

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title>smartcab project answers</title>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/require.js/2.1.10/require.min.js"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
```

```
<style type="text/css">
```

```
  /*!
```

```
  *
```

```
  * Twitter Bootstrap
```

```
  *
```

```
  /*! normalize.css v3.0.2 | MIT License | git.io/normalize */html{font-family:sans-serif;-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}body{margin:0}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inline-block;vertical-align:baseline}audio:not([controls]){display:none;height:0}[hidden],template{display:none}a{background-color:transparent}a:active,a:hover{outline:0}abbr[title]{border-bottom:1px dotted}b,strong{font-weight:bold}dfn{font-style:italic}h1{font-size:2em;margin:.67em 0}mark{background:#ff0;color:#000}small{font-size:80%}sub,sup{font-size:75%;line-height:0;position:relative;vertical-align:baseline}sup{top:-0.5em}sub{bottom:-0.25em}img{border:0}svg:not(:root){overflow:hidden}figure{margin:1em 40px}hr{-moz-box-sizing:content-box;box-sizing:content-box;height:0}pre{overflow:auto}code,kbd,pre,samp{font-family:monospace,monospace;font-size:1em}button,input,optgroup,select,textarea{color:inherit;font:inherit;margin:0}button{overflow:visible}button,select{text-transform:none}button,html input[disabled]{cursor:default}button::-moz-focus-inner,input::-moz-focus-inner{border:0;padding:0}input{line-height:normal}input[type="checkbox"],input[type="radio"]{box-sizing:border-box;padding:0}input[type="number"]::-webkit-inner-spin-button,input[type="number"]::-webkit-outer-spin-button{height:auto}input[type="search"]{-webkit-appearance:textfield;-moz-box-sizing:content-box;-webkit-box-sizing:content-box;box-sizing:content-box}input[type="search"]::-webkit-search-cancel-
```

```
button,input[type="search"]::-webkit-search-decoration{-webkit-appearance:none}fieldset{border:1px
solid #c0c0c0;margin:0 2px;padding:.35em .625em
.75em}legend{border:0;padding:0}textarea{overflow:auto}optgroup{font-weight:bold}table{border-
collapse:collapse;border-spacing:0}td,th{padding:0}/*! Source: https://github.com/h5bp/html5-
boilerplate/blob/master/src/css/main.css */@media print{*,*:before,*:after{background:transparent
!important;color:#000 !important;box-shadow:none !important;text-shadow:none
!important}a,a:visited{text-decoration:underline}a[href]:after{content:" (" attr(href)
")"}abbr[title]:after{content:" (" attr(title)
")"}a[href^="#"]:after,a[href^="javascript:"]:after{content:""}pre,blockquote{border:1px solid
#999;page-break-inside:avoid}thead{display:table-header-group}tr,img{page-break-
inside:avoid}img{max-width:100% !important}p,h2,h3{orphans:3;widows:3}h2,h3{page-break-
after:avoid}select{background:#fff
!important}.navbar{display:none}.btn>.caret,.dropdown>.btn>.caret{border-top-color:#000
!important}.label{border:1px solid #000}.table{border-collapse:collapse !important}.table td,.table
th{background-color:#fff !important}.table-bordered th,.table-bordered td{border:1px solid #ddd
!important}}@font-face{font-family:'Glyphicons Halflings';src:url('../fonts/glyphicons-halflings-
regular.eot');src:url('../fonts/glyphicons-halflings-regular.eot?#iefix') format('embedded-
opentype'),url('../fonts/glyphicons-halflings-regular.woff') format('woff'),url('../fonts/glyphicons-
halflings-regular.ttf') format('truetype'),url('../fonts/glyphicons-halflings-
regular.svg#glyphicons_halfingsregular')
format('svg')}.glyphicon{position:relative;top:1px;display:inline-block;font-family:'Glyphicons
Halflings';font-style:normal;font-weight:normal;line-height:1;-webkit-font-smoothing:antialiased;-moz-
osx-font-smoothing:grayscale}.glyphicon-asterisk:before{content:"\2a"}.glyphicon-
plus:before{content:"\2b"}.glyphicon-euro:before,.glyphicon-eur:before{content:"\20ac"}.glyphicon-
minus:before{content:"\2212"}.glyphicon-cloud:before{content:"\2601"}.glyphicon-
envelope:before{content:"\2709"}.glyphicon-pencil:before{content:"\270f"}.glyphicon-
glass:before{content:"\e001"}.glyphicon-music:before{content:"\e002"}.glyphicon-
search:before{content:"\e003"}.glyphicon-heart:before{content:"\e005"}.glyphicon-
star:before{content:"\e006"}.glyphicon-star-empty:before{content:"\e007"}.glyphicon-
user:before{content:"\e008"}.glyphicon-film:before{content:"\e009"}.glyphicon-th-
large:before{content:"\e010"}.glyphicon-th:before{content:"\e011"}.glyphicon-th-
list:before{content:"\e012"}.glyphicon-ok:before{content:"\e013"}.glyphicon-
remove:before{content:"\e014"}.glyphicon-zoom-in:before{content:"\e015"}.glyphicon-zoom-
out:before{content:"\e016"}.glyphicon-off:before{content:"\e017"}.glyphicon-
signal:before{content:"\e018"}.glyphicon-cog:before{content:"\e019"}.glyphicon-
trash:before{content:"\e020"}.glyphicon-home:before{content:"\e021"}.glyphicon-
file:before{content:"\e022"}.glyphicon-time:before{content:"\e023"}.glyphicon-
road:before{content:"\e024"}.glyphicon-download-alt:before{content:"\e025"}.glyphicon-
download:before{content:"\e026"}.glyphicon-upload:before{content:"\e027"}.glyphicon-
inbox:before{content:"\e028"}.glyphicon-play-circle:before{content:"\e029"}.glyphicon-
repeat:before{content:"\e030"}.glyphicon-refresh:before{content:"\e031"}.glyphicon-list-
alt:before{content:"\e032"}.glyphicon-lock:before{content:"\e033"}.glyphicon-
flag:before{content:"\e034"}.glyphicon-headphones:before{content:"\e035"}.glyphicon-volume-
off:before{content:"\e036"}.glyphicon-volume-down:before{content:"\e037"}.glyphicon-volume-
```

up:before{content:"\e038"}.glyphicon-qr-code:before{content:"\e039"}.glyphicon-barcode:before{content:"\e040"}.glyphicon-tag:before{content:"\e041"}.glyphicon-tags:before{content:"\e042"}.glyphicon-book:before{content:"\e043"}.glyphicon-bookmark:before{content:"\e044"}.glyphicon-print:before{content:"\e045"}.glyphicon-camera:before{content:"\e046"}.glyphicon-font:before{content:"\e047"}.glyphicon-bold:before{content:"\e048"}.glyphicon-italic:before{content:"\e049"}.glyphicon-text-height:before{content:"\e050"}.glyphicon-text-width:before{content:"\e051"}.glyphicon-align-left:before{content:"\e052"}.glyphicon-align-center:before{content:"\e053"}.glyphicon-align-right:before{content:"\e054"}.glyphicon-align-justify:before{content:"\e055"}.glyphicon-list:before{content:"\e056"}.glyphicon-indent-left:before{content:"\e057"}.glyphicon-indent-right:before{content:"\e058"}.glyphicon-facetime-video:before{content:"\e059"}.glyphicon-picture:before{content:"\e060"}.glyphicon-map-marker:before{content:"\e062"}.glyphicon-adjust:before{content:"\e063"}.glyphicon-tint:before{content:"\e064"}.glyphicon-edit:before{content:"\e065"}.glyphicon-share:before{content:"\e066"}.glyphicon-check:before{content:"\e067"}.glyphicon-move:before{content:"\e068"}.glyphicon-step-backward:before{content:"\e069"}.glyphicon-fast-backward:before{content:"\e070"}.glyphicon-backward:before{content:"\e071"}.glyphicon-play:before{content:"\e072"}.glyphicon-pause:before{content:"\e073"}.glyphicon-stop:before{content:"\e074"}.glyphicon-forward:before{content:"\e075"}.glyphicon-fast-forward:before{content:"\e076"}.glyphicon-step-forward:before{content:"\e077"}.glyphicon-eject:before{content:"\e078"}.glyphicon-chevron-left:before{content:"\e079"}.glyphicon-chevron-right:before{content:"\e080"}.glyphicon-plus-sign:before{content:"\e081"}.glyphicon-minus-sign:before{content:"\e082"}.glyphicon-remove-sign:before{content:"\e083"}.glyphicon-ok-sign:before{content:"\e084"}.glyphicon-question-sign:before{content:"\e085"}.glyphicon-info-sign:before{content:"\e086"}.glyphicon-screenshot:before{content:"\e087"}.glyphicon-remove-circle:before{content:"\e088"}.glyphicon-ok-circle:before{content:"\e089"}.glyphicon-ban-circle:before{content:"\e090"}.glyphicon-arrow-left:before{content:"\e091"}.glyphicon-arrow-right:before{content:"\e092"}.glyphicon-arrow-up:before{content:"\e093"}.glyphicon-arrow-down:before{content:"\e094"}.glyphicon-share-alt:before{content:"\e095"}.glyphicon-resize-full:before{content:"\e096"}.glyphicon-resize-small:before{content:"\e097"}.glyphicon-exclamation-sign:before{content:"\e101"}.glyphicon-gift:before{content:"\e102"}.glyphicon-leaf:before{content:"\e103"}.glyphicon-fire:before{content:"\e104"}.glyphicon-eye-open:before{content:"\e105"}.glyphicon-eye-close:before{content:"\e106"}.glyphicon-warning-sign:before{content:"\e107"}.glyphicon-plane:before{content:"\e108"}.glyphicon-calendar:before{content:"\e109"}.glyphicon-random:before{content:"\e110"}.glyphicon-comment:before{content:"\e111"}.glyphicon-magnet:before{content:"\e112"}.glyphicon-chevron-up:before{content:"\e113"}.glyphicon-chevron-down:before{content:"\e114"}.glyphicon-retweet:before{content:"\e115"}.glyphicon-shopping-cart:before{content:"\e116"}.glyphicon-folder-close:before{content:"\e117"}.glyphicon-folder-open:before{content:"\e118"}.glyphicon-resize-vertical:before{content:"\e119"}.glyphicon-resize-horizontal:before{content:"\e120"}.glyphicon-hdd:before{content:"\e121"}.glyphicon-bullhorn:before{content:"\e122"}.glyphicon-bell:before{content:"\e123"}.glyphicon-certificate:before{content:"\e124"}.glyphicon-thumbs-up:before{content:"\e125"}.glyphicon-thumbs-down:before{content:"\e126"}.glyphicon-hand-right:before{content:"\e127"}.glyphicon-hand-left:before{content:"\e128"}.glyphicon-hand-up:before{content:"\e129"}.glyphicon-hand-

down:before{content:"\e130"}.glyphicon-circle-arrow-right:before{content:"\e131"}.glyphicon-circle-arrow-left:before{content:"\e132"}.glyphicon-circle-arrow-up:before{content:"\e133"}.glyphicon-circle-arrow-down:before{content:"\e134"}.glyphicon-globe:before{content:"\e135"}.glyphicon-wrench:before{content:"\e136"}.glyphicon-tasks:before{content:"\e137"}.glyphicon-filter:before{content:"\e138"}.glyphicon-briefcase:before{content:"\e139"}.glyphicon-fullscreen:before{content:"\e140"}.glyphicon-dashboard:before{content:"\e141"}.glyphicon-paperclip:before{content:"\e142"}.glyphicon-heart-empty:before{content:"\e143"}.glyphicon-link:before{content:"\e144"}.glyphicon-phone:before{content:"\e145"}.glyphicon-pushpin:before{content:"\e146"}.glyphicon-usd:before{content:"\e148"}.glyphicon-gbp:before{content:"\e149"}.glyphicon-sort:before{content:"\e150"}.glyphicon-sort-by-alphabet:before{content:"\e151"}.glyphicon-sort-by-alphabet-alt:before{content:"\e152"}.glyphicon-sort-by-order:before{content:"\e153"}.glyphicon-sort-by-order-alt:before{content:"\e154"}.glyphicon-sort-by-attributes:before{content:"\e155"}.glyphicon-sort-by-attributes-alt:before{content:"\e156"}.glyphicon-unchecked:before{content:"\e157"}.glyphicon-expand:before{content:"\e158"}.glyphicon-collapse-down:before{content:"\e159"}.glyphicon-collapse-up:before{content:"\e160"}.glyphicon-log-in:before{content:"\e161"}.glyphicon-flash:before{content:"\e162"}.glyphicon-log-out:before{content:"\e163"}.glyphicon-new-window:before{content:"\e164"}.glyphicon-record:before{content:"\e165"}.glyphicon-save:before{content:"\e166"}.glyphicon-open:before{content:"\e167"}.glyphicon-saved:before{content:"\e168"}.glyphicon-import:before{content:"\e169"}.glyphicon-export:before{content:"\e170"}.glyphicon-send:before{content:"\e171"}.glyphicon-floppy-disk:before{content:"\e172"}.glyphicon-floppy-saved:before{content:"\e173"}.glyphicon-floppy-remove:before{content:"\e174"}.glyphicon-floppy-save:before{content:"\e175"}.glyphicon-floppy-open:before{content:"\e176"}.glyphicon-credit-card:before{content:"\e177"}.glyphicon-transfer:before{content:"\e178"}.glyphicon-cutlery:before{content:"\e179"}.glyphicon-header:before{content:"\e180"}.glyphicon-compressed:before{content:"\e181"}.glyphicon-earphone:before{content:"\e182"}.glyphicon-phone-alt:before{content:"\e183"}.glyphicon-tower:before{content:"\e184"}.glyphicon-stats:before{content:"\e185"}.glyphicon-sd-video:before{content:"\e186"}.glyphicon-hd-video:before{content:"\e187"}.glyphicon-subtitles:before{content:"\e188"}.glyphicon-sound-stereo:before{content:"\e189"}.glyphicon-sound-dolby:before{content:"\e190"}.glyphicon-sound-5-1:before{content:"\e191"}.glyphicon-sound-6-1:before{content:"\e192"}.glyphicon-sound-7-1:before{content:"\e193"}.glyphicon-copyright-mark:before{content:"\e194"}.glyphicon-registration-mark:before{content:"\e195"}.glyphicon-cloud-download:before{content:"\e197"}.glyphicon-cloud-upload:before{content:"\e198"}.glyphicon-tree-conifer:before{content:"\e199"}.glyphicon-tree-deciduous:before{content:"\e200"}\*{-webkit-box-sizing:border-box;-moz-box-sizing:border-box;box-sizing:border-box}\*:before,\*:after{-webkit-box-sizing:border-box;-moz-box-sizing:border-box;box-sizing:border-box}html{font-size:10px;-webkit-tap-highlight-color:rgba(0,0,0,0)}body{font-family:"Helvetica Neue",Helvetica,Arial,sans-serif;font-size:13px;line-height:1.42857143;color:#000;background-color:#fff}input,button,select,textarea{font-family:inherit;font-size:inherit;line-height:inherit}a{color:#337ab7;text-decoration:none}a:hover,a:focus{color:#23527c;text-decoration:underline}a:focus{outline:thin dotted;outline:5px auto -webkit-focus-ring-color;outline-offset:-2px}figure{margin:0}img{vertical-align:middle}.img-responsive,.thumbnail>img,.carousel-inner>.item>img,.carousel-inner>.item>a>img{display:block;max-width:100%;height:auto}.img-rounded{border-radius:3px}.img-

thumbnail{padding:4px;line-height:1.42857143;background-color:#fff;border:1px solid #ddd;border-radius:2px;-webkit-transition:all .2s ease-in-out;-o-transition:all .2s ease-in-out;transition:all .2s ease-in-out;display:inline-block;max-width:100%;height:auto}.img-circle{border-radius:50%}hr{margin-top:18px;margin-bottom:18px;border:0;border-top:1px solid #eee}.sr-only{position:absolute;width:1px;height:1px;margin:-1px;padding:0;overflow:hidden;clip:rect(0, 0, 0, 0);border:0}.sr-only-focusable:active,.sr-only-focusable:focus{position:static;width:auto;height:auto;margin:0;overflow:visible;clip:auto}h1,h2,h3,h4,h5,h6,.h1,.h2,.h3,.h4,.h5,.h6{font-family:inherit;font-weight:500;line-height:1.1;color:inherit}h1 small,h2 small,h3 small,h4 small,h5 small,h6 small,.h1 small,.h2 small,.h3 small,.h4 small,.h5 small,.h6 small,h1 .small,h2 .small,h3 .small,h4 .small,h5 .small,h6 .small,.h1 .small,.h2 .small,.h3 .small,.h4 .small,.h5 .small,.h6 .small{font-weight:normal;line-height:1;color:#777}h1,.h1,h2,.h2,h3,.h3{margin-top:18px;margin-bottom:9px}h1 small,.h1 small,h2 small,.h2 small,h3 small,.h3 small .small,.h1 .small,h2 .small,.h2 .small,h3 .small,.h3 .small{font-size:65%}h4,.h4,h5,.h5,h6,.h6{margin-top:9px;margin-bottom:9px}h4 small,.h4 small,h5 small,.h5 small,h6 small,.h6 small,h4 .small,.h4 .small,h5 .small,.h5 .small,h6 .small,.h6 .small{font-size:75%}h1,.h1{font-size:33px}h2,.h2{font-size:27px}h3,.h3{font-size:23px}h4,.h4{font-size:17px}h5,.h5{font-size:13px}h6,.h6{font-size:12px}p{margin:0 0 9px}.lead{margin-bottom:18px;font-size:14px;font-weight:300;line-height:1.4}@media (min-width:768px){.lead{font-size:19.5px}}small,.small{font-size:92%}mark,.mark{background-color:#fcf8e3;padding:.2em}.text-left{text-align:left}.text-right{text-align:right}.text-center{text-align:center}.text-justify{text-align:justify}.text-nowrap{white-space:nowrap}.text-lowercase{text-transform:lowercase}.text-uppercase{text-transform:uppercase}.text-capitalize{text-transform:capitalize}.text-muted{color:#777}.text-primary{color:#337ab7}a.text-primary:hover{color:#286090}.text-success{color:#3c763d}a.text-success:hover{color:#2b542c}.text-info{color:#31708f}a.text-info:hover{color:#245269}.text-warning{color:#8a6d3b}a.text-warning:hover{color:#66512c}.text-danger{color:#a94442}a.text-danger:hover{color:#843534}.bg-primary{color:#fff;background-color:#337ab7}a.bg-primary:hover{background-color:#286090}.bg-success{background-color:#dff0d8}a.bg-success:hover{background-color:#c1e2b3}.bg-info{background-color:#d9edf7}a.bg-info:hover{background-color:#afd9ee}.bg-warning{background-color:#fcf8e3}a.bg-warning:hover{background-color:#f7ecb5}.bg-danger{background-color:#f2dede}a.bg-danger:hover{background-color:#e4b9b9}.page-header{padding-bottom:8px;margin:36px 0 18px;border-bottom:1px solid #eee}ul,ol{margin-top:0;margin-bottom:9px}ul ul,ol ol,ol ol ol{margin-bottom:0}.list-unstyled{padding-left:0;list-style:none}.list-inline{padding-left:0;list-style:none;margin-left:-5px}.list-inline>li{display:inline-block;padding-left:5px;padding-right:5px}dl{margin-top:0;margin-bottom:18px}dt,dd{line-height:1.42857143}dt{font-weight:bold}dd{margin-left:0}@media (min-width:541px){.dl-horizontal dt{float:left;width:160px;clear:left;text-align:right;overflow:hidden;text-overflow:ellipsis;white-space:nowrap}.dl-horizontal dd{margin-left:180px}}abbr[title],abbr[data-original-title]{cursor:help;border-bottom:1px dotted #777}.initialism{font-size:90%;text-transform:uppercase}blockquote{padding:9px 18px;margin:0 0 18px;font-size:inherit;border-left:5px solid #eee}blockquote p:last-child,blockquote ul:last-child,blockquote ol:last-child{margin-bottom:0}blockquote footer,blockquote small,blockquote .small{display:block;font-size:80%;line-height:1.42857143;color:#777}blockquote footer:before,blockquote small:before,blockquote .small:before{content:'\2014 \00A0'}.blockquote-reverse,blockquote.pull-right{padding-right:15px;padding-left:0;border-right:5px solid #eee;border-left:0;text-align:right}.blockquote-reverse

footer:before,blockquote.pull-right footer:before,.blockquote-reverse small:before,blockquote.pull-right small:before,.blockquote-reverse .small:before,blockquote.pull-right .small:before{content:''}.blockquote-reverse footer:after,blockquote.pull-right footer:after,.blockquote-reverse small:after,blockquote.pull-right small:after,.blockquote-reverse .small:after,blockquote.pull-right .small:after{content:''}\00A0 \2014'}address{margin-bottom:18px;font-style:normal;line-height:1.42857143}code,kbd,pre,samp{font-family:monospace}code{padding:2px 4px;font-size:90%;color:#c7254e;background-color:#f9f2f4;border-radius:2px}kbd{padding:2px 4px;font-size:90%;color:#fff;background-color:#333;border-radius:1px;box-shadow:inset 0 -1px 0 rgba(0,0,0,0.25)}kbd kbd{padding:0;font-size:100%;font-weight:bold;box-shadow:none}pre{display:block;padding:8.5px;margin:0 0 9px;font-size:12px;line-height:1.42857143;word-break:break-all;word-wrap:break-word;color:#333;background-color:#f5f5f5;border:1px solid #ccc;border-radius:2px}pre code{padding:0;font-size:inherit;color:inherit;white-space:pre-wrap;background-color:transparent;border-radius:0}.pre-scrollable{max-height:340px;overflow-y:scroll}.container{margin-right:auto;margin-left:auto;padding-left:0;padding-right:0}@media (min-width:768px){.container{width:768px}}@media (min-width:992px){.container{width:940px}}@media (min-width:1200px){.container{width:1140px}}.container-fluid{margin-right:auto;margin-left:auto;padding-left:0;padding-right:0}.row{margin-left:0;margin-right:0}.col-xs-1,.col-sm-1,.col-md-1,.col-lg-1,.col-xs-2,.col-sm-2,.col-md-2,.col-lg-2,.col-xs-3,.col-sm-3,.col-md-3,.col-lg-3,.col-xs-4,.col-sm-4,.col-md-4,.col-lg-4,.col-xs-5,.col-sm-5,.col-md-5,.col-lg-5,.col-xs-6,.col-sm-6,.col-md-6,.col-lg-6,.col-xs-7,.col-sm-7,.col-md-7,.col-lg-7,.col-xs-8,.col-sm-8,.col-md-8,.col-lg-8,.col-xs-9,.col-sm-9,.col-md-9,.col-lg-9,.col-xs-10,.col-sm-10,.col-md-10,.col-lg-10,.col-xs-11,.col-sm-11,.col-md-11,.col-lg-11,.col-xs-12,.col-sm-12,.col-md-12,.col-lg-12{position:relative;min-height:1px;padding-left:0;padding-right:0}.col-xs-1,.col-xs-2,.col-xs-3,.col-xs-4,.col-xs-5,.col-xs-6,.col-xs-7,.col-xs-8,.col-xs-9,.col-xs-10,.col-xs-11,.col-xs-12{float:left}.col-xs-12{width:100%}.col-xs-11{width:91.66666667%}.col-xs-10{width:83.33333333%}.col-xs-9{width:75%}.col-xs-8{width:66.66666667%}.col-xs-7{width:58.33333333%}.col-xs-6{width:50%}.col-xs-5{width:41.66666667%}.col-xs-4{width:33.33333333%}.col-xs-3{width:25%}.col-xs-2{width:16.66666667%}.col-xs-1{width:8.33333333%}.col-xs-pull-12{right:100%}.col-xs-pull-11{right:91.66666667%}.col-xs-pull-10{right:83.33333333%}.col-xs-pull-9{right:75%}.col-xs-pull-8{right:66.66666667%}.col-xs-pull-7{right:58.33333333%}.col-xs-pull-6{right:50%}.col-xs-pull-5{right:41.66666667%}.col-xs-pull-4{right:33.33333333%}.col-xs-pull-3{right:25%}.col-xs-pull-2{right:16.66666667%}.col-xs-pull-1{right:8.33333333%}.col-xs-pull-0{right:auto}.col-xs-push-12{left:100%}.col-xs-push-11{left:91.66666667%}.col-xs-push-10{left:83.33333333%}.col-xs-push-9{left:75%}.col-xs-push-8{left:66.66666667%}.col-xs-push-7{left:58.33333333%}.col-xs-push-6{left:50%}.col-xs-push-5{left:41.66666667%}.col-xs-push-4{left:33.33333333%}.col-xs-push-3{left:25%}.col-xs-push-2{left:16.66666667%}.col-xs-push-1{left:8.33333333%}.col-xs-push-0{left:auto}.col-xs-offset-12{margin-left:100%}.col-xs-offset-11{margin-left:91.66666667%}.col-xs-offset-10{margin-left:83.33333333%}.col-xs-offset-9{margin-left:75%}.col-xs-offset-8{margin-left:66.66666667%}.col-xs-offset-7{margin-left:58.33333333%}.col-xs-offset-6{margin-left:50%}.col-xs-offset-5{margin-left:41.66666667%}.col-xs-offset-4{margin-left:33.33333333%}.col-xs-offset-3{margin-left:25%}.col-xs-offset-2{margin-left:16.66666667%}.col-xs-offset-1{margin-left:8.33333333%}.col-xs-offset-0{margin-left:0}@media (min-width:768px){.col-sm-1,.col-sm-2,.col-sm-3,.col-sm-4,.col-sm-5,.col-sm-6,.col-sm-7,.col-sm-8,.col-sm-9,.col-sm-10,.col-sm-11,.col-sm-12{float:left}.col-sm-12{width:100%}.col-sm-

11{width:91.66666667%}.col-sm-10{width:83.33333333%}.col-sm-9{width:75%}.col-sm-8{width:66.66666667%}.col-sm-7{width:58.33333333%}.col-sm-6{width:50%}.col-sm-5{width:41.66666667%}.col-sm-4{width:33.33333333%}.col-sm-3{width:25%}.col-sm-2{width:16.66666667%}.col-sm-1{width:8.33333333%}.col-sm-pull-12{right:100%}.col-sm-pull-11{right:91.66666667%}.col-sm-pull-10{right:83.33333333%}.col-sm-pull-9{right:75%}.col-sm-pull-8{right:66.66666667%}.col-sm-pull-7{right:58.33333333%}.col-sm-pull-6{right:50%}.col-sm-pull-5{right:41.66666667%}.col-sm-pull-4{right:33.33333333%}.col-sm-pull-3{right:25%}.col-sm-pull-2{right:16.66666667%}.col-sm-pull-1{right:8.33333333%}.col-sm-pull-0{right:auto}.col-sm-push-12{left:100%}.col-sm-push-11{left:91.66666667%}.col-sm-push-10{left:83.33333333%}.col-sm-push-9{left:75%}.col-sm-push-8{left:66.66666667%}.col-sm-push-7{left:58.33333333%}.col-sm-push-6{left:50%}.col-sm-push-5{left:41.66666667%}.col-sm-push-4{left:33.33333333%}.col-sm-push-3{left:25%}.col-sm-push-2{left:16.66666667%}.col-sm-push-1{left:8.33333333%}.col-sm-push-0{left:auto}.col-sm-offset-12{margin-left:100%}.col-sm-offset-11{margin-left:91.66666667%}.col-sm-offset-10{margin-left:83.33333333%}.col-sm-offset-9{margin-left:75%}.col-sm-offset-8{margin-left:66.66666667%}.col-sm-offset-7{margin-left:58.33333333%}.col-sm-offset-6{margin-left:50%}.col-sm-offset-5{margin-left:41.66666667%}.col-sm-offset-4{margin-left:33.33333333%}.col-sm-offset-3{margin-left:25%}.col-sm-offset-2{margin-left:16.66666667%}.col-sm-offset-1{margin-left:8.33333333%}.col-sm-offset-0{margin-left:0}}@media (min-width:992px){.col-md-1,.col-md-2,.col-md-3,.col-md-4,.col-md-5,.col-md-6,.col-md-7,.col-md-8,.col-md-9,.col-md-10,.col-md-11,.col-md-12{float:left}.col-md-12{width:100%}.col-md-11{width:91.66666667%}.col-md-10{width:83.33333333%}.col-md-9{width:75%}.col-md-8{width:66.66666667%}.col-md-7{width:58.33333333%}.col-md-6{width:50%}.col-md-5{width:41.66666667%}.col-md-4{width:33.33333333%}.col-md-3{width:25%}.col-md-2{width:16.66666667%}.col-md-1{width:8.33333333%}.col-md-pull-12{right:100%}.col-md-pull-11{right:91.66666667%}.col-md-pull-10{right:83.33333333%}.col-md-pull-9{right:75%}.col-md-pull-8{right:66.66666667%}.col-md-pull-7{right:58.33333333%}.col-md-pull-6{right:50%}.col-md-pull-5{right:41.66666667%}.col-md-pull-4{right:33.33333333%}.col-md-pull-3{right:25%}.col-md-pull-2{right:16.66666667%}.col-md-pull-1{right:8.33333333%}.col-md-pull-0{right:auto}.col-md-push-12{left:100%}.col-md-push-11{left:91.66666667%}.col-md-push-10{left:83.33333333%}.col-md-push-9{left:75%}.col-md-push-8{left:66.66666667%}.col-md-push-7{left:58.33333333%}.col-md-push-6{left:50%}.col-md-push-5{left:41.66666667%}.col-md-push-4{left:33.33333333%}.col-md-push-3{left:25%}.col-md-push-2{left:16.66666667%}.col-md-push-1{left:8.33333333%}.col-md-push-0{left:auto}.col-md-offset-12{margin-left:100%}.col-md-offset-11{margin-left:91.66666667%}.col-md-offset-10{margin-left:83.33333333%}.col-md-offset-9{margin-left:75%}.col-md-offset-8{margin-left:66.66666667%}.col-md-offset-7{margin-left:58.33333333%}.col-md-offset-6{margin-left:50%}.col-md-offset-5{margin-left:41.66666667%}.col-md-offset-4{margin-left:33.33333333%}.col-md-offset-3{margin-left:25%}.col-md-offset-2{margin-left:16.66666667%}.col-md-offset-1{margin-left:8.33333333%}.col-md-offset-0{margin-left:0}}@media (min-width:1200px){.col-lg-1,.col-lg-2,.col-lg-3,.col-lg-4,.col-lg-5,.col-lg-6,.col-lg-7,.col-lg-8,.col-lg-9,.col-lg-10,.col-lg-11,.col-lg-12{float:left}.col-lg-12{width:100%}.col-lg-11{width:91.66666667%}.col-lg-10{width:83.33333333%}.col-lg-9{width:75%}.col-lg-8{width:66.66666667%}.col-lg-7{width:58.33333333%}.col-lg-6{width:50%}.col-lg-5{width:41.66666667%}.col-lg-4{width:33.33333333%}.col-lg-3{width:25%}.col-lg-2{width:16.66666667%}.col-lg-1{width:8.33333333%}.col-lg-pull-12{right:100%}.col-lg-pull-11{right:91.66666667%}.col-lg-pull-10{right:83.33333333%}.col-lg-pull-9{right:75%}.col-lg-pull-

8{right:66.66666667%}.col-lg-pull-7{right:58.33333333%}.col-lg-pull-6{right:50%}.col-lg-pull-5{right:41.66666667%}.col-lg-pull-4{right:33.33333333%}.col-lg-pull-3{right:25%}.col-lg-pull-2{right:16.66666667%}.col-lg-pull-1{right:8.33333333%}.col-lg-pull-0{right:auto}.col-lg-push-12{left:100%}.col-lg-push-11{left:91.66666667%}.col-lg-push-10{left:83.33333333%}.col-lg-push-9{left:75%}.col-lg-push-8{left:66.66666667%}.col-lg-push-7{left:58.33333333%}.col-lg-push-6{left:50%}.col-lg-push-5{left:41.66666667%}.col-lg-push-4{left:33.33333333%}.col-lg-push-3{left:25%}.col-lg-push-2{left:16.66666667%}.col-lg-push-1{left:8.33333333%}.col-lg-push-0{left:auto}.col-lg-offset-12{margin-left:100%}.col-lg-offset-11{margin-left:91.66666667%}.col-lg-offset-10{margin-left:83.33333333%}.col-lg-offset-9{margin-left:75%}.col-lg-offset-8{margin-left:66.66666667%}.col-lg-offset-7{margin-left:58.33333333%}.col-lg-offset-6{margin-left:50%}.col-lg-offset-5{margin-left:41.66666667%}.col-lg-offset-4{margin-left:33.33333333%}.col-lg-offset-3{margin-left:25%}.col-lg-offset-2{margin-left:16.66666667%}.col-lg-offset-1{margin-left:8.33333333%}.col-lg-offset-0{margin-left:0}}table{background-color:transparent}caption{padding-top:8px;padding-bottom:8px;color:#777;text-align:left}th{text-align:left}.table{width:100%;max-width:100%;margin-bottom:18px}.table>thead>tr>th,.table>tbody>tr>th,.table>tfoot>tr>th,.table>thead>tr>td,.table>tbody>tr>td,.table>tfoot>tr>td{padding:8px;line-height:1.42857143;vertical-align:top;border-top:1px solid #ddd}.table>thead>tr>th{vertical-align:bottom;border-bottom:2px solid #ddd}.table>caption+thead>tr:first-child>th,.table>colgroup+thead>tr:first-child>th,.table>thead:first-child>tr:first-child>th,.table>caption+thead>tr:first-child>td,.table>colgroup+thead>tr:first-child>td,.table>thead:first-child>tr:first-child>td{border-top:0}.table>tbody+tbody{border-top:2px solid #ddd}.table .table{background-color:#fff}.table-condensed>thead>tr>th,.table-condensed>tbody>tr>th,.table-condensed>tfoot>tr>th,.table-condensed>thead>tr>td,.table-condensed>tbody>tr>td,.table-condensed>tfoot>tr>td{padding:5px}.table-bordered{border:1px solid #ddd}.table-bordered>thead>tr>th,.table-bordered>tbody>tr>th,.table-bordered>tfoot>tr>th,.table-bordered>thead>tr>td,.table-bordered>tbody>tr>td,.table-bordered>tfoot>tr>td{border:1px solid #ddd}.table-bordered>thead>tr>th,.table-bordered>thead>tr>td{border-bottom-width:2px}.table-striped>tbody>tr:nth-child(odd){background-color:#f9f9f9}.table-hover>tbody>tr: hover{background-color:#f5f5f5}table col[class\*="col-"]{position:static;float:none;display:table-column}table td[class\*="col-"],table th[class\*="col-"]{position:static;float:none;display:table-cell}.table>thead>tr>td.active,.table>tbody>tr>td.active,.table>tfoot>tr>td.active,.table>thead>tr>th.active,.table>tbody>tr>th.active,.table>tfoot>tr>th.active,.table>thead>tr>td.active>td,.table>tbody>tr>td.active>td,.table>tfoot>tr>td.active>td,.table>thead>tr>th.active>th,.table>tbody>tr>th.active>th,.table>tfoot>tr>th.active>th{background-color:#f5f5f5}.table-hover>tbody>tr>td.active: hover,.table-hover>tbody>tr>th.active: hover,.table-hover>tbody>tr>td.active: hover>td,.table-hover>tbody>tr>th.active: hover>td,.table-hover>tbody>tr>th.active: hover>td,.table-hover>tbody>tr>td.active: hover>th{background-color:#e8e8e8}.table>thead>tr>td.success,.table>tbody>tr>td.success,.table>tfoot>tr>td.success,.table>thead>tr>th.success,.table>tbody>tr>th.success,.table>tfoot>tr>th.success,.table>thead>tr>td.success>td,.table>tbody>tr>td.success>td,.table>tfoot>tr>td.success>td,.table>thead>tr>th.success>th,.table>tbody>tr>th.success>th,.table>tfoot>tr>th.success>th{background-color:#dff0d8}.table-hover>tbody>tr>td.success: hover,.table-hover>tbody>tr>th.success: hover,.table-hover>tbody>tr>td.success: hover>td,.table-hover>tbody>tr>th.success: hover>td,.table-hover>tbody>tr>th.success: hover>td{background-color:#d0e9c6}.table>thead>tr>td.info,.table>tbody>tr>td.info,.table>tfoot>tr>td.info,.table>thead>tr>th.info,.table>tbody>tr>th.info,.table>tfoot>tr>th.info,.table>thead>tr>td.info>td,.table>tbody>tr>td.info>td,.table>tfoot>tr>td.info>td,.table>thead>tr>th.info>th,.table>tbody>tr>th.info>th,.table>tfoot>tr>th.info>th{background-color:#d9d9d9}



```

able>tfoot>tr.info>td,.table>thead>tr.info>th,.table>tbody>tr.info>th,.table>tfoot>tr.info>th{background-color:#d9edf7}.table-hover>tbody>tr>td.info:hover,.table-hover>tbody>tr>th.info:hover,.table-hover>tbody>tr.info:hover>td,.table-hover>tbody>tr>tr.info>th{background-color:#c4e3f3}.table>thead>tr>td.warning,.table>tbody>tr>td.warning,.table>tfoot>tr>td.warning,.table>thead>tr>th.warning,.table>tbody>tr>th.warning,.table>tfoot>tr>th.warning,.table>thead>tr.warning>td,.table>tbody>tr.warning>td,.table>tfoot>tr.warning>td,.table>thead>tr.warning>th,.table>tbody>tr.warning>th,.table>tfoot>tr.warning>th{background-color:#fcf8e3}.table-hover>tbody>tr>td.warning:hover,.table-hover>tbody>tr>th.warning:hover,.table-hover>tbody>tr.warning:hover>td,.table-hover>tbody>tr>tr.warning>th{background-color:#faf2cc}.table>thead>tr>td.danger,.table>tbody>tr>td.danger,.table>tfoot>tr>td.danger,.table>thead>tr>th.danger,.table>tbody>tr>th.danger,.table>tfoot>tr>th.danger,.table>thead>tr.danger>td,.table>tbody>tr.danger>td,.table>tfoot>tr.danger>td,.table>thead>tr.danger>th,.table>tbody>tr.danger>th,.table>tfoot>tr.danger>th{background-color:#f2dede}.table-hover>tbody>tr>td.danger:hover,.table-hover>tbody>tr>th.danger:hover,.table-hover>tbody>tr.danger:hover>td,.table-hover>tbody>tr>tr.danger>th{background-color:#ebcccc}.table-responsive{overflow-x:auto;min-height:.01%}@media screen and (max-width:767px){.table-responsive{width:100%;margin-bottom:13.5px;overflow-y:hidden;-ms-overflow-style:-ms-autohiding-scrollbar;border:1px solid #ddd}.table-responsive>.table{margin-bottom:0}.table-responsive>.table>thead>tr>th,.table-responsive>.table>tbody>tr>th,.table-responsive>.table>tfoot>tr>th,.table-responsive>.table>thead>tr>td,.table-responsive>.table>tbody>tr>td,.table-responsive>.table>tfoot>tr>td{white-space:nowrap}.table-responsive>.table-bordered{border:0}.table-responsive>.table-bordered>thead>tr>th:first-child,.table-responsive>.table-bordered>tbody>tr>th:first-child,.table-responsive>.table-bordered>tfoot>tr>th:first-child,.table-responsive>.table-bordered>thead>tr>td:first-child,.table-responsive>.table-bordered>tbody>tr>td:first-child,.table-responsive>.table-bordered>tfoot>tr>td:first-child{border-left:0}.table-responsive>.table-bordered>thead>tr>th:last-child,.table-responsive>.table-bordered>tbody>tr>th:last-child,.table-responsive>.table-bordered>tfoot>tr>th:last-child,.table-responsive>.table-bordered>thead>tr>td:last-child,.table-responsive>.table-bordered>tbody>tr>td:last-child,.table-responsive>.table-bordered>tfoot>tr>td:last-child{border-right:0}.table-responsive>.table-bordered>tbody>tr>th:last-child>td,.table-responsive>.table-bordered>tfoot>tr>th:last-child>td{border-bottom:0}}fieldset{padding:0;margin:0;border:0;min-width:0}legend{display:block;width:100%;padding:0;margin-bottom:18px;font-size:19.5px;line-height:inherit;color:#333;border:0;border-bottom:1px solid #e5e5e5}label{display:inline-block;max-width:100%;margin-bottom:5px;font-weight:bold}input[type="search"]{-webkit-box-sizing:border-box;-moz-box-sizing:border-box;box-sizing:border-box}input[type="radio"],input[type="checkbox"]{margin:4px 0 0;margin-top:1px \9;line-height:normal}input[type="file"]{display:block}input[type="range"]{display:block;width:100%}select[multiple],select[size]{height:auto}input[type="file"]:focus,input[type="radio"]:focus,input[type="checkbox"]:focus{outline:thin dotted;outline:5px auto -webkit-focus-ring-color;outline-offset:-2px}output{display:block;padding-top:7px;font-size:13px;line-height:1.42857143;color:#555}.form-control{display:block;width:100%;height:32px;padding:6px 12px;font-size:13px;line-

```

height:1.42857143;color:#555;background-color:#fff;background-image:none;border:1px solid #ccc;border-radius:2px;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);-webkit-transition:border-color ease-in-out .15s, box-shadow ease-in-out .15s;-o-transition:border-color ease-in-out .15s, box-shadow ease-in-out .15s;transition:border-color ease-in-out .15s, box-shadow ease-in-out .15s}.form-control:focus{border-color:#66afe9;outline:0;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,.075), 0 0 8px rgba(102, 175, 233, 0.6);box-shadow:inset 0 1px 1px rgba(0,0,0,.075), 0 0 8px rgba(102, 175, 233, 0.6)}.form-control::-moz-placeholder{color:#999;opacity:1}.form-control:-ms-input-placeholder{color:#999}.form-control::-webkit-input-placeholder{color:#999}.form-control[disabled],.form-control[readonly],fieldset[disabled].form-control{cursor:not-allowed;background-color:#eee;opacity:1}textarea.form-control{height:auto}input[type="search"]{-webkit-appearance:none}@media screen and (-webkit-min-device-pixel-ratio:0){input[type="date"],input[type="time"],input[type="datetime-local"],input[type="month"]{line-height:32px}input[type="date"].input-sm,input[type="time"].input-sm,input[type="datetime-local"].input-sm,input[type="month"].input-sm{line-height:30px}input[type="date"].input-lg,input[type="time"].input-lg,input[type="datetime-local"].input-lg,input[type="month"].input-lg{line-height:45px}}.form-group{margin-bottom:15px}.radio,.checkbox{position:relative;display:block;margin-top:10px;margin-bottom:10px}.radio label,.checkbox label{min-height:18px;padding-left:20px;margin-bottom:0;font-weight:normal;cursor:pointer}.radio input[type="radio"],.radio-inline input[type="radio"],.checkbox input[type="checkbox"],.checkbox-inline input[type="checkbox"]{position:absolute;margin-left:-20px;margin-top:4px \9}.radio+.radio,.checkbox+.checkbox{margin-top:-5px}.radio-inline,.checkbox-inline{display:inline-block;padding-left:20px;margin-bottom:0;vertical-align:middle;font-weight:normal;cursor:pointer}.radio-inline+.radio-inline,.checkbox-inline+.checkbox-inline{margin-top:0;margin-left:10px}input[type="radio"][disabled],input[type="checkbox"][disabled],input[type="radio"].disabled,input[type="checkbox"].disabled,fieldset[disabled] input[type="radio"],fieldset[disabled] input[type="checkbox"]{cursor:not-allowed}.radio-inline.disabled,.checkbox-inline.disabled,fieldset[disabled] .radio-inline,fieldset[disabled] .checkbox-inline{cursor:not-allowed}.radio.disabled label,.checkbox.disabled label,fieldset[disabled] .radio label,fieldset[disabled] .checkbox label{cursor:not-allowed}.form-control-static{padding-top:7px;padding-bottom:7px;margin-bottom:0}.form-control-static.input-lg,.form-control-static.input-sm{padding-left:0;padding-right:0}.input-sm,.form-group-sm .form-control{height:30px;padding:5px 10px;font-size:12px;line-height:1.5;border-radius:1px}select.input-sm,select.form-group-sm .form-control{height:30px;line-height:30px}textarea.input-sm,textarea.form-group-sm .form-control,select[multiple].input-sm,select[multiple].form-group-sm .form-control{height:auto}.input-lg,.form-group-lg .form-control{height:45px;padding:10px 16px;font-size:17px;line-height:1.33;border-radius:3px}select.input-lg,select.form-group-lg .form-control{height:45px;line-height:45px}textarea.input-lg,textarea.form-group-lg .form-control,select[multiple].input-lg,select[multiple].form-group-lg .form-control{height:auto}.has-feedback{position:relative}.has-feedback .form-control{padding-right:40px}.form-control-feedback{position:absolute;top:0;right:0;z-index:2;display:block;width:32px;height:32px;line-height:32px;text-align:center;pointer-events:none}.input-lg+.form-control-feedback{width:45px;height:45px;line-height:45px}.input-sm+.form-control-feedback{width:30px;height:30px;line-height:30px}.has-success .help-block,.has-success .control-label,.has-success .radio,.has-success .checkbox,.has-success .radio-inline,.has-success

.checkbox-inline,.has-success.radio label,.has-success.checkbox label,.has-success.radio-inline label,.has-success.checkbox-inline label{color:#3c763d}.has-success .form-control{border-color:#3c763d;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);box-shadow:inset 0 1px 1px rgba(0,0,0,0.075)}.has-success .form-control:focus{border-color:#2b542c;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #67b168;box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #67b168}.has-success .input-group-addon{color:#3c763d;border-color:#3c763d;background-color:#dff0d8}.has-success .form-control-feedback{color:#3c763d}.has-warning .help-block,.has-warning .control-label,.has-warning .radio,.has-warning .checkbox,.has-warning .radio-inline,.has-warning .checkbox-inline,.has-warning .radio label,.has-warning .checkbox label,.has-warning .radio-inline label,.has-warning .checkbox-inline label{color:#8a6d3b}.has-warning .form-control{border-color:#8a6d3b;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);box-shadow:inset 0 1px 1px rgba(0,0,0,0.075)}.has-warning .form-control:focus{border-color:#66512c;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #c0a16b;box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #c0a16b}.has-warning .input-group-addon{color:#8a6d3b;border-color:#8a6d3b;background-color:#fcf8e3}.has-warning .form-control-feedback{color:#8a6d3b}.has-error .help-block,.has-error .control-label,.has-error .radio,.has-error .checkbox,.has-error .radio-inline,.has-error .checkbox-inline,.has-error .radio label,.has-error .checkbox label,.has-error .radio-inline label,.has-error .checkbox-inline label{color:#a94442}.has-error .form-control{border-color:#a94442;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);box-shadow:inset 0 1px 1px rgba(0,0,0,0.075)}.has-error .form-control:focus{border-color:#843534;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #ce8483;box-shadow:inset 0 1px 1px rgba(0,0,0,0.075),0 0 6px #ce8483}.has-error .input-group-addon{color:#a94442;border-color:#a94442;background-color:#f2dede}.has-error .form-control-feedback{color:#a94442}.has-feedback label~.form-control-feedback{top:23px}.has-feedback label.sr-only~.form-control-feedback{top:0}.help-block{display:block;margin-top:5px;margin-bottom:10px;color:#404040}@media (min-width:768px){.form-inline .form-group{display:inline-block;margin-bottom:0;vertical-align:middle}.form-inline .form-control{display:inline-block;width:auto;vertical-align:middle}.form-inline .form-control-static{display:inline-block}.form-inline .input-group{display:inline-table;vertical-align:middle}.form-inline .input-group .input-group-addon,.form-inline .input-group .input-group-btn,.form-inline .input-group .form-control{width:auto}.form-inline .input-group>.form-control{width:100%}.form-inline .control-label{margin-bottom:0;vertical-align:middle}.form-inline .radio,.form-inline .checkbox{display:inline-block;margin-top:0;margin-bottom:0;vertical-align:middle}.form-inline .radio label,.form-inline .checkbox label{padding-left:0}.form-inline .radio input[type="radio"],.form-inline .checkbox input[type="checkbox"]{position:relative;margin-left:0}.form-inline .has-feedback .form-control-feedback{top:0}}.form-horizontal .radio,.form-horizontal .checkbox,.form-horizontal .radio-inline,.form-horizontal .checkbox-inline{margin-top:0;margin-bottom:0;padding-top:7px}.form-horizontal .radio,.form-horizontal .checkbox{min-height:25px}.form-horizontal .form-group{margin-left:0;margin-right:0}@media (min-width:768px){.form-horizontal .control-label{text-align:right;margin-bottom:0;padding-top:7px}}.form-horizontal .has-feedback .form-control-feedback{right:0}@media (min-width:768px){.form-horizontal .form-group-lg .control-label{padding-top:14.3px}}@media (min-width:768px){.form-horizontal .form-group-sm .control-label{padding-top:6px}}.btn{display:inline-block;margin-bottom:0;font-weight:normal;text-align:center;vertical-align:middle;touch-action:manipulation;cursor:pointer;background-image:none;border:1px solid transparent;white-space:nowrap;padding:6px 12px;font-size:13px;line-height:1.42857143;border-radius:2px;-webkit-user-select:none;-moz-user-select:none;-ms-user-

```
select:none;user-
select:none}.btn:focus,.btn:active:focus,.btn.active:focus,.btn.focus,.btn:active.focus,.btn.active.focus{o
utline:thin dotted;outline:5px auto -webkit-focus-ring-color;outline-offset:-
2px}.btn:hover,.btn:focus,.btn.focus{color:#333;text-
decoration:none}.btn.active,.btn.active{outline:0;background-image:none;-webkit-box-shadow:inset 0
3px 5px rgba(0,0,0,0.125);box-shadow:inset 0 3px 5px
rgba(0,0,0,0.125)}.btn.disabled,.btn[disabled],fieldset[disabled] .btn{cursor:not-allowed;pointer-
events:none;opacity:.65;filter:alpha(opacity=65);-webkit-box-shadow:none;box-shadow:none}.btn-
default{color:#333;background-color:#fff;border-color:#ccc}.btn-default:hover,.btn-default:focus,.btn-
default.focus,.btn-default:active,.btn-default.active,.open>.dropdown-toggle.btn-
default{color:#333;background-color:#e6e6e6;border-color:#adadad}.btn-default:active,.btn-
default.active,.open>.dropdown-toggle.btn-default{background-image:none}.btn-default.disabled,.btn-
default[disabled],fieldset[disabled] .btn-default,.btn-default.disabled:hover,.btn-
default[disabled]:hover,fieldset[disabled] .btn-default:hover,.btn-default.disabled:focus,.btn-
default[disabled]:focus,fieldset[disabled] .btn-default:focus,.btn-default.disabled.focus,.btn-
default[disabled].focus,fieldset[disabled] .btn-default.focus,.btn-default.disabled:active,.btn-
default[disabled]:active,fieldset[disabled] .btn-default:active,.btn-default.disabled.active,.btn-
default[disabled].active,fieldset[disabled] .btn-default.active{background-color:#fff;border-
color:#ccc}.btn-default .badge{color:#fff;background-color:#333}.btn-primary{color:#fff;background-
color:#337ab7;border-color:#2e6da4}.btn-primary:hover,.btn-primary:focus,.btn-primary.focus,.btn-
primary:active,.btn-primary.active,.open>.dropdown-toggle.btn-primary{color:#fff;background-
color:#286090;border-color:#204d74}.btn-primary:active,.btn-primary.active,.open>.dropdown-
toggle.btn-primary{background-image:none}.btn-primary.disabled,.btn-
primary[disabled],fieldset[disabled] .btn-primary,.btn-primary.disabled:hover,.btn-
primary[disabled]:hover,fieldset[disabled] .btn-primary:hover,.btn-primary.disabled:focus,.btn-
primary[disabled]:focus,fieldset[disabled] .btn-primary:focus,.btn-primary.disabled.focus,.btn-
primary[disabled].focus,fieldset[disabled] .btn-primary.focus,.btn-primary.disabled:active,.btn-
primary[disabled]:active,fieldset[disabled] .btn-primary:active,.btn-primary.disabled.active,.btn-
primary[disabled].active,fieldset[disabled] .btn-primary.active{background-color:#337ab7;border-
color:#2e6da4}.btn-primary .badge{color:#337ab7;background-color:#fff}.btn-
success{color:#fff;background-color:#5cb85c;border-color:#4cae4c}.btn-success:hover,.btn-
success:focus,.btn-success.focus,.btn-success:active,.btn-success.active,.open>.dropdown-toggle.btn-
success{color:#fff;background-color:#449d44;border-color:#398439}.btn-success:active,.btn-
success.active,.open>.dropdown-toggle.btn-success{background-image:none}.btn-success.disabled,.btn-
success[disabled],fieldset[disabled] .btn-success,.btn-success.disabled:hover,.btn-
success[disabled]:hover,fieldset[disabled] .btn-success:hover,.btn-success.disabled:focus,.btn-
success[disabled]:focus,fieldset[disabled] .btn-success:focus,.btn-success.disabled.focus,.btn-
success[disabled].focus,fieldset[disabled] .btn-success.focus,.btn-success.disabled:active,.btn-
success[disabled]:active,fieldset[disabled] .btn-success:active,.btn-success.disabled.active,.btn-
success[disabled].active,fieldset[disabled] .btn-success.active{background-color:#5cb85c;border-
color:#4cae4c}.btn-success .badge{color:#5cb85c;background-color:#fff}.btn-info{color:#fff;background-
color:#5bc0de;border-color:#46b8da}.btn-info:hover,.btn-info:focus,.btn-info.focus,.btn-
info:active,.btn-info.active,.open>.dropdown-toggle.btn-info{color:#fff;background-
color:#31b0d5;border-color:#269abc}.btn-info:active,.btn-info.active,.open>.dropdown-toggle.btn-
```

info{background-image:none}.btn-info.disabled,.btn-info[disabled],fieldset[disabled] .btn-info,.btn-info.disabled:hover,.btn-info[disabled]:hover,fieldset[disabled] .btn-info:hover,.btn-info.disabled:focus,.btn-info[disabled]:focus,fieldset[disabled] .btn-info:focus,.btn-info.disabled.focus,.btn-info[disabled].focus,fieldset[disabled] .btn-info.focus,.btn-info.disabled:active,.btn-info[disabled]:active,fieldset[disabled] .btn-info:active,.btn-info.disabled.active,.btn-info[disabled].active,fieldset[disabled] .btn-info.active{background-color:#5bc0de;border-color:#46b8da}.btn-info .badge{color:#5bc0de;background-color:#fff}.btn-warning{color:#fff;background-color:#f0ad4e;border-color:#eea236}.btn-warning:hover,.btn-warning:focus,.btn-warning.focus,.btn-warning:active,.btn-warning.active,.open>.dropdown-toggle.btn-warning{color:#fff;background-color:#ec971f;border-color:#d58512}.btn-warning:active,.btn-warning.active,.open>.dropdown-toggle.btn-warning{background-image:none}.btn-warning.disabled,.btn-warning[disabled],fieldset[disabled] .btn-warning,.btn-warning.disabled:hover,.btn-warning[disabled]:hover,fieldset[disabled] .btn-warning:hover,.btn-warning.disabled:focus,.btn-warning[disabled]:focus,fieldset[disabled] .btn-warning:focus,.btn-warning.disabled.focus,.btn-warning[disabled].focus,fieldset[disabled] .btn-warning.focus,.btn-warning.disabled:active,.btn-warning[disabled]:active,fieldset[disabled] .btn-warning:active,.btn-warning.disabled.active,.btn-warning[disabled].active,fieldset[disabled] .btn-warning.active{background-color:#f0ad4e;border-color:#eea236}.btn-warning .badge{color:#f0ad4e;background-color:#fff}.btn-danger{color:#fff;background-color:#d9534f;border-color:#d43f3a}.btn-danger:hover,.btn-danger:focus,.btn-danger.focus,.btn-danger:active,.btn-danger.active,.open>.dropdown-toggle.btn-danger{color:#fff;background-color:#c9302c;border-color:#ac2925}.btn-danger:active,.btn-danger.active,.open>.dropdown-toggle.btn-danger{background-image:none}.btn-danger.disabled,.btn-danger[disabled],fieldset[disabled] .btn-danger,.btn-danger.disabled:hover,.btn-danger[disabled]:hover,fieldset[disabled] .btn-danger:hover,.btn-danger.disabled:focus,.btn-danger[disabled]:focus,fieldset[disabled] .btn-danger:focus,.btn-danger.disabled.focus,.btn-danger[disabled].focus,fieldset[disabled] .btn-danger.focus,.btn-danger.disabled:active,.btn-danger[disabled]:active,fieldset[disabled] .btn-danger:active,.btn-danger.disabled.active,.btn-danger[disabled].active,fieldset[disabled] .btn-danger.active{background-color:#d9534f;border-color:#d43f3a}.btn-danger .badge{color:#d9534f;background-color:#fff}.btn-link{color:#337ab7;font-weight:normal;border-radius:0}.btn-link,.btn-link:active,.btn-link.active,.btn-link[disabled],fieldset[disabled] .btn-link{background-color:transparent;-webkit-box-shadow:none;box-shadow:none}.btn-link,.btn-link:hover,.btn-link:focus,.btn-link:active{border-color:transparent}.btn-link:hover,.btn-link:focus{color:#23527c;text-decoration:underline;background-color:transparent}.btn-link[disabled]:hover,fieldset[disabled] .btn-link:hover,.btn-link[disabled]:focus,fieldset[disabled] .btn-link:focus{color:#777;text-decoration:none}.btn-lg,.btn-group-lg>.btn{padding:10px 16px;font-size:17px;line-height:1.33;border-radius:3px}.btn-sm,.btn-group-sm>.btn{padding:5px 10px;font-size:12px;line-height:1.5;border-radius:1px}.btn-xs,.btn-group-xs>.btn{padding:1px 5px;font-size:12px;line-height:1.5;border-radius:1px}.btn-block{display:block;width:100%}.btn-block+.btn-block{margin-top:5px}input[type="submit"].btn-block,input[type="reset"].btn-block,input[type="button"].btn-block{width:100%}.fade{opacity:0;-webkit-transition:opacity .15s linear;-o-transition:opacity .15s linear;transition:opacity .15s linear}.fade.in{opacity:1}.collapse{display:none;visibility:hidden}.collapse.in{display:block;visibility:visible}tr.collapse.in{display:table-row}tbody.collapse.in{display:table-row-group}.collapsing{position:relative;height:0;overflow:hidden;-webkit-transition-property:height,

visibility;transition-property:height, visibility;-webkit-transition-duration:.35s;transition-duration:.35s;-webkit-transition-timing-function:ease;transition-timing-function:ease}.caret{display:inline-block;width:0;height:0;margin-left:2px;vertical-align:middle;border-top:4px solid;border-right:4px solid transparent;border-left:4px solid transparent}.dropdown{position:relative}.dropdown-toggle:focus{outline:0}.dropdown-menu{position:absolute;top:100%;left:0;z-index:1000;display:none;float:left;min-width:160px;padding:5px 0;margin:2px 0 0 0;list-style:none;font-size:13px;text-align:left;background-color:#fff;border:1px solid #ccc;border:1px solid rgba(0,0,0,0.15);border-radius:2px;-webkit-box-shadow:0 6px 12px rgba(0,0,0,0.175);box-shadow:0 6px 12px rgba(0,0,0,0.175);background-clip:padding-box}.dropdown-menu.pull-right{right:0;left:auto}.dropdown-menu .divider{height:1px;margin:8px 0;overflow:hidden;background-color:#e5e5e5}.dropdown-menu>li>a{display:block;padding:3px 20px;clear:both;font-weight:normal;line-height:1.42857143;color:#333;white-space:nowrap}.dropdown-menu>li>a:hover,.dropdown-menu>li>a:focus{text-decoration:none;color:#262626;background-color:#f5f5f5}.dropdown-menu>.active>a,.dropdown-menu>.active>a:hover,.dropdown-menu>.active>a:focus{color:#fff;text-decoration:none;outline:0;background-color:#337ab7}.dropdown-menu>.disabled>a,.dropdown-menu>.disabled>a:hover,.dropdown-menu>.disabled>a:focus{color:#777}.dropdown-menu>.disabled>a:hover,.dropdown-menu>.disabled>a:focus{text-decoration:none;background-color:transparent;background-image:none;filter:progid:DXImageTransform.Microsoft.gradient(enabled = false);cursor:not-allowed}.open>.dropdown-menu{display:block}.open>a{outline:0}.dropdown-menu-right{left:auto;right:0}.dropdown-menu-left{left:0;right:auto}.dropdown-header{display:block;padding:3px 20px;font-size:12px;line-height:1.42857143;color:#777;white-space:nowrap}.dropdown-backdrop{position:fixed;left:0;right:0;bottom:0;top:0;z-index:990}.pull-right>.dropdown-menu{right:0;left:auto}.dropup .caret,.navbar-fixed-bottom .dropdown .caret{border-top:0;border-bottom:4px solid;content:""}.dropup .dropdown-menu,.navbar-fixed-bottom .dropdown .dropdown-menu{top:auto;bottom:100%;margin-bottom:1px}@media (min-width:541px){.navbar-right .dropdown-menu{left:auto;right:0}.navbar-right .dropdown-menu-left{left:0;right:auto}}.btn-group,.btn-group-vertical{position:relative;display:inline-block;vertical-align:middle}.btn-group>.btn,.btn-group-vertical>.btn{position:relative;float:left}.btn-group>.btn:hover,.btn-group-vertical>.btn:hover,.btn-group>.btn:focus,.btn-group-vertical>.btn:focus,.btn-group>.btn:active,.btn-group-vertical>.btn:active,.btn-group>.btn.active,.btn-group-vertical>.btn.active{z-index:2}.btn-group .btn+.btn,.btn-group .btn+.btn-group,.btn-group .btn-group+.btn,.btn-group .btn-group+.btn-group{margin-left:-1px}.btn-toolbar{margin-left:-5px}.btn-toolbar .btn-group,.btn-toolbar .input-group{float:left}.btn-toolbar>.btn,.btn-toolbar>.btn-group,.btn-toolbar>.input-group{margin-left:5px}.btn-group>.btn:not(:first-child):not(:last-child):not(.dropdown-toggle){border-radius:0}.btn-group>.btn:first-child{margin-left:0}.btn-group>.btn:first-child:not(:last-child):not(.dropdown-toggle){border-bottom-right-radius:0;border-top-right-radius:0}.btn-group>.btn:last-child:not(:first-child),.btn-group>.dropdown-toggle:not(:first-child){border-bottom-left-radius:0;border-top-left-radius:0}.btn-group>.btn-group{float:left}.btn-group>.btn-group:not(:first-child):not(:last-child)>.btn{border-radius:0}.btn-group>.btn-group:first-child>.btn:last-child,.btn-group>.btn-group:first-child>.dropdown-toggle{border-bottom-right-radius:0;border-top-right-radius:0}.btn-group>.btn-group:last-child>.btn:first-child{border-bottom-left-radius:0;border-top-left-radius:0}.btn-group .dropdown-toggle:active,.btn-group.open .dropdown-toggle{outline:0}.btn-group>.btn+.dropdown-toggle{padding-left:8px;padding-right:8px}.btn-group>.btn-lg+.dropdown-toggle{padding-

left:12px;padding-right:12px}.btn-group.open .dropdown-toggle{-webkit-box-shadow:inset 0 3px 5px rgba(0,0,0,0.125);box-shadow:inset 0 3px 5px rgba(0,0,0,0.125)}.btn-group.open .dropdown-toggle.btn-link{-webkit-box-shadow:none;box-shadow:none}.btn .caret{margin-left:0}.btn-lg .caret{border-width:5px 5px 0;border-bottom-width:0}.dropup .btn-lg .caret{border-width:0 5px 5px}.btn-group-vertical>.btn,.btn-group-vertical>.btn-group,.btn-group-vertical>.btn-group>.btn{display:block;float:none;width:100%;max-width:100%}.btn-group-vertical>.btn-group>.btn{float:none}.btn-group-vertical>.btn+.btn,.btn-group-vertical>.btn+.btn-group,.btn-group-vertical>.btn-group+.btn,.btn-group-vertical>.btn-group+.btn-group{margin-top:-1px;margin-left:0}.btn-group-vertical>.btn:not(:first-child):not(:last-child){border-radius:0}.btn-group-vertical>.btn:first-child:not(:last-child){border-top-right-radius:2px;border-bottom-right-radius:0;border-bottom-left-radius:0}.btn-group-vertical>.btn:last-child:not(:first-child){border-bottom-left-radius:2px;border-top-right-radius:0;border-top-left-radius:0}.btn-group-vertical>.btn-group:not(:first-child):not(:last-child)>.btn{border-radius:0}.btn-group-vertical>.btn-group:first-child:not(:last-child)>.btn:last-child,.btn-group-vertical>.btn-group:first-child:not(:last-child)>.dropdown-toggle{border-bottom-right-radius:0;border-bottom-left-radius:0}.btn-group-vertical>.btn-group:last-child:not(:first-child)>.btn:first-child{border-top-right-radius:0;border-top-left-radius:0}.btn-group-justified{display:table;width:100%;table-layout:fixed;border-collapse:separate}.btn-group-justified>.btn,.btn-group-justified>.btn-group{float:none;display:table-cell;width:1%}.btn-group-justified>.btn-group .btn{width:100%}.btn-group-justified>.btn-group .dropdown-menu{left:auto}[data-toggle="buttons"]>.btn input[type="radio"],[data-toggle="buttons"]>.btn-group>.btn input[type="radio"],[data-toggle="buttons"]>.btn input[type="checkbox"],[data-toggle="buttons"]>.btn-group>.btn input[type="checkbox"]{position:absolute;clip:rect(0, 0, 0, 0);pointer-events:none}.input-group{position:relative;display:table;border-collapse:separate}.input-group[class\*="col-"]{float:none;padding-left:0;padding-right:0}.input-group .form-control{position:relative;z-index:2;float:left;width:100%;margin-bottom:0}.input-group-lg>.form-control,.input-group-lg>.input-group-addon,.input-group-lg>.input-group-btn>.btn{height:45px;padding:10px 16px;font-size:17px;line-height:1.33;border-radius:3px}select.input-group-lg>.form-control,select.input-group-lg>.input-group-addon,select.input-group-lg>.input-group-btn>.btn{height:45px;line-height:45px}textarea.input-group-lg>.form-control,textarea.input-group-lg>.input-group-addon,textarea.input-group-lg>.input-group-btn>.btn,select[multiple].input-group-lg>.form-control,select[multiple].input-group-lg>.input-group-addon,select[multiple].input-group-lg>.input-group-btn>.btn{height:auto}.input-group-sm>.form-control,.input-group-sm>.input-group-addon,.input-group-sm>.input-group-btn>.btn{height:30px;padding:5px 10px;font-size:12px;line-height:1.5;border-radius:1px}select.input-group-sm>.form-control,select.input-group-sm>.input-group-addon,select.input-group-sm>.input-group-btn>.btn{height:30px;line-height:30px}textarea.input-group-sm>.form-control,textarea.input-group-sm>.input-group-addon,textarea.input-group-sm>.input-group-btn>.btn,select[multiple].input-group-sm>.form-control,select[multiple].input-group-sm>.input-group-addon,select[multiple].input-group-sm>.input-group-btn>.btn{height:auto}.input-group-addon,.input-group-btn,.input-group .form-control{display:table-cell}.input-group-addon:not(:first-child):not(:last-child),.input-group-btn:not(:first-child):not(:last-child),.input-group .form-control:not(:first-child):not(:last-child){border-radius:0}.input-group-addon,.input-group-btn{width:1%;white-space:nowrap;vertical-align:middle}.input-group-addon{padding:6px 12px;font-size:13px;font-weight:normal;line-height:1;color:#555;text-align:center;background-color:#eee;border:1px solid #ccc;border-radius:2px}.input-group-addon.input-sm{padding:5px 10px;font-size:12px;border-radius:1px}.input-group-addon.input-lg{padding:10px

16px;font-size:17px;border-radius:3px}.input-group-addon input[type="radio"],.input-group-addon input[type="checkbox"]{margin-top:0}.input-group .form-control:first-child,.input-group-addon:first-child,.input-group-btn:first-child>.btn,.input-group-btn:first-child>.btn-group>.btn,.input-group-btn:first-child>.dropdown-toggle,.input-group-btn:last-child>.btn:not(:last-child):not(.dropdown-toggle),.input-group-btn:last-child>.btn-group:not(:last-child)>.btn{border-bottom-right-radius:0;border-top-right-radius:0}.input-group-addon:first-child{border-right:0}.input-group .form-control:last-child,.input-group-addon:last-child,.input-group-btn:last-child>.btn,.input-group-btn:last-child>.btn-group>.btn,.input-group-btn:last-child>.dropdown-toggle,.input-group-btn:first-child>.btn:not(:first-child),.input-group-btn:first-child>.btn-group:not(:first-child)>.btn{border-bottom-left-radius:0;border-top-left-radius:0}.input-group-addon:last-child{border-left:0}.input-group-btn{position:relative;font-size:0;white-space:nowrap}.input-group-btn>.btn{position:relative}.input-group-btn>.btn+.btn{margin-left:-1px}.input-group-btn>.btn:hover,.input-group-btn>.btn:focus,.input-group-btn>.btn:active{z-index:2}.input-group-btn:first-child>.btn,.input-group-btn:first-child>.btn-group{margin-right:-1px}.input-group-btn:last-child>.btn,.input-group-btn:last-child>.btn-group{margin-left:-1px}.nav{margin-bottom:0;padding-left:0;list-style:none}.nav>li{position:relative;display:block}.nav>li>a{position:relative;display:block;padding:10px 15px}.nav>li>a:hover,.nav>li>a:focus{text-decoration:none;background-color:#eee}.nav>li.disabled>a{color:#777}.nav>li.disabled>a:hover,.nav>li.disabled>a:focus{color:#777;text-decoration:none;background-color:transparent;cursor:not-allowed}.nav .open>a,.nav .open>a:hover,.nav .open>a:focus{background-color:#eee;border-color:#337ab7}.nav .nav-divider{height:1px;margin:8px 0;overflow:hidden;background-color:#e5e5e5}.nav>li>a>img{max-width:none}.nav-tabs{border-bottom:1px solid #ddd}.nav-tabs>li{float:left;margin-bottom:-1px}.nav-tabs>li>a{margin-right:2px;line-height:1.42857143;border:1px solid transparent;border-radius:2px 2px 0 0}.nav-tabs>li>a:hover{border-color:#eee #eee #ddd}.nav-tabs>li.active>a,.nav-tabs>li.active>a:hover,.nav-tabs>li.active>a:focus{color:#555;background-color:#fff;border:1px solid #ddd;border-bottom-color:transparent;cursor:default}.nav-tabs.nav-justified{width:100%;border-bottom:0}.nav-tabs.nav-justified>li{float:none}.nav-tabs.nav-justified>li>a{text-align:center;margin-bottom:5px}.nav-tabs.nav-justified>.dropdown .dropdown-menu{top:auto;left:auto}@media (min-width:768px){.nav-tabs.nav-justified>li{display:table-cell;width:1%}.nav-tabs.nav-justified>li>a{margin-bottom:0}}.nav-tabs.nav-justified>li>a{margin-right:0;border-radius:2px}.nav-tabs.nav-justified>.active>a,.nav-tabs.nav-justified>.active>a:hover,.nav-tabs.nav-justified>.active>a:focus{border:1px solid #ddd}@media (min-width:768px){.nav-tabs.nav-justified>li>a{border-bottom:1px solid #ddd;border-radius:2px 2px 0 0}.nav-tabs.nav-justified>.active>a,.nav-tabs.nav-justified>.active>a:hover,.nav-tabs.nav-justified>.active>a:focus{border-bottom-color:#fff}}.nav-pills>li{float:left}.nav-pills>li>a{border-radius:2px}.nav-pills>li+li{margin-left:2px}.nav-pills>li.active>a,.nav-pills>li.active>a:hover,.nav-pills>li.active>a:focus{color:#fff;background-color:#337ab7}.nav-stacked>li{float:none}.nav-stacked>li+li{margin-top:2px;margin-left:0}.nav-justified{width:100%}.nav-justified>li{float:none}.nav-justified>li>a{text-align:center;margin-bottom:5px}.nav-justified>.dropdown .dropdown-menu{top:auto;left:auto}@media (min-width:768px){.nav-justified>li{display:table-cell;width:1%}.nav-justified>li>a{margin-bottom:0}}.nav-tabs-justified{border-bottom:0}.nav-tabs-justified>li>a{margin-right:0;border-radius:2px}.nav-tabs-justified>.active>a,.nav-tabs-justified>.active>a:hover,.nav-tabs-justified>.active>a:focus{border:1px solid #ddd}@media (min-width:768px){.nav-tabs-justified>li>a{border-bottom:1px solid #ddd;border-radius:2px 2px 0 0}.nav-tabs-



```

justified>.active>a,.nav-tabs-justified>.active>a:hover,.nav-tabs-justified>.active>a:focus{border-
bottom-color:#fff}}.tab-content>.tab-pane{display:none;visibility:hidden}.tab-
content>.active{display:block;visibility:visible}.nav-tabs .dropdown-menu{margin-top:-1px;border-top-
right-radius:0;border-top-left-radius:0}.navbar{position:relative;min-height:30px;margin-
bottom:18px;border:1px solid transparent}@media (min-width:541px){.navbar{border-
radius:2px}}@media (min-width:541px){.navbar-header{float:left}.navbar-collapse{overflow-
x:visible;padding-right:0;padding-left:0;border-top:1px solid transparent;box-shadow:inset 0 1px 0
rgba(255,255,255,0.1);-webkit-overflow-scrolling:touch}.navbar-collapse.in{overflow-y:auto}@media
(min-width:541px){.navbar-collapse{width:auto;border-top:0;box-shadow:none}.navbar-
collapse.collapse{display:block !important;visibility:visible !important;height:auto !important;padding-
bottom:0;overflow:visible !important}.navbar-collapse.in{overflow-y:visible}.navbar-fixed-top .navbar-
collapse,.navbar-static-top .navbar-collapse,.navbar-fixed-bottom .navbar-collapse{padding-
left:0;padding-right:0}}.navbar-fixed-top .navbar-collapse,.navbar-fixed-bottom .navbar-collapse{max-
height:340px}@media (max-device-width:540px) and (orientation:landscape){.navbar-fixed-top .navbar-
collapse,.navbar-fixed-bottom .navbar-collapse{max-height:200px}}.container>.navbar-
header,.container-fluid>.navbar-header,.container>.navbar-collapse,.container-fluid>.navbar-
collapse{margin-right:0;margin-left:0}@media (min-width:541px){.container>.navbar-header,.container-
fluid>.navbar-header,.container>.navbar-collapse,.container-fluid>.navbar-collapse{margin-
right:0;margin-left:0}}.navbar-static-top{z-index:1000;border-width:0 0 1px}@media (min-
width:541px){.navbar-static-top{border-radius:0}}.navbar-fixed-top,.navbar-fixed-
bottom{position:fixed;right:0;left:0;z-index:1030}@media (min-width:541px){.navbar-fixed-top,.navbar-
fixed-bottom{border-radius:0}}.navbar-fixed-top{top:0;border-width:0 0 1px}.navbar-fixed-
bottom{bottom:0;margin-bottom:0;border-width:1px 0 0}.navbar-brand{float:left;padding:6px 0;font-
size:17px;line-height:18px;height:30px}.navbar-brand:hover,.navbar-brand:focus{text-
decoration:none}.navbar-brand>img{display:block}@media (min-width:541px){.navbar>.container
.navbar-brand,.navbar>.container-fluid .navbar-brand{margin-left:0}}.navbar-
toggle{position:relative;float:right;margin-right:0;padding:9px 10px;margin-top:-2px;margin-bottom:-
2px;background-color:transparent;background-image:none;border:1px solid transparent;border-
radius:2px}.navbar-toggle:focus{outline:0}.navbar-toggle .icon-
bar{display:block;width:22px;height:2px;border-radius:1px}.navbar-toggle .icon-bar+.icon-bar{margin-
top:4px}@media (min-width:541px){.navbar-toggle{display:none}}.navbar-nav{margin:3px 0}.navbar-
nav>li>a{padding-top:10px;padding-bottom:10px;line-height:18px}@media (max-width:540px){.navbar-
nav .open .dropdown-menu{position:static;float:none;width:auto;margin-top:0;background-
color:transparent;border:0;box-shadow:none}.navbar-nav .open .dropdown-menu>li>a,.navbar-nav
.open .dropdown-menu .dropdown-header{padding:5px 15px 5px 25px}.navbar-nav .open .dropdown-
menu>li>a{line-height:18px}.navbar-nav .open .dropdown-menu>li>a:hover,.navbar-nav .open
.dropdown-menu>li>a:focus{background-image:none}}@media (min-width:541px){.navbar-
nav{float:left;margin:0}.navbar-nav>li{float:left}.navbar-nav>li>a{padding-top:6px;padding-
bottom:6px}}.navbar-form{margin-left:0;margin-right:0;padding:10px 0;border-top:1px solid
transparent;border-bottom:1px solid transparent;-webkit-box-shadow:inset 0 1px 0
rgba(255,255,255,0.1),0 1px 0 rgba(255,255,255,0.1);box-shadow:inset 0 1px 0 rgba(255,255,255,0.1),0
1px 0 rgba(255,255,255,0.1);margin-top:-1px;margin-bottom:-1px}@media (min-width:768px){.navbar-
form .form-group{display:inline-block;margin-bottom:0;vertical-align:middle}.navbar-form .form-
control{display:inline-block;width:auto;vertical-align:middle}.navbar-form .form-control-

```

static{display:inline-block}.navbar-form .input-group{display:inline-table;vertical-align:middle}.navbar-form .input-group .input-group-addon,.navbar-form .input-group .input-group-btn,.navbar-form .input-group .form-control{width:auto}.navbar-form .input-group>.form-control{width:100%}.navbar-form .control-label{margin-bottom:0;vertical-align:middle}.navbar-form .radio,.navbar-form .checkbox{display:inline-block;margin-top:0;margin-bottom:0;vertical-align:middle}.navbar-form .radio label,.navbar-form .checkbox label{padding-left:0}.navbar-form .radio input[type="radio"],.navbar-form .checkbox input[type="checkbox"]{position:relative;margin-left:0}.navbar-form .has-feedback .form-control-feedback{top:0}@media (max-width:540px){.navbar-form .form-group{margin-bottom:5px}.navbar-form .form-group:last-child{margin-bottom:0}}@media (min-width:541px){.navbar-form{width:auto;border:0;margin-left:0;margin-right:0;padding-top:0;padding-bottom:0;-webkit-box-shadow:none;box-shadow:none}}.navbar-nav>li>.dropdown-menu{margin-top:0;border-top-right-radius:0;border-top-left-radius:0}.navbar-fixed-bottom .navbar-nav>li>.dropdown-menu{border-top-right-radius:2px;border-top-left-radius:2px;border-bottom-right-radius:0;border-bottom-left-radius:0}.navbar-btn{margin-top:-1px;margin-bottom:-1px}.navbar-btn.btn-sm{margin-top:0;margin-bottom:0}.navbar-btn.btn-xs{margin-top:4px;margin-bottom:4px}.navbar-text{margin-top:6px;margin-bottom:6px}@media (min-width:541px){.navbar-text{float:left;margin-left:0;margin-right:0}}@media (min-width:541px){.navbar-left{float:left !important;float:left}.navbar-right{float:right !important;float:right;margin-right:0}.navbar-right~.navbar-right{margin-right:0}}.navbar-default{background-color:#f8f8f8;border-color:#e7e7e7}.navbar-default .navbar-brand{color:#777}.navbar-default .navbar-brand:hover,.navbar-default .navbar-brand:focus{color:#5e5e5e;background-color:transparent}.navbar-default .navbar-text{color:#777}.navbar-default .navbar-nav>li>a{color:#777}.navbar-default .navbar-nav>li>a:hover,.navbar-default .navbar-nav>li>a:focus{color:#333;background-color:transparent}.navbar-default .navbar-nav>.active>a,.navbar-default .navbar-nav>.active>a:hover,.navbar-default .navbar-nav>.active>a:focus{color:#555;background-color:#e7e7e7}.navbar-default .navbar-nav>.disabled>a,.navbar-default .navbar-nav>.disabled>a:hover,.navbar-default .navbar-nav>.disabled>a:focus{color:#ccc;background-color:transparent}.navbar-default .navbar-toggle{border-color:#ddd}.navbar-default .navbar-toggle:hover,.navbar-default .navbar-toggle:focus{background-color:#ddd}.navbar-default .navbar-toggle .icon-bar{background-color:#888}.navbar-default .navbar-collapse,.navbar-default .navbar-form{border-color:#e7e7e7}.navbar-default .navbar-nav>.open>a,.navbar-default .navbar-nav>.open>a:hover,.navbar-default .navbar-nav>.open>a:focus{background-color:#e7e7e7;color:#555}@media (max-width:540px){.navbar-default .navbar-nav .open .dropdown-menu>li>a{color:#777}.navbar-default .navbar-nav .open .dropdown-menu>li>a:hover,.navbar-default .navbar-nav .open .dropdown-menu>li>a:focus{color:#333;background-color:transparent}.navbar-default .navbar-nav .open .dropdown-menu>.active>a,.navbar-default .navbar-nav .open .dropdown-menu>.active>a:hover,.navbar-default .navbar-nav .open .dropdown-menu>.active>a:focus{color:#555;background-color:#e7e7e7}.navbar-default .navbar-nav .open .dropdown-menu>.disabled>a,.navbar-default .navbar-nav .open .dropdown-menu>.disabled>a:hover,.navbar-default .navbar-nav .open .dropdown-menu>.disabled>a:focus{color:#ccc;background-color:transparent}}.navbar-default .navbar-link{color:#777}.navbar-default .navbar-link:hover{color:#333}.navbar-default .btn-link{color:#777}.navbar-default .btn-link:hover,.navbar-default .btn-link:focus{color:#333}.navbar-default .btn-link[disabled]:hover,fieldset[disabled] .navbar-default .btn-link:hover,.navbar-default .btn-

link[disabled]:focus,fieldset[disabled] .navbar-default .btn-link:focus{color:#ccc}.navbar-inverse{background-color:#222;border-color:#080808}.navbar-inverse .navbar-brand{color:#9d9d9d}.navbar-inverse .navbar-brand:hover,.navbar-inverse .navbar-brand:focus{color:#fff;background-color:transparent}.navbar-inverse .navbar-text{color:#9d9d9d}.navbar-inverse .navbar-nav>li>a{color:#9d9d9d}.navbar-inverse .navbar-nav>li>a:hover,.navbar-inverse .navbar-nav>li>a:focus{color:#fff;background-color:transparent}.navbar-inverse .navbar-nav>.active>a,.navbar-inverse .navbar-nav>.active>a:hover,.navbar-inverse .navbar-nav>.active>a:focus{color:#fff;background-color:#080808}.navbar-inverse .navbar-nav>.disabled>a,.navbar-inverse .navbar-nav>.disabled>a:hover,.navbar-inverse .navbar-nav>.disabled>a:focus{color:#444;background-color:transparent}.navbar-inverse .navbar-toggle{border-color:#333}.navbar-inverse .navbar-toggle:hover,.navbar-inverse .navbar-toggle:focus{background-color:#333}.navbar-inverse .navbar-toggle .icon-bar{background-color:#fff}.navbar-inverse .navbar-collapse,.navbar-inverse .navbar-form{border-color:#101010}.navbar-inverse .navbar-nav>.open>a,.navbar-inverse .navbar-nav>.open>a:hover,.navbar-inverse .navbar-nav>.open>a:focus{background-color:#080808;color:#fff}@media (max-width:540px){.navbar-inverse .navbar-nav .open .dropdown-menu>.dropdown-header{border-color:#080808}.navbar-inverse .navbar-nav .open .dropdown-menu .divider{background-color:#080808}.navbar-inverse .navbar-nav .open .dropdown-menu>li>a{color:#9d9d9d}.navbar-inverse .navbar-nav .open .dropdown-menu>li>a:hover,.navbar-inverse .navbar-nav .open .dropdown-menu>li>a:focus{color:#fff;background-color:transparent}.navbar-inverse .navbar-nav .open .dropdown-menu>.active>a,.navbar-inverse .navbar-nav .open .dropdown-menu>.active>a:hover,.navbar-inverse .navbar-nav .open .dropdown-menu>.active>a:focus{color:#fff;background-color:#080808}.navbar-inverse .navbar-nav .open .dropdown-menu>.disabled>a,.navbar-inverse .navbar-nav .open .dropdown-menu>.disabled>a:hover,.navbar-inverse .navbar-nav .open .dropdown-menu>.disabled>a:focus{color:#444;background-color:transparent}}.navbar-inverse .navbar-link{color:#9d9d9d}.navbar-inverse .navbar-link:hover{color:#fff}.navbar-inverse .btn-link{color:#9d9d9d}.navbar-inverse .btn-link:hover,.navbar-inverse .btn-link:focus{color:#fff}.navbar-inverse .btn-link[disabled]:hover,fieldset[disabled] .navbar-inverse .btn-link:hover,.navbar-inverse .btn-link[disabled]:focus,fieldset[disabled] .navbar-inverse .btn-link:focus{color:#444}.breadcrumb{padding:8px 15px;margin-bottom:18px;list-style:none;background-color:#f5f5f5;border-radius:2px}.breadcrumb>li{display:inline-block}.breadcrumb>li+li:before{content:"\00a0";padding:0 5px;color:#5e5e5e}.breadcrumb>.active{color:#777}.pagination{display:inline-block;padding-left:0;margin:18px 0;border-radius:2px}.pagination>li{display:inline}.pagination>li>a,.pagination>li>span{position:relative;float:left;padding:6px 12px;line-height:1.42857143;text-decoration:none;color:#337ab7;background-color:#fff;border:1px solid #ddd;margin-left:-1px}.pagination>li:first-child>a,.pagination>li:first-child>span{margin-left:0;border-bottom-left-radius:2px;border-top-left-radius:2px}.pagination>li:last-child>a,.pagination>li:last-child>span{border-bottom-right-radius:2px;border-top-right-radius:2px}.pagination>li>a:hover,.pagination>li>span:hover,.pagination>li>a:focus,.pagination>li>span:focus{color:#23527c;background-color:#eee;border-color:#ddd}.pagination>.active>a,.pagination>.active>span,.pagination>.active>a:hover,.pagination>.active>span:hover,.pagination>.active>a:focus,.pagination>.active>span:focus{z-index:2;color:#fff;background-color:#337ab7;border-

color:#337ab7;cursor:default}.pagination>.disabled>span,.pagination>.disabled>span:hover,.pagination>.disabled>span:focus,.pagination>.disabled>a,.pagination>.disabled>a:hover,.pagination>.disabled>a:focus{color:#777;background-color:#fff;border-color:#ddd;cursor:not-allowed}.pagination-lg>li>a,.pagination-lg>li>span{padding:10px 16px;font-size:17px}.pagination-lg>li:first-child>a,.pagination-lg>li:first-child>span{border-bottom-left-radius:3px;border-top-left-radius:3px}.pagination-lg>li:last-child>a,.pagination-lg>li:last-child>span{border-bottom-right-radius:3px;border-top-right-radius:3px}.pagination-sm>li>a,.pagination-sm>li>span{padding:5px 10px;font-size:12px}.pagination-sm>li:first-child>a,.pagination-sm>li:first-child>span{border-bottom-left-radius:1px;border-top-left-radius:1px}.pagination-sm>li:last-child>a,.pagination-sm>li:last-child>span{border-bottom-right-radius:1px;border-top-right-radius:1px}.pager{padding-left:0;margin:18px 0;list-style:none;text-align:center}.pager li{display:inline}.pager li>a,.pager li>span{display:inline-block;padding:5px 14px;background-color:#fff;border:1px solid #ddd;border-radius:15px}.pager li>a:hover,.pager li>a:focus{text-decoration:none;background-color:#eee}.pager .next>a,.pager .next>span{float:right}.pager .previous>a,.pager .previous>span{float:left}.pager .disabled>a,.pager .disabled>a:hover,.pager .disabled>a:focus,.pager .disabled>span{color:#777;background-color:#fff;cursor:not-allowed}.label{display:inline;padding:.2em .6em .3em;font-size:75%;font-weight:bold;line-height:1;color:#fff;text-align:center;white-space:nowrap;vertical-align:baseline;border-radius:.25em}.a.label:hover,.a.label:focus{color:#fff;text-decoration:none;cursor:pointer}.label.empty{display:none}.btn .label{position:relative;top:-1px}.label-default{background-color:#777}.label-default[href]:hover,.label-default[href]:focus{background-color:#5e5e5e}.label-primary{background-color:#337ab7}.label-primary[href]:hover,.label-primary[href]:focus{background-color:#286090}.label-success{background-color:#5cb85c}.label-success[href]:hover,.label-success[href]:focus{background-color:#449d44}.label-info{background-color:#5bc0de}.label-info[href]:hover,.label-info[href]:focus{background-color:#31b0d5}.label-warning{background-color:#f0ad4e}.label-warning[href]:hover,.label-warning[href]:focus{background-color:#ec971f}.label-danger{background-color:#d9534f}.label-danger[href]:hover,.label-danger[href]:focus{background-color:#c9302c}.badge{display:inline-block;min-width:10px;padding:3px 7px;font-size:12px;font-weight:bold;color:#fff;line-height:1;vertical-align:baseline;white-space:nowrap;text-align:center;background-color:#777;border-radius:10px}.badge.empty{display:none}.btn .badge{position:relative;top:-1px}.btn-xs .badge{top:0;padding:1px 5px}.a.badge:hover,.a.badge:focus{color:#fff;text-decoration:none;cursor:pointer}.list-group-item.active>.badge,.nav-pills>.active>a>.badge{color:#337ab7;background-color:#fff}.list-group-item>.badge{float:right}.list-group-item>.badge+.badge{margin-right:5px}.nav-pills>li>a>.badge{margin-left:3px}.jumbotron{padding:30px 15px;margin-bottom:30px;color:inherit;background-color:#eee}.jumbotron h1,.jumbotron .h1{color:inherit}.jumbotron p{margin-bottom:15px;font-size:20px;font-weight:200}.jumbotron>hr{border-top-color:#d5d5d5}.container .jumbotron,.container-fluid .jumbotron{border-radius:3px}.jumbotron .container{max-width:100%}@media screen and (min-width:768px){.jumbotron{padding:48px 0}.container .jumbotron,.container-fluid .jumbotron{padding-left:60px;padding-right:60px}.jumbotron h1,.jumbotron .h1{font-size:58.5px}}.thumbnail{display:block;padding:4px;margin-bottom:18px;line-height:1.42857143;background-color:#fff;border:1px solid #ddd;border-radius:2px;-webkit-transition:border .2s ease-in-out;-o-transition:border .2s ease-in-out;transition:border .2s ease-in-out}.thumbnail>img,.thumbnail a>img{margin-left:auto;margin-

right:auto}a.thumbnail:hover,a.thumbnail:focus,a.thumbnail.active{border-color:#337ab7}.thumbnail .caption{padding:9px;color:#000}.alert{padding:15px;margin-bottom:18px;border:1px solid transparent;border-radius:2px}.alert h4{margin-top:0;color:inherit}.alert .alert-link{font-weight:bold}.alert>p,.alert>ul{margin-bottom:0}.alert>p+p{margin-top:5px}.alert-dismissible,.alert-dismissible{padding-right:35px}.alert-dismissible .close,.alert-dismissible .close{position:relative;top:-2px;right:-21px;color:inherit}.alert-success{background-color:#dff0d8;border-color:#d6e9c6;color:#3c763d}.alert-success hr{border-top-color:#c9e2b3}.alert-success .alert-link{color:#2b542c}.alert-info{background-color:#d9edf7;border-color:#bce8f1;color:#31708f}.alert-info hr{border-top-color:#a6e1ec}.alert-info .alert-link{color:#245269}.alert-warning{background-color:#fcf8e3;border-color:#faebcc;color:#8a6d3b}.alert-warning hr{border-top-color:#f7e1b5}.alert-warning .alert-link{color:#66512c}.alert-danger{background-color:#f2dede;border-color:#ebccd1;color:#a94442}.alert-danger hr{border-top-color:#e4b9c0}.alert-danger .alert-link{color:#843534}@-webkit-keyframes progress-bar-stripes{from{background-position:40px 0}to{background-position:0 0}}@keyframes progress-bar-stripes{from{background-position:40px 0}to{background-position:0 0}}.progress{overflow:hidden;height:18px;margin-bottom:18px;background-color:#f5f5f5;border-radius:2px;-webkit-box-shadow:inset 0 1px 2px rgba(0,0,0,0.1);box-shadow:inset 0 1px 2px rgba(0,0,0,0.1)}.progress-bar{float:left;width:0;height:100%;font-size:12px;line-height:18px;color:#fff;text-align:center;background-color:#337ab7;-webkit-box-shadow:inset 0 -1px 0 rgba(0,0,0,0.15);box-shadow:inset 0 -1px 0 rgba(0,0,0,0.15);-webkit-transition:width .6s ease;-o-transition:width .6s ease;transition:width .6s ease}.progress-striped .progress-bar,.progress-bar-striped{background-image:-webkit-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:-o-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-size:40px 40px}.progress.active .progress-bar,.progress-bar.active{-webkit-animation:progress-bar-stripes 2s linear infinite;-o-animation:progress-bar-stripes 2s linear infinite;animation:progress-bar-stripes 2s linear infinite}.progress-bar-success{background-color:#5cb85c}.progress-striped .progress-bar-success{background-image:-webkit-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:-o-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent)}.progress-bar-info{background-color:#5bc0de}.progress-striped .progress-bar-info{background-image:-webkit-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:-o-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-image:linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%, rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent)}.progress-bar-warning{background-color:#f0ad4e}.progress-striped .progress-bar-warning{background-image:-webkit-

linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-  
image:-o-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-  
image:linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent)).progress-bar-  
danger{background-color:#d9534f}.progress-striped .progress-bar-danger{background-image:-webkit-  
linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-  
image:-o-linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%, transparent);background-  
image:linear-gradient(45deg, rgba(255,255,255,0.15) 25%, transparent 25%, transparent 50%,  
rgba(255,255,255,0.15) 50%, rgba(255,255,255,0.15) 75%, transparent 75%,  
transparent)).media{margin-top:15px}.media:first-child{margin-top:0}.media-right,.media>.pull-  
right{padding-left:10px}.media-left,.media>.pull-left{padding-right:10px}.media-left,.media-  
right,.media-body{display:table-cell;vertical-align:top}.media-middle{vertical-align:middle}.media-  
bottom{vertical-align:bottom}.media-heading{margin-top:0;margin-bottom:5px}.media-list{padding-  
left:0;list-style:none}.list-group{margin-bottom:20px;padding-left:0}.list-group-  
item{position:relative;display:block;padding:10px 15px;margin-bottom:-1px;background-  
color:#fff;border:1px solid #ddd}.list-group-item:first-child{border-top-right-radius:2px;border-top-left-  
radius:2px}.list-group-item:last-child{margin-bottom:0;border-bottom-right-radius:2px;border-bottom-  
left-radius:2px}a.list-group-item{color:#555}a.list-group-item .list-group-item-heading{color:#333}a.list-  
group-item:hover,a.list-group-item:focus{text-decoration:none;color:#555;background-  
color:#f5f5f5}.list-group-item.disabled,.list-group-item.disabled:hover,.list-group-  
item.disabled:focus{background-color:#eee;color:#777;cursor:not-allowed}.list-group-item.disabled  
.list-group-item-heading,.list-group-item.disabled:hover .list-group-item-heading,.list-group-  
item.disabled:focus .list-group-item-heading{color:inherit}.list-group-item.disabled .list-group-item-  
text,.list-group-item.disabled:hover .list-group-item-text,.list-group-item.disabled:focus .list-group-item-  
text{color:#777}.list-group-item.active,.list-group-item.active:hover,.list-group-item.active:focus{z-  
index:2;color:#fff;background-color:#337ab7;border-color:#337ab7}.list-group-item.active .list-group-  
item-heading,.list-group-item.active:hover .list-group-item-heading,.list-group-item.active:focus .list-  
group-item-heading,.list-group-item.active .list-group-item-heading>small,.list-group-item.active:hover  
.list-group-item-heading>small,.list-group-item.active:focus .list-group-item-heading>small,.list-group-  
item.active .list-group-item-heading>.small,.list-group-item.active:hover .list-group-item-  
heading>.small,.list-group-item.active:focus .list-group-item-heading>.small{color:inherit}.list-group-  
item.active .list-group-item-text,.list-group-item.active:hover .list-group-item-text,.list-group-  
item.active:focus .list-group-item-text{color:#c7ddef}.list-group-item-  
success{color:#3c763d;background-color:#dff0d8}a.list-group-item-success{color:#3c763d}a.list-group-  
item-success .list-group-item-heading{color:inherit}a.list-group-item-success:hover,a.list-group-item-  
success:focus{color:#3c763d;background-color:#d0e9c6}a.list-group-item-success.active,a.list-group-  
item-success.active:hover,a.list-group-item-success.active:focus{color:#fff;background-  
color:#3c763d;border-color:#3c763d}.list-group-item-info{color:#31708f;background-  
color:#d9edf7}a.list-group-item-info{color:#31708f}a.list-group-item-info .list-group-item-  
heading{color:inherit}a.list-group-item-info:hover,a.list-group-item-

info:focus{color:#31708f;background-color:#c4e3f3}a.list-group-item-info.active,a.list-group-item-info.active:hover,a.list-group-item-info.active:focus{color:#fff;background-color:#31708f;border-color:#31708f}.list-group-item-warning{color:#8a6d3b;background-color:#fcf8e3}a.list-group-item-warning{color:#8a6d3b}a.list-group-item-warning .list-group-item-heading{color:inherit}a.list-group-item-warning:hover,a.list-group-item-warning:focus{color:#8a6d3b;background-color:#faf2cc}a.list-group-item-warning.active,a.list-group-item-warning.active:hover,a.list-group-item-warning.active:focus{color:#fff;background-color:#8a6d3b;border-color:#8a6d3b}.list-group-item-danger{color:#a94442;background-color:#f2dede}a.list-group-item-danger{color:#a94442}a.list-group-item-danger .list-group-item-heading{color:inherit}a.list-group-item-danger:hover,a.list-group-item-danger:focus{color:#a94442;background-color:#ebcccc}a.list-group-item-danger.active,a.list-group-item-danger.active:hover,a.list-group-item-danger.active:focus{color:#fff;background-color:#a94442;border-color:#a94442}.list-group-item-heading{margin-top:0;margin-bottom:5px}.list-group-item-text{margin-bottom:0;line-height:1.3}.panel{margin-bottom:18px;background-color:#fff;border:1px solid transparent;border-radius:2px;-webkit-box-shadow:0 1px 1px rgba(0,0,0,0.05);box-shadow:0 1px 1px rgba(0,0,0,0.05)}.panel-body{padding:15px}.panel-heading{padding:10px 15px;border-bottom:1px solid transparent;border-top-right-radius:1px;border-top-left-radius:1px}.panel-heading>.dropdown .dropdown-toggle{color:inherit}.panel-title{margin-top:0;margin-bottom:0;font-size:15px;color:inherit}.panel-title>a{color:inherit}.panel-footer{padding:10px 15px;background-color:#f5f5f5;border-top:1px solid #ddd;border-bottom-right-radius:1px;border-bottom-left-radius:1px}.panel>.list-group,.panel>.panel-collapse>.list-group{margin-bottom:0}.panel>.list-group .list-group-item,.panel>.panel-collapse>.list-group .list-group-item{border-width:1px 0;border-radius:0}.panel>.list-group:first-child .list-group-item:first-child,.panel>.panel-collapse>.list-group:first-child .list-group-item:first-child{border-top:0;border-top-right-radius:1px;border-top-left-radius:1px}.panel>.list-group:last-child .list-group-item:last-child,.panel>.panel-collapse>.list-group:last-child .list-group-item:last-child{border-bottom:0;border-bottom-right-radius:1px;border-bottom-left-radius:1px}.panel-heading+.list-group .list-group-item:first-child{border-top-width:0}.list-group+.panel-footer{border-top-width:0}.panel>.table,.panel>.table-responsive>.table,.panel>.panel-collapse>.table{margin-bottom:0}.panel>.table caption,.panel>.table-responsive>.table caption,.panel>.panel-collapse>.table caption{padding-left:15px;padding-right:15px}.panel>.table:first-child,.panel>.table-responsive:first-child>.table:first-child{border-top-right-radius:1px;border-top-left-radius:1px}.panel>.table:first-child>thead:first-child>tr:first-child,.panel>.table-responsive:first-child>.table:first-child>thead:first-child>tr:first-child,.panel>.table:first-child>tbody:first-child>tr:first-child,.panel>.table-responsive:first-child>.table:first-child>tbody:first-child>tr:first-child{border-top-left-radius:1px;border-top-right-radius:1px}.panel>.table:first-child>thead:first-child>tr:first-child td:first-child,.panel>.table-responsive:first-child>.table:first-child>thead:first-child>tr:first-child td:first-child,.panel>.table:first-child>tbody:first-child>tr:first-child td:first-child,.panel>.table-responsive:first-child>.table:first-child>tbody:first-child>tr:first-child th:first-child,.panel>.table-responsive:first-child>.table:first-child>thead:first-child>tr:first-child th:first-child,.panel>.table-responsive:first-child>.table:first-child>tbody:first-child>tr:first-child th:first-child{border-top-left-radius:1px}.panel>.table:first-child>thead:first-child>tr:first-child td:last-child,.panel>.table-responsive:first-child>.table:first-child>thead:first-child>tr:first-child td:last-child,.panel>.table:first-child>tbody:first-child>tr:first-child td:last-child,.panel>.table-responsive:first-child>.table:first-

[illegible]



child>td,.panel>.table-bordered>thead>tr:first-child>th,.panel>.table-responsive>.table-bordered>thead>tr:first-child>th,.panel>.table-bordered>tbody>tr:first-child>th,.panel>.table-responsive>.table-bordered>tbody>tr:first-child>th{border-bottom:0}.panel>.table-bordered>tbody>tr:last-child>td,.panel>.table-responsive>.table-bordered>tbody>tr:last-child>td,.panel>.table-bordered>tfoot>tr:last-child>td,.panel>.table-responsive>.table-bordered>tfoot>tr:last-child>td,.panel>.table-bordered>tbody>tr:last-child>th,.panel>.table-responsive>.table-bordered>tbody>tr:last-child>th,.panel>.table-bordered>tfoot>tr:last-child>th,.panel>.table-responsive>.table-bordered>tfoot>tr:last-child>th{border-bottom:0}.panel>.table-responsive{border:0;margin-bottom:0}.panel-group{margin-bottom:18px}.panel-group .panel{margin-bottom:0;border-radius:2px}.panel-group .panel+.panel{margin-top:5px}.panel-group .panel-heading{border-bottom:0}.panel-group .panel-heading+.panel-collapse>.panel-body,.panel-group .panel-heading+.panel-collapse>.list-group{border-top:1px solid #ddd}.panel-group .panel-footer{border-top:0}.panel-group .panel-footer+.panel-collapse>.panel-body{border-bottom:1px solid #ddd}.panel-default{border-color:#ddd}.panel-default>.panel-heading{color:#333;background-color:#f5f5f5;border-color:#ddd}.panel-default>.panel-heading+.panel-collapse>.panel-body{border-top-color:#ddd}.panel-default>.panel-heading .badge{color:#f5f5f5;background-color:#333}.panel-default>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#ddd}.panel-primary{border-color:#337ab7}.panel-primary>.panel-heading{color:#fff;background-color:#337ab7;border-color:#337ab7}.panel-primary>.panel-heading+.panel-collapse>.panel-body{border-top-color:#337ab7}.panel-primary>.panel-heading .badge{color:#337ab7;background-color:#fff}.panel-primary>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#337ab7}.panel-success{border-color:#d6e9c6}.panel-success>.panel-heading{color:#3c763d;background-color:#dff0d8;border-color:#d6e9c6}.panel-success>.panel-heading+.panel-collapse>.panel-body{border-top-color:#d6e9c6}.panel-success>.panel-heading .badge{color:#dff0d8;background-color:#3c763d}.panel-success>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#d6e9c6}.panel-info{border-color:#bce8f1}.panel-info>.panel-heading{color:#31708f;background-color:#d9edf7;border-color:#bce8f1}.panel-info>.panel-heading+.panel-collapse>.panel-body{border-top-color:#bce8f1}.panel-info>.panel-heading .badge{color:#d9edf7;background-color:#31708f}.panel-info>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#bce8f1}.panel-warning{border-color:#faebcc}.panel-warning>.panel-heading{color:#8a6d3b;background-color:#fcf8e3;border-color:#faebcc}.panel-warning>.panel-heading+.panel-collapse>.panel-body{border-top-color:#faebcc}.panel-warning>.panel-heading .badge{color:#fcf8e3;background-color:#8a6d3b}.panel-warning>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#faebcc}.panel-danger{border-color:#ebcc11}.panel-danger>.panel-heading{color:#a94442;background-color:#f2dede;border-color:#ebcc11}.panel-danger>.panel-heading+.panel-collapse>.panel-body{border-top-color:#ebcc11}.panel-danger>.panel-heading .badge{color:#f2dede;background-color:#a94442}.panel-danger>.panel-footer+.panel-collapse>.panel-body{border-bottom-color:#ebcc11}.embed-responsive{position:relative;display:block;height:0;padding:0;overflow:hidden}.embed-responsive .embed-responsive-item,.embed-responsive iframe,.embed-responsive embed,.embed-responsive object,.embed-responsive video{position:absolute;top:0;left:0;bottom:0;height:100%;width:100%;border:0}.embed-responsive.embed-responsive-16by9{padding-bottom:56.25%}.embed-responsive.embed-responsive-4by3{padding-bottom:75%}.well{min-height:20px;padding:19px;margin-bottom:20px;background-

color:#f5f5f5;border:1px solid #e3e3e3;border-radius:2px;-webkit-box-shadow:inset 0 1px 1px  
rgba(0,0,0,0.05);box-shadow:inset 0 1px 1px rgba(0,0,0,0.05)}.well blockquote{border-  
color:#ddd;border-color:rgba(0,0,0,0.15)}.well-lg{padding:24px;border-radius:3px}.well-  
sm{padding:9px;border-radius:1px}.close{float:right;font-size:19.5px;font-weight:bold;line-  
height:1;color:#000;text-shadow:0 1px 0  
#fff;opacity:.2;filter:alpha(opacity=20)}.close:hover,.close:focus{color:#000;text-  
decoration:none;cursor:pointer;opacity:.5;filter:alpha(opacity=50)}button.close{padding:0;cursor:pointe  
r;background:transparent;border:0;-webkit-appearance:none}.modal-  
open{overflow:hidden}.modal{display:none;overflow:hidden;position:fixed;top:0;right:0;bottom:0;left:0  
;z-index:1040;-webkit-overflow-scrolling:touch;outline:0}.modal.fade .modal-dialog{-webkit-  
transform:translate(0, -25%);-ms-transform:translate(0, -25%);-o-transform:translate(0, -  
25%);transform:translate(0, -25%);-webkit-transition:-webkit-transform 0.3s ease-out;-moz-transition:-  
moz-transform 0.3s ease-out;-o-transition:-o-transform 0.3s ease-out;transition:transform 0.3s ease-  
out}.modal.in .modal-dialog{-webkit-transform:translate(0, 0);-ms-transform:translate(0, 0);-o-  
transform:translate(0, 0);transform:translate(0, 0)}.modal-open .modal{overflow-x:hidden;overflow-  
y:auto}.modal-dialog{position:relative;width:auto;margin:10px}.modal-  
content{position:relative;background-color:#fff;border:1px solid #999;border:1px solid  
rgba(0,0,0,0.2);border-radius:3px;-webkit-box-shadow:0 3px 9px rgba(0,0,0,0.5);box-shadow:0 3px 9px  
rgba(0,0,0,0.5);background-clip:padding-box;outline:0}.modal-  
backdrop{position:absolute;top:0;right:0;left:0;background-color:#000}.modal-  
backdrop.fade{opacity:0;filter:alpha(opacity=0)}.modal-  
backdrop.in{opacity:.5;filter:alpha(opacity=50)}.modal-header{padding:15px;border-bottom:1px solid  
#e5e5e5;min-height:16.42857143px}.modal-header .close{margin-top:-2px}.modal-title{margin:0;line-  
height:1.42857143}.modal-body{position:relative;padding:15px}.modal-footer{padding:15px;text-  
align:right;border-top:1px solid #e5e5e5}.modal-footer .btn+.btn{margin-left:5px;margin-  
bottom:0}.modal-footer .btn-group .btn+.btn{margin-left:-1px}.modal-footer .btn-block+.btn-  
block{margin-left:0}.modal-scrollbar-measure{position:absolute;top:-  
9999px;width:50px;height:50px;overflow:scroll}@media (min-width:768px){.modal-  
dialog{width:600px;margin:30px auto}.modal-content{-webkit-box-shadow:0 5px 15px  
rgba(0,0,0,0.5);box-shadow:0 5px 15px rgba(0,0,0,0.5)}.modal-sm{width:300px}}@media (min-  
width:992px){.modal-lg{width:900px}}.tooltip{position:absolute;z-  
index:1070;display:block;visibility:visible;font-family:"Helvetica Neue",Helvetica,Arial,sans-serif;font-  
size:12px;font-weight:normal;line-  
height:1.4;opacity:0;filter:alpha(opacity=0)}.tooltip.in{opacity:.9;filter:alpha(opacity=90)}.tooltip.top{ma  
rgin-top:-3px;padding:5px 0}.tooltip.right{margin-left:3px;padding:0 5px}.tooltip.bottom{margin-  
top:3px;padding:5px 0}.tooltip.left{margin-left:-3px;padding:0 5px}.tooltip-inner{max-  
width:200px;padding:3px 8px;color:#fff;text-align:center;text-decoration:none;background-  
color:#000;border-radius:2px}.tooltip-arrow{position:absolute;width:0;height:0;border-  
color:transparent;border-style:solid}.tooltip.top .tooltip-arrow{bottom:0;left:50%;margin-left:-  
5px;border-width:5px 5px 0;border-top-color:#000}.tooltip.top-left .tooltip-  
arrow{bottom:0;right:5px;margin-bottom:-5px;border-width:5px 5px 0;border-top-  
color:#000}.tooltip.top-right .tooltip-arrow{bottom:0;left:5px;margin-bottom:-5px;border-width:5px 5px  
0;border-top-color:#000}.tooltip.right .tooltip-arrow{top:50%;left:0;margin-top:-5px;border-width:5px  
5px 0;border-right-color:#000}.tooltip.left .tooltip-arrow{top:50%;right:0;margin-top:-5px;border-

```

width:5px 0 5px 5px;border-left-color:#000}.tooltip.bottom .tooltip-arrow{top:0;left:50%;margin-left:-5px;border-width:0 5px 5px;border-bottom-color:#000}.tooltip.bottom-left .tooltip-arrow{top:0;right:5px;margin-top:-5px;border-width:0 5px 5px;border-bottom-color:#000}.tooltip.bottom-right .tooltip-arrow{top:0;left:5px;margin-top:-5px;border-width:0 5px 5px;border-bottom-color:#000}.popover{position:absolute;top:0;left:0;z-index:1060;display:none;max-width:276px;padding:1px;font-family:"Helvetica Neue",Helvetica,Arial,sans-serif;font-size:13px;font-weight:normal;line-height:1.42857143;text-align:left;background-color:#fff;background-clip:padding-box;border:1px solid #ccc;border:1px solid rgba(0,0,0,0.2);border-radius:3px;-webkit-box-shadow:0 5px 10px rgba(0,0,0,0.2);box-shadow:0 5px 10px rgba(0,0,0,0.2);white-space:normal}.popover.top{margin-top:-10px}.popover.right{margin-left:10px}.popover.bottom{margin-top:10px}.popover.left{margin-left:-10px}.popover-title{margin:0;padding:8px 14px;font-size:13px;background-color:#f7f7f7;border-bottom:1px solid #ebebeb;border-radius:2px 2px 0 0}.popover-content{padding:9px 14px}.popover>.arrow,.popover>.arrow:after{position:absolute;display:block;width:0;height:0;border-color:transparent;border-style:solid}.popover>.arrow{border-width:11px}.popover>.arrow:after{border-width:10px;content:""}.popover.top>.arrow{left:50%;margin-left:-11px;border-bottom-width:0;border-top-color:#999;border-top-color:rgba(0,0,0,0.25);bottom:-11px}.popover.top>.arrow:after{content:"";bottom:1px;margin-left:-10px;border-bottom-width:0;border-top-color:#fff}.popover.right>.arrow{top:50%;left:-11px;margin-top:-11px;border-left-width:0;border-right-color:#999;border-right-color:rgba(0,0,0,0.25)}.popover.right>.arrow:after{content:" ";left:1px;bottom:-10px;border-left-width:0;border-right-color:#fff}.popover.bottom>.arrow{left:50%;margin-left:-11px;border-top-width:0;border-bottom-color:#999;border-bottom-color:rgba(0,0,0,0.25);top:-11px}.popover.bottom>.arrow:after{content:" ";top:1px;margin-left:-10px;border-top-width:0;border-bottom-color:#fff}.popover.left>.arrow{top:50%;right:-11px;margin-top:-11px;border-right-width:0;border-left-color:#999;border-left-color:rgba(0,0,0,0.25)}.popover.left>.arrow:after{content:" ";right:1px;border-right-width:0;border-left-color:#fff;bottom:-10px}.carousel{position:relative}.carousel-inner{position:relative;overflow:hidden;width:100%}.carousel-inner>.item{display:none;position:relative;-webkit-transition:.6s ease-in-out left;o-transition:.6s ease-in-out left;transition:.6s ease-in-out left}.carousel-inner>.item>img,.carousel-inner>.item>a>img{line-height:1}@media all and (transform-3d),(-webkit-transform-3d){.carousel-inner>.item{transition:transform .6s ease-in-out;backface-visibility:hidden;perspective:1000}.carousel-inner>.item.next,.carousel-inner>.item.active.right{transform:translate3d(100%, 0, 0);left:0}.carousel-inner>.item.prev,.carousel-inner>.item.active.left{transform:translate3d(-100%, 0, 0);left:0}.carousel-inner>.item.next.left,.carousel-inner>.item.prev.right,.carousel-inner>.item.active{transform:translate3d(0, 0, 0);left:0}}.carousel-inner>.active,.carousel-inner>.next,.carousel-inner>.prev{display:block}.carousel-inner>.active{left:0}.carousel-inner>.next,.carousel-inner>.prev{position:absolute;top:0;width:100%}.carousel-inner>.next{left:100%}.carousel-inner>.prev{left:-100%}.carousel-inner>.next.left,.carousel-inner>.prev.right{left:0}.carousel-inner>.active.left{left:-100%}.carousel-inner>.active.right{left:100%}.carousel-control{position:absolute;top:0;left:0;bottom:0;width:15%;opacity:.5;filter:alpha(opacity=50);font-size:20px;color:#fff;text-align:center;text-shadow:0 1px 2px rgba(0,0,0,0.6)}.carousel-control.left{background-image:-webkit-linear-gradient(left, rgba(0,0,0,0.5) 0, rgba(0,0,0,0.0001) 100%);background-image:-o-linear-gradient(left, rgba(0,0,0,0.5) 0, rgba(0,0,0,0.0001)

```

```

100%);background-image:linear-gradient(to right, rgba(0,0,0,0.5) 0, rgba(0,0,0,0.0001)
100%);background-repeat:repeat-
x;filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#80000000',
endColorstr='#00000000', GradientType=1)}.carousel-control.right{left:auto;right:0;background-image:-
webkit-linear-gradient(left, rgba(0,0,0,0.0001) 0, rgba(0,0,0,0.5) 100%);background-image:-o-linear-
gradient(left, rgba(0,0,0,0.0001) 0, rgba(0,0,0,0.5) 100%);background-image:linear-gradient(to right,
rgba(0,0,0,0.0001) 0, rgba(0,0,0,0.5) 100%);background-repeat:repeat-
x;filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#00000000',
endColorstr='#80000000', GradientType=1)}.carousel-control:hover,.carousel-
control:focus{outline:0;color:#fff;text-decoration:none;opacity:.9;filter:alpha(opacity=90)}.carousel-
control .icon-prev,.carousel-control .icon-next,.carousel-control .glyphicon-chevron-left,.carousel-
control .glyphicon-chevron-right{position:absolute;top:50%;z-index:5;display:inline-block}.carousel-
control .icon-prev,.carousel-control .glyphicon-chevron-left{left:50%;margin-left:-10px}.carousel-control
.icon-next,.carousel-control .glyphicon-chevron-right{right:50%;margin-right:-10px}.carousel-control
.icon-prev,.carousel-control .icon-next{width:20px;height:20px;margin-top:-10px;font-
family:serif}.carousel-control .icon-prev:before{content:'\2039'}.carousel-control .icon-
next:before{content:'\203a'}.carousel-indicators{position:absolute;bottom:10px;left:50%;z-
index:15;width:60%;margin-left:-30%;padding-left:0;list-style:none;text-align:center}.carousel-indicators
li{display:inline-block;width:10px;height:10px;margin:1px;text-indent:-999px;border:1px solid
#fff;border-radius:10px;cursor:pointer;background-color:#000 \9;background-
color:rgba(0,0,0,0)}.carousel-indicators .active{margin:0;width:12px;height:12px;background-
color:#fff}.carousel-caption{position:absolute;left:15%;right:15%;bottom:20px;z-index:10;padding-
top:20px;padding-bottom:20px;color:#fff;text-align:center;text-shadow:0 1px 2px
rgba(0,0,0,0.6)}.carousel-caption .btn{text-shadow:none}@media screen and (min-
width:768px){.carousel-control .glyphicon-chevron-left,.carousel-control .glyphicon-chevron-
right,.carousel-control .icon-prev,.carousel-control .icon-next{width:30px;height:30px;margin-top:-
15px;font-size:30px}.carousel-control .glyphicon-chevron-left,.carousel-control .icon-prev{margin-left:-
15px}.carousel-control .glyphicon-chevron-right,.carousel-control .icon-next{margin-right:-
15px}.carousel-caption{left:20%;right:20%;padding-bottom:30px}.carousel-
indicators{bottom:20px}}.clearfix:before,.clearfix:after,.dl-horizontal dd:before,.dl-horizontal
dd:after,.container:before,.container:after,.container-fluid:before,.container-
fluid:after,.row:before,.row:after,.form-horizontal .form-group:before,.form-horizontal .form-
group:after,.btn-toolbar:before,.btn-toolbar:after,.btn-group-vertical>.btn-group:before,.btn-group-
vertical>.btn-group:after,.nav:before,.nav:after,.navbar:before,.navbar:after,.navbar-
header:before,.navbar-header:after,.navbar-collapse:before,.navbar-
collapse:after,.pager:before,.pager:after,.panel-body:before,.panel-body:after,.modal-
footer:before,.modal-footer:after,.item_buttons:before,.item_buttons:after{content:"
";display:table}.clearfix:after,.dl-horizontal dd:after,.container:after,.container-
fluid:after,.row:after,.form-horizontal .form-group:after,.btn-toolbar:after,.btn-group-vertical>.btn-
group:after,.nav:after,.navbar:after,.navbar-header:after,.navbar-collapse:after,.pager:after,.panel-
body:after,.modal-footer:after,.item_buttons:after{clear:both}.center-block{display:block;margin-
left:auto;margin-right:auto}.pull-right{float:right !important}.pull-left{float:left
!important}.hide{display:none !important}.show{display:block
!important}.invisible{visibility:hidden}.text-hide{font:0/0 a;color:transparent;text-

```

```

shadow:none;background-color:transparent;border:0}.hidden{display:none !important;visibility:hidden !important}.affix{position:fixed}@-ms-viewport{width:device-width}.visible-xs,.visible-sm,.visible-md,.visible-lg{display:none !important}.visible-xs-block,.visible-xs-inline,.visible-xs-inline-block,.visible-sm-block,.visible-sm-inline,.visible-sm-inline-block,.visible-md-block,.visible-md-inline,.visible-md-inline-block,.visible-lg-block,.visible-lg-inline,.visible-lg-inline-block{display:none !important}@media (max-width:767px){.visible-xs{display:block !important}table.visible-xs{display:table}tr.visible-xs{display:table-row !important}th.visible-xs,td.visible-xs{display:table-cell !important}}@media (max-width:767px){.visible-xs-block{display:block !important}}@media (max-width:767px){.visible-xs-inline{display:inline !important}}@media (max-width:767px){.visible-xs-inline-block{display:inline-block !important}}@media (min-width:768px) and (max-width:991px){.visible-sm{display:block !important}table.visible-sm{display:table}tr.visible-sm{display:table-row !important}th.visible-sm,td.visible-sm{display:table-cell !important}}@media (min-width:768px) and (max-width:991px){.visible-sm-block{display:block !important}}@media (min-width:768px) and (max-width:991px){.visible-sm-inline{display:inline !important}}@media (min-width:768px) and (max-width:991px){.visible-sm-inline-block{display:inline-block !important}}@media (min-width:992px) and (max-width:1199px){.visible-md{display:block !important}table.visible-md{display:table}tr.visible-md{display:table-row !important}th.visible-md,td.visible-md{display:table-cell !important}}@media (min-width:992px) and (max-width:1199px){.visible-md-block{display:block !important}}@media (min-width:992px) and (max-width:1199px){.visible-md-inline{display:inline !important}}@media (min-width:992px) and (max-width:1199px){.visible-md-inline-block{display:inline-block !important}}@media (min-width:1200px){.visible-lg{display:block !important}table.visible-lg{display:table}tr.visible-lg{display:table-row !important}th.visible-lg,td.visible-lg{display:table-cell !important}}@media (min-width:1200px){.visible-lg-block{display:block !important}}@media (min-width:1200px){.visible-lg-inline{display:inline !important}}@media (min-width:1200px){.visible-lg-inline-block{display:inline-block !important}}@media (max-width:767px){.hidden-xs{display:none !important}}@media (min-width:768px) and (max-width:991px){.hidden-sm{display:none !important}}@media (min-width:992px) and (max-width:1199px){.hidden-md{display:none !important}}@media (min-width:1200px){.hidden-lg{display:none !important}}.visible-print{display:none !important}@media print{.visible-print{display:block !important}table.visible-print{display:table}tr.visible-print{display:table-row !important}th.visible-print,td.visible-print{display:table-cell !important}.visible-print-block{display:none !important}@media print{.visible-print-block{display:block !important}}.visible-print-inline{display:none !important}@media print{.visible-print-inline{display:inline !important}}.visible-print-inline-block{display:none !important}@media print{.visible-print-inline-block{display:inline-block !important}}@media print{.hidden-print{display:none !important}}/*!

```

\*

\* Font Awesome

\*

\*//\*!

\* Font Awesome 4.3.0 by @davegandy - <http://fontawesome.io> - @fontawesome

\* License - <http://fontawesome.io/license> (Font: SIL OFL 1.1, CSS: MIT License)

```

*/*font-face{font-family:'FontAwesome';src:url('../components/font-awesome/fonts/fontawesome-webfont.eot?v=4.3.0');src:url('../components/font-awesome/fonts/fontawesome-webfont.eot?#iefix&v=4.3.0') format('embedded-opentype'),url('../components/font-awesome/fonts/fontawesome-webfont.woff2?v=4.3.0') format('woff2'),url('../components/font-awesome/fonts/fontawesome-webfont.woff?v=4.3.0') format('woff'),url('../components/font-awesome/fonts/fontawesome-webfont.ttf?v=4.3.0') format('truetype'),url('../components/font-awesome/fonts/fontawesome-webfont.svg?v=4.3.0#fontawesomeregular') format('svg');font-weight:normal;font-style:normal}.fa{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0)}.fa-lg{font-size:1.33333333em;line-height:.75em;vertical-align:-15%}.fa-2x{font-size:2em}.fa-3x{font-size:3em}.fa-4x{font-size:4em}.fa-5x{font-size:5em}.fa-fw{width:1.28571429em;text-align:center}.fa-ul{padding-left:0;margin-left:2.14285714em;list-style-type:none}.fa-ul>li{position:relative}.fa-li{position:absolute;left:-2.14285714em;width:2.14285714em;top:.14285714em;text-align:center}.fa-li.fa-lg{left:-1.85714286em}.fa-border{padding:.2em .25em .15em;border:solid .08em #eee;border-radius:.1em}.pull-right{float:right}.pull-left{float:left}.fa.pull-left{margin-right:.3em}.fa.pull-right{margin-left:.3em}.fa-spin{-webkit-animation:fa-spin 2s infinite linear;animation:fa-spin 2s infinite linear}.fa-pulse{-webkit-animation:fa-spin 1s infinite steps(8);animation:fa-spin 1s infinite steps(8)}@-webkit-keyframes fa-spin{0%{-webkit-transform:rotate(0deg);transform:rotate(0deg)}100%{-webkit-transform:rotate(359deg);transform:rotate(359deg)}}@keyframes fa-spin{0%{-webkit-transform:rotate(0deg);transform:rotate(0deg)}100%{-webkit-transform:rotate(359deg);transform:rotate(359deg)}}.fa-rotate-90{filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=1);-webkit-transform:rotate(90deg);-ms-transform:rotate(90deg);transform:rotate(90deg)}.fa-rotate-180{filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=2);-webkit-transform:rotate(180deg);-ms-transform:rotate(180deg);transform:rotate(180deg)}.fa-rotate-270{filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=3);-webkit-transform:rotate(270deg);-ms-transform:rotate(270deg);transform:rotate(270deg)}.fa-flip-horizontal{filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=0, mirror=1);-webkit-transform:scale(-1, 1);-ms-transform:scale(-1, 1);transform:scale(-1, 1)}.fa-flip-vertical{filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=2, mirror=1);-webkit-transform:scale(1, -1);-ms-transform:scale(1, -1);transform:scale(1, -1)}:root .fa-rotate-90,:root .fa-rotate-180,:root .fa-rotate-270,:root .fa-flip-horizontal,:root .fa-flip-vertical{filter:none}.fa-stack{position:relative;display:inline-block;width:2em;height:2em;line-height:2em;vertical-align:middle}.fa-stack-1x,.fa-stack-2x{position:absolute;left:0;width:100%;text-align:center}.fa-stack-1x{line-height:inherit}.fa-stack-2x{font-size:2em}.fa-inverse{color:#fff}.fa-glass:before{content:"\f000"}.fa-music:before{content:"\f001"}.fa-search:before{content:"\f002"}.fa-envelope-o:before{content:"\f003"}.fa-heart:before{content:"\f004"}.fa-star:before{content:"\f005"}.fa-star-o:before{content:"\f006"}.fa-user:before{content:"\f007"}.fa-film:before{content:"\f008"}.fa-th-large:before{content:"\f009"}.fa-th:before{content:"\f00a"}.fa-th-list:before{content:"\f00b"}.fa-check:before{content:"\f00c"}.fa-remove:before,.fa-close:before,.fa-times:before{content:"\f00d"}.fa-search-plus:before{content:"\f00e"}.fa-search-minus:before{content:"\f010"}.fa-power-off:before{content:"\f011"}.fa-signal:before{content:"\f012"}.fa-gear:before,.fa-cog:before{content:"\f013"}.fa-trash-o:before{content:"\f014"}.fa-home:before{content:"\f015"}.fa-file-

```

o:before{content:"\f016"}.fa-clock-o:before{content:"\f017"}.fa-road:before{content:"\f018"}.fa-download:before{content:"\f019"}.fa-arrow-circle-o-down:before{content:"\f01a"}.fa-arrow-circle-o-up:before{content:"\f01b"}.fa-inbox:before{content:"\f01c"}.fa-play-circle-o:before{content:"\f01d"}.fa-rotate-right:before{content:"\f01e"}.fa-repeat:before{content:"\f01e"}.fa-refresh:before{content:"\f021"}.fa-list-alt:before{content:"\f022"}.fa-lock:before{content:"\f023"}.fa-flag:before{content:"\f024"}.fa-headphones:before{content:"\f025"}.fa-volume-off:before{content:"\f026"}.fa-volume-down:before{content:"\f027"}.fa-volume-up:before{content:"\f028"}.fa-qr-code:before{content:"\f029"}.fa-barcode:before{content:"\f02a"}.fa-tag:before{content:"\f02b"}.fa-tags:before{content:"\f02c"}.fa-book:before{content:"\f02d"}.fa-bookmark:before{content:"\f02e"}.fa-print:before{content:"\f02f"}.fa-camera:before{content:"\f030"}.fa-font:before{content:"\f031"}.fa-bold:before{content:"\f032"}.fa-italic:before{content:"\f033"}.fa-text-height:before{content:"\f034"}.fa-text-width:before{content:"\f035"}.fa-align-left:before{content:"\f036"}.fa-align-center:before{content:"\f037"}.fa-align-right:before{content:"\f038"}.fa-align-justify:before{content:"\f039"}.fa-list:before{content:"\f03a"}.fa-dedent:before{content:"\f03b"}.fa-indent:before{content:"\f03c"}.fa-video-camera:before{content:"\f03d"}.fa-photo:before{content:"\f03e"}.fa-pencil:before{content:"\f040"}.fa-map-marker:before{content:"\f041"}.fa-adjust:before{content:"\f042"}.fa-tint:before{content:"\f043"}.fa-edit:before{content:"\f044"}.fa-share-square-o:before{content:"\f045"}.fa-check-square-o:before{content:"\f046"}.fa-arrows:before{content:"\f047"}.fa-step-backward:before{content:"\f048"}.fa-fast-backward:before{content:"\f049"}.fa-backward:before{content:"\f04a"}.fa-play:before{content:"\f04b"}.fa-pause:before{content:"\f04c"}.fa-stop:before{content:"\f04d"}.fa-forward:before{content:"\f04e"}.fa-fast-forward:before{content:"\f050"}.fa-step-forward:before{content:"\f051"}.fa-eject:before{content:"\f052"}.fa-chevron-left:before{content:"\f053"}.fa-chevron-right:before{content:"\f054"}.fa-plus-circle:before{content:"\f055"}.fa-minus-circle:before{content:"\f056"}.fa-times-circle:before{content:"\f057"}.fa-check-circle:before{content:"\f058"}.fa-question-circle:before{content:"\f059"}.fa-info-circle:before{content:"\f05a"}.fa-crosshairs:before{content:"\f05b"}.fa-times-circle-o:before{content:"\f05c"}.fa-check-circle-o:before{content:"\f05d"}.fa-ban:before{content:"\f05e"}.fa-arrow-left:before{content:"\f060"}.fa-arrow-right:before{content:"\f061"}.fa-arrow-up:before{content:"\f062"}.fa-arrow-down:before{content:"\f063"}.fa-mail-forward:before{content:"\f064"}.fa-expand:before{content:"\f065"}.fa-compress:before{content:"\f066"}.fa-plus:before{content:"\f067"}.fa-minus:before{content:"\f068"}.fa-asterisk:before{content:"\f069"}.fa-exclamation-circle:before{content:"\f06a"}.fa-gift:before{content:"\f06b"}.fa-leaf:before{content:"\f06c"}.fa-fire:before{content:"\f06d"}.fa-eye:before{content:"\f06e"}.fa-eye-slash:before{content:"\f070"}.fa-warning:before{content:"\f071"}.fa-exclamation-triangle:before{content:"\f071"}.fa-plane:before{content:"\f072"}.fa-calendar:before{content:"\f073"}.fa-random:before{content:"\f074"}.fa-comment:before{content:"\f075"}.fa-magnet:before{content:"\f076"}.fa-chevron-up:before{content:"\f077"}.fa-chevron-down:before{content:"\f078"}.fa-retweet:before{content:"\f079"}.fa-shopping-cart:before{content:"\f07a"}.fa-folder:before{content:"\f07b"}.fa-folder-open:before{content:"\f07c"}.fa-arrows-v:before{content:"\f07d"}.fa-arrows-h:before{content:"\f07e"}.fa-bar-chart-o:before{content:"\f07e"}.fa-bar-

chart:before{content:"\f080"}.fa-twitter-square:before{content:"\f081"}.fa-facebook-square:before{content:"\f082"}.fa-camera-retro:before{content:"\f083"}.fa-key:before{content:"\f084"}.fa-gears:before,.fa-cogs:before{content:"\f085"}.fa-comments:before{content:"\f086"}.fa-thumbs-o-up:before{content:"\f087"}.fa-thumbs-o-down:before{content:"\f088"}.fa-star-half:before{content:"\f089"}.fa-heart-o:before{content:"\f08a"}.fa-sign-out:before{content:"\f08b"}.fa-linkedin-square:before{content:"\f08c"}.fa-thumb-tack:before{content:"\f08d"}.fa-external-link:before{content:"\f08e"}.fa-sign-in:before{content:"\f090"}.fa-trophy:before{content:"\f091"}.fa-github-square:before{content:"\f092"}.fa-upload:before{content:"\f093"}.fa-lemon-o:before{content:"\f094"}.fa-phone:before{content:"\f095"}.fa-square-o:before{content:"\f096"}.fa-bookmark-o:before{content:"\f097"}.fa-phone-square:before{content:"\f098"}.fa-twitter:before{content:"\f099"}.fa-facebook-f:before,.fa-facebook:before{content:"\f09a"}.fa-github:before{content:"\f09b"}.fa-unlock:before{content:"\f09c"}.fa-credit-card:before{content:"\f09d"}.fa-rss:before{content:"\f09e"}.fa-hdd-o:before{content:"\f0a0"}.fa-bullhorn:before{content:"\f0a1"}.fa-bell:before{content:"\f0f3"}.fa-certificate:before{content:"\f0a3"}.fa-hand-o-right:before{content:"\f0a4"}.fa-hand-o-left:before{content:"\f0a5"}.fa-hand-o-up:before{content:"\f0a6"}.fa-hand-o-down:before{content:"\f0a7"}.fa-arrow-circle-left:before{content:"\f0a8"}.fa-arrow-circle-right:before{content:"\f0a9"}.fa-arrow-circle-up:before{content:"\f0aa"}.fa-arrow-circle-down:before{content:"\f0ab"}.fa-globe:before{content:"\f0ac"}.fa-wrench:before{content:"\f0ad"}.fa-tasks:before{content:"\f0ae"}.fa-filter:before{content:"\f0b0"}.fa-briefcase:before{content:"\f0b1"}.fa-arrows-alt:before{content:"\f0b2"}.fa-group:before,.fa-users:before{content:"\f0c0"}.fa-chain:before,.fa-link:before{content:"\f0c1"}.fa-cloud:before{content:"\f0c2"}.fa-flask:before{content:"\f0c3"}.fa-cut:before,.fa-scissors:before{content:"\f0c4"}.fa-copy:before,.fa-files-o:before{content:"\f0c5"}.fa-paperclip:before{content:"\f0c6"}.fa-save:before,.fa-floppy-o:before{content:"\f0c7"}.fa-square:before{content:"\f0c8"}.fa-navicon:before,.fa-reorder:before,.fa-bars:before{content:"\f0c9"}.fa-list-ul:before{content:"\f0ca"}.fa-list-ol:before{content:"\f0cb"}.fa-strikethrough:before{content:"\f0cc"}.fa-underline:before{content:"\f0cd"}.fa-table:before{content:"\f0ce"}.fa-magic:before{content:"\f0d0"}.fa-truck:before{content:"\f0d1"}.fa-pinterest:before{content:"\f0d2"}.fa-pinterest-square:before{content:"\f0d3"}.fa-google-plus-square:before{content:"\f0d4"}.fa-google-plus:before{content:"\f0d5"}.fa-money:before{content:"\f0d6"}.fa-caret-down:before{content:"\f0d7"}.fa-caret-up:before{content:"\f0d8"}.fa-caret-left:before{content:"\f0d9"}.fa-caret-right:before{content:"\f0da"}.fa-columns:before{content:"\f0db"}.fa-unsorted:before,.fa-sort:before{content:"\f0dc"}.fa-sort-down:before,.fa-sort-desc:before{content:"\f0dd"}.fa-sort-up:before,.fa-sort-asc:before{content:"\f0de"}.fa-envelope:before{content:"\f0e0"}.fa-linkedin:before{content:"\f0e1"}.fa-rotate-left:before,.fa-undo:before{content:"\f0e2"}.fa-legal:before,.fa-gavel:before{content:"\f0e3"}.fa-dashboard:before,.fa-tachometer:before{content:"\f0e4"}.fa-comment-o:before{content:"\f0e5"}.fa-comments-o:before{content:"\f0e6"}.fa-flash:before,.fa-bolt:before{content:"\f0e7"}.fa-sitemap:before{content:"\f0e8"}.fa-umbrella:before{content:"\f0e9"}.fa-paste:before,.fa-clipboard:before{content:"\f0ea"}.fa-lightbulb-o:before{content:"\f0eb"}.fa-exchange:before{content:"\f0ec"}.fa-cloud-download:before{content:"\f0ed"}.fa-cloud-upload:before{content:"\f0ee"}.fa-user-md:before{content:"\f0f0"}.fa-



stethoscope:before{content:"\f0f1"}.fa-suitcase:before{content:"\f0f2"}.fa-bell-o:before{content:"\f0a2"}.fa-coffee:before{content:"\f0f4"}.fa-cutlery:before{content:"\f0f5"}.fa-file-text-o:before{content:"\f0f6"}.fa-building-o:before{content:"\f0f7"}.fa-hospital-o:before{content:"\f0f8"}.fa-ambulance:before{content:"\f0f9"}.fa-medkit:before{content:"\f0fa"}.fa-fighter-jet:before{content:"\f0fb"}.fa-beer:before{content:"\f0fc"}.fa-h-square:before{content:"\f0fd"}.fa-plus-square:before{content:"\f0fe"}.fa-angle-double-left:before{content:"\f100"}.fa-angle-double-right:before{content:"\f101"}.fa-angle-double-up:before{content:"\f102"}.fa-angle-double-down:before{content:"\f103"}.fa-angle-left:before{content:"\f104"}.fa-angle-right:before{content:"\f105"}.fa-angle-up:before{content:"\f106"}.fa-angle-down:before{content:"\f107"}.fa-desktop:before{content:"\f108"}.fa-laptop:before{content:"\f109"}.fa-tablet:before{content:"\f10a"}.fa-mobile-phone:before,.fa-mobile:before{content:"\f10b"}.fa-circle-o:before{content:"\f10c"}.fa-quote-left:before{content:"\f10d"}.fa-quote-right:before{content:"\f10e"}.fa-spinner:before{content:"\f110"}.fa-circle:before{content:"\f111"}.fa-mail-reply:before,.fa-reply:before{content:"\f112"}.fa-github-alt:before{content:"\f113"}.fa-folder-o:before{content:"\f114"}.fa-folder-open-o:before{content:"\f115"}.fa-smile-o:before{content:"\f118"}.fa-frown-o:before{content:"\f119"}.fa-meh-o:before{content:"\f11a"}.fa-gamepad:before{content:"\f11b"}.fa-keyboard-o:before{content:"\f11c"}.fa-flag-o:before{content:"\f11d"}.fa-flag-checkered:before{content:"\f11e"}.fa-terminal:before{content:"\f120"}.fa-code:before{content:"\f121"}.fa-mail-reply-all:before,.fa-reply-all:before{content:"\f122"}.fa-star-half-empty:before,.fa-star-half-full:before,.fa-star-half-o:before{content:"\f123"}.fa-location-arrow:before{content:"\f124"}.fa-crop:before{content:"\f125"}.fa-code-fork:before{content:"\f126"}.fa-unlink:before,.fa-chain-broken:before{content:"\f127"}.fa-question:before{content:"\f128"}.fa-info:before{content:"\f129"}.fa-exclamation:before{content:"\f12a"}.fa-superscript:before{content:"\f12b"}.fa-subscript:before{content:"\f12c"}.fa-eraser:before{content:"\f12d"}.fa-puzzle-piece:before{content:"\f12e"}.fa-microphone:before{content:"\f130"}.fa-microphone-slash:before{content:"\f131"}.fa-shield:before{content:"\f132"}.fa-calendar-o:before{content:"\f133"}.fa-fire-extinguisher:before{content:"\f134"}.fa-rocket:before{content:"\f135"}.fa-maxcdn:before{content:"\f136"}.fa-chevron-circle-left:before{content:"\f137"}.fa-chevron-circle-right:before{content:"\f138"}.fa-chevron-circle-up:before{content:"\f139"}.fa-chevron-circle-down:before{content:"\f13a"}.fa-html5:before{content:"\f13b"}.fa-css3:before{content:"\f13c"}.fa-anchor:before{content:"\f13d"}.fa-unlock-alt:before{content:"\f13e"}.fa-bullseye:before{content:"\f140"}.fa-ellipsis-h:before{content:"\f141"}.fa-ellipsis-v:before{content:"\f142"}.fa-rss-square:before{content:"\f143"}.fa-play-circle:before{content:"\f144"}.fa-ticket:before{content:"\f145"}.fa-minus-square:before{content:"\f146"}.fa-minus-square-o:before{content:"\f147"}.fa-level-up:before{content:"\f148"}.fa-level-down:before{content:"\f149"}.fa-check-square:before{content:"\f14a"}.fa-pencil-square:before{content:"\f14b"}.fa-external-link-square:before{content:"\f14c"}.fa-share-square:before{content:"\f14d"}.fa-compass:before{content:"\f14e"}.fa-toggle-down:before,.fa-caret-square-o-down:before{content:"\f150"}.fa-toggle-up:before,.fa-caret-square-o-up:before{content:"\f151"}.fa-toggle-right:before,.fa-caret-square-o-right:before{content:"\f152"}.fa-euro:before,.fa-eur:before{content:"\f153"}.fa-gbp:before{content:"\f154"}.fa-dollar:before,.fa-

usd:before{content:"\f155"}.fa-rupee:before,.fa-inr:before{content:"\f156"}.fa-cny:before,.fa-rmb:before,.fa-yen:before,.fa-jpy:before{content:"\f157"}.fa-ruble:before,.fa-rouble:before,.fa-rub:before{content:"\f158"}.fa-won:before,.fa-krw:before{content:"\f159"}.fa-bitcoin:before,.fa-btc:before{content:"\f15a"}.fa-file:before{content:"\f15b"}.fa-file-text:before{content:"\f15c"}.fa-sort-alpha-asc:before{content:"\f15d"}.fa-sort-alpha-desc:before{content:"\f15e"}.fa-sort-amount-asc:before{content:"\f160"}.fa-sort-amount-desc:before{content:"\f161"}.fa-sort-numeric-asc:before{content:"\f162"}.fa-sort-numeric-desc:before{content:"\f163"}.fa-thumbs-up:before{content:"\f164"}.fa-thumbs-down:before{content:"\f165"}.fa-youtube-square:before{content:"\f166"}.fa-youtube:before{content:"\f167"}.fa-xing:before{content:"\f168"}.fa-xing-square:before{content:"\f169"}.fa-youtube-play:before{content:"\f16a"}.fa-dropbox:before{content:"\f16b"}.fa-stack-overflow:before{content:"\f16c"}.fa-instagram:before{content:"\f16d"}.fa-flickr:before{content:"\f16e"}.fa-adn:before{content:"\f170"}.fa-bitbucket:before{content:"\f171"}.fa-bitbucket-square:before{content:"\f172"}.fa-tumblr:before{content:"\f173"}.fa-tumblr-square:before{content:"\f174"}.fa-long-arrow-down:before{content:"\f175"}.fa-long-arrow-up:before{content:"\f176"}.fa-long-arrow-left:before{content:"\f177"}.fa-long-arrow-right:before{content:"\f178"}.fa-apple:before{content:"\f179"}.fa-windows:before{content:"\f17a"}.fa-android:before{content:"\f17b"}.fa-linux:before{content:"\f17c"}.fa-dribbble:before{content:"\f17d"}.fa-skype:before{content:"\f17e"}.fa-foursquare:before{content:"\f180"}.fa-trello:before{content:"\f181"}.fa-female:before{content:"\f182"}.fa-male:before{content:"\f183"}.fa-gittip:before,.fa-gratipay:before{content:"\f184"}.fa-sun-o:before{content:"\f185"}.fa-moon-o:before{content:"\f186"}.fa-archive:before{content:"\f187"}.fa-bug:before{content:"\f188"}.fa-vk:before{content:"\f189"}.fa-weibo:before{content:"\f18a"}.fa-renren:before{content:"\f18b"}.fa-pagelines:before{content:"\f18c"}.fa-stack-exchange:before{content:"\f18d"}.fa-arrow-circle-o-right:before{content:"\f18e"}.fa-arrow-circle-o-left:before{content:"\f190"}.fa-toggle-left:before,.fa-caret-square-o-left:before{content:"\f191"}.fa-dot-circle-o:before{content:"\f192"}.fa-wheelchair:before{content:"\f193"}.fa-vimeo-square:before{content:"\f194"}.fa-turkish-lira:before,.fa-try:before{content:"\f195"}.fa-plus-square-o:before{content:"\f196"}.fa-space-shuttle:before{content:"\f197"}.fa-slack:before{content:"\f198"}.fa-envelope-square:before{content:"\f199"}.fa-wordpress:before{content:"\f19a"}.fa-openid:before{content:"\f19b"}.fa-institution:before,.fa-bank:before,.fa-university:before{content:"\f19c"}.fa-mortar-board:before,.fa-graduation-cap:before{content:"\f19d"}.fa-yahoo:before{content:"\f19e"}.fa-google:before{content:"\f1a0"}.fa-reddit:before{content:"\f1a1"}.fa-reddit-square:before{content:"\f1a2"}.fa-stumbleupon-circle:before{content:"\f1a3"}.fa-stumbleupon:before{content:"\f1a4"}.fa-delicious:before{content:"\f1a5"}.fa-digg:before{content:"\f1a6"}.fa-pied-piper:before{content:"\f1a7"}.fa-pied-piper-alt:before{content:"\f1a8"}.fa-drupal:before{content:"\f1a9"}.fa-joomla:before{content:"\f1aa"}.fa-language:before{content:"\f1ab"}.fa-fax:before{content:"\f1ac"}.fa-building:before{content:"\f1ad"}.fa-child:before{content:"\f1ae"}.fa-paw:before{content:"\f1b0"}.fa-spoon:before{content:"\f1b1"}.fa-cube:before{content:"\f1b2"}.fa-cubes:before{content:"\f1b3"}.fa-behance:before{content:"\f1b4"}.fa-behance-square:before{content:"\f1b5"}.fa-steam:before{content:"\f1b6"}.fa-steam-square:before{content:"\f1b7"}.fa-recycle:before{content:"\f1b8"}.fa-automobile:before,.fa-

car:before{content:"\f1b9"}.fa-cab:before,.fa-taxi:before{content:"\f1ba"}.fa-tree:before{content:"\f1bb"}.fa-spotify:before{content:"\f1bc"}.fa-deviantart:before{content:"\f1bd"}.fa-soundcloud:before{content:"\f1be"}.fa-database:before{content:"\f1c0"}.fa-file-pdf-o:before{content:"\f1c1"}.fa-file-word-o:before{content:"\f1c2"}.fa-file-excel-o:before{content:"\f1c3"}.fa-file-powerpoint-o:before{content:"\f1c4"}.fa-file-photo-o:before,.fa-file-picture-o:before,.fa-file-image-o:before{content:"\f1c5"}.fa-file-zip-o:before,.fa-file-archive-o:before{content:"\f1c6"}.fa-file-sound-o:before,.fa-file-audio-o:before{content:"\f1c7"}.fa-file-movie-o:before,.fa-file-video-o:before{content:"\f1c8"}.fa-file-code-o:before{content:"\f1c9"}.fa-vine:before{content:"\f1ca"}.fa-codepen:before{content:"\f1cb"}.fa-jsfiddle:before{content:"\f1cc"}.fa-life-bouy:before,.fa-life-buoy:before,.fa-life-saver:before,.fa-support:before,.fa-life-ring:before{content:"\f1cd"}.fa-circle-o-notch:before{content:"\f1ce"}.fa-ra:before,.fa-rebel:before{content:"\f1d0"}.fa-ge:before,.fa-empire:before{content:"\f1d1"}.fa-git-square:before{content:"\f1d2"}.fa-git:before{content:"\f1d3"}.fa-hacker-news:before{content:"\f1d4"}.fa-tencent-weibo:before{content:"\f1d5"}.fa-qq:before{content:"\f1d6"}.fa-wechat:before,.fa-weixin:before{content:"\f1d7"}.fa-send:before,.fa-paper-plane:before{content:"\f1d8"}.fa-send-o:before,.fa-paper-plane-o:before{content:"\f1d9"}.fa-history:before{content:"\f1da"}.fa-genderless:before,.fa-circle-thin:before{content:"\f1db"}.fa-header:before{content:"\f1dc"}.fa-paragraph:before{content:"\f1dd"}.fa-sliders:before{content:"\f1de"}.fa-share-alt:before{content:"\f1e0"}.fa-share-alt-square:before{content:"\f1e1"}.fa-bomb:before{content:"\f1e2"}.fa-soccer-ball-o:before,.fa-futbol-o:before{content:"\f1e3"}.fa-tty:before{content:"\f1e4"}.fa-binoculars:before{content:"\f1e5"}.fa-plug:before{content:"\f1e6"}.fa-slideshare:before{content:"\f1e7"}.fa-twitch:before{content:"\f1e8"}.fa-yelp:before{content:"\f1e9"}.fa-newspaper-o:before{content:"\f1ea"}.fa-wifi:before{content:"\f1eb"}.fa-calculator:before{content:"\f1ec"}.fa-paypal:before{content:"\f1ed"}.fa-google-wallet:before{content:"\f1ee"}.fa-cc-visa:before{content:"\f1f0"}.fa-cc-mastercard:before{content:"\f1f1"}.fa-cc-discover:before{content:"\f1f2"}.fa-cc-amex:before{content:"\f1f3"}.fa-cc-paypal:before{content:"\f1f4"}.fa-cc-stripe:before{content:"\f1f5"}.fa-bell-slash:before{content:"\f1f6"}.fa-bell-slash-o:before{content:"\f1f7"}.fa-trash:before{content:"\f1f8"}.fa-copyright:before{content:"\f1f9"}.fa-at:before{content:"\f1fa"}.fa-eyedropper:before{content:"\f1fb"}.fa-paint-brush:before{content:"\f1fc"}.fa-birthday-cake:before{content:"\f1fd"}.fa-area-chart:before{content:"\f1fe"}.fa-pie-chart:before{content:"\f200"}.fa-line-chart:before{content:"\f201"}.fa-lastfm:before{content:"\f202"}.fa-lastfm-square:before{content:"\f203"}.fa-toggle-off:before{content:"\f204"}.fa-toggle-on:before{content:"\f205"}.fa-bicycle:before{content:"\f206"}.fa-bus:before{content:"\f207"}.fa-ioxhost:before{content:"\f208"}.fa-angellist:before{content:"\f209"}.fa-cc:before{content:"\f20a"}.fa-shekel:before,.fa-sheqel:before,.fa-ils:before{content:"\f20b"}.fa-meanpath:before{content:"\f20c"}.fa-buysellads:before{content:"\f20d"}.fa-connectdevelop:before{content:"\f20e"}.fa-dashcube:before{content:"\f210"}.fa-forumbee:before{content:"\f211"}.fa-leanpub:before{content:"\f212"}.fa-sellsy:before{content:"\f213"}.fa-shirtsinbulk:before{content:"\f214"}.fa-simplybuilt:before{content:"\f215"}.fa-skyatlas:before{content:"\f216"}.fa-cart-plus:before{content:"\f217"}.fa-cart-arrow-down:before{content:"\f218"}.fa-diamond:before{content:"\f219"}.fa-ship:before{content:"\f21a"}.fa-user-

secret:before{content:"\f21b"}.fa-motorcycle:before{content:"\f21c"}.fa-street-view:before{content:"\f21d"}.fa-heartbeat:before{content:"\f21e"}.fa-venus:before{content:"\f221"}.fa-mars:before{content:"\f222"}.fa-mercury:before{content:"\f223"}.fa-transgender:before{content:"\f224"}.fa-transgender-alt:before{content:"\f225"}.fa-venus-double:before{content:"\f226"}.fa-mars-double:before{content:"\f227"}.fa-venus-mars:before{content:"\f228"}.fa-mars-stroke:before{content:"\f229"}.fa-mars-stroke-v:before{content:"\f22a"}.fa-mars-stroke-h:before{content:"\f22b"}.fa-neuter:before{content:"\f22c"}.fa-facebook-official:before{content:"\f230"}.fa-pinterest-p:before{content:"\f231"}.fa-whatsapp:before{content:"\f232"}.fa-server:before{content:"\f233"}.fa-user-plus:before{content:"\f234"}.fa-user-times:before{content:"\f235"}.fa-hotel:before,.fa-bed:before{content:"\f236"}.fa-viacoin:before{content:"\f237"}.fa-train:before{content:"\f238"}.fa-subway:before{content:"\f239"}.fa-medium:before{content:"\f23a"}/\*!

\*

\* IPython base

\*

\*/.modal.fade .modal-dialog{-webkit-transform:translate(0, 0);-ms-transform:translate(0, 0);-o-transform:translate(0, 0);transform:translate(0, 0)}code{color:#000}pre{font-size:inherit;line-height:inherit}label{font-weight:normal}.border-box-sizing{box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box}.corner-all{border-radius:2px}.no-padding{padding:0}.hbox{display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}.hbox>\*{-webkit-box-flex:0;-moz-box-flex:0;box-flex:0;flex:none}.vbox{display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}.vbox>\*{-webkit-box-flex:0;-moz-box-flex:0;box-flex:0;flex:none}.hbox.reverse,.vbox.reverse,.reverse{-webkit-box-direction:reverse;-moz-box-direction:reverse;box-direction:reverse;flex-direction:row-reverse}.hbox.box-flex0,.vbox.box-flex0,.box-flex0{-webkit-box-flex:0;-moz-box-flex:0;box-flex:0;flex:none;width:auto}.hbox.box-flex1,.vbox.box-flex1,.box-flex1{-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}.hbox.box-flex,.vbox.box-flex,.box-flex{-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}.hbox.box-flex2,.vbox.box-flex2,.box-flex2{-webkit-box-flex:2;-moz-box-flex:2;box-flex:2;flex:2}.box-group1{-webkit-box-flex-group:1;-moz-box-flex-group:1;box-flex-group:1}.box-group2{-webkit-box-flex-group:2;-moz-box-flex-group:2;box-flex-group:2}.hbox.start,.vbox.start,.start{-webkit-box-pack:start;-moz-box-pack:start;box-pack:start;justify-content:flex-start}.hbox.end,.vbox.end,.end{-webkit-box-pack:end;-moz-box-pack:end;box-pack:end;justify-content:flex-end}.hbox.center,.vbox.center,.center{-webkit-box-pack:center;-moz-box-pack:center;box-pack:center;justify-content:center}.hbox.baseline,.vbox.baseline,.baseline{-webkit-box-pack:baseline;-moz-box-pack:baseline;box-pack:baseline;justify-content:baseline}.hbox.stretch,.vbox.stretch,.stretch{-webkit-box-pack:stretch;-moz-box-pack:stretch;box-pack:stretch;justify-content:stretch}.hbox.align-start,.vbox.align-start,.align-start{-webkit-box-align:start;-moz-box-align:start;box-align:start;align-items:flex-start}.hbox.align-end,.vbox.align-end,.align-end{-webkit-box-align:end;-moz-box-align:end;box-align:end;align-items:flex-

```

end}.hbox.align-center,.align-center,.align-center{-webkit-box-align:center;-moz-box-align:center;box-align:center;align-items:center}.hbox.align-baseline,.vbox.align-baseline,.align-baseline{-webkit-box-align:baseline;-moz-box-align:baseline;box-align:baseline;align-items:baseline}.hbox.align-stretch,.vbox.align-stretch,.align-stretch{-webkit-box-align:stretch;-moz-box-align:stretch;box-align:stretch;align-items:stretch}div.error{margin:2em;text-align:center}div.error>h1{font-size:500%;line-height:normal}div.error>p{font-size:200%;line-height:normal}div.traceback-wrapper{text-align:left;max-width:800px;margin:auto}body{background-color:#fff;position:absolute;left:0;right:0;top:0;bottom:0;overflow:visible}#header{display:none;background-color:#fff;position:relative;z-index:100}#header #header-container{padding-bottom:5px;padding-top:5px;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box}#header .header-bar{width:100%;height:1px;background:#e7e7e7;margin-bottom:-1px}@media print{#header{display:none !important}}#header-spacer{width:100%;visibility:hidden}@media print{#header-spacer{display:none}}#ipython_notebook{padding-left:0;padding-top:1px;padding-bottom:1px}@media (max-width:991px){#ipython_notebook{margin-left:10px}}#noscript{width:auto;padding-top:16px;padding-bottom:16px;text-align:center;font-size:22px;color:red;font-weight:bold}#ipython_notebook img{height:28px}#site{width:100%;display:none;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;overflow:auto}@media print{#site{height:auto !important}}.ui-button .ui-button-text{padding:.2em .8em;font-size:77%}input.ui-button{padding:.3em .9em}span#login_widget{float:right}span#login_widget>.button,#logout{color:#333;background-color:#fff;border-color:#ccc}span#login_widget>.button:hover,#logout:hover,span#login_widget>.button:focus,#logout:focus,span#login_widget>.button.focus,#logout.focus,span#login_widget>.button.active,#logout.active,span#login_widget>.button.active,#logout.active,.open>.dropdown-togglespan#login_widget>.button,.open>.dropdown-toggle#logout{color:#333;background-color:#e6e6e6;border-color:#adadad}span#login_widget>.button.active,#logout.active,span#login_widget>.button.active,#logout.active,.open>.dropdown-togglespan#login_widget>.button,.open>.dropdown-toggle#logout{background-image:none}span#login_widget>.button.disabled,#logout.disabled,span#login_widget>.button[disabled],#logout[disabled],fieldset[disabled] span#login_widget>.button,fieldset[disabled] #logout,span#login_widget>.button.disabled:hover,#logout.disabled:hover,span#login_widget>.button[disabled]:hover,#logout[disabled]:hover,fieldset[disabled] span#login_widget>.button:hover,fieldset[disabled] #logout:hover,span#login_widget>.button.disabled:focus,#logout.disabled:focus,span#login_widget>.button[disabled]:focus,#logout[disabled]:focus,fieldset[disabled] span#login_widget>.button:focus,fieldset[disabled] #logout:focus,span#login_widget>.button.disabled.focus,#logout.disabled.focus,span#login_widget>.button[disabled].focus,#logout[disabled].focus,fieldset[disabled] span#login_widget>.button.focus,fieldset[disabled] #logout.focus,span#login_widget>.button.disabled:active,#logout.disabled:active,span#login_widget>.button[disabled]:active,#logout[disabled]:active,fieldset[disabled] span#login_widget>.button:active,fieldset[disabled] #logout:active,span#login_widget>.button.disabled:active,#logout.disabled:active,span#login_widget>.button[disabled].active,#logout[disabled].active,span#login_widget>.button.active,fieldset[disabled]

```

```

button[disabled].active,#logout[disabled].active,fieldset[disabled]
span#login_widget>.button.active,fieldset[disabled] #logout.active{background-color:#fff;border-
color:#ccc}span#login_widget>.button .badge,#logout .badge{color:#fff;background-color:#333}.nav-
header{text-transform:none}#header>span{margin-top:10px}.modal_stretch .modal-dialog{display:-
webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-
orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-
direction:column;align-items:stretch;min-height:80%}.modal_stretch .modal-dialog .modal-body{max-
height:none;flex:1}@media (min-width:768px){.modal .modal-dialog{width:700px}}@media (min-
width:768px){select.form-control{margin-left:12px;margin-right:12px}}/*!

```

\*

\* IPython auth

\*

\*/.center-nav{display:inline-block;margin-bottom:-4px}/\*!

\*

\* IPython tree view

\*

```

*/.alternate_upload{background-
color:none;display:inline}.alternate_upload.form{padding:0;margin:0}.alternate_upload
input.fileinput{display:inline;opacity:0;z-index:2;width:12ex;margin-right:-12ex}.alternate_upload
.input-overlay{display:inline-block;font-weight:bold;line-height:1em}ul#tabs{margin-bottom:4px}ul#tabs
a{padding-top:6px;padding-bottom:4px}ul.breadcrumb a:focus,ul.breadcrumb a:hover{text-
decoration:none}ul.breadcrumb i.icon-home{font-size:16px;margin-right:4px}ul.breadcrumb
span{color:#5e5e5e}.list_toolbar{padding:4px 0 4px 0;vertical-align:middle}.list_toolbar .tree-
buttons{padding-top:1px}.dynamic-buttons{display:inline-block}.list_toolbar [class*="span"]{min-
height:24px}.list_header{font-weight:bold;background-color:#eee}.list_placeholder{font-
weight:bold;padding-top:4px;padding-bottom:4px;padding-left:7px;padding-
right:7px}.list_container{margin-top:4px;margin-bottom:20px;border:1px solid #ddd;border-
radius:2px}.list_container>div{border-bottom:1px solid #ddd}.list_container>div:hover .list-
item{background-color:red}.list_container>div:last-child{border:none}.list_item:hover
.list_item{background-color:#ddd}.list_item a{text-decoration:none}.list_item:hover{background-
color:#fafafa}.action_col{text-align:right}.list_header>div,.list_item>div{padding-top:4px;padding-
bottom:4px;padding-left:7px;padding-right:7px;line-height:22px}.list_header>div input,.list_item>div
input{margin-right:7px;margin-left:14px;vertical-align:baseline;line-height:22px;position:relative;top:-
1px}.list_header>div .item_link,.list_item>div .item_link{margin-left:-1px;vertical-align:baseline;line-
height:22px}.new-file input[type=checkbox]{visibility:hidden}.item_name{line-
height:22px;height:24px}.item_icon{font-size:14px;color:#5e5e5e;margin-right:7px;margin-left:7px;line-
height:22px;vertical-align:baseline}.item_buttons{padding-top:4px;line-height:1em;margin-left:-
5px}.item_buttons .btn-group,.item_buttons .input-
group{float:left}.item_buttons>.btn,.item_buttons>.btn-group,.item_buttons>.input-group{margin-
left:5px}.item_buttons .btn{min-width:13ex}.item_buttons .running-

```

```

indicator{color:#5cb85c}.toolbar_info{height:24px;line-
height:24px}input.nbname_input,input.engine_num_input{padding-top:3px;padding-
bottom:3px;height:22px;line-
height:14px;margin:0}input.engine_num_input{width:60px}.highlight_text{color:blue}#project_name{di
splay:inline-block;padding-left:7px;margin-left:-2px}#project_name>.breadcrumb{padding:0;margin-
bottom:0;background-color:transparent;font-weight:bold}#tree-selector{display:inline-block;padding-
right:0}#tree-selector input[type=checkbox]{margin-left:7px;vertical-align:baseline}.tab-content
.row{margin-left:0;margin-right:0}.folder_icon:before{display:inline-block;font:normal normal normal
14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-
osx-font-smoothing:grayscale;transform:translate(0, 0);content:"\f114"}.folder_icon:before.pull-
left{margin-right:.3em}.folder_icon:before.pull-right{margin-
left:.3em}.notebook_icon:before{display:inline-block;font:normal normal normal 14px/1
FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f02d";position:relative;top:-
1px}.notebook_icon:before.pull-left{margin-right:.3em}.notebook_icon:before.pull-right{margin-
left:.3em}.running_notebook_icon:before{display:inline-block;font:normal normal normal 14px/1
FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f02d";position:relative;top:-
1px;color:#5cb85c}.running_notebook_icon:before.pull-left{margin-
right:.3em}.running_notebook_icon:before.pull-right{margin-left:.3em}.file_icon:before{display:inline-
block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-
font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0,
0);content:"\f016";position:relative;top:-2px}.file_icon:before.pull-left{margin-
right:.3em}.file_icon:before.pull-right{margin-left:.3em}#notebook_toolbar .pull-right{padding-
top:0;margin-right:-1px}ul#new-menu{left:auto;right:0}.kernel-menu-icon{padding-
right:12px;width:24px;content:"\f096"}.kernel-menu-icon:before{content:"\f096"}.kernel-menu-icon-
current:before{content:"\f00c"}#tab_content{padding-top:20px}#running .panel-group .panel{margin-
top:3px;margin-bottom:1em}#running .panel-group .panel .panel-heading{background-
color:#eee;padding-top:4px;padding-bottom:4px;padding-left:7px;padding-right:7px;line-
height:22px}#running .panel-group .panel .panel-heading a:focus,#running .panel-group .panel .panel-
heading a:hover{text-decoration:none}#running .panel-group .panel .panel-body{padding:0}#running
.panel-group .panel .panel-body .list_container{margin-top:0;margin-bottom:0;border:0;border-
radius:0}#running .panel-group .panel .panel-body .list_container .list_item{border-bottom:1px solid
#ddd}#running .panel-group .panel .panel-body .list_container .list_item:last-child{border-
bottom:0}.delete-button{display:none}.duplicate-button{display:none}.rename-
button{display:none}.shutdown-button{display:none}/*!

```

\*

\* IPython text editor webapp

\*

```

*/.selected-keymap i.fa{padding:0 5px}.selected-keymap i.fa:before{content:"\f00c"}#mode-
menu{overflow:auto;max-height:20em}.edit_app #header{-webkit-box-shadow:0 0 12px 1px
rgba(87,87,87,0.2);box-shadow:0 0 12px 1px rgba(87,87,87,0.2)}.edit_app #menubar .navbar{margin-

```

```

bottom:-1px}.dirty-indicator{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0);width:20px}.dirty-indicator.pull-left{margin-right:.3em}.dirty-indicator.pull-right{margin-left:.3em}.dirty-indicator-dirty{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0);width:20px}.dirty-indicator-dirty.pull-left{margin-right:.3em}.dirty-indicator-dirty.pull-right{margin-left:.3em}.dirty-indicator-clean{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0);width:20px}.dirty-indicator-clean.pull-left{margin-right:.3em}.dirty-indicator-clean.pull-right{margin-left:.3em}.dirty-indicator-clean:before{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0);content:"\f00c"}.dirty-indicator-clean:before.pull-left{margin-right:.3em}.dirty-indicator-clean:before.pull-right{margin-left:.3em}#filename{font-size:16pt;display:table;padding:0 5px}#current-mode{padding-left:5px;padding-right:5px}#texteditor-backdrop{padding-top:20px;padding-bottom:20px}@media not print{#texteditor-backdrop{background-color:#eee}}@media print{#texteditor-backdrop #texteditor-container .CodeMirror-gutter,#texteditor-backdrop #texteditor-container .CodeMirror-gutters{background-color:#fff}}@media not print{#texteditor-backdrop #texteditor-container .CodeMirror-gutter,#texteditor-backdrop #texteditor-container .CodeMirror-gutters{background-color:#fff}}@media not print{#texteditor-backdrop #texteditor-container{padding:0;background-color:#fff;-webkit-box-shadow:0 0 12px 1px rgba(87,87,87,0.2);box-shadow:0 0 12px 1px rgba(87,87,87,0.2)}}/*!

```

\*

\* IPython notebook

\*

```

*/.ansibold{font-weight:bold}.ansiblack{color:black}.ansired{color:darkred}.ansigreen{color:darkgreen}.ansiyellow{color:#c4a000}.ansiblue{color:darkblue}.ansipurple{color:darkviolet}.ansicyan{color:steelblue}.ansigray{color:gray}.ansibgblack{background-color:black}.ansibgred{background-color:red}.ansibggreen{background-color:green}.ansibgyellow{background-color:yellow}.ansibgblue{background-color:blue}.ansibgpurple{background-color:magenta}.ansibgcyan{background-color:cyan}.ansibggray{background-color:gray}div.cell{border:1px solid transparent;display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch;border-radius:2px;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;border-width:thin;border-style:solid;width:100%;padding:5px;margin:0;outline:none}div.cell.selected{border-color:#ababab}@media print{div.cell.selected{border-color:transparent}}.edit_mode div.cell.selected{border-color:green}@media print{.edit_mode div.cell.selected{border-color:transparent}}.prompt{min-width:14ex;padding:.4em;margin:0;font-family:monospace;text-align:right;line-height:1.21429em}@media (max-width:540px){.prompt{text-

```



align:left}}div.inner\_cell{display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch;-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}@-moz-document url-prefix(){div.inner\_cell{overflow-x:hidden}}div.input\_area{border:1px solid #cfcfcf;border-radius:2px;background:#f7f7f7;line-height:1.21429em}div.prompt:empty{padding-top:0;padding-bottom:0}div.unrecognized\_cell{padding:5px 5px 5px 0;display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}div.unrecognized\_cell .inner\_cell{border-radius:2px;padding:5px;font-weight:bold;color:red;border:1px solid #cfcfcf;background:#eaeaea}div.unrecognized\_cell .inner\_cell a{color:inherit;text-decoration:none}div.unrecognized\_cell .inner\_cell a: hover{color:inherit;text-decoration:none}@media (max-width:540px){div.unrecognized\_cell>div.prompt{display:none}}@media print{div.code\_cell{page-break-inside:avoid}}div.input{page-break-inside:avoid;display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}@media (max-width:540px){div.input{display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}}div.input\_prompt{color:navy;border-top:1px solid transparent}div.input\_area>div.highlight{margin:.4em;border:none;padding:0;background-color:transparent}div.input\_area>div.highlight>pre{margin:0;border:none;padding:0;background-color:transparent}.CodeMirror{line-height:1.21429em;font-size:14px;height:auto;background:none}.CodeMirror-scroll{overflow-y:hidden;overflow-x:auto}.CodeMirror-lines{padding:.4em}.CodeMirror-linenumber{padding:0 8px 0 4px}.CodeMirror-gutters{border-bottom-left-radius:2px;border-top-left-radius:2px}.CodeMirror-pre{padding:0;border:0;border-radius:0}.highlight-base{color:#000}.highlight-variable{color:#000}.highlight-variable-2{color:#1a1a1a}.highlight-variable-3{color:#333}.highlight-string{color:#ba2121}.highlight-comment{color:#408080;font-style:italic}.highlight-number{color:#080}.highlight-atom{color:#88f}.highlight-keyword{color:#008000;font-weight:bold}.highlight-builtin{color:#008000}.highlight-error{color:#f00}.highlight-operator{color:#a2f;font-weight:bold}.highlight-meta{color:#a2f}.highlight-def{color:#00f}.highlight-string-2{color:#f50}.highlight-qualifier{color:#555}.highlight-bracket{color:#997}.highlight-tag{color:#170}.highlight-attribute{color:#00c}.highlight-header{color:blue}.highlight-quote{color:#090}.highlight-link{color:#00c}.cm-s-ipython span.cm-keyword{color:#008000;font-weight:bold}.cm-s-ipython span.cm-atom{color:#88f}.cm-s-ipython span.cm-number{color:#080}.cm-s-ipython span.cm-def{color:#00f}.cm-s-ipython span.cm-variable{color:#000}.cm-s-ipython span.cm-operator{color:#a2f;font-weight:bold}.cm-s-ipython span.cm-variable-2{color:#1a1a1a}.cm-s-ipython span.cm-variable-3{color:#333}.cm-s-ipython span.cm-comment{color:#408080;font-style:italic}.cm-s-ipython span.cm-string{color:#ba2121}.cm-s-ipython span.cm-string-2{color:#f50}.cm-s-ipython span.cm-meta{color:#a2f}.cm-s-ipython span.cm-qualifier{color:#555}.cm-s-ipython span.cm-builtin{color:#008000}.cm-s-ipython span.cm-bracket{color:#997}.cm-s-ipython span.cm-tag{color:#170}.cm-s-ipython span.cm-attribute{color:#00c}.cm-s-ipython span.cm-header{color:blue}.cm-s-ipython span.cm-quote{color:#090}.cm-s-ipython span.cm-link{color:#00c}.cm-

s-ipython span.cm-error{color:#f00}.cm-s-ipython span.cm-tab{background:url(  
AAAAAXNSR0IArs4c6QAAAGFJREFUSMft1LsRQAQheHPowAKoACx3IgEKtaEHujDjORSgWTH/ZOdnZOcM  
/sgk/kFFWY0qV8foQwS4MKBCS3qR6ixBJvEI0obYAtivseIE120FaowJPN75GMu8j/LfMwNjh4HUpwg4LUAA  
AAASUVORK5CYII=);background-position:right;background-repeat:no-repeat}div.output\_wrapper{position:relative;display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}div.output\_scroll{height:24em;width:100%;overflow:auto;border-radius:2px;-webkit-box-shadow:inset 0 2px 8px rgba(0,0,0,0.8);box-shadow:inset 0 2px 8px rgba(0,0,0,0.8);display:block}div.output\_collapsed{margin:0;padding:0;display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}div.out\_prompt\_overlay{height:100%;padding:0 .4em;position:absolute;border-radius:2px}div.out\_prompt\_overlay:hover{-webkit-box-shadow:inset 0 0 1px #000;box-shadow:inset 0 0 1px #000;background:rgba(240,240,240,0.5)}div.output\_prompt{color:darkred}div.output\_area{padding:0;page-break-inside:avoid;display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}div.output\_area.MathJax\_Display{text-align:left !important}div.output\_area.rendered\_html table{margin-left:0;margin-right:0}div.output\_area.rendered\_html img{margin-left:0;margin-right:0}div.output{-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}@media (max-width:540px){div.output\_area{display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}}div.output\_area pre{margin:0;padding:0;border:0;vertical-align:baseline;color:black;background-color:transparent;border-radius:0}div.output\_subarea{padding:.4em;-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}div.output\_text{text-align:left;color:#000;line-height:1.21429em}div.output\_stderr{background:#fdd}div.output\_latex{text-align:left}div.output\_javascript:empty{padding:0}.js-error{color:darkred}div.raw\_input\_container{font-family:monospace;padding-top:5px}input.raw\_input{font-family:inherit;font-size:inherit;color:inherit;width:auto;vertical-align:baseline;padding:0 .25em;margin:0 .25em}input.raw\_input:focus{box-shadow:none}p.p-space{margin-bottom:10px}div.output\_unrecognized{padding:5px;font-weight:bold;color:red}div.output\_unrecognized a{color:inherit;text-decoration:none}div.output\_unrecognized a:hover{color:inherit;text-decoration:none}.rendered\_html{color:#000}.rendered\_html em{font-style:italic}.rendered\_html strong{font-weight:bold}.rendered\_html u{text-decoration:underline}.rendered\_html :link{text-decoration:underline}.rendered\_html :visited{text-decoration:underline}.rendered\_html h1{font-size:185.7%;margin:1.08em 0 0 0;font-weight:bold;line-height:1}.rendered\_html h2{font-size:157.1%;margin:1.27em 0 0 0;font-weight:bold;line-height:1}.rendered\_html h3{font-

size:128.6%;margin:1.55em 0 0 0;font-weight:bold;line-height:1}.rendered\_html h4{font-size:100%;margin:2em 0 0 0;font-weight:bold;line-height:1}.rendered\_html h5{font-size:100%;margin:2em 0 0 0;font-weight:bold;line-height:1;font-style:italic}.rendered\_html h6{font-size:100%;margin:2em 0 0 0;font-weight:bold;line-height:1;font-style:italic}.rendered\_html h1:first-child{margin-top:.538em}.rendered\_html h2:first-child{margin-top:.636em}.rendered\_html h3:first-child{margin-top:.777em}.rendered\_html h4:first-child{margin-top:1em}.rendered\_html h5:first-child{margin-top:1em}.rendered\_html h6:first-child{margin-top:1em}.rendered\_html ul{list-style:disc;margin:0 2em;padding-left:0}.rendered\_html ul ul{list-style:square;margin:0 2em}.rendered\_html ul ul ul{list-style:circle;margin:0 2em}.rendered\_html ol{list-style:decimal;margin:0 2em;padding-left:0}.rendered\_html ol ol{list-style:upper-alpha;margin:0 2em}.rendered\_html ol ol ol{list-style:lower-alpha;margin:0 2em}.rendered\_html ol ol ol ol{list-style:lower-roman;margin:0 2em}.rendered\_html ol ol ol ol ol{list-style:decimal;margin:0 2em}.rendered\_html \*+ul{margin-top:1em}.rendered\_html \*+ol{margin-top:1em}.rendered\_html hr{color:black;background-color:black}.rendered\_html pre{margin:1em 2em}.rendered\_html pre,.rendered\_html code{border:0;background-color:#fff;color:#000;font-size:100%;padding:0}.rendered\_html blockquote{margin:1em 2em}.rendered\_html table{margin-left:auto;margin-right:auto;border:1px solid black;border-collapse:collapse}.rendered\_html tr,.rendered\_html th,.rendered\_html td{border:1px solid black;border-collapse:collapse;margin:1em 2em}.rendered\_html td,.rendered\_html th{text-align:left;vertical-align:middle;padding:4px}.rendered\_html th{font-weight:bold}.rendered\_html \*+table{margin-top:1em}.rendered\_html p{text-align:left}.rendered\_html \*+p{margin-top:1em}.rendered\_html img{display:block;margin-left:auto;margin-right:auto}.rendered\_html \*+img{margin-top:1em}div.text\_cell{display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}@media (max-width:540px){div.text\_cell>div.prompt{display:none}}div.text\_cell\_render{outline:none;resize:none;width:h:inherit;border-style:none;padding:.5em .5em .5em .4em;color:#000;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box}a.anchor-link:link{text-decoration:none;padding:0 20px;visibility:hidden}h1:hover .anchor-link,h2:hover .anchor-link,h3:hover .anchor-link,h4:hover .anchor-link,h5:hover .anchor-link,h6:hover .anchor-link{visibility:visible}.text\_cell.rendered .input\_area{display:none}.text\_cell.unrendered .text\_cell\_render{display:none}.cm-header-1,.cm-header-2,.cm-header-3,.cm-header-4,.cm-header-5,.cm-header-6{font-weight:bold;font-family:"Helvetica Neue",Helvetica,Arial,sans-serif}.cm-header-1{font-size:185.7%}.cm-header-2{font-size:157.1%}.cm-header-3{font-size:128.6%}.cm-header-4{font-size:110%}.cm-header-5{font-size:100%;font-style:italic}.cm-header-6{font-size:100%;font-style:italic}.widget-interact>div,.widget-interact>input{padding:2.5px}.widget-area{page-break-inside:avoid;display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}.widget-area .widget-subarea{padding:.44em .4em .4em 1px;margin-left:6px;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch;-webkit-box-flex:2;-moz-box-flex:2;box-flex:2;flex:2;-webkit-box-align:start;-moz-box-align:start;box-align:start;align-items:flex-start}.widget-area.connection-problems .prompt:after{content:"\f127";font-family:'FontAwesome';color:#d9534f;font-

```

size:14px;top:3px;padding:3px}.slide-track{border:1px solid #ccc;background:#fff;border-
radius:2px}.widget-hslider{padding-left:8px;padding-
right:2px;overflow:visible;width:350px;height:5px;max-height:5px;margin-top:13px;margin-
bottom:10px;border:1px solid #ccc;background:#fff;border-radius:2px;display:-webkit-box;-webkit-box-
orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-
align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-
items:stretch}.widget-hslider .ui-slider{border:0;background:none;display:-webkit-box;-webkit-box-
orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-
align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-
items:stretch;-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}.widget-hslider .ui-slider-
handle{width:12px;height:28px;margin-top:-8px;border-radius:2px}.widget-hslider .ui-slider .ui-slider-
range{height:12px;margin-top:-4px;background:#eee}.widget-vslider{padding-
bottom:5px;overflow:visible;width:5px;max-width:5px;height:250px;margin-left:12px;border:1px solid
#ccc;background:#fff;border-radius:2px;display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-
align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-
orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-items:stretch}.widget-vslider .ui-
slider{border:0;background:none;margin-left:-4px;margin-top:5px;display:-webkit-box;-webkit-box-
orient:vertical;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-
align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-
items:stretch;-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}.widget-vslider .ui-slider .ui-slider-
handle{width:28px;height:12px;margin-left:-9px;border-radius:2px}.widget-vslider .ui-slider .ui-slider-
range{width:12px;margin-left:-1px;background:#eee}.widget-text{width:350px;margin:0}.widget-
listbox{width:350px;margin-bottom:0}.widget-numeric-text{width:150px;margin:0}.widget-
progress{margin-top:6px;min-width:350px}.widget-progress .progress-bar{-webkit-transition:none;-
moz-transition:none;-ms-transition:none;-o-transition:none;transition:none}.widget-combo-btn{min-
width:125px}.widget_item .dropdown-menu li a{color:inherit}.widget-hbox{display:-webkit-box;-webkit-
box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-
align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-
items:stretch}.widget-hbox input[type="checkbox"]{margin-top:9px;margin-bottom:10px}.widget-hbox
.widget-label{min-width:10ex;padding-right:8px;padding-top:5px;text-align:right;vertical-align:text-
top}.widget-hbox .widget-readout{padding-left:8px;padding-top:5px;text-align:left;vertical-align:text-
top}.widget-vbox{display:-webkit-box;-webkit-box-orient:vertical;-webkit-box-align:stretch;display:-
moz-box;-moz-box-orient:vertical;-moz-box-align:stretch;display:box;box-orient:vertical;box-
align:stretch;display:flex;flex-direction:column;align-items:stretch}.widget-vbox .widget-label{padding-
bottom:5px;text-align:center;vertical-align:text-bottom}.widget-vbox .widget-readout{padding-
top:5px;text-align:center;vertical-align:text-top}.widget-box{box-sizing:border-box;-moz-box-
sizing:border-box;-webkit-box-sizing:border-box;-webkit-box-align:start;-moz-box-align:start;box-
align:start;align-items:flex-start}.widget-radio-box{display:-webkit-box;-webkit-box-orient:vertical;-
webkit-box-align:stretch;display:-moz-box;-moz-box-orient:vertical;-moz-box-
align:stretch;display:box;box-orient:vertical;box-align:stretch;display:flex;flex-direction:column;align-
items:stretch;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;padding-
top:4px}.widget-radio-box label{margin-top:0}.widget-radio{margin-left:20px}/*!

```

\*

\* IPython notebook webapp

\*

```
*/@media (max-width:767px){.notebook_app{padding-left:0;padding-right:0}}#ipython-main-app{box-
sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-
box;height:100%}div#notebook_panel{margin:0;padding:0;box-sizing:border-box;-moz-box-
sizing:border-box;-webkit-box-sizing:border-box;height:100%}#notebook{font-size:14px;line-
height:20px;overflow-y:hidden;overflow-x:auto;width:100%;padding-
top:20px;margin:0;outline:none;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-
sizing:border-box;min-height:100%}@media not print{#notebook-container{padding:15px;background-
color:#fff;min-height:0;-webkit-box-shadow:0 0 12px 1px rgba(87,87,87,0.2);box-shadow:0 0 12px 1px
rgba(87,87,87,0.2)}}div.ui-widget-content{border:1px solid
#ababab;outline:none}pre.dialog{background-color:#f7f7f7;border:1px solid #ddd;border-
radius:2px;padding:.4em;padding-left:2em}p.dialog{padding:.2em}pre,code,kbd,samp{white-space:pre-
wrap}#fonttest{font-family:monospace}p{margin-bottom:0}.end_space{min-
height:100px;transition:height .2s ease}.notebook_app #header{-webkit-box-shadow:0 0 12px 1px
rgba(87,87,87,0.2);box-shadow:0 0 12px 1px rgba(87,87,87,0.2)}@media not
print{.notebook_app{background-color:#eee}.celltoolbar{border:thin solid #cfcfcf;border-
bottom:none;background:#eee;border-radius:2px 2px 0 0;width:100%;height:29px;padding-
right:4px;display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-
moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-
align:stretch;display:flex;flex-direction:row;align-items:stretch;-webkit-box-pack:end;-moz-box-
pack:end;box-pack:end;justify-content:flex-end}@media
print{.celltoolbar{display:none}}.ctb_hideshow{display:none;vertical-align:bottom}.ctb_global_show
.ctb_show.ctb_hideshow{display:block}.ctb_global_show .ctb_show+.input_area,.ctb_global_show
.ctb_show+div.text_cell_input,.ctb_global_show .ctb_show~div.text_cell_render{border-top-right-
radius:0;border-top-left-radius:0}.ctb_global_show .ctb_show~div.text_cell_render{border:1px solid
#cfcfcf}.celltoolbar{font-size:87%;padding-top:3px}.celltoolbar
select{display:block;width:100%;height:32px;padding:6px 12px;font-size:13px;line-
height:1.42857143;color:#555;background-color:#fff;background-image:none;border:1px solid
#ccc;border-radius:2px;-webkit-box-shadow:inset 0 1px 1px rgba(0,0,0,0.075);box-shadow:inset 0 1px
1px rgba(0,0,0,0.075);-webkit-transition:border-color ease-in-out .15s, box-shadow ease-in-out .15s;-o-
transition:border-color ease-in-out .15s, box-shadow ease-in-out .15s;transition:border-color ease-in-
out .15s, box-shadow ease-in-out .15s;height:30px;padding:5px 10px;font-size:12px;line-
height:1.5;border-radius:1px;width:inherit;font-size:inherit;height:22px;padding:0;display:inline-
block}.celltoolbar select:focus{border-color:#66afe9;outline:0;-webkit-box-shadow:inset 0 1px 1px
rgba(0,0,0,.075), 0 0 8px rgba(102, 175, 233, 0.6);box-shadow:inset 0 1px 1px rgba(0,0,0,.075), 0 0 8px
rgba(102, 175, 233, 0.6)}.celltoolbar select::-moz-placeholder{color:#999;opacity:1}.celltoolbar select:-
ms-input-placeholder{color:#999}.celltoolbar select::-webkit-input-placeholder{color:#999}.celltoolbar
select[disabled],.celltoolbar select[readonly],fieldset[disabled] .celltoolbar select{cursor:not-
allowed;background-color:#eee;opacity:1}textarea.celltoolbar select{height:auto}select.celltoolbar
select{height:30px;line-height:30px}textarea.celltoolbar select,select[multiple].celltoolbar
select{height:auto}.celltoolbar label{margin-left:5px;margin-right:5px}.completions{position:absolute;z-
index:10;overflow:hidden;border:1px solid #ababab;border-radius:2px;-webkit-box-shadow:0 6px 10px -
```

```
1px #adadad;box-shadow:0 6px 10px -1px #adadad}.completions
select{background:white;outline:none;border:none;padding:0;margin:0;overflow:auto;font-
family:monospace;font-size:110%;color:#000;width:auto}.completions select
option.context{color:#286090}#kernel_logo_widget{float:right
!important;float:right}#kernel_logo_widget .current_kernel_logo{display:none;margin-top:-1px;margin-
bottom:-1px;width:32px;height:32px}#menubar{box-sizing:border-box;-moz-box-sizing:border-box;-
webkit-box-sizing:border-box;margin-top:1px}#menubar .navbar{border-top:1px;border-radius:0 0 2px
2px;margin-bottom:0}#menubar .navbar-toggle{float:left;padding-top:7px;padding-
bottom:7px;border:none}#menubar .navbar-collapse{clear:left}.nav-wrapper{border-bottom:1px solid
#e7e7e7}.i.menu-icon{padding-top:4px}ul#help_menu li a{overflow:hidden;padding-
right:2.2em}ul#help_menu li a i{margin-right:-1.2em}.dropdown-submenu{position:relative}.dropdown-
submenu>.dropdown-menu{top:0;left:100%;margin-top:-6px;margin-left:-1px}.dropdown-
submenu: hover>.dropdown-menu{display:block}.dropdown-submenu>a:after{display:inline-
block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-
font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0,
0);display:block;content:"\f0da";float:right;color:#333;margin-top:2px;margin-right:-10px}.dropdown-
submenu>a:after.pull-left{margin-right:.3em}.dropdown-submenu>a:after.pull-right{margin-
left:.3em}.dropdown-submenu: hover>a:after{color:#262626}.dropdown-submenu.pull-
left{float:none}.dropdown-submenu.pull-left>.dropdown-menu{left:-100%;margin-
left:10px}#notification_area{float:right !important;float:right;z-index:10}.indicator_area{float:right
!important;float:right;color:#777;margin-left:5px;margin-right:5px;width:11px;z-index:10;text-
align:center;width:auto}#kernel_indicator{float:right !important;float:right;color:#777;margin-
left:5px;margin-right:5px;width:11px;z-index:10;text-align:center;width:auto;border-left:1px
solid}#kernel_indicator .kernel_indicator_name{padding-left:5px;padding-
right:5px}#modal_indicator{float:right !important;float:right;color:#777;margin-left:5px;margin-
right:5px;width:11px;z-index:10;text-align:center;width:auto}#readonly-indicator{float:right
!important;float:right;color:#777;margin-left:5px;margin-right:5px;width:11px;z-index:10;text-
align:center;width:auto;margin-top:2px;margin-bottom:0;margin-left:0;margin-
right:0;display:none}.modal_indicator:before{width:1.28571429em;text-align:center}.edit_mode
.modal_indicator:before{display:inline-block;font:normal normal normal 14px/1 FontAwesome;font-
size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f040"}.edit_mode
.modal_indicator:before.pull-left{margin-right:.3em}.edit_mode .modal_indicator:before.pull-
right{margin-left:.3em}.command_mode .modal_indicator:before{display:inline-block;font:normal
normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-
smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0, 0);content:'
'}.command_mode .modal_indicator:before.pull-left{margin-right:.3em}.command_mode
.modal_indicator:before.pull-right{margin-left:.3em}.kernel_idle_icon:before{display:inline-
block;font:normal normal normal 14px/1 FontAwesome;font-size:inherit;text-rendering:auto;-webkit-
font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;transform:translate(0,
0);content:"\f10c"}.kernel_idle_icon:before.pull-left{margin-right:.3em}.kernel_idle_icon:before.pull-
right{margin-left:.3em}.kernel_busy_icon:before{display:inline-block;font:normal normal normal 14px/1
FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f111"}.kernel_busy_icon:before.pull-
```

```

left{margin-right:.3em}.kernel_busy_icon:before.pull-right{margin-
left:.3em}.kernel_dead_icon:before{display:inline-block;font:normal normal normal 14px/1
FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f1e2"}.kernel_dead_icon:before.pull-
left{margin-right:.3em}.kernel_dead_icon:before.pull-right{margin-
left:.3em}.kernel_disconnected_icon:before{display:inline-block;font:normal normal normal 14px/1
FontAwesome;font-size:inherit;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-
smoothing:grayscale;transform:translate(0, 0);content:"\f127"}.kernel_disconnected_icon:before.pull-
left{margin-right:.3em}.kernel_disconnected_icon:before.pull-right{margin-
left:.3em}.notification_widget{color:#777;z-
index:10;background:rgba(240,240,240,0.5);color:#333;background-color:#fff;border-
color:#ccc}.notification_widget:hover,.notification_widget:focus,.notification_widget.focus,.notification
_widget:active,.notification_widget.active,.open>.dropdown-
toggle.notification_widget{color:#333;background-color:#e6e6e6;border-
color:#adadad}.notification_widget:active,.notification_widget.active,.open>.dropdown-
toggle.notification_widget{background-
image:none}.notification_widget.disabled,.notification_widget[disabled],fieldset[disabled]
.notification_widget,.notification_widget.disabled:hover,.notification_widget[disabled]:hover,fieldset[di
sabled]
.notification_widget:hover,.notification_widget.disabled:focus,.notification_widget[disabled]:focus,field
set[disabled]
.notification_widget:focus,.notification_widget.disabled.focus,.notification_widget[disabled].focus,fields
et[disabled]
.notification_widget.focus,.notification_widget.disabled:active,.notification_widget[disabled]:active,fiel
dset[disabled]
.notification_widget:active,.notification_widget.disabled.active,.notification_widget[disabled].active,fiel
dset[disabled] .notification_widget.active{background-color:#fff;border-color:#ccc}.notification_widget
.badge{color:#fff;background-color:#333}.notification_widget.warning{color:#fff;background-
color:#f0ad4e;border-
color:#eea236}.notification_widget.warning:hover,.notification_widget.warning:focus,.notification_widg
et.warning.focus,.notification_widget.warning:active,.notification_widget.warning.active,.open>.dropdo
wn-toggle.notification_widget.warning{color:#fff;background-color:#ec971f;border-
color:#d58512}.notification_widget.warning:active,.notification_widget.warning.active,.open>.dropdow
n-toggle.notification_widget.warning{background-
image:none}.notification_widget.warning.disabled,.notification_widget.warning[disabled],fieldset[disabl
ed]
.notification_widget.warning,.notification_widget.warning.disabled:hover,.notification_widget.warning[
disabled]:hover,fieldset[disabled]
.notification_widget.warning:hover,.notification_widget.warning.disabled:focus,.notification_widget.wa
rning[disabled]:focus,fieldset[disabled]
.notification_widget.warning:focus,.notification_widget.warning.disabled.focus,.notification_widget.war
ning[disabled].focus,fieldset[disabled]
.notification_widget.warning.focus,.notification_widget.warning.disabled:active,.notification_widget.wa
rning[disabled]:active,fieldset[disabled]

```

.notification\_widget.warning:active,.notification\_widget.warning.disabled:active,.notification\_widget.w  
arning[disabled].active,fieldset[disabled] .notification\_widget.warning.active{background-  
color:#f0ad4e;border-color:#eea236}.notification\_widget.warning .badge{color:#f0ad4e;background-  
color:#fff}.notification\_widget.success{color:#fff;background-color:#5cb85c;border-  
color:#4cae4c}.notification\_widget.success:hover,.notification\_widget.success:focus,.notification\_widge  
t.success:focus,.notification\_widget.success:active,.notification\_widget.success.active,.open>.dropdown  
-toggle.notification\_widget.success{color:#fff;background-color:#449d44;border-  
color:#398439}.notification\_widget.success:active,.notification\_widget.success.active,.open>.dropdown  
-toggle.notification\_widget.success{background-  
image:none}.notification\_widget.success.disabled,.notification\_widget.success[disabled],fieldset[disabl  
ed]  
.notification\_widget.success,.notification\_widget.success.disabled:hover,.notification\_widget.success[di  
sabled]:hover,fieldset[disabled]  
.notification\_widget.success:hover,.notification\_widget.success.disabled:focus,.notification\_widget.succ  
ess[disabled]:focus,fieldset[disabled]  
.notification\_widget.success:focus,.notification\_widget.success.disabled:focus,.notification\_widget.succ  
ess[disabled].focus,fieldset[disabled]  
.notification\_widget.success:focus,.notification\_widget.success.disabled:active,.notification\_widget.succ  
ess[disabled]:active,fieldset[disabled]  
.notification\_widget.success:active,.notification\_widget.success.disabled:active,.notification\_widget.suc  
cess[disabled].active,fieldset[disabled] .notification\_widget.success.active{background-  
color:#5cb85c;border-color:#4cae4c}.notification\_widget.success .badge{color:#5cb85c;background-  
color:#fff}.notification\_widget.info{color:#fff;background-color:#5bc0de;border-  
color:#46b8da}.notification\_widget.info:hover,.notification\_widget.info:focus,.notification\_widget.info.f  
ocus,.notification\_widget.info:active,.notification\_widget.info.active,.open>.dropdown-  
toggle.notification\_widget.info{color:#fff;background-color:#31b0d5;border-  
color:#269abc}.notification\_widget.info:active,.notification\_widget.info.active,.open>.dropdown-  
toggle.notification\_widget.info{background-  
image:none}.notification\_widget.info.disabled,.notification\_widget.info[disabled],fieldset[disabled]  
.notification\_widget.info,.notification\_widget.info.disabled:hover,.notification\_widget.info[disabled]:ho  
ver,fieldset[disabled]  
.notification\_widget.info:hover,.notification\_widget.info.disabled:focus,.notification\_widget.info[disabl  
ed]:focus,fieldset[disabled]  
.notification\_widget.info:focus,.notification\_widget.info.disabled:focus,.notification\_widget.info[disabl  
ed].focus,fieldset[disabled]  
.notification\_widget.info:focus,.notification\_widget.info.disabled:active,.notification\_widget.info[disabl  
ed]:active,fieldset[disabled]  
.notification\_widget.info:active,.notification\_widget.info.disabled:active,.notification\_widget.info[disabl  
ed].active,fieldset[disabled] .notification\_widget.info.active{background-color:#5bc0de;border-  
color:#46b8da}.notification\_widget.info .badge{color:#5bc0de;background-  
color:#fff}.notification\_widget.danger{color:#fff;background-color:#d9534f;border-  
color:#d43f3a}.notification\_widget.danger:hover,.notification\_widget.danger:focus,.notification\_widget  
.danger:focus,.notification\_widget.danger:active,.notification\_widget.danger.active,.open>.dropdown-  
toggle.notification\_widget.danger{color:#fff;background-color:#c9302c;border-



color:#ac2925}.notification\_widget.danger:active,.notification\_widget.danger.active,.open>.dropdown-toggle.notification\_widget.danger{background-image:none}.notification\_widget.danger.disabled,.notification\_widget.danger[disabled],fieldset[disabled]

.notification\_widget.danger,.notification\_widget.danger.disabled:hover,.notification\_widget.danger[disabled]:hover,fieldset[disabled]

.notification\_widget.danger:hover,.notification\_widget.danger.disabled:focus,.notification\_widget.danger[disabled]:focus,fieldset[disabled]

.notification\_widget.danger:focus,.notification\_widget.danger.disabled:focus,.notification\_widget.danger[disabled].focus,fieldset[disabled]

.notification\_widget.danger.focus,.notification\_widget.danger.disabled:active,.notification\_widget.danger[disabled]:active,fieldset[disabled]

.notification\_widget.danger:active,.notification\_widget.danger.disabled.active,.notification\_widget.danger[disabled].active,fieldset[disabled] .notification\_widget.danger.active{background-color:#d9534f;border-color:#d43f3a}.notification\_widget.danger .badge{color:#d9534f;background-color:#fff}div#pager{background-color:#fff;font-size:14px;line-height:20px;overflow:hidden;display:none;position:fixed;bottom:0;width:100%;max-height:50%;padding-top:8px;-webkit-box-shadow:0 0 12px 1px rgba(87,87,87,0.2);box-shadow:0 0 12px 1px rgba(87,87,87,0.2);z-index:100;top:auto !important}div#pager pre{line-height:1.21429em;color:#000;background-color:#f7f7f7;padding:.4em}div#pager #pager-button-area{position:absolute;top:8px;right:20px}div#pager #pager-contents{position:relative;overflow:auto;width:100%;height:100%}div#pager #pager-contents #pager-container{position:relative;padding:15px 0;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box}div#pager .ui-resizable-handle{top:0;height:8px;background:#f7f7f7;border-top:1px solid #cfcfcf;border-bottom:1px solid #cfcfcf}div#pager .ui-resizable-handle::after{content:"";top:2px;left:50%;height:3px;width:30px;margin-left:-15px;position:absolute;border-top:1px solid #cfcfcf}.quickhelp{display:-webkit-box;-webkit-box-orient:horizontal;-webkit-box-align:stretch;display:-moz-box;-moz-box-orient:horizontal;-moz-box-align:stretch;display:box;box-orient:horizontal;box-align:stretch;display:flex;flex-direction:row;align-items:stretch}.shortcut\_key{display:inline-block;width:20ex;text-align:right;font-family:monospace}.shortcut\_descr{display:inline-block;-webkit-box-flex:1;-moz-box-flex:1;box-flex:1;flex:1}span.save\_widget{margin-top:6px}span.save\_widget span.filename{height:1em;line-height:1em;padding:3px;margin-left:16px;border:none;font-size:146.5%;border-radius:2px}span.save\_widget span.filename:hover{background-color:#e6e6e6}span.checkpoint\_status,span.autosave\_status{font-size:small}@media (max-width:767px){span.save\_widget{font-size:small}span.checkpoint\_status,span.autosave\_status{display:none}}@media (min-width:768px) and (max-width:991px){span.checkpoint\_status{display:none}span.autosave\_status{font-size:x-small}}.toolbar{padding:0;margin-left:-5px;margin-top:2px;margin-bottom:5px;box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box}.toolbar select,.toolbar label{width:auto;vertical-align:middle;margin-right:2px;margin-bottom:0;display:inline;font-size:92%;margin-left:.3em;margin-right:.3em;padding:0;padding-top:3px}.toolbar .btn{padding:2px 8px}.toolbar .btn-group{margin-top:0;margin-left:5px}#maintoolbar{margin-bottom:-3px;margin-top:-8px;border:0;min-height:27px;margin-left:0;padding-top:11px;padding-bottom:3px}#maintoolbar

```
.navbar-text{float:none;vertical-align:middle;text-align:right;margin-left:5px;margin-right:0;margin-top:0}.select-xs{height:24px}@-moz-keyframes fadeOut{from{opacity:1}to{opacity:0}}@-webkit-keyframes fadeOut{from{opacity:1}to{opacity:0}}@-moz-keyframes fadeIn{from{opacity:0}to{opacity:1}}@-webkit-keyframes fadeIn{from{opacity:0}to{opacity:1}}.bigtooltip{overflow:auto;height:200px;-webkit-transition-property:height;-webkit-transition-duration:500ms;-moz-transition-property:height;-moz-transition-duration:500ms;transition-property:height;transition-duration:500ms}.smalltooltip{-webkit-transition-property:height;-webkit-transition-duration:500ms;-moz-transition-property:height;-moz-transition-duration:500ms;transition-property:height;transition-duration:500ms;text-overflow:ellipsis;overflow:hidden;height:80px}.tooltipbuttons{position:absolute;padding-right:15px;top:0;right:0}.tooltiptext{padding-right:30px}.ipython_tooltip{max-width:700px;-webkit-animation:fadeOut 400ms;-moz-animation:fadeOut 400ms;animation:fadeOut 400ms;-webkit-animation:fadeIn 400ms;-moz-animation:fadeIn 400ms;animation:fadeIn 400ms;vertical-align:middle;background-color:#f7f7f7;overflow:visible;border:#ababab 1px solid;outline:none;padding:3px;margin:0;padding-left:7px;font-family:monospace;min-height:50px;-moz-box-shadow:0 6px 10px -1px #adadad;-webkit-box-shadow:0 6px 10px -1px #adadad;box-shadow:0 6px 10px -1px #adadad;border-radius:2px;position:absolute;z-index:1000}.ipython_tooltip a{float:right}.ipython_tooltip .tooltiptext pre{border:0;border-radius:0;font-size:100%;background-color:#f7f7f7}.pretooltiparrow{left:0;margin:0;top:-16px;width:40px;height:16px;overflow:hidden;position:absolute}.pretooltiparrow:before{background-color:#f7f7f7;border:1px #ababab solid;z-index:11;content:"";position:absolute;left:15px;top:10px;width:25px;height:25px;-webkit-transform:rotate(45deg);-moz-transform:rotate(45deg);-ms-transform:rotate(45deg);-o-transform:rotate(45deg)}.terminal-app{background:#eee}.terminal-app #header{background:#fff;-webkit-box-shadow:0 0 12px 1px rgba(87,87,87,0.2);box-shadow:0 0 12px 1px rgba(87,87,87,0.2)}.terminal-app .terminal{float:left;font-family:monospace;color:white;background:black;padding:.4em;border-radius:2px;-webkit-box-shadow:0 0 12px 1px rgba(87,87,87,0.4);box-shadow:0 0 12px 1px rgba(87,87,87,0.4)}.terminal-app .terminal,.terminal-app .terminal dummy-screen{line-height:1em;font-size:14px}.terminal-app .terminal-cursor{color:black;background:white}.terminal-app #terminado-container{margin-top:20px}/*# sourceMappingURL=style.min.css.map */
```

</style>

<style type="text/css">

.highlight .hll { background-color: #ffffcc }

.highlight { background: #f8f8f8; }

.highlight .c { color: #408080; font-style: italic } /\* Comment \*/

.highlight .err { border: 1px solid #FF0000 } /\* Error \*/

.highlight .k { color: #008000; font-weight: bold } /\* Keyword \*/

.highlight .o { color: #666666 } /\* Operator \*/

```
.highlight .cm { color: #408080; font-style: italic } /* Comment.Multiline */
.highlight .cp { color: #BC7A00 } /* Comment.Preproc */
.highlight .c1 { color: #408080; font-style: italic } /* Comment.Single */
.highlight .cs { color: #408080; font-style: italic } /* Comment.Special */
.highlight .gd { color: #A00000 } /* Generic.Deleted */
.highlight .ge { font-style: italic } /* Generic.Emph */
.highlight .gr { color: #FF0000 } /* Generic.Error */
.highlight .gh { color: #000080; font-weight: bold } /* Generic.Heading */
.highlight .gi { color: #00A000 } /* Generic.Inserted */
.highlight .go { color: #888888 } /* Generic.Output */
.highlight .gp { color: #000080; font-weight: bold } /* Generic.Prompt */
.highlight .gs { font-weight: bold } /* Generic.Strong */
.highlight .gu { color: #800080; font-weight: bold } /* Generic.Subheading */
.highlight .gt { color: #0044DD } /* Generic.Traceback */
.highlight .kc { color: #008000; font-weight: bold } /* Keyword.Constant */
.highlight .kd { color: #008000; font-weight: bold } /* Keyword.Declaration */
.highlight .kn { color: #008000; font-weight: bold } /* Keyword.Namespace */
.highlight .kp { color: #008000 } /* Keyword.Pseudo */
.highlight .kr { color: #008000; font-weight: bold } /* Keyword.Reserved */
.highlight .kt { color: #B00040 } /* Keyword.Type */
.highlight .m { color: #666666 } /* Literal.Number */
.highlight .s { color: #BA2121 } /* Literal.String */
.highlight .na { color: #7D9029 } /* Name.Attribute */
.highlight .nb { color: #008000 } /* Name.Builtin */
.highlight .nc { color: #0000FF; font-weight: bold } /* Name.Class */
.highlight .no { color: #880000 } /* Name.Constant */
.highlight .nd { color: #AA22FF } /* Name.Decorator */
.highlight .ni { color: #999999; font-weight: bold } /* Name.Entity */
.highlight .ne { color: #D2413A; font-weight: bold } /* Name.Exception */
```

```
.highlight .nf { color: #0000FF } /* Name.Function */
.highlight .nl { color: #A0A000 } /* Name.Label */
.highlight .nn { color: #0000FF; font-weight: bold } /* Name.Namespace */
.highlight .nt { color: #008000; font-weight: bold } /* Name.Tag */
.highlight .nv { color: #19177C } /* Name.Variable */
.highlight .ow { color: #AA22FF; font-weight: bold } /* Operator.Word */
.highlight .w { color: #bbbbbb } /* Text.Whitespace */
.highlight .mb { color: #666666 } /* Literal.Number.Bin */
.highlight .mf { color: #666666 } /* Literal.Number.Float */
.highlight .mh { color: #666666 } /* Literal.Number.Hex */
.highlight .mi { color: #666666 } /* Literal.Number.Integer */
.highlight .mo { color: #666666 } /* Literal.Number.Oct */
.highlight .sb { color: #BA2121 } /* Literal.String.Backtick */
.highlight .sc { color: #BA2121 } /* Literal.String.Char */
.highlight .sd { color: #BA2121; font-style: italic } /* Literal.String.Doc */
.highlight .s2 { color: #BA2121 } /* Literal.String.Double */
.highlight .se { color: #BB6622; font-weight: bold } /* Literal.String.Escape */
.highlight .sh { color: #BA2121 } /* Literal.String.Heredoc */
.highlight .si { color: #BB6688; font-weight: bold } /* Literal.String.Interpol */
.highlight .sx { color: #008000 } /* Literal.String.Other */
.highlight .sr { color: #BB6688 } /* Literal.String.Regex */
.highlight .s1 { color: #BA2121 } /* Literal.String.Single */
.highlight .ss { color: #19177C } /* Literal.String.Symbol */
.highlight .bp { color: #008000 } /* Name.Builtin.Pseudo */
.highlight .vc { color: #19177C } /* Name.Variable.Class */
.highlight .vg { color: #19177C } /* Name.Variable.Global */
.highlight .vi { color: #19177C } /* Name.Variable.Instance */
.highlight .il { color: #666666 } /* Literal.Number.Integer.Long */

</style>
```

```
<style type="text/css">
/* Overrides of notebook CSS for static HTML export */

body {
  overflow: visible;
  padding: 8px;
}

div#notebook {
  overflow: visible;
  border-top: none;
}

@media print {
  div.cell {
    display: block;
    page-break-inside: avoid;
  }
  div.output_wrapper {
    display: block;
    page-break-inside: avoid;
  }
  div.output {
    display: block;
    page-break-inside: avoid;
  }
}
</style>
```

```
<!-- Custom stylesheet, it must be in the same directory as the html file -->
```

```
<link rel="stylesheet" href="custom.css">
```

```
<!-- Loading mathjax macro -->
```

```
<!-- Load mathjax -->
```

```
<script src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML"></script>
```

```
<!-- MathJax configuration -->
```

```
<script type="text/x-mathjax-config">
```

```
MathJax.Hub.Config({
```

```
  tex2jax: {
```

```
    inlineMath: [ ['$','$'], ["\\(", "\\)"] ],
```

```
    displayMath: [ ['$$', '$$'], ["\\[", "\\]"] ],
```

```
    processEscapes: true,
```

```
    processEnvironments: true
```

```
  },
```

```
  // Center justify equations in code and markdown cells. Elsewhere
```

```
  // we use CSS to left justify single line equations in code cells.
```

```
  displayAlign: 'center',
```

```
  "HTML-CSS": {
```

```
    styles: { '.MathJax_Display': { "margin": 0 } },
```

```
    linebreaks: { automatic: true } 
```

```
  }
```

```
});
```

```
</script>
```

```
<!-- End of mathjax configuration -->
```

```
</head>
```

```
<body>
```

```
<div tabindex="-1" id="notebook" class="border-box-sizing">
```

```
  <div class="container" id="notebook-container">
```

```
<div class="cell border-box-sizing text_cell rendered">
```

```
<div class="prompt input_prompt">
```

```
</div>
```

```
<div class="inner_cell">
```

```
<div class="text_cell_render border-box-sizing rendered_html">
```

<p>First project in driverless car Smartcab project 1. Code below and in github as the compiled .py file in the github repository file named agentO.cpy. I am controlling the red car. I have set display to False in the notebook so the simulator can be run inside the notebook. I do not know how to get pygames visual to play inside the notebook and it kept crashing the Kernel so I set display to False and run the display True in the command line for the visual on my computer.</p>

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="cell border-box-sizing text_cell rendered">
```

```
<div class="prompt input_prompt">
```

```
</div>
```

```
<div class="inner_cell">
```

```
<div class="text_cell_render border-box-sizing rendered_html">
```

<p>QUESTION1: Observe what you see with the agent's behavior as it takes random actions. Does the smartcab eventually make it to the destination? Are there any other interesting observations to note?</p>

<p>Actions requested in project:To complete this task, simply have your driving agent choose a random action from the set of possible actions (None, 'forward', 'left', 'right') at each intersection, disregarding the input information above. Set the simulation deadline enforcement, enforce\_deadline to False and observe how it performs.</p>

<p>Answer:I changed the line 44 enforce\_deadline=False (was previously set to =True); I implemented line 28 action='forward'. When the code was implemented the red car being controlled moved in the forward direction to the end of the grid. The destination is set in code line 15 as destination= None so the car did make it to the None destination which is continue to move forward until you can't move

forward. Interesting observations: After the car reached the end of the grid it moved in reverse to the starting position and moved forward again. The forward action continued regardless of other cars being in the way. This is sub-optimal in a real world situation. The reward as you can see below was -0.5

```
<div class=" highlight hl-ipython2"><pre><span class="kn">from</span> <span
class="nn">IPython.core.display</span> <span class="kn">import</span> <span
class="n">Image</span>
```

```
<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span
class="o">=</span><span class="p">(</span><span class="s">#39;smartcab1.png&#39;</span><span
class="p">))</span>
```

```
<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span
class="o">=</span><span class="s">#39;smartcab1.png&#39;</span><span class="p">,</span><span
class="n">width</span><span class="o">=</span><span class="mi">500</span><span
class="p">,</span><span class="n">height</span><span class="o">=</span><span
class="mi">300</span><span class="p">)</span>
```

```
</pre></div>
```



<div class="output\_area"><div class="prompt output\_prompt">Out[4]:</div>

<div class="output\_png output\_subarea output\_execute\_result">

<img

src="  
IArs4c6QAAARnQU1BAACx

jwv8YQUAAAAJcEhZcwAAFnQAABKbAfPP1EgAAP+ISURBVHhe7J0FfBTHF8cvSIJL8eUp07bf91d  
oUK9helU6kbdaSktNUqFFmixUooXd3d3h0BwJxB//3m7s7m9yybZl1y4hft95/O429nZuXdv5x6/  
rMz6CIAzBGOWzhxal05sqEkp99mVLrl+IE9+h3RiVT9BoBsSEhIMEyKMVjHT51FFFudSpWpalSe  
UPbXzXfQ5N6f0xudR9D9tS5XNUm04l/36bWP3zPajFw9ImZ89SrtSiO64qbeNOhglu2b9DZ936oz  
TTxuNAERyttvv00vv/yyXsrl/v379bvMOX484yAyBuvLHa+nft+WMCoYNf5o3/4jJp00ngPgBeA  
ZgVnDBis4lzh9AzW6Gj9xoZPf7T1CkA2hHak/P470ejRekExYQLRqlXmgLTMgt/nz2++li1L9J76  
w81a/+qrgW0BUiR+REyebL5Om0Y0cybR9u1Ec+YQ7dhBIJhINGWkuX74cPN15Eii3buJNmwgGjXK  
rGMmTdJvADDJ+/SVnJx9loyJ0W9cgqwbkYR2r//7r5IJLTztMgeWZT17EtWsqVcqOOMyIsTVahg  
ygFut3Qp0cUXE7VsSdSkibkdc8stRFFRZhsQcYR2rx85QtSrl/neGIDWQGVJUL48UcmSZj3Tu7f5  
Wrgw0aOPeRvR59/u3XfNgblsGdGgQWYdc/PN5qAFEUdoB2tuiY/XbwDISGgHa5cuZmZ8802ideul  
Dh0yjpgkJRG1bWvKBF7P2fl2fR3C3r3mK8N/dFmZtVatwHUG4gn9YGVYg546ZQ68cuXMukqViD7/  
3Kzjgcwaldm3z3xleN1DD5m6tmJFotmz9QoAQj1YAchDMFjBGQMKGKzhjCOlg9YWwuwULFuh3uSeU  
fS1cuFC/yz2hjFdk9BUifOqPly6V+/ajKqvX0XOPPUa1Y2OpxuHDVG3nLt3KHUZF2nKLp/vigr5c  
E5peFJZj1fbupxppadSuVSvic1V878FF731grs/ORipzKk5ts7N/IDkVp7bZ2Z/KnlpTW4mFog/L  
PN5XKAjZYD1y5ljhWCilj48P2RdMSkoKWW9p6kcYqr6YUMWLiYy+QkhmjvnmzdPv/Dy+caN+50xl  
B4VX+wrpjoyEvkJIZjuSB6tv1iy9ZPIQXxKYBRExWNGXiND1pEDwZaAvGaHrSYHgy0BfMkLXkwLB  
l4G+ZISuJwWCLwN9yQhdTwoEXwb6khG6nhQlvgz0JSN0PSkQfBkh66tQDO0uEE1ze/6mK3KHZ+OI  
X0PCWT8oFF7sa1nxknS4aHEaUf5cXZM7PBsv/RoSzvZBwXixr/EVKlByTAz1eqCJrskdno2XfgOJ  
vj4e/ZKh9MuDfbXq2JHatGID4+66S9fkDs/GXr/mmvXr16vOQtPdtm3bQhawuLi4kPnF84qGqq+j  
R4+GrK8h0ZWlYqLpn8v+p2tyTmJiYshib1z4E6LvlySkpylTphhf0Cjqddy4cTmzfeNoiiq+O1Q/  
2lQjtRtlayOwqVOnhsYvZdOnTw9ZXzNmzAhNXzvG0UxV7PGisuWISui5di+6iNMkUcGC5ivPy9Cv

n3Nf28bRLFXsfalaGvefrY3AZs+eHfAdQ0FoelGE0jGjL225JeR+ebUvbcasNhaXXMIriWrXNI+Z  
vn3N10zgPjr4Opp95RLDpxB9RyY0vWjYsVARqi/IhNQvr/YVonj9UaAfJauudvlst8jngtDGK4SE  
dlChLxGh6uv76j9Syw4tablvla7JHSH9jvo1JGBQyPBIX22fbGtk1k/Kfqprckdlv6N+DQkYFDK8  
2Nevxf6gNm3b0Clfkq7JHSH9jvo1JGBQyPBIX0P6jKCH8z1CW3dupb2/7qXjs47Tnh7m887SktJo  
f9/9tOf7PbTrs1108N+DILAxwXh/cvVJ2v3tblp1RaB8COL31K8hAYNCxpnQV8rhFON1YdGFIHw4  
meb45hiDNH5ZPM31zTVsls9/yxIGay5BXzl825d+DQkIvgz0JSN0PSkQfBnoS0boellg+DLQl4zQ  
9aRA8GWgLxmh60mB4MtAXzJC15Mitl5FQvDxHSWErCf+gmwNGzbUNTnH6EuVevXq6ZqcY/IVp04d  
XZNzrL5q8cM5con1HWvUqKFrco7IV7Vq1XRNzrH6qIKliq7J0dZ3rMTPkwbBuR6st6liuhRU9JfO  
seW2jyHKNlpT2+ysnzKn4tRWYrnto4cyp+LUNjv7RplTcWorMd1HKAh5Zs2tY9a0kqH4kul9hdKv  
UPaF7ygiNL1oQuUUw18yVITUL3xHESH9jvo1JHg2YOhLhGf70q8hAcGXgb5khK4nBYIvA33JCF1P  
CgrfBvqSEbqeFAi+DPQlI3Q9KRB8GehLRuh6UiD4MtCXjND1pEDwZaAvGaHrSYHgy0BfMkLXkwLB  
l4G+ZISuJwWCLwN9yQhdT4rQOubRvrz6HSMh9vo113Cw2KKionRNzjH6UsWTFXn1O3rVrxD1xeR6  
sHr2ela75bavSLie1W6h6MMY3VcoCFImLVGihOHU6tWrdU3OMfpSZcmSJbom51h9zZ07V9fknJlI  
Sxrfcdq0abom5xh9qcIT7+YWY68RI0bompxTqlQpw69//vIH1+Qcq6+//vpL1+SO0AkKRah+QQx/  
yVAR0r68+h0jlf6NSENDoS0RE9KVfQwKCLwN9yQhdTwoEXwb6khG6nhQlvgz0JSN0PSkQfBno  
S0boellg+DLQl4zQ9aRA8GWgLxmh60mB4MtAXzJC15MCwZeBvmSEricFgi8DfcklXU8KBF8G+pIR  
up4UONctlyLi5cXByk4ZJQTOGX1ppy0h98urfWnLLV71iwlNLwr7l3zllVdyZqtfoTdU8X2t+tgM  
aumVdrY2AnvzzTdD45eyt956y3t9zXqFOquSIV7P29olrHNN1Vco/Jr0Cr2tSrBfuSVkgzUlJcX4  
kqEgNTXVCFYoMOYJdd3X9G5fiYoX412/Qohvq0e/ZCj98mpfXo2XZwdrKB1DXylioi/9GhIQfBno  
S0boellg+DLQl4zQ9aRA8GWgLxmh60mB4MtAXzJC15MCwZeBvmSEricFgi8DfcklXU8KBF8G+pIR  
up4UCL4M9CUjdD0pEHwZ6EuGr27duhQqY8ec6sVWrx7VD1Vfynz1s+6radOmmVpw2+z6kphn+wpl  
7EPZlx60AHgaY6AeOLSf4hON5RySrF8ByJ6EhATDMuP48ePGK1/FZ+E7deoUrZpQlKaMvNGo2Leh  
N73f6VFq92Jfy/nTux+n8W89Tx8PXktpq7vSnwv2Ubc//qY0t7ah1FPDqW3DDqpVHP2z9zAl7xpG  
n3b8ygtgORCZ8Lezbb7+tlzJnx44d+p0zL770kn5nYmTUru+Yc3oyqSn+kb5v3z79DoDwAo0Kzggw  
UMEZQd4MVC0jsqRkSaKFC/UCAfKt2oHKT94YO9YcqDyHvTVgp0/POHhTU4lmzSLq3t1cbtTlfoV2  
lgGgCe1o4AHJA3DtWqL164k2bzbrV64ksh7kwOuYtDSi7dvN9zy4LYYPJwrBgy3A2UX401Z8PNGh  
Q3rBJUeP6jcgUgjtQO3Y0f/fdokSREeO6BUKzp6FCxNdeaW5/umnzfrPPiP67jvz/cCBRJdfTtSm  
DVFMDFGXLkQzZvj7XLDableBGnGovR9CeDDdeCPRjz8StWhBdPIk0dKI/nV8XLZDB//AY17RExpY  
y0yRIkT//mu+5/qbbvKvj442szCIKGyj4zTj9N+9PQMDYCO0AzVfPqIGDYggVzazlmfB8883/8vn

9+vWma92syhQQL9RXHGF2RcAGttICQGLSxNdfbV/APKFB82amXqT63igMk4Dlf87r1DB1LmsU0P0  
kFlwdmAbKQB4FwxUcEaAgQrOCEI2UKuqEipC8fh2izVr1uh3uWetdVYtBFyhSqi4VpVQcYsqoeJu  
VZJDdFF9SAZqlPrDxxfrU38b+ajCiLHU9tlnqVpilp23YSPd+sWXVOEP98+IN/pS/bDddtttjZn  
2Pu6/vrrdW3OsPf1v//9T9fmDKOvBWZfF110ka7NGUZfs1Rf+XxUv359XZszjL4mq76ifVSrVi1d  
mzOMvsaovor4qDlfBcolIRmoxg5UAzXKV5AqDBlB7Vq3pppJSVRTrWvftg1VHDs+fSdnaZkVp7bZ  
WWbFqW12llxauvW9EANiemB6rhOanqgOq6Tmh6oderUMQdKLgjJQOWZoAvtL0SP3feYrsk53Ffh  
woXpoYce0jU5h/sqWrQoNWnSRNfkHO6rePHidO+99+qanMN9lvxfku68405dk3O4r1KrS9Ftd+Tu  
fx+G+zpn2TI00x036Zqcw32VXVCWrr8zd/+TWYRMO1ZRhdL0Qhb4Zs2i67PRjbn9b8dObv87tHPh  
hRfqd7nnclVCxTWqpKoSCm5WJVGvUHCXKvGqhILQDdQqaqC6gAfq1dn8sRQRA5VPaoSla65RA5Uv  
rwwBN9+sBqr6+yIU3HWXGqghui7jtA9UN2CgysBAFYCBKgMDVQYGqgAMVBkYqAlwUGVgoArAQJWB  
gSoAA1UGBqoMDFQBxhyo02fPoeP5ChAVitE1uQMDVQAGqntmp9OCjhQrQfGFi+ia3lGBKgAD1T1r  
v/ySkmNiKLVQIV2TO87+gfgPRwfsBAO1BEhHKijQzNQV3T7htq0aUNt27bVNbkjpAN1ohqoyR4b  
qFGxUVSxeEW9IDv4ErEyZcropdwRtS6KShYqqZdyR9SKKCoaU1Qv5Y6ohVEUE517XTm383t0lqYQ  
TY0OTbzy5ctnxD8U5BuXj3yFQjPEQtKLcUmXvh6Vr/scN25czkwV3x2qH21XfXiVUefY1oUzf01  
/brssssc27g1o6/lzl98DalTG7dm9KUv82vYsKFjG1cWFK/znj+PqHx53iHmnbw8CUinTua8COaO  
cu6HLaivGm1rGHWOvV2Y8R31ZX7nnnuu+fm5ICQD1bhIVg/Ue+65R9fmjCif6ou/pLJbrs7d1eaG  
X3qg8iGc3GD0pQcqD/rcYPSIB+r5fDt5LrDHq3bl2kSffGIOVL79vKjK/jxQLbg+C7iv23x3UmFf  
EapWtpquzRnGd9QDtVy5cro254RkoDLGZX4hlQaVZVQcaEqoSkkI/mF6DjqtB7TKFmNiKEFRuma  
3OHNy/yKeXSgVgzhQK0ewoFa13sDtcsDX1HLDi2p1fOtaP+R/bo259x1DQ5PuQZnptwzc8osl6Om  
hWiSGhxHFYCB6p6pfadT2/Ztqdlzj+qa3lGBKgADVUbH6p1o2OvDjfdzfHMo5XAKHRp+iHZ9tot2  
f73bqLduOYrrGme+UctpyWm08dGN5rIGA1UABqoM+wH/2b7Zh2ddJRm+WYZdQsKL6BNj2+i47PN  
h5ZZ8KBee2vgvAcYqAlwUGXgXL8ADFQZGKgCMFBIYKDKwEAVglEqAwNVAAaqDAXUARioMjBQBYRO  
oKoSKjw7UD14rp8J6UD12rn+Pn36pF/Bw5Nj5YYBAwaQb3No+hoyZEjI/Bo5cmTI+ho7dmz61VO5  
7Wvy5Mkh82vWrFkh62v+/PnpV0/lti8m1wNVueJc9BfOkemBmivLrDi1zc4yK05t3Voopp3MrDi1  
zc4yK05t3ZoeqPw+t4Qko/LV+JZzuaV8+fLpAzW3VKxYMWWR+8WS0oeqratWq6QM1t9SoUSNkfp13  
3nkh66tu3brpAzUUhKYXBTSqDGhUGd4cqPirXwT+6heAgSoDA1UGBqoADFQZGKgCMFBIYKAKwECV  
gYEqAANVBgaqDAXUARioMjBQBWCgysBAFYCBKgMDVQAGqgwMVBkYqAlwUGWc/QM1lOf6Qzn3VCgH  
qlfP9Z/Nj5i0rrhhW7p0qa7NGUY/+uqpxYsX69qcYfSiZ/NbsGCBrs0ZRI/a5s2bp2tztGPvnpq  
9uzZujZn2P2aMWOGrS0ZRj/66dJTp07VtTnD6EtfPTVp0iRdm3NyPVCVK87FFkCxheJ6VMv0QM2V

ZVac2ro1r12PalkePAad3+eWkP3XbzkXCk1PSpjDdRQENLvGKLrUZmQ+sUDtUCI+uKB6qUZpxlo  
VBnQqDK8OVDxv78I/NUvAANVBgaqDAXUARioMjBQBWCgysBAFYCBKgMDVQAGqgwMVBkYqAlwUGVg  
oArAQJWBgSoAA1UGBqoADFQZGKgyQjdQz/HoQKOWwoFaO4QDtZfHB+pVZ/lAraZKqKhdu7Z+l3sa  
qRIqLIElVFylSqI44YYb9Lvcc7sqoeL+++X73JP5AaqcfWOfp0s2bNdG3OsK4EYnvg/gd0bc4w  
+tFXT9155526NmcYfemns99kPVY8hxx96aunrr76al2bM4y+tDW+tLGuzRIGP/oyvwsuuEDX5gy7  
X3Xr1NW1OSfkA5WnG3zlVdyZqr4vIb9aDtv+HlGnWNbF2b4pQcqT8/o1MatGX3pgcrTRjq1cWtG  
X3qg8rPsndq4sqB4VehXlffx0gO1bNmyjm1cWZBfPf8obdTlhpD9118xuSLt2bxHL+UOHghxcfrx  
hbmKvLitit0Uq5dyR/3E+rRt0za9lDsuTLiQNm/arJdyR+PGjWnDhg16KXdcdfIqWrNpjV7KHSJ  
VqxYoZdyR+j+mNrh0T+mdobwj6ldlfxjKs6jf0ztVn9MpeHwIctweEoGDk8JwECVgYEqAwNVAAaq  
DAXUARioMjBQBWCgysBAFYCBKgMDVQYGqgAMVBkYqAlwUGVgoArAQJWBgSoAA1UGBqoMH19EEgor  
UKCAMSic1kmtns9HdevVc1wnteh60VSvbuZ98SWFTZs2dbSLL744oG1MvZgs+5JYoXqFH0tzYoXr  
FXasz4kVKVKE6tSp47hOakWLFg1ZXyHLqAAAAGwydea/FWjzHPoxPpatHLFcNoyxy87E45vp+nD  
htCOU0SrDqqKE1vUP2nU76+/aO2aOFo35Q/qv2gXxa1aSWP69KXEXUtoyIQ5xrZ/9/2DThrvIPYI  
EC0druqTd9LhLasoLSWO/vtvKK09pFam7aQxq0dS/yNE80YMpjFz19COhf+qFcdov/p3+uLQXPkH  
Alg89u/nLGkyY8cO/U7O5f/7n/HapOmDxqsTRILdu3cvjRo7iLp8/Q1Nmz2VzvX+jj7p1s1owOzb  
0JtGTJxEB1OIUIPSaPFevZmaQGPHjqO3u01SCfg/WrN4ER3azJfwptGWvSfoaBwnXqLJY8bQvG3H  
jfexq1apf1No26EEOrxjE6WI7qeENKJ1C9YbiXraqtW0J1kl7umTadH63aptKm3ZF0+Htqv1AACQ  
Q5YsWUKrV682bOHChbpWxhiVy06dUspSsXv3btq3b5/xPhj8+Q8AACEESRUAAEIlkioAAIQQbydV  
ngDHP1ycM4dowACiqCiiokWJBg826/laqbQ0okcfNZfvyYfo88/N9//ySS5F2bJEnTqZ7xlru169  
iPjYCi8zXbua77kvht+PH2++Wg9z5ImCeHn7dnMZAACC0BnFg7z+OtHWrXyE2UxkjNPrGj0nDb/n  
M3znnmsu33uv+frdd0QTJ5rvGW5nXdDJ79nGjiUqXNisu/RSojZtzHoLft+7t5lwk5OJHnErwAA  
gEBsmQMAAEbu8XZSZaXKI2FZt+QkJRGtW2cqUmbCBFNFcv3PP5t1zMaNRFvMS7qM93v3mtvxK8Pq  
9uhR8z3DyrRcOVOfHjxltHYt0SG+eFbBbY8dM+v4cw4cINq2jSghgWiznlGSP4P7t+Bt+HILVsRH  
jpjr2U/7ZwIAzKq8m1SHDCGKiTHfW3+KW6/ffMNx8JpJN3id9cqJml/B8v2ur71m1lnrgnnnHaJK  
lfimSqlmTcy6a68lat7cvw0nSH7Phxf4UIF1mIGPs7ZtSzRvXkYf+Nhv/vzme65DUgXgrEf/+gEA  
AISCyEyqlpLMjIEDic4/n2j58sC2fEKLH33DKpbVMgAABOHdpNqIC1Hp0uZ7Tmz8pzRfJjVzJIHf  
vub6W2/IRwL5E1+LFkQXXURUsCDRTz+ZxzmtdfxqXRplT5QWx4+b9YUK8XxbRP37m/VObfnyrFM9  
ACDiQWYAAIAQgqQKAAAhBEKVAABCiCeTahVVfMj3II+4XBUvj9rVGG/UIXxEjerwn4lquill7IKF  
/YpXxUt4amStXlmSfD4VpIglSrq/csdO9HrU6bSE6NG0c/LV9AVP/1BP69dZ9j7c+ZSx6kz6PIJ

U+nveQso4cQJ3VNoWbNmjemXz dq0aUNpPO9AGFm/fn0Gv5577rmw+7Vp06YMfj399NNh92vr1q2m  
PwuU6fH16KOPht0vnjTZ8GuW9iufjx566KGw+7Vr1y7Tr8nar2gf3X///WH3a8+ePaZfY7RfRXzG  
c3vC7ZeFZ5Jqv nz5zECx6aQa5StKNdU6tmoHD1Hb tm2pXatWVDMpKb2+qrJ2qr6VSnKVfY/x95Fb  
G6IMUpz6yAv7R5mkOPWRF/anMklx6uN0mS2pespsdVxfbjMllQd14fLbEnVqvMCnlKq6cGyKVvf  
iZJUOyGBapyMp3oJSVR99Vq6ZMRoumj4CKr/629U89gxqnXyJDWI202/Dxupewot6X4FWTjh/5Wd  
fGILJ2eEX0FJNZwE+BWUVMNJgF9BSTWcBPgVIFS9gnc8sZGTy6of7dyp32Vkzalt9EVc7p9xxY94  
9dLOs+Ang3rRL370rBf9wjFVGTimKsN7I0vBz7qW/hh9fO+9ft2amEj3rV9Pvfbto5OpqeSbPZsO  
JSfTZStX0t32iU+ElKnK8GxSvVwIVQ/6xc8ZZ79C9azxUMHPLGe/QvXc8IDBx1HZr1A9Az1UeG9k  
KXKUVOfoNV7f4oP+KolGz59vLDM+vgTL0WnbNmNdTg9nl6nKQFKVgaQqA0IVQE6S6ukASVUGkqoM  
JFUZSKoCkFRlIKnKQFKVgaQqw3sjS4GkKgNJVQaSqgwkVRneG1kKJFUZSKoykFRlIKnK8N7IUiCp  
ykBSIYGkKgNJVYb3RpYCSVUGkqoMrybV/113nSf9QIKV4b09qEBSIYGkKsNrSXVWp5foULHixm3Y  
PKfEb1Vq0oB6DfXa8IOkKsN7I16BpCoDSVWG15Lqojfept/OPZeSY2IMO1moMH16zfV6bfhBUpxh  
vRGvQFKVgaQqw2tJdVCjS2jEuTWpW+PG9JWy7y68hHrUa6TXhh8kVRneG/EKr86n6tmkqooX43Wh  
KI70y2v3/k956GHaH12K1pUpRyvKlqMJORXp7yLF9drw49mkinv/s4d3nGG2Waqi+Nn9YSbdL5t5  
yq+1yrwYr+Ue9cs2SxX7xTMfhRP247zadal3gQo0r2ApurpqXflpvxyvAnDytVMnotq19YIN1W9u  
SI+XzTheycnJukV4SPfHNksV+5WUIKRbhBfPJFX7jguY+i9cNkqZpDj1cbrMllQ9Zbak6inzwnyq  
PZS5KFS2PNFHH1k/ksDXqVPN9/akaq1j9HvHz8/MvlEmKU59nC7DfKruMIJjS6rDhg3Ta8KLtdMs  
GzRokF4TXgx/bEm1Lz++2wMYftmS6m+//abXhBfDL1tS/eGHH/Sa8GL4ZbNu3brpNQ6ULUuUKEB0  
1VVEdevqytCzfNNyiiuw1blW0dpKlOciDpFPxT6Sa8NL0acbEn1/fff12vCj+eSKoNjqjJwTFWG  
Z+dT9djF/7Pmz6HnqrejtU3bGpd7tXq+Fx3DdZrww+OqQrA2X8ZOPsvA3dUuWPx4mXU9equZpZg  
iyL6sNTHxjovgLP/ApBUZSCpykBSdce8kYvpmQeepWQVKrbUfETji0/Sa8MPkqoAJFUZSKoykFTd  
sWLLCkpV6vSd/71Lrz74Gq2KWUP9CgzUa4kWFI9le3/aq5dCz7p719Gyesv0UkaQVAUgqcpAUpWB  
pCojs+tUZ/tmU1pCmvHKxL4fSysuWEGzfLNoaa2lLlAlgRYUXUA7399JiyssJkohWnXFKjq59iIt  
7r6bknYn0Vyf+cQO3mbIpSspeW+y2V+aSurnrzDaZwaSqgAkVRllqjKQVGVklVSZo5OP0sLCC2n9  
PeuNZU6QyYfMa1mPTjhKi8stptTjqTQvynyOHMNJ9tCIQ7SI5RZKPZlqbGNxbNoxWhCzgPb+ufi  
umb+wE4kVQFIqjKQVGUGqcrAbaoyvDeyFEiqMpBUZSCpykBSleG9kaVAUpWBpCoDSVUGkqoM740s  
BZKqDCRVGUiqMpBUZXhVZCmQVGUGqcpAUpWBpCrDeyNL4dmkWgNJVYJnk2pjJFUJnk2qdyCpuqZa  
tWqeHPS1VfHiPeMNGzb0ZLwuueQST/p1pSpe3I/XX3+9J+N1++23e9Kv+1Tx4n70IEe845zs4Ycf  
1i3CQ7ovm5Vx0cv33XefbhEe0v0KMv4RhBMnn9huuOEG3SI8pPsSNPXflVdeqVuEh3S/goz/Uwon

Tj6xNWoU3qcSpPsSNPVfbac5ZcNA2JNqS1UaqNJQFd8aFRwHK7GzhNHGKOcqa3D6zFKBhgUl1eLF  
iztuk6c2XpkqWcWr2l5iVrSoQWVITv2E2kYpU6WRKk4+sRXZVsTyihpUUebUTx4ZJ4L0/RiUVAsX  
Luy4TZ7aAGWqZBWvQlsKWdGiBtWUOfUTauujTJWs/lrZHGN5RQ2qK3PqJ4/s/PPP9+/HoKTK672A  
p5Qqz8CekJBAPuqVMoL0119/6TXhhf1KSUmhskfLGjuxT88+ek14Yb/YKISoYMT999/12vCizWT  
fuXKIQ2/fv31V2M53Fh+Vd9b3diPP/XwxytYgl/WiVCvzPFq+VW3bl3Dr+7duxvL4cbyq+F2JSxU  
6fZTFnPPHgHvHZBQePZEISq8E70GTITJwHyqMjx7ogrzqboHl1TJQFKVgUuqZOCsKhneG1kKJFUZ  
SKoykFRlIKnK8N7IUiCpykBSIYGkKgNJVYb3RpYCSVUGkqoMJFUZSKoyvDeyFEiqMpBUZSCpykBS  
leG9kaVAUpWBpCoDSVUGkqoM740sBZKqDCRVGUiqMpBUZXhvZCmQVGUGqcpAUpWBpCrDeyNLgaQq  
A0lVBpKqDCRVGd4bWQokVRllqjKQVGUGqcrw3shSeDap7lFJdZf3/Kq/VyVVD/rl2aS6TyVVD8bL  
s0n1gEqqKl6JaR5LqgdUUIV+xachqWbJsGHDjIFlWVJSkl4TXsaMGUO+jXynsemXV/7XnjhxlvlW  
+f3iCWm8wNspUwP248mTJ/Wa8DJ79mzyzfXHyysqZ/78+QHxOnHihF4TXhYtWkS+8f54HTt2TK8J  
L8uXlyffSL9fR48e1WwCT9iT6veqdFCloyq+HipADlZ7fG2jjVGeVdbh9FmnTp38gz1o6r+aNWs6  
bnM67IUXxvD7tdbwKn2ZJ/l22iZPbYEyVv5UxWkfsIUeXtnai9ShhTKnfvLIXnrxJX+8gqb+q1ix  
ouM2eWrTlanykipOsWlRn7icFS3q8Jwyp37yyF5++WV/vCYr4xJtLpcpU8Zxm9NhAX4FTf3HU3F6  
AU8p1fRgBVmTB5roFuEh3ZegpMrHdMJJul9BSfWmm27SLcJDIC/K75vNrr7qat0iPERFab+Ckmrj  
xo11i/CQWbzOb3S+bhEe0uMVIFR5KsBwku5XUFJlMeEFvHdgSeHZY6penfpPFS/65dljqjp6T8TN  
qnC8ElXxExpj6T4Bnk2olj579r46z/xluPx9n/yXcfLVHz/7fhLP/rsElVTJwSZUMXFIIA5dUyfDe  
yFigqcpAUpWBpCoDSVWG90aWAKlVBpKqDCRVGUiqMrw3shRlqjKQVGUGqcpAUpXhvZGlQFKVgaQq  
A0lVBpKqDO+NLAWSqgwkVRllqjKQVGv4b2QpkFRlIKnKQFKVgaQqw3sjS4GkKgNJVQaSqgwkVRne  
G1kKJFUZSKoykFRlIKnK8N7IUiCpykBSIYGkKgNJVYb3RpaiiipevDfbs0nVq/f+q+JFv3DvwwzP  
3vuPpJo96dN6xfluVIXfK0tLS9MtwkPnzp3TfbFbuP368MMPTV+CZqkKt19ffPGF6ctyb/nVvXt3  
05egWarC7VfPnj3TfbFbuP3q3bu36UvQLFXh9mvAgAEBcbls3H5ZeCapxsTE+AMUIFTDYqOUSYpT  
H6fLgpKqZywoqXrGgpJqWlznTJUUpz5OlwUl1bDY58oEJZx4SqleeumlZgCDkmq4ufLKK/0712bh  
5rrrrjN9CUqq4eaWW24xfQlKquGEVcydd95p+hKUVMMJ+3X//fen+2K3cMJ+Pfzww6YvQUk1nLBf  
jz/+eECcLPMK3vHEBo6pysAxVRk4pioDx1RleG9kKXD2XwbO/svA2X8ZOPsvw3sjS4GkKgNJVQaS  
qgwkVRneG1kKJFUZSKoykFRlIKnK8N7IUiCpykBSIYGkKgNJVYb3RpYCSVUGkqoMJFUZSKoyvDey  
FEiqMpBUZSCpykBSleG9kaVAUpWBpCoDSVUGkqoM740sBZKqDCRVGUiqMpBUZXhvZCmQVGUGqcpA  
UpWBpCrDeyNLgaQqA0lVBpKqDCRVGd4bWQokVRllqjKQVGUGqcrw3sgCAIAzGCRVAAAIUzSPRGf  
RgePpdKYKUtp5OAONGLMZFWXYjTIG9Jo//GMf0oc3Ban3wEAQGjgwykJCQnpltPJrl8cOaLfEe3e

vVu/y4iRVJMO9abEzRUP9cAjtHL6lbR0TBlavfRvowGzdtEy6j94OO2YM9JYHjxpHa2bMYamr9yo  
0iNRn9591b9ptG33Rho9dy31/fMv4px5LHYR/dV/sbHN1sWj1b+HVSGauXQ3rd5xnFZNnE39Bg42  
1o/q3ZteiL5KvYunP//qZ/QbG0+0bMgMopTMvwAAAGSFPRkynFhzSr9+/alr16/0kjNGUh02cgjt  
XP8qpW6vSrT/fPr8wztp7dr1RgPmlRa9KDF+K+2kBPpr3WI6peoGztpB2+f/RiknllCbT9+mfoel  
vpgcR1O/elOtTaW3fl5KI5a/gzq/95HRR+zCP6lbnw7U4+seNFtlzHf7raN2V3dWyXMC7ds/jpJV  
m3cKX0PfPviC0b5l71X0/ns/0vgdE+mtZ14y6gAAkcXw4cPp7bffTrecEMqk+kCTptSk6YN6yRkj  
qT71bDMApMYSjRw7mTpfdbF9/PgjxkqL9ve9QxMmT1fv0qjLC6wmiVYunEHDfv1cJcNUGjV1Ae1J  
Umr0wCk6tH2T0W7zrqMUv28djRs71mhPquWOo8kUt261ep9GW/bE07rF3PYgnVT/jhk9mjZMW6Le  
nalxY8cbSnX/5nVG242qLwAAyAmcVFevXp1uOU2qTz/9jH5HdO211+p3GUK/UfXHVyXoiQdL07j+  
FTlcLzhxjLWpyvDHDhQKkjm4fy8dOmLWAwAAMEIPqgAAAHIPkioAAIQQJFUAAAghSKoAABBCkFQB  
ACCEnDIJIS/NymwSDL684fHH9UIW8IQQ3lc1YUVM/QEAQA7xblaZNMIMetdcYy7ze7aVK4mqVTPf  
Dx1KIJTKX3fqFNE55xDVq+ecOK2katVZr7xd9epEhQsTbdxINHiwua50aaIGDcw2a9YQRUURNWli  
LgMAgAO2jOMxOKk99xzR6NFER48S/feFwbdpk/n+1Vf9SfHKK4meeMJctqxAAaLge3ytpDptGIHZ  
sv7tOVkeO2a+57opU/zr+PXwYfPVsuefN9cBAEAQOnN4DJ4f8e67zfecxZM8CfVl14iqI078HDA  
FVcQNWtmLi9YQDR+PFHXrua6Dz80XxkrqTKciK330dFEgWYR7dXP1vHnWev49eBB85WT9GuvEU3n  
u8sAACAJOnN4kB07iMaNI1q1ylzmhMbjkhMjJz3+s3813/Kq4PdJxpjvJ08mWrTIfM9wMrbgQwLD  
h+sFxUhzhgiDmTOJpk413+/Z49+O2/Nn8mfW57NSBgCATPBuUgUAgDOQMzupJuuZCPjP/weznjkm  
S55+mujqq/VCHmEdTgAAAnNV495fOf3Lzn/5bt+oKBR8S2LDBfM/HXTIRbdtm/ok+caJZv3cv0dq1  
/hNPfNZ+/36idTzjlcY6bMAcOGD2w6/M5s3+P/F37TI/n/uztrFeuV8+JMDTii1dShSnJ9jmbS0f  
GPaxDzsgqQIQEXj3l964sfNkyYiPrd50E1GbNmaiLV/ev4656iqiRx4hevFF/yVQfPKJ23lbTqq8  
LV8ixTz6qPlqYfXDr3xs1bqci68+yJfPVMG8zFOGlsVnP6HF8Gcz1vJ555IXHjRvbrb9+Weitm39  
6wEAZzXe/aVzElq2zHzlpMrJjVUK062b+WpPbJxUa9Qg+krPyl2qFFH//v42vXqZidbpUQpWG+s1  
JcV8z4n40kvNunbtiDp3Ni/vuuyyWGtKitbalpMqJ3eG67iv7dv96wEAZzXe/aU/+SRRTIx5/Wn3  
7ubhgKpVzeTEf14zrAZ5+cIL+Xm1Zt2tt5p1771nLlvJrEcP/3vr1cJa5j/pWWUWL060ZYt5oT8n  
Sca6VpWxb883GjRqRHTvvabKLVOGqFURcx3fSMBtBw4kuu46sw4AcFYTIF0AAADkhshLqsePm5YV  
fliAlSfvgOAAAlil6nynVq33KIXMoH/ZOeTUnxCbO5cXangep68hS/hAgAAB7ybVDMB8QQnfLkS  
T3byxhtE+fOblyXvrkw0apTZ5uRJ89U609+6tflq9cFn4HmClEKfzGOdfJ+/IVstdsFY9TzzlTWWh  
C8PHW3mdmxmxAAARSSZZxQNYia1jR9M4MVq3mPI6Pmn1wQdEf/9tzgPAf65XrEj044/+ba1XxnrP  
J7ScIOpFF5lt+BZVq+3DDzvPSmXvFwAAbHg3O1ijy5oEhaf7Y8XKtGhh1i1fbr7yMVA+887vWZla

21qvDL+3pgy8/np/nRMXXOBvy9hf+bltVr0AAOBAJlKFAABATkBSBQCAEIKkCgAAIQRJFQAAQgiS  
KgAAhBAkVQAACCGeTKo+XQDIC7w6vuCXDO/65UG8GqzFixdTsvW0AQ/BfiXy9bweg/1K4Nt9PYZX  
xxf8kuFdvzyEz+ejKF9UerB4eeisOfTk6NH00Zy59MuKlXTv2In0y7r11G3RYnpzwWJ6bsxY+nzm  
bFowd77uJfQYfkUpv9SrZV7Aq37ly5fPk34VLFgww/jyAkWKFPgkX8WLF/ekX6VLI/akXxae8YYD  
k266lKhdl6ps2UaVJkyhy/r0pfYtWtDlf/Smmvv2Uf2tW6netu10Rddu9L8eP1FN1YfD9NO5JsCv  
lAsnTv5YfK6c/LEsnAT4YhX4lSlmI/8n7hXCG+EbDgFq7JSo5ws2Zrfcw+1adOGru32dXod233N  
HjXqqy5fGdhHbk1anPrIC5MWpz7ywqTFqY/TZV7wwcngl8yC/Proo49URgg/nk6qJZ57gcr++gdV  
3bmL7mjdRifVboFJ9eFm1KHlc8SzAgT0kVuTFqc+8sKkxamPvDBpcerjdJkXfHAy+CWzIL+8gmc8  
Wbt2rWOw6q9dTzUSTIGdY8epXlIKNRRowiC4e+R9d0Lcv1Z02g2oeP0511bpG/f/WPYWWLVu2+P2y  
WZUqVXSL8BAbG+voV6VKISjN6Tlcp4m9e/c6+lW+fPmw+nX48GG/P7bxVaZMmbD6deLECue/SpUq  
FVa/+MSnk198nDWcfvFnO/lvtGjRsPplxvpXXPjRTemB8tL3HDDdek700tcf/31nvTruuuu86Rf  
1150rSfHl1f9uubia+CXEO95pJAG6/d9+++ifgwf1Uka6795NsfqSo9z8X+bFJMHALxnm6lJfboFf  
Mrznkcp60mBdtWoVPbJhg17ywwn0ZGoq+WbNMPz98+YZrkFyUuGJ/3Kwfg6LcAvGV71S+E9jxTS  
H6OVVI+mpNCTa9dSbFIS+fSzpRbxMavZs+mYWsdJlZNSvFkklDALxnwSwb8kuE9jxTSYFIJ9c0d  
O+g79ae+nV18wB1KNSzALxnwS4Zn/dKvnklarMtVUn1Q//kfM3++oUyb6uWdnFT5USuKAmrdXevW  
Ge9zgmd3lwwSAb9kwC8Z3vNlgZ0oA37JgF8y4JcM73mkwE6UAb9kwC8Z8EuG9zxSYCfKfG8y4JcM  
+CXDex4psBNlwC8Z8EsG/JlHPY8U2lky4JcM+CUdfsnwnkcK7EQZ8EsG/JlBv2R4zyMFdqIM+CUd  
fsmAXzK855ECO1EG/JlBv2TALxne80iBnSgDfsmAXzLglwzveaTATpQBv2TALxnwS4b3PFJgJ8qA  
XzLglwz4JcN7HimwE2XALxnwSwb8kuE9jxTYiTLglwyv+bWJTAWi6GhaF1OYqEAB2u3LZ8wF7BWw  
H2V4zyMFdqIM+CXDS34lJifTiQIF6YGnnjEebPlcp45E+QtQ4qkE3SL8YD/K8J5HCuxEGfBLhpf8  
OpWQQAcLRNPxYsXpcFFlxUrQEwWaePKkbhF+sB9leM8jBXaiDPglw0t+paSm0gMNGhHFxFCytjsu  
aQyl6gLP+qVfPQV2ogz4JcNrfr10w40BSfWNG26i5KQkvTb8YD/K8J5HCuxEGfBLhpf8WrlkCb1X  
vRZ1veJy+qpxY8Ngl6lAP773oW4RrAfZXjPI4WvjwrWr9iJbjHi9Qv8covX/BpZoDC1bdvWOFHF  
1v38iyglJUWvDT8Y9zK855FChcooXsOzg8ur8YJf2XLI5EmakP8cWli8DC0pV47iSpakhQXPocT4  
eN0i/GDcy/CcR8YO9GCwrIHlthl6XjBr2zhE1Un8hekIdGljOOOpU6LL0vz8pSgplVG3CC/WmGfz  
El4dX4xnPBo6dKh/B1rFAzvzv//+8/sVZOfk4sSjfl88FK/p06d70q+5c+d60q9FixaRL8pH7xas  
SvEqqX6RvwrlK1xl5pdTW64bNUovyFmxYkV6flItNkXdu9bvi4f2ox3PeJleKldgTZ06lcaNG3f6  
7D9luviuVj7c4WxTVUlv6dRPHlpW8Zo2bZrjNqfDsvKLE67TNqfDPOXXWGW6+K5TPjiMLbb9lGom  
R7ZPPiEqUYIoOdlcvulK89X88RBdcAFRhQpE111HdOiQWdewIY2aMMHZByez+3W9s09sM1VJb+nU



Tx5aVvtxz549ZjzCjN4r4ScqKirTYIXFpMWpj9NIXvDByeBX5uai/EgD/Inzm2/MpHrLLUQPPGDW  
WeusV04q9roRI9SL6kli0uLUx+myIB/eeecd87uHGb0Hwk9WwQonAX4FWTgJ8MWr8YJf2RLgl82i  
fFFWA/PVSqp795p1V1/tX2e92pNqoUJE551nvs8Blh9FfMWova8jPeB7xPh094sJwwQFI+G2fyj  
izKvoPeAN8iXL5+x0+zB8gLR0dGBStojfhUrVsyT8SpVqpQn/Spfvrwn/apSpYr78fXuu0R33kl0  
/Lg/geYRo6uMpR75fqVk9TET802ncfkmU9Kp8N+UUL9+fU/uRwtveaOxguU1sh3wYcKz8YJflrw2  
vnb79tHi/MuNpMrh2plvD332+mfmSg/g2f2oXz2FZ4OFpCoCfsnwXFItv4+eeP4Jate2HbVp24ae  
b9yJdm2K02vDj2f3o371FJ5NXvBLBPYs4SW/Tp08RbMKzaOnX3yaWnZoaViHa56n9y/9XLclP57d  
j/rVU3g2WPBLBPYs4TW/BtcdppxSb7SNqDKKDsYeMtZ5Ac/uR/3qKTwbLPglAn7J8JpfvWr9rpwi  
45gqW9ervqJVC1frteHHs/tRv3oKzwYLfomAXzK85teC4kuVU/6k+vP5v9HYfybpteHHs/tRv3oK  
zwYLfomAXzK85tfuAnvpsTueNE5StW3XlVb6DILPr37Ra8OPZ/ejfvUUnG0W/BIBv2R4za8V1VfR  
pkJb6bWmr9Nr176hHCRKOG4+keDAXwdopm+m8T6vmOWbpd8549n9qF89hWeDBb9EwC8ZXvNr1/5d  
NO7DcfRgvofDt+dNKzXaKM+aU8SzfHNoe1vbDeWDw07RHt+3EMpR8w5YHd9tosO9D9AcV+bl1/F  
fRFHyQeT6di0Y3Ri8QIKjU+l3d/upv199hvrU46lGNvs/XmvsXxo+CE6Puc4zfPNM5Yzw7P7Ub96  
Cs8GC36JgF8yziS/ZvtmU+L2RFp23jLa9dGu9Dpmrm8u7fttHy2ru4yOzzaT4443dqQnyU1PbzJe  
FxZbSLu67KLE2ERjm/1/7KeNj2ykTU9soqRdSuiqocSzwYJfluCXjDPJL06gp9adogVFFID8InNC  
bSOppvn/bE9LSKOVjVfSzo93GkmT2zLz88+n1GOptLCUSqpfmEnV2mZJ1SWG6k1LTENSdSWeDRb8  
EgG/ZJxJfllJlVl4xpaXHoxS81k6v9WKj1fnmD5RT7fqzxfk+PPbSk0hLjsEDc53EBSZVZXnc5  
bXthG23rsE3XOOPZeOIXT+HZYMEvEfBLBvyS4Vm/9KunwE6UAAb9kwC8Z8EuG9zxSYCfKgF8y4JcM  
+CXDex4psBNlwC8Z8EsG/JLhPY8U2lky4JcM+CUDfsnwnkcK7EQZ8EsG/JIBv2R4zyMFdqIM+CUD  
fsmAXzK855ECO1EG/JIBv2TALxne80iBnSgDfsmAXzLglwzveaTATpQBv2TALxnwS4b3PFJgJ8qA  
XzLglwz4JcN7HimwE2XALxnwSwb8kuE9jxTeDZZZvIZn44X9KAL7UYZn96N+9RTWTkxLS9M13sDY  
hWke9Mur8cJ+FIH9KMOz+1G/egLecVFRUek7kc0LGH75IF9W8ZJfXo0X9qNr8uXL58l45c+f35N+  
FShQwJP70clz3th3XLCfkwBfbDvRU34FWThx8seycBLgC/Zjtj5Y1k4CfDFQ/vRTtg9SQ+M22IP  
6uk2L/ngtjj1kRcmLU59nC7zkg9ui1MfeWHS4tTH6blgHz799FOdVcJL2JNqA118K1Vg1jhbQ1Ws  
dg0anF7Laic2bKj8ctgmz6ymMl2yilcjVdJbOvUTaqlTBffKmf2E67XzbLaj82aqT8ctgmz6y2  
MI08Fa86ynQ5E/ejV/CMJwcPHgwMmDI+ntOqVauwHog+duyY3ydd2K/mzZuH1a/4+PiAWLGxX08/  
/XRY/UpMTMzgF9vjz8eVr/4s9P9sYp636xZs7D6xaT7ZbMHH3zQk3498MAD3vHLth/vvffesPtl  
4Z30rvn555/Tg+Ylenbpmb4TvcRPP/3kyXj16NHDk3792OVHT+7HH39UfnkwXj/88IMn/fr+i+89  
uR8Z73mk8OJOZDy7E70aL+xHEdiPMjy7H/Wrp/DkTIR/WXh2J3p10HvVL+xHEdiPMrznkQKDSwb8

kgG/ZMAvGd7zSIGdKAN+yYBfMuCXDO95pMBOIAG/ZMAvGfBLhvc8UmAnyoBfMuCXDPglw3seKbAT  
ZcAvGfBLBvyS4T2PFNiJMuCXDPglA37J8J5HCuxEGfBLBvySAb9keM8jBXaiDPglA37JgF8yvOeR  
AjtRBvySAb9kwC8Z3vNlgZ0oA37JgF8y4JcM73mkwE6UAb9kwC8Z8EuG9zxSYCfKgF8y4JcM+CXD  
ex4psBNlwC8Z8EsG/JLhPY8U2lky4JcM+CUDfsnwnkcK7EQZ8EsG/JIBv2R4zyMFdqIM+CUDfsmA  
XzK855ECO1EG/JIBv2TALxne80iBnSgDfsmAXzLglwzveaTATpQBv2TALxnwS4b3PFJgJ8qAXzLg  
lwz4JcN7HimwE2XALxnwSwb8kuE9jxSeDdYu5Vcs/HIL9qMMxEuGZ/3Sr57Cs4NLF6/hWb+wH0Ug  
XjK865eHWL9+ffrAioqKop07d+o14WXDhg2mX6pEqbJr1y69Jrxs3LjRk35t2rQpYD96xa/Nmzch  
xCsuLk6vCS9btmwJiJdX/Nq6dasn47Vt27YAv7wyviw8k1Rfe+219IEVbOGkc+fOfI+s4gG/3nvv  
PU/69fHHH/v9CrJw0qVLF78vHorXV1995fcryMLt99+6/ffQ/H68ccfPemXnbB70kEX39cqMD2c  
7QVVrHYdOpXeS9+BDjvxhReUXw7bnA7Lyq8XX3zRcZs8s9bKdPF9o3xw2ldsL6qS3tKpnzw0T8Wr  
rTJdfN2VDw6xYntJlfsWTv3koWUVr5deUn45bHM6LCu/Jk+erLNKeAl7Uk0PjNtiD+rpNi/44GRe  
8EtanPo4XeYIH9wWpz5OI3nBBycL8uvtt9/WWSW8eEYzBwQryMJJgC9BOzGcnBF+BvK4CfAF8cqW  
AF+8Gi8P+WXHO54oatasGRC0ggUL6jXhpU6dOqZPukRHR+s14aVRo0YBfsXExOg14eXiyy8O2I+F  
ChXSa8LLFVdcERCvwoUL6zXh5brrrgulV5EiRfSa8HLzzTcHxKto0aJ6TXi56667AvwqVqyYXuMN  
PJVULazB5TWSneg1POsX9qMlxEuGd/3yIbHcMjzrF/ajCMRLhnf98iCeHVzwSwT8kgG/ZHjWL/3q  
KbATZcAvGfBLBvyS4T2PFNiJMuCXDPglA37J8J5HCuxEGfBLBvySAb9keM8jBXaiDPglA37JgF8y  
vOeRAjtRBvySAb9kwC8Z3vNlgZ0oA37JgF8y4JcM73mkwE6UAb9kwC8Z8EuG9zxSYCfKgF8y4JcM  
+CXDex4psBNlwC8Z8EsG/JLhPY8U2lky4JcM+CUDfsnwnkcK7EQZ8EsG/JIBv2R4zyMFdqIM+CUD  
fsmAXzK855ECO1EG/JIBv2TALxne80iBnSgDfsmAXzLglwzveaTATpQBv2TALxnwS4b3PFJgJ8qA  
XzLglwz4JcN7HimwE2XALxnwSwb8kuE9jxTYiTLglwz4JQN+yfCeRwrsRBnwSwb8kgG/ZHjPlwV2  
ogz4JQN+yYBfMrznkQI7UQb8kgG/ZMAvGd7zSKFCZRSv4d2d6NF4wS8RGF8yvBsvD2LswjQfpaWl  
6RpvYO1Ez/nl1XjBLxEXXzK8Gy8PYQUpvaj3XiDdL5t5gXR/vBov+OWKdL9slpqaqteGj3R/bPFK  
TEzUa8NHul82i4+P12vDjzdGlsSladyBZOAvwKsnAS4ltX4wW/siXArYALJwG+IF4iPONJQICC  
dmJYTFqc+jhd5gUfnAx+ZW7S4tTH6Tlv+eCyhBNPJFU+JulYQL38xhtv0CuvvHL6rJMyXXyfKR++  
drY3VUlv6dRPHlpW8XrzTeWXwzanw85Ev9566y3HbfLMXISmi6+L8sFhbLG9pUp6S6d+8tCylfn  
zp0dt8kze0mZLr4vIA8OsWLrrAq3aa9KOPGMUK1KSsqwE6OiomjUqFG6RXhISUnx+6WN/Ro2bJhu  
ER4C/iOyxWvIkCG6Rfhw8mvQoEF6bfhw8qt///5hP9GR7pc29qtPnz7e8csWr99//91z8WL7+eef  
w+6XhXcORGj2rN2TvhO9xO7du9N3oJfYvVb55cF4xa2N86Rfu9bu8qRfO3fu9OT4il0X68l47dix  
w5PxYrznkcKLO5Hx7k70aLzglwiMLxnejZch8W1VwdqMnegWz8YlfonA+JLh2XjpV0/h2WDBLxHw

Swb8kuFZv/Srp8BOIAG/ZMAvGfBLhvc8UmAnyoBfMuCXDPglw3seKbATZcAvGfBLBvyS4T2PFNiJ  
MuCXDPglA37J8J5HCuxEGfBLBvySAb9keM8jBXaiDPglA37JgF8yvOeRAjtrBvySAb9kwC8Z3vNI  
gZ0oA37JgF8y4JcM73mkwE6UAb9kwC8Z8EuG9zxSYCFkgF8y4JcM+CXDex4psBNlwC8Z8EsG/JLh  
PY8U2lky4JcM+CUDfsnwnkcK7EQZ8EsG/JIBv2R4zyMFdqIM+CUDfsmAXzK855ECO1EG/JIBv2TA  
Lxne80iBnSgDfsmAXzLglwzveaTATpQBv2TALxnwS4b3PFJgJ8qAXzLglwz4JcN7HimwE2XALxnw  
Swb8kuE9jxTYiTLglwz4JQN+yfDVrVuXvGQNGjRID5bT+rDZOedQfeUTm+P6MFn9uvXJV1/FS5nT  
ejf28MMPU9OmTUX20EMPOfZlWSj8yguDXzKrX1/55cHfo1f9YvNemgcAAAAAAEARJFTTjH+/6uyj  
hMN96l/u1Shx3zu0au4r1Kb51fTVRzXp3fe/MdoAAAAAACQlwQl1aXzfqHJA4rQmMkLaOC/46nb  
R5fQiQ21qfNHPWnarJm0bMOKGvF7Cer+7W96izONVDq4fRntP3BSL2fFRno+3+XU77BeBAAAAADw  
IEeOHKHY2Fg6ceJEgMXHx9OmTZsolSFBt8x74uLi6JLJG9O2bdt0jcn4CRPowYceolOnTukadwQl  
1d96/UZPNi1JCvVOUwKVrbZ+rUUnN9aiqX+fS6unVKbJEwfrLfwc2NSTqkY1plFbjtDuye/QvQ//  
RMkz3qMCL04z1q/86kJ6dkQqbf3vdXq4fV86fjKRej1TI+54fzQlphGIJCerVpvp5dl30ICjRDM+  
bkq+mi/QjqTD9M2Tdaha2wGUcvlo/dTiQXp3/BZKSxtH7S7uqOSkYkwHeuafg/zOYNEfrei9iYel  
FrXL5//vTuq1iujowKZ0409baNH3b9A7/dZR6ql/6YLCF9DwtWpHDmlO1W/4mZJoJ71cOlo+XkSU  
MOtTKuy7hladIBr3yi30RI/1qnlf9WzxCL07YR8dGtycHu2/k2htd7rssmvow1plDy2Pf2vm2oH  
AAAAAHCaYKG6f/9+vRTlJh07TqtQtUhLS6MffvzRuO51ghKpOSVdqKaqDnfu3kJXXHkBPdumDW3Z  
sYtefK0zNX28BVUpW5ZuueceeuKhB+nDLI30FoHs39iTrrngMVq27RCtH9WZWYrYfQXzcMnF9T7q6  
zi302ehlRru1I9+iJq1+pt17D9BPTzeguz6ZQjT9dTr3oV/p2JGpdHth8yjm+Lfa0YejWY0n0aQu  
L9M7/ZXapBSa+9s79OZPSknSJGpa+SL6Z2Ec7d27lw7Fs9A1ST2yjT5qWosu+WKJcmAf/fni1XTF  
F7xNMk3u+iq989daJVSHUcsrXqMdqRzM/+i5+h0ojvZR55JR9MbUNDo4+UMq4ruYhiqhOrnzfnRo  
tyV0fP9m6v7MC/TXur1KtJ6gX1rUpoZvjVcdJNPozpFTfe9NMh0AAAAAADhNeE2obtiwgYoWLOHH  
jh0zltmHKtWq0549e4xICQFHVleP+I9uuLoU7VtagWjPRZS0tRolxt1KX3R5iK697Tpq064ttW3T  
VrcOZO/636jNw9/SYSX8TNJoz+p5NLxHJyUs1+g6olOHT9OM8aNp7Ljx9PQ1tejnhXv1lbEAAAAA  
AAD4CRCqfJgWAAAAAALxAgVAEAAAAAAPAKEKoAAAAAAMCTQKgCAAAAAABPAqEAAAAAAA8CYRq  
XhPOG9RSU4n++ouof3+irVt1pebQlbOeDQAAAADAg0CohhJLIM6YoSKrQssm4dprzW2efJXZEN2  
ljgx0e8H2wMPmOKVWbjQXw8AAAAA4EGgUqTwEUoWd4UKEY0cSTRmDFGRImYdi78DB4i++MlvAjdv  
JvrmG/P9eecRTxI1Lu3f/369abgjI0latjQrLvXrn4GGdH55xOdcw7RkCFEy5YRde5srv/kE+2M  
omRJs65DB11hwy5UDx8mKlvWfM8+LV7sX8fs3EIUp465fOutRJMnE3Xr5m8zZQRqFH+5UcfJZo+  
neiZZ8zl4sXNz1i3zlx+6CGipUvNNjVrmjGaPdv8LAAAAAAAFyhFACsSwkXUsiVR4cKmKctYkKhB

A1MAMv/9Z9azMSxgWSSyWOvYkWj7dv967sviyivNOj6i+u+/5nuuYwHIduQIET/IQT/plVvsQtU6  
kvrqq+ZymTL+dQyLX35fsaK5bNGzp1lfqhTRtGmB21hYdezfu0+Z70ePJjp61PSbX9ln4fN9AQAA  
ABDZBCkOkC3ffWceTbXEGVuBAkTXX0908KDZho8cxsSY68qVM4+EWkdd2VgMRkeb78ePN7dhWORF  
RZn1jz1G9P33/m0sq1HD/zlMtWpm/Wuv6QobTkKV4SO4JU411m8+y5R/vz+ejb25847zfVDh2bc  
hrHq9u0zBeltt/m/Bxu/f/BB3RgAAAAAwB1BigMAAAAAABvAKHqFcI5OwAAAAAAGaEUA0XW7b4  
T41b8A1MgwcTzZmjK04jX35p+sl3RfFNYgkJesUZhhVTvi4WAAAAAGc0NpUEXMF39V9+OVG+fH5R  
xNeoXnqpf65Sfr3kErPeasMC8NNPzfV8XSrX8bWb1rWcfF3r1Veb7/mOeYtHHgm8vpWNZwL48Udz  
feXKZt255/qvi2Xja2D79DHbMFZ9v366wkbttv71bHj1/HjRC+8QFSsWOA6Xn7jDXO7xx8369if  
K64w3zdvr7y7AZ8XazVN/fJsBDn5WuuMZdnzflfGYZ35z24ovmeqZ6dbP+wguJqly11ycnE/39  
t38mA7aLL/a/h1AFAAAAznjU/+hADB/xbNrUnBrKfvPRuHHmemuZby6yYNHGP3iZzFrtbO46ipz  
mcUpw+KWI3//3Vy2YGHl9TzNVa1a5nv7zVS9ep1LFbd8s475jaVKpnLli/Bc7ryjVvc36yZX4Ty  
NFoW/J15miuODYtdXt+3r/nao4cpaDImO3boDRS//ELUuLEp/u3i3oKn9eLIFi10hWLAALOO+7Lg  
o8DWthCqAAAAAwBmPTQ0AV/Bd/yyE+CgnT9fER/4sccRzqjlvvWQu8/yhPJdo167mkVMWcDyFE2Of  
oornNGWCheqwYf42f/5pzrl6333mMt/9z33xKy9nJ1StfuxHWe0EC9V58/xHez/+mGjTJqLXXzeX  
+cjtihX+o6d8NNkOHZW1ZkZgQc9YR1/Z+NICi/LizTpev3atKVitdhaWUOUjvBZ86UTt2mY9z+nK  
87fWr+/fFkIVAAAAOOOxqQHgKdzcXOW2jfRGrZx8dlbSD/fjn1aLQAAAAABEFBCqAAAAAADaK0Co  
RgL8SFQ+HX7ttboih/Cpf+u6WTa+kYmfXOXEiRNE11nTv5v2Q03EP3wg24AAAAAAJA1EKpS+BpV  
vpmpbl3zOk4Wa3xjFD8F6r33iBo1Mm9wat/e/8jQn34yn0Z1xx3mMIOOqHkNK1+rynew86NN+bpO  
6658fhwrX9/Jd7nzdZz85Cv7tZ18zSi352tl+VrVevXMx5Yyu3cTPfywue1nn5kikYXlbeY6xm+  
fpWNZzFwA1+Hyn3wjU8Mn5K//36zzn6TkwxPDMDfkb8fAAAAEAOgFCV0qWLKc54iiQWoAwfJeQ6  
J2PByPDUSjwN04wZ5k1lfEMSi9oOHUyhyW35Tnpm2zbzSOXUqUQpKUSTJpmfx22sz+Sbh6zPeP55  
UygzDRuadV9/bS7zkU1r6ii7UM0MvIPfuqmLjYU4i1lr2S6WeUYAruP+7cTHmyK8dGmiCRNMH/ho  
Lre9+27dCAAAAAAga5RyACIsocoizIKnieK6Nm3Mo6PM2LFE3boRzZxpLjPchgUcv/Ld99acov/7  
n3n0kwUqM3y4Wc/G7N/vnyPVSaja4flbua5zZ3M5Ls4UxFxnf6pLlphm+ZsdfMSX++ApqfjmKJ6G  
yrrDn6fPYjG7fLk5MwHPdHDzUR33WUKbcb6rlWqmMsAAAAANmgIAMQwafXn3mGqGNHXaFhwcfi  
k4+QdupkCrNgVq40p1Ji0cfwvJ98irxIS/MhAHb4yCtPc8XXhbKA5adW8dyhLID5s/j0Pvf12GN6  
Axt8uv3tt01ROWqUKZb/+Ydo+nTdQMfHb9mOHdMVLuDP/eMP87IGnirKmo6L6d/fPFL85pu6QjFo  
kBknjgfpocpHVgEAAAAAXAKhCgAAAAAPAmEKgAAAAA8CQQqgAAAAAAwJNAqAIAAAAAAE8CoQoA  
AAAAADwJhKpLflSliY4fqAIAAGckfVsx8tf7qoCs+VMVe7zSVAGZM0AVe7xSVAGZ87cq9nglqQly

B0LVJVU8dkKAACcKVyuCvKXe65RxR6vVFVA5tysij1eiaqAzLILFXu84lUBmYOMIQlpPKm9Ii4u  
jsrkK0P54/KnDyoul9S9x5FifHfHHj9HeXbv00tmLFa8DBw5QmTJlKF++fOTzqUgppq1SpEjXihw6A  
dKx4HTt2jM4555yAeFWsWJHq8SNxQTpWvE6ePEmlS5em/PnV71HHq0KFClS7dm1jPTCx4pWSkkKI  
8pei6OXRtuzloxrVaxjrgYkVL4bjFbMwxhYtH1WtXlWvBYw9XiULIKRCswvZouWjc2ucq9cCJjhe  
hacWtkVL5bCaFfRa4ASEqgOzZs1K/08w3WJ5ONmKri/96GPEKb9GYiJVXrSMKo6ZQBVGjKpQ/w6n  
6pOmUrVFi+mC19+g259tTg90eJ7ub9eeHnjmGbritlvpnldfo7qr11JNtX0tZfzKViU1jR7q+esZ  
c7Jp4cKFgbHKxliURTLlly93jEtWFsmsXbvWMSZZWSSzZcuWjDFZoMxegtZHMrgXsRni4ZulzF7y  
KbOtj2T27t0bEAvDJiuzl2hltvWRzMGDBwNiYdgYZfZSRJltPQgEEQli+/btAQMm3YKEapSvEJ3z  
0TdUYdhoJUzHUvWjR9PF5kMPP0Lt2raltsratW5N1339NdVMSjLWB1s1Zfc3e4w6tG9PHdq1o7sf  
f4L4b/eio8fQho0baaOXbIOyE8psZYcqvnlqIrVldmu7WylWFXtfrImvzOmzz0TjeB1XZis7VclJ  
vK5/7npjW3tfrjnb4nVMma3sUsVX3jkmWdmVT11pbGvvyhU7wcbPfu3c75Kxuheumlxpnpj5z6  
PJttz549GWJhWDZC9fzzzzdi7dTn2WyOlPUtG6HKZ4g41k59ns22b9++gDikWzZCtUaNGpTs9vHm  
EQCEahA8SKKioglgjWEZhGoRqjjqP6qycXP6kVC2akeP0VN33221DZt2lC7Vq3o2m6mULW3s4xF  
6X3NHjWEbRtlzZs3p3O3x9K58xeQz8mPcNsoZXlZnD7zTLYhyvKyOH3mmWz9lOvlcfrMSLBshCos  
yLIRqrAgy0aowoIsG6HK9v77uOnRAKLVAR4kGcSqw6n/YoUKUZV33qdzlyyjCqPHUfn+g6nCvyOp  
Zp9+dN8jj9GDLZ6jO956m84fOl gumjGDrh4+nG7s149u+6M3XfRlV7rl19+oztx5dN7iJVTrx55U  
bZMpevlSgnq/9DZ8ORMw4uWTierbr7xdbx158PeXxuv6i6/XW9s4U64NySX8/aXxuqL+FXprG5EU  
r+d8lY1QvfDCC/XWkQd//wzxykKocctlvobcikF6rNiyEKrclo8QRipWDNJjxZaFUOW25cuX11sD  
Bkl1Ew4dOuQfOPyfZCbXqP7www90eN9+WqVE5oltW2nB5i20fss22rF9B8Xu2Ekf/Pw7fdC7L22P  
3Umxu3bTjp1xtHnbNlq3ZQut2R5Ly7fuoMVqm+Vbt9PGzVtp365dIJZ85k1VceTlEX+8gn+UNvvq  
q6+M9vaLyyORo0ePUoECBYyYZBWvzz//3Ggf6fHim84KqT8MOSZZxevDDz802kd6vE6cOEHFihUz  
YmLkr0yEaufOnY32kR6v+Ph446bG9HhlllRffvVvUVkwzc18k2M6fHKRKh26tRjbxHZnDp1iqpU  
qeKPV5BQjSpi5jQ+CwsyAqHqEkxPJaNWrvrGD88ykDX169dHvATwEUDEyz2YnkpG8PRUmEc1a4Kn  
p8K8oFkTPD3VSVVA5iBjueR0CNWrVq0i37x59MiGDbrGHfuTk6n0okXkmzuXyqrXZfHxdAX3NwCo  
3bd+vW51eoFQlQGHkgNcVQaEqgzMoyoD86jKwDyqMpCxXFKlmHnYPi//Y3QSqmtPnqR7ldhkEeqb  
PZsuXrGCxh85otcS9eK7CqdPp3xqOzbfzJkUs2CBIVK5L97mlrVrT/vxAAhVGRcQMiBUZVxeVwIV  
xMs111yghKotXqmpEKpZcfNISqja4pWYCKGaFXddq4SqLV586QnIHGQsl1jXl1iWFWQL1a5xcabg  
VGKTxahhPMerFqwWu1RSMOqVWVy9erXR10PCo7OhAkJVBoSqDAhVGZdfDqEq4ZprlFQl3HwzhKqE  
u+6CUJWAjOWScAjVhSdOGEdlL165kvbpOdXe2LGDLIPLffbvN5YZJ6F6pe6rCU79nxFAqMqAUJUB

oSoDQlUGhKoMCFUZyFguOR1CdXtCAq07dcoQnhZ8yn5vUhJtVPVs/D6Y5LQ0WnPyJK1WZpGi6rar  
fri/PQ7b5DUQqjlgVGVAqMqAUJUBoSoDQlUGhKoMZCyXnA6hejYBoSoDQlUGhKoMCFUZEKoyIFRI  
QKjKQMZYcYSqDAhVGRcQMiBUZUCoyoBQlQGhKgNCVQYylksgVGVAqMqAUJUBoSoDQlUGhKoMCFUZ  
EKoyLFcAqEqA0JVBoSqDAhVGRcQMiBUZUCoyoBQlYGM5RIIVRkQqjlgVGVAqMqAUJUBoSoDQlUG  
hKoMZCyXQKjKgFCVAaEqA0JVBoSqDAhVGRcQMiBUZSBjuQRCVQaEqgwIVRkQqjlgVGVAqMqAUJUB  
oSoDGcsIEKoyIFRIQKjKgFCVAaEqA0JVBoSqDAhVGchYLoFQlQGhKgNCVQaEqgwIVRkQqjlgVGVA  
qMpAxnIJhKoMCFUZEKoyIFRIQKjKgFCVAaEqA0JVBjKWSyBUZUCoyoBQlQGhKgNCVQaEqgwIVRkQ  
qjKQsVwCoSoDQlUGhKoMCFUZEKrZ8/brb1LHitWpW4my1DOmOP0WXYx6FixO3QoUptdKV6AXHniY  
duzqcVsDOxQMiBUZSBjuQRCVQaEqgwIVRkQqjlgVLNmRo9fKC46hhIKFaZu9S6gR5q3oJYdOtAT  
bdrQs9fcTicLF6XU/AVp6KOPU1Jyst4KWECoYoBQlYGM5RIIVRkQqjlgVGVAqMqAUM2c+FOnaOpj  
T1BawYK0tGhpatLyOXq2eXN6xjK1/PRFV9PSmKL0b6lZKBGilmQqjlgVGUGy7kEQlUGhKoMCFUZ  
EKoyIFQzJzE5mZa370gfX3UtDalbnyi6ICXHxARYYqFC1PHe+6IPvUaUIJCgtwQWEKoyIFRIIGOS  
BEJVBosQDAhVGRcQMiBUS2blp19Q2+tvpcH16imhGp1BqCYUjqEO991PP9xzHyVBhGUAQlUGhKoM  
ZCyXQKjKgFCVAaEqA0JVBoRq1qz/qhu1uOYG2hUTTaSEqWFKsKZboRj67oKL6bcmD1MaZgDIAISq  
DAhVGchYLoFQlQGhKgNCVQaEqgw1cw5fw4DbjnAdriy0/3PfkUtW/Xjtq0aRNgbdu2pUEVq9Cg  
6ufRkoUL9ZbAAkJVBosQDGQs11RRRQ2p9AKyBkJVRn1VrLHFBWTNhaogXu65XBXEyxk+Prrw409p  
hO8cevnqyq6hdi+eoRctW1KJ1a2rRqjW1at6C2j34MP1esDwNveFmnPp3AEJvxl2qWL9FLvGqgMxB  
xnJJVVXSh1WaGba0tDTjFWSkdu3aAYmLQbwyp6EeqnRhflNgYIUQL/f8TxXEy5kTJ0/StAcolUF  
itPGmFI0J6acceo/RVmqst3RxWhhdBkaHl2ehhcuirv+HbjtttuQ7wXcq4r+NeL36AlzQiAD48eP  
T//RRfmiyBebPqzMotfdfffdRvtIH2RTpkzxytKxUu/D7bbb7/daB/p8Zo5c2Z6TlzxtVa9txe9  
jo9UMJEer7lz5wbGa7mOk1X0uhtuuMFoH+nxWrRoEeXLI88frwXpkTKLjhc/gYmJ9HgNGTilehcq  
SiNiKtLcgmVojxKnfBPVgZiitDy6FP2SvzKNK1ycmjS60GifZbw+/FD9z6r+a2VzS+vWZvuKFXVF  
FrzwAtHzzxOtXKkrTj+rV69WWj7GGENZ5fvLLrvMaB/p42v9+vVUpEgRlybG73GMio+9FFGm1l18  
8cVG+0iPVzAQqg7Yf2jplolQtaxp06Z668gjOBZujK/RiVSc4pGZULXMEqyRSHAsDMtEqFpmCbBI  
JDgWhmUiVC2zBEUkEhyLwkp4VY7KT+cpqxZVgEpFmYlfbhdccIG5MR+ouPJKov79ib7+2lz+5Rei  
unX5wmCzJSU6evbkC+REr7zCyoXosceInnjCXNepEztibrd1K9HjjxPdeivRZ58RWQ8YOHHiQ6Omn  
zXZsNWsS/f67ue40EhwLN1aPZ1OIUJzikZlQtaxGjRp6a8BAqAYRHR0dMGDSLruhGjE2SIlleFqfP  
jATLRqjCgiwboQoLsmyEasTYj8pCVH4gJU7LljdFY5UqfFqJ6NAhom7dzLoSjcz/VF5/3Vxm4crr  
Wbi+955Zx8ZYQrV6daKUFLNuzZrANhZW3ciRelF5k1f2tbK8LE6fGQmWjVBI++STT4z9C9QY169A

YR1uDx4whmUjVAsWLKjyUomz2wooG6ssqPiOqBgclVmBkwVsPdiKT5nTZ59FFjx2DMtGqBYooOLI  
0FckWHAsDMtGqObPn9+xr0iw4FgYlo1QjYh4FVL2m7Kg4pSfsrUTPvqVhviFql1UfPONWac+08A6  
rc9HXo8dM+u++MKsY2MsoVq7trnM7NkT2MbCqhs+XC/6Mn7XUFhhZT8qs5WSqjjGlXvLdyKfrRdb  
ya/M6bPPEitZUsVL7Z8Mlo1QjeQj0E4E/QIAExsbm/HlahZCNZKvi+PvvHfv3vTrldxYJF8Xx9/5  
0KFDVKhQocC4ZCFUr+T/4PS2kQZ/Z54+yLq+K92yEKqRfF0cf+eEhAQqWrRoQEyyEqqXXHJJ+raR  
Bn/nIJQUKlasWEBMsjKeGs3Ylv+54w5SAy7wFPyAAUSNGxPddJO5zPOuHj1K1KYNqeRH9O67RNOm  
+QUnw0dh+YjrQw+Zywyf6m/QwDQ7mzYRPfig2f6ff3Tl6cEalyVK8h9EUXRjmUvpgzKf0Jf5vjfU  
xBHfCRoY9S91i/merqp5FeXz5adGjRoZ20Tq+GlyCNYshCpPVchEYrwyA0l1E6xBwvObPf/E81Ts  
sEpkvtLeK+8Z64GJFa+kpCTq0KFDhh9m586djfXAxlpXqvpPrPOT7an0ntK20eWjn158w1gPTOxJ  
u90T7ajsrK2aPnoIRde0WsBExyvCtsr2KLloxc6vaDXAsYeL54ztVKISgH563m+eSmnqJxoXI+q  
/jhVnZlWqhTRAw8QxcXpRmcWk4dMod2F9lLvAv1pbv6FFOfbR8nqa6UoO+E7RePzTaOZ+ebRLt9e  
+vfPoXqryMU+vto82Yaqr65u+zX6qNULrfRa4ASEqswj6oMzMqA/OoysA8qjlwj6oMPutjz1/8  
ByXwM6XfVNPryBeNzTeFJhadSj0a9KRPrvqUPrvqc/rugh9oYKnBtM93mJb4VtAf3/XWWwELzKMq  
AxnLDeqPoQxC1f8HEnAAQlWAGksQqgJUvCBUBah4QajKgFDNmmnLF9CB8keo7XXtqE2HwKd4WU/y  
6tz0bRrs+5f2bt+ntwIG6vcloSoDGcslelSqDAhVGXiEqgw8QlUGHqEqA0l1c/ia3sGDhtKA+n/T  
2ErjaGyx8TSk1L80pGSgzS29gD4r3lWmjZ5JiUI4UpUdPEJVBjKWSyBUZUCoyoBQlQGHkgNCVQaE  
atYsWriEhtQdRp3ufcFQEXx9qt1S8xH1bPgzfXZOFzqw46DeClhAqMpQQwq4AUJVBosQDAhVGRCq  
MiBUZUCoZs20kdPpp8o/093X32OoCL6Jyi5U05RQ7Vz/bfqq+Nc0ZdQUvRWwgFCVoYYUcAOEqgwI  
VRkQqjlgVGVAqMqAUM2aqb2n0858uym24C6aVHEqDawzmAbV9tvfNQbTrgJ76JQviX7o0kNvBSwg  
VGUGy7kEQlUGHKoMCFUZEKoylFRIQKhmdzSJ02lriVhKzpdKn1/2BbV+vrX/RqoObenhO5sZ6mK5  
bxXNn7pQb5UJZ9qNych+5sB/CFUZYfgugVCVAaEqA0JVBosQDAhVGRCq2fP9jT1NBaEt0ZeqLMVc  
jjLr/rnnX0pITFAlfub55tFs32w60P8ApRzXj4s9w9j42Ebj6y6YpWukQGHKoOHFHABhKoMCFUZ  
EKoylFRIQKjKgFCVcfPNNwfEKzEx413+JxafoEXlFhkCz7KdH+yk47OO04liC4zl+dHzDeP3C0su  
pONzjxvzbvLNorm+ubTmxjU0xeHNjTdQAtLLaTFFRZT6qlUil8cb9SzhZt2jNKS02jtlWuN7ZLi  
kijlcAqtvHRIhs/gzz0y7ojxGYmxiQGfw+/3/7mf0hLSaMVFK4x63m5B4QW0IMb0F0L19ICM5RII  
VRkQqjlgVGVAqMqAUJUBoSrDjVC14KOpbKfWnaLkA8mGuGRRaCftVJpRx+v41Dq/Z0vamaRbEMV1  
jTPE457ue2hl45W0rPYy2vn+Tlpx/go6NuMYzfTNpD099ujWREcmHDHE68JiC2lp1aU0P/98Y/td  
X+wy1ltClX2zMz9qvuHH4VGHDQ3RxmY4ono6QcZyCYSqDAhVGRCqMiBUZUCoyoBQlZFToZqWkmal  
SxaMCwotoLgv42jXZ7toFj5TRMZ1Mx8xawnV5EPJxRLF+nvWGyKS+0vYnEDxS+JpcbnFxrZrrl9D

qfHmfjs285ghLLI+x5s7aNMTm4xl3nbnRzuNNpZQZbNzaOgho938AvMprkscbW23Nb0vPkqbEyBU  
ZSBjuQRCVQaEqgwIVRkQqjlgVGVAqMpwLVTPIBun8thPCFUZYfgugVCVAaEqA0JVBoSqDAhVGRcQ  
MiRHVAGEqhRkLJdAqMqAUJUBoSoDQlUGhKoMCFUZEKoyIFRIIGO5BEJVBoSqDAhVGRcQMiBUZUCo  
yoBQlQGHkgMZyyUQqjlgVGVAqMqAUJUBoSoDQlUGhKoMCFUZYfgugVCVAaEqA0JVBoSqDAhVGRcQ  
MiBUZUCoykDGcgMEqgwIVRkQqjlgVGVAqMqAUJUBoSoDQlUGMpZLIFRIQKjKgFCVAaEqA0JVBoSq  
DAhVGRcQMpCxXAKhKgNCVQaEqgwIVRkQqjlgVGVAqMqAUJWBjOUSCFUZEKoyIFRIQKjKgFCVAaEq  
A0JVBoSqDGQsI0CoyoBQlQGHkgNCVQaEqgwIVRkQqjlgVGUGY7kEQlUGhKoMCFUZEKoyIFRIQKjK  
gFCVAaEqAxnLJRCqMmotVkJ1tooV23TEKzsgVGVAqMq4fLkSqv9ugZCVQaEqoy71iihavs9xqdC  
qGYFMpZLIFRI1FJFRSg9gKyBUJUBoSrjclX4d2gVkDUQqjlgVGXcpQr/Dq0SrwrIHGSsLDh27BJN  
nTs34AfIFhUVRyUKFTLaIIH5OX78OC1ZssSM0yb++ZklSpX8vVxGm5SUFOMVmPFatWqV4/jiVwbx  
8sPxWr9+PeLIEo7Xli1bzDjNU2b7PVrxSk5ONI4B0YkTJyg2NjZ9XFlmH1+Ilx+O1+7du7OMV1JS  
kvEKyDi9f+DAATNOo5U5/B4RL2fM6IAafvjhh/QfXXZm/Sj379+vt448fVnllwxx8W1WZi+63orX  
3r179daRR+/evf1xysaseO3Zs0dvHXn069cvQ1wyMytecXFxeuvIY/DgwRni4lugzF50vRWvnTt3  
6q0jj2HDhvnjl1Z8dqY4feOviYPXp0hrhkZla8tm/frreOPMaPH58hLr4xyuyliDJVb8WL/8AE  
fiJeqA5T5Xtb6aGKr6UaNJ1kdsKf1BPVex9GaWrsu/PXvvp58Cf4CWZSJULbvggguoZ08VL4c+  
zxr7RtkmZbbysyo5GV8NfmpgbGvvyhfKXP67DPROF7rINnKL6r4WjnHJCur830dY1t7X0Y5m+LI  
YL/++muG35phmQhVY2rWrGnwenU51lj3ZWtUGYrv6ria61i4DCGsrKqXasa29r7Mko3ZU6ffSba  
t8qWKRov31TtXGOSVZ2bpdzneP1tTKnzz5LrFevXhl+a4ZllQtK1u2LK1cuVKrFBDxQvU2VYyB  
klFFNvgiyrIRqhFjQ5TlZXH6zDPZ+inLy+L0mZFg2QjViLEeyvKyOH3mmWzfKMvL4vSzkWDZCFW2  
zp07a5UCcOrfAR4kUT7zELxbu+eOe/TWkQd/f+uURbplI1RvueUWvXXkwd9fOr6uvfJavXXkwd9f  
Gq/LLr5Mbx158PfP8HvMRqjyGY5Ihb+/dHzVqVIHbx158PeXxlvbVq9cXW8deRgxCP49ZiFUuW2F  
ChX01oCBUM2CEiVKBA4uB5s9e7ZuDcqUKeOPTSZCdcqUKbo1qFixoj9emdjYsWN1a1C1aIXHGNlt  
+PDhujU477zz/LHJRKGOGjRltwYNGzb0xysT69u3r24NLr74YscY2Y0vRQEm//vf//yxyUSo8uUC  
ICMQqi7B9FQyMD2VDExPJQPTU8nA9FQyMD2VDExPJQPTU8IAxnIJhKoMCFUZEKoyIFRIQKjKgFCV  
AaEqA0JVBjKWSyBUZdQqh0eoSoBQlQGHkPuPy+niEqgQIVRkQqjLuuh6PUJWAjOUSCFUZeNa/DAhV  
GRcQMvCsfkQqjlgVGXgWf8ykLFcAqEqA0JVBoSqDAhVGRcQMIBUZUCoyoBQlYGM5RIIVRkQqjlg  
VGVAqMqAUJUBoSoDQlUGhKoMZCyXQKjKgFCVAaEqA0JVBoSqDAhVGRcQMIBUZSBjuQRCVQaEqgwI  
VRkQqjlgVGVAqMqAUJUBoSoDGcsIEKoyIFRIQKjKgFCVAaEqA0JVBoSqDAhVGchYLoFQlQGHkgNC  
VQaEqgwIVRkQqjlgVGVAqMpAxnIJhKoMCFUZEKoyIFRIQKjKgFCVAaEqA0JVBjKWSyBUZUCoyoBQ



IQGhKgNCVQaEqgwIVRkQqjKQsVwCoSoDQlUGhKoMCFUZEKoyIFRIQKjKgFCVgYzIEghVGRcQMiBU  
ZUCoyoBQlIQGhKgNCVQaEqgxkLJdAqMqAUJUBoSoDQlUGhKoMCFUZEKoyIFRIIGO5BEJVBosQDAhV  
GRcQMiBUZUCoyoBQlIQGhKgMZyyUQqjlgVGVAqMqAUJUBoSoDQlUGhKoMCFUZyFgugVCVAaEqA0JV  
BosQDAhVGRcQMiBUZUCoykDGcgmEqgwIVRkQqjlgVGVAqMqAUJUBoSoDQlUGMpZLIFRIQKjKgFCV  
AaEqA0JVBoSQAahVGRcQMpCxXAKhKgNCVQaEqgwIVRkQqjlgVGVAqMqAUJWBjOUSCFUZEKoyIFRI  
QKjKgFCVAaEqA0JVBoSdGQsl0CoyoBQlIQGhKgNCVQaEqgwIVRkQqjlgVGUGY7kEQlVGrSoQqhLq  
14JlIQChKuPyiyFUJUCoyrj5OghVCXfdCqEqARnLJRCqMmqpoiKVXkDW1FcF8XlPhKqMy1WxxhYX  
kDUQqjJuVsUaW1wSVQGZc5cq9njFqwlyBxnLJdWqVQtIXEaWprxCjJSRxxR+hLQ7yyo5EqId7  
LrnkEvweBVylih5dGF8uuOGGGzC+BNyuih5dGF8uuF8VxMs9ZoRABrp06RKQqDKz4sWLG+0jfZB9  
8803gbHzrMxedH2xYsWM9pEerx49egTga60ye9H1RYoUMdpHerx69eoVGK9MrHDhwkb7Sl/Xn3/+  
GRibBcrsRdcXKITIaB/p8Ro8eHBgvDKxmJgYo32kx2vEiBGBsZmszF6ilan66Ohoo32kx2vs2LGB  
8RqjzF6KKFP1BQsWNNpHeryCgVB1lGBAubRmzZrprSMPp3hkIJQte+CBB/TWkUdwLAzLRKhadued  
d+qtl4/gWLiXm266SW8deTjFlzOhatnVV1+tt448gmPhxho3bqy3jjyc4pGZULXsggsu0FtHHvY4  
pFsmQtWyunXr6q0BE/FC9TZVjIGSV8U2+CLKshGqsCDLRqhGjPVTlPff6TMjwbIRqhFjPZTIZXH6  
zEiwblQqLMiyEapsn376qVYplOKF6ruq3Gsr96niG6oGyiiZVVpcybjuxN6XUW5Qdu/Za/fff3+G  
H5hh2QjVChUqGEdVnfo8m42/c3AsDMtGqJYrV+7sj9ftyqYqs5UHVmnJ77HMnDLGtva+jHKTMqfP  
PkusSZMmGcaOYdkl1dKlSxvbOvV51tgdykYrs5UmqviGqRg4jKGsrOSMksa29r6McrMyp88+S6xp  
06YZxo5h2QhVvkTurB9fDpZpvLIRqg0aNNAKBTA49e8ATxURPA+ok0VFRRmvX331lbFdpF5XkpCQ  
YJyqCliPg1C14vXFF18Y20VivPg7JyUlUcOGDQPj5SBUrXhZf1lHarz4juvgu/ydzlrXR99l5t  
pGF950svwTQwPg5C1YrXe++9Z2wTyfG64oorAuPIYFa8OnfubGwTyfEKnhUHK6H6xhtvGNtEcryC  
b87LSqi+8sorxjaRGK/MgFDNBGuQsKiYMWOGcUTLPtDmzp1rrAcmVrxYVEwfMZ0qnazk/xGqMnvS  
bGM9MLHixa/TRkyjqieq2qLloxkTZxjrgYk9aU+bNo1q1qwZ8HucOnWqXgsYe7ymjphK9Y7Ws40u  
H02cMFGvBcFMmTKFGjVqFDC+xo0bp9eCYCaPmEwXH7zYNrp89N+E//RaEMykkZPo8v2B08WNmDhC  
rwVOQKi6BPOoysA8qjlWj6oMzKMqA/OoysA8qjlWj6oMzKMqAxnLJRCqMmolKqGaoGKIDWRN/UQl  
VBEv10Coyrg8SQlVjC/XQKjKuDIJCVXb+IJQzZq7kpRQtULQjVrkLFcAqEqA8/6l4Fn/cuAUJWB  
Z/3LgFCVgWf9y8Cz/mUGY7kEQlUGhKoMCFUZEKoyIFRIQKjKgFCVAaEqAxnLJRCqMiBUZUCoyoBQ  
IQGhKgNCVQaEqgwIVRnIWC6BUJUBoSoDQlUGhKoMCFUZEKoyIFRIQKjKQMZYCYsQDAhVGRcQMiBU  
ZUCoyoBQlIQGhKgNCVQYylksgVGVAqMqAUJUBoSoDQlUGhKoMCFUZEKoykLFcAqEqA0JVBoSQAahV  
GRcQMiBUZUCoyoBQlYGM5RIIVRkQqjlgVGVAqMqAUJUBoSoDQlUGhKoMZCyXQKjKgFCVAaEqA0JV

BoSqDAhVGRcQMiBUZSBjuQRCVQaEqgwIVRkQqjlgVGVAqMqAUJUBoSoDGcsIEKoyIFRIQKjKgFCV  
AaEqA0JVBoSqDAhVGchYLoFQIQGhKgNCVQaEqgwIVRkQqjlgVGVAqMpAxnIJhKoMCFUZEKoyIFRI  
QKjKgFCVAaEqA0JVBjKWSyBUZUCoyoBQIQGhKgNCVQaEqgwIVRkQqjKQsVwCoSoDQIUGhKoMCFUZ  
EKoyIFRIQKjKgFCVgYzIEghVGRcQMiBUZUCoyoBQIQGhKgNCVQaEqgxkLJdAqMqAUJUBoSoDQIUG  
hKoMCFUZEKoyIFRIIG05BEJVBoSqDAhVGRcQMiBUZUCoyoBQIQGhKgMZyyUQqjlgVGVAqMqAUJUB  
oSoDQIUGhKoMCFUZyFgugVCVAaEqA0JVBoSqDAhVGRcQMiBUZUCoykDGcgmEqgwIVRkQqjlgVGVA  
qMqAUJUBoSoDQIUGMpZLqqQooZqiBpU2kDUQqjLqpyihivHlmgTIVBFvFxzeaoSqoiXayBUZdyc  
qoSqbXwlqglyB0JVBjKWS6qoooZUegFZA6Ego74q1tjiArLmQIUQL/dcrgri5R4IVRk3q2KNLS4Q  
qlkDoSoDGSSLxo8ft99//705mLabP0AuUapw3eTJk+nUqVO6NZgwYQL17Nkz4AfIFhXljd+kH4m  
TZpEvXr1MuO0ikeWWazxNWXXFDp27JhuDTgeffr0MeO1JGO8pk6dSkeOHNGtwbRp06hfv35mvOY4  
x+vgwYO6NZgxYwYNGjTlJfNrPzF8dq/f79uDWbNmKX//POPGadxPLLMYh9fe/bs0a3B3Lladiw  
YenjyjL7+IqLi9OtgR0IVQceeuIHDPJF6vMXnS9NchGjhypt448Hn30UX+csjErXkOHDtVbRx5P  
PPFEhrj41iqzF11vxYv/Q4hUnnnmGX+cLFuuzF50vRWvgQMH6q0jj5YtW/rjZNkCZfai6614/fXX  
X3rryKNNmzb+OGVjVrx69+6tt448OnTokCEuvsNk7CVamaq34sV/kEcqL7zwQmCssjArXj///LPe  
GjAQqkHcfffDGQaPYZkIVcsqVKhAt99+O916661nr12rbK4yW7ldFd9YFYOJMiu3rJyxb0vo/xP  
mdNnnyXGYyR47BiWiVC1rEyZMnTHHXc49nk2G3/n4FgYlOlQtax06dKlI90yEaqWISxZ8uyP1w3K  
RiizItVMY4GOuSorKz4/OJ0hyr2voxytTKnzz5L7M47VbyCxo5hmQhVy4oVK2Zs69TnWWM3KRug  
zFaM8TVexcBhDGVIRWYXMba19/U/VSIVCNUg+Edl/VUTYNkl1YixUcrysJh9ZiRYNklIVfMTZCFVY  
kGUjVCPGeijLy+L0mZFg2QjViLHPleVhiVQgVB1wHIDZCNUnn3xSbx15BMfCjT388MN668jDKR7Z  
CdX77rtPbx15BMfCsGyEKH+5jISCY2FYnKL1xhtv1FtHHsGxcGNXXXWV3jryclpHdkK1cePGeuVl  
wx4Ht3b++efrrQEDoZoF1113HdWpU8ccPDuU2Yuqq1u3Lm3fvl23BjddBPVq1cv4AdnN163ceNG  
3RqwmEqfP3WNMntRdbxu9erVujW45557qEGDBma8lqVHyijqtctWbJEtwZNmjShhg0bmvGanx4p  
s6g6Xsc3eACTZs2aUaNGjcx4ORjHa/r06bo14IMzLKiM+ExSZi+qjmM5ceJE3Ro899xzdMEFFwSM  
KbtXLEeNGqVbAzsQqi7B9FQyMD2VDExPJQPTU8nA9FQyMD2VDExPJQPTU8IAxnIJhKoMCFUZEKoy  
IFRIQKjKgFCVAaEqA0JVBjKWS6pUwCNUJUCoyqh/nr4EAPFyxYUN8AhVCZdfjEeoSoBQIXHz9XiE  
qgQIVRnIWC7Bs/5lQKjKwLP+ZeBZ/zLwrH8ZEKoy8Kx/GRCqMpCxXAKhKgNCVQaEqgwIVRkQqjlg  
VGVAqMqAUJWBjOUSCFUZEKoyIFRIQKjKgFCVAaEqA0JVBoSqDGQsl0CoyoBQIQGhKgNCVQaEqgwI  
VRkQqjlgVGUGy7kEQIUGhKoMCFUZEKoyIFRIQKjKgFCVAaEqAxnLJRCqMiBUZUCoyoBQIQGhKgNC  
VQaEqgwIVRnIWC6BUJUBoSoDQIUGhKoMCFUZEKoyIFRIQKjKQMZYCYsQDAhVGRcQMiBUZUCoyoBQ

IQGhKgNCVQYylksGVGAqMqAUJUBoSoDQlUGhKoMCFUZEKoykLFcAqEqA0JVBoSqDAhVGRcQMiBU  
ZUCoyoBQlYGM5RIIVRkQqjlgVGVAqMqAUJUBoSoDQlUGhKoMZCyXQKjKgFCVAaEqA0JVBoSqDAhV  
GRcQMiBUZSBjuQRCVQaEqgwIVRkQqjlgVGVAqMqAUJUBoSoDGcsIEKoyIFRIQKjKgFCVAaEqA0JV  
BoSqDAhVGchYLoFQlQGhKgNCVQaEqgwIVRkQqjlgVGVAqMpAxnIJhKoMCFUZEKoyIFRIQKjKgFCV  
AaEqA0JVBjKWSyBUZUCoyoBQlQGhKgNCVQaEqgwIVRkQqjKQsVwCoSoDQlUGhKoMCFUZEKoyIFRI  
QKjKgFCVgYzlEghVGRcQMiBUZUCoyoBQlQGhKgNCVQaEqgxkLJdAqMqAUJUBoSoDQlUGhKoMCFUZ  
EKoyIFRI+Lp3706wrO3nn3+mUqVKBQysHj16OLaFqXj98guVLIMmIF4//vijY1uYOb7KlyOfEK8f  
fvjBsW1e2TfffEMDBw6k/v3757kNGDAgV+OhZ8+edG7FcwPi9d133zm2hZnxqlaIWkC8vv32W8e2  
sO70008/Uc3qNQPixb8Pp7YwM1516tQJiFe3bt0c28K6G9qhQYMGAfHq2rWrY1uYafjTGgAAAAAA  
eA6IVAAAAAAA4DkgUgEAAAAAgOfIIFKXzutFq+a1p71bv6J/f6tFu9e/S9Nht6YH76tOzZ++iZKS  
knRLAAAAAAA8oYAkbp503JaOv0p2nfoGO05sI9mTvmIRvcpr4eOnaKN2+Jo2sg7aMvmdbr1GUjq  
cYrdtItOpunlLEhOOEYHD54gF00BAAAAAMIGz6yQkJDgaKd71oVNmzbpd4HwbBnbtm3TS+4IEKk9  
PiLC0/57ivYfTaUrr72O1k2vS6OGfUxtOr2ghOsRst7fzrgb7Ywl5SCtmr+WjrlQntvmfk/Pd/iT  
4jEDCQAAAAA8zMKFC+nUqVN04sSJADt58iQtXbpUtzo99OrVi+68+269ZJKWlka169aj0aNH6xp3  
BCjOPUsq0OrJ59KK8SWI9talU3FNaMPU4rR1/Xe0aOYHdGhIBfnty3K6tZ14Wjl6Dm3ds5aG9PqD  
xq4+qCTzQVo2bx7Fp/D6NNqxdimt35eg6nfQyD/+oGGLFtPyodNop1p/cscC+nvAX9Rv8Ehaf5A3  
OELRsYiFctmUb8+f9Lao0qZzx5CvdX72bHW5QaHaMqAftTnz39pRezR9COeR7fOoxW7uU0qxS1f  
Rhv3q8+kFNq+fRudTN5Hi4fNVL2n0IYpC2j75oX0xx996L8llrJPoY1zxIO/P/vRP8N/oNde7Efx  
quOUI2vp3z59qG9f5Us8UfLe1eozj6j2aXRw61pat918v3/3FiXm+fMAAAAAAE4PixYt0u8ycrpF  
KjP6v/+o2aOP0fHjx42jpszfedDNt3rxZr3VPgEitXr04Df/9XDq1+Tw6sYGttn6tSdvn16DDq2pQ  
yyecROpOev/c4nRO076UmraEmpa/nvrEJtMPXV+naZuOUWpyEn33ys20Li2WOkVH0ZfLIDCd+g4V  
9F1GI47x9imUorTprpW/UqeWfSiZFIFTXwF6rOcSovkfky+mJH0x8xTR4k/IV+hF9WmJ1POmB6n7  
2mSlZ1fS6zOG06kkLVPX/k5F2g1Vbw7Trx2vpQs/nEOUdli6fPlm7UiZR8/lu5FmUxL1uKIKFWj/  
n1K1i+iaWo/S2NQ02jrxfxR42R/pyMkU6t+6Pt346jAldRdQ8xKVAMgW1eXO3+iK4k1pzeYZVLNj  
X1WRRGM+uJXqvzxCuV+lvp8/RZM2YTJjAAAAAJw+vCZSmbi4OCpdpixdetlIdOQIH8yTky5S/xsz  
hmbNm0ILF/5Nc0fVpqQt51LajupKODyGOnAD/dDtVqpRowy9++6Hegs72+j10rfQEHOQcfqLJanN  
eKL9Y3+hl/ovp6NzXqSXv1Uq78S/dPINfZukZXbSmwWupn+USF36zcf00Otv0msdW1Gbd/5UllWJ  
ySL30kKj3VJql/9Wmq70KNEq6uC7miYfPUAvX34Xtfj4XXrnr701u/jKSnFOpa6iV68sjmNjltF  
A3p+Re2ve59m7x1K7741SK1bqkXqKep6wd30Wxy3P0w/XX8/dVuTRksGf06/j99iHJXdvepPeu3F

f4hif6ZrKn5Ge7ipYtxLUTRgwyn66K42NPbEFvr9nXfp1Qe70ILjo+jFp76mQ7odAAAAAM5uXn75  
ZXr77bczWOfOnWnkyJG6Vd7jRZHa5Ysv6PmOnahIq9Y0bNhwXSsjXaQeOXaUJkwZQtVqVaVlq9ZQ  
nwGD6dFnWtMljS6gc6vXoKeefYbue+D+TO7u30nvlC9Kd/+2gWjfRLrv3E40n6tTNIGnZ9vSE9ff  
STNOccUWer2Yj96dqQTs+LeokO8yGrv7ELV9riON2xRHSwe+Sg+34iOp86h5zB002/iohdQi6gaa  
bAjgpdTSdw1NUu8GNG1EI747i04e3U+7Dx2nNEujKg4MbEL1L7uH+m1U32tIM3rw8U7040JWuYuM  
vmYZIvV2+n4rtz5I315zH32zLpV2Tf+Ymj7ZjWL3HqbeLRvSja/wkdQl1KZ8ZfpzeRrFr/2Orjv3  
E2Jte2J0a7rowmvoSzUuEiZ2oicffY46TznBHQIAAAAAAnDa8JFJTUIlovffpy+/7Gos8/WoTR58  
kP7++29jWUK6SL34kkt047ZFNHj4YProi+7076ihNGnmPPqwU0cacn4d+m/0QHqqaZNMROo2eqPM  
TfTNnKU0fvRoWrHHf13mhi9r0C1/ntRLiuObVF9jaNnaf+gBXzNSepUSd62mSePG0OQFa2lr7B5K  
o6O0bvpyOmlIz2O0btpSOMzCWHS1k1dQgf1+yUTx9OYMWNo5Ny1IJqU6kpe2nJjFVksMbU/bRo  
6jLVmjH7OqKk546Fy2mnlZyTKXbJSorli0/TEmntkpk0VvW5ZEssbd20xjqm3J8K00ZPYpGj55N  
+3kTJu0ALZ6yVHnKHkKFkxbTYeM9AAAAAMDpY/78+bRmzRpavXp1gHHdkiVLdKvTQ/Xq1em//8bo  
JT+9fv+drnmGr3kDkOkssplGjasR90/KEFp28+ILD23UGpsDZoyogk998J9dOotd9B1112XyVQG  
2+jllv+j3/bpRUVqylFaOfo7avJsby3kmBTaNncs/Td+EvVoeTXV6TCaCAUnAAAAAMCZj6Uns8JN  
G4uAG6dOnDhCrR7z0cAfStCzzUrRq21L0oFl5almrVJ0WePLdSsnUin+4BFKsE3XlJaWTEf276Pj  
ieYVqCZpdPLwPtq7T9nel+R0TBYAAAAAIAAkZodEvULAAAAAABATHGJVAAAAAAAE4HEKkAAAAA  
AMBzQKQCAAAAAADPAZEKAAAAAA8B0QqAAAAAADwHBCpAAAAAADAc0CkAgAAAAAAzwGRCgAAAA  
AA  
PAdEKgAAAAAA8BwQqaHGS0/lmjQV6O+/icaNI0pi0JWaoUOJBg0iWrdOVwAAAAAAeAel1LygSBEV  
WRXaP//UFS44csTchi1U3Hyzv8++fXWlPkQJs75bN10BAAAAAOAdQqilQDoFCpgCcMgQXeESS1Dm  
BKcjuHaRmj8/0fr1eoWiZEmz3q1I9dlRYgAAAAAcc9eRQEUUwfmTznXelzjnHFHllyhB98ol5On3/  
fqLatf3CkO3Il4mOHSN66im/eC1YkOiZZ4iOHzf7nDXLrl+KMo3f8+n4JUulrr7av83TTxPt22du  
w2zaZK5jc8ISqcWLM8bCNDXVXOckUhcsILrrLqJChcx1desS/fwzUUqKud76rG++lbr9dvM9f6eO  
Hf1t+LVnT6JKlcz1VasS/fILUVKSUR4AAAAAwAVKRQARLNxYfL36KtHMmUTPP28uX3WVuZ6Flx+1  
5LrOnYI27yY6/3xzmCXd5MIet91mLI98sbkNi9ixY806tjIziAYONN+3akW0ciXRxlmsKxcmWjt  
WnM7tyL1vvtMYc3vWYQywSL1jz/8ff32G9H48UQNGpjLd99ttrHWsw/sH1/TyiKU6x57zGxzyy2m  
cO3Vy/S7e3dzPQtZAAAAACXKPUARJx7rim6Lr3UFJ1TphANH24ehbSwjPj26WMus8Dkl6Msanv3  
NsUpry9f3lzPHD5s1rExTz7pXw42PlIZTFan+++5x1y+5BJzmW+mCr4mtXRpc5mFpwUfteU6/j58  
qQC/Z+vXTzdQsljmOhbiGzaYR2ytdsEGAAAAAASKAcpp06ZQvOii/zii0/R85FKi2CR+sYb5nKj

RqagfeUVczkrkXrHHeZ7PuXPR1p5Pb+yJsaabbljWKTyKXc+gspC0rqswBKp1mcvX24uW9jrrff/  
/KNXKiyResEFRAsXEhUtap7qP3TivDSCzflbAAAAAMAlSIOA14webQq+rl11haJtW1Ok8R39Fvny  
mXW//24uW+Ju8WJzmU+f83KFCuYyY7+7n4+K8nWu/L5DB3M9X7/aurV5hJVPozMsXN9+2zQngkUq  
M2MGUUYm/7Mskdqkib8//1EJ0+adZYP7Cdfb2tt4yRS+UgqX5fLR2iLFSMaNcpcz8KV+3z/fXMZ  
AAAAAMAFSIOA1yQn+6/BDLZhw3QjxYUX+utZQLJItLflm634RqiKFf03HMXHm+LOatOjB9FI/mP  
eLLxEVq+BtZiyxb/OiduuslcZ11TavHm/7tvvrKrOMjn3w01BLYlrFPAwaYbaw6J5HasKG53L+/  
v51l/D1XrzbXaWAAAAC4QCkIIlKF6tKI5nWmPEk+3+R08KBeqWHBOX060X//mUKST7PPnm3ejMTX  
dh49ak6iz8bvLXbuNPvldnyUlI9M8rWu/DI87ev27bqhjh9nxAjTnJg3z/SBj2ba4X65Tz4yzP5Z  
8Hfj0/qWD7y93T/ui4+Q8s1gFitWmPV8hNaC1/ODBPgzuA9rFgMAAAAAAJdApAlAAAAAM8BkQoA  
AAAAADwHRCO4O8ETsgAAAIazGojUcME3TPE1oNYToPi6TetGo3Awdy5RrVrm5/MUW2cizZqZ/jdu  
rCsAAAAAcKYCkRourlc8BRWDatWniTfPIH+6SQ62vSHH+Nqf8b/mQQ/9Yq/wxVX6AoAAAAAnKIA  
pOaEAwfMpyvxXf4s6Pbu1Sts8JROGzeajzDdts0/AT+fhuZ11lRWTzXHptmJRep335lmh+/g5+15  
JgD+zLg4/9FXnjWA50xdtcqs4zv1+fP41ZrrlOGjtNzGaRoofjgBzyrAvrCNGWP6w7DPO3aY35E/  
PzY28EEC/NIs/NlbtXktWWPa5s26gYl/k7fIGQWYPXvMbawZEfh782fwdjxTAG9rn/jf+o5sHDuO  
Ac8py/Dn8rb8nbnu0UchUgEAAICzBlhUKTwfqnxU0TKeHP/jj3UDBT9hyj7nKdv//mckRxad1iT7  
lj38sCkWrWWLIUPM5+Tb50otVMh89j+LPhZ0Vj3PqWq9z5+f6MYb/WKTP42y1gXDp/m5nj/D+hx+  
lhYlBh5CFs/nam3L35sf6crrGKu+RQvz9brrTP94XIRLUfpteKouFqTWnLEsXvk7N20a+Bk8T+t5  
5xHNn29+BotQa91zz5mv1pO8HnzQ/K5cx5/L88/ye4hUAAAA4IxH/Y8ORLDAZDFmHS187TVTGPFE  
+0zfVuyyP6WJ5whleEJ/ruNn/VvUrWvWsaBlgq9J5aOPLPZ4+a23zDo+wnjttWZd8+bmUVVrm19+  
Mdvw0V2rbto0f90jxA9/ri57IS1DR/pZIF56aXmMh+dtGAxzXUsgBlrG37KfAtGnne1Rg3zsav8  
dC1+UACvL1zYFPEsUvk9f3eGH/nK14/yU7QYnpPVevb/F1+YdXz01vocjik/GWvRlvOyBK675hrz  
+l4+ElulilKHKQoAAACc8aj/OYEIPt3Nj0Xlo4wsiKwjeZZIZSFoiScLFn0spKzT9AwfLeR2nTqZ  
y8EilU+725ct+lgk1/Hn8mUHTm2susmTdYULrG1YHPopeGvZOk3Psi06vk7We/5UglLftRc9+uv  
pqBmoc1HZvIlMotSXteli26s4G2feoqoXLnAl8ZOlpUfcGBhHc0eOIRXKCwRDZEKAAAAnPGo/9GB  
iMsvN4UQCy1+nv1vv5nLiki1TkmzKLPgo5PvvEM0cqSuUFgi9YUXzOVgkcoC0xJt9ms0//7brOOj  
l/bn6dux6pxEKotLJ6xtWKTyNbPW41H5elgLfhoV1/E6u0i1rhFI+Lty3S23EJUsSfTqQ0QDB5p1  
LMj5O1IPwGLRykdWeR0/kYspXdpdchKpdqx21qI/xrqUACIVAAAAOOMJ+p8fZlt1BI9P8y9bRISn  
jrnMRzYZfowpXx/Jy59+aoo+PqXNbfjaUAtrO74mk4XdiRPmMhvdRzOvuspCzmHMRxz58aeWcOVn  
+2cm4Kw6S6QOH+7czo61nv3lm5WsZ/Jbly3MmmUKY66zhLW1DYtaO5bfbHzan68vPeccc/nKK/1H

RPIRr1Y79pEvD7COTH/+udkms+/lI05wHfvEvv3+u//aVo4XAAAAAM5ogv7nB9nCz6S3rn3kU/p8  
JPXuu4nuvNM80sjw3fAsPi1xxYL0338Dj2Ly8+7LljXX89FXvqnKam/BYU6DD/zXprLdcIP5bH2G  
b2AK3oax6vj5+wzf7OXUzo613poBgG9qYiFsiWm2Cy8kGjDAFLGMVW/dqW9hHe1lY3i99R06djTr  
GP7OfKTVOmrlWpTFMR+F5SPP7MP27YF9WXAse/b0C2c+ivrSS+a+4FcAAAAAnNEE/c8PzlgSAZZ  
6XwLp/XZbZMbQvI5ku3y8jsBAAAAIM+BSAUAAAAAAJ4DIhUAAAAAAHgOiNRlgK8JDb5uNCfwtag8  
DRVfK8szDvCOWgAAAAAAeQBE6tkOz81q3Xhkf6SpFL6DvkEDf1885Rbf4MRPhAqGrwflhwvwwg/4  
6ViW8c1lTo9mBQAAAAAIail13OT1DT7cf25FKj9etXZtsw9raqhmzczlZ57x3+1vwXft33efOXsB  
P52LJ/Vn45kj+Bn8AAAAAADZAJEqPWhRoiJFzHIL+VGgjz1m1vME9Xy0sGpVoquvJvrnH7Oe4XIS  
eTuefJ/57DOiUqX8jxxdscKcnJ6PTFrP2+d5Py+6iKh8eaJatcxpleyn188/3xSBPLcoP+WJHw7A  
4pDp3ZuoXj3z0ab2aaoskcp98ROeWDS6Ydlkc+5XnqOVp4VirKmheP5T+8MGGL60oHJloi+/1BUA  
AAAAADlgUqVYgo/FI7/yvJwffWS+Z9HK85o+/bS5zHOAMjypPy//+acpFHkbXi5TJnA9zxHktG9v  
LvNRSBaU119vLlvPvGeseUeteUL59DuLQ2t+Vhax3l+1ns0SqdaJW2vWNJezw5r3IEWqhf0lrf1x  
pYw1Af8ll/jbslj+4w/dAAAAAAAgA5R6ACIs0cXCkY+M8jWZ1iNOg42fTsVPklqzxlzmazL5xiM+  
+nj//UQxMeYDAFiM8tOSRo82P4Mf9dmvn3lUIT/j+ef9fVpYlpWPsvKTr5ghQ8zP5KOeLBSZGTP8  
27o53c/Xklrt2Vhws1/83i5SGatNsEgdO9asb97cPPrbv795JJkn7R80SDcAAAAAMgcpSSACEuY  
8eM+GRaILBgtAcan+QcPNp/yxKlxOdI8bKj1qNBFi8xXvlyAX19/3bzek0/3W/DjTOvXN0/J89Oo  
3nzT/7kWKljlzNZ/PSTWceC0IKFqbWtG5H6ww/+9myPPEl0bZopUO0ilU/x83oWxSy8s+Oyy8z2  
zz6rKwAAAAAAMkepBiDCEm9LI5rLcXHmo0NZpE6fbtaxaOvenahXL1OkMm+8YW7H14myYOXT5Xy6  
nZ+Nz0dRX3nFbMcUL2625SOQzPff+z/XwhKp776rKxT8bH8+OmsdoWX4FLu1rSVS+Sgrr+fHt9rJ  
7CYuFtSVKpkidfZss46P9nKffF0r97tkiWnMhx+aInvOHhN5717/41V5HQAAAAABANijVAERYgs8S  
qcyvv5p1hQubwpOPivly3/xkCT9+5j8f4eR669nyfBTv6m/PHrOOsa5B5etJq1Qx+7XaWTiJVlan  
fbK25Zu4ihXzb2uJVD4Nz8sNG5rLbuja1dyGvwN/x/z5zcsK+CgrY33Gr1mbPg9r+fLEfjaW16+  
6y6zLQAAAAABANijlAETwTUd8Rz/f3W6H5//kU++tWxN9/DHRhg16hYbnK+3UiahS7+w4yOUPJUT  
H2W1w8L2m2+l2rQxBTAfmeVLBwYONLdhWOjyqfOhQ81lO337mtuygOVtBwwwj8qyDwwfXWWh+t57  
5rJb+Ohr587md+SpqFiQWvD3YONLG5ht28w4cFuOy/z5Zn1mR2sBAAAAAGxApAlAAAAAAM8BkQoA  
AAAAADwHRCOAAAAAAPAcEKkAAAAAAMBzKQKQAAAAAADPAZEKAAAAAAA8B0QqAAAAAADwHBCpAA  
AA  
AADAc0CkAgAAAAAAZwGRCgAAAAAAPAdEKgAAAAAA8BwQqS44pMoHqnRR5XNVJqoCAABnAgmqvK8K  
5y8u/6oCMidJFXu8/lYFZE6KKvZ4DVAfZE6aKvZ4/aUKyBylVBdsVMVnKy+qAgAAZwJHVLHnr6aq

gMyJV8Uer7tVAZmTqlo9XreoAjlnVRV7vK5VBWQORKoLIFIBAGcqEKkyIFJIQKTKgEiVAZHqgm2q  
2AfVK6qAzEILS6MFCxbQ4sWLafny5boWZMX8+fONeC1btKzXgKyw4rV06VJdAzLjpCr2/PWQKiBz  
klWxx+teVUDW2ON1qyoga+zxuk4VkDkQqdlw2WWXUeUbKqcPKKN866Mdm3YY61mQZcUq9R9pJHHV  
VVdR5coqXj4VJ23lypWjDRs2GOuzi1ekcf3111OVKIUC4IW+fHlavXq1sR7xCuTWW2+IqIWrbSsr  
QoUKtGTJEmM94hXIXfdRVXPV/Gyl2E+mjd1nrEe8QqkSZMmVK1+NXu0yDfGRzPGzjDWI16BNGvW  
jKrVDorXZB9NHG7et4F4BfL0009T9ZrV7dGiQFIRNHrgaGM94pURiNQgrEHY4osv+v8jrG0NJ7NE  
fReVvm7oiFH07azZ1GrseHp05Eh6fuJk+mnpmvplXUrqNHg4FflnGH09ex79qkTar+vX0w8rV9Gn  
c+fR2/Pm06vzFIOb8ROppdr2rRlZqOecuTRu9hzavMoUKGcCVrxe+01f7yCLCrKHy8++hXJWPF6  
6623AmJkN3u8+IhhJGPF69133w2lkd3s8ZozZ47RPIKx4vXRRx/5Y1RSma1EDfPHa+bMmUb7SMWK  
12effeaPVxEzTlaJGuOP19SpU432kYoVry+//Nlfr2gzTlaJmuyP16Rjk4z2kYoVr6+//tofr3xm  
nKzCitVaN27cOKM98AOR6kCRIkX8A4otSKTykVSuL1CkMJ07ajTVVNtU27efKk2YShVGjacKw8dS  
reEjqrqtVTrhx/p6lat6d6WremBjh3pvuda0o333kv3tmhB1377PVVPSKBaans27sfs6wAN6Hvm  
3PFXsmTJwHhIY2PHjtVbRiZly5Z1jEtmNnz4cL1IZFKpUiXHuGRmf/8d2XdjV6tWLTAmQSLVN1SZ  
bX3fvn31lpFJ7dq1A+IRLFJ9/ymzrf/111/1lpFJw4YNA+IRLFJ9k5TZ1n///fd6y8jkkksuCYhH  
sEj1zVRmW//VV1/pLQEDkRrEww8/HDBgDAsWqd2VqfrqzVtR5cXLqNLs+VRhyEg6d8Fiqq7ssee  
o+dbtqS2TzxJTygxWnnDRqpx8iTVioujetu3U51Nm6jelq1UZ+dOqj9rNt3y2ed0bfdvqdGYsVQj  
KYlqqr++OvTtr3yNk899VTGeGVh1lGvSKVVq1YZYuLGUIJSdA+RRUf1h51TPLKzBPXHXyTieEYj  
G5HKduLECd1DZPHee+9liEV2IpXt8OHDuofl4tNPP80Qi+xEKtu+fft0D5FFwBFUy7IRqWw7ITYA  
JhCpNIJTU+n+++/PMGAYiNSuPip299NU7vc/qcKIMVR5xWrjSGgNZVd9053atGIDbdu2NaydEiVV  
IBC1jpYG2yXjx9NzStS2a9eOmj/+ODVcsMAQqU0HDabYLZtp48aN3jB2a5Ks7ebke88FROOkVsr  
67P1Yit7IT95plstsI34D390dPkq+UQk6xMxWtdyjbT7rsV+b0mWeqqT/m7GWrKi27tMxRvFYn  
rrb1pMsBZU6fe5bYli1bqEOHDhnlwuRumLFCsc+z2bjeL388ssZYuFGpPI10E59ns22efNmevPN  
NzPEwo1I5Rtpnfo8m43j9f7772eIhRuRGumX4diBSLUxefLkDIPFMP6Pz166+Kj0W+9T9fiT6afo  
jdP0ym5Tg5JFqmVtW7akyjNnBbSz28XjxIOL5i2Mtu2VWG3c4yeqoURq8edzdgQpzy0vS0tITp95  
JltelueVOX3mmWx5WV5V5vSZZ7u5EKkwm7kQqTCbuRCpMJU5EKlswASRsDFx4kTHwZJBpH7ho5Id  
X6HKy1dTtQMH0wUni9SbPvggUKS2ciFSW5gitd0zz9BFfw0wRGqJ9g5HRMJtUcrysRRR5vS5Z7LI  
ZXlBmdNnsmWI+V1ZU6febYbRKRmIFJlBpEqS2xEaqRfEhcMIhHEk08+GTBgDAsWqV/7KKb25VS+  
/2Cqsm4DlftzIFWcMIWqq+0v7N2HWrdunS5S+XR/liJ17Dhq8WxzaqvaPtS6DdXatJlqpKZSkcee  
yOiHFywwC46kygqOpMoKjqSaBSI1a4NIIRIEqsxcHEndtm2bUgeAgUgNomnTphkGTAArqu/ur9Kw

EVUePZYqTp5OFdVr+b8GUXn1/n9Pt6BHn3iS7u/Yia757HOqO306XTFhAl03aBDd0rcvXfnd93TT  
L7/R1X36UI0NG6nWp12o1qrVVO3QYeM6VRavH/zRz/DHc5gzaqTzTJNnMsYrC4vyRdhfiUHxavt4  
2wwwxycrS4xUp900FxeVFFrap4FxYerxO6Q4iDMepzblQqZF+1Objjz/2x8kyFyl1Um9k7Nq1a4ZY  
uBGpJ0+e1D1EFj/88EOGWODGKRkQqQ5kmJlE5HqUwm+dKs2VO/oMaqZclqqnzpJtY4fp3rHT1K9  
5FSqdfQE1RowkC4dNpluHvkfXTx6DF3492Bq+Ouv1Ojvv6nmxs1U89QpZSep9rETVP/wEaq/YCHd  
2PV72rFxo/bG+zRo0CAwXtnYwoUL9ZaRyYUXXugYl8xs9uzZesvlpHHjxo5xycymTJmit4xM+IEa  
ATHJ5kjqmDFj9JaRCT9Qwx6P7ERqpE8Jxw/UsMcjO5E6aNAgvWVks899wTElzuRGulTwgUDkRqE  
NfnuTz/95B84QSLVPpn/rh07aPv27bRk81aav2ULLd+8hbZv20GxO3bS1Blz6JLX3qAvy1bQrrg9  
FLtrN21T9RtUu7XbttGq7XG0ZMtWw9ZujaXtW7dS/JEjxuefafTq1csfryCzT7a+f/9+uvUVkw4nl  
HiO72eO1d+9evUVkM2DAglAY2c0er927d+stlpshQ4b4YxQkUu2T+e/atUtvEdmw8EyPV5BltU/m  
Hxsbq7elbP777z9/vlJEqn0yf/6/ERCNHZ/eH68gkWqfzJ9nnACBQKRmw+OPP07XtbgufUBxKftX  
WTq456Cx3hK1wITnTQ0+MnHOOefQnj17jPWIVyB809yNN94YEK/SpUunn+5BvALh67xvuummgHiV  
KlUqPbkjXoHwIFS3NL3Flr18VGJSCVq3dJ2xHvEK5KWXXqJb77/VFi0fFZ9RnFbOW2msR7wCeeON  
N+i2e26zRctHxeYVo8UzzCcLlI6BvPPOO3T7nfbouWjokuK0tyJc431iFdGIFJdsEkV+6B6SRWQ  
OTt27AgQEe3bt9drgBMs4O3xat68uV4DnDh48GBAvPgPSZA5R1VRkUovTVUBmXNSFXu87IYFZE6S  
KvZ43aIKyJw0VezxulYVkdKqQs7YqIp9UL2oSqHJVH9BvaXEHZsU/tur/4ED9PHOnTTOA5cL8Cke  
u4jgBxWAZOFT1PZ4QaRmzQE11u3xgkjNmiOqqEilF4jUrllXr4viNSsSVTFHi+11KxJvcUeL4jU  
rIFldcHpEKnHU1LIN2uWYVKe2LSJfHPnGvbitm10PFx9CGbMMcwcQKTKgEiVAZEqAyJVBkSqDihU  
GRCpMiBSXeB1kVp/+XLyzZtHr9guUs9pX6EAILUGRkoMiFQZjilVI75liqNIRbwyxVGklI6Z4ihS  
Ea9MgUh1wcZ1SqTa/IN8sfXpEalJaWn04549VH7xYqM+esEC6rh1K+1PTjbWH1XbXLhixkUVYIU  
3+zZVGrRivLnnEn51DIbvX980LzJ63QBkSoDIIUGRKqMI4eVSLXFq+mtOJKaFfHxSqTa4nX3dTis  
mhWJiUqk2u1y+U4kpoVqXym0xavay/EkdSsgEh1wcaNqSL1xdMjUp/ZvNkQoL/s20dH1cCeePSO  
sXzjmjV0Qi3zH18sZGsuXWql1Oaq/SElYNeeOpXe1+6kJDql2p5OIFJlQKTKgEiVceRIkEhtCpGa  
FRIE6t0QqVmRQaTeApGaFRIE6rUQqVkBkeqCcljU7eqHX3fZMvMI6cyZ5Js+3bzOdPZso80eJT4t  
6uh2z2/daiwf4R+Bra/TDUSqDihUGRCpMiBSZUCkyoBIIQGRKgMi1QXhEKkbTp2iGnyEdO5c487/  
riBIPbOBSJUBkSoDIIUGRkoMiFQZEKkyIFJdEA6Ryqfor1y92hCfn+inwixTyfPyVavoqU2bjNP9  
FsEilS8NsPpKDsPkwBCpMiBSZUCkyoBIIQGRKgMiVQZEggyIVBecNpHKp/WVWSw4cYKqLlIi3hjF  
onPOHOPo6qSjR3ULE+OaVNWmvRapLHAr6+tUebt/DxOy6k8XEKkyIFJlQKTKgEiVAZEqAyJVBkSq  
DihUF5wOkcrHRVefPGmYnXg1oLcmJND6U6eMV/sRVlStqn6dWm+/TpXv/N+o6viyAe7jdAKRKgMi



VQZEggyIVBkQqTlgUmVApMqASHXB6RCpZxMQqTlgUmVApMqASJUBkSoDIIUGRkoMiFQXQKTKgEiV  
AZEgAyJVBkSqDIhUGRCpMiBSZUCkugAiVQZEggyIVBkQqTlgUmVApMqASJUBkSoDitUFEKkyIFJI  
QKTKgEiVAZEgAyJVBkSqDIhUGRCpLoBIIQGRKgMiVQZEggyIVBkQqTlgUmVApMqASHUBRkoMiFQZ  
EKkyIFJIQKTKgEiVAZEgAyJVBkSqCyBSZUCkyoBIIQGRKgMiVQZEggyIVBkQqTlgUI0AkSoDIIUG  
RkoMiFQZEKkyIFJIQKTKgEiVAZHqAohUGRCpMiBSZUCkyoBIIQGRKgMiVQZEggyIVBdApMqASJUB  
kSoDIIUGRkoMiFQZEKkyIFJIKS6ACJVBkSqDIhUGRCpMiBSZUCkyoBIIQGRKgMi1QUQqTlgUmVA  
pMqASJUBkSoDIIUGRkoMiFQZEKkugEiVAZEgAyJVBkSqDIhUGRCpMiBSZUCkyoBldQFEggyIVBkQ  
qTlgUmVApMqASJUBkSoDIIUGRkoLIFJIQKTKgEiVAZEgAyJVBkSqDIhUGRCpMiBSXQCRKgMiVQZE  
ggyIVBkQqTlgUmVApMqASJUBkeoCiFQZEKkyIFJIQKTKgEiVAZEgAyJVBkSqDIhUF0CkyoBIIQGR  
KgMiVQZEggyIVBkQqTlgUmVApLoAIIUGRkoMiFQZEKkyIFJIQKTKgEiVAZEgAyLVBRCpMiBSZUCk  
yoBIIQGRKgMiVQZEggyIVBkQqS6ASJUBkSoDIIUGRkoMiFQZEKkyIFJIQKTKgEh1AUSqDIhUGRCp  
MiBSZUCkyoBIIQGRKgMiVQZEgsgUmVApMqASJUBkSoDIIvGojJ7vO64915zBXAEIIUGRkoMiFQX  
QKTKgEiVAZEgAyJVBkRq9sTu2EFPNL6S3ilVgX4pUYZ+LICUfo4uQT0KFqEPY4rTczXqUt8+fXVr  
YAcIVQZEggyIVBdApMqASJUBkSoDIIUGRGr2fPlcaz58SIQohqhAAdpRIIbWRRei1Kj8RNEFIUXT  
7wWK0/G0VL0FsIBIIQGRKgMi1QUQqTlgUmVApMqASJUBkZo9XxctbQjRRcXPoYfuaUJpt29PrTp0  
oPuffJJ+rVaH0mJi6ESBwjRg1Ci9BbCASJUBkSoDitUFEKkyIFJIQKTKgEiVAZGaNcs3b6alvgJE  
BQvQq9feSO3at6M2bdoY1lbZKw8+TLFFSiCaErE/v/OB3gpYQKTKgEiVAZHqAohUGRCpMiBSZUCk  
yoBlzZrhU6bQUV9+is9fkF646256tkULekb9Bi174sFm9F25apQaE0O9mrfUWwELiFQZEKkyIFJd  
AJEqAyJVBkSqDIhUGRCpWTNhzIza5ytiW4oUpb3FitORYiXocNHi6Xa8aDFaWLykccq/T9vn9VbA  
AiJVBkSqDIhUF0CkyoBIIQGRKgMiVQZEatYs37iBluQvQu/deDMdKFGCUgoVomQISC3jyWA+v/U2  
ml6uAg186129FbCASJUBkSoDitUFEKkyIFJIQKTKgEiVAZGaNbGHDIL/qELU7MILaU+xYhIFanQ0  
vdLwAupRsRL16vqN3gpYQKTKgEiVAZHqAohUGRCpMiBSZUCkyoBlzZokZb/Ik0yv3XZbpdSv7n1  
NupfuQoN+qOPuRFIByJVBkSqDIhUF0CkyoBIIQGRKgMiVQZEavb0KICAPr72OjpcpliDSC1IP195  
FY04twoN+auf3gJYQKTKgEiVAZHqAohUGRCpMiBSZUCkyoBlzZ5f8xekry+5jFKio4IYmPKrZUqk  
jqhfn0ZWqkZD+w/QWwALiFQZEKkyIFJdAJEqAyJVBkSqDIhUGRCpmXPs5Enq9+pbNCqmCH1Vsy71  
uPBi6nbZZfRV48bp1u3SxjS4SgOadG5V+uqOe2jdyIV6a8BApMqASJUBkeoCiFQZEKkyIFJIQKTK  
gEjNmn+/odm+aKpT/2G9CxP4N+2bfpk/myt27ejN6+/ifbGRFP3Vu3pWHy83hlwEKkyIFJIKS6  
YOMxJVJ/VQOqj7LeSqTOh0jNCohUGbtPKpFqjS9IzWdBpGbfGQQIUn/xx+vx6RCpWXEkSYIUW7ya  
ToFiDaavLx/9cMwV1LpD+wwitU27tvRE04cpOaYQ9R+A0/3BQKTKSE1TitX2e7x2DERqVkCkumCj

Kmo4pZcXVQGZA5EqY7cq1tji0lwVkDkHVLHH63FVQOYcUcUer6aqAD9Hk5NooK8A9SxXhzo93lxa  
PvcctWjdm pore65IS2rz1NPuO5jii9YIH7v+YveClhApMplVcX6LXK5VhWQORCpLoBIIQGRKgMi  
VQZEgqyl1KwZ+N9/tM8XTfOjy9L8QmVoe+kytLxsOVpchrxtPKcMrS9amoYXrEQpMYWo52NP6a2A  
BUSqDIhUGRCpWZCWlma87ITFl+YfVK+rAjJixWv//v0BSatTp05GPQjEitdRvezjq7UqlCNWvE6q  
Yo/X06qAjFjxSlbFHq9HVAf+Bo4eTcd9BWhVdCkaGn0uUaEYJUhN4ymo1qr63tGVDZH6x6NP6q2A  
Nb4Ye76//fbbdS2wExAv2+/xBIVA5kCkZkOVKIWoYIOC6QPKKJ/6aNVS3OHpRM2aNalgQRUvW9Ji  
W7x4sW4B7NSrV48KVgsaXz18NG/aPN0C2LnggguoYKWgeP3uo+njpusWwE7jxo0pumy0PVrkG+Cj  
8cPG6xaAnzg1zVeQNseUosnRZWhLdHFK1QKVbVDBijQyujlFlymX/HEqQBuvPFGio5W4yso3//z  
zz+6BbBzxx13UHShoN/jFB/9+dOfugUIBil1COuvnXvvvdf/o6utB5MuUd9Fpa8bM2aM0T5SseLV  
pEkTf7yCLCrKH68RI0YY7SMVK14PP/ywP0YVldIKVB9/viYMGWK0j1SseD366KP+eJUx42SVqIH+  
eA0aNmhoH6lY8XryySf98SppxskqUcP88frrr7+M9pGKFa/rSpelCTGlaU3B0vS3EqUUE01pyhYW  
PlmFihLk4uUp875i9AHXboY7bNExdWwmTN1hQusbbZv1xWZwH/sP/880Usv6YrTixWv5557zhg/  
2dmvv/5qtI9UrHjDXjpccmnzF5mKdPrevToYbQHfBSHeDBYhdWwSLV960ya52yv//+W28ZmeTL  
ly8wXtnYn39G9l+NMTExgfEKEqk8g4Q9Xr/99pveMjlpWrRoYLyCRCofGbTH68cff9RbRialS5cO  
jFeQSPUNVWaLV7du3fSWkUn58uXJp+IVXeYcerxkdfuQBWaE12S0qJj6LUC1ahzsepUrmwFow3H  
6/PPP9dbhhDvr2GxsboiEwYO9LcNE3x2UZLv33//fb1IZFKrVq3AmASL1JnKbOvffPNNvSVglFKD  
eOyxwIgjGHZiFS2SOXZZ5/NEAs3FqnwTWQZ4pGNSGVLsUnRPUQWPCdxcCyyE6lsCQkJuoflqv+D  
C45FdiKV7cSJE7qHyOLDDz/MEIvyUfmoelR+qhVVGcooiwlaz3b48GGizZuJLrqI6OKLiXju1Dvv  
JFq+nK95lqpWjWjFCv0pijVriJ54guiuu4jWrSNq2ZIPdxNNm2auV30atn8/0RdfEPHNR82aEc2Y  
Ya5nxo9nhehvq/6vylbUhpguXboY319q+/bt0z1EFt9++23GeGQjUtI27dqlewCRqxYcOH78uHGN  
TfCAcSNSJ06cSOPGjTv7zVYmq3Jn9zvJd7uKwR0Cu8pn68VWVitz+syzxCZNmkt3339/hrHjRqTy  
ZSVOFz7Nxf66KGHMsTCjUgdNWqUY59ns3EO4gcbBMfCjUgdPny4Y59nIY1RZisTVXI2gPoj+1YV  
A6c8lZld7aPBjWYTrVRiU8XOMBam/Dplir9u/nzzP5Zhw4iioszHrd50E9F55/nb/KKntLKWq1cn  
uvxyoksv5VNUZI3//mabyZOJ6tTxt23ThpbIZV74T5mtTFCl1T+tyHeLQ0yyMhWvv4/8betJl63K  
xipz+uyzwMarPyo6dOigdpWKgd1ciNRIP9toR410YDFFJZjgwWKYC5EaMZaXpbUyp888282FSIXZ  
zIvIhdnMhUiNGAthoZU7/YLxqaeldqplxpqjkZqYaApPXIZ/OBnwUdWCBc0665pNa5vZs81lpm5d  
s659e12h4lcJWG0VjzzyiHqrvMkry8vSTZnTZ57t5kKksgETRMLGhAkTHAcLRKq2KGV5Wdooc/rc  
s90gUmUGkSoziFS/hbAEiFQ7Vh2L1JMn/cupqbqBolAhsy5YpFpCl2nQwKxr21ZXKPioqtVW0axZ  
M/VWeZNXlpfla2VOn3m2WzYi1breF5ggEkG0aNEiYMAY5kKkIhR4uy3ksqCSsFTBcl3VMVAYkdU

vJzKC8qKKXP67LPEnKZrcSNSnfqKBHOMlwuRWrx4ccf+znbjm/KCY+FGpEZEVAoqCyoxCSpeTjkq  
K1P5K39afiVSYwMEYzpWHYtUvja6VClzmS8FYPgO/syOpNqvMXUhUvmegDzbd/mVBZWYxJzFq3ia  
8jG4/KCcssDKnzz5LrJD6YyT4t+bmSOqhQ4fSZwaldMyRDtJxnEopC5EacX/1BP1unnzQNTWNC4vy  
RfZfia1atcoQk6xEaqT/Vd2xY0d/nCzLQqRGerxef/11f5wsy0KkWvGKmBvzgvLX+6+/74+TC7Py  
16IDpyht5coAwZiOVccilYUGT1vFy0WKEN16K1GNGv42UpHKU9JZbfkSA77RKi8Jitfn732uPjow  
JnV9F9BTvqepk+8letbXgi72/S9Dm6N7juoelovu3btuilUbkbrTfkQ9wlEjHVhYf7nwY90CBk0W  
lpWP9GzdutXYLLhCYoD4pWFFShQgNavX6+3jEwC5uBly0KkcrxW8n+GEUzTpk0D45WFSM2fP3/E  
PzgiwyngLEQqTx83d+5cvWVkJCnDbGon76dP3gCL67v1EjovPPN5ctLrzQrFu92lzm/1f4RhiV  
J+mee0whyjdRqf7SRSoLUra9e81l5tFHia64wrzb34Kvce3alejGG83ZAvbs0StOH9bZxnKFytGd  
le+msfkm08B8Qw01MTNqPvXM9zu1qNaKqkRXpXwF8xuX0UybdUfGQHjKBuRONr0aL0IYCBSM4Hv  
Dk4fOEEi1T6ZpZDJ9HpeZdbRGjZgMm3aNH+MgkSqfTJ/YDJnzhx/vIJEqn0yf2CycOFCf7yCRKp9  
Mn+cUjRZtmyZP15ZWkr9uli3sKi84ALzKKol1r7/3rxzX/0Batz5fwbyXZfvab1vE/2bfyT1z/c3  
HYs6SSnq58c2Jv9kmpL/Lh2OOkbvPfkB3iKy2bBhg38cBYtU22T+SUIJegtggayeDe+++y498/Ez  
/gGlsr0x9ejEYXNeQST5QD744ANq3ry5/weprHbt2nTkyBFjPeIVyMcff0wt32lpG10+qjmtJh3c  
ddBYj3gFwhOpt36rtS1aPqo+uzrt3WYegUK8AunatSu1faOtLVo+qrKgCu3caJ5ORLwC+eabb6h9  
+/YB+evcc8+lbdU2GetzFK+JE/2n7i3jI7Dqs1SHutGZxduN3qNtvp20oMASGpJvpKEkrVtjtpO  
Q/OPphNRp2juefNp4+6NeivQs2dP6vhCx/Tflpdy68rR2oVrjfX4PWZEDSmQHRtVsQ+qF1UBmbN9  
+/b0BM/GE9iDzNmtijW2uDRXBWTOAVXs8XpcFZA5R1Sxx6upKiBz4uPjA/LX3XffrdcAi099n9G2  
fDtpcP7hdDzfkEmlqiVSd0XtodX519PUfLNpy7nbaPZy27RagFJVsx6LXK5VBWQORKoLIFJlQKTK  
gEiVAZEqAyJVBkRq9rxX9QPa5ztE031zaGC1QfTlpV/RJ9d8SI9c9iX9VuN32pZ/J43KN4GW11hF  
K7dE9jX1wUCkyoBldQFEqgyIVBkQqTlgUmVApMqASM2eL9p8TfPyLaJHHnqUWndoTW3btaV2bdsZ  
NwnxcrNHH6F/Co2kYY2HU5lqwA9EqgyIVBdApMqASJUBkSrDUaTiUq5McRSpiFemQKRmT9f239Ci  
Ukup5dOtDGHapk2bAHu207M0psQEGnHZCEpRogz4cRSp+D1mCkSqCzauUyLVlrRebA2RmhUQqTJ2  
71li1Rav5g9DpGbFgX1KpNri9fh9OJKaFUcOK5Fqi1fTW3EkNSsgUrPm7z5/U/eyP900otOpw6Md  
qNXzrahllh5bp1ur5ltTkqSY0udhUGl9kEg3/bbR8VoSzGI6FfXxdehyGOpgYFRKoLnm4MEqkvQqRm  
BUSqjN27gORqc4jUrDhwIEikPg6RmhU8s4Y9XjzvlMgciNSs2bZtK3343Jc0ttw4mlhiCv1baigN  
Kflvuv1TcgitjdIAz1/QiYbWG0Hb9uzAXes2MojUayFSswli1QUQqTlgUmVApMqASJUBkSoDijV7  
vuzUnf6o1Zt2INIJKUqWEnYbG+B/dT6jrY07uIJIz2ZQFApMqASHUBRkoMiFQZEKkyIFJlQKTK  
gEjNnq4vdKfvGn5P6ypsoBQlUq3pp9jSIC0puYyev6cjjb14vHENJvADkSoDitUFEKkyIFJlQKTK

gEiVAZEqAyl1ezo/+jZ9dtIntLzqSkeROrfyGr+QHMa1fA/OpZ0TG8FGIhUGRCpLoBIIQGRKgMi  
VQZEggyIVBkQqdnz8nWv0Gv1X6f5ZRZmEKmsKqaWmU53X303zag7m3Yd3mVuBAwgUmVApLoAIIUG  
RKOmiFQZEKkyIFJIQKRmzxdX9qA/6/xFKyqtcjySuqD8Yvrwqo9pXe1NtHX/dr0VYCBSZUCkugAi  
VQZEggyIVBkQqTlgUmVApGbPx2U/N9TDiqKraFCdf2hQ7cHpNvC8v2luyfl0MI8iram6jhavX6y3  
AgxEggyIVBdApMqASJUBkSoDIIUGRKOmiNTs+bjS57Qraje1rt3BeNpUm7Zt0u3Zds/SQCVWk6PS  
aGX1NbR041K9FWAgUmVApLoAIIUGRKOmiFQZEKkyIFJIQKRmz/t3f2Koh0mFp1Krjv6nTrVt05ae  
bvE0Tas8k9KiiMY2HE/74/frrQADkSoDItUFEKkyIFJIQKTKgEiVAZEqAyl1e37v/gcd88UbCuLX  
yr3o5bteoVcfeo1eu/d16IG7Jx2PMtd988C3eguTxNhEWnPjGprjm0PL6y+n5APJes2ZxTTfNMNS  
jqToGvdApMqASHUBRKOmiFQZEKkyIFJIQKTKgEjNnqP7j9IQ3whTQUQp4aVeE6KSKYWXtR33nal/  
/unDzdPZ+clOmuebR4vKLKKkXUIqQ73iDGOWb5ZhKUchUvMaHk4gGyBSZUCkyoBIIQGRKgMiVQZE  
avakpqXS3PnzaMiXQ2ncq+PpLt899HC+R+jBfA9R2xrtaMzXE2noqKG6tcmxGceMo6gsUheWWEgH  
Bx+k10OmSo1fGk/7eu+j3d/upn1/7KP4ZfGFPXN4zGGK6xZHB/ofofMbTxltdnffTXFfxlHKMVMk  
nlhwwljmPhk+QsvbcJ9Wm5NrT9K+3/ZR3BdxtOf7PXRkwhGj3mJfL7VObcN+Hp91nPZ8u4fSksyn  
ZZ1ae4r29txLe3/dS6knUyFSTyMQqS6ASJUBkSoDIIUGRKOmiFQZEKkyEhMTA+J1yy236DWB7Hhj  
B831zTVEKtuKC1dQyuEUWnfXOuP0//z882l+jDL1yssbmm4wttv01KZOYbu02Ij3c53d9JM30w6  
NOwQpaWk0ebmm42+F5ZcaGzD4pS3WXvHWmP50D+HDFGZ/hkFzM9Yf+96Yz2z7LxljZLayylhcUW  
0oLCCygtMY3iusaZfuebZ2y3qOwiox1E6ukBlitUFEKkyIFJIQKTKgEiVAZEqAyJVhluRyuz8aKch  
+Jacu8RYju0cm3596ql1p4y6k6t00tJaS412e77bQ5tbbTZE4fzo+bS1zVba32c/nVx9kmb7ZtOm  
xzdRaklqLau7zOiD2yRsTKCNj22keVHzaE+PPUaf217aRls7bKXErYnGctzXccbnch8Wlklglbrz  
/Z108O+DxIFVFr4sUI+MNY+8sk+8LUTq6QEi1QUQqTlgUmVApMqASJUBkSoDIIWGRKTGvhNriM/F  
lcy5U1ddscoQhpuf2WwsW6y7e51Rv+HBDbsI7Rbj/YqLVui1RMkHk43IBQUXUPK+ZEM0HhI9xOh7  
1+e7aOWIK2l+wfnGkVomNT7VEMRLqy+lhcUXGq/cp5NI3dp+q65RuabfAaNPPgKbTiquST2dQKS6  
ACJVBkSqDIhUGRCpMiBSZUCKysiNSN3w0AZjed2d69Kv/0w9kUqrLjFFKx/93NLGFKksPO1sf2W7  
IU75cgAWmCxEx+YYsPqrKr+vv85/KX3HBCKOQ8nWrqadS6ejEo8ayk0jd9sI2XUPG5QTGaf58SvDq  
O/n5elneDiL19ACR6gKIVBkQqTlgUmVApMqASJUBkSojNyKVT78vKr3IEjsbm22kv/vpXX3rjNE  
IF8SkLQ3iTY9a16TGixSD48+TAuKljD649P7DF8OYJ3GPz7nuFHHLD9vuVG/+trVtPu73bS0inlt  
KwtNCyeRyjdgrbx4pfeZq65cRXt+2EOLyy82tuVrYiFS8x6IVBdApMqASJUBkSoDIIUGRKOmiFQZ  
uRGpDN+cFPtuLC2tupQWFFhgXfVkd++TeWCvNj2pReolgsI1LSGNIIReYhzp5JubmKOTjhrCc0kl  
85pXCz7tv/aWtbQgZoFxfhJWvad3WcRtteW4LHZ181GhjXQdrF6kMH+Hd3GizLSi0gJbVWkYnl5+k

LS230JZntxhHZaVApMqASHUBRkoMiFQZEKkyIFJIQKTKgEiVIRGpAWgRKiln27ghr/p1ACJVBkSq  
CyBSZUCkyoBIIQGRKgMiVQZEqowci9QIBSJBkSqCyBSZUCkyoBIIQGRKgMiVQZEqgyIVBkQqTlg  
UI0AkSoDIIUGRkoMiFQZEKkyIFJIQKTKgEiVAZH6f/buAk6K8o/j+NlpoWKCrdqJiQUWKil2FogS  
CordLYqigqJYqBhY2N2N0khlG3R3N7//85uZ5WZ35/b2OfbxL7ef9/P6eXg7O7M799ze7743O5MB  
mlQ7NKL2aFLt0KTaoUm1Q5NqhybVDk2qHZrUDNCK2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpGaA  
JtUOTaodmlQ7NKL2aFLt0KTaoUm1Q5NqhyY1AzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qRmg  
SbVDk2qHJtUOTaodmlQ7NKL2aFLt0KTaoUnNAE2qHZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5NagZo  
Uu3QpNqhSbVDk2qHJtUOTaodmlQ7NKL2aFlzQJNqhybVDk2qHZpUOzSpdmhS7dCk2qFJtUOTmgGa  
VDs0qXZoUu3QpNqhSbVDk2qHJtUOTaodmtQM0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhS7dCkZoAm  
1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk2qHJjUDNKL2aFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpGaBJ  
tUOTaocm1Q5Nqh2aVDs0qXZoUu3QpNqhSc0ATaodmlQ7NKL2aFLt0KTaoUm1Q5NqhybVDk1qBmhS  
7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUjNAK2qHJtUOTaodmlQ7NKL2aFLt0KTaoUm1Q5OaAZpU  
OzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2a1AzQpNqhSbVDk2qHJtUOTaodmlQ7NKL2aFLt0KRmgCbV  
Dk2qHZpUOzSpdmhS7dCk2qFJtUOTaocmNQM0qXZSmtS2NKnP0KTaoUm1k9KkNqVJTYcm1Q5Nqp2U  
JvVomtR0aFlzQJNqZ+oa06T+ZvbVH361m0STmg5Nqh2aVDtLNpgm9Vf/e1Gr2Via1HRoUu3QpNrZ  
aEb4+7HBSJrUdGhSM0CTameqGWZPbR7tzED+aFLt0KTaWWKGfh/GRzMzkD+aVDs0qXa8JjU0GpiB  
/NGkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk2qHJjWNTZs2eR/nzp2b8E142223eZ9Hovj+WmxG  
bFP8W9A09WYgVXx/Jf9QbNu2rfd5Jlrvr3Xr1iXsrxYtWnifR6L4/ILh78cLzECqhP0Vml9NmzYN  
Pouw/PbXqaeGnwWYfl9P55gBvJHk5oPPbhZbbPNnt43XrFixRK+EX/77TdZtmyZtwzy9te2224r  
sR3M/jlj/k0YeyQm33/5PfsrJP6CteOOO0bOr6+++or9FRLfXzVq1ljcXx9//LEsXbrUWwZ5+6tm  
zZoSK530/fh6TN555R32VxLdZ/vss0/k/HrttdfYX0n0Nf/ggw+O3F8vvvgi+yvJhg0b5Mgjj/T3  
V/j78buYPN3paV7v80GTmiT+4r7ffvslfNOFK/wN2bNnT2/5XBXfXwceeGDePtrNVHg8byq4rUeP  
Ht7yuSq+v+lv7gVVt27dvOVzVXx/HXbYYZH7J7m6dOniLZ+r4vvriCOOyNsvlU2Fx8emgts6derk  
LZ+r4vvr6KOPzttfaeree+/1ls9V8f117LHHRu6f5Lrjjju85XNVfH+dcMIJefuluKnw6GsquO2m  
m27ylkcemtQk8ePdkn8zTFddu3YN7p17Fi5c6O2DhP2VpknVevTRR4N7557Fixd7+8Bmfj344IOb  
X+xyjZ4+qUSIElb76+67787Z/aXpVdmyZRP3V3KT+pGp0P665ZZbNv8IJNdoelWxYkWr+aXvScjV  
/bV8+XKpWrWq1f7S82RriPiLVqxY4f21LGF/JTepv5sK7a9WrVrl7P6KQpOa5JJLLkmYMJlWrtJv  
qJT9UUCTqpWr2rdvn7lvMqlcfdG68cYbl/dHQbVmzZpgDbnljvvnTN0fBTSpWvrDNBfpL4DJ+yKT  
0l82c9Fjjz0WuT8Kqnnz5gVryC3du3dP3R8FNKlas2bNCTaAnG5S9V12XczoHoznzKj/Rn2JXWsm

ik1dGZNnzYivZ/P4ztTTpsxELYr13HPPyVFHHZXyDZZJkxq1viJX+rUPjR5mHPf+cRLrYPZB1DzK  
r1rF5KmNT4XWFlxfTRWI+fWUqdDQ/dXw44aF2l9d13cNrSkYfU0V4e/HZ5991nvTSvL3WiZN6hNP  
PBG5ziJVXU2Fhr5mn/HNGRJrb/ZB1DzKr8zr/WOrHgutKRiDTOkcjtr21lhPmgqNZ8xo+kNTiVOT  
sU/Sldlfjyx/JLSmYAw1VZT2V1I988wzct5556V8r2XSpOb6YSVhOd2krjPDTAl3o4WppMmXE5VB  
k5oz5XJcZypqm1tzuRy3mlraZlGvDJrUnCmX4xFTUdvcmsvleNJU1DaLemXQpGrBR5PqctCk+oMm  
1c2gSbUbNKn+oEl1M2hS7QZNqj+SmtT48avw5fSeWG/GzmbshYx9zKg8s7LERpvJYIMjY5vXkTDu  
MFXP1N5Fs/ROLVWqVEn4BvMqgyY1an1FruqaCg2dX1VnV42eQ+nKzK96ZoTX5Y2HTek2ora9NVZt  
U6Gh+2vbudtG75N0pftrU8T+etxUUpdfSaXfj9tvv33K91omTWrdunUj11mkak9TSaPa/GoSG2X2  
QdQ8yq/M/KqzoU5oLcF40ZTO4ahtb421u6mkscOCHbK3v3qZKkr7K6J22mmnIO+1TJLUVatW5eyb  
P5PRric5t+m5KRMmXRWL5fZvPRdffHHKPkNxpOb6b4IXtrgybz9lUjvnV46+2bN9W7s3msX314Y1  
ubnDbr755pR9kq5JjX8/5uobze654568/WRRYxb1jktH77/4cj9UVAtnLswWENuefLJJ1P3RwZN  
6owZM4l1gCY1JP6bi54yQydKJqfZ0NNx6BWpclmHDh28fbF5f6VpUitXriwzZ84M7pmbrr/++sT9  
lab0YhJTpkwJ7pmb4o1XJvurQoUKMnHixOCeuUnPTan7YvP+StOk6umqRo8eHdwzN+mbVBL2V5oq  
Xbq0DB8+PLhnboqfESGT/VWYzEKZPHhwcM/cpKdc1H2xeX+laVKLFy8u/fr1C+4JRZOaj4EDB26e  
OPIV+fLlg6WhL9yb900+TWqpUqWCpTFmzJi8/ZVP6YsafJMMTYrcR8kF3/Tp0/P2S5omFb7Zs2fn  
7a80BV/8fOIFFXx6/uLN+yVNk4pU7JUC9OrVS+6///68CWZKR06ydu1a73aOG0n0xhtvyEOvP5T3  
DWjGEUOOkNVLVnu3s78S9e7dWzp37pwww455JDN561kfyV65513vKtKhfeXXu0sfklB9leiPn36  
yBM9nwh9N8Zkn3H7yOKZ/nk+2V+JPvzwQ+8qb+H5pcfralOm2F+JPvnKE3n66acT9pdeijf+10X2  
V6lvvvhCnn3u2dB3Y0z2mL6HzBrnnxeV/ZWKJjUDEyZMSPgm1CuOIH9TzYh/A+poZwbyl5zitGzZ  
MrgFUZJTnObNmwe3IMoSM/T7MD6amYH8rVy5MmF+nX766cEtiKKBTXh/NWrUKLgFUfT87Pp9GB8N  
zED+aFlzQJNqhybVDk2qHZpUOzSpdmhS7dCk2qFJtUOTmgGaVDs0qXZoUu3QpNqhSbVDk2qHJtUO  
TaodmtQM0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhS7dCkZoAm1c7UyaZJDe2vdi1oUtOhSbVDk2pn  
yWLTPlb2V7NTaFLToUm1Q5NqZ+NG06SG9leDg2hS06FJzQBNqp2pU5Oa1HY0qenQpNqhSbWzZEIS  
k9qMJjUdmlQ7NKI2UprUBjSp6dCkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk2qHJjUDNKI2aFLt  
0KTaoUm1Q5NqhybVDk2qHZpUOzSpGaBJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpNqhSc0ATAodmlQ7  
NKI2aFLt0KTaoUm1Q5NqhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUjNAK2qHJtUO  
TaodmlQ7NKI2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2a1AzQpNqhSbVD  
k2qHJtUOTaodmlQ7NKI2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk2qFJtUOTaocmNQM0qXZoUu3Q  
pNqhSbVDk2qHJtUOTaodmlQ7NKKZoEm1Q5NqhybVDk2qHZpUOzSpdmhS7dCk2qFJzQBNqh2aVDs0

qXZoUu3QpNqhSbVDk2qHJtUOTWoGaFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhSM0CTaocm1Q5N  
qh2aVDs0qXZoUu3QpNqhSbVDk5oBmlQ7NKI2aFLt0KTaoUm1Q5NqhybVDk2qHZrUDNCK2qFJtUOT  
aocm1Q5Nqh2aVDs0qXZoUu3QpGaAJtUOTaodmlQ7NKI2aFLt0KTaoUm1Q5NqhyY1AzSpdmhS7dCk  
2qFJtUOTaocm1Q5Nqh2aVDs0qRmgSbVDk2qHJtUOTaodmlQ7NKI2aFLt0KTaoUnNAE2qHZpUOzSp  
dmhS7dCk2qFJtUOTaocm1Q5NagZoUu3QpNqhSbVDk2qHJtUOTaodmlQ7NKI2aFlzQJNqhybVDk2q  
HZpUOzSpdmhS7dCk2qFJtUOTmgGaVDs0qXZoUu3QpNqhSbVDk2qHJtUOTaodmtQM0KTaoUm1Q5Nq  
hybVDk2qHZpUOzSpdmhS7dCkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk2qHJjUDNKI2aFLt0Kta  
oUm1Q5NqhybVDk2qHZpUOzSpGaBJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpNqhSc0ATaodmlQ7NKI2  
aFLt0KTaoUm1Q5NqhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUjNAK2qHJtUOTaod  
mlQ7NKI2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2a1AzQpNqhSbVDk2qH  
JtUOTaodmlQ7NKI2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk2qFJtUOTaocmNQM0qXZoUu3QpNqh  
SbVDk2qHJtUOTaodmlQ7NkkZoEm1Q5NqhybVDk2qHZpUOzSpdmhS7dCk2qFjzQBNqh2aVDs0qXZo  
Uu3QpNqhSbVDk2qHJtUOTWoGaFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhSM0CTaocm1Q5Nqh2a  
VDs0qXZoUu3QpNqhSbVDk5oBmlQ7NKI2aFLt0KTaoUm1Q5NqhybVDk2qHZrUDNCK2qFJtUOTaocm  
1Q5Nqh2aVDs0qXZoUu3QpGaAJtXO1ClJTEpVNKnzJ6V1KReSpOaDk2qnZQm9Uya1HRoUu2kNKnH  
0aSmk9KkHkGTmg5NagZoUu1M3WCa1OlMx830q91imtR0Zm8wTWpof7VcTJOaDk2qnSUbtZMaml/N  
FtKkpOTamftJtOkhuZXowU0qelsNCO8vvrMp0lNhyY1AzSpdqaaYfbU5tHODORvthnh/dXSDOSP  
JtXOEjPic0tHMzOQP5pUO2vNiM8tHY3MQP68JjU0GpiB/NGkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqh  
SbVDk2qHJjWNTZs2eR/nz5+f8KJ1++23e59Hovj+WmpGbFP8WzAmN5iBVPH9tcaM8P6iqY8W31/J  
x3TxRrNo8f2lwwPrQjOQKmF/hebX2WefHXwWYfnNr9PMQKr89teJZiB/NKn5WL9+vUyfPl1Klizp  
vVAVK1Ys4YWrT58+Mnbs2GBpbNiwwdtfZcuWlVhls7/MiH8TxjrFpPervdlfldpoTZs2TbbZZhuJ  
lUvaX11j0uv5XjJmzJhgaegLvJ41Yrvttov8fnzhRdk9OjRwdJQU6ZMkZ122snfX+H59VpMnu78  
NPsrjeTJk2W33XaLnF9dunSRUaNGBUtC/fPPP1KrVq3U+fVpTDrd0Yn9lWTSpEmy7777pu6vb2Ny  
z/X3sL/yQZOaJP7bTqVKIRJepPKrBx54wFs+V8X3V5UqVfL2y26mwuN5U8Ftd955p7d8rorvr223  
3TZvf+1kKjx6mQpuu+mmm7zlc1V8f22//fZ5+ytNXXvttd7yuSq+v3bYYYe8/WJ+aUwYH5kKbmvb  
tq23fK6K76+dd945b3+lqVxP7eP7a9ddd83bL6VNhccPpoLbLr74Ym/5XBxfX/Fffrwqbio8fjcV  
3Hbuued6yyMPTWqSESNGeJMI+TfpdJXLx6j+/fff3j5I2F9pmlSta665Jrh37tF0VPdBwv5K06Rq  
tW7devOLXa4ZN26ctw9svh+1kdCkOhfp8fPFixdP3F9pmlQtbST0LyG5SNPA0qVLW82v888/X9at  
WxeslbdoOI++fPnE/ZWmSdVq2rSpd5qqXKR/LdPAK2F/pWlStc444wxZs2ZNsAbQpCa57LLEiZM

ppWrrrrqqtT9UUCTqpWrOnTokLlvCmpStXK1idAkOXlfZFK5+iJ/1113pe6PAppUrRURVgRryC2d  
OnVK2ReZ1OLFi4M15JbHH388dX8U0KRq6fs6ctEzzzyTsi8KaK1Zs2aFawBOd2kbjCjhRnXBONa  
M+r+UFdiPcxEsaknY9LejPh6No83TF1TdEv/tLrPPvukfINlOqRGra+ol+6v/fffP2VfZNKkXn31  
1ZHrLFLV1lRodDDjoD8OktizZh9Efd/IV+b7se2GtqE1BeNjU1ebitp2ESj9BejQQw9NmTuZnKlt  
2rSJXGeRqtamQkNfs48YfiTEnjH7lGoe5Vdmfl219qrQmoLxjal2pqK2XQsqffv2cvTRR6fMnUya  
1FatWkWusyiX7q9jjz02ZV9k0qRedNFFQZeCnG5S15lhp0S70cJU0uTLicqgSaVCIUGTmjPlctxi  
KmqbRb0yaFJzplyOR0xPbbOoVwZnKhWqDjPULfhoUlO0mlR/OKSmL5rUvHI5aFL9QZPqZtCk+oMm  
NX0V0KTGj1+FL+f/3H+cGWcG4ywwzqv9VXWKfm8liU+ZFv4kZ8fVsHt1MnWHqzKJZZ511VuK7FuOV  
QZMatb6iXk2aNJE99tgjZV9k0qTqwRR6yxSdZqp0NDvqb1G7RX9PZeuzPffj6RtPD60pGK+aOt1U  
1LaLQOn8ql27dsrcaRJbdy4ceQ6i1SdbCo0dH7VHV9XYp+ZfRA1j/lrs/9OXXdqaE3B6GOqsamo  
bReR2nvvvVPmTiZN6imnnBK5vqJe++23X8q+yCRJ1WPqc/XNsslo15NceOGFKRMmk8pVLvq0SNOf  
vHEqX3rKn+R9kUmTmqvvVr/uuutS9kUmtXZNbr6b+LbbbkvdHxk0qXqVpVx03333peylTGrJ4iXB  
GnJL586dU/dHBk3qvHnzgjXklm7duqXsi0ya1JkzZwZrAE1qhPjEyuS0JLvvrssW7YsuGduir+D  
cfP+StOk6vn1cvWdsXHPPfdc4v5K06TuuOOOOfvO2LiXXnopcx+IKT2f6pw5c4J75qZXX33V2xeb  
91ealXPbzjxozgnrnpjTfeSNxfaUovvqEXlchlb7/9trcvNu+vNE1quXLlvJPY5zK98E/C/sqn  
SdXb9XR048ePD+4JRZOaD50o8atNRZW+WNWoUSNYGnr+vBllSvjfiDXMPgqPHmZ/lD7Gu/oNfHqK  
kc3ns9wxtK90vBqTiqUqeg0X/BNi6/X6dX9pRX0/VqxYUapWrRrcI7fp/lq6dKk3t7z9tY3ZR+Hx  
YUwqFK/gnb8R/v5atWqVN4/ym18VKITw5lh8+Vymz1+vyKj7RV/zY6XMPgqP72NSPIbea1Djy+ey  
+PPfvL+KhfaVDtOk6v4qU6aMt1yu769kNkkF+PXXX+W1115LeMHSY7/if35lQix67bffpPdvfvO+  
Ac04bfJpsn7leu929leivn37ynu/vhfaWzFpOLWhrFvunyy/ZXozz//lA8++CDh+/G4447bfLJw  
9lei/v37yyc/frKaXTE5ctaRsnRau929leigQMHyueff54wv/S0XtrEKvZXosGDB8tXP3wVml0x  
OWD+AbJyrn/4CPsr0dChQ+Xb778N7a2Y7L1wb1k+a7l3O/srFU1qBvQqLuEXrVy+wLQmppor/wbU  
0c4M5G+2GeH91dIM5E9T1fD3Y/PmzYNbEGWJGfG5paOZGcifHp8bnl+nn356cAuirDUjPrd0NDID  
+dtoRnh/NTAD+aNJzQBNqh2aVDs0qXZoUu3QpNqhSbVDk2qHJtUOTWoGaFlt0KTaoUm1Q5NqhybV  
Dk2qHZpUOzSpdmhSM0CTamfqJtOkRJH7Kqh262lS05m9yTSpof3Vcj1Najo0qXa8JjU0v5qto0lN  
hybVjtekhuZXo3U0qel4TWpofzVYR5OaDk1qBmhS7UydYprU0P5q14omNZ3Zs0yTGtpfLzVtpKZD  
k2pnyRLTpIb2V7PGNKnP0Kta0TcthdvXo2NoUtPRN12H91eDw2hS06FJzQBNqh09j2B4f7VrR5Oa  
zuzZSU1qS5rUdGhS7aQ0qc1oUtOhSbWT0qQ2oklNJ6VJbUCTmg5NagZoUu3QpNqhSbVDk2qHJtUO  
TaodmlQ7NKl2aFlzQJNqhybVDk2qHZpUOzSpdmhS7dCk2qFJtUOTmgGaVDs0qXZoUu3QpNqhSbVD



k2qHJtUOTaodmtQM0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhS7dCkZoAm1Q5Nqh2aVDs0qXZoUu3Q  
pNqhSbVDk2qHJjUDNkI2aFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpGaBjUtUOTaocm1Q5Nqh2aVDs0  
qXZoUu3QpNqhSc0ATaodmlQ7NkI2aFLt0KTaoUm1Q5NqhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5N  
qh2aVDs0qXZoUjNAk2qHJtUOTaodmlQ7NkI2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOT  
aocm1Q5Nqh2a1AzQpNqhSbVDk2qHJtUOTaodmlQ7NkI2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk  
2qFJtUOTaocmNQM0qXZoUu3QpNqhSbVDk2qHJtUOTaodmlQ7NkKZoEm1Q5NqhybVDk2qHZpUOzSp  
dmhS7dCk2qFJzQBNqh2aVDs0qXZoUu3QpNqhSbVDk2qHJtUOTWoGaFLt0KTaoUm1Q5NqhybVDk2q  
HZpUOzSpdmhSM0CTaocm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk5oBmlQ7NkI2aFLt0KTaoUm1Q5Nq  
hybVDk2qHZrUDNck2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpGaAjUtUOTaodmlQ7NkI2aFLt0Kta  
oUm1Q5NqhyY1AzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qRmgSbVDk2qHJtUOTaodmlQ7NkI2  
aFLt0KTaoUnNAE2qHZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5NagZoUu3QpNqhSbVDk2qHJtUOTaod  
mlQ7NkI2aFlzQJNqhybVDk2qHZpUOzSpdmhS7dCk2qFJtUOTmgGaVDs0qXZoUu3QpNqhSbVDk2qH  
JtUOTaodmtQM0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhS7dCkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqh  
SbVDk2qHJjUDNkI2aFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpGaBjUtUOTaocm1Q5Nqh2aVDs0qXZo  
Uu3QpNqhSc0ATaodmlQ7NkI2aFLt0KTaoUm1Q5NqhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5Nqh2a  
VDs0qXZoUjNAk2qHJtUOTaodmlQ7NkI2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOTaocm  
1Q5Nqh2a1AzQpNqhSbVDk2qHJtUOTaodmlQ7NkI2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk2qFJ  
tUOTaocmNQM0qXZoUu3QpNqhSbVDk2qHJtUOTaodmlQ7NkKZoEm1Q5NqhybVDk2qHZpUOzSpdmhS  
7dCk2qFJzQBNqh2aVDs0qXZoUu3QpNqhSbVDk2qHJtUOTWoGJoxLalKvpkInhybVzuxZSU1qc5rU  
dBbMT2pSm9GkppPSPDamSU2HJtVOSpN6NE1qOjSpdmhSMzDBjNgGM6GC6riRjJUdmlQ7s80Iz6+W  
G2IS01lgRnh/Nd9Ik5rOEjPC+6vZRprUdGhS7aw1Izy/Gm2kSU2HJtUOTWoGvCY1NDqagfzRpNrx  
mtTQaGkG8uc1qaHR3Azkz2tSQ6OZGcgfTaodr0kNjUZmIH80qXZoUjNAk2qHJtUOTaodmlQ7NkI2  
aFLt0KTaoUm1Q5OaxqZNm7yP88yIbYp/C8bkdjOQKr6/li5dmvBNeP3113ufR6L4/lptRnh+tTMD  
qeL7a4MZ4f1FUx8tvr9UeH9daAZSJeyv0OvX2WefHXwWYfnNr9PMQKr85tcJJ5wQfBZRaFLzsWbN  
Gvnhhx/8ibR9TIqZ4X8LmnFPTLo+2lW++eabYGmsX79evv/+eylevLi3z4oVM/sr9I346KOPYldf  
fRU5Df1tWvdXmTJIJFYmaX51jUmnuzvJI19+GSyd2+lv7t99951UqFBBYsWT9tfzMbn31nvliy++  
8JbLdfH99e2330qVKIX878fw/nrN/KJ93e3y+eefe8vluvj+0tfzatWqRb5+3XjjfLZZ595y+W6  
+P76+uuvZdddd02dX5/G5NorrpVPP/3UWy7cnOWi+PPXn3977LFH5Pxq06YN+ysfNkIJ4hMkPIFi  
tUyFx1Omgvtat2/vLZ+rlvdXmrryyiu95XNV5P7ayVR49DIV3HbppZd6y+eqyP21nanweNtUcNv5

55/vLZ+rIvdXZVPh8ZGp4LazzjrLWz5X2b5+nXrqqd7yuS6hySptKjx+MBXcRkroK1GiRN7+SINH  
HXVUcA/E0aQm0fRKJ0vCN2GaJIWradOmwb1zj6Y1ug+SfzNMV2eccUZw79zz448/evsgYX+laVK1  
9AejJq+56JdffvH2QcL+StOkauI5Gjds2BCslbf07dvX2wcJ+ytNk6p1/PHHy7p164I15Jb+/ft7  
f/2xef065phjvL+05aJBgwZJqVKIEvdXmiZV64gjvCO881Ff/31I5QtWzZhfxRUhx56qCxfvjxY  
A2hSk1x22WWpE6eAJIUrV1115VUp+yKTylXXXntt6v4ooEnVytUm9eabb07ZFwU1qVp67sZcdPfd  
d6fsi4KaVK1cbSI6PdQpZV9kUksXLw3WkFsef/zx1P1RQJOqpRfgyEXPdn82ZV9kUnrubPhoUkOG  
Dx+++RiuhMqgST355JPlpJNOKvoVGiebsevoXSX2vdkHNvVLLSW0OhmqpGpqO0WgdI5UqNgjZS5  
k0mTqulgl1DqLcun+2n333VP2RSZNasOGDSPXWZRL99eee+6Zsi8yaVJPPPHEyHUWqTrWVGjo61fN  
f2pk7DuzD6Jep/Ir8/p1wvoTQmsKxtumTjQVte0iUrVr106ZO5k0qccdd1zk+opUHW0qadSdVtd+  
fn0dk2PXHBtayOneGRO6m5GLaFJDfvrpp5RvLq8yaFJzooqZcjnamIrablGvDJpUKIQZNKIUqDJo  
UnOmXI5HTEVts6hXBk1qzpTDcY0ZuYgmNUTfbR058WhS/aJdVM0qXZFk2pXNKI55XLQpPqDJtXJ  
oEmF56qrlo6xzKBJ1WPn9DQIRbpuMhUat5ix/3f7S+wJsw+etKhOsdBaQuN9U1HbLSJ1yy23yEEH  
HZQydzJpUm+44YbldRbl0u+pww47LGVfZNNkduzYMXKdRbl0f+mbVJL3RSZNqh4rHbXOIXXmgqN  
m804pv8xEnvc7IOo16n86uGYtF/TPrSmYPxg6npTUdsuAnXTTtd5f7ZPnjuZNKIXX3115DqLVLU3  
FRo3mdHwr4YS62L2QdQ8yq/M/Gq7om1oTTfKDWZ8aEYuoklN0qpVq5RvMN44IT998UneF5IUrtJm  
M2V/ZNck5uq71W+77baUfZFJk5qr776+7777UvYfB5zKX+fOnVP2RSa1ZMmSYA25pWvXrqn7I4Mm  
df78+cEacstzzz2Xsi8yqVvmzZgVrAE1qBD3prk6UzafZSNokHnDAATn7TuI4m9NQ1atXT1avXh3c  
MzfFLxKxeX+laVJr1aqVsw1E3M8//5y4v9I0qfpGq1w/fctvv/2WuL/SNKm77LKLd4W4XPbHH38k  
7q80pSf7X7x4cXDP3KSn7dJ9sXI/5dOk6u36RuSFCxcG98xNgwcPTtxf+ZTerhcrydWGPj80qRH0  
BM8zZ87MeyfjHqbC43HTPOxea/OJd+MnhM5V+vz1G0sbquRvwHjVrFIT6tevv3n5XKbPXxsD3Sfe  
C9f2Zh+FR8+Y7LXLXnLwwQdvXj6X6fNftWqV7LXXXv5JsZObrjdisdOe8iBBx64eflcps9frwCn  
+6tkyZISKx/aVzrej8lu1XaT/fbbb/PyuSz+/PXMCHoO0KjXLz0rxz777OMtx/7yn79ePcm7YI7y  
exW+jUn1KtWlbt263nLsr7z9ld85U/WXxTp16njL5fr+SkaTWoDJkyfLj1N/zPsGNOPyRZeLrA8W  
QIlpU6ZsTnLiddFFFwW3ItUqVOI3+R+odkVk2ZLmonk5I+rCzRt2jQZNHIQaG/F5IylZ4jkdjif  
r+nTp8tf//wV2lsxOWnZSSKrggWQYMaMGfL3338nvH7pcZilpmHOMEljQrMrJkeuPFKEc9FH0j/j  
jx8/PmF+HXLITSmadCkZmCCGfFvQB0dzUD+tPEKfxO2a9cuuAVRZpsRn1s6WpqB/C0wl7y/mpuB  
/C0xl7y/mpmB/OnhNeHXR9NPPz24BVHwMhGfWzr0nJ7In16YJTjy/GjRoENyCKDspGaBjtUOTaocm  
1Q5Nqh2aVDs0qXZoUu3QpNqhSc0ATaodmIQ7NKI2aFLt0KTaoUm1Q5NqhybVDk1qBiasM03qJDOh  
JvvVcSFNaJo0qXZmbzBNamh+tZxPk5rOgo2mSQ3tr+bzaFLTwbLRNkmh/dVsLk1qOjSpdtZuMk1q

aH41mk2Tmg5Nqh2a1AxMGG+a1NCK6tiBJjUdmlQ7s2eZJjW0v1peRpOazoL5pkkN7a/m59OkpqPn  
9Azvr2ZNaFLToUm1o6dgDO+vRsFtpKZDk2qHJjUDEyYkNakdaVLToUm1M3t2UpPakiY1nQULkprU  
5jSp6aQ0qc1oUtOhSbWT0qQ2oklNhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUjNA  
k2qHJtUOTaodmlQ7Nkl2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2a1AzQ  
pNqhSbVDk2qHJtUOTaodmlQ7Nkl2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk2qFJtUOTaocmNQMO  
qXZoUu3QpNqhSbVDk2qHJtUOTaodmlQ7NkkZoEm1Q5NqhybVDk2qHZpUOzSpdmhS7dCk2qFJzQBN  
qh2aVDs0qXZoUu3QpNqhSbVDk2qHJtUOTWoGaFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhSMOCT  
aocm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk5oBmlQ7Nkl2aFLt0KTaoUm1Q5NqhybVDk2qHZrUDNCK  
2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpGaAJtUOTaodmlQ7Nkl2aFLt0KTaoUm1Q5NqhyY1AzSp  
dmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qRmgSbVDk2qHJtUOTaodmlQ7Nkl2aFLt0KTaoUnNAE2q  
HZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5NagZoUu3QpNqhSbVDk2qHJtUOTaodmlQ7Nkl2aFizQJNq  
hybVDk2qHZpUOzSpdmhS7dCk2qFJtUOTmgGaVDs0qXZoUu3QpNqhSbVDk2qHJtUOTaodmtQM0Kta  
oUm1Q5NqhybVDk2qHZpUOzSpdmhS7dCkZoAm1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk2qHJjUDNkl2  
aFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpGaBJtUOTaocm1Q5Nqh2aVDs0qXZoUu3QpNqhSc0ATaod  
mlQ7Nkl2aFLt0KTaoUm1Q5NqhybVDk1qBmhS7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qXZoUjNAk2qH  
JtUOTaodmlQ7Nkl2aFLt0KTaoUm1Q5OaAZpUOzSpdmhS7dCk2qFJtUOTaocm1Q5Nqh2a1AzQpNqh  
SbVDk2qHJtUOTaodmlQ7Nkl2aFLt0KRmgCbVDk2qHZpUOzSpdmhS7dCk2qFJtUOTaocmNQMOqXZo  
Uu3QpNqhSbVDk2qHJtUOTaodmlQ7NkkZoEm1Q5NqhybVDk2qHZpUOzSpdmhS7dCk2qFJzQBNqh2a  
VDs0qXZoUu3QpNqhSbVDk2qHJtUOTWoGaFLt0KTaoUm1Q5NqhybVDk2qHZpUOzSpdmhSMOCTaocm  
1Q5Nqh2aVDs0qXZoUu3QpNqhSbVDk5oBmlQ7Nkl2aFLt0KTaoUm1Q5NqhybVDk2qHZrUDNCK2qFJ  
tUOTaocm1Q5Nqh2aVDs0qXZoUu3QpGaAJtUOTaodmlQ7Nkl2aFLt0KTaoUm1Q5NqhyY1AzSpdmhS  
7dCk2qFJtUOTaocm1Q5Nqh2aVDs0qRmgSbVDk2qHJtUOTaodmlQ7Nkl2aFLt0KTaoUnNAE2qHZpU  
OzSpdmhS7dCk2qFJtUOTaocm1U6sW7duQqWvu+66K2FSHXfccdK9e/flZalucu/99yfsr6OOPlqe  
eeaZyGWpbvLAaw8k7K/DDzv8X99fPXv2llfeutfqd69e8uTTz4Z+TgyqU6dOiXsr0MOOkSeffbZ  
yGWpbtk5c+eE/bX/fvuzv9JUl8e6JOyvvevtLT169IhcluomTzzxRML+ql27NvsrTXXt2jVhf+25  
557y3HPPRS5LdZNYnTp1hEpFOonCk6pq1apSt27dyGVzvurVkpzmH9ULVd0qVaQO+yvf2qvOXhKr  
Z+ZWUJXqVpK6df69/aU/VPQXL03Y/o1q2rSp1KxZM/KxZFI169RM2F/b1N3mX91fW1vVqlMrYX9V  
rFuR/ZWmatepnbC/KtStIPXqmNe1iGUp//Uj/POxfPnyUk9/DkQsS/kV3l/lypVjf6Up/twPAAAA  
AAAAABkgTAUAAAAAACADGQUpv780+dyY4ejZPRvh4ssPEqWTTHaHr3nQHm+x8PBEnnatesgpUqX

lx123FN+/PHX4LMimzZtCv4FAAAAAAAAAAFufyDA1HHwumvOnjP/rQRn5ZxuZnN2UzF8mMmveMun7  
490y7lcDZPmYbeW3d2Py3nM7yGdvny1ffvqY/DW4pyyYfre0uKCCnN/OgGBNBKr/ttS9nYX9n7IK  
vqYAAAAAADlJmzlH65yyHyPTH32+VelXr1q0v/TKrJp0U0yd8FSmTN/kSxYvND8e758/dMA+f2H  
W2XF+Jry87d3yfe/9Jd5C+fL/EULZLZZbva8+SKru8nnbx8ipUsVk+fN+vAv2rRaZEKf6XPTJXJk  
qYZy41t/y69rgtu2yESZOuA5uaHisXLD9W/LO0tEVm0MbglAAAAAEDGli1bJkOHDpWBAwfK4MGD  
M64BAwblyJEjZe3atcGaipY+ffpl8eLF5aGHHpIFCXYEn83f+vXrZeLEidK4cWOpUaOGzJo1K7gl  
+/INU9cs7CUzBu4gv7xXVX75+BAZ+eflMqLfLpW2fLUw9sJ39+uZ8sGnOQLB9XU3784Gjp3q2l  
dOn+jAwfO1tmzhotP3zWQhaMOID6fVJFTjyqrHz8Ue9gzfkrXGDS3yl+35RVZLROXSj9grr+LQ20  
832MYSkbSf3/6MdiPu/9NzvyT+9nyPyhPaRj2aPl2g695a2MwtSoR+b/f9TzyPtvnwdfGAAAAAA  
wFZqyZlIXjiaSWAYNn36dPnrr79kzZqsHDn3n7RixQp57rnn5Oxm58hpjU+XL778Mrglz6pVq6RL  
l8flxlaNpEXLK+Sbb74JbnEn3zB13PCe8tbTleWDF7aTv76plasm1pQ1/9SSVRP2IBXja8uKaafL  
iqmnmH/vZW7bS9ZN3tPctqt880YV+aLXdrJk9F6yYVotmTloF/n6jWoydfx7wZrTmSxxz70kHcse  
Kq0bXii3vvqAnFI7D9krFpOKVavIri3eli59F8iqFUPI00t3IA7nN5WLPhMZszi4+8b1lv3vkJ63  
XyixBs9L9/6zZZ38l0t+v0tuPG4f2TIWTCqWryD7Xn25nNLuejk7tr9ce9sH8u5KEXNPmfV5jflC  
uTGpXq2SLC4Vk7l71pDYSU/L3R9PN7duMPWHfHjJa2lu7ndbtwfkqg4XyVEISooV8/j2OKW1XPbW  
DPn5pWvk0Yv2IDLlyktp8/lT6jWSA+4flb9Njf+IYLXItA/I+8cukmMqVpFqZUpKmdK7S8USF8s1  
r/YzW/C3FPbP6xfJa212kHNemybvT9LPmAcS38m7bZrLOSVOIBs/mSK/6PfOpuUiU5+WHs8+JYdf  
841MWf6jDOh2g1wcO1xu6TVKfjPb3rDqfXlk70ZyxQ5Hyu0Pt5Z9D6knJfVxmudbtUkXaf3efFkZ  
Dy5XTZAV/e+XO085UKrHikvFsuWkXquL5dQb75ALix0hN930jry91Cymyy78Xkb1vJJO32kn2T5W  
RspXKCdlzXrrnn6ttP9kvUwxT3/+jw/Kx1eVkoBp/i5Pj9A76V7/Q37p2kEuiB0iVz/9p3xsnoL5  
QorMeU2+fPthObDVV/LNoPn6SQAAAAAAGCIIHqbOn2+XfUybNq3lh6lxsDf5MmT5dTtGsve++wr  
H3/yiXdk7hNPPiIVtt1O7r3vPlm6dGmwthSjYWr4aMThI/+WTo8+Kt2f6y59v71M1k2qJhum7OYF  
qvrWfg1R/TL/HreHLJ90qKyYfrbl/INI4Yid5P3nt5Ub2tWQgw7YRmKxmLz88ovBmtOZKvMmvCK3  
b3+wXHjSPXJ3vxUyOzjqcdPCMfJ5693kjo7t5JY/RZau/1G+79pJTt7mHnmn30xZ6h3LOEheaHWr  
dLr6Bf9hfnfDT9JzvfvlhPqdZAn+vwt/3hrUn/KsHdvkXNKHClT7vxY3l8u4k291f1lzpe3SMOj  
DpVadapL7T1rSIXYPnLq1W/JW4s15BwgP9x1pVxW8ghp+XQ/+XDzEcO/yIf3tZDTYgdlq+cHy2eb  
v35/yG9P3yxNYo3k1l4j5I+N5IFO7SH3HVNxasVKSrWa+0nN/Q6WA8229t+5uMQa3CAHPT5KVqxJ  
iIMXfyVjPr9X9j+ph3R6f4YsW7tQRnU+S958+Axp2/V+OfD0l6Tzu8NI7tpx8mWrhvLwzQ/IA39v  
ICUyVPp1vVlaFT9B7u491jyaNbJx9cfy5EEN5IL9W8v1f6yUSZpnyiKR5W/L/fXPkmb73ijPTtkk

C9f1lREf3icN67WXh3sNlgneA1GDzGO5Uy4ofaS0uv5DeX+xeayL35IXLrhUTqzcSp4ZtUQmBkvq  
Hhv1ekfpdub20u7tCfLV6HfMf98vJ53dVdo8MdJsdZNMERg1fHjviXJjzvl0GY9pMPjf8hsmSW/  
3nmRdG3VUm4fsEb+9gJWAAAAAACwtbv33ntlxx13lfQ1aknt2rUzKl1W7/Pggw8Gayk6CFMzt2nj  
RnnjzTe9nFGrVu06MmKEd7TevyrlyFTvENoXX5RevV+U/oO/le49npQuTz0tr73TW957o530/fJM  
mT78TBnzxwky6NsJ5LfDPdSv3qstX7+zq3z15q7Sus3JcnKzq+Xya26UXfesJU2aNAAnWnBjWRpsm  
88a/LLdt69cctbD8sg483i8z6+SDXO+ICeOqi2tL7xOHholMktXNftrmdnnDDnrznfkwjad5buO  
B0qLHl/KfYP1PmaBTd/Jb2/cJSfWvFlee3tEKBD8RYa8ca2cXuJlufLe7+SzBUtF/r5fOjZsLXXr  
PSe/B0vJ7G/kqaOOI9tve1leX6jHUfaXb++4Wq4od5rc/8UU+WOdLqT/GSQ/PHqjtCpxotz99t/y  
kzePNRAdJv163i1XxRrI7S/+ZT6/RDZNf15u3fdsOf3Ae+XVld5xqpGSd9Wyaf3li4uPkcufvUDa  
vvGcnNHkLXnjzGyYM1E+erK1vLOPZfK0188lvUv6CV3vTDG7DFdwRD5uctNoTB1tRemPn7AqXLV  
EbfIM5PNHvfC6oVme1/K08eeJVfu00G6jt0k89b2k5Ef3Ssn1bpCHuw5UMboYp4/5O8PbpSzSh0h  
l8fD1KXvylsXXCQnVrxYug1fGtrPK2RYzxul80k15aaPxsrXegjr6knya4cm0u26U+W6Pt3ljAve  
kMefGWwewRz55c7b5O3258vL3z0gJ171krS8t59wTCoAAAAAAEVDNi4lI11/JcQphZMz4X6+BNP  
yNHHNJdZL7jQO8WB+vnenn+XQ+ofLOeeeJ2/27u3ty39DSpiqJ6797Y/f5fd+P8gPv3wkBx26tzQ7  
t7EMGPK9jB7ftZ767B1p2/4aqVV3Tzm5aWN59/135ZeTj5MeNWvlGRdeJB07d5ZeN1wn3+25s9x7  
8onS5YXnZd167/DHDEyV+RNfl9t3Olla1j1MLmpxuGy7nX9ka4Uq28lJw6Rd8YHiwY2rF4sQx85  
SFqcvZ+UbP2ZfDkq/p7/wlKhsqbPpXL6ITt76ylRvLjUaddKTr3+bmKRO1xuu/dz+WClvrH8H/nz  
8fPkquoxqVgyJrE9G0jVoy+Xc+sdIDc/3Et6LdAwtZ98ffNVcknsRLn708ny++YwdYB81+lauTR2  
tNz25kj5fnOYOIT+eOE2s536ctMzf8oX3ufN8gs/kZ+7XiCHxUpIBfOYihWPSckSxSR29I1y3DOj  
ZMXapCNT1QbzAGb0kMeanCwHVTbdbv1+jgz0dusykbbkvy4uXnykHlzpJrv94kvzmJbT6uAbK949c  
J5fFjpLbXh8tv3tv8/9QHqlzgrQ46Hrp9o/Z496mFpgXo8/lycNPk8v2ulK6jNnoB6JLR8nGj6+Q  
C46u7u274pr6X9FcTrrpLrkwdrDccN2b8qaeM1WXXdIXJn7UXprsUNU77UGseFlzn9pyTlse8sJE  
kfnxKbDRPLiFveWdjufIQWZ/tX11mHznPV4NZd+Xr+6/2OyXI+SKbr/LZxyRCgAAAAAAijDOmRpt  
3bp1Mnr0aDnq6GO8t/Z///33smRxUuYXmDlzpjbzo4dU3KaSdOx4vcybN082bnR3tfSUMHX58uXS  
rWs3aX7xRXLBURU3kscc6yHdf3CR/928tn751svzw+WUYa2p3WbPoWVPd5Z/xT0jn526Ray48Vx47  
/Ch59s6b5dk3e0qn7k/JPgcfLKc2Pt3iCztV5o1/VW6t2kDaX/qivLHcO740SfgzZscs6yvdr2wm  
t3S8V96bITivIEPm+8eKyV3ki9uPlm2r3CA3vTrUbNWPPguUsj7/E6nbKeDz3n/Ti1om9XP5XXTK  
/0xmf6zxL2iVyl93wYJtef/NT+KtqcvmT470awUAAAAAANia6UGNeuV5DUf1aFP9mK50mfhyC+fO  
lQ0bMkq0tjrDhg2Tm2++Wfr27Rt8pmBr166R119/3TsdhMtZqEZegCp+yPTGDWvI9ptOldYXl5E3

ntpRjvfbXZaP21PmD99D5g3fUxaNMvX37jLs253lmzd3lJ6PVZN9axf3jmlsVaQufP755956MjdN  
5k/sLffufprcdOWL0nuByNKUIHmDrF0+WD5tu4+cWbmkIKlOplz70TgZGBk4r5IZ/V+UZ44vLftX  
jkmsWDHzuEqYx9dQDj7lFflm4ybxdi2ZHQAAAAAAAIACRIapcRs2rJdxYwbJ6BHfycwpP8mMyT/J  
uNE/yaCBP8nAAT/JiGE/ybRJP8mGFb/Igtm/yg8//iyffPqFDBkyJFhDXjCbKWyHmoSkAAAAAA  
ALikbZgKAAAAAAAPARpgIAAAAAABABghTAQAAAAAACADhKkAAAAAAAKAHCVAAAAAAADl  
AGEqAAAAAAAGSAMBUBAAAAAAAMkCYCgAAAAAAAZIEwFAAAAAAAgAwQpgJbm02bgn+kEbV  
M  
uvtlsk4AAAAAAIAcR5iKrcOqVSJTp4pMmZJXK1YEN2bZxo0is2eLTJ7sb1M/zp8f3PgfsXatyBFH  
mO9g8y0cr2rVRG6/XWTcuGChCP37ixQrlni/hx8ObgQAAAAAEa6seAj8N/2+eeJAaDWq68GN2bZ  
4sUihx6auK1mzYlb/yM0TG3YMPExxqtRI5E33ggWTDJokEjlyonLP/FecMAAAAAAADSiQUfAbfW  
rBGZPt0P8378UeTLL/2A9Kuv/P/Xz8+a5YeEcfrvmTP92+68U6REicQQsEMHkT/+EBk9WmTISv8+  
y5ajB0r8uuvlt98l/LxxyLvvy/yySf+//ft69+uy4Xp29wXLRIZM0bkrbdEatdO3NZRR/mPd/hw  
kblz/fts2OAfwRp/TvpcvvgicTvLI/vLRtHH/P33lh98lPlpp/7HUaOCGwsQFabqEafFi/v/3mkn  
kbvuEhk/PrhDwCZM1ec3Z47lsGH+/vz2W//5aX33nf8cdd8vWBDclUS/lvqcPvrl/xrov3Vfr7t  
r0/3l65H9+nXX/tx0mT/COQo+jRwgsX+vf9+Wf/PvH76mMbOdK/HQAAAAAAwKFY8BFwY8IEkSef  
FNI9dz+4i4d9yRX//DHH+AGcvoVfA8933xXZZ5/U5eO13XYit94q8tlnls8/L7L99tHLxSv+Fvf9  
9hO55x4/LFTr1vmB3qWXpi4bflv8BReI9Owp8vbbbg0a5H0+qvQ5Va0qcuGFfiic7J9/8vZLvG6+  
ObixAFFh6uGHi7z5ph/8xj93yin+UaoajKpMwlQ9Mlf3zf77+88hv69ZvMqX9x/L668HKzD0a5i8  
3C67iFSvnvr5cJ1+usg77+SFqho4azitRwZHLR+ubbbxvyb6HAEAAAAAABylBR8BNx57LDHw0uBN  
j5gcMkRk/Xp/GT3H5333iVxyiR+mXXyxfwRimB59WLk4rr69AluNDREPOsskTPPFLnsMpFnn/WP  
gowbMcJfb/j+O+wg8vffwQllepSqhrrhZXWdStf56KN5n69SReTEE0Wuu07kkUf884/edpv/WCpW  
9JcpXVqkfn3/CM0wDS31cd57r0inTv5HPWlze1Fh6tln+7fpc2rfPi8E1nOpauCs533VfZ0uTNWj  
PfVrUKqUf1u5ciJNmvihdnh/6tG5XbsmBrd6NOzVV/tHquqRq/HPx+uqq0R+/90/yITpftbgs3Hj  
xOX0FAt6lKkG3L17+4G0fr5MGf88sboe3V+6v3UuaQCu247fv04d/7EBAAAAAABkWSz4CLihb7/W  
oLRmTT8M1UAsHnrFS0PNU0/1w8Sffkp8q3+cnhYgOUx97bXgxoAGcPp2/htvFDn2WJFdd/W3p0dW  
VqokUrZs4v3zC1M15DzyyMRlNehVesTlaaflv4fkwaq227rHyWrH+P/jv+/3q7/fvllfx3ZEBWm  
nnFGXlCpwanun8MOy7tdw+SbbkrdD+Ew9YEHEm/TkPKvv4lbl7zwQuLy+jw1uP7lI8TPa+lpDKJo  
OBpeTo+IXbJEZPBgP8SuUMH/fPzrmLx/k/+ty+scAAAAAAALJY8BHlJ3qdNo0/5yaepSpXkVe  
z0eq583UgFWPwOzVS+SGGxIDP6299hL57bdgRYGoMDV8kaXHH098C3mNGv45OefN898yrkdb3nFH

4v0LE6bq49a3+sc/r2+tHzjQvy2ZHr2pb+XXUx3ox6VLgxsMfTz5SXdbXEFhapwGoRqghpdLrnCY  
+tRTfiAZf2u/njZBzzmrp11INnWqfxRufD16n7p1/XPf6nIN45+PI64nSISYql8DnTvXXuu/fv8/  
r8GuhuVRYbsuP3myyMSJfmmYDAAAAAAAGWx4COQXRpgapipR5zGQzI9UITfBq7nG9UjF/XiS3r0  
5Lnn5i2jR02ecELqeS/1vJnJR7VeeaX/9n8N9TR0C9+uRynq5/T8m7o9fft/8vIUNTSMOupSQ8/k  
t/nr2/R1PRol67IB9Ujb+G36tnl9XnrkpV6kqXt3/63y8ds1tH3olT9kDNPQL/kt99dcE9xYgEzD  
VKUX1nr1VT/4DS8fr3CYqueQ1bfWaygaXqZRI/+ctPp10PPT6tvsNfQOL6Pbj5+mQE8XEL5NK9Mw  
dd99/RBc6QW5jj467zY9ZYKeP1VP66CnEtDnpUevxi9Oph/1wmT9+vn3BwAAAAAAyKJY8BFwQ4NJ  
PSpVr8Z/3HEiO+6Ydz5OLT2Hql45X8/3qW8Z1yNFo4481Ld9a1DZvLI/DtD4/Xfe2T9Pqb61fOhQ  
kRYtRPbcM+98oXq0qoZteqTre+/5oae+3V0DOV32pZf8lyDD9GJNetX+++/3z7+pAV58e5dfLvLD  
D35YrOvUizMPPjjvrehaeiSlBpd6nllNe/UiSIFHm2qYGn4uWnru1UzoPgqfr1RLw9WoMDVOjxDW  
xxS+j5aezSZrn/AAP82vYiVBuHxwFJLz2OqF/Fq1coPmTXQDm9bz2cb3oaWBttR9OsTXk6/fnpU  
b5x+PfQcu3quWp1D8XOoamn4rhcoa9vWP2pVj1BNtw8AAAAAAC2QCz4CODftCVv89/S26MU5j75  
sXluNssCAAAAAAD8nxGmAgAAAAAAAEAGCFMBAAAAAAAIAOEQuAyPefm+PEigwf7F6jSj3ql+Fyx  
Zo3l9On+xb03LG6DyZMEFm40D9/Kf4dU6YkzsFx49j/AAAAAAD8nxGmIrdOSHjCCf6V/Hff3b/q  
v15oKmzZMv+iRuGLijVuHNxYxH38sX/BqSpVEp+/XmH/2WdFFiWIFoRzF1yQ+DU49FD/gm4AAAAA  
AOD/xvyGDuSQ+fNFatdODKluuim48V+U7YsrZetCTg89lLhvdtlFZNKk4MbA//vCUJlsP1vLFMTI  
vrjoosSvxeGHE6YCAAAAAPB/Zn5DBxbu1akXz+Ru+8WOewwkR12ECIVKi8k0qMgDzplpG1bkU8/  
FVm/PrhjIEDRR55RKRRi5HddhOpUEGkTBmRatVEjppK5LbbRPr3DxYOrFol8tNPIs2a5W0zucqW  
FbniCn/9q1eLNGiQePv55wcrS6JHaurRnO3bixxxhEiNGiLly4uULu0f/brffn4o1rNnaih5zz/+  
0bHh7Vx3nX9KgZtvFtl/f//5lSwpUrGif3Sobue774IVJPniC5HixRPX99ZbwY1pDBokcuONItts  
k3jfcNWtK3LXXSKzZvn30a9R374iDz4octJlRvqiVSu7D9vXc9ee/lHuHbqJPLnn6mh46WXJq7/  
4lP9t7W/+qo/F/Q569dEjxwuVsxfRo8WTt6Huv74OnR5/br9/bd/m56uQd8if+21ecvoPnzxxbzn  
oUaNEnn6af+I5V139dcTX173f506/tfwlVdEFi8O7hSiRzvXrJl3Hy3d5pw5ltdfL7L99v4633hD  
ZMWK4E6GPt9nnhE5+WR/rug81vC6XTv/cZ91VuI6CVMBAAAAAPi/M7+hAw59/bUfqsUDouQVRfQ3  
9kMkDbUeftgPk8Kh0d57+0Fd3LRplk895QdN8WU0fNJArnt3keeeE2nRQmTPPfNu13BOA1w9x6QG  
ebNn++vUIFaD1/hyWhqivf22H/otXy6ycqXI0UcnLhMOUzUQ02XPpTsP/eLLaJcQYag+puef98Nj  
fWt2eD0aiH31lb+eqVNF9tgj8XYNIg85ROTWW/11aMjXvHliWFeunL/Phg3z1xOnoW54XVq9egU3  
pqFH62qg+v77/ukMwvfxgFQDUw1wBwwQGTHCPy2ChpxzZTQo1Lekd+nif00fe8zfN+HAfKedRJ58  
0t+WuvzyvNu0SpTw50b4czpP3nnH/1rr/+tH3bcakK5bj/LZZyJNmiTeR4Psd9/1t6GBr369NUCN

367b1XPCKg1RdQ7Fw1MNbfVxP/GEyEsv+Y/3kkv8gDh+f/1DwMsv+/ePiwpTdZ0aglc/p18LnYc6  
L884w3/O+nndrgbxjz7qn0pBw1TdTnlwTpgKAAAAAMD/nfkNHXBIL1z0wQf+EZIfijyyy/+EZka  
JGo4p+GbBkvh0EhDpm++CVZg6FF+GrTFj1DU5FHvq0e8ho0e7R99qke/6lF9Gupp+BemR31GHUUY  
poGqHuUaXiYcpupz0KMkw0Hq7benHjWp9KJNerSlhpK6nD43DV0/+khk3jz/CM74OrQ6dli+yJCG  
e+HlNKj88cfgRiPd281t3oquR5+Gt7PzziJx/q36VGZ55zjh7nx2zXU1XAwHIDG6dG9GI4feWTe  
svo1jO/rNm3yPq+l4awGnd9/n/h49ejN44/3l9HnrCG3HoGr67/ySv+oZg07NczV23T/akD6++/+  
uW/jQb0Gk3p06ZdfBis2dP9rIKtzs08fPzDWi25pOKoX3tlgt1WrxCNVtTp3DIYQiApT9UhenRPx  
fRf3+ef+bfGgWR+Xhrdz5wYLGBoUjxyZ+kcGwlQAAAAAAP7vzG/ogEMaXmoQqEGQHnWpgZqGiXpE  
4eOPi/z6a2qApyFLOEzVIDN8+7HH+oFnpsLhXDbCVD2yNPmowzffDG6MoEd9xo+ujJe+7V+D1uQw  
9ZZbgjsl0aMhw8tpgBgOU7MlXZiqpzQlh6NaekRwfvT0CslHuuqRn/r10FM6hD+vRXLr/kimRwHr  
qR9OPTVvWT3HrYbp8a+jho7xt+rr/+u+vuopPySNH5Wqc0+D0zB9Phrw61ysWtVfZscd/fl1773+  
HNS39uvn49vW0iNlw6LC1OQ5FafzJLychr8aDifTfZR8ASrCVAIAAAAAA/u/Mb+iAQ/pW6XAgpEcS  
/vxzcKOh57fU84CGl9Ew9dtvgwUMDV3Db9XWALrV/+o1zg9sk9DW307uS6j57584AE/ZAvTMDXq  
aNCwgsJUfXu9HuUYfqv76af758QMn1NTj5zVIFDfph8/mlND0IYN/SMn9Zya4VMTaOm5UqNkEqbq  
29/1KGB9m7uWHm2pRwDbSBem6n7R28MX8NljdPVUDEPH+8vE6Tk/9VQF4XPC6rI6H5SGnfHPa+nR  
xosW+bdF0aOM9Zyiuqy+BV5PBaH/1rfHv/aaf35UnTN6nlk92IMDUP23ziUN7/WxTJwYrCzwww+p  
pxbQt9nHzZpHzWaHJxnEqbq6R6i6Nw/8UT/XKy6nD7Wli39xxlP/XW73bolznktwlQAAAAAAP7v  
zG/ogEOffCjy3HGJb5XWgEvf5qyBoL79Xc9zGQ6N9038vXsHKwjoW8n1QkfxECq+Hg3YtPTf8c/r  
eUj1gkpr4ZwGjvp44svGSx+HnqNz8ODMLkCl5+PUoE0Du/gy+rj1ecUfkz6/8Hkv9SJNGurpEZE  
6gJUWxKmFvYCVGHpwtQ4PWJTjwYNv90//LXQxxY/H6iWBui6//Ro0rjkC1AVFKbq0b033OAFQRq+  
nx7hGj/dgwa6Dz0kUr164jJ6yof4uVrD9K38elqI8Dr1cevXUCt+JHV4XVr69v3waQ1swlSlj0UD  
9vD3hG5X951uV78fNKjXf4fXqaevIEwFAAAAAOD/yvyGDjim54DUq+Tr0ZJ65J9eTErPI6pH42k4  
tGSJH2L+9pt/1X196//Qof4Resk0iNjg9b33RF54wX9rt65Pz72q5yHVgEwvlBUXfou/0v/X23X9  
+pZrftwaiupH3a6+1VyDUj1fpgZzept+TH6LeJge/annUdWLWOLFo/SoQn1M+vg0zNTnqcvo0ZNh  
+rz1aFq9lr0efasfdT1R9NyvekSkHqGpF1XSl3OTjzrVc7bqkZ96Tk9dVi8lpeeRtaFHzOqRpnpf  
/ajPJ/z2+/D+1FByQ059q7yetkC/Fvq4evTwjxbVfaZHBoevYB+n5y7V/Rp/3npUb/J5V8N0uzNm  
+F9nXV6fp+5fnTdx+nXTZfRCT7qMfu10//bvHyyQDz1Pql68S9cXn0saTGvwro9J55TuFz2qVEvn  
sgbhuj2l52bVsDv8ddR5XBDdrxql677S7ca/L/Rcs3qKBH1cup7494Ue7Rt1PI0AAAAAAPCvIUwF  
AAAAAAAAGAwQpglAAAAAABABghTAQAAAAAACADhKnA1k7PxXrBBEa72SKBMXAAAIHUSURBVHw7



64WL9OPZZ+ddmOn/YfFi/+Jj7dqJ1K8vsttuljvtJFK7tn/xKj3vq557FAAAAAAYCtCmAps7TRM  
Pf10P0SNV6NG/58wVS8aplfYD18hv149kauvFrnlFpFzzxWpXDNvtr33Fnn9df+xJl8sLJleyGzc  
OJFLL827f0F10UWZXQwKAAAAAAGa7Hgl+CGhmTLI+eVXtldwz+9KrlesVw/t3p14pXu9Ta9irre  
rsvHS6/Cr8vGr6lepKgb3h7fjt43OZWLPxZdX/J29d+63fA6dL3h9egy+jm9Tbe3dKnl/Pn+kZjx  
x6jriHqMSu+vy8W3oRUPEvV+8eeafH+9Pb7d+DK6LaXLNm6cGCJGham6fr3yvG5TP+r9CwowbeiV  
9S+7TKRaNf8xFCsmctddfgAap9v+4QeR447Le6wHHugfqRp/PvnRff388yKHHCSurQf0HJ0KwAA  
AAAA+BfFgo+AGw8+mBeaxat9e5GLLxYpV87//9NOExk7VmT8eD8s07eFV6mSej+tihVFDjhApGVL  
kf79g40YffqInHRS3nJ77CEyaJB/mwaYkyaJXH554rp0mb//9pfRoFHXceqpebcfe6zlkCF+WKIB  
poaFut3q1fMee3KVLOklfXnMMSJPPOE/p7AxY1KfmwaLt98ustde/v+XKSPy7rt5Qe+XX4q0aSNS  
q5ZI2bKJ991xR/+oVH0LffjzUWFq8hGd+pb7efOCG7NA39YfD1K1NEzV55FMQ9Pzz098LDoHliwJ  
FsjHnDn+/XQfVKggcscd/j7u0ME/zYHepqcVePpp/2sFAAAAAACQZbHgl+BG586JoZmWBo4a/ukR  
ifo2744dRb7+WuSgg/zbixf3Q83u3UWGDvUDSQ3HevcWadAgbz377CPy0EN+EKq++EJku+382/Sj  
vq1cQ1oN77p08bent1WqIPdRt7Fwob9MixZ+WKshoAabep+4X3/1Q8/ddxfZc08/uP322+BGY8EC  
kW7d/CA0/vi0NBwO08eTHHzq891mG//t8Lo/TjxRpG9fPwy+886851SihL+PnnrKD3eVLtekSeLb  
6rWiwlQNtuvWFTnsMP+jvgVej6hNRwNdDVw1BB4wwD8SNF76/3/9JTJ5sr+shtHhx6D78bPP/NuS  
XXJJ4rlaoBf0WIYPF9l117z76L6uUcPfj1p6XtZSpfJu32UXkbff9o+GBQAAAAAYIJY8BFwIzIM  
1QBTQzd9m3rcqFF+8KnhV3jZTOqoo/y3rKtp0/zzdcbXs/POlh9+6B/xePjh/uf0tpdf9oNT/X8N  
Rd96yw8F9YhX/ZyGdl89lhCsxukRqiNHivTq5b/FXC+kpAGoBpnly/tvPdf7h+u554I7B6LCVD3q  
VM81mvzW/p49E8NB3YaGzsk0hNTHEl5nts6ZqkHkSy/5R9qG1x+vHXbw96We6kAfW/jlWQ1TP/oo  
WFGIBrTNmyeuR4PqdGGq3kcDb/0aT5woMnWqH4KHn6N+nTXg1kA6vn0NovWoXwAAAAAAGCyIBR8B  
N5LDVL3avB5hGKZhqgZe8aMO9QjM114Lboyg5/mMv/Vez68ap/8/YYLhRfmbe+880Qeftg/4ITD  
1Ztv9gNCDTk1bNNtafCoYa6+ZV7vc8QRfmgapkeJ6sWTwm/v16MrNSzUbapffkl9+38mYeo99wQ3  
JtHQNxOapj66afBjSHTp/tHs4bXaROmpjtvqh71q0e/du3qB9UaIl9zjV/6//pW+zfe8M8/26+f  
f2oEfQu+Pgbdv888kxd2K/166ZHGZ5yR91j1tAnXXpv+CFINa4cNE/n5Z5GvvhL5/Xd/m8lmzBA5  
9NC8dWtdcUVwIwAAAAAAwJaJBR8BN5LDVD2KU48CTaZHGoyvSqRHsOp5VfWcpXoU65Qpli++mHf0  
qJYeVRq/EnyYHkm5336JR3Vq6Xk14+dR1Su8n3mm//b68DJ6GgE9mjX5KEk9H6gGsuFI9fQEvoB  
XVbfzn/00X44G14mkzD17ruDG5NooKshc3x5PR2A/vuGG/z9oRdk0nOS6ukLks+lGHwM6vlew8vs  
u292z5mqYaiGneEwUx+vBq/69dUjS/X8rxpWazAcX+bWW0VmzfLXofs0/nmtmjVF5s71Q9L77ss7  
5YHeX8+Bq0fD6kW59GugRwyH161f/1atuJo/AAAAAADImljwEXDj3nvzgrF45XdxID368PPPRa66

yg9D9dyq4ftpYKjn+7zuOv8lfyCwNmz/RAzF+tTp2CBQxd5pVXUsPUG28MFkiibynXlyL17e6  
hGv4PnreTg0B9YrOGtBqqKpXnNfSAFDD4/g5TvUo3OTg86ab/Nui6BG4GIBq4KhBcnJARAG0nttV  
L76klaae9kBDUj3SM77NuPARu1p6jIENKrNNg1ENpDW81bfc6xGq4e1qoH7CCSIPPODPBQ1Z4zT8  
DC+7/fb+W/uVHomqb+PXo4v1HKvhQFZLt6MXFdN9oeV1fO6Rh29CgAAAAAAUEix4COQfenePp7u  
Npf+X9tV/8X9of6f2w77r+4fAAAAACAAGEqAAAAAAAAGSAMBUAAAAAAAAMkCYCgAAAAAAAAGZ  
IEwFAAAAAAAAgAwQpgIAAAAAABABghTAQAAAAAACADhKkAAAAAAAkAHCVAAAAAAAAADIAGEq  
AAAAAAAAGSAMBUAAAAAAAAMkCYiqzYZEYmMI0AAAAA6dn0VfRgsMX8gkvML7jE/IJrhKnImrFm  
VDQjls9obQYAAACyZ7EZdcwI91zhcZIZQGGtNGN/M8JzKjyONWOjGUBhrDXjMDPCcyo86puhywCF  
oa9NDcwlz6nwOMAMfY0DCoMwFVkwYzqZoRfoMKjoxkAAADIniVmaOAQ7rnCo5kZQGfP0HCMGeE5  
FR6nm0GYisLSolShGeE5FR6NzCBMRWHpa1NjM8JzKjw0aCVMRWERpiJrCFPhyqZNBb/1lpNlgCjM  
L7jE/IJrhKlwiTAVLhGmwiXCVLhEmIotor8Arlu3TmSNyOh1o6X6pvzD1A4bOuhPTHMn/77Zoo+B  
X0SLps3zy5gwYYLUqFFDYjEzm0JVqVILGTrokLfM2rVrmQvIWHh+TZ06VWrWrJkyv8qVKyd9+/b1  
lmF+wVZ8fs2aNUv23nvvlPIVsmRJ+fHHH71lmF+wFe+/5q+dL4dtv9tsk03NpV1a82y5F2wEJ9f  
S9YukaM2HRWaUYnjtl2nyeq1q5lfsBKfX8vXLpcTNp4QmlGJ48SNJ3rLML9gY/369d780temUzee  
GppRiePojUfL4rWlMv8oFMJUzCz+S96iRYvkiuvlHPPPVdOOeWUvF8MdzE1LSbFzPBfnvKG97IH  
zL/Mcicdf5Kcd9550rJIS5k3b563TrVm0UKZPGKE9P/2e/np44/I2/fflx8/+khG9P1DFkyeYn7T  
ND90N5hXuo3mcZgaPeAv6dbrHTmz56vy5m9/yPr5C83nk14J9SGbH9brly6VZXNmy/zJk2XWuLEy  
9e/RMnHoSBnxRz8Z9MOP0u/rr73tjv9T3PbGfK2a7b+ZuuvA/+K+PxasWKftG3bVs455xpx3Lhx  
3vwKqlgxM5ci/h2vhg0byvnnny+XXXaZF2DEEVlktvjXf82aNdK+fXtvp1xxhISvHjxhPIT0Pw6  
4YQT5IILLpCLL77YC2DjmF+5Lf7137Bhg3Ts2FGaNWsmTZo08cLS8PwpaH4de+yxcuGFF8pFF10k  
EydO9NapmF+5Lfz1v+mmm+Tss8+Wpk2bSunSpf25U9JUfzOn8uu/3jH/MssdddBR3vzSn5GjR48O  
1sj8ynXhr//tt9/uzS19DdM/Jm5+ffotzfz62PzLLHPEfkd4r136+8GwYcOCNTK/cl3463/33XfL  
WWed5c2RChUq5M2v79LMr6/Mv8wyh9U7TJo3b+71b/GDKBTzK7eFv/4PPvigN7/OZ5webLN5fn2e  
Zn79ZP5Iijm41sHe/NKfr3/88UewRuYX0iNMRcb++eefhPA05RfBWqamm8pvPGUqft/Q/UofeqTU  
7vqMVPv2e6k8foLssHS57Lhiley0fKXssGipbDt2gmzz489StvfbUvaNN6Xcm29J2bffkUo//Sw7  
mGatxtC/ZlcPPpQq3Z+RbU1t98yzsl2P52S7556Xbz/tlds+08P7/22ff1G2fe1N2enrb2Xnfvl  
p1GjZYfZc2UHs60d162XHdeskWqLI8i2/0yRSr/1lUrvvCc7Pd1DDn+8q1z01DNy/5tvS98hf8mS  
RYuCPYJsmjJlihc+bJ4jEUFDPHw+79FHHy1jx44NtoJcNX36dO+POFFzxLbC9z3ssMNk1KhRNFs5  
bubMmXLJJZdEzhHbCt93//33l+HDhzO/ctycOXPkiiuiJwjXIU2NdBUfuMjUxH3rVu3rgwdOIQ2

Jv8hGjIFD2xo165d5BzxqrypqbG1+aKm4q6b577rmnDBgwgPmV4xYsWCDXX3/95nmRMr9Km/rR  
VH7jB1O6TPg+pnbddVcv9NI/YiJ36UFet9566+Z5kTK/9LVJA/n8xu+m9DUu6f7VqlWtX375hfmF  
tAhTUaDbbrtNKlasmPeilF9ZhKmVS5eS2nXqyO6dHpGdP/ICdvj8G9nxq+9lpx9+kV0HDJHdpk6X  
3efNlxozZ8muy1dlzf4D5fBrr5Nm11wjLW68US6+4QY5r93VclqrK+XltldLvfc/ID0nTJJaw4ZL  
vR9/kv3ef18OevNNOezll+Wo556XY7t3l0MeeUT2vf9+OarrU3LUCy/KwW+/LbXMD+Hdpk6T6suW  
y86z58gui5flbuY5725qD1N7BqX/v7t5MT3w5bflwV7vZftMBTntnnvuSfzroaMqX768d0Qickun  
Tp2kcuXKkXMim1W2bFlp3bo1TVeO6dKli1StWjVyTmSzypQply1atPBOBYDc8fTTT8t2220XOScS  
yiJMjSo9wlWPyFm1aWwZeSCF154wQsMouZEQlmeEqVFVqlQp70jEZcuWBVtGLnj11Vdlxx13jJwT  
CVXIMDVe+g4QPRpRQzXkjrfeekt22WWXyDmRUJZhanLp/DrttNMS3k0LxBGmllL4KJhWrVpFvrik  
VEFh6pMxKRYrlaV3ri3bnHOxbP94d9nhnQ9k5z8HSI3JU2XXUeNk1yHDZZcBQ2QX87ldhv8tNSb8  
l/u89LKcfF1Hade2rVzTpo33FnAt/Sv6NeZj+9at5bguj0uNX3+TvdavI73MY9bSEFQD0XAlf06D  
04O++lrOOKOJXNnsHGLzzrlyadOz5ULz75M7XCv7vfKq1Jw9x7+v2SflzQt3qw8+MP+HjBWQPF99  
2dXR88lBnXPKOcFWUWQUML9uaH1D5FwxUWccf4YIWWrRUsD8uuPaOyLngos66ciTvPN/FYi/9hUZ  
DzzwQORcSKktDFO1GjRoEGwVuUL/GBQ1F1JqC8NUrr16/vnMETO0D8GRc2FINrCMFXrwAMP5I9B  
Oeall16KnAsptYVhqla9evVk6dKI3nZ5pxDCCFORr59//jnxfEkFVUFh6qMappaTym2vk2q93vSO  
RN1t/gl/4Ny4yQsr40eCanlHiC5fLic88KBcfGYTaRMEqfoxXu2uukraXnWINHjiSdn1976y57p1  
CesoqGqYOuiabb6XJBRfKFS2v8AJbb90a2F5+ubRo2VL2f+8D2XnWHPMYN8ou/QZlpauvkellSpjn  
EvPOt0hFVMxUCVNfmCpgFntUTGlbzfz4F0q3Fd522tHWID6PYqainiP1/yv9umh9aKqA8Z+dXx1N  
xZ9H1HOk/n8V/7q8ZaqA8W/Or9gm8zMn03GrKebXVI8pb1fMr7lQpmpFPQaq6FbG8ysLYapW1GOg  
im5IPL+yEKZqRT0GquhWxvMrC2FqvB5++GEBwghTka+ffvrJe2th1ItJZBUUpnbWMLWCVL3zAak+  
fsLmo0fzKw1TNWw9+a675YrGjf/dMNxu1VdeKa2vvloOffoZqT7kL9lj0ybZZfBgqdy+g8QyfQHP  
5Spm6nNTW+toYyrqeVH/nfrA1NY6rjMV9Zyo/071NrW1jltMRT0nqmhWlsJUioqsLIWpFBVZWQpT  
KSqyshim6qnDgDDCVOSrX79+Urt27cz/8lNQmNpFw9QyUvGsC6TKDbdL1TselB1ef0d26T9lasye  
l7stXSa7r1xpapXssWHD5vOUHvX4E3LuOed6AWfblOiMlx+mXiUNnnhiC8Lub6TJ+Rdlq5ZXyNV6  
RKqu19QVFzWXps0vIr37/iG7muX0yNSdzb+3ad0m+vltqfWZqa11XGUq6jIR/51639TWOq41FfWc  
qp9OvWlqax03m4p6TITRLMJUymURplluizCVcllZCFM1C9HrLzz77LMChBGmokB60YuoF5aUKihM  
7WYqVkyKxcpKlXr7yV4tWslOL/SU7Xq/K9s/11N2+OAT2WXIX7LbnHmy6/CRsvPv/WTXPwZKjZ9/  
k8NffEmuuOxSudo8Fg08r27dWq4x1b5lC2l74YVydLenZBdzHy+ANeUd1RqUBqbVg4/6/+F/67J1  
hw6VQ+64S4674GI57so2cmTnx+Tg7s/l3l98KXtNn54X0G7cKDu+8aZc9HR3GTx4slz86y/vlXVR

A4OPS0zIM0aYMdKMSx86W2K1zdw4xNShDsvMz+NvPN7b9t9m5D2SfMZUUwNMJT836v9f8fm12FQ+  
Y7gZ+nW+8LEL/fl1cNJ8yHaZ+XVUh6Nk4MaBMsqM8GOJHNNMMb/+mzUoqlWm8hnDzBhtxqXdLpVY  
HfP1/xfm12FtD5N+a/t52w0/lsgxwxTza6utv0x/MWbMGLnyyitTe62oykKYesABB0jfvn1l9Ggz  
vyleE1V0SufX2LFj5eqrMzxnfRbC1L333tu7MrbO66jHRBWdGmp+rxo3bpx07Ngxci6kVBbC1Fq1  
askPP/zgzeuox0QVndL5NX78+lQr+KetLISpdevWlfnz5wuQjDAVBerRo4fsv//+m19Q8j1StaAw  
NXQ1/5KmKlauLDs2v0R2+fhT2WnZctlp0ITZ8dc/ZMePP5Odv3BvwjVsBGy88jR5vN/yg69ess+  
z70o9Z9/QQ7s9brs2+d9qfXxJ1L9g49kry+/kv0++0wO69ITjnnnySTnhwYfkpDvukFNvuknOMD/M  
m7bvIMdd1FwOPaupnHH1NXLmjTfKcQ88ILW//VZ2W7LEC1V3GT5Cdp04KTGI3bRJdlq6VKrNmCk1  
zOM456U35K0PPuP6Hln04osvykEHHbR5bmR8JHQGFV6Xnjz88ccfD7aKXNGrVy859NBDI+fElIZ4  
XXXq1JFHHnlENm7cGGwZuaB3797ehVWi5sSWVnhdNWvW9C5GtG7dumDLyAXvvfeeHHXUUZFzIqEK  
EaaG17XnnnvK3XfflWvWZHKFMxQVH3/8sRxzzDEJ8yKyChGmhufX7rvvLrfdpusXLky2DJyweef  
fy7HHXdcwT8XCxGmhtdZvXp1ucn8vhe/QBByw7fmd/gTTzxRSpQokTA3UqoQYWp4fu2yyy5y3XXX  
yaJFi4ItA3kIU5ExDQkuueSShBebhLIU8N19LHHyWt9PpAOT3aT4+99UPZ++Amp3vKx2bFzZ9n1  
4UfkkPsflrMf7CLXPd5dbnjyWbmh27NyY7ce0vSBp2Tvzs9JtXf6yKnPvyI3dX5abu7aQ27p9pyp  
5+WWp16Qm57sldd17ipX3f+gNL/jTml66+1y2l13SaNOXeW4x1+Vwx7rlfs9+qTU6/yoWdfjqhZ  
x8mPPCsXPfiYXN+5i3R9qad8/dNPMmHKVFm1dq1sMPtgE2GJMxkfhWNRTZo04RdEeK655prlObll  
dcopp/ALLjw33nhj5BzZkjr++OP5BRGeO+64Y/O8SAknCnlk6pFHHskviPDoH2vi8yJlhfUiTNU6  
5JBDZO7cucEWkMsee+xyzfMiZX4V8sjUfffdV2bOnBlSAbnsqae2jwvUuZXIY9M1aOdp06dGmwB  
iEaYCmsaHOiLVpcuXeTaa6/Ne+HZwdRU8yJmhvIXwvA+d7/5l1numtbXePd98sknZdmyZcFa7a1d  
uULGjRsrH377nYwcO1Y2csTOVm3TJv94Xw0+u3fv7jVeN9xwQ978CirIh2TS5/RcunoEqtbixYu9  
dar4+pGb4l//DRs2eEfbP/roo3LLLbek/EW7oPnVqIUreeKJJ7z5GX7LD/Mrt4W//i+88IJ07txZ  
br/99pSLOBY0vy6//HJvfun9Z8+eHayR+ZXrwl//nj17ekfB33XXXVK+fPm8udTPzKX8+i89/69Z  
pvnZzb3eS+8/ffr0YI3Mr1wX/vrruzn0itX33nuvVKpUKW9+/ZJmfun5y80y559+vnTt2tW7SMvk  
yZODNTK/cl346//GG2948+v++++XbbfdNm9+fZtmfunFbM0yzU5u5s2vhx56SCZMmBCskfmV68Jf  
/7fftt7/dE5ssMOO+TNR8/SzK/vzb/MMmcef6Y3vx588EHvdCRxC+kQ5iKQkt+cZlpRg0zYpvy  
XqQ2D/O5W8yl44UJBumelxpc6V8JN/9gTCoN9uOYXyhi8hzRo//22WefyLml1bp162BJ5hcKljxH  
Vq1alXA6k+S67LLLgiWZxYhY8hxZb8YRZuTXf51vRhzzCwWJmiPHmpHf/GpiRhzzCwWJmiMnm5Hf  
/DrFjDjmFwoSNUfONCO/+XW8GXHML9giTEXWTDCjuhmhl6iE0dGM/7qCXkI5if3/Obda7LXXXpFB  
hFa7du2CJQF7ehSgnlc3am5ptWzZMlgSsLdgwQLvAj9Rc0urefPmwZKAvm1DfDzKbl0cwMoLBW

mnGMGeE5FR6nm7HRDKAw1prR0IzwnAqPRmboMkBh6GtTYzPCcyo8Gpihr3FAYRCmImuKQpiqlm3Y  
IPsMHy6xX36RWN++3sfTxo4NbnVj7aZNMmDFCrll2jTzc9gwKTZggMR+/11if/4pl0+aJL8Fp0MY  
sXKI/LBkifyydKn3cdSqVd7ncwFhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFORNUUITF2+YYMc  
9ffffpDav7/38fzx44Nb3Zi1bp0cPHKkF57G+vWT2B9/yDIImmyNWzKZF69d7yyzduDHLcekyuYlw  
FS4RpsllwIS4RJgKlwhT4RJhKlwiTIVLhKnlmgmLJkj156tL7EHZ0tQIVI+ZMp/r+GPRDFM17Pxj  
+XLpbX5Zfm7OHOk6e7Z0M9Vz3jz5fPHifl8e1aNRR5vbPI20SG6fNk22HTzYD1N1m+bj0eYxPGPW  
12v+fHnFrOuBGTNk97/+SlhGj6B9fNYs+dJsZ0IRv2I9YSpclkyFS4SpcGnJmiVS/436EnvAzKdw  
/6VIPtfsY8JUFN7K9SvlmLeOyXd+nf7B6bJxE2EqCmfthrXS8N2G+c6vRu828pYBCkNfmqx/3zjf  
+dXgrQaych1hKqgHMBVZo1dWrF69euQvilodOxatMFXfen/VP/9ITN+S36+fIDQfyw4cKOVm6cfS  
+nm9v6lqQ4ZI9zlzZH3oxNa6nXfNL9iNx47dvFy4ipmqOGiQ7D9ihNQZPlzKmHXq5xKWM9stbz5/  
8cSJ8sXixUX6nK6EqXCJMBUuEabCpSVLlkj9+vUj55ZW52aEqSi8lStXyjHHHBM5t7ROP/102biR  
MBWFs3btWmnYsGHk3NJq1KiRtwxQGpra1Lhx48i5pdWgQQPvNQ4oDMJUZE2uhKLZe3fLI0q9UPL  
7Dp0qBw9apS33CUTJ8oFZI+cOnas7KZHkgahpx55eqm5Lf62/bBF5nN1hw1L2Ob1U6YEt/rWmWpg  
thFeRreTKwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmIqsyZUwdfzq1XLL1KkJb7nXMPXAESPk  
oKQ6WGvkSK/2Hj5cmpt9tCAiTJ1pmoTaSWFq+8mTg1t9S8wPAw1sw8ucm+b0A0UNYSpclkyFS4Sp  
clkwFS4RpsllwIS4RJgKlwhTkTW5EqZOMz/Q9byodfWK/7pMv35y9/Tp3ueTaXA6fNUqGbRihVfj  
1qxJekt/XGHD1PMIUzcXYSq2BGEqXCJMhUuEqXCJMBUuEabCjCUuESYiqzJpXOmryYvzDdPmybb  
DRniH53ar593XtNbzeenTNHusyaJReb/VFVLyr1++/eMheZ//9p2TJZU8gwdbnZ5nGjR3tX+ttf  
6TrmW0+PHOmG1+kZpcxBsNwIS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmImty7QJUSo88/XjR  
Imk1aZL3Nv7KgwZJ6YEDpcrgwbLfBFy5T//yHsLF3rnWU0nkzBV29R/1qyRp+fMkcPM49t2yBAp  
NmCAdwGq1mY7Py9bxgWogEliTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFORNUUHTP1/hJFFOQDNJsJU  
uESYCpciU+ESYSpclkyFS4SpclkwFS4RpiJrisqRqfhvIkyFS4SpclkwFS4RpsllwIS4RJgKlwhT  
4RJhKrKGMBUuEabCjCUuESYCpciU+ESYSpclkyFS4SpclkwFVIDmAqXCFPhEmEqXCJMhUuEqXCJ  
MBUuEabCjCUuESYiqwhTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPhEmEqXCJMRdYQpsllwIS4RJgK  
lwhT4RJhKlwiTIVLhKlwiTAVLhGmImSIU+ESYSpclkyFS4SpclkwFS4RpsllwIS4RJgKlwhTkTWE  
qXCJMBUuEabCjCUuESYCpciU+ESYSpclkyFS4SpyBrCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBUu  
EabCjCUZA1hKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPhEmEqsoYwFS4RpsllwIS4RJgKlwhT  
4RJhKlwiTIVLhKlwiTAVWUOYCpciU+ESYSpclkyFS4SpclkwFS4RpsllwIS4RjiKrCFMhUuEqXCJ  
MBUuEabCjCUuESYCpciU+ESYSpclx1hCmwiXCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBUuEaYi

awhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFORNYSpcikwFS4RpsllwIS4RJgKlwhT4RJh  
KlwiTIVLhKnIGsJUuESYCpciU+ESYSpclkyFS4SpclkwFS4RpsllwIRkDWEqXCJMhUuEqXCJMBUu  
EabCJcJUuESYCpciU+ESYSqyhjAVLhGmwiXCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBVZQ5gKlwhT  
4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmIqslUyFS4SpclkwFS4RpsllwIS4RJgKlwhT4RJhKlwi  
TEXWEKbCJcJUuESYCpciU+ESYSpclkyFS4SpclkwFS4RpiJrCFPhEmEqXCJMhUuEqXCJMBUuEabC  
JcJUuESYCpciU5E1hKlwiTAVLhGmwiXCVLhEmIps2LBxoyxZulQWzJsn2bNliUzZ8IG8//zJ02S  
k+of7s2IKsWKmSoulc3HyrFi3ueaNmokC2fMIFVm2cULF8rKVatkw4YNwVqB9AhT4RJhKlwiTEXW  
EKbCJcJUuESYCpciU+ESYSq21Nhp0+T9l1+Rn/Y7RIYWLy3jSpaSicVKyJRixWWGqQWlysJSMmVI  
WWnzsXRZWWz+vcDUXFPTzW3ji5eQiSVKyfeVt5MHDj1KhN3/U7BmID3CVLhEmAqXCFORNYSpcikw  
FS4RpsllwIS4RJiKlFXThx/LA4ceKTMqVRYpWULWly0rK8qUk6mVqkj/nXaVD3evKS/W3ke67XuA  
PLX3/tKrVj35pvruMmTbajKzQkXZaJaXsmVkabESMqxYWfnxtjtl8Pjx3tGuQDqEqXCJMBUuEaYi  
awhT4RJhKlwiTIVLhKlwiTAVhbVx0YaZv3KF/HrvA/JBrJQsKFIKpFRJWVGhgjxywKFyadNzpXWb  
NtK6bVvvY5ug9N9XtW4tLU3vdUfjM+TTKtVkdplyliVLysZSpaV3g+PkuWd6yJo1a4ItAdEIU+ES  
YSpclxkF1hCmwiXCVLhEmAqXCFPhEmEqCmv12rXyw4D+8lvLVjlrVILWISol60uUkvEVq8j1jc+Q  
c9q2lXZXXy1ttUyflVxtOnSQNhdeLB33rS9vb7urTCpdXjaWKSPv77Sr9Lz6Wlm7alWwJSAaYSp  
lkyFS4SpyBrCVLhEmAqXCFPhEmEqXCJMRWGtXL1aPvn5Z/m1+cWyLFZcNpQqJWw0HKjIK8u9emTq  
CY2kdcOGcpXWiSemVMuTTpLrjzhaXqpeSyZvU0XE3H9TmTLSZ9vt5JUrrpS1hBQoAGEqXCJMhUuE  
qcgawlS4RJgKlwhT4RJhKlwiTEVhrVm3TgaMGC5/XdVWJhYrK+/vUUtter7uPfldrDVIWoYKlVu2/  
dGmRUlp6CoCIKlISpExpef6Ag+XRgw+T6RW2kc9q7CXvX3+zrFu9OtgSEI0wFS4RpsllwIRkDWEq  
XCJMhUuEqXCJMBUuEaaisNZt2CBjpk+XodfdKD+WKC/tj24gHU44UXrXriPzKlXygtQNZcvK+jl  
8i0NVFeVLSedjterjitsfSuUUMeO6S+vHz/Q7KWc6aiAISpcikwFS4RpiJrCFPhEmEqXCJMhUuE  
qXCJMBWFpdfaX7JunQy95wF5pXg5uaLB8dLhpJPkbfPzCH4GYeoGU5tKlZSVZcvlow1PklZnniUv  
19hNupx0irze4wXCVBSIMBUuEabCJcJUZA1hKlwiTIVLhKlwiTAVLhGmYkuNfrSL9CxRWjocd5zc  
dcLx8unuu8vCihUzPj1Q9ky8vRxx0vHU06VD3fZRV5q2lQ+eLO3rF+3LtgCEI0wFS4RpsllwIRk  
DWEqXCJMhUuEqXCJMBUuEaZiS415rIu8XKK0tD/2ZHnkyGOIf5WqsqxcOZEyZQoOU0uXko1m2TcP  
OUQeOviY+WTnPeWlpufKB2++SZiKAhGmwiXCVLhEmIqslUyFS4SpclkwFS4RpsllwIQUll7Nf8SM  
6fLNDTfKSyXlyJUHHilv1q4nS4qXkrGIS8vgsmVlqFaZMjIknxpZqrSMN/f9fpdd5YM995LPd6kh  
TzY+U7p37Sajx42VlbzVH2kQpsllwIS4RJiKrCFMhUuEqXCJMBUuEabCJcJUFMa8pUvltY+/lh77  
HSEfxcrJ8DIV5YNK28o9tfaWixo1lhZXtZYrTe/eqkMHuaJ9+8hqZepSc3vztm2ly/6HyvvbbCcf

br+j9CITSZ6rtpvcfdZ5MujX34MtAqklU+ESYSpclxf1kwYN0GqV0oTprYhTEXhEabCpdkzZ0u9  
XdKEqRcQpqlWfSxbIAfsmSZMPYswFYW3ZPESeqV8vTZh6CmEqom0ytXT9ehnwYCd5N1Zc+sZKSs8D  
D5FL27SRK01f1a5tW2lrqo35/3zLLNfGLNO68ZnyUs06sqZMGfnixlby+ksvE1KgQISpcMkLU49N  
E6YeSJiKwiNMRXaYbmyCGdXNMC9NkaOjGV7XBhQCYSqcMa9Ls82oZ4aZTZGjpRnxZQErZs4sMOMA  
M8JzKjyamxFfFrBi5swSM+qbEZ5T4dHMjPiyQDK9ov9fjz0ur8VKyEcIKskT+x0ml1x6ubTSkLT9  
NdLWfNRANao0SG11zdVymfn3+Sc3kSd3rydrSpeTL09tLG++2VvWcc5UFIAwFc6Yn3kbzWhshpIN  
kaOBGSvN4OcJCoMwFVmTUZgKFBjHklzKOEwFCiHjMBUohlzDVCDJmnVrpe+ww+SX1m1lcKyMfFZ8  
exlSaluZW24beXen3eWO/Q+Vlg1PkfOaNJOm518oTS++WJo0by5Nzz1PLjztDGI7zAnyRJ395Mdt  
dpAvymwv/UtWkXWIK8ibu9WU7tfflmtXrw62BEQjTIVLGyepQCEQpiJrCFPhEmEqXCJMhUuEqXCJ  
MBWFtXzISun14UfyY5OmsiJWUn4tUUUGI95W5peplr+W3kH+LIVFpFQpkZILZWOJErIhqE2m9HNS  
spTMLVVRPi+9k7xRahdzn+28q/9/WLK09Gx2vqxdQUiB9AhT4RjHklwiTMUW2bBhg8yYMUMWTVok  
v878VXbesPPmF6fk0XJJS1k4baHI+uDOQAH0PDc6vxYuXCh//vmn7LbbbpGNltbFF18s8+bNC+4J  
FGzTpk0yc+ZMWTBpgQyaMUhqrqsZesVKHOcuO1fmTJ1juv7gzkAGZs2aJfMnzZdh04dJvXX5h/VN  
ljeRmVNninDRa1jQC+fNmzRP/p7+txy49sDQjEocp6w4RaZNnSbCQYJIsnrtWvmhf3/p2+IKmR0r  
KYtLlpNxZarIN6WrSZ/SO8uAMtvL+rLIZEOZMrl+VJvKIPY+/l6qqvxcanuZW6ayvFu6uvQptbO3  
7Kc77SqvXn2trF21KtgSkGjOnDkyd+5cmThxohx99NGRvb2W3qYXOQZs6NyaM2mOTJ46WU5YdULO  
J2LiOGz1YTJm6hiRfEdAqUEqciYBg9Km/cTTzxR9t13X9l7773zfuDtbmqa/8IUObqYMsvVq1nP  
u+9xxx0n06aZ5j4QXz9yU/zrr8HpySefLPvss483TzbPrwyrTp06st9++3nNlzZoccyv3Bb/+i9b  
tkzOOOMM77VL50mxYsX8ubOdqdGm8hsvmTLL1d6ttuy///5y+OGHy+jRo711KuZXbot//VevXi1N  
mzaVevXqefOkRlKS/vyqZGqYN5OixxumzHK1dq3lXfX/sMMOk2HDhnnrVMYv3Bb/+usfsM877zyp  
W7euN09Klizpz69ypgZ4Myl69DFllttrp73kwAMPLIIMPPIgGDBjgrVMxv9C/9zvy2j4HyaqKFWRa  
iQryfqmdZXbpbWRi6cryeakdZW6ZCiJBgKq10dSK0uXI01Lby1Nm2TdL7SQjS1WRxbEyMq9EeRlw  
wy0ycORib85ukd69xTSEYiaumBdWMb88iPz9d3BjFnXpIlK7tphvDv/j2Wdr2hfcWAjpvqdy7Pst  
/PrSokULr08/6KCDplyZQ97rV4a1++67e69d+rP1xx9/DNbl61euC3/99YJ4tc33r86TcuXK5c2f  
L03IN34xVSIImNarU8O6nv3t+9dVXwRqZX0iPMBUZWb9+vXz33XdSv37e1Wl3hxDxqmVquqn8xlOm  
lu6rP1C/+eYbTIKfw/Ql1J9++kmOOeaYyDliW+H7aij7xRdfyJo1HPKVq7QR+vXXX+WEE06InCNe  
7WRqjKn8Ri9TEffVpu3TTz+VVRx9k9P69u0rp5xySuQc8UrD+uGm8htvm4q47x577CEffvghV5rN  
QeFf4Pr16+f9EShqjnhV2dRAU/mNj0xF3HfXXXeVPn36yPLly4MtIVeE55cG6+dedJE3Jw4tXlqu  
rLKjdNphD3m1fHV5s9Su8napXWREqSqytHR5kTJITJWS8aUryfPmtr9KbS8Dyu4od21TU67ftro0

q1xFtitRylvXLmZ+vfX227J06dJgS4UIIj77TOT000XOOkvktNNElrlEZPz44MYsuvZa81ux+bU4  
XrVqiUyfHtxYSL16Ja5T64MPghuLtvDXeciQIXLhhReap5/6GmRbet/4/atWrSpvvPGGLF68ONhS  
leYXtkrhr7P+4fnSSy9NmCPxf3tV3NRXpvlbv5sqbyrp/hUrVpRXXnnFO8gnjvmFZOZVHUivdevW  
eS9I6coiTM2vLr/88mCryBxt27ePnAsuSps55Jabbropci6klEWYml+dffbZW34UDrYqd955Z+Rc  
SCmLMDW/Ov300/mjUI558MEHI+dCSImEqfmVnpeQ0D63dOnSJXluxloV846qr1SipBxesqxcXbKi  
3FuqsvQoXVn6lN5G3itVSZ4qVVXaldpWzilZQQ4qUUrKmGWLFS+eb1DWoEEDWbJkSbBlCzNnivz6  
q8jvv/sf+/UTcRH+uwhT+/RJXKfW118HNxZ9Tz31lHnKqXPBRem7hRYsWBBsGbngRdeiJwLKWUZ  
pkaVHvil78wFkplXdSBV+C8vV111VeQLS0plIUy9RP/ijKKlgD/iXXPxNZFzwUWd1+i8YKsF4A+P  
Rca/GaaeeeaZwVaRK+66667luZBSWQhT9fQnyCOPPFRQ5FxlqSyEqccee2ywVRQZBfQyT9z9RORc  
cFGH1z5cZJG/3U3D/xlpXToxZNQJT59+WmTfff3/1yNg9a3c992XuJxe+EoD1bBFZsUffSRyxRUi  
deqIFCuWt7xePKtVK9NsXiNSoYJI8eJ5t7/ySrAClzlMrVZN5P77RfQgj6pVE2/T+1evLnLGGSLv  
vCMyY0awEkPD3xde8LcVXj70mDbpvzt21CurBnfaShUwv5596FnzdKPNQ7broN0PknVzM3iHI/19  
kfHyyy9HzoWUykKYqqcGi78DjaNTEWZe1YFo33///eYXkfz+0pxQWximZrQNauuqT0xIMlqZEf5/  
F8N6Gy1MRT0naqurjF5btjBM5fUrduyr/0WhqnMr9ytjL72WximMr+KYGnPnW5s8j/+2/1XN3IN  
ZORMke13yAsb41WypEilSn7QqqHon3+KdOqUulzeHj/fr74TRANXPRAjfs224iceKLITz/5y+jR  
ZNdfL7Ljjokhq9ZLL/nLqOQwtVw5fz36Vv34aQqGDBG58UaR7bbLW07D2Vtv9W9P9uGHievU+vbb  
4EaRs84+23zK7JWtsR4zlW78n+ZXfLtpx6Omop4TVTRC8PU8M/He+65J/juBXzmVR2IpuewtDo5  
eBaOTKWKUBUz9bmprXW0MRX1vKiiWVvk4MpWi8q0sHJlKUfIWfFo5MpYpY9TD1HxzPytsil2ekhqkt  
WpjPjxRJPpVJ166Jy4XD1NWrRZo2Tbxzd6s6b55/e9j776ceDZsuTM3vbf5z54rsvXfism3bBjcm  
eeutxOW0Pv00uFHkpgsuMJ8ye2VrrK6mttbxpKmo50QVzcrCkanx6qR/3AFCzKs6EE1P6KxvK6xk  
GpeoF5SU4shUKrk+NZXB+Nf/cp3JuMJU1HOitrrK6LWFI1Mpl8WRqZTL4shUKrm6m/qPJHD/9ZS8  
EX1kan4hRUfhql6cKny7vvU+dMGrzfRcpXq0aXjZwoSpeoX/LQIT9YJagWbNmplPmb2yNVYXU/+R  
Yd3f61G1Uc+JKpqVhTC1dOnScsABB8jbb78dfPcCPvOqDqSnF4WKemFJqSwcmdq0aVNztGiRd2XZ  
+fPnU1trzQs+rjWVz1hqxjlzLr3+UomVNF9/DRu2d1glYnLaZad5215hRt4jyWcsNzXXVPJzo7aK  
OgtldrFixlvNzPmfhyNSGDRua37PmeBdxiXpMVNEpvXqwnj/rmmsyPOdzFo5M1Qu4zJxg/mVA6V9  
0OrVq+XGG2+MnAsplYUjU+vXry9Tp0715nXUY6K2ktL+S2u1qXzGljPWmHHhk3dlrKz5+lc1FdU3  
ZatMj3dAowNkwtIJ/i8WIOeJbF8tMWR86CH/tmTpwtR16/zznurb8eO3V64scuqplr/84i+j51O9  
916RmjX9t+SH1/X/CFNffz24UWSp+V1n7ty50V/H/2rF59cqU/mMhWasNeP+Z++XWDnz9a+SNB+y  
XWZ+7X3c3j2wVhvxocfS+TQxx5/HIHPkfpPI15obJ353u/cubP5lJf/4lqC2FqnTp1uAAVlplX



dSC9Tz/9VM4666zNLyJ5HsFQIDA1vC69UvGHen4h5JQvv/xSzjnnnMg5saUVXtepprnuo1dWRU75  
9ttv5fzzz4+cEwViDA1vC69Erb+xXrjxo3BlpELfvjhB2nevHnknEioQoSp4XUdf/zx8sYbb8j6  
9euDLSMX/PLLL3LZZZdFzomEKkSYGI7XMcccl6+++qr3CypyR9++faVly5ZSvHjxIDmxxpRve1xFH  
HCEvvfSS9wcCz/Dh/gWmzG2b6667/NuSde6cuJyWnks1bPRokdtvFznySD9sDS+rF4LSUwjoRaj0  
4lXh28JhauvWibfpOVajLhA1a5Z/8anwsnqRqij6PNu1E6ldO/F8rfqY7rxTZPz4YMGiqX///t4f  
tEuWLGmedmh+IhJiXlpGrVqnJkpQZydekb5YaSt8sLJV6RkcXHyMTik6V/iUHyRom35fGSz8q1  
pW6RhmVOkRrb7ialtjXrOnAstL5DDz1UnnvuOe8PjcgdgwcPNt9e7Qo+JWEhwtTw65deyb979+7e  
gV5AMvOKDmTuuuuuS3ixSahCHpmqL4SAuvXWWyPnyJaU/qIAKD1xfNQc8aqQR6ZqkAaox9+OHKO  
eFXII1P1D02AeuKJNFdeL+SRqWfoW6IB49Ins3/I9ZNOOsntH4D0j5f//ONfGGRQIJGhQ6OPJtU/  
Erz5ZuI5U/Uo1VdfDRaAay/OfMGbE7vF9pAryl0pr1V5Q54u3UO+in0vS2lr5Mfiv8nvxfrLBvOl  
2WRqo6m1sQ3ya7E/5b1iH3u3f17ma3m9wpvSqXxn2TNWUw458rC8gB457U3z/Z3vH4MKeWTq4Ycf  
ToCKApmXKsDOmjVr5PPPP/eOKHzyySfzXnj0rRZTY5HnrvE+d4/5l1nusYcek6+++ko+++wz7+1k  
QJg23jq3tPQvgfEjJulV9cMy/LmHHnplvv76a/nkk0+8t3kDYXrkqM6PL774Qp5//nnvPEje3Clj  
6u80r1/dzL/Mcvfdp93/48//liWRp2XDTlp06ZNwb/8o6H1Z6QeiVW+fHl/fmkzPzTN/Hre/Mss  
d+cNd8o333wjH330kXcqAUCF59f333/v9U+9evWSypUr+/NL68808+t18y+zzE3X3OTNrw8++MB7  
qySgwwNLLz6r70jTI+G33377vPlqqD+Sw+4iM8vfft6XHj9WaVX8//hB5Hzz88LSUuWFDnoIJHG  
jfWkpP5b/vfcm/Go1N12E9NgikyZEqlrvUf2F+6tu0mU6vNkMWx5TKo+DD5usQP8maId6VX8d4y  
tdhML5VYH5SGqfpxTLEJZtm/ZGLxqfJ18R/lx+K/e8sNiA2W9475QMYOGysrNtDr57Lw68tvv/3m  
9efvvfee7L777t7rklefpvn5+K35l1mm1QWtvP5N38Wop1SKc/b6hSLBvBwBhZP84jLPjN3MiG3K  
e5HaPMznbcjjhcmFCR5jug5MGvVqpX3gzGprr/++mBJ5hcKljxHVpuxrxn5vX61NSOO+YWCJM+R  
DWYcbEZ+86ulGXHMLxQkao4caUZ+8+tCM+KYXyhl1BzRU41E9V5aer2DuP/L/Njt6hGpL7zgv+Xe  
PFY5+GCRAw4QOfRQ/yJVHTr4t48bF9wJ/6afX/5FHqjYSeYUmyeji4+XbqV6yLCSO2Riianye4kB  
Mr/Y4oQwNV76uVnmPn1KfCrflvhJhhUfJf8Uny5LYytkXPUJ0vWBbjJw9EB/I4AR9Rp0lhn5/Xw8  
wYw4fj7Clr5MAVkwYzqZoReohJGRzOAwtILY+y1116RjBWp4vAlphtRj0zzGyKHOGwC7C1wlwD  
zAjPqfBobgZQWEvMqG9GeE6FRzMzgMLSc1HqeXWjei8tveYB5wtHOj8/+7M8EHtQZsbmyJziC+Sj  
El/IB8U/l+HFxvhphCk9GnVVbJ2sjK2R1ebj+tgM7/MrYqtlQLG/5J9i02VSsWnyZYnvZWpslvzy  
8xR56OaHpO+www5GgAgbzWhshnm1ihwNzFhpBlAYhKnIGsJUuESYCpciU+ESYSpclkyFS4Sp2FK/  
fPeb3H/awzK90iyZEJsz5V7RV7c82XpfOSj0vr0NnLB+RdK08uaypmtzpQmVzWRs648S5pe3ITO  
vfBcuaLJFXLXsXdL79pvyTeVf5BPIn8jX8a+kx/3/Vn6/dBf5i6fF2wFSEWYCpciU5E1hKlwiTAV

LhGmwiXCVLhEmAqXCFOxpfr2+1MeveoJmV9tsfTfbqBc3rCFXNHiCml7dVtp27attGnTJm3pMq2u  
bSW3nXu7vL/9h/JO7D359JDPZNq46bJG1gZbAVIRpsllwIRkzYS5E6T6TdUldpF5aWoVqitMmc91  
fJswFYVHmAqXZi+ZLfXuqiexC818Sn79Mp9r+QphKgpvwfIFcsC9B6TOLy3zueYvEqai8JasWiL1  
H6wfPb8uiEmzZWlTUXiEqSisjRs3yKwFM+W9Nz+U+09+XGZUmCPDK4yUOw+5S66+5Bpp0751gWGq  
3q51Xrtz5abGN8mwsioIf2ygfFTzE3nrlfdl7PQJwdaAVBs3bZTG3Rp7Pwujfj426NxAVq4ITEXh  
EKYiayZMMCDVq1ePbLS0OnYkTEXhEabCpdmzZ0u9evUi55ZWY5aEqSg8vXL6AQccEDm3tJo3J0xF  
4ekFGuvXrx85t7Sa6VXNgUiTEVhrVi6Qp68oZs8WLeTvFXxHXm92pvybtU+0qfKB3JHwzul7SVt  
5fpLrpeOF3eU65p39D6Gy/vcJX7dfPytckfdu+XB6p3kiwpxf/F+0vPSr3kg9s/khETRsj69euD  
rQJ59LWpcePGka9dWgOaNPBe44DCIExF1hCmwiXCVLhEmAqXCfPhEmEqXCJMxZb6s99AeaTDk9Kn  
3ofyXvU+0muf1+SfbabKmtH6WVJimSwunr6WmFoZWY1/bt9fWh1/pdxT6T758sBvZMakmbKWt/kj  
DcJUUESYiqwhTIVLhKlwiTAVLhGmwiXCVLhEmLotNXjwX9KI41PyQe1P5Lk6L8idJ98pE3f4x0si  
NhQTWW8+5ldlf+NxUVGlX8rr9V6Xdqffq102vYR+fqgb2XB9IXeOTGB/BCmwiXz8gRkB2EqXCJM  
hUuEqXCJMBUuEabCJcJUBKkB/QbJQ20fk7dqVStdDnxc2p13tYzaZUxGYeomreliA3cYlk8d0l2u  
OrO1PFi1k3x5wNcyY4Iembou2AqQijAVLpmXJyA7CFPhEmEqXCJMhUuEqXCJMBUuEaZiS/381c9y  
/anXyzO7PSs373OLND3pbBIYbfDmMHVDEJxGVTxM/aXab3LXfnfLGcedlfdXelC+qvet/PH9HzJv  
+bxgK0AqwlS4ZF6egOwgTIVLhKlwiTAVLhGmwiXCVLhEmLot9edn/eSeo++T93f+ULod9rS0bt5G  
htcYaXV6kP+7DJAnj+gmHc7sIO9u30eG1/xbfvzOJ5m+aHqwFSAVYSpCMi9PQHYQpsllwS4RJgK  
lwhT4RJhKlwiTMWW+v3dfvJI7W4ytcxM+XubUfLybq/lnPLz/CQiwpxfYaL8sv1vMni7v2RKbLqM  
3nOcvPn8OzJq6hizABCNMBUu6csTkBWEqXCJMBUuEabCJcJUUESYCpciU7Glfnr5Z3mg8oMyrdgM  
L32YUnqGPLvzC3LIEW3knPPPNXWOnHvuuabOS6zzpMm550ITc48S96v/qEsi630LkilpwUYXX2c  
dH2gqwwcMzDYCPCKMBUuEaYiawhT4RJhKlwiTIVLhKlwiTAVLhGmYkv1/XSAPhLYuzK13GwvfVhU  
cqM8u0MfufPYu6Rtu7bS+urW0qZnm5RqZ2679MpL5ezmsnXu3/n3XdjMZF5sYUytO5w+fK9b2Ty  
vCnBVobUHklwiTAVWUOYCpciU+ESYSpclkyFS4SpclkwFVvqtwG/y4NtOsmkaIO89GF9sU0yvfgs  
6bP7B9L2jHZyweUXyIXtLpQr2l8hV3ZoJS3bt5RLrr5EzrviPLnijFbSpe4TMnabCX5yUuZk69i3  
0qv+6zJj0gzB0Z+NizelEu+WSJzX5wrs7rMkpkPzZT5b8yXVSNWycbVzNI/y6KPF8n0u6fLzE4z  
vY/zXp0nm9ZsCm51izAVLhGmlmsIU+ESYSpclkyFS4SpclkwFS4RpmJLLVi5QAb9Mki+2vcBGR4b  
KZuKiVcajEpxkRWx1TKmxAT5sty38sE2H8uXfB6RwSWGy6JiS71ldFk9IlWTC70g1Rv1essT9zwp  
q1au0tVHmvfKPBleb7gMKDFA+sX6Sf9Yf69GnzDaC/PWL8o/hEV2TbhogvSN9fX2v378+/C/ZcPS  
DcGtbhGmwiXCVGQNYSpclkyFS4SpclkwFS4RpsllwIRkw7q16+XzN76QHhe/JC/U6SUvIoSpr8Z6

yXuD+Tn2K8ysMRgGVB8iAwu/pdXA4oPkj9K9PeOQn019rq8UvZNeXGP16T7sc/KN698K9PmTk87  
72Y+PFP+ip3hBaka4A3ddaism7UuuDWwJQdHZuPAYkzWka1ICILYdWRwvwnNCVNRNBGmlmsIU+ES  
YSpclkyFS4SpclkwFS4RpsK1Y07Jf34de+qxwVIFW/7ncvmn9T/yZ+XP+T32u/Qv5h+Nqh81VP0t  
9psMrjpYpnacKuvM+MHq6nGrZV7PeTLpsknekawDyw/07q9BrH4cWHagjNhnhExqMck7onXNP2tk  
07q8BHHc2ePkl9gvXkj4a+XGVP9qCz7ZZnMuHeGDNlhiLeegaUGypBq/r91mSE7DZFVI/OOqt20  
epNMu3Wa95j1Mepyw2sPI5VD/ZBv04ZNsuz3ZTLx0onednR7f+35l8x/fb6sm+c/jw3LNsijzxfL  
5HaTZVjtYTKg+ABvPbpOLX0ugysPIlFHjZJpd0yTlX+tlI2rEr9v10xdl4O3G5zwfCacP0FW/LIC  
xp8z3tuXuh91f21c6d9Xt7v0h6Uy5dop3v7TZXS7+nHMyWNkQe8FMvr40Zs/R5iKooQwFVIDmAqX  
CFPhEmEqXCJMhUuEqXCJMBUuV27Vho2bBg5t7QaNWrkLZOJTWs3yYblG2TDkg0y+drJXoCn4Z9+  
1GBz6c9LZdP6Tbj+/nov8Jx0xSTpX9wPCDXoG7rDUO+8qisGrPBC0xX9V8j0u6Z7AWM8SBxYZqBM  
vWGqrBrIh6GTWk7yAkk93SsNclN16uc1yJx641RZ8vUS7zHo/w/edrB3/tB4MLvg7QVe8Kj30fJC  
3PIDzf7L82XTqk3eY9bHoQGqbkmD3snXTBYJvu1WDF7hnb5AT2mgz3XQNoNk+h3TZeWQlf7zGLhC  
Zjw0QwZWGLg51NSAd073Of4KAmunr5VhNYelPh9T+pg06NXHp+edXTNpjSz8YKGMPhSkd5suO6ji  
IJlw4QQv1F49drXMe3meDK87XAaUyJvVAmEqihLCVGQNYSpclkyFS4SpclkwFS4RpsIlwS4IM0w  
NUzDx4QwdechXsCn9HypY08dKwOKDfACPr193FnjZM2ENalvWzf/r/fToyvjlaMGkf9c+Y932z9t  
/tn8eQ0V9XQCMx+ZKatH+9uKWzdjnYw/e7wXaOpRoyP2HSGlPlzkXYhpgwUTZEDpAV6AOqXjFBnX  
ZJwXYOryS75cluvnrPfcUg1i9fP/tP1HlvVdFqxZZOOKjd6Rrst+XuZd7EmPBtXS88bqY9HHOOqY  
UZtDzfg+mdl5ZrAGX3KYqs9n1NGjvCNg189LPMfsoo8WydCdh3qPR9elQe7cl+amXNhr3fR1Mu7M  
cZvXSZiKooQwFVIDmAqXCFPhEmEqXCJMhUuEqXCJMBUu/V/C1IXrZXSD0QlHS+pb6MNV3w/bsHiD  
jDI1TElIOP7c8X6Y2jYvTNWPlw8ZKesXp17cSt9Sv+y3ZTK+2Xjv8WjpKQk0LNVTA+gRrxPOmyBr  
p66VeS/M8x6bhqvzx8vc56bl8PqDvOfhx5h++PSYK0+DUFnPzHbe5t9/DnrUawTL5soC/sslPVz  
18uyX5fJ4CqDN9+uH2c+mj5M1Y9TrpsS3Jplw1oNhXVduj80TF38yeLg1jwaFo8/b3zCOglTUVQQ  
piJrCFPhEmEqXCJMhUuEqXCJMBUuEabCpf9HmKrn+5z77Fwv1NMg0DuidOehMvHiid65STcz03rx  
V4u9Iyv1rfm6Hu9ozWNHyYI3F3iLTloy723++tELUxelhqxc56Z4wWiekSnhqX6b32MeuSrHu2p  
Ae3Sn5Z6R5JqwDqg5AAZutNQb/3D6gzzAtC10xL3x9Lvl8qgCoM2B5v6US/AFT9KVM+9Orn9ZBIY  
zj8fbHyfzHys8GHq8v7Lvbf0xwNafT569Ky+tV+Cp796zGqZ3HayFxbRdglTUdQQpiJrCFPhEmEq  
XCJMhUuEqXCJMBUuEabCJVdhql5kSYM7DfH0o4aWq8bkXfRJeSFjh8kyfO/h/tvvSw3YvLxWPlzV  
UHNQpUHe+UE1pF0zcU2wBvP7b/MJ3tvvdTn9OOKAEWnDVH37v56XNXyUqG4zHFqunbJW5jw9xwsh  
9XHEI9HwMn7hp7DVE1bLtJumec9Dz9eqoaUuH1+/nnpATyugF8HS56Ghqh5JOu22af7b94NVakj7

1+5/JTwf3T/50aNOp9853duuni81vt34vtPtjDxopPc8NOzVbet6R5842rtw1b+BMBUuEaYiawhT  
4RJhKlwiTIVLhKlwiTAVLhGmwiVXYeq6uetk5ciV3oWi9KNetV8vUBVFj+Bcv2C99xZ7DSb1iMpV  
o1d5R7JqcKpHbOpb/aNOA7B2xlpZOSLYjvmoF2bSq++no4Ho6vGr/fvpdsxj0+1vZr6ddBnddnzd  
urxeOCsdvfCWBqK6Pm+95j762HVD+pg05NX9sm62X7rNjct1Y/799fnp8w8/H12ulBqq6nL6eDWw  
1tL16Lb04ln6uLxtBtvW0yyknJvWEcJUuESYiqwhTIVLhKlwiTAVLhGmwiXCVLhEmAqXXIWpgCJM  
hUuEqcgawlS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVGQNYSpclkyFS4SpclkwFS4Rpsll  
wLS4RJgKlwhT4RJhKrKGMBUuEabCJcJUuESYCpciU+ESYSpclkyFS4SpclkwFVIDmAqXCFPhEmEq  
XCJMhUuEqXCJMBUuEabCJcJUuESYiqwhTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPhEmEqXCJMRdYQ  
psllwLS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmlmsIU+ESYSpclkyFS4SpclkwFS4RpsllwLS4  
RJgKlwhTkTWEqXCJMBUuEabCJcJUuESYCpciU+ESYSpclkyFS4SpyBrCVLhEmAqXCFPhEmEqXCJM  
hUuEqXCJMBUuEabCJcJUZA1hKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPhEmEqsoYwFS4Rpsll  
wLS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVWUOYCpciU+ESYSpclkyFS4SpclkwFS4RpsllwLS4RJiK  
rCFMhUuEqXCJMBUuEabCJcJUuESYCpciU+ESYSpclxkF1hCmwiXCVLhEmAqXCFPhEmEqXCJMhUuE  
qXCJMBUuEaYiawhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFORNYSpclkwFS4RpsllwLS4  
RJgKlwhT4RJhKlwiTIVLhKnIGsJUuESYCpciU+ESYSpclkyFS4SpclkwFS4RpsllwIRkDWEqXCJM  
hUuEqXCJMBUuEabCJcJUuESYCpciU+ESYSqyhjAVLhGmwiXCVLhEmAqXCFPhEmEqXCJMhUuEqXCJ  
MBVZQ5gKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmIqslUyFS4SpclkwFS4RpsllwLS4RJgK  
lwhT4RJhKlwiTEXWEKbCJcJUuESYCpciU+ESYSpclkyFS4SpclkwFS4RpiJrCFPh0tQpU2Wv6mnc  
1KslU1F4hKlwiTAVLnlh6oFpwtQzCVNRelSpclkwFS55YWqjNGHq4YSpKDzCVGQNYScq2SQy1Yy9  
zDCzKXK0M0OXAwwqDMBUuEabCGfNzb4kZ9c0wsylyNDMjvixgizAVLhGmwhnzM2+jGY3NMLMpcjQw  
Y6UZ/HxEYRCmlmsIU+FSRmEqUEiEqXCJMBUuZRymAoVAwAqXCFPhUsZhKlAIhKnIGsJUuESYCpci  
U+ESYSpclkyFS4SpclkwFS4RpsllwIRY2bQp8Rj49evXy59//ikDBgyQt956S6pVqxb5g1Dr7LPP  
liFDhqQ0XMnrRO5Kngs6V/r37y/9v+kvffr3kV1W77L5h1/yOGPaGTLwt4GyYdWG4N4+5hfikueC  
/r++dvXr108++ugj2W233SJfu7ROPvlkby7qa14Y8wvpDBo0yPsZ+dlnn0nNmjUj55bWCSeclH/8  
8YesW7cuuKeP+YV0tKf685s/5cu+X0q9ZfVCPxETx9FzjpZff/IV1i5ODCSYX0jnr7/+8l6Xvv32  
W9lvv/0iX7u09OJnP/30k6xZsya4p4/5hXSGDRsmffv2IR9++EEOOeSQyLmlddBBB8n3338vq1at  
Cu7pY34hnREjRsjv3/4uP//6sxyx8ljQT8TEse+ifeXrX7+WIXMSA1XmFzJBmAprEydOlFq1aknZ  
smWlatWqkT/40IWVKlWkXLLysueee8qYMWOCtQK+adOmyd577y1lypTx5lexYsX8uVPD1CRT+Y0e  
pnR+lasi5cuX946S1kYNUPGmaM6cOXLggQd682vbbbfNm18ZVuXKlaVChQqy0047eUEsoOLza9Gi

RXLYYYdJ6dKlvfIVvHjxyHmUX1WqVEkqVqzo/WHy999/99YJxOfX8uXL5aijppJSpUrJdtttlze/  
tjE10FR+40NTOOr9KVpJtttnGm5saTsTxS2Nui3/9NQzVP+yULFnSm18lSpRleH0qqHRu6WuY9vlf  
fPGFt07F/Mpt8a//hg0b5JRTTtmi+aU9mH788MMPvXUq5lduC3/9mzRp4s2r7bff3ptnm+fPV6by  
G7+ZKhuTirGK3muX9vh6gFgc8wvpEKYil/rXwB49enghavyFyTaECFf4vhqqPv30094vCXG8cOUW  
ffvOiy++6lWoUXPEq91MpQtTnzcVcd8aNWpI165dvasdxzG/ckP866xHk7788suy//77R84R2wrf  
d+edd5bHH39cFi5c6G1LMb9yQ/zrrEfQv/baa3LwwQdHzhHbCt9XQ9XOnTvLvHnzvG0p5lduCH+d  
e/fu7R39FzVHvKpsKI2Y+pGpiPtqqPrQQw/JrFmzgi0xv3JF+Ov8zjvvvNFHHx05R2wrfF8NVu+7  
7z6ZPn16sCXmV64lf53ff/99Of744yPnyJaUHjh9913y5QpU4ItMb9yRfjr/PHHH3ungojPi5T5  
VdxUujD1d1PITSXdXw+6uP32272DyOKYX0hGmloCXXTRRXkvSI6rWTPO65Vr9FYUUXMhpSzC1PxK  
z+uF3NKuXbvlueCitJnTly+QO/Rc4FFzwUXpOQuT30aLok1/kYuaCyllEabmV3pE9YoVK4Itlxfc  
f//9kXPBRekfMxcvXhxsGblA/xAYNRdcVN26dRP+6lii76mnnoqcCyllGaZG1R577CEzZ84Mtgzk  
IUzNUZvMyNRVza6KfGFxUZc2vjTYahr8Uahlad++feRcSKkshKnnn39+sFUUGQW8Htx8xc2Rc8FF  
NTm6iUgmWSqvYVuPaR5Wd199d+RccFGnHHqKSCZZKvOryHj44Ycj50JKZSFMPe6444Ktosgo4LXg  
yTufjJwLLurwWoeLLAo2nA6vX1uPaR5WPR7sETkXXNRBNQ6SDXMzaMCYX0XGq6++GjkXUioLYaq+  
czL+x2yOTkUYYWoOW2fGkWaYl4nosSnv38XMyPs/NyNhG6Ft5zvOMxXgkdtfZXRW362MEzN1tuK  
qP9QvWMq3QheR/6Tr19XmYp6TtR/p141lW78l+dXB1NRz4na6iqjn11bGKby87EIVldT6cZ/+fXr  
HlNRz4n679QjptKN//L8ethU1HOitrK6GfXFoap4W3ceeedQYoC+AhTc5iGqSebYV4ets7RwlTo  
xY4q4pWFI1OpIlyfmNpax3Wmop4T9d+p3qa21nGLqajnRBXNysKRqVQRK70o59Y6NKiLek7Uf6cK  
Cuv/y+NJU1HPiSqaLYUjU+PVqVOnIEUBfISpOUzD1GPMMC8PBY5//S+LmYwLTUW80FFbX2X0lOWO  
TKWS611T6cZ/+ciltqainhp136leptKN//L8lqwvMpXRzy6OTKWSq5updOO//Pp1v6mo50T9d+pR  
U+nGf3l+dTYV9Zyora4y+tmVhTC1YsWKcs4558h3330XpCiAjzAVBbrssssiX1hc1AXnXxBsFbni  
qqsyPCdvFo5Mbdq0abBV5loOHTpEzgUXddqpp3EBqhxz0003Rc4FF3XiCSfk2jVrgy0jF+iVqqPm  
Qkpl4chUvZr7ypUrgy0jF3R6qFPkXHBRRhx5yqCxZvCTYMnLB410ej5wLLmr//faX+fPnB1tGLnjm  
mWci50JKZSFMrVOnjsyaNSvYmpCHMBUFgJjkiFx33XWbX1CyeQRDeF3XXHONDBo0KNgqcsWwYcPk  
5pvzLhKU7/wqRJgaXlfr1q2lf//+wVaRK0aOHJlWRWxXr18tW7aUP//8kxPT55hRo0YIBF6u5tcl  
l1wiv//+u2zcuDHYMnLB2LFj5cEHH5TixYunzImEKkSYGI7XBRdcIL/88ousX78+2DJywfjx4+WR  
Rx6RUqVKpcyJLa3wupo1ayY//vijrFu3LtgycsHEiROI55cuUq5cuZQ5saUVXteZZ57pHTEYv0AQ  
csPkyZOLA9euss0226TMiYQqRJgaXtepp54qX3/9taxevTrYmpCHMBUFCocDDz30kQvXz7hBWdL  
Sn/A3nvvvcHauUJeLgp/zR9//HHvrrRRRc6UwYWRzsmXl1ItvDdbO/MpF4a/5008/LZUqVUqZJ4Wt

0qVLS8eOHYO1M79yUfhr/uKLL0qVKIUi50phSgOOq6++OlG78ysXhb/mvXr1km233TZyrhQmTC1Z  
sqS0atUqWDvzKxeFv+bvvPOObL/99inzpLBVokQJufTSS4O1M79yUfhr/tFHH8mOO+4YOvcKUzq/  
zj///GDtzK9cFP6af/nll7LzzjtHzpXChKn6B8yzzjpr8x+wmV/ID2EqrOmRC3o0zj///CPvv/++  
9wMt/AIU9Zeh8Ofefvtt769Jf//9N3+lRgp9m7TOr0mTJsmnn34qZcqU8eeO/rl4wcyliPMveZ/r  
ZP5llnvtpde8+aVHJPJXaiTTxmJmMDeERNffVVSnhf0OtxZ549ZcqUKTJixAj+So0U2nDr0YQ6  
v77//nupWrVqvnMp6nPPpvusN7+GDx/OW64Rady4cTJhwgTvSNlddtghby71C34WJg3vc6+Zf5ll  
nuz05Ob5tWLFimCNQB49WIWrb9++UqNGjbz5Zaqg16/OnTt780vfcRs2bJgjUAefe3S1zB9p1jN  
mjXznUtRn3vggQc2z6+lS5cGawTy6O+O2oMNHjxY9t5777y59GnwszBpeJ/71vzLLHPHTXdsnl+L  
Fy8O1gikR5iKQkv+K42eq2a33XbLe+FKKn2rbRx/4UFBkufUjNqmxF5cnnzuRvMiGN+oSDJc0SD  
93333TfytUurXbt2wZLMLxQseY5oiH/wwQdHzitPU1EHPMLBYmal0eZkd/PxwvNiGN+oSBRc+T4  
44+PfO3SOvvss4OlmF8oWNQc0bdSR80trdNOOy1YivmFgkXNkbPMYO/n44lmxDG/YlswFVmjf22s  
Xr165A9CrFDbYQFbU83YywwzmyJHOzOAwp09e7bUq1cv8rVLKxx2AbYWLFggBxxwQOTc0mrevHmw  
JGBviRn1zTCzKXIOMwMoLD1K/phjjol87dl6/ftTOZ8zCm3t2rXSsGHDyLml1ahRI28ZoDA2mtHY  
DDObIkCDM1aaARQGYsqyhjAVLhGmwiXCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBVZQ5gKlwhT4RJh  
KlwiTIVLhKlwiTAVLhGmwiXCVLhEmIqslUyFS1PnTJW9LtxLYgeZ+XRsqBqYOjAm7Z4gTEXhEabC  
JcJUuLRkxRKp37q+97Mw4eej1gExaXYnYSokjzAVLhGmwiV9bWp8c2PvZ2HKz0fzM7PBNQ1k5RrC  
VBQOYSqyhjAVLk2dOIX22muvyLmlFb5AEGCLMBUuEabCpSVLlkj9+vUj55ZWs2aEqSg8wls4RJgK  
l7wwtXHjyLml1aBBA+81DigMwIRkDWEqXCJMhUuEqXCJMBUuEabCJcJUuESYCpciU+ESYSqyhjAV  
LhGmwiXCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBVZQ5gKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXC  
VLhEmIqslUyFS4SpclkwFS4Rpsllwls4RJgKlwhT4RJhKlwiTEXWEKbCJcJUuESYCpciU+ESYSpC  
lkyFS4SpclkwFS4RpiJrCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUuESYCpciU5E1hKlwiTAVLhGm  
wiXCVLhEmAqXCFPhEmEqXCJMhUuEqcgawls4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVGQN  
YSpclkyFS4SpclkwFS4Rpsllwls4RJgKlwhT4RJhKrKGMBUuEabCJcJUuESYCpciU+ESYSpclkyF  
S4SpclkwFVIDmEqXCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUuESYiqwhTIVLhKlwiTAVLhGmwiXC  
VLhEmAqXCFPhEmEqXCJMRdYQpsllwls4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmlmsIU+ESYSpC  
lkyFS4SpclkwFS4Rpsllwls4RJgKlwhTkTWEqXCJMBUuEabCJcJUuESYCpciU+ESYSpclkyFS4Sp  
yBrCVLhEmAqXCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUZA1hKlwiTIVLhKlwiTAVLhGmwiXCVLhE  
mEqXCFPhEmEqsoYwFS4Rpsllwls4RJgKlwhT4RJhKlwiTIVLhKlwiTAVWUOYCpciU+ESYSpclkyF  
S4SpclkwFS4Rpsllwls4RjiKrCFMhUuEqXCJMBUuEabCJcJUuESYCpciU+ESYSpclkxF1hCmwiXC

VLhEmAqXCFPhEmEqXCJMhUuEqXCJMBUuEaYiawhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqX  
CFORNYSpcikwFS4RpsllwIS4RJgKlwhT4RJhKlwiTIVLhKnIGsJUuESYCpciU+ESYSpclkyFS4Sp  
clkwFS4RpsllwIRkDWEqXCJMhUuEqXCJMBUuEabCJcJUuESYCpciU+ESYSqyhjAVLhGmwiXCVLhE  
mAqXCFPhEmEqXCJMhUuEqXCJMBVZQ5gKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVLhEmIqsIUyF  
S4SpclkwFS4RpsllwIS4RJgKlwhT4RJhKlwiTEXWEKbCJcJUuESYCpciU+ESYSpclkyFS4Spclkw  
FS4RpiJrCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUuESYCpciU5E1hKlwiTAVLhGmwiXCVLhEmAqX  
CFPhEmEqXCJMhUuEqcgawIS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmwiXCVGQNYSpclkyFS4Sp  
clkwFS4RpsllwIS4RJgKlwhT4RJhKrKGMBUuEabCJcJUuESYCpciU+ESYSpclkyFS4SpclkwFVID  
mAqXCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUuESYiqwhTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPh  
EmEqXCJMRdYQpsllwIS4RJgKlwhT4RJhKlwiTIVLhKlwiTAVLhGmlmsIU+ESYSpclkyFS4Spclkw  
FS4RpsllwIS4RJgKlwhTkTWEqXCJMBUuEabCJcJUuESYCpciU+ESYSpclkyFS4SpyBrCVLhEmAqX  
CFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUZA1hKlwiTIVLhKlwiTAVLhGmwiXCVLhEmAqXCFPhEmEq  
soYwFS5NnTZV9tonCFPLhaqMP7/aXUuYisKbPWe21DswCFPD86usP79atiZMReERpsKIJUuXSP0G  
QZgafv3SMp9rdiFhKgqPMBUueWHqaUGYGvH61agxYSoKzwtTmwZhasT8atCwgaxcRZiKwiFMRdYQ  
psKZTSJTzdlDDObIkC7M4BCMfNrthn1zAjPqfBoaQZQWISpcMa8fi0xo74ZZjZFjmZmAIVFmAqX  
1prR0AwzmyJHlZn0GaAwNprR2IzwnAqPBmasNEN/IgK2CFORNYSpcikwFS4RpsllwIS4RJgKlwhT  
4RJhKlZKOEwFCoEwFVIDmAqXCFPhEmEqXCJMhUuEqXCJMBUuEabCJcJUuESYCiubNiUeA6/nsHnr  
rbekV69e8vDDD0vVqlUjGy2tY489Vt577z1Zt25dcG9f8jqRu5LnwoYNG+Sdd96RV7u+Ko+985hU  
W1Zt8w+/5HHUsKPk7VfeltVLVwf39jG/EBc1F/r06SOvdH1FnnjrCdlp8U6hGZU46v9dX958+U1Z  
uSix4WJ+IS5qLnzwwQfy8ssvS7du3dL+sfHggw+W1157TZYvXx7c08f8QlzUXPj444+Iz9ee0v31  
7rLHvD1Cr1iJY/8J+8vLPV+WJbOXBPF0Mb8QFzUXPvvsM+nZs6f06NFDatWqFfnabpX33nvLiy++  
KlsWLQRu6WN+IS5qLnz55ZfyUteX5PIXnpd6M/P/Y3btqbWIR88eMn/K/OCePuYX4qLmwjffffCMv  
dntRXnr5Jdlv2n6hGZU49pyxp3R/ubvMmTgnuKeP+YVMEKbC2pAhQ6RKISqRDZVNVaxYUfr37x+s  
FfD9/fffUq1atdQ5s5upSabyG8+bCi1fpkwZ+fXXX4O1Ar7x48fLrrvumjBXvNrJ1BhT+Y1epkLL  
lyhRQR777rtgrYBvypQpssceeyTmlcJUsWLF5PPPPw/WCvhmzZoltWvXTp0zIU0NNJXf+MhU0n3e  
f//9YK2Ab/78+bLffvulzJXCVO/evYO1Ar7FixfLIYcckjpfSpv60VR+4wdTukzoPq+88kqwVsC3  
YsUKOeqooxLmiVfFTX1IKr/xu6nypkL30T8gAZkgTEW+wn+R0b82X3fddbLDDjtsfqHRX/bCLzw2  
Fb7vdtttJ9dcc43XxMXx16DcEP8669FYN954o+y0006Rc8QrizA1fF89WrpNmzbeL6FzxK/cEP86  
r1q1Sm699daEEDVlflmEqeH7VqpUSa688kqZNm2aty3F/MoN8a+zvtvizjvvlN122y1yjmxJ6R8d

W7Rolf/884+3LcX8yg3xr7N+vO+++xJC+pT5VYgwVat8+fJy6aWXytixY71tKeZXbgh/nR966KGE  
IO+z9fqIf9S+6KKLvD+SxzG/ckP46/zoo49K3bp1N8+LIPiViDBVq1SpUnL++efLX3/9FWyJ+ZUr  
wl/nJ598UvbZZ5/N8yJlfhUiTNXSgybOOeccGThwYLAi5hdSEaaiQPpCEn9hyVaDFa7wOvW8S8gt  
/2vvXoCkqu48js8wglAoigygbBHIWFAYESoMGEJQmSCiYoT4KCosjo81IMhDxYhFjHE1WZESoelj  
usq6SdiYCqtVmNVSV5PUatasa+ljxrXGEFEQCQGCYQB5Ow5d27D7e7Tj7lzf1mm7/dz6l9LsKe7  
qP7vneE7zbT7WYG+Xcibdr4yNTPR+2xsbAwfFWlx5ZVXenchb9r5ytTMRO/T/Uw596MpkB7uG4G+  
XUhqovc5duxYs3fv3vCRkQbum4y+XcibmDE1Oqeddlrwyh6kx9KlSw89/4rrV3TcK6pzfwWAKpuL  
9Jnnv+h+xYypbjL3e/LJJ5stW7aEj4w0WL58ed4eeCdmTHWTuV/3Yp+NGzeGjwwcRkxNqYP2lOuq  
mVfIXvXUM/u82eGjFsE3hSrKvHnzvLuQNZfJanQuvvji8FGRFosXL/buQt7EjKnRmT59evioqBgI  
Pt8svfZwjFDpILFTjCmnpfI5smLceeed3l3lmwRi6qRjK8JHRcUocS24Z+k93l1QzLhTxlITtkvl  
+lUx7rvvPu8u5E0HYmpmRo8ebfbv3x8+MtLA/agH3y7kTQdiambcz4V2/8rN4dWpiCKmptg+e8bb  
Yy8T/nPw8K+r7Tn8vzQn6zEij13wfNWO54LHdL4p6xURHYypZT0GU5FT1nPfwZjKfXgPGqn2Ak/  
Tx2Rnx/n2fH9mZhON2VdWzoYU7l+VeDcY6fYOZKvX7fa8f2ZmE43ZV1bOhhTuX6ld8p67jsYU6OP  
ccstt4QVBWhDTE0xF1On2GMvD53zXG4ncrFjKnwSeGUqwxScBF6ZylTY/MhOZz032fH9mZjKnARe  
mcpU2Nxxvp7Oe79rx/ZmYypwEXpnKMAUngVemZsb9axEgipiaYi6mnmGPvTz4T+S7x0fkd64vtuO5  
ODGdb8r6ziKvTGVITInPPa9MZXJntZ1i50h+Zdd1dnx/JqbTTVnXFl6ZyuTOCjvFzpF8/fqWHd+f  
iel0U9a1hVemMjGnrOc+gZjq3ix7yZlI5vXXXw8rCtCGmJpS7fmZqU0XN3kvLlqZdcGs8FGRFnPn  
zvXuQt4k8MrUGTNmhl+KtFi0aJF3F/ImgVem8gZ6FajEp8ol85Z4d0Exk8dN5mempsy3v/1t7y7k  
TQKvTHVvolcKU+JasOzWZd5dUMzY4WPN/h1l/ExLrI8VY9WqVd5dyJsEXpnq3kAv8zMtkQ4PPfSQ  
dxfyJoGYWI9fbz766KPwkYHDIKko6eOPPzYrVqw4dEFJ8juA0ftatmwZF6oUcu++Gf0h9QX3K0ZM  
jd6Xe1fRTZs2hY+KtNi6dWvWF1wF9ytGTI3e16233mo+/PBDfjB9ymzbtS2sXr3adOnSJW8nOjrR  
+7r55pvNhg0b2K+U+fOf/2zWrFljunfvnrcTWRMjpkbvY33T6Y9//KP59NNPw0dGGuzYscP85Cc/  
MUcffXTeTnR0ovd17bXXmj/84Q/mwIED4SMjDVpaWswTTzxhjvuuLydyJoYMTV6X1dffBvpbm7m  
DahSZufOneapp54ytbW1eTuRNTFiavS+5syZY9599132C17EVJQU/cvbl488YoYPH551wenInHLK  
KebBBx8M7z37sZAO0ef8Bz/4QfCOibl7UI1IP6nFiKnDhg0zK1euDO+d/Uqj6HP++OOPm8997nN5  
exLsV4yYOmTIELN8+fLw3tmvNlo+5+4vjaNGjcrbk4Jf4JeYuro6853vfCe8d/YrjaLP+c9+9jMz  
ZsyYvD0Jrl+IYuqTdnl+btCgQea2224L7539SqPoc/7cc8+ZsWPH5l2v4l6/TjrppKw3a2G/0if6  
nP/yI78048ePP/SNx8wE169SMdX9t5yYOnDgQHPjTeG985+pVH0Of/Vr35lJkyYYGpqarL2JNiv  
UjH1JTs5MbV///5m/vz54b2zXyiMmlp2c69ccN9tdF5++eVD39HOjO8Lr8zvUvDx/OIXvwg+1n1H



/ADfpUaOzH65T1yvvvrqoe9oV9Xaec/ukufnewW/9922T5rPrns2uB/2Cz5ur9x+uT174403gi+Y  
gv1yMeLtlvt1r/2Vvd1Ta58K7of9go/bL/dqCbcbv//974NgFexXOMU+P7pZu3ZtcD9uv3gVBHwy  
+/Xee++ZoUOHtu1ONzuvFLI+/Yv9lb3dj/7pR8F9bN++3ezbty/4NRD1ySefBNee999/34wYMeLQ  
tcINqevXo48+GtwH+4VC3H653di4caM5/fTTD+/S80WuX+vsr+xtHlj+QHAfbr9aW1uDXwNRu3bt  
CnbD/avHcePGHd6vp4rs13P2V/Y2K/5hRXAf7usv9gvlIqYittzvOrgLI3slzaELV864f6aYwXd4  
UErujuypw94e75sX2N9bZE8G+4VScndkjz2n2INov75uTwb7hVJyd8TFCd+rCjNz+eWXh7dkv1Ca  
b0eCNxQtCP26zJ4M9gul+HZk0qRJ3muXm6985SvhrdgvIObbkXPskXT9OteeDPYLpfh2ZL09hfbr  
LHsy2C+0FzEVIXE/r2bw4MHeL7TcLFy4MLwI0H4b7Blmj90m75lrDxDXZntG2BPdqehpsgely/1s  
Vd+PAMjMrFm8+SLia7GnwR67Td4zwx4grt27dwvUOa7drlxb77Iz9tFXK32TLbHbP3NNrjbgPE  
8ak90+yJ7IT0TLRntz1AHMRUJlaYCiViKpSlqVAipkKJmAolYiqUiKlQlqZCiZiKxBBToURMhRix  
FUEVCgRU6FETIUSMRVKxFQoEVORGGIqIDzS3GCGTRhmqo6x+zQ4MoPs2N+bewsxFFft3rLZjDhr  
hH+/eleZphulqYiPmAqllk9aTMP5DcG1Kuv65eboKjPjamlq4iOmQql1X6uZfNnk4Frlu341XtYY  
3Aalw12bpv3dNP9+2c+ZE2dMNLv3EFMRDzEViSGmQmnDhg1m2LBh3t1yM3cuMRXxbd68Oe+di6PT  
1ERMxZEVci1tLSYhoYG7265mTGDmIrl4iKlQcu+cPnnyZO9uuWlsbOTd1RFbEFOntfPulpuJEycG  
1zggDmlqEkNMhRixFUEVCgRU6FETIUSMRVKxFQoEVOhRExFOipUCKmQomYCiViKpSlqVAipkKJ  
mAolYiqUiKIIDDEVSSRUKBFToURMhRixFUEVCgRU6FETIUSMRWJlaZCiZgKJWlqllipUCKmQomY  
CiViKpSlqVAipilxxFQoEVOhREyFEjEVSSRUKBFToURMhRixFUEVCsGmAolYiqUiKlQlqZCiZgK  
JWlqllipUCKmQomYisQQU6FETIUSMRVKxFQoEVOhREyFEjEVSSRUKBFTkRhiKpSlqVAipkKJmAol  
YiqUiKlQlqZCiZgKJWlqEkNMhRixFUEVCgRU6FETIUSMRVKxFQoEVOhRExFOipUCKmQomYCiVi  
KpSlqVAipkKJmAolYiqUiKIIDDEVSSRUKBFToURMhRixFUEVCgRU6FETIUSMRWJlaZCiZgKJWlq  
llipUCKmQomYCiViKpSlqVAipilxxFQoEVOhREyFEjEVSSRUKBFToURMhRixFUEVCsGmAolYiqU  
iKlQlqZCiZgKJWlqllipUCKmQomYisQQU6FETIUSMRVKxFQoEVOhREyFEjEVSSRUKBFTkRhiKpSI  
qVAipkKJmAolYiqUiKlQlqZCiZgKJWlqEkNMhRixFUEVCgRU6FETIUSMRVKxFQoEVOhRExFOip  
UCKmQomYCiViKpSlqVAipkKJmAolYiqUiKIIDDEVSSRUKBFToURMhRixFUEVCgRU6FETIUSMRWJ  
laZCiZgKJWlqllipUCKmQomYCiViKpSlqVAipilxxFQoEVOhREyFEjEVSSRUKBFToURMhRixFUE  
VCSGmAolYiqUiKlQlqZCiZgKJWlqllipUCKmQomYisQQU6FETIUSMRVKxFQoEVOhREyFEjEVSSRU  
KBFTkRhiKpSlqVAipkKJmAolYiqUiKlQlqZCiZgKJWlqEkNMhRixFUEVCgRU6FETIUSMRVKxFQo  
EVOhRExFOipUCKmQomYCiViKpSlqVAipkKJmAolYiqUiKIIDDEVSSRUKBFToURMhRixFUEVCgR  
U6FETIUSMRWJlaZCiZgKJWlqllipUCKmQomYCiViKpSlqVAipilxxFQoEVOhREyFEjEVSSRUKBFT

oURMhRlxFUrEVCSGmAolYiqUiKlQlqZCiZgKJWlqllipUCKmQomYisQQU6FETIUSMRVKxFQoEVOh  
REyFEjEVSsRUKBFTkRhiKpSlqVAipkKJmAolYiqUiKlQlqZCiZgKJWlqEkNMhRlxFUrEVCgRU6FE  
TIUSMRVKxFQoEVOhRExFYoiPUCKmQomYCiViKpSlqVAipkKJmAolYiqUiKlIDDEVsRUKBFToURM  
hRlxFUrEVCgRU6FETIUSMRWJlaZCiZgKJWlqllipUCKmQomYCiViKpSlqVAipilxxFQoEVOhREyF  
EjEVSsRUKBFToURMhRlxFUrEVCSGmAolYiqUiKlQlqZCiZgKJWlqllipUCKmQomYisQQU6FETIUS  
MRVKxFQoEVOhREyFEjEVSsRUKBFTkRhiKpSlqVAipkKJmAolYiqUiKlQlqZCiZgKJWlqEkNMhRlX  
FUrEVCgRU6FETIUSMRVKxFQoEVOhRExFYprfbaDJww2Vf3sxWlYzIbYsb+38B+lqYiPmAqlzX/a  
bEZMGWGqTrD7FL1+DbXTt8o03URMRXzbtm8zo84blb9fbux+zVplTEV8LZ+0mlaLGoJd8u3XjLnE  
VMRHTIVS675WM/lrkwtvxpNwa3AelgpkKJmlrENNs2B57afKehfYAcRFTobTZnhH22G3yniZ7  
gLi22TPKnuhORc8se4C4WuxpsCe6U9Ezwx4gLmlqlFrtmWyP3SbvabTH3Qalg5gKJWlqEkNMhRlX  
FUrEVCgRU6FETIUSMRVKxFQoEVOhRExFYoiPUCKmQomYCiViKpSlqVAipkKJmAolYiqUiKlOl4MH  
D4a/arNnzx5z++23mxuuuMFc9Y9XmWNajjn0yS/3jHx2pLlJyR1md0v2BSv3PpFeubuwb98+c+ed  
d5rrr7/eXHPNNaZPnz7eT4Ruhg8fbm677Tazc+fO8KPbsF/lyN0F97/vuusus+iKRWbu7XNN3y19  
l1es7FP/n/Xmmzd902zfvD386DbsFzJ8u7B8+XKz8lqFZt635pnaj2ojG5V9hv7XULNk8RLzpw/+  
FH5kG/YLGB5duPfee4P9mr90vhn4wcDIRmWfuv+pM4tvXGw+eu+j8CPbsF/l8O3C9773PbNgwYLg  
DWQHDRrk/drLjXvzWfd12gcffBB+ZBv2Cxm+XXjwwQfN/Cvnm+tvvt7UNddFrIjZ58S3TzQLFi8w  
699aH35kG/YLGB5deOSRR8z8+fPNDTfcYIYOHeq9drkZMGCAmTdvnnn33XfDj2zDfqEcxFs02zPP  
PGNqamqyLkTVVdWmq7++kM7BU71KnubyMe4WbduXXivQJsXX3zR9OjRI2tPqqvzdyd3fLdZu3Zt  
eK9Am5deesccc8wxWXSXL8G2l+/Y6fAqX4sf7/WrFkT3ivQ5te//rXp27dv1p4E++XeeOoNOwVO  
9Y/z92v16tXhvQJtfvub35qBAwdm7UmwX33sr//bToFT/WT+fj3wwAPhvQJtfve735m6urqsPSnn  
6y/fuNAPRLIYVV9fn7UnwfWru/31C3YKHfff3G0iH7ds2bLwXoE269evNyNHjszak7jXL/dCMWlq  
ykFMRUHRi8imTZuCl8jnRq6sKRFTq1ba8XzcUUcdZaZMmRL8M+4MLmDpkHmet27daqZPn2569erl  
3ZGOTpfu3c3ZZ59tmpubg8dy2K90yDzPO3bsMDNnzjRHH320d0eCKRFTq/7ZjufjunXrZiZNmmTe  
fvvt4LEc9isdMs/zX/7yF3PppZea3r17e3ckmBlxtepf7Xg+rmvXrmbChAnm9ddfDx7LYb/SlfM8  
u3+hMXv2bHPsscd6dySYejG16gk7no9z3xgfP3588E2ADPYrHTLPs/snsFdccUXWv/yJGyBypOuX  
Lmbs2LHBNzEz2K90iD7P7l+WHX/88Yf2Im+/SsXU/7CTE1PduPsZM2aMeeGFF8JHYr/Sivo8X3fd  
deaEE07I2ovonnRkRo0aFbylLIP9Qi5iKkq66KKLDl1Uil6gYsbU6H2ee+654aMiLS677DLvLiQ1  
0fs866yzwkdfWJq1NXl3IW9ixtTofX7hC18wBw4cCB8ZaeB+VrNvF/lmZkyN3qf7S+PevXvDR0Ya  
uH867duFvIkZU6P36V7Rs2vXrvCRkQa33HKLdxeSGnefmft1P/N++/bsH5ODyuZe3Rfdhcyv8yZm

THWTuV/3oya2bNkSPjLS4O67787bg6Qnc7+1tbVm48aN4SMDhxFT4RX9zov7jnXuxcU7MWNqdGbN  
4k04Kk6Jb+LNNXM4Rqhn5tSZ4aMiLalxoujEjKnROf/888NHRVosWbLEuwt5EzOmRqexsTF8VKRF  
NEYUnZgxNTruTTh41U2FKf0LrttmXcXFNmwssG07ijjTYRYwYqxcuVK7y7kTQdiamZOP/304F+J  
oIKUuBY8tOIh7y4oZvhnpsdm3aEjwwcRkxFQc8999yhi0hZ3/HpYExVfVeJ+X8c9xe4Yudg2/+t  
tif8HdnJeoZwcYue2XZ8fyam001Z15YOxISuX+mdsp77DsZU9iu9U9Zz38GYyn5V4KywU+wcyV9/  
fcuO78/EdLop69rSwZjK9asCx10Dip0j+Pp1hT1ID2lqCnJvBOR+3qT3luebBF6ZylTQVnt5yk5n  
PdfY8f25mMqcBF6ZyJAFJ4FXpjJMwUnglalMhc39djrr+a4d35+JqcXJ4JWpTIWNuwZ00nOtPUgP  
YioK2rlzp3n22WeDN1fxXuhyh1emMrlT7JWp7rt7R/lrl75mx/dnYjrdlHVt4ZWpTMwp67nnlalM  
zCnrueeVqUzu3GON2DmSv/661Y7vz8R0uinr2slrU5nccdeAYuclvn5dbg/Sg5iKktzPMfVe6HIn  
gVemujcQoUp8TNvvn7J1727oJiLlOUPmoJ/MyuirFo0SLvLuRNAq9M5Q300udv+TNTeQO99PIb  
/sxU9wZ6qDALpa5e8nhN3BRz5iTxxhTzvtP8fVxVi1apV3F/ImgVemnnrqQWb//v3hl6MilLgW  
PHzXw95dUMxnj/+s2f3h7vCRSzvIhSw1iKkoy09/+tNDF5SC3wGMEVOj97VmzRre/CCl1q1bd2gX  
kvwOc/S+Vq9ebT799NPwEZEmzzzzjOnWrVveTmRNjJgava/vf//77FdKvDCC6Zxr155O5E1MWJq  
9L7cX0oPHDjA58gUeumllOyfPn3ydiJrYsTU6H0tW7YsCBHsV/q88sorpl+/fnk70dGJ3tcdd9xh  
Wltb2a8U+s1vfmNOOumkvJ3ImhgXNXpf55cuNXv37mW/Uuitt94yQ4YMydqNjCa6X4sXLz79uxh  
v+BFTEVJ0YvH008/baZONzP1wXFTXWUvOiViavWq/E+iU6ZMCUJaBheq9lk+588//3zwjui5e1LO  
F/i+25x99tlm7dq14b2zX2kUfc5//vOfmwsvvdBvT4LrV4mYWv1Y/n65H4Hy+OOPh/fOfqVR9DI3  
0WvmzJI5exLsV4mYWv3j/P2aMGGC+eEPfxjeO/uVRtHn3EWvSy65xHTp0iVrT4L9KhFTq5/M369x  
48YF32TMYL/SJ/qcv/baa8G/ROvatWvWnpTz9ZdvPv/5z5uHH344vHf2K42iz/mbb75pZs+enfde  
HMH1q1RMdf8tJ6aOHj3a3H///eG9s19pFH3O33nnHdPU1GR69uyZtSdxr1+nnXaaWblyZXjv7BcK  
l6aiQ9zF68QTTwwuPDXDa4q/MvWOKIPbq9a8+dqb4UcDxa1fv97U1dW17VeN3a/IJzrfHHfcebV  
V18NPxoobuPGjaa+vJ7YnZpBdr/+1+5RoXNflendrbd5+cWXw48Givv444/NyJEj2/arf42pftN+  
UV/oPFJlenTtYV58+sXwo4Hitm/fHgSFYL+Ot/v1apH9WINlutZ0NU//29PhRwPF7dq1y4wfp75t  
v+zXX6WihPvvTz75ZPJRQHf//etfD70nR01Pu18vFrl+/budrlXm8UcPf/MaKObAgQPBC7aC/Srj  
+uXmscceCz8aKB8xFbHlfpdmkz2fscf7w5nt791kTwbf4UEpuTuydevWQ+HLNwsWLAhvyX6htNwd  
2WnPSHsKXb/+3p4M9gul5O7IHntG21Nov+bYk8F+oZTcHdlvz3h7Cu3XJfZksF8oxbcjX/rSI7xf  
e7lx/+ljg/1CKb4dmWJPoexXOfZksF8oxbcjF1xwgffa5ebMM88Mb8V+of2lqUhMsz2D7bGXJu9Z  
aA8Q14YNG8ywYcO8nwjdZJ07N7wl0H6b7Rlhj90m72myB4hrmz2j7InuVPTMsgelq8WeBnuiOxU9  
M+wB4tq9e7f54he/6P3ay815553HzwxHbK32TLbHbpP3NNrjbgPE4a5N06ZN81673EycODG4xgFx

EFORGGIqllipUCKmQomYCiViKpSlqVAipkKJmAolYiqScZCYCi1iKmTs9YuYChm7X8RUyNj9lqZC  
iZgKGXv9Kjum2tsC7UVMhRlxFYlpfq/ZDB4yuO3i1DUyXdouVguvJ6YiPmIqIDZv3mxGnDqibZ+i  
16+atv1qupKYivi2bdtmRo0ZVXC/Zn2NmIrr4Wna2mIYzGvL3y439vRlfJaYiPmIqIFpbW83kKZPb  
9slz/Wo8pzG4DRAHMRVKxFQkprmm52QweHMZUzyxcSExFfMRUKAUxdUQYUz3T1ERMRXxBTB0VxITP  
zJpFTEV8LS0tpqEhjKmemTGDmIrr4iKlQCMlq5DCmeqaxkZiK+lipUCKmIjHEVCgRU6FETIUSMRVK  
xFQoEVOhREyFEjEVSsRUJlaYCiViKpSlqVAipkKJmAolYiqUiKlQlqZCiZiKxBBToURMhRlxFuR  
VCgRU6FETIUSMRVKxFQoEVORGGIqllipUCKmQomYCiViKpSlqVAipkKJmAolYioSQ0yFEjEVSsRU  
KBFToURMhRlxFuREVCgRU6FETEViiKlQlqZCiZgKJWIqllipUCKmQomYCiViKpSlqUGMMRVKxFQo  
EVOhREyFEjEVSsRUKBFToURMhRlxFYkhpKJmAolYiqUiKlQlqZCiZgKJWIqllipUCKmIjHEVCgR  
U6FETIUSMRVKxFQoEVOhREyFEjEVSsRUJlaYCiViKpSlqVAipkKJmAolYiqUiKlQlqZCiZiKxBBT  
oURMhRlxFuREVCgRU6FETIUSMRVKxFQoEVORGGIqllipUCKmQomYCiViKpSlqVAipkKJmAolYioS  
Q0yFEjEVSsRUKBFToURMhRlxFuREVCgRU6FETEViiKlQlqZCiZgKJWIqllipUCKmQomYCiViKpSl  
qUGMMRVKxFQoEVOhREyFEjEVSsRUKBFToURMhRlxFYkhpKJmAolYiqUiKlQlqZCiZgKJWIqllip  
UCKmIjHEVCgRU6FETIUSMRVKxFQoEVOhREyFEjEVSsRUJlaYCiViKpSlqVAipkKJmAolYiqUiKlQ  
lqZCiZiKxBBToURMhRlxFuREVCgRU6FETIUSMRVKxFQoEVORGGIqllipUCKmQomYCiViKpSlqVAi  
pkKJmAolYioSQ0yFEjEVSsRUKBFToURMhRlxFuREVCgRU6FETEViiKlQlqZCiZgKJWIqllipUCKm  
QomYCiViKpSlqUGMMRVKxFQoEVOhREyFEjEVSsRUKBFToURMhRlxFYkhpKJmAolYiqUiKlQlqZC  
iZgKJWIqllipUCKmIjHEVCgRU6FETIUSMRVKxFQoEVOhREyFEjEVSsRUJlaYCiViKpSlqVAipkKJ  
mAolYiqUiKlQlqZCiZiKxBBToURMhRlxFuREVCgRU6FETIUSMRVKxFQoEVORGGIqllipUCKmQomY  
CiViKpSlqVAipkKJmAolYioSQ0yFEjEVSsRUKBFToURMhRlxFuREVCgRU6FETEViiKlQlqZCiZgK  
JWIqllipUCKmQomYCiViKpSlqUGMMRVKxFQoEVOhREyFEjEVSsRUKBFToURMhRlxFYkhpKJmAol  
YiqUiKlQlqZCiZgKJWIqllipUCKmIjHEVCgRU6FETIUSMRVKxFQoEVOhREyFEjEVSsRUJlaYCiVi  
KpSlqVAipkKJmAolYiqUiKlQlqZCiZiKxBBToURMhRlxFuREVCgRU6FETIUSMRVKxFQoEVORGGIq  
llipUCKmQomYCiViKpSlqVAipkKJmAolYioSQ0yFEjEVSsRUKBFToURMhRlxFuREVCgRU6FETEV  
iKlQlqZCiZgKJWIqllipUCKmQomYCiViKpSlqUGMMRVKxFQoEVOhREyFEjEVSsRUKBFToURMhRl  
FYkhpKJmAolYiqUiKlQlqZCiZgKJWIqllipUCKmIjFbtmwxDXV13guVm2984xvhLYH227Fjh6mv  
r/fulhtiPTrcfSE1cuRI7265ueaaa8JbAu23b98+M2bMGO9uuZkzZ054SyCeM844w7tbbi699NLw  
VkA8kyZN8u6WmwsvvDC8FRDPOeec490tN1OnTg1vBcQzffp07265OfPMM8NbAe1X1b9/f8MwHZ0B  
AwaYE044wXTp0sV7oXLTq1ev4Ha+j2eYgmN3ZkDfvqBW7k/v6mrT0+1Szrjf69OzZ3DbYHz3wzAF

xl2X+tX2MzW9a0xVT3u96pUz9vd69OlhBvS3u2jHdx+dfWpra02/fv3M6NGjg7+4TJkyxXz5y1+u  
iHF/SXOvehk6dGjwecr351dOZr+69u5ael+O7dF22wrdL0Y3br/c//92O6Zbwf066tij2m7LfjHt  
nGC/+hffr+7Hdm+7LfvFtHMy16/u3bt7/+7oxv03dxt3W999MEyhyexMsf3q1q0b+8XEnP7m/wAb  
H7VxJCnhKQAAAABJRU5ErkJggg==

"

width=500

height=300

>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[4]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="c">#code from project1 see file in repository  
named: agentO.cpy</span>

<span class="kn">import</span> <span class="nn">random</span>

<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>  
<span class="n">Agent</span><span class="p">,</span> <span class="n">Environment</span>

<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>  
<span class="n">RoutePlanner</span>

<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>  
<span class="n">Simulator</span>

```
<span class="k">class</span> <span class="nc">LearningAgent</span><span class="p">(</span><span class="n">Agent</span><span class="p">):</span></span>
```

```
<span class="sd">&quot;&quot;&quot;An agent that learns to drive in the smartcab world.&quot;&quot;&quot;</span>
```

```
<span class="k">def</span> <span class="nf">__init__</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">env</span><span class="p">):</span></span>
```

```
<span class="nb">super</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="o">.</span><span class="n">__init__</span><span class="p">(</span><span class="n">env</span><span class="p">)</span><span class="c"># sets self.env = env, state = None, next_waypoint = None, and a default color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">color</span><span class="o">=</span><span class="s">'#39;red'#39;</span><span class="c"># override color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">=</span><span class="n">RoutePlanner</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="c"># simple route planner to get next_waypoint</span>
```

```
<span class="c"># TODO: Initialize any additional variables here</span>
```

```
<span class="k">def</span> <span class="nf">reset</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">destination</span><span class="p">)</span><span class="o">=</span><span class="bp">None</span><span class="p">(</span><span class="p">):</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">route_to</span><span class="p">(</span><span class="n">destination</span><span class="p">)</span><span class="p">(</span><span class="p">)</span>
```

```
<span class="c"># TODO: Prepare for a new trip; reset any variables here, if required</span>
```

```
<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">t</span><span class="p">)</span><span class="p">(</span><span class="p">):</span></span>
```

```
<span class="c"># Gather inputs</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span>
<span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next_waypoint</span><span class="p">()</span> <span class="c"># from route planner, also displayed by simulator</span>
```

```
<span class="n">inputs</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>
```

```
<span class="n">deadline</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>
```

```
<span class="c"># TODO: Update state</span>
```

```
<span class="c"># TODO: Select action according to your policy</span>
```

```
<span class="n">action</span> <span class="o">=</span><span class="s">"&#39;forward&#39;</span>
```

```
<span class="c"># Execute action and get reward</span>
```

```
<span class="n">reward</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">action</span><span class="p">)</span>
```

```
<span class="c"># TODO: Learn policy based on state, action, reward</span>
```

```
<span class="k">print</span> <span class="s">"&quot;LearningAgent.update(): deadline = {}, inputs
= {}, action = {}, reward = {}&quot;</span><span class="o">.</span><span class="n">format</span><span class="p">(</span><span class="n">deadline</span><span class="p">,</span><span class="n">inputs</span><span class="p">,</span><span class="n">action</span><span class="p">,</span><span class="n">reward</span><span class="p">)</span> <span class="c"># [debug]</span>
```

<span class="k">def</span> <span class="nf">run</span><span class="p">():</span>

<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of trials.&quot;&quot;&quot;</span>

<span class="c"># Set up environment and agent</span>

<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span class="p">()</span> <span class="c"># create environment (also adds some dummy traffic)</span>

<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span class="o">.</span><span class="n">create\_agent</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">)</span> <span class="c"># create agent</span>

<span class="n">e</span><span class="o">.</span><span class="n">set\_primary\_agent</span><span class="p">(</span><span class="n">a</span><span class="p">,</span><span class="n">enforce\_deadline</span><span class="o">=</span><span class="bp">False</span><span class="p">)</span> <span class="c"># specify agent to track</span>

<span class="c"># NOTE: You can set enforce\_deadline=False while debugging to allow longer trials</span>

<span class="c"># Now simulate it</span>

<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span class="p">(</span><span class="n">e</span><span class="p">,</span><span class="n">update\_delay</span><span class="o">=</span><span class="mf">0.5</span><span class="p">,</span><span class="n">display</span><span class="o">=</span><span class="bp">False</span><span class="p">)</span> <span class="c"># create simulator (uses pygame when display=True, if available)</span>

<span class="c"># NOTE: To speed up simulation, reduce update\_delay and/or set display=False</span>

<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span class="p">(</span><span class="n">n\_trials</span><span class="o">=</span><span class="mi">10</span><span class="p">)</span> <span class="c"># run for a specified number of trials</span>

<span class="c"># NOTE: To quit midway, press Esc or close pygame window, or hit Ctrl+C on the command-line</span>



```
<span class="k">if</span> <span class="n">__name__</span> <span class="o">==</span> <span class="s">"&#39;__main__&#39;</span><span class="p">:</span></span>
```

```
    <span class="n">run</span><span class="p">()</span></span></pre></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="output_wrapper">
```

```
<div class="output">
```

```
<div class="output_area"><div class="prompt"></div>
```

```
<div class="output_subarea output_stream output_stdout output_text">
```

```
<pre>Simulator.run(): Trial 0
```

```
Environment.reset(): Trial set up with start = (6, 6), destination = (6, 2), deadline = 20
```

```
RoutePlanner.route_to(): destination = (6, 2)
```

```
LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5
```

```
LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5
```

```
LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0
```

```
LearningAgent.update(): deadline = 17, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0
```

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (5, 6), destination = (4, 2), deadline = 25

RoutePlanner.route\_to(): destination = (4, 2)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'oncoming': None, 'right': 'forward', 'left': None},  
action = forward, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 0, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -54, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (7, 5), destination = (8, 1), deadline = 25

RoutePlanner.route\_to(): destination = (8, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -45, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -85, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None, 'right': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (2, 3), destination = (7, 5), deadline = 35

RoutePlanner.route\_to(): destination = (7, 5)



LearningAgent.update(): deadline = 35, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 30, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'right'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -38, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -39, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -40, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -54, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -55, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5



LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -75, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (3, 3), destination = (8, 3), deadline = 25

RoutePlanner.route\_to(): destination = (8, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

Simulator.run(): Trial 5

Environment.reset(): Trial set up with start = (7, 6), destination = (2, 3), deadline = 40

RoutePlanner.route\_to(): destination = (2, 3)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 6

Environment.reset(): Trial set up with start = (4, 1), destination = (7, 5), deadline = 35

RoutePlanner.route\_to(): destination = (7, 5)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': 'forward'},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': 'forward'},  
action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -45, inputs = {'light': 'red', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -85, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 7

Environment.reset(): Trial set up with start = (8, 1), destination = (3, 2), deadline = 30

RoutePlanner.route\_to(): destination = (3, 2)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 0, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -7, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -39, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -79, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -80, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 8

Environment.reset(): Trial set up with start = (8, 1), destination = (3, 1), deadline = 25

RoutePlanner.route\_to(): destination = (3, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 12.0

Simulator.run(): Trial 9

Environment.reset(): Trial set up with start = (7, 4), destination = (1, 1), deadline = 45

RoutePlanner.route\_to(): destination = (1, 1)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5



LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'forward'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -48, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

```
LearningAgent.update(): deadline = -98, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0
```

```
LearningAgent.update(): deadline = -99, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0
```

```
LearningAgent.update(): deadline = -100, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5
```

```
Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.
```

```
</pre>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="cell border-box-sizing text_cell rendered">
```

```
<div class="prompt input_prompt">
```

```
</div>
```

```
<div class="inner_cell">
```

```
<div class="text_cell_render border-box-sizing rendered_html">
```

```
<p>Question 2
```

QUESTION: What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?

OPTIONAL: How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?

```
</div>
```

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[15]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="kn">from</span> <span class="nn">IPython.core.display</span> <span class="kn">import</span> <span class="n">Image</span>

<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span class="o">=</span><span class="p">(</span><span class="p">(</span><span class="s">&#39;stateChange1.png&#39;</span><span class="p">))</span>

<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span class="o">=</span><span class="s">&#39;stateChange1.png&#39;</span><span class="p">,</span><span class="n">width</span><span class="o">=</span><span class="mi">500</span><span class="p">,</span><span class="n">height</span><span class="o">=</span><span class="mi">300</span><span class="p">)</span>

</pre></div>

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt output\_prompt">Out[15]:</div>

<div class="output\_png output\_subarea output\_execute\_result">

<img  
src="  
jwv8YQUAAAAJcEhZcwAAFPsAABHWAVNM8g4AAP+ISURBVHhe7J0HnBRF9sdnExkRFQOYBcyKgpJO  
gGL4nzlnUdQlGg+9O3POZw4YzqxnlGJMYBKEpRglOecc4Z9//p11ZupHXpnunYbpmDetz5vp7u6  
u/Z1dfXrX1enBAmxsXbtWjNUNrKyMu9sYyRorD/27Uc9v/8yyBAEn7B3rMSnn39MayfuQE8/clyQ  
ceUOben9maoBj1yhxgZR684dqNRt9NotcxJWxwazPP5H/3p3oc+p3ubnECP/IVGcye8R1NU/Int  
rqP+86TR5zsdO3YMrCK4AU6cODH4rYhLLr3UDGmCyHr+2XvS5MmTgwxB8BXRrMJGgzRWYaNh/TTW  
RIRin3yS6Nhj9bywVavMBEwLFmZAwdOZisr+/nuipUvNSCWp4Gy+HHvTbTDDmYkC40a6d+VK/Uv  
GDTIDCgqWrej9a/mfjiC6JOncylgcsLK7NWLf1bEXPmmAFF1676194mxxxjBhTp/4NB3k036XUc  
MsRkxkflf6wCW29N9L//aafR4HiFrruO6Lnn9LDNH3/oX1552M03r9tY00He5pvrMnnjo7GGgXlb  
tUqVz3mgXz/9C049lejZ4lGjCg/bxg1ahAVFOjhl1/Wv5gfO+Dzz+tx5uSTU+VhZ7D9Tf8fGK9d  
WzfWe+5J1eHIkUSHHbZuHaKx4n9yPv8fG4zvuadurEcdRfTPf6bmOf/88svaYNtweQ89tG5jtcH4  
iSem8ivaFIUk7b9uQO67j+jww82IAStbr54ZyTHNmHdcYcZceTii82ABdbtgAPMilfMnUv0zDNm  
xAJ+169vRtLo0sUMhNC/vxmlj/XbWEtLzUAG2rWldvi1QQQU8o54G2uDBnpP3HJLPd6+vf5ds0b/  
YlqvXvqXQRRq0kQPn366/sV02KJFWvfxOosqaax5idVqBMFvctdY7ejKcB4iKp+4CIlh3sb6/vu6  
wR1xBNH//R/Ry4/pcZxN4veBB/R8CxakGibYf49/nj9KwgWVosRBL+RxipsNEhjFTYaYmusCU/b  
/RKVWqkkRCfObRlvWTGQUcdGQbJPmirJ0UcfrctTNioxlt5M/l/urnkvPVD9IVqWWEmf3t7dzJmd  
3r17U6K2Kuv7BM2cOdPkVh7264knnjA5lYfLglUF3BsaV1mgoKAg2Jb4rSqBT6ZdxPF0QuyNtUqW  
llar+mrfvj1NK5xJq5Wnpe1KaXziXviyFZlprKHTolpFKWzebFZRCps3mz2jLCxhncPmj2pIYfmV  
MVPWkaecYlpL5Ymlsd5yyy1Jp6rKuHHjkiv6U4+f6JffBIH/XwbSz9/0oT7vWzeeRAQbruCHqkcJ  
EESdGNbRjoaff/65ya08XNZhuNmligRlxbQtd95559jKAvGUooBTcdGiRQtaw5doQ/ht+9/MENHP  
iZ/NENFPiZ/MUIq4NWtcFQ+8LSvGbRlvWTERZ2VlaqxrFqwJGmifRB/ql+hXrrH2TfQ1QymWLFGN  
FbclxkReNFZfyzK/VSZOpzJGVqPTJ/97MvVPmNvQkFeBfpgG6o63ZZnfKhOnU9lkgAvSWN3xtizz  
W2XidEoaqzt5UZb5rTjXoiWN1Z28KMv8Vpk4nZLG6k5elGV+q0ycTkldScvyjK/VSZOp6SxupMX  
ZZnfKhOnU9JY3cmLssxviYnTKWms7uRFWea3ymzly60uxN5YY1xPb8uKtYHFWVYMHnkkYFTcazk  
119/HZQDi+W2MINWHBQk9O1zVSV5I4tKXTK9KClIXNZ2221ncioP11ccdVa3bt3ArzjKArGUwpVI  
r2ilrKIUNm82qyiFzeticZTB5ltZYWmQsrB5oxqS+vXmFkHbqarStm3b5lpWIWXLIsVWFgjKUqmf  
/Y6sSsJl1aITx+RUHi4rjvXk2yDjvvk6DulpRQGn4kl0qzuxlhVT4wLxrmNMxLmC0hvgTl6UZx6r

TJxOSWN1Jy/KMr9VJk6npLG6kxdlmd8qE6dT0ljdyYuyzG+VidMpaazu5EVZ5rfKxOmUNFZ38qls  
81tl4nRKGqs7eVGW+a0ycToljdUdH8vao8llql48gwoTVb9sDmJbQ2ms7mqzZRUmZlG1lql69eL5  
RGpsayIN1Z1NvawWLW5SkbWMjj/uRpNTNWJbw+CyWjzRnlqotEalOlj1jSxq/fLiEmIMZSUSn1Bp  
aTs6oNn1JqdqxOLV1KITg4rfrHgzk1M1EgNVaQVVdy24Fa+mKuu7GG83jKmBFRcXx1bWVlttFfj2  
EF6HX0WKioqCskpKSkxO5dmi7uQgspYkFpkcUuWaAYXrv4iltngjBr9xGBprUVpeZc28RZDvJqqS  
JZePcz2rWtZ/IJUpS09h80axsDRAWdi82Swt/Z2OppNOQlvRH2fEL2zsWN2GdtxR/2Yi9sZaVYKy  
TGOtKs2aNUs21ufTP1PpylIFC5LrWVhYaHlrT1x1Zr+RsKplgfVXlj6yqcBNV16JaXqYiXLfeNU9  
MiiXzFDV8VazKujcz1jLiqFxmD6WZX6rTJxOSW+AO3IRlvmtMnE6JY3Vnbwoy/xWmTidksbqTl6U  
ZX6rTJxOSWN1Jy/KMr9VJk6npLG6kxdlmd8qE6dT0ljdyYuyzG+VidMpaazu5EVZ5rfKxOmUNFZ3  
8qls81tl4nRKGqs7eVGW+a0ycToljdWdvCjL/FaZxG3xORVrY12IGutbMTbWGNczcb+njSJOv+6L  
sSzzWyXWrl1LBSq9/NTLJqdqoOLjqvygrO6J4DaKtWVINGv2bFqrhpetWBFMdwFIYT3j4IADDICV  
H886TpkyJfBt8eLFJqfyzJs3L/Arjg8zz58/Pyhr4cKFJqdqxFJbQYNAUr/7778/NW/evHKmUvDW  
OmM1/qgR5IXOG8ECv8xdV/Or16l2V1xB3erXp9XVq9M5559Pq4pLQpcLs6Asaz3D5olqjRo1iqes  
95pTS5XsOI051Pxwax5Hi2sdYXZZsba0/xWieBeT+NUVffuYAWN/etf/zK5ISMoxzTW6Ynq1K1o  
C/qyYSO6qvXxNL+kOo1S06Ny7LHH6vLMelaFL7/8MrayQFCWMt/e/BdnWSCeUhRwKi5i1axyi6Az  
3q6j+a0ycVbWBusNqF0bjhNtsQVRtWpEHTsSvfCCnIbB+sTaKKQsJ2IrKU6nNIhj/eQTOE50yCFE  
Bx51lK6dmaCo4JAqDcyNWMsyv1UmTqekn9WdvCjL/FaZOJ2SxupOXpRlftMnE5JY3UnL8oyv1Um  
TqeksbqTF2WZ3yoTp1PSWN3Ji7LMb5WJ06k4G2vjXd+mLevfasajjQwn2Ity/xWmTidiqux9uv7  
DV14wXl0yskXmpyqlw3MjVjLMr9VJk6n4oysNUqISqr+ApUk0sDciLUs81tl4nQqrsa6atUiateu  
HbVvX2pyqk5s62le+xMXeVGW+a0ycCqOW8Hw7qZddtmFXnnlFZNteYaPGETVi5ZRtaKeJqc8ri8W  
fP3Z11WFxVBIN91E3377bVBngQu//hpkV4V4G4WvZVWRtSolCpVLxmbNmRXk4R5XZ0sra4+99qha  
WWoDJsVdsFpbvFnt+nS8JlL7j9cu/uc/2NgYSyvDMr28Mit1GzZPFMM/euChB/Q/RHnKN9xfGzZv  
VivDkqoYXsegrLR5HMz+WndV1pHNLisOYiklcMrYkplLTG7lCD6TbsrabrOqfZLc9kubfnPdu+/i  
DYNmJgXuY8G0TCTLUKkqt+KV/fkn3fvve+lK/ENTJr32gZlaOVLrV/XNyevoW1kgnlIUcTkEvO5n  
jaPKbtSvLQ/q7KOP3PVICHWWfyzrali3rJlJm8YaZ6OQspylraQ4nZLG6k5elGV+q0ycTkljdScv  
yjK/VSZOp6SxupMXZZnfKhOnU9JY3cmLssxvlYnTKWms7uRFWea3ysTpIDRWd/KiLPNbZeJ0Shqr  
O3IRlvmtMnE6JY3Vnbwoy/xWmTidksbqTl6UZX6rTJxOSWN1Jy/KMr9VBt8QjYtD8NKJGDn++OPN  
UNWJcz1jvW4eU6PgO6/iAmXF8ZFnlEtXcChl6nfRotQXkCtDUJaxPXfd0+RWDrssWfVllmPWsy08  
8sgjsZUFkr5twmWBWEoJHDIVP2rUKJo7d27ITKXEFqocY00ObhLkhc6bzVDW5qmy8OHilDds3giW

rHiznmHzRLXbb789nrKWzw2+cWvXmcoNnzebzZIL81Rap6zV1jxRTZWFZJdVptJslapCPE1eqBAC  
An2zjZHElCID6e1n9jCjguAniamjbiSavgd17/6FySK69c8ldP4139Ofa5fRzV9MUDk9qbTB2dSp  
fQc1PJgua34LXV7jBDU8j7p+PobWTn+Vel+foOfatifMLeQvb731FnXEK0OryMMPP2KGNIm3ntmd  
Jg460lwKgp+IRhU2CqShChsF67+hJkL+xddfm4EKOOccM6Cl8qkf/A/7/4T9z2ysWmUG0ggr97rr  
9G8m7rvPDKSRybceddiA68UQzkgU8jrvrrmbEsHKlGbCoU4fo8cfNSAU8/7wZUNSrZwbSyHYBhddr  
t92lLriA6MEH9XhMVGKLVsAff5TfCGecoX+Rd/75RC+9pMcB8IDRYWy7rW6omTYoePZZouee0/OF  
zct5p5+uf21GjkwTbZv8cKLq1cuXM3Vq+XHA86Ohfv450Wab6fFu3YiaNNH+2KBcG14eVhF4acE2  
2+jhq67Sv5gfG56/acCgoelV8nbdhrHnnkSnnaaHuXGjzBtu0Mumv4Ak3T+sJxpqmN+bb07qTJzo  
ssulxo4leucdPV/YvFUG3tKiUtEbRoYnk//eflfatDEDMXDTTUTFxWbEgYr6KT/9lGjvvc1lFYga  
cV244w69Y4YxZEh4vS5bZgYq4KGHzEA8rN+GutNOei/LxM47m4GIPIJeUd8DXXUKH14uPdek6HA  
YcY+DIQdDvbbj+jaa4mOO05PX7JE/556KhE+tWjrVeRLQ81LQlqOIPiHPw3VPrtH5AyLvKLeEl9r  
QMOaPFmf6f75JxFu+7v0Uj0NXR54DR+fJdvMmpU6y2Vwpi4NVbCQ1iBsFEhDFTYKpKEKGWwXNdSE  
p23eV798ZheV4iKu+o+lFH60Yit+Z3kVCMpSVpQopqmJWtQs8Qc9WPdhGl4wit5NuL3ZGW+Whl9V  
ecM0w37BqnqXvF1WVbnwwgtjK2vw4MG6rHEJGjdunMmtPEFZKv3rX/8yOZWnymvHIRQkHq6Mfa3M  
SqTa1vLEKiotLaXV8FLZZZ0vC182kyGF5btYRSls3mxWUQqbN5tVIMLmdTHVUEPzK2PGn6pS5RIQ  
Xdihn3/+2eRWHl7Bs885m35+5Wd69s6XaMiQ3+mX7r/RwD8GmrmiEWdF1a5dO+lBVXniiSdiK2v2  
7NIBOXEcNUDgl2qocRB8z0GlFVHugMtCPB4p4FBcuGzAVdP17XnzP5kf/KaTK7+yEWdZ5557Lk1H  
P3VMbLlaFWyljdgV0S/4XTFB76Fz359Lq6atov6J/rTgiwVBXjq+Ni6vG+ouMTbUmNYzttrakBtx  
zQL9+p/53cOjq12vjSvOsqShOpAPDcLXsqShOiCNy404y5KG6oA0LjfiLEsaqgPSuNyIsyxpqA5l  
43ljzrKkoTogjcuNOMuShuqANC434ixLGqoD0rjciLMSaagO5MOlylJXMcayzlVpukpxEWtDjWk9  
q1yKfVMKxv1dVYKyLDXBm0eqCJcFqypFRUXBOsZR1imnnBJbWUOHdQXEewkq2K4glpf0xlvfgG9K  
wY0zVaXKHvGKBYmHK2Npt/klU9i8USwsva8sbN6ohhSWXxmLsyZVUBPbpuW5WkUpbF4XM2VUIVh2  
HXZo9913NzmVx17JqhLcOK3Kie3G6bgqPcay7rvvrmRDrSpr167VvsXgF+D1/OADtxvew4jHlwUc  
iou4KgrEWlac6xhjWaJRHfC2ceVBWXLW74A0LjfiLEsaqgPSuNyIsyxpqA5l43ljzrKkoTogjcuN  
OMuShuqANC434ixLGqoD0rjciLMSaagOSONyl86yfG2oJQVExTgtZmy1JY3LjU29oeK+g+pFRNWU  
/fXnNJNbeWKrLWlcbmzqDRVUL4aVKd9mmZzKE1tt+Xp50euyYvoia8NtUOHUmrXrjQW32KpeX5r  
XhxsscUWQVnd8JGxKlJSUhKUhfGdGVZVWrVoFZdWvX9/kVJ7g5g+VaiVqmZyqgegcR4RO3rKpLI5b  
BqsVzqOSwn60bKn+4NrEicFPpajy2vGKBYmHq2pSVjSrKIXNm83SuoFKwVDYvNksLKmWhm+r8bfw  
EGRffx3thyjKWyljCYNJ5/Bfq8jGUFZVo03sfhmrKsEN3aasLI26mNzKY/umDpQ0YYLOP/po3VBB



LXVQmT+faEH4q8OSVH3tDModM1R1pCw3pB/VgbgcAlKWG9JQHZDG5UacZUIddUAalxtxliUN1QFp  
XG7EWZY0VAekcbkRZ1nSUB2QxuVGnGVJQ3VAGpcbcZYIDdUBaVxuxFmWNFQHPhHG5EWdZ0lAdkMbl  
RpxlSUN1IHGfwp3ifk/Lui2+smJvqM/G2FBvj2c9YykFH+uN69o1fzh2+fLIJqfyLF68OPALG3HJ  
smW01uSvWr2a1jreWBJ8ylGIOPwCuDvp5k43m7GqgfqKLXKhrJHxIYX1jIMqe1SrVi29ckjqt3nz  
5pWzvs3plJUSg1Q5xlQuNd/fmsfBWRokfTrjZltaG31avoDwNWRB1barh3t3bJl6LjhZq9jS4fl  
wswua6+99gqdJ5KpZNdX0eAiXWdh80awwC+Y+Whv2DxRLVmWWc+qUuUSXn311XgdMisY5xv4imrU  
oJ+rbUHvbb8T3X7o4fSvo47BC0/NXNGIfR1NWZMnTza5ISMoy9gVV1xhcitHtWrVdFmqoelG9qpW  
xhln6LLiqjPzW2XgUFzEsWJMrH55WlbsGIU+2huNDVIW8vl9rv/4hx4vLta/P/xANGiQnmbh6zrK  
Wb8Dvm7ECsuy8zEM23HHVH779vrXwtd1lIbqgK8bMR/KkobqgDQuN+IsSxqqA9K43lizLGmoDkjj  
ciPOsqShOiCNy404y5KG6oCPG/HGLs9TtaKuNCikq6kySEN1RxpqBloSq+n888+laoUmo4plIQ3VH  
GmoEjrq5uBtcg23e8/kVA1pqO5IQ41As/1fCm5EqVa40uRUjbxpqFtsog31448/DhyK4w1wlK6V  
q148Loio1Rlx+RVHdam6AljHqVOnEnXtGoxXFtQ57uZ69NFHdUYVefvVt2nLhVuasarxyvOvxFNn  
iiqVslaltle2pUShcscY8vBaRWcr03eLxIKWsvSyZsxA4yB6+WX9a9vAgfo3rBw2NKwgYUZF2DyR  
zPjF/3zIsCHBb+i82Uwlex3r1qtbpToL1hFmbvMLmyeqJcuy6qwwqVLmEi06/KOkUPntdVZlRGMfK  
WWVNN6GjKhpqSUnQNgK7444gm447Tv9WRFBOHJU+baEuQ9mZBQkaPnC4mVA5gk+NwzdjVSFZjmmo  
VSFZlkqx3LJpfqtMVVfMxtuy4qguyKMRIJ+qcN2VYldo8ptftGQstylvaHuJGf9kZCy3JDukQek  
cbkRZ1nSUB2QxuVGnGVJQ3VAGpcbcZYIDdUBaVxuxFmWNFQHPhHG5EWdZ0lAdkMblRpxlSUN1QBqX  
G3GWJQ3VAWlcbSRZljRUB6RxuRfNwDjQHYjrmi6la+X4I7RxxgXWM7VbGsvj8uvjii2nlynjuuQWN  
Gzc2Q1XHm4aKG5PhTJBicCooy1hVib2sONfRIDV8eNXungrKMrZZ9c1MbuWwy4JVhTjLAIUu4cor  
r9TOmlqfO3du5WzVXFqqUmlVY4xlUtz51jzONhCldYpa2X5eVzMXkfcxbk2T1Szy/rxxx9D54lk  
KtnrWHenuno9w+bNZiirTqqSRHVTZ2HzZjOUtZIVljLkVYWqN3VBEAQhSWLt0m9pxZhGRLP/Tn0+  
3Z1+/nR/+vh/N5jJRHMMzqDRo0bRqkWzgvHJ0+bSqIWzafKChbRKjY/EtDVEc6fNotFjxtKMSWNo  
zjLMuZRGjJyIAVoxf4r6u5YWriyJBQsX0+zx02n+5Jk0YsTIYPrCmZNpylAXaOfaovEjR6qjtCpv  
4RJatXwh4exl4RL8JOEQhMqR3g8ycaKOTZWhS5cbac6cOTRkyBCTU57EwYccSJ98+Q39PGAgIR55  
FD3Q5RozSXP51mfTsGHD1FAZfdIOC9ulcybStLGv0iQ1POz3P2nO0IU0fQKC7hJCV8m0UQiiS9Ry  
Q9WvZuLY6URfTikNTxtxCSaMxZLrwkC6PSJo2jC7Hkq7BKNGDqEFgTrv5SWqN+Jw8djRBAEodlg  
qA4dOjRpkyYh/rhTt25dM0T06WefmaHyyOm/IAhCjEhQFQRBiJH8CqoV3V+yVr8RKPkb030osZUD  
uKw4y4yDivzJhZ+u/9O3uqwI9nN9+utS9obwZyPGz6B67bVE221HtOuuRJ07E/3738pT5erDD+tf  
vBgHv+DUU/Uwv2bqsMP0p84A8r/7Tg+DBg3MgOKii8yAgcvj3513JtpvP6KaNYm23pqoRQuiOnWI

mjdPzXPSSXr4zTf1OOBp6SAf9sQTRHvsQbTXXkRPPpmaxuV+/73OywTqBQa/jj2W6MQT9bofdFDq  
/19yCdGWWxLVr0/OzjtEv/2mp2E9eB78Nmumf9FXVL26fj0Xfp99Vudj+b33Ti2Dd8zh7VK77ZbK  
Y8LGUC9duhDtsAPRddcR4SObyLd9xS8+VwcfLr+c6PPPCZus9g3jFXH44UT16unluazLLiPaZhvd  
fpCH94/jF9sSv6NGpdbt6qv1MoCXZzB+771EF1yg6/foo4kWLDL5dvvDL+oCH+IEe8D/a9RI1/MR  
R+h5wujUSW9DgDKwLLwmXo2GYfxv/J5zjs4HGB882lwoMP7FF/p/Pf44PlesxQHy2c8VK/QvvqyG  
etl3X6JZs/S2Pfjg1H7Rty/R3Ay3E6EMbKfztdPDoeKTXV7t2tqOOUb7YO8nZ56ph195RY9nAv5w  
PYMDDtAxxNcB8K+n+Okddq7NN9cB7Z57dB4q8sYb9S/vSkvMXQEYfuQR/YvGjcbD+XaQatjQDCgu  
vtgMGDBv+i8aHQLpttvqnYQbN88DMMzBEDjTbDj/mWfwyVy9bt266Tz+X/i1DwIVYUAAb8Q/Hin  
b9VKT+NI8FVf/B8E2x49iEaP1tNQPzXP+i92bByk7KDKN9HyPF9/rYPqVlul8phM4xhGUJ0/Xw8j  
4PN0/v3gAz2MoFpYqNfx9tuJcKc4z2MDf085RQ/zdARr1C0CK/l4qAL8cnvAMNaV4XkYexzDCKoc  
rMJ8HzFCD//6q/7dbLPUGZvnsUkPqigT68vwMvjFwYXBUH3V2S4bwwhoPGxvo/T5Zs/Wv9jPDjxQ  
5yOozplDdOeduj2mg/n/+is1zL8oA/ss1hB9fjjU9MYDGMfBbjwbU9LZ+bM1PR+/XR7RDvm9c60  
rAf47V0UKnPalghCeWTfiA0/g+oLL+gjLsw+6uFoJiMWj/NLjvnIhSM9jpoYf/11/YsjLucB/A5N  
3eoVIAFvtHixnpauQrbfnuj++4n++1+ttJD36af6NOirr7QaxBF6n31Sy0ApT5hAVKNGSoFC+QmC  
sMljoodnoD8Gp284jbMDHPpbePwf/yjfdwoQHfFqjED51FM63z7d41/79AndAFhm+XI9zf4f/AvQ  
7/R//0d0332pfPQtlaCmB23Ap4rchwl/BEHY5DERIE+QUxxBENYzfgZVKDtcxcVPz59Tlel/Msd  
6lCnyIMtXEj0wAN6GYzzRRq+wAX4N501a/Rv+nR0wJ9xhh6uaFIBEPleP6MDghb6LXm4TRv9u2yZ  
vhrNQQ2/9hV0vtVj1CgdVBFEAU690b+JWz/sZTPB02+4QV/ZRLBFHq62o99VEAQhhCyRRRAEQXBB  
gqogCEKMeBtUy4KX/gmClGxceBVUV6xYoRzSKXgPoEfsu+++hLefL8etVx4xd/ncol6OWnuUyfEH  
1NfLeHzYI/Adj+Bt9srW8EVJTzjttNMCv+L81kgcrF2+NmhjO5ftbHL8YO0y7ReST3jhTadOnZIN  
PZnMOD4HkUuSfln2asHbNLxgNC1OLKPSTqXUvn17ml40S9emsUs7tqEOR3ak4Qn9lu64SfpTWxnS  
92ZcWS5J+hViStbkSYj5X0/d8VSoT7CJv1X+hcVxEoyTrF7CXGjNEeX8QRqXGs8l6/iF5IFfjBde  
XHrppRVWUq7VRNlvyzpuCTX1Kv6ZuhV2p3PPO4+uPv9qGlVnLP217Qjqs0U/mlBvEpU27kRLCpbT  
OKI/TEnxkvRHgmpmzP967JbHqN2CjR+U2xehFyQKQv2qnqhu5sgN5fxB8jyoFvATljnGj9BuwZXk  
Gy1atAg2XK6VczrBB+hUaqWSb3DD9w1f/cL3cOHXjBkzTI4/ol3topJvwC8kn/AvqHra4Dmouijn  
nxi/0ZTbptDSIUvpx8SPQd7oc0bTz4mf6ZdqV9DcD+cGw1PunhLMO+LvI4J5eiV6Uf+C/rRkwJjg  
PBNLliwJ/Gp1kATVqPjqFwVOD80Hhfw5etPAyqHm5LCaoRqUxQ7ZfoZ4YoCJ5rFq0J8laOW0n9  
E/1p7vs6qK6arl9hiDyw4KsFwXxlq7KfJyeDKr/6zyN83Za++uV9UMU7bj3Dx20pQTUilQmqYayc  
spKW/LKEVs2I5wuxEITd8dUvCaru+LgtJahGJK6gGjSVN3x1S8Jqu74uC0lqEZEgqo7vm5LX/2S

oOqOj9tSgmpEJKi64+u29NUvCaru+LgtJahGRIKqO75uS1/9kqDqjo/bUoJqRCSouuPrtvTVLwmq  
7vi4LSWoRkSCqju+bktf/ZKg6o6P21KCakQkqLrj67b01S8Jqu74uCOlqEZEqo7vm5LX/2SoOqO  
j9vSv6BapirJP7foolMO8nJHRJCHX8cee6zJ8YdgS6rtWebZBxd9bWMXqAS/5qvkG2hju+66qxnz  
B25jPuGFN1dccUXyiJNMZnz27NlmrtyQ9CvNdsd3/3NImE9suaScL2nbMpc88MADFfo1dOhQM1du  
SPr1XuAVJbbV43XxbbQckvQrxHJJOV84eeAX44UXHTp0qLCS7rnnHnrsscc2rE1QptLTKiWuV36E  
2NYPbB3ME6RHIIWVsz5MpSdVSlwX7leifYK9osdGKgsrYz1acjuGbMvnn38+dJkNYSeddFKFfnXp  
0iV0mQ1hqJOKX2IBtUaNGqHLrFd7SJlKj6uUuFr5EdbGOlptblaysHLWoyXrC8bJjPuAH15Y+Hpq  
dvDBB3uz0Rg+rYZfx13XDDsE8GWlNP/yFyEvzC6xx9wW5ju+22WzDsEz62Mf+C6g6qkvbxr8HL  
hSp3sB2xPX0jsb2fbezcf5wb+DV9tlyoikpib7Utd/JrW/oXVNXGg/mGBFV3fN2WvvoIV//d8XFb  
SICNiARVd3zdlr76JUHVR+3pQTViEhQdcfXbemrXxJU3fFxW0pQjYgEVXd83Za++iVB1R0ft6UE  
1YhIUHXH123pq18SVN3xcVtKUI2IBFV3fN2WvvoIQdUdH7eIBNWISFB1x9dt6atfEITd8XFbSICN  
iARVd3zdlr76JUE1GrjRP5FYQOeddwqVlpZS585XULVCorFjppk5cosE1YhIUHXH123pq18SVKOz  
265tgkBavbgssKOOak0TJ04xU3OLBNWISFB1x9dt6atfEISj0b//QNpi8/9Ru3Yd6lor2gdWp/bH  
1LLFHWaO3OJfUDXJN1qoBL/WqOQTS1SCX61U8g1ft6WvfkIQjcbatWupIDGESgp/pSK1GYuVVSv+  
nAYPnmzmyC1etKzzzz8/2GiBcTLjs2bNMnPlhoKCAu3LQOOX2ooYb9iwoZkjN3D9JGoHXIHiezOu  
LJck/YJxUsOox1xyxx13hPoF++2338xcuSHZxkIsI4T5w5ZL2leiooZUUVQAFRcdF4xzGxszhmjx  
YqL584mefz7lokWLiCZO1laTTV6FsWOJVqzQw23aEG27rZ5eFbwlqmeeeWayopKJx32xtKDqaUF  
1Uw76lawcv+fkzXdC/PBr/8oQzJvzMqawspYX5YIFaiUHBugLKyM9WER0t/paDWvjvivjung2a0b  
UWGhzsM4Tgr4HsDD111HtP32ergqWEXnnttuuy1ZOROHT/TidV7sQ4MJDQK/rii9Ihj3hbad2wZ+  
Nfi9gcnxg0vPujTwC8knnnzzyaRfA78d6FUbw5v1ETzOOuusYNwX2rdvH/hVu3Ztk+MHF198ceAX  
LBOo3q231sGzZUs9vj7xq8UrEitVJSnzjRarWgR+edmnqvqxqtdrDPIVPt6WvfkmfqjtRguqGxr+g  
6mElAbn6746v29JXvySouuPjtpSgGhEJqu74ui199UuCqjs+bksJqhGRoOqOr9vSV78kqLrj47aU  
oBoRCaru+LotffVLgqo7Pm5LCaoRkaDqjq/b0le/JKi64+O2IKAAEQmq7vi6LX31S4KqOz5uSwmq  
EZGg6o6v29JXvySouuPjtpSgGhEJqu74ui199UuCqjs+bksJqhGRoOqOr9vSV78kqLrj47aUoBoR  
Caru+LotffVLgqo7Pm5L/4KqSb7hbVCVV/8546tfXgdVIXZRyTd83JZeeNOvX7/U2404mSPQCn4v  
V45gP2yDrx9++KGZlzc/bHeUIVzqy1pQnFdGlqtPr241W708n4H0Es7N6Z7Dj+COh5zDN17yOG0  
sqgaUUkxCjAlxcu///3vIG+czHgu+fPPPytsY/PmzTNz5Qb2wzb4+t///tfMkRs233zzIE9I4/Rw  
06ZNzRy54ZFHHqlwW/qAF15whYRVUK7sa2UuKayMDWVWUP1H8U60tHpNWlOtOI12edvg+z3d6ten  
1dWrJ+3s88+jq1Q+FZeElxencQqblkvzwa9nLlmkWsrCytIQhmSCqlfGyYyfffbZJqrkDj9CuWKn

1TVq1Ei+q/Hf1/7bTMk9b731VrkNOXPmTDMld+B1cd999532CaI7gr744gszNcX8BQvM0Iajb9++  
Sb+wPX/66SczJffsvPPOyTZ28akXm9zcg21nt7FxeBlojkEb69GjR6qNjdRtziXJQ4dOrRcG+vV  
q5eZknu8CaqM2XxmzB/yrk+1Y0e84NOMWKg6ilqv29JXv6RP1R0ft6V3LSs4+jjsuBuKTerqP+r3  
009Tw/bvZ5/pYTuo8jRgD2fB123pq19y9d8dH7eldy3Lx0oCm/wtVajzTp2ISkqI7rnHZFYNX7el  
r35JUHXXh23pXcvysZKA3Kfqjq/b0le/JKi64+O2IKAAEQmq7vi6LX31S4KqOz5uSwmqEZGg6o6v  
29JXvySouuPjtpSgGhEJqu74ui199UuCqjs+bksJqhGRoOqOr9vSV78kqLrj47aUoBoRH4NqSQHR  
6aefQZ2vbktNGj9FhYnVZoof+LotffVLgqo7Pm5LCaoR8S2ozpw5jlq3voVqIBBVly6jkkKiHbc3  
9556gq/b0le/JKi64+O2IKAAEf+Uahk13e0IFVDHU7WicVRcMJm23eZLM80PfN2WvvolQdUdH7el  
BNWI+BZUx437k4oL/6lj/taRzjrrHNp1l6eppGCTmeoHaksGyRvQrkz7CtrYZpvpPE/wOqiqJl+p  
RsMbb7ih21ZYqM5pcwheHFG3bt1Q33LNJZdcQg0a7E8lRe9Q9SKiOjW7UVFRTfr888/NHJqffya  
McOMpBG2GsirytsWcdBJ1hMnNbzvvvm5kUco0cTvftuMAg/CpTtbvzDMH4DbrhB/25gUCcNGjTQ  
9ZVmueaf//xn+VfsjdOvJHz++efNHLkjWU+c1DCUtA8ve8nplrOqJFriilzf5vOr/yKkpSqliklD  
ilTAfflIPT5ihM7jafx7551EW2yhh5GHoBr6v12NU9i0DWTqT9KPWkv1uG12CsY3lLm++g+veAw  
rZ0MZ0kbw6r/jjz9eN+QcolqSH/z666/lK0uZL6T79fTTT5spuSXdl1hFHHgg0VNPEX3wASm1YTLX  
E3iJcUFCv14vSBn82iCY/z9t2rRydVXOL3s4B6T7dc0115gpuSXpE5lJqj5wwgknpPxCSQv4l8n  
Bt6lviH3qbrDDd4L+LTwlVeI0K2k6mztCy/oPE+QPIV3vGpjBgmqEZGg6o6v29JXv+Tqvzs+bksJ  
qhGRoOqOr9vSV78kqLrj47aUoBoRCaru+LotffVLgqo7Pm5LCaoRkaDqjq/b0le/JKi64+O2IKAA  
EQmq7vi6LX31S4KqOz5uSwmqEZGg6o6v29JXvySouuPjtpSgGhEJqu74ui199UuCqjs+bksJqhGR  
oOqOr9vSV78kqLrj47aUoBoRCaru+LotffVLgqo7Pm5LCaoRkaDqjq/b0le/JKi64+O29C+omuQb  
ElT8XVb+uqXBFV34BfMJ7zxhisnmdTwUUcdZabmjttvvz3lm2U+4L1f1rb0IVCE+eVDoHjjjTdS  
vlnmAxtXwMGDDBTc4sXNVSucqWgnzPbyF/9l0zvKwsrY0MZp7BpuTQf/PL51X8uaZCysDLWhzmm  
XOFFUMWLZTfffPPyFaeGL730UjNH7ujatWv5DauspKTEj5fhpvkF84GkP9a2XLZsmZmaG7C9GjVq  
tl5fxxxzjJkjd/ToOSNVZ8bwgnZpYxUT5teYMWPM1NziRwOJQhVA8BHLbsKGIZFe2VL5giAIlSfx  
Z9/2NGvuPFo0ch+aMXsmdXthBzNp/aBDtgRuQRA2LqIKzsSAz7aigd3r07Kxe9KY8cNo/p+7mklg  
Gc2eMZVmLSuj2YtWEa1dQeovjRkxguZNmkXL54yjkZNMUNnaJTRj8kRavHltjcKH1hSTR06khSv1  
7UdzlhEtmjFLDa2ltWXLalUKqosWzqKJsxcHeaNHjaFrLutKtGoBjVTLla1cFCw3d+4SWr1iYTAs  
CIJQWYYPH26G8A22yn/Z8oknnqRVq1bRYX87wuSsS6JHrz405Ms6VDZpJxrerxUliuqYSaAnjVJ/  
X7uoJn199tn0xcv30No/HwmmndN7mPOp+1ZV07pGtafXiDwkf7Oywe3v19ycavWgoXfCva6nTS72D  
ea848Erq1u1p6jX2PTXWnwbSEnplJtGzfzFhna7P9CtN1z+KrWsc1sw/5NDyqjHssl023VPU7uj

rg/yBEHITzp06EAdO3YMrE2bNibXjbiCKri87RX0zTffmLF1SfQf9A3dcte91P+3ftT/lmvoqquv  
NpNAT/py5ASaA3mqeHCg/h069C86dYd/0plpl2nMuMlKqS6i1Sp/2ljci7iUlqu/Y4cNoyHjZ2N2  
otXzA4U7YeQU9XeZmr6W5qmMeRMmK029moapeWdPn6+U6lwaMgRhngjiXxPU3+W0SHoKBEGolV36  
9VNxa2hggwcPNrnu/Pbbb2alaJ999jFD5Uk89djV9Mz9u9MLDzelguLNTHEYENgJgiCsg9xSJQiC  
ECMSVAVBEGJEgqogCEKMSFAVBEGIEQmqUeCbftfn02a+PMm2PvzldZIV+f+8bJzrsHat/s1WZmX+  
Z5x+ApfyeN64fdjl2DiCauPGRKWlZiSNhFqFH380I1koLta/RUVEixcTff+9Xr5LF6KGDYlWr9YW  
BuZLB3n9++vfVea+Mxv8v08+MSMRQDlokGH/6+9/JzrpJDOynlm50gxEOHp1ovffNyMWrvvr9eB7  
AsPWqbKkl4U6q8jnsP973nlEzZoR7bYb0c03m0yLG2/Uy02cqMfXh++ZysS0Dz8kevVvoolCityb  
ecghZqLft9/qeb/+Wo9XxU+8J7ii5dPzK7Of4L5O/GK/y8Rnn5UvA8OTJhFtsYUer1dP/3pMSA14  
ACry4INTIVutGIH9+kQPPEC0ww5EO+1EdNBBRHhHJ+bBznHyybrh1a1LNG2aXs7eONGwDz2kh+18  
wEH1gAOIzj+faP58Pc/uu6fmxW/TpkSbbUZ09tn6fyCvSRM9HbRtq31LB/PtuSdRu3Z6+lgj9C+C  
wWOPedWqpRsN8rKxcKGer2XL1Pz4xYEHZZxySirPnqd2baK99yaqUYPoppuISkp0QGzUSC+HIGOX  
x7/77qsPQi+8oPNQzoEH6p39kkt0Hvj119RyAEH18MPXLbNBA12vqMeLLybaeWftU/Pm5eflOqpo  
5wWYjnX4/PPyy+63n/7Fm7Hwi/+HdrP11noetBH8Tz4QYkfn5QGCKua1ywRoZ3j/KqbhpeA4yKEe  
ORZ5HmxL9j3TTeY8f/v2uj6xHVCHf/6pp+EeSNQNhqfg/m4DLwc4qHle//7tb0TbbKN9RZt95hm9  
vVq00PMsXarbi7YPTi3aFLDLDoP/16GH6oM819see6SWxS/qG8Hv1FOJZs7UeVHeW4vyOnRIlQUw  
jDaA/XLYML3OXbuaiX5iee8RV16pKxM7NGCl2qeP3ik7dUpVPH55h372WV3hH3ygp9nght8JE4jw  
Itv0DZweVGvW1CoB2P8H/PRT+Two1WxgPhwAoKhR9nPPEb300rrqwB6uCKw/Di4AgdBuhLxDonwE  
DZv0/4NggLoaNSo1raLfI4/UOxHgPGybTE+3IKjCxo3T2y+9TIBhBKprkrkmN8476/PO6jmbh8eYK  
4LL4bAH1svnmOg/jHFQZHmalWhElqmgPs2fr+ubl0svCwRUBIMf5F76/+CLRiBE6Lwx7fmwvLPPy  
y6m8338neuUVHQwrgoMq1hPB0S6TwfBbb6XaA8btOsbB8rvv9DS0/ae0sNhcLncHIDXjz+u83ga  
/w4ZUj6voieQ/vpLT8c+iV/et7INQOCgHeCF9fz6yLPO0r+eYtbaI3AURaVutZX+BYcdpoePO043  
DqgBngblhNPs00/XR0fk86kgzwNwtL/9dj1s54P0oDp1qp4HjYbn5V8ERh5GgERwYq66SgeQdDA/  
BwcMb7llqgyoRvhv/y+AJzfcYYDsV0G//7xR/k81CFUPsAw6gc76cMPVy2oQtnCZzuopvvLQRXw  
6TSA+oJKxflXX6/H+Sk++/mr19Y9wpP56AK8lvy8VtRUOD0ZllejxMqal9gP/+NzUNgbhOH2O  
zz1XK9X0oIrtaj914HfkSD1sw9PhC8osLEwpUkxDUEW3CobT94TycoCDKhg6NDWMbYX9BPsg9p2n  
ny4fVOfN0wdFnP3Z5fGwnWfD+dwe0KaRx/sd4F+IGB7G9kDQZzi/luzpGEbb5YMLujqwTTwmy9pt  
YnDDQgMeP14Ph3HGGXqHeeed7A1gfYGDCwKub6A+oMAQFLILYH1zzz2pizuZQP81/GOV5NI3vL54  
7TWttnxn7txUWw/rvw0DAgQv+Ub/r8t+gm6yyglh4zkONSFUijy/Eiol+cbGG1T5yDh2LNFyVvMKI

kqAcLE+cTmKC4KwUePn3o4gZPepXXGF7sdCP+APP+g+QUz7v//Tv+jn7NdPD9v9RPjFIV6c7vMV  
Q54G0JeHcZzCoO8JfbU4vUC56C9EnxSmw+xbnbifEuPsJ/o6cTENy2Mc/V34v+n9VolgbNL4ubfj  
olPYLT429jS+eLRlv9w6GObbRQD6fnARKgy7HAbDuAq51156HEF3xgyijz/WgZPnvfzy1G1XuJCC  
aXyhKr08QRDyAj/3dgSh/fdPBaMTT9S3tkB12jc6o8MbvwiqX32lh3HPHy/Hv3ZQ5TyGxxEccY/p  
ttvq+/wu0zf1wkGDSpf5kDzYlkMc/DH/XglqnzfKtQ01DXuF+RIBUHY5JG9XRAEIUYkqAqCIMRI  
fgVV3NCNG/2zgUc9+ekWBt0CuCkc92dan1QQBEGw8TOoog8ST2Cg7xLDsNtul7rjDqJHH9WGPDy2  
iqc1+GkSmN13imey8ejmjvqcZSJoloLWhgPA89B46o9btq24fl79Eg9RCAlgpBGBZEIx3AAw8Uj  
vLwBL37Ay0H4EU0Ygimu0ON5flx5x0sd7ltb/lsncXi4lqWKR/YwD8oCmYlqLpThgpkgCEIJJ4  
BgcwvPABz1LjZQp4eQXA24bwBh9WqwB3AeB+UFakgH8BhvFGJPxed11mpQrsoMrz4S1Kxx6rx/mV  
cllgCGlkiCyCIAiCKxJUBUEQYkSCqiAQoxIUBUEQYgRCaqCIAgxIkF1E6FMJUfYn0gbi4ZXQXWW  
SgmVzIRJiAbqC0kQ1he+trF2KsGvkSr5hFc1NWnFpKCSTltzmsnxh0QiQR/isxEhlOXo7f4rV64M  
6ito8GtMpccccQRQZ2tyPRF0RywbNmywC8fgV8P4+1mHrFq1apUG8vwYdtccOWaKwO/hq0YZnL8  
wlwW1aNHdyoKKBEA7P5PlSmGtg8fKAshyBYwo90e6LW0zQ/sZh6JfrQu4n36aOan9ADdR6iB+o9  
SB/X+4xWJdbSiMQoemnl2n+wwXzfaKkP5zUcC28ZDuHoL5OOeWUIG+W5Zr+/fvrNmb5hPEp9uef  
c4TtE9u9996bs4M1k/THamOwXII6ad++vfaIq/Grae79svHCE95Y6UE11xWV9MuyS6qX0jeFvWhy  
4XRq/7cO1K5dO7r55JupDB97hLvKrj3serqo7cVUVkg05dupQVlxcvtt6d84mTG5+f4I3Njv9Js  
+5rbmzlyQ5hPbLkzB+2JDMlrW+//XbKI7Q2NgQfVswHsb/SgmpxcXHOD0Qgty1K0bt371QlhQTV  
nFiG9EPiJ1qtAugaZSdefCKvtiulm/9+M61WNcl2w2FdqLS0ID6t/QXtX9As/H/EZZzCpm0oe0eZ  
SypUFlBO+jCX9K2ysDLWh7mmsDI2lPngQ5ilBVWYD3jhRbKSNgKIWqN6Dfq46HPqXdyHvq7zLV1x  
yRX0QMsHaczW42jljsNo8I5D6fvNe1PPut8F6nV0P+ub7THx73//O+UTJzOe6y6TpF9p1qCwgZkj  
N4T5xJZLwvxhS5ID8fXGG2+kfElrY7/++quZKzcku3EkqGYmqBQTVAs+OpXmA0VfRev0xf01cTj1  
eaMv/dyjL/06bDA92OphdUr0O/3S9xca2GMQ9XmnD/X7/heas3SOKSveclpz//33a384qeFv8bHB  
HDN16tRydYW6O/744/04LbP8YvOBbgOarNPG1uLtajkE2+vFF1/U/lhtrFu3bmaO3LFw4ULtlwmq  
BbsX0H777Wem5h5vgiqYrRlqycdbqrix+4ZuVv751bp166C+Fi9ebHL8AX41w7fFEPamBt8S9Uo  
IXzCq5ry+T7VqjT437b/jVbPDb8f5efEz2YoxY+JH81QdnRz9zeoLulv3HoCFBj8kqAaHV/bmNyn  
GoFZM1RQVY3qzBM3/qA64+kZ1CfRh35K/BQEYVUzV9HKKSupf6l/Teg8lfgFRN9adDmg4LxlZNW  
OqxXZ1H/gv5Bfhr83RElqLrj67b01a92l6qgqvwaOVyCaoXMmmWC6pkbf1Cd+cJMGlA8IBiGGI01  
axX1S/SjspW6b5GDJivVoXsPpbFtxgbDUQMqCPVrQyFB1R1ft6WvfuF2Rvg1cqQE1QrZ5lJqSfmg  
OvzY4fTnIX/S6DNHhwbVMW30nQKYtmJ0tCeRXP3aUEhQdcfXbemrXxJUI7ApBdUwJt8ymcpWINHi

fouTp/9VJQ6/1gcSVN3xdVv66pcE1Qhs6kEVLB2yIjb9scyMVZ24/lobCaru+LotffVLgmoE8iGo  
xo2vfkIQdcfXbemrXxJUlyBB1R1f/ZKg6o6v29JXvySoRkCCqju++iVB1R1ft6WvfkIQjYAEVXd8  
9UuCqju+bktf/ZKgGgEJqu746pcEVXd83Za++iVBNQISVN3x1S8Jqu74ui199UuCagQkqLrjq18S  
VN3xdVv66pcE1QhIUHXHV78kqLrj67b01S8JqhGQoOqOr35JUHxH123pq18SVCmGdQdUdX/2SoOqO  
r9vSV78kqEZAgqo7vvolQdUdX7elr35JUI2ABFV3fPVLgqo7vm5LX/2SoBoBCaru+OqXBFV3fN2W  
vvolQTUCEITd8dUvCaru+LotffVLgmoEJKi646tfEITd8XVb+uqXBNUISFB1x1e/JKi64+u29NUv  
CaoR2FS/pro+8dWv1iphWy5SySfKVIJfzVTyDfIF5Bu+tjH5mmoWoCAWqoRKOkl3+CGBT99wtcG  
f7xK2JarVfIN+HWgSr4BvxJIEISj0IEI1NIEIXzCm5oKNlyDoFIR4iNlnmxE+FFQUJBsWD74hcB+  
/PEqaFk+we68804zR+7o06eP9qenMqS6Cdppp528OBgFfhUYvwYrU+M+UKNGDSplqDbGSfm1dOIS  
MzU3YHude+65us4s69ixY8635bhx47Q/XZWpVLB7AW255ZZmau7xolUINxoH1Q/NuLJckvQrxHLJ  
jTfeGOoTbPz48Wau3JD0xQqqGG/YsKGZlzcK/UoLqrBckvQLxskDv1588cXyvlNwu3dvM1duSPpi  
gmqIQr4vKioyc+SWnAdVbKBkYUE1ZyYaworY32YaworY0NZWID1xkKCqjGKWzahjLXFFbGhrK0  
oArzAS+8SFZSSFB97LHHNqw9pMykRHvIw/Xh9pxKyTnDylkfZIKrz1qF+gS7de6tyfkee1hZWDnr  
yZ599tnUtkwLqvXq1QtdZkPY448/nvIrJKiGLbOhLOkXjJMZ79q1a+gy69VMOfrefOvUPaV+wTmM7  
Jed77AFIYeWsJ0OdJOsrLagWFxf70c1kfnPKwoULdb9lWID1geQGtOyPP/4wU3NHmF+5PsUGXbpO  
0f6kBdVcs2rVKt3GrKCKcR9IbKNOarhHjx5mau5I+mWZD3X23HPPaX/SgqoveOMJjDLVUIInaeS  
b3Cj8uFlaMN++caJkmFb+gRvO/jVXCXfgF+4+i9tLBpXq4Q6m6aST3hVU3Kfqju++iX3qboDv5B8  
w9c2JvepRmDWXBVU91FB9ToJqIHx1a/WV6qgqrbloqUePIGI/Gp2lodBdW+1LXeQNhaVdnepoKq2  
5cgJElQRb5TdcdXv+QxVXd83Za++iWPqUZAgqo7vvolQdUdX7elr35JUI2ABFV3fPVLgqo7vm5L  
X/2SoBoBCaru+OqXBFV3fN2WvvolQTUCEITd8dUvCaru+LotffVLgmoEJKi646tfEITd8XVb+uqX  
BNUISFB1x1e/JKi64+u29NUvCaoRkKDqjq9+SVB1x9dt6atfElQjIEHVHV/9kqDqjq/b0le/JKhG  
QIKqO776JUHVVHV+3pa9+SVCNgARVd3z1S4KqO75uS1/9kqAaAQmq7vjqlwRVd3zdlr76JUE1AhJU  
3fHVLwmq7vi6LX31S4JqBCSouuOrXxJU3fF1W/rqFwFvUaMkqFaIBFV3fPVLgqo7vm5L3/yaNGkc  
FSI3YCUF85UR3XDDt2Zq7vFqC0pQdcdXvySouuPrtvTNr+ee/ZOqFa2h9u3bU2lpKe2x+xO0287D  
zNTcl0E1ItLg3ZCg6o6v29I3v+684xfarO6vVL2YVHAqq5s6y2Gm6m5R4JqRKTBuyFB1R1ft6Vv  
fr3x+hjaZeenVFAtC6xYudb8wGfM1Nzj1RaUoOqOr35JUHxH123pm1+ffTqNtt36q2RQRZ/qkUd8  
bKbmHgmqEZEG74YEVXd83Za++ZVIfEjNm3emK65oH1jbtPDQ/Xrvmqm5x6+g6uvXVMtQUTr5Rtlv  
v75qLF9TrQQSVKPx2acT6ZKLrwxuqSotbUcdOlXOB+4/3kzNPRJUIwK/kHzDV78kqLojQTUaV3fu  
SdWLFgR9qbitqlRz1S7+u9mau7xpqaqV69Oia3Q3FXqlqDNN988OFLNS1atNCNyqR69eqZKbkF

p9VFRUVJv+ok6pgpuWfLLbekRE/jWa0Evfjii2ZK7kBbKi4uTm3L3xJBm/OhjZ1++unJwAVD/flic  
SUIJ0q+aNWua3NzS5+eRKpB+ogLr6KBptVrRK3TK39cNqsr1UCZNqnhaHM3Bi6CabFANICF9aMaV  
5ZKkXzBOarigoMDMkRs6d+4c6hds7NixZq7cgLoJfOGgWlf7Vbt2bTNHbuD6SRQEXIFisBIXikuS  
fqVZrtvY008/HeoX7LvvyjNz5QZuYyVFV1JJ4S1quFrSt+nTiaZMIRozhuj5580CivHjdcCcOJFo  
wgTUO9G0aUSTJ+vp6PpHXhyXAHleVLGBuELSG2pyB82Rlfv/nKzpXpivfqUFVW8sLajmrI2ZVKBS  
aiwtlVnDYWWsD3NNYWWsD7tZGZJdJyHp0rNlnZEQ3XijDpJgiy2I/vc/oi5ddB4HVVdDfd801Ls6r  
KjEVU3nWrFmjVkZVB0yCqrv56pcE1cxmkgTViOYQVFu21LFFLRZQWKgVKAfT9KD69tup4TilqZiq  
sdtuu6kVUIWSFITfe+89M0dumDRpkvYLxkKN+9AXF+YXLJfw7UqBpQXVXHPkkUdqv9KC6n333Wfm  
yA2rV69O1VmaedPG0iyXlGtjaQbatMHTfMEgFRXp31WriBo2JJo/XwdO9KnyNKjYbt308AUXEJ1x  
hh6uCrIv7RaX/OOSoME37KdqwCMuO/MyvSOq5BMPPPBA0q+/fvkr5zshYB8a/dUo8Oui9hcF475Q  
2q408KveeD8uODKXXKLavhUcfOG///1v0q8ffvjBqza25557Bn6dfPLJwXhF3H67DqZQRn27m8z1  
iFdbUG6pcsdXv+SWKnd8DKrAV7/kfaoRCILqShVUV3sYVJVfMN/w1a/Wq1VQVX55GVS VX81WSVCN  
igRVN/wKqvKYqjO++iWPqbrj67b01S8JqhGQoOqOr35JUHxH123pq18SVCMgQdUdX/2SoOqOr9vS  
V78kqEZAqgo7vvolQdUdX7elr35JUI2ABFV3fPVLgqo7vm5LX/2SoBoBCaru+OqXBFV3fN2Wvvol  
QTUCEITd8dUvCaru+LotffVLgmoEJKi646tfElTd8XVb+uqXBNUISFB1x1e/JKi64+u29NUvCaoR  
kKDqjq9+SVB1x9dt6atfElQjIEHVHV/9kqDqjq/b0le/JKhGQIKqO776JUHVVHV+3pa9+SVCNgarV  
d3z1S4KqO75uS1/9kqAaAQmq7vjqlwRVd3zdlr76JUE1AhJU3fHVLwmq7vi6LX31S4JqBCSouuOr  
XxJU3fF1W/rqlwTVCEhQdcdXvySouuPrtvTVLwmqEZCg6o6vfkIqdcfXbemrXxJUlyBB1R1f/ZKg  
6o6v29JXvySoRkCCqju++iVB1R1ft6WvfkIqJcCspSqo3q+C6ocSVKOSuE35perMN1q/o4Kq8mvR  
Ss+CKj78p/xq9rIE1aj42sbafaWCqvJr5FwJqhXi9SeqfW3wJvmGfKLaHWljbrRTCX6NVMknvKmp  
pUuXUqKm2XyvJmjatGlmSm7hLgIYQUEBrVixwkzJLTiN3XXXAlAJdTZI3c8EuSvWLMslhkfB/zQ  
K/gFq9esoeVq2opVq0zO+mPMmDGU+NpsS1Vv2LY+sGDBAr0tkfonaMaMGWZKbpbk7d265NuZLfYFW  
rVol21i789qZ3Nwze/ZsSjxttmXtBC1cuNBMyT1eBFVuUlkGppl+1OMnnHCCmSM3XHPNNSnf0iyX  
/Pbbb8HOF/jCSQ2/XWsrGII9c/q1eEt64uDD6K4me9FjB7Sgf6kd48pDD6VxtWoTFZcQlZTQBU33  
NqXFS7KOehq/6upA8fbbb5s5ckP16tW1XwXGr8HK1PgBBxxg5sgNd999d2pbplkuCQ6M7AsnMz5z  
5kwzV25I+tU18loSTXUbe/bZZ80cuSXnQXXw4MGpSkoLqrDmzZtvWNtLmUmJAcqHQeHWUqXknGHI  
rEfbZpttUnVm0v8latLsarVpjQoeHzVuSqWlpXR5586IjRTa2Lw6mwWd+7cdfSy9Xlw7tOyqWMuW  
LVN+WUEV4yUqklcts6Es6VdaUIWFzb9ebT9JIv+UT6EtC9YLtvYTjvtIKozTma8adOmoctsCCvX  
xqygineEVh/leVD9+eefU5UUEIRzYq4prlwnZSb9TQXVb0oa0AoVPLvvsIsqqFarlgyqc1VQLVVB



9c7D/kavl9QKLy8uSwuq3lhlUM2JuaawMjaU+eBDmKUFVZgPeOFFspLSgupuu+1m5sgNhx9+eMq3  
NMslH3zwQejp/xN16tHnxVvTH8Wb04tN96QPN9uSntpnf/rHscfSpSqQ/qlO/5cXV6OliSK67qTT  
TWnxkqyjtNP/q6++2syRGwoLC7VfaUEVqj+X/P3vf0/VWZrlkt69e6d84WTGc30LU9KvtNP/iy66  
yMyRW/wl7YoffviBEsWmkl5N0Jdffmmm5JYff/wxuRGx4aCsfWGrrbZKXkTo1KaTyU0xdvgIM7Rh  
+eijjyxpdmWqt4mT55spuQOXNjr2bOn3pZI/RL0xRdfBPm5ZuDAgeXaGNqcL+y3337JNnbKEaeY  
3NwTbMsnzLasmaARI3LT1sPwJqgCuaXKHdOszJg/yC1V7kgbc0NuqYqABFV3dHN38CuqMgtb15o1  
iZ57zoxkRoKqO963sdyL+nJIUI2APKbqTqX8wvywyy4jatKEaPx4PX7XXfqX5znnHKJDDyXacUci  
3PtaUkJ0/vl6ehbkMVV3KrUtNwC++iWPqUZAgqo7IfKL51eNMgiq++1HdOWVOo+n8e+8ealhKNUn  
n9TDWZCg6k6ltuUGwFe/JKhGQIKqO5Xyi+dHIN1Ifz1cWKjzeRr/2kEVD2PwcBYkqLpTqW25AfDV  
LwmqEzCg6k4sfr3yCIG9ekRXXIECTWbVkkDqTizbcj3gq18SVCmgQdUdX/2SoOqOr9vSV78kqEZA  
gqo7vvolQdUdX7elr35JUI2ABFV3fPVLgqo7vm5LX/2SoBoBCaru+OqXBFV3fN2WvvolQTUCEITd  
8dUvCaru+LotffVLgmoEJKi646tfElTd8XVb+uqXBNUISFB1x1e/JKi64+u29NUvCaoRkKDqjq9+  
SVB1x9dt6atfElQjIEHVHV/9kqDqjq/b0le/JKhGQIKqO776JUHVVHV+3pa9+SVCNgATV6CxeVJj+  
dthzVJR4hqoVPUYXX+THR88YCaru+NbGGF/9kqAaAQmq0dm6wX+p/mY/U6dOF1GbNqeowFpGPXp8  
bKbmHgmq7vjWxhhf/ZKgGgEJqtFJJbRMcd0pvPPPzewLet/RG0ufdVMzT0SVN3xrY0xvvolQTUC  
ElSjU1zQnzarPY+qF5cF1uroZ+jWW940U3OPBFV3fGtjjK9+SVCNgATV6NSq9TLVrTk/GVT/dnhX  
evPNT8zU3CNB1R3f2hjqq18SVCmgQTU6tVVQLSmcStWKxgXWaLv6b33PjNTc48EVX8a2OMr35J  
UI2ABNXoFCY+ovbtr6DS0tLATvp7Kb3Q9X9mau5pfbAJqgs9C6rK4NeuElQj46tf7S4xQXW4BNUK  
kaAanURiKG215Ud0+unnU+tjL6eSoh/oxi7/NVNzj3dfU8VnY+rVSwbVPYqK9NcO2rfX0z3AtzbG  
YDsi+YZ8TTUCzzzzTNC09t9/f5PjB2+++aZ3DT6RmETViz6n6sWkfsuoekl3uqbzS2aqRp3pbnBw  
eg32nrZ30OCffcOD+2cvvZRowYJg8LXXXgu24/bbbx+MU/fukb8Quz559dVXvWtj4Keffgq2I9K4  
38aZXD9oNbJV4Nfd795tcwvg51sQO+EZZ5yRbFC2YYPmkoULF4b61bx582TwyBXwo17d+6ha4SQq  
KXyMttn6aCosLDBTU/z4oxlIA1+b7trVjBiWLUO5ZqQSoE7q16+v66mnMqS6us5yRrE66ig6dOhA  
BQUFgS/D4Z+xd955J5hOjRrp3xzAvtjWtGITL9pYYJzMeC5BnTRs2FD70tX41TT3ftnk1JO1KoFE  
gaqUwhBTFYV5glSmbO0GMpOCDRfmlzKQnDOsjPVhJnW5qUuoT7Cx48cqV4mOO47o7bdTQbJxY1JB  
BXCnkdQl6aCKM2L+iCrAL+x//1PrFvb/s5guQ/kBSwuq2BHCllmvpvwpeOp/Uhs+tFhtLFhBY41Q  
Z+oXc5apCsP8oeWsDzMpQKu0bcgGknOglBee7YUXXtC+wTiZ8V69eoUusyEMJP1KC6rF5gCaa/SW  
ywHLVDJVEi29olwrc32baworY31Y1KS2KoscDNu/Bx+cCqpqvwngaaxU+/fvr35RThUsLajmyq55  
Uv2aNH5ntXlqj21Ay9S0b45TfwcrM8utd3NNYWVsKPPBhzBLC6owH/DCi3IVIWa5JMwfWEHCOs3e

kGdo5n9de+W1oX7Bpo+ern71fiCH8XvaaUS4PsNBIU//eZ6VK/Vw3756vDLwKXZ6UK0FiZwLzMrB  
h50xrGytsvnKaipDvplB/25ggroKsZy1MUNX1TiS/nAy43369DFz5YZkG5OgmhmuFNt8YLvttvPS  
r9GjRzv5xZPr1iW6+WY9vL4l/LGC6p133mmm5ICOHYI22yOYTK8vmJIA1KOHHS4BLVu2TAUK268c  
k7ymwMkTv0DglxVUL774YjMI9/hTSwq5pcodX/3y6pYqPIBgIdIudKC+mh1wQDBO55yjfz3A2zZm  
km/ILVURkKDqjq9+eXefKnjwQVSYNIztf/hhM8EPvG1jJvMGBNUISFB1x1e/Wh8jj6m64uu29NWv  
dqXymGpWJki646tf8uy/O75uS1/9kmf/lyBB1R1f/ZKg6o6v29JXvySoRkCCqju++iVB1R1ft6Wv  
fklQjYAEVXd89UuCqju+bktf/ZKgGgEJqu746pcEVXd83Za++iVBNQISVN3x1S8Jqu74ui199UuC  
agQkqLrjq18SVN3xdVv66pcE1QhIUHXHV78kqLrj67b01S8JqhGQoOqOr35JUHxH123pq18SVCmG  
QdUdX/2SoOqOr9vSV78kqEZAgqo7vvolQdUdX7elr35JUI2ABFV3fPVLgqo7vm5LX/2SoBoBCaru  
+OqXBFV3fN2WvvolQTUCEITd8dUvCaru+LotffVLgmoEJKi646tfEITd8XVb+uqXBNUIISFB1x1e/  
JKi64+u29NUvCaoRkKDqjq9+SVB1x9dt6atfEIQjIEHVHV/9kqDqjq/b0le/JKhGQIKqO776JUHv  
HV+3pa9+SVCNgARVd3z1S4KqO75uS1/9kqAagVkqqc1HZ6rkG9Lg3fDya6qKmpXgVzOVfEPamBsS  
VLMABbFaJTT4C1TyDW5Y8NMnfG3w/6cStqVP8LaDxy1U8g1pY25ce+21gV84w/UJb2oq2HAN0Nxv  
+tCfjcgNyrBFixebqbkjzC8fTmmfeOIJ7U9PZUh1tW8+EPHVYPwa7JlfaTZixAgzNXcUFBSs49fW  
W29tpuaOzz//fB2/YL7ghSfjikkLqrmuqKRfIZYTjID51w3/CvUJNnXcVD1Tjkj6khZUGzVqZObl  
DUm/0oIqLJck/QqxXPLK86+E+gTr26uvmSs3hPkEK04UmzlyS86Dau/evVMVExJUc2KuKayM9WGu  
KayMDWVpQdUbCwmqOTHXFFbG+jDXFFbG+rC7ITmkmSrlpwHVZCsuJCgOnfu3A1rc5SZlChSPmWR  
buj/Tc4ZVs76MJOUvuPqUJ9ggycOTs4XrEtYOevJVq5cmdqWaUF12223DV1mQ9j8+fNTfoUE1bBl  
1qvZbaym8iFkO8JwUS05Z1g568NMevz1x0N9gnXv2z053wZrY8uUqbRWpTCfYMVbF7NXwXy5woug  
ColGvpH0qeYaXMho0KDBOn6dfvrpZo7c8e6772p/rKBaVFTkxcWXwK+NpE915szcKS2A7bXrrruu  
49fBBx9s5sgdAwYMWMcV9P/6gjdBVRAEQRAEQRBskkK17fl1iaYfSk/dXYcWzB5AyxaPoVv+3Zmq  
FSVozNgJZi5BEARBEARB2DAkBvb/nD5+oZB+HvArDRs+gR59qJQWj9yFnnm5O33b63saO3k2TRvW  
nl5pfZZZCNk6W/07K2v0IQ1ZjwDU4c+S23PfYoW5O7SoSAlgiAlghO4veDXX3+IqVOn0pQpU8rZ  
hAkTAtuQnH322fTggw+ZsRQXX3lp3XjjjWYsO4me335FZ55YQguH70pLR+1GS0bvTktGNaalo3dV  
eTtRv08b0cpxu9LLz//DLGLzLR2T2lw6v9qPpva/i5pvX0qjf/+YdrjhvWDqvP4vUJuHPqZV9AND  
sUdL+njolvrliWOp3g6X0F9KNK5YOIPmzptK7/yrMz3YfSkTzvwe7Z/YhR795k8a+8KZVLjHsfT+  
gKk09fXzqf6BT9NymksP730SPTdiGa0e8gFd+oL1WMfPt1D163upgUI0z3kH0+FPjyBaNoH+ec/9  
aqkfQ3JcfSLKuE/e29HdTt2o2V/fkoHNR6Evlcbdmqfh+mcc++iP8dOodc77k+HI76pfB5O12+3  
JT3WayHNGfIYtax/Af06cQi1uulZWqNE7J8vnE27X9aVlqjhPk+dpObz660hgiAlgiDkDxCqw4cP

N2PIWbFiBU2cONGMbViGDh1KO+68Mx11dCuaNm2ayY1OYurMifT3k4+msy44n6bNnkc33XI3nXNp  
O9qlQQM6+Kij6eKzz6lRO3Uys6fzHZ1afQd6oe8smj/4fjq14dpOriXfU5HNT+Trv7nwzRTCbky  
Nd+luzan7qNW0+CnIFDdpS1NG9eDGh51M/01dwI9dekZ1OXD8UqodqMrD/OnTQ56M7+gyxtfSSOD  
XtDv6Lltz1HScQk9sld9avXIEFo4expNnlP+pSaTXz6GGh9+CfVU9TDrrf+jky68id7+Azec96XL  
qh2vhOoSune3Y+iZSZh7Dj1+6Mn0nxFraXqf++mMc+6hMVPm0tsd96FDroJQ/ZOu3qo+Pd1vJS0Z  
8RwdvsVthHORVT/eRgfvT91/m410a+PUrtzz6GLPpqPAgVBearBEHJCJqGKp503tFDF/7zmmup  
a9euwTj8O/W00+I///tfMB6VxNixE+jE1rvSkK+3lprZIMqmNFa/B9NLTx1H/3faQXTEUUFrjV26  
mNnT+ZrOr3s6DVaaTVNGM//4lp64/hw6/95hJk9RNoc+efxmuvuBh+jcxg2p9JNRZolgCllGCIKw  
KQFRmo0o8wB56l8QBEEQBEHwEhGqgiAlgiAlpelUBUEQRAEQRC8RISqIAiCIAiC4CUiVAVBEARB  
EAQvEaEaJ/36EX33HdGIESYjIj17En3zDdH8mF9z1a6d2sJqE48ercfffFOpt2ypx7/8Uo/DQMeO  
erhmTT3eqJEep/54PR6FJk30MieeaDlygHWGLVylI7ylFm1ysyQBZ7f8VUXkeCy4R+eTOTxxeVf  
iRYK6hfczj1vCvD6P/ywyYgRLvvVV01GFrh+0U4HDTKZhrp19bTSUpPhCf/8p/Yr0/ezf/wxVRdR  
GTag+jJ//KHj0y+/6PFatRy2O+jgNcU8v/69FOTaTjsMJ1/4IEmwxM+/DDIM7jsMj28xRZ63IXX  
XiOqU0cv/+STOI4jdn//vZ5+8sl62s476/FsfPVVyreH0l6K/sADqWkbgm4d9P+9/HK8X8hkZoF9  
ffRRkxHCnnvqeVy+bd+ihV7mkENMRgb4ODp3rh5nn3B8qSrYz779lqhHD5NhgXax6676f1WvTnTn  
nUTLI+u6O/10//aJRBVs0JW+BUKAwcS/fvfege+4YZUgEKDfPbZVBDbZReiF1/U03r1Irr2WqIL  
LyS66iqi//5X54M5c3Q5vEO1akX02Wd6GgLS1VcTXXmlLntN2me1zjuP6JxziP76y2RYrF5NdP75  
RFttZTIM+B+XXGJGFHjx7tZbp4lkXkOGeRo21OPwFz7dfbceB/DjmWel2rYleustXQc330z04IN6  
OgcWrO+wYXrdIRpsMTI7ts7n9T72WH3gO+ssvU5rzWfBXnmF6OKLtV9hQDTrVr080ZYoeP+MMojPP  
JFq6VAcL3gZjxxLdeqv2Gdvvt990vg18vf56HaSxXdg33sZnn010yim6brOBYGYvy3zyiV4XtB/4  
AWHCPPKI3jY4QOGEB+LhiitS6wC4Hf76q65zXp8hQ3Q+g/V/+WXdfIDGPfcQ/fmnmajAwfqCC/SJ  
Cv4fthXKQTsePFif4MAXBGAGJyt/+5teB4BtBZs+XR+0sU74f2jv6cyapbcH2hNE0tNPE91yC1G3  
bnr6Y49pgTkpeMExEb6egvqGhdG6ta7fwkL9a58U7bCDzrvuOpOhQHt6+22dhzq777Zdhwy2O9oZ  
5sH+h/YKXzEOuN5nziS67z5dp9hvcWCMytCh2i8uE/sU6hh1hwMg/j/iCw7IfFDm//vDD0Tt2+tt  
jm2E/w3DCR2EOsqFAezLWEeUP3WqzgPvv5868dx8c70N0EYwDt8AThaxD8HCwJdkMD/ENn4POshM  
UGAbIO/oo02GAfs1t/I//Yuod28zQQFRgzpAnUIIQaijbl94IbXuAELjP//R2wRt7OOPzYQITJ6s  
1/fxx/U4i20IC5uffil6xz903SGWIUYxiB98Eg9DHMC6YDtxHuIL6hjDX3xhFIRwOw57fyW2Oy9f  
VES0006pzgr4i3ycTACuD8QMtAM+BvH+CCDSsF9jHRDbsT6YD+szb56ZSYF2g+MQYh32dXs/Z/r3  
1/8fsTAKOEZhHcaM0eMcH7Dvl8Y89xxR5876eIK6s8E7NuEz4n3fvjoW8THnmGO0H3//O9HIkXpf

xf6JThcGwpTbJuzlI3W9oG393/9F7/glAyD3WK97702Vb8MnxtifGMQX5PGxAsfiGjWidXIloaTV  
ulAh1arpxnfCCfpgiwMGDjpPPWVmUDRurOfh3hwchBB0FyzQAg87KKbDbDgP5b7+uh4uLtbCFzsC  
hpGH/80sWqQtXcAC7FyYH4KHQaBEXoUfb1CkC9UDDtDj2NEACwSUGZ1wyRKi007Tedtuq+dhoYrA  
woLz8891HsyG83CQCgNnpVhH/J8ocHkQVBA7EJ/wC3nofUNdci8S6hYH5lGj9lEMeT//rMuBkOWy  
0sVmZYDIQ1kQ0bwuEGU4MEFsPfEE0amn6nnQ28DbtE+flB/go4/0GTva4rhxOg8HAqwbizpug3zi  
APigg3VGsC8pKT/P77+n/g/7d+ihehx+AfxjOOABHh+iF2GlznpJJOh2GMPnce9pTip2H57ndem  
jc5LB+2G23cY3A6PO06PY/vBPxz4UaeYxkIVvRkYx4kR1ytO7lBnWObr109IRCLPA/8RR4MHH64  
HsYBB+2odm09jm3BQs8FLg8nWu++q4VR+vZGPnzE/4JYAmgDPM+yZeWFql1fnAeRy2A/RI6zZiYj  
DYghiMKKeqBYDEDwgv331+0XcY7jAAAtVxCqMo0eJD9AQjTiJh9CF8MT+gHmaN08d1BELkAcDEF4Y  
Rj2g3tGGsTzEwwcf6HlCSBeqLBaxHqhn/A9u6+liFnkwxFFAbSRTLzm3Y46FNrZQBRDiGEYnBIQk  
hlmo3nSTHm/aVNcjQLnci4stFAhODMN/xE6AbYk8GOZnwYUYwPWJusQ62Cd3HDPCTuqjwP8TxwPE  
WHDEEToPxwaAOIFxiGkcT9GeeXtvtpmeh4UqlmWxjpNh5MFsOM/IBBLgiuNee6WODzgBQ3uEsMZ+  
inqbMSP8f2K/5Hwsi/jA8RXHH4CyMI71FCpFWq0LoaD3BWf93AOEneqaa1INIGGRglAD+GDGvSjP  
P7/uMoDzlCwXL4YvTQVbCCW0OOGwMag5wsWdpbGQhXBxgZ5OMusiGxCFQIQ41g/nKViB+Zgky5U  
ETiZii5pcp4tqG0QkCECot5KweVBvAGluW220aKExTB6FtCDhwMBRAmEW716ejn0SAD8cllhQhUB  
les/CqgLIWze74shWW3204fINAuWKiix4Hh7QgD6O3jQMjiCF8hQRlc3yzM0CvBgZ0vfyL4o8eD  
A6n5WkhwlH/D2AhlU2o2l9BQU8G8vhABHAZFhnoUQE4geCevYqEarb6ZaGKXwaCE3lsfNBfWlc  
PTb2/oR6xDrhUjgLVfTWMRBBXBbg+uBL3igLVxawXugFQ13joA7jes8El4feOYbFAf9P9BphWyGO  
cD2/805qnnShasN5dvvFAEIEBGYYEFO8DmGwUOX4ACD6+X/BWKjy/4Jg5V459PxjWeSjBwo9txiG  
CGGwr3NZ4KKL9DBfcll++9JLWhxzLyX7jBPPbKQLVQh5jON2ElwdAFgnxFuO9wz7hRM7gB5AjGcS  
qmjD8l3bnk26UAWIT9wpAmOhih5GjMNVjocQTxyT+QolhtG2bbgsiFa+pQC9fgziA04c0KPP8AkD  
x2b8L2w/FunZ4P9pX/FLjw8sQrGNse1wJQgn3MhLF6o42WdQp8iD2XBe2OV51D+2Q0UxJR10ruCk  
YL/9tEHlcvk4seJy0LuPjga+YoX/gZMeLMPgGI3l0HstVApVe0IkEMRxTxEO1Lgs88YbqXs/Gexo  
OLggyPEBAsEO86PhY2fBJSZM4x4SgEuK6HnFpWw+C8MBFL21COgYTuf++7V4DrukhamlbnhgnZ3B  
zpJjqOJMFZcOIZwAenfhU/fuehzrB/Glglh1QjDDZSiUC9EFUC9Yf/seNiWDX+GzDYIfrczHfxSE  
HEzDJcoolIjjMjf8tIFgRRkQqDgR4AOSDQlyDn4QbxCTqAscXMLuG4YQhqiBiHABPbYoH36gnrin  
AUAYod7Qi8aglw09lDAb1BWvD9pb+reT0c7Q04m2gyD63nu6rhkclNCbyrcM4OCleoMxEETwBz14  
AP8Ly3CAZr9YeAPc3pF+gGfRCX/hD+p+7711m8EtCGFgfwhbbwaXWPF/8GuD/wXxiDZjCzTsu9iH  
eN9FG+DbDAC3QXs/wwwM2qt97yB8x3ZDXUck2Ps/BBR6vbDPRbnUmF6/ANsR+wkMoBwcwCEkEAvQ

NrBtUXcwrC+2K+qJl2HC4gNOSNHOSp1xaZR7MRn8P+xDsDDS4wOD5Tju2e0XoJeY2zziCeqVwXph  
GdQFg/aUvu2xD6JclIFbYThGMnfcoXvS7f2pliAOsa3tW2oARBpOORC/4lQQdZUO+8W3BWCbo/cO  
yyAm4sQ9HV6GT55tsG2wjSAebRC/cbKMukZ75TyAE1SlmdxD/Nr3aEMkYR9FPdnwNmWxjDaFdYtf  
2CfCOgpYqPJJbQ724WTPPkNJBK+3fetAenxA+8U+hXaB9UR8xDrj//KxC7ELy9i95xXFB2wXbFds  
3/TboeAHOpdwElAZcKUpPUYyuMWGtwN2A/hng5NR/F+uf8EZ1SKETZa77tI7PYtAnP1h3O7BcAEH  
NgS2LbfU5cDQYwVBF6UXaVMBB2YEo3xa58oCkYF75biXCOIe965F7ZnZWEBbwAGLbx2ICwgGXlRk  
/Q09ahA2uJ1I0OB2kjBhKVQOfn4A95Fze0ZvrH2faxzgBBK941tvnWrfODZBplfdKIEVELNxEurch  
YzZOCHDvNnq9hSqhWoawQcGBG4HV7tXjndTuBdpQYOflyyu4NLOx3vDNdcg9ElIlgCllgbPSol7sQ  
GVxywmVC9GjgUhzf0wj4TA2Xn3CTPQQLhPalxjt3hG8HoUv++EeHTx5bV+6wqUGXHJHDyYuSePS  
h335Fj7g8h5EL54yxf/DvLj8Y8OXEsMuPeEyGO5Xgz+4dl//wZcncEkJlzdRLn7tB0ZwCwQuX+By  
Fy6r4xISXrmy2K4Zwf+4VIlwBk51gM+si+49IVLJny5Cpcj7csjqF/8H8yDextxSYcvr0Howy9s  
B5zpp99DKQICIAjCRo8I1ajg5nPcE8P3LOK+KO7FA7j/Bpc38RQI31gNcYfpeJKRL6Hwzfy77abH  
AZeDh45wyYMfyrLvXeKHKPDQEt+cDYP4Y/gJUDzQxEC44mGwiu6PwUNbWlYfsOD3DOIjbr73EQ/g  
4Mzy5EN84IU5GIY4xOUhvFIeghx5uNSBdcA78/heRNx7xO+R5Hvp4Dde6cJ1ZT8kgss0AG9PwDjq  
FW9MwP+ByOWHX3BvE8D9ZfxmBBGqgiAlgrDJoI7sQiTwGhkIIX5YAL2lGlcB9O7h/jvcBM7vL4R4  
g3DFaylYqLLI46dOAZfDQpVf38NCDODmd+ThCXXcn8bL2Lcl8Otn8LQ+A1ELq+ieH7ySB8vwU4r8  
hDquesuT74NBry089QjDyOkCI2qBnlZfF094Q9ZgHPkNg41U1DF7DhHnx0Bewn+RMF6r4iAADn/gp  
fV53+5VLIIQFQRAEYZNBHdmFSEDo4YIEPPmIp15xKRtPJOKI9/ZlcVxOx4uYIU7xkvPx480EA26w  
xtOB6LnEZW0AcYjXctivYULPIYQfXgUFwYZ57Rd4n3uuXsZ+/QeeelbwRC8ngwdX8L5C+xYEG6wL  
bj3A64wA38KAp/bxih/4iWn8jIGA2x+wDHPjbbAshCzeLIC6Afi/eA0NXpid/sADv+gZPceYhqeA  
cQKAJ4UBLvnj/4S98B9PaOOFTDUAZ4AR69s+psYBEEQBEHYaBGhKgiCIAiCIHiJCnVNGX65dliP  
5PoEtzmGv7dBA/2ycvSo4mEoQRAEQRAEB0SoRgVfvsFL1PFCarxMmC9P46XFeJgHI/xxj6I9iR0P  
/+AJeX4SHk+n4wl9voyO2wDw7Wssyw9p4R5Mli358MJ7PAiF/8HgKXv4gkvfuPcU8/BnL/GLlzej  
PLz0PI2oQkDiwwOwqGAd+EEINjztXxEQqJjnqKNSD3DhRe/I409eColgCllgRECpByESLKN22oro  
0Uf1PZjHHqvz8EJf3EuKe0HxOT1+8h+iFTmXh8B8eA0U8vBFE9zriWHcZwnwKijcE8rft+fP+8H4  
vau4nxPj+Ewbvq+NMvB/+OEK/jQeBCu+XYw8Fqp4YT9enYUHV9LBa58gMvGGAhi+vMQCm8H/RHmZ  
hCo+RYh57E+oQowjDyYlgiAlghARUQ5RYaHFnwbFw078TIT7c23oDeXeTsCvXcKDT5tvr/1BFGJ  
S+J4hdMhh+j5AH8fm79hjvem8v9NF6p48p3Bg1542wDyIW4BRCZ/QSrKpX88CIWeW/TKwtAjm/4V  
jyhCFa+vwjwQ5VwH/Fk81IEgCllgCEJElHoQlrHvvlpY4nvXNhCp6IFETyJ6VHEp3gYiEq9zatYs

1duJ2wPwXlJ8Ccp+iT/Au1oPO0y/3xS3AeDJeIhTFsjoDYUvYZ9BRU8sLq+jXPiB3la8SB+3JACI  
T7wWy341lgsHHqhFU4VXZTF44T5ebcXimsUtnthHTzLWBT21LFoFQRAEQRAiIkJVEARBEARB8BIR  
qolgCllgCIKXiFAVBEEQBEEQvESEqAlgiAlguAlIIQFQRAEQRAELxGhmoGZKiWsdJJKghAnS1Sy  
29hhKgmClGxs2HFsO5WEzFyukl1nI1QSwhGhmoFZKtK6UyVhlopS3/vqpCVdKHaSiWhYqSNVR2p  
Q2F9YMexXVQSMtNOJbvORqokhCNC1QIBfNGiRXTwwQdTYWEhJTZLUIFKyab0boL22nkv+hteZm/m  
F4guvPBCOvTQQymRUHUUYkcddRT17tU7mHfQ6EH0ysevUNf3utKA4b/QUpVWqvT8C6/Qe5d2o/m0  
Ql2tCmzm0pn0x4Tf6dcxv1LPQd/TW1++RR98143+mPRXUNbGCNrM2rVrqWXLlStWjVKFKW1sW8S  
1GS7JnSI+RCeTDFNaWlpsN+FtS/Y0UcfTZ/Z7/fNY7jNtGjRgmrVqhXUT0GBamNWfe288850ED5Y  
opA2pvnHP/4RxCq7nmxr1aoVvfnmm2bu/lbbTPPmzalevXpB/ZSLY2MS1LBWw2A6kDamueuuu4J2  
FLSpZ5O1pVOjBB1zzDH0zDPPmLkFRoSqoW3btuWCUmANINnpQ2Vp88xO/8xoHpFeFxVZjTo16F8N  
/03vFX1MHxV+TsMLR9PMgrk0seYUuuCM8+mqdlfRGeedTo83eYpmF82jHzf/iT7c/mN6e6936JlG  
z9Jbe/2Pvt3me5pQMpn+KhIjyOtW0ZpCojUFRMNqDadhvf80HvINly5d1q2f2srs9L2ytHnGjx9v  
Ssg/0usiim277bZm6fzjgQceCK2TbDZ06FBTQv4RVh/ZrA4+k52nvPDCC6F1Ui6NU5Y2vTc+pZ2n  
pNdFYF2V2ampMms6OssEjQhVxbffluugSQti1BN76XYpK27MjuVpZ1BZ0jvJ7rR7IIFpGantcog  
MFCwRkGTLjyJ2rVvR6XtSoMes5v/fjNNK5xJq1WrTDdVDN1wWJfkbD1vpC+q9FbITuPdks0Dvd5  
Y7AsQjWv2thryuzk0MbWSWrZdVLY/9zYLYZUYR1/pSzsF27M9qgy00kbq5SFxiU7pQnVvIpijUS2L  
UIUdia9MCqp2BJo/fz4dccQR6zQSEaqWVUGoXrldZ+qTGEhfF/WkHwv70cTCaUHLu+yYy6i0oxKe  
GYTqGvOL+a854jppq16mU2I5zOV1wwow0omA1TSqYSjsldgr3eWMwEaopE6HqbjEkEarSxIxNhGoM  
lkWoFhUV0RB8ol1QtSMkqVmzZrmGkk2oPvHEE2bj/GTgwIHL6iOTbVXcgAYlBpOK7bSmslzWFupe  
1WXFk2lVvYqargTqG9u+RZ9v+yV9ss3n9PF2n9Ib279JT2/7LHVv+AXNqjY36JHFsmpt0MrEGvo2  
8T09fMxjNHPODOOR/2y99dbl6yaLUL399tvNkvnJqFGjytVHFMt3dtttt9B6qcg6d+5slsxPpk6d  
Glovmszfwb2n69SLndKEKp5jyGcWLFhQrj4CyyJUV61aZZYWZI+rgDPOOIP2P27/VCNSqcYXNWhw  
38HBdLk5XMP1sGbNGtp9992TD2+wHXDAACf0YV0uvfRSOvBvB1otLEHV+ISj3l/oe7mkjWnsemja  
tCltttlm5drYXnvtZaYK6UCE4qEqu75KSkqSD55JG9PY9dCkSROqX79+uTqD+BfCueWWW+ig/Q+y  
oliCiicV0+tPvh5MIzamKdfGdmpCDT5sYNNVYgnZsvaOZKqQjQjUD8noqd9IPIhCwG4qxV4wNbM3i  
6P8zucxCBz9jjLvyeip3WrduXa6N4U0dQsXgAGnXV7NmzcwUoSLOPffccnU2ffp0M0WoCFVTySSv  
p8qOvJ4qOiJUMyBC1Z1cCtUIA5TsU1a2OrqS/Dnxc2Crpme+zLLoh0XUJ9GH+ib6mpx4CBWq0gGR  
ERGqbhQdUeEqjuqpplpEKoSxzliQjU6IlQzMGUWEqpWsDrzTBGq2dgQQnXmCzOpX6lf9U/Op4F1  
BtKgLQbR8hHLAYeZiM5Zq2jNojX0V6u/AmE5+tzRwXLj248PlkHe3PfnBnmYH2WNOGkEIS0ro2V/

LKNft/41mG9C5wnBPGBR7/UkVJcooWrVF96xJ2RGhKobIItdEaHqjl1fu+wiParZaNdOCVWrzkaO  
FKFaESJUMyBC1Z0NKVQHIAwwOZpk76gSqismrAjEJsTI2iVrzRwUKISDZawe1aF7Dw3mG9NmjMkh  
WvDFgma5Eaq5R4SqGyJU3RGh6o5dXyJUsyNCNToiVDMgQtWdDSFUZzw1g35K/LSOaOyV6BXYqhla  
dK6es5pGXzhai9ppqA2j+Z/PXEaq8zMqpK4Nx8NsOvwXiddQ5o0yOLmvUaaOCsgZuNjDoeY0DEaru  
iFB1Q4SqOyJU3bHrS4RqdkSoRkeEagZEqLqTy3tUbcRWINGkmyYFva4QpuhZhjc8pcYvNPq80VS2  
0o8bqESouiNC1Q0Rqu6IUHXHri8RqtKRoRodEaoZEKHqji9CdWNBhKo7IITdEKHqjghVd+z6EqGa  
HRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHq  
jghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVD  
hKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGh  
mgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6  
EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITd  
EKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6I  
UHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq  
0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghV  
d+z6EqGaHRGq0RGhmgERqu6IUHVDhKo7IITdEKHqjghVd+z6EqGaHRGq0RGhmoFZC5RQba0a0ena  
znxQhGo2RKi6IULVnda3K6Fq9snEsUqoLhOhmolAqKp64jprdr0I1Wyc+4gSqqeZOIPHgOnzRKhm  
w45jIISz0+55JVS5jR2nhOpUEaoVIUI1A7NUUk0omc5USciMCFU3RKi601ol7I+cFqkkVEyZSnZ9  
NVNJyMy5Ktl1NI0IITN2HBOhmp12KnH7QhqpkhCOCNUK2H333am4YXGyEQXprQR92e1LM4dgM3v2  
7CBAFRQUIAtYhYWFZg4hnZYtW1JJSUm5+oK9/fbbZg7BZvny5UH9FHyn2pidakoYq4jjjuOSqqp  
NmangQl6+t6nzRxCOMhjhR8U2jVGic2ljVUEbpMli2N33HGHmUNIJ2hJL6a1sYbSxipCasbA96OW  
E1oNTAPi9KEynqasQ4cOed1j+PLLL5erj0zG9Tp69GizdP4xb948Ki4uXkfMZ7ILL7yQVq1aZUrI  
P957771166WnMjvVvabyuV7/+usvs3T+sWDBAqpXr175NIYQ1FIqDVbG05SddNJtHLISINC/vH5  
558H9VCuzt5TZqdtlVnzDB482CydfyxevJi23377VF1FsL/97W/BiWa+8v333wf1UK6NdVVmp6bK  
rHn69+9vlhZEqCrGjh1LO++8c6oBsWURqrBLLrmErrrqk3bLIX2hzIrdVAp8bSqgxfdbPP3N6dO  
KtlIBamtsrD/vTGblXCZZ7/++1HihfB6qdBUHZ+/6HyrJJMGKWujLOz/biLWsWPHdfa3wCoQqmy1  
atWizp07h5a5KVtpaWnQS2/XRWBZhCrsrLPOCi1zUze0k/S6CKwCocqGHsS8aGOI/VYqVemi4UdQ  
4nIVD2Hxqij7NkGnTD/FKsmkCcouURb2vzcRq7CNVSBu2SBYX3rpJaNS8hsRqgo+21nHlgjVvLGv  
la3PPFY/N3Zbn+I9ZWH/Mx8si1AVS7MIQlUuszBII1byy9ZkGKQv7n/IgWYQqDG84EUSoJrn99tvX  
aSTZhGq1atVo/vz5poT8o379+uXql4qdfPLJZun84z//+U9onWQy3CowZcoUU0L+sfXWW69bL1mE

6IFHHWWWzj9Cb8fJlISLiory+tU4jRo1KlcfgWURqnhoNF/p1q1bubqIYnhWYdCgQaaE/GPXXXdd  
t16yCNW9997bLC2IULXAa1xwP+A999wTBO9E9QQVqBQ0lqQ3EnTOSefQnXfemZxfIHrmmWfoxhtv  
LLeTwfhem4ceekjeEWfgNnPXXXdRnTp1ytUTG+4ZxIkTkDamgQC79dZbdR19pcykYP9UeQ8//HBe  
3zdow20GD7M0aNAgqJ9ycWxAglod2CqoTyBtTIOHGLFfBm3srWRt6bor1HGst58+Zu78htsM4tRO  
O+2k21haHDvoolOkjaXx0Ucf0X333afr6EndvpCCNIYvQQ8++CD17NnTzC0wllQzIK+nckdeT+WG  
vJ7KHxk9IRvyeip35PVU7thxTF5PIR15PVV0RKhmQISqOyJU3RCh6o4IVTdEqLojQtUdO46JUM2O  
CNXoiFDNgHxC1R0Rqm6IUHVHPqHqBi672vUln1DNjnxC1R27vkSoZkc+oRodEaoZEKHqjghVN0So  
uiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZ  
EKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8R  
qtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0R  
qu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghV  
N0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRod  
EaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXH  
ri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC  
1Q0Rqu6IUHXHri8RqtkRoRodEaoZEKHqjghVN0SouiNC1Q0Rqu6IUHXHri8RquH88svP9M+b3qez  
z3qHdt7hMSouupVKcv+t6uxKOvOM5+iZZ96nxYvnmrkFRoRqBkSouiNC1Q0Rqu6IUHVDhKo7IITd  
setLhOq6fP/9SKpdYyTVKCbavuGbdOGFZ1P79ldQaelldMwxF1OdmlOoupp27NHDaP78WWYpAYhQ  
zYAlVXdEqLohQtUdEapuiFB1R4SqO3Z9iVAtz9q1a+nf//qRCpTiKikcT23anEJtLr2ELr304sAu  
u+wiatmyDdWs1o+22OwP+v33KWZJAYhQzYAlVXdEqLohQtUdEapuiFB1R4SqO3Z9iVAtD/bBjx4f  
Sc32e5T22fstqlZIVL24rJwVKzV2wYWXUdPGH9PoOdPMkgIqoZoBEaruiFB1Q4SqOyJU3RCh6o4I  
VXfs+hKhui4ffzSZDjzwetq96duhQrVlqbFzzrmCDj/sfzRt2kyzIABEqGZAhKo7IITdEKHqjghV  
N0SouiNC1R27vkSorkv37INon306UPWiGVS9ZJ4Sp3OpWtLmUHFiGe215yvU6uhuNGPGDLOUAESo  
ZmDWjDShepw1WyoaCJC1YUli9OEanMRqtlo3TJNqC4UoZqJdYTqLiJUyFC1R27vkSorssLXQep  
uvmV2rS5hNq3b0+lpaXlrGPHq2ifve+jPRqPohEjRpulBCBCNQOzVFK7XTKdqZKQmRYq2XW2RiWh  
YpaoZNdXK5WEzLRWya6zRSofJFVomkl1fzVQSMiNC1R27vkSolgcni089MUzVTU+qWW0m7b3P7XTa  
qWfRxRefTOeffxldc8yltHWDz6h68be0yw7DaMzoyWZJAYhQrQC8flf3X72TwR2p1fxWtHbxWjOH  
kM6oQaNo72V7WzWWoMHDBpupQjpjxoyhAX8MsGorQS0WtaDV81ebOYR0RgwaQYcsPsSqsQT1/b2v  
mSqkM378eBo6bKhVWwlquqwprZq9yswhpPPXX3/RCSecUE54ffvt2aqkM6kSZNo+PDh5eqrUaNG  
tGLFCjOHUFa2lu68o6+qm8VUvaiXEqQ/KeP7VImqFf5F1UumqeH/0jZb/El//SIC1UaEqsV9990X



7GQFBQV6h2ugzE4fKjM7Yq1atWjUqFFmyfwDZ4grV66kK2aJOskslHK7FSkzEzD5Q1eNI95/PHH  
g7pltrHaQS2l0vfKzPTi4mlaMmSIWTL/4Hay3377JdtQYD2V2amuMjPtksu CZbJ5zbWtWvXoC6S  
bawgqKVUGqzMmt63b/4KfW4nBx98sK6rChb22WcHy+RzG3vzzTeDuki2sQqMp/fs2dMsmX9wO2nd  
upWqi21ohx3aUrWiN6mkUlnVliVWixap4e60Zf07qX59fVvTySefHCxTURtbsIDogQe0RWXCBPR6  
a8tG+/Z6vnr1TEaOEaFq6NChQ9BAylkGocoGsZavpNdFYBmEKmyvvfYyS+cft956a7m6CKwCoWrb  
7NmzTQn5R3pdBJZBqMK23357s3T+wSdC5awCoWrb2LFJTQn5R3pdRLHNEpuZpS02Vd2atl5vPqNF  
qqN+n6QKSH/CKuPbFacKA6WPf0M3GNOtNVWRAccgLKIfv5Z/8IYde4QCEvOP+qo1DDuXLGF6nHH  
pYaLIhFhGuePH63Juuy01DdakCR4s19NyhXJD6N27t9ogIY0IglAVS7MsQlUszSIIVbE0yyJUxdIs  
glDNG3tG2fpM2J/D/u/GbOszDVIW9j83ZrtbWUzpxDqFLjILC8YDDzSCRTFuXEplgife0MMQnTY8  
T7pQteG8t982GYrrrtN5vpzpz7mcv1xwwQVqw6imYVsEobpqVf7e64X1X+fSTwahinl32GEHs3T+  
0aVLI/J1BYsgVOFmMWNKyD+w/uu0sQxCFfNuttIb1ee8NBDD5WvK1gEoYr7pfMVrH9BlvMlBnt4  
3nXlKx7VV/7zyjp1EsV+6fGLKSH/wPpXto2deZbuUW3ZMhgNSBeqOETsu68eb9GC6O67iRo2TM3j  
KlRvvFHn1amjtvcrRMuWmQk5Is1l4f3331cbyDSYNKF8FGqoeWzeLDBDfO4Xxd1EuxcFQjVG9Hy  
hYAvv/wy2Y7ShWrBD6k2Ju/S0+BkcOuttw7qJGhjaUK1oK6us6uuusosIZS7SpQuVlcoM9MmT5aH  
NsCaNWuocePGQZ2sc2Jk5Z1zzjlmCWHQILxuqXw9hVhK+31Zig8+oHnjggUGdZGpjx+G6vCM//qiF  
KoxfUMG9ojVqEG3sd4+JUM2AvJ7KHxk9IRvyeip35PVUbsjrQdyR11O5Y9eXvJ4qO+3atStXZ3jT  
kBCOCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMzJqZ9mWqk0WoZqPFvJlKhfw  
+YTq4SJUs9H6yLQvU8knVDOCV9zY9dVsTxGq2RCh6o5dXyJUsyNCNToiVDMg3/p3R77174Z8698d  
+da/G+sIVfnWf1ZEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2O  
CNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj  
15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh  
6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1  
R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eyl  
UI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdE  
qLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU  
3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUM  
iFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl  
1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYI  
VXdEqLpj15cl1eylUI2OCNUMiFB1R4SqGyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMiFB1R4Sq

GyJU3RGh6oYIVXdEqLpj15cl1eylUI2OCNUMzFJJNaFkOIMiITMiVN1YohK3L6RWKgmZaa2SXWeL  
VBIqpkwlu76aqSRkRoSqO9y+kHZRSchMO5XsOhupkhCOCNUQ0AMBVqiUKes1pPNUeJlZ8MEHlwww  
gOtTSGHXid3GjINJyMyJKiVrTNUdkDa2LhW1seYqCZm56KKLysWxpUuXShsLoal2tptKQmauVilZ  
Y6rupqskbSwcEaoGbiBbbrllKkA1MI2IO4fKzLTLL788mD/feHFF1P1lcGqV68ezJ/POyKv+7bb  
bpuqm9rK7PS9MjPtsJYH04cnnEqjdmlC07ffkcbstCt9vtf+9Om+B9Bne+5L/avXpfd22oW+3mc/  
+nLvfeGLNe2rpvvSgN32pDG7NKWxysar6X/sujt126sZPXrGufTciy/TshUrAj82Ft5///1UfcF6  
KrNTXWUqv6SkhJYvXy5tTLHjjum6qsgqKVUGqzMTDvuODkxAt98802qvjjYUVFRcEuYtDGilK2a  
IK8fO41TZvIPO+ywYP58Z+DAgVRQUJCqr67K7NRUmcovLCyksWPHmqUEIELVcOWVV6YaEFsGoco2  
d+5cU0L+kV4XUWYp3fcwS+cfN91007p1EiJUayQKqEv1HahfyZY0pVod+qzattQdttm2dPWWFFwX3  
NsHaXnMNfbDFFqTOAkJtQZ3N6KSzzqVTL7yY2hxA1RZ3OiGtXpnoIaxiP/Wae+YBUIVbaGDRua  
pfOPBx98sFxdBJZBqLINGzbMIJB/pNdFFNus7mZm6fzjpZdeCq2TcskSqmW//vijKSH/SK+LwCoQ  
qmzFxcVmaUGEqqJ3797IGkjSlgjVvLGvla3PFPY/88FChGqbRE0INKtRmbLVSnAGVq0avb3nXnTp  
VVdRaWlpYJd37kzd6tcnUtOS81k2TwnVs88/j0qVqMX8/zr5VJpVvQa9U61uuC8bi2URqmJpFkGo  
5o09o2x9JuzPYf83H8xOIUJVLm2yCFXYKaecYIRKfiNC1YADeXojiSJUu3fvHpwPbtLWS9k8ZVb6  
RaVEM1UHR7jZjhftSL+pZlcVpN7Kwv73JmJ9+/aLCy64YJ32EyZUS1T+uds1pfcLt6M3SrahCdU2  
pz7VtqLeNRpQp7PPoXYdOtBVHTvSuddeQ1/X3ZzUqTdRUZGyQm0YLykZbVq0+Utg6SbDj6Cujbe  
k2bVqhP0qN5SVJP6Dxkc6qdP9uuvv65bX7AsQnWrrbaiwYP9X7+4rU+fPus8SRxYBKH66quvhpa5  
SRlizFRIVhqsUulGVQchsSqT1TqhFg1RyS4rSH2Uhf3vTcR++uknuvHGG9dpP4HZKUSoPv3006FI  
buo2ZMiQdeoiMOIRjYwl1TTshpJJqB555Jfmifzm1ltvLV9nGUxel6QpVy8Z7lFt0qQpdXv5FXr0  
b63o4V2a0F37NqeHr72enr7zLnqs83X0ar1t6bbL2tIL99xLXe+6m56982566Lp/0D0XtaEXTzqd  
Hjv0SHq4WXN6sNXxdHena+iDDz+kxUuWGC82Hh5//PHydZZBqE6dOtUsld+Uq68MQnXnnXc2S+Q3  
r7/+evk6y2DDhw83S+U369SNnSyh2qBBA7NEfoNOrXL1IUgo9u/f3ywlABGqlfDN4qtVsp9kvEAl  
ITOHhHJluZOR5PODBxVh14ndxo5XScjM/6mUrDFVd0Da2LpU1MZaqCRk5uKLLy4Xx1auXClTlISK  
2lhjYTMXKtSssZU3eF1mNLGwhGhmgF5j6o78h5VN3L6HtVzz1URQlWAOnVMRgSuvFlvs912JiMD  
F6gTu9NOI/riC5MRD/leVTfkParuyHtU3eH2hSTvUc2OvEc1OiJUM5D8MIV1bWeeK0I1GyJU3Viy  
1HyZyrSxVsevJ6FaWKgFZvfuRPjC2sEHa9GJ+6D23NPMpFi9mqhjR6LNNtMis2dPor32ltp3Xz0d  
01DOrrsS/fil0d57E+GBLojSOXP0PHgTxuGH6/nYunTR02Kg9f8poWrqC3W3aLEl1Uygl8ZuY80O  
EqGaDRGq7thtbJfdRahmo10nJVRLTJ2puhs5SoRqRagjiFAR8glVd0SourHBPqGqyg7s//6PaPfi  
ndeunc5r0kSPn3yyHj/jDFibTuf95z+pZQELVfsTifPnl5+HqVIT5z35pMmIB/mEqhtJoWpMPqGa

HRGq7tj1JZ9QzY58QjU6aUcWwUaEqjsiVN3Y4EL1009NhiJdqKLXE+N77EE0eTLRypVExx6bWhbY  
ParMvHnl52GqVdN5jz1mMuJBhKobIItdEaHqjl1fIISzI0I1OmIHFsFGhKo7IITd2GBCdeZMbfZX  
qSDwkMcfrZgxg2j8eKKJE/VlfpS8vvlmeRHKy/BlfoAHALAszAbbHmVj/qVLTWbVEaHqghghVd0So  
umPXlwjv7IhQjY4I1QyIUHVHhKobG0yoRmXECH1vauPGRlwTnTVVaQiqJnoByJU3RCh6o4IVXfs  
+hKhmh0RqtERoZoBEaruiFB1wzuhuhEgQtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdEqLoj  
QtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdEqLojQtUNEaruiFB1x64vEarZEaEaHRGqGRCh  
6o4IVTdEqLojQtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdEqLojQtUNEaruiFB1x64vEarZ  
EaEaHRGqGRCh6o4IVTdEqLojQtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdEqLojQtUNEaru  
iFB1x64vEarZEaEaHRGqGRCh6o4IVTdEqLojQtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdE  
qLojQtUNEaruiFB1x64vEarZEaEaHRGqGRCh6o4IVTdWrVpWrr6OOeYoM0WoCBGqbbohQdUeEqjt2  
fYlQzY4I1eiIUM2ACFV3RKhm54brX1V1Q1StkKhQ/RyXTKRqxX/pvAKilvVbq3o3NecqvYBQDhGq  
bohQdUeEqjt2fYlQzY4I1eiIUM2ACFV3RKhmZtq0capeRIGx2vOOOvIGat++HZWWlibtnLOvpzq1  
xgaC9Zab3zVLCTYiVN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQzcywYQNUvcykLTYfRNWL1ICx  
6UG1rcEWP6ppy6lj+/fMUoKNCFU3RKi6I0LVHbu+RKhmR4RqdESoZkCEjsiVDOzdu0SKky8Rocc  
Opbq1pxDNUqlqheXJa2kkKj5gfdTo+2eoyeefM0sJdilUHVDhKo7IITdsetLhGp2RKhGR4RqBkSo  
uiNCNTt167xKB7d8gGpWW7KOUK1WRHRgs3eo4bbd6e23PzNLCDYiVN0QoeqOCFV37PoSoZodEarR  
EaGaARGq7ohQrZhFi2bSLju9SLWqD6XGjW+htm3bUseOV1CHDm2T1rHjpbT/fg/T1lu9R1ts/hp1  
/6yXWVpgRKi6IULVHRGq7tj1JU1OyJUoyNCNQMivN0RoZqdwsTbtM/ej1L79pdT+w7tgoAFw4NV  
HTq2peOPP4+qFS6nt9/6wCwhJCITQIU1l1bqSaZFkgGJ0iSPPkr0xhtQEKrRFepf2Pbb09r//pfo  
scfMjEI6IITd4f0RaReVsK8KFaoivlvjsiqjJlQjQJUDIITdEaGamR49uql6WUbi/6kasW91C8F  
96WWFOr7U6sVzqbqxe9Q9cK1dMbpSkwl6yBCNQqI1MWLiW68kahmTal//jATQnpUDzjATFG8/z6p  
wEZ06KF63Ba6eYwIVXewL3IKhKqQERGq0RGhmsaQIUPKBaiKTEjRsGHD0Dqy7dlnnzVzC82bPUZF  
idFKlC6gakUfB2K1ejFE6glqKeqq8v5QdfYnDR3axyxRMapqA3vxRZMRAV7muedMRgZWwNCWa/bY  
Yw/lS2IPPZXZqa4ylX/PPfeYufOMTp30BrUut86cOTNVXxmsHBiH9ehhMvKLQw45pFzdhNn1119v  
5hbSv6pXLo1TZvKFFCeeGKqvroqs1NTZSq/TZs2Zm6BkVZkSD+DzmYFBQXBL3pd8xW7Hqly5t1m  
m23M0vIHx44d0+ojQUVFrai48BIqLmhHxUWtVd625eaBjRkzhpYvJ/rf/4jeeUeX9dVXRFOnt1y  
i+5EGzhQ5wPM26sX0SefEE2cqKd9+y0p4aunl5RoPYKrv6NGEX32GVH37qT+j54O5s0jevvtIHbB  
/540yUzcgOh6SmtjFQhVnrdGjRpm6Tzgr7/0Bpo2LRI9/fbbk3Wxk7ITMM3YWvM7T9mpymoq43l/  
/fXXYHI67TU9P8iT3IVuN1wX2YznzVeeuqpdeokMDtZQpXtWwShPAXrv04bq0Co2vMKmrysiWUq

maZRpVSgkj2eTM8q48a4qdiXytJSheufLZVZw5zC/ufGbjEkruOIM7R+gOFZGAjW+fNJCv2d9/LL  
um0fc4we/+c/9TiuADdqpPPOOEPnsVDt0EGLWlBaqvO23VaPM8iDgQED+qph5c0GtFABkUGo5ptR  
ooA6PKmG01IttY+N2dNsvAqszzF63vT9+Ovj9PTEr8rwuynZE8rsFBaLoqawZQuVhf3ffDA7hQhV  
sTTLIFTZjjvuOB188xwVkfKPuIRqhQkNMK3BbfT2lbL1mcL+58ZuMSZbqNqkC9UGDfT4e+ZbAUuW  
EO21l85LF6pdu+pxcMcdOm+LLUyGYtkynQdbuZKof//+alh5k2sToZo09Yc6P62GQ1KbN8zGq8Aa  
TbfnTqUex+vpicHK8LspWQV1FVuqoSzs/+aD2UmEanbLIISLi4tNJBZURBLwsAE45ZRTyjWUTFZY  
WEi9e/cOlstHUGcLFiygOnXqhNZPmO2///7JZfOVCy64ILRuKrLPcF1eMXcuqfrTZnPUUUQHh0zO  
+ed6HM+uffcd0dln434oop9+wr13WpuwUD36aL0Mbg1gXnmF6LDDiE46yWQosJl69sR+oXtqBwww  
EzYQaCd4GK9evXrl6yWDUN1zzz2Ty+YFuHcDT/R362YyiK666qqgLTAbXcD1omy0GQ6mWcNvvmm  
WVKB+0jSu9U3YbidbLnllsn6yGa77bZbsEw+x7F//OMfQV2Uu+JhpzSh+swzz5gl8w9uJ7jtza6T  
TEJ1hx12CJbJ5zZml0K1ArqpWp/EE0+Ua1j77bdfIM6ANCCNXQ+vKLWz8847l6uzF12e8skzIEkf  
e+65cvXVuHHj5H3PIWlJuCf11FO1MGXDQ97oPV271sy0kWHXw8tPvkx7Td0rFdxVevKVJ83UPOaD  
D4jattVC08Qo0KNHD3r55ZeT7QvWqFEjmx5cjA9qNkvviDCw46XXEL0yy9Bfr5htzHErIMOOqhc  
nT344INmqpBOR1696LWXXrP2yARtuXBLGjNI3/Qux8p1eeGpF+io4UdZNZagO96+w0wV0hGhmgF5  
PZU78noqN9KfnG3VqpWZlSEvJ4qCzvtPM9Qttwy+IVMsNvY7rhfpH59fZ8H5sMNyKI55PVU7mBf  
5CSvp8qOvJ4qOijUMyBC1R0Rqm6IUHWn9UQIVP9S9QX7XQnVtSJUK6SsbB2huqt8mSorlITdwb7I  
++Uuo0WoZqPddCVUOY79oYTqShGqFSFCNQMiVN0RoeqGCFV35BOqbuDSq11f8gnV7lhQdceuL/mE  
anbkE6rREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0Q  
oeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQ  
dUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarR  
EaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV3  
7PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsiv  
VN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq  
7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZod  
EarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqO  
CFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOE  
qsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGa  
ARGq7ohQdUOEqsivN0QoeqOCFV37PoSoZodEarREaGaARGq7ohQdUOEqsivN0QoeqOCFV37PoS  
oZodEarREaGagVkrIVD9QjWib7SdOUSEajZEqLohQtWd1oOUUDX7JPbPRWtEqGaiTCU7jjXrJOI1

GyJU3bHrS4RqdkSoRkeEagZmqSaUDKdqZKQGRGqbohQdae1StgfOS1SSaiYQKhaqZIKQmZEqLpj  
15cl1eylUI2OCNU0IKx+/vlnqlmzJiWKE1SgkmpGOr2aoFtvuJW++eYbM7cA+vTpQ++++265nQ5W  
UKDqTv32798/ul1CSPHjjz9SgwYNytUT2zXXXEM9evQwcwpgwIAB9Nlnn+k6+IKZSch+qfLQxqZO  
nWrmzm9wqR/06tUrEAyon3JxrF+C2pzeJtnGeP5857fffgvqhPdDNT4/+btS+PHjzdz5zfcZr77  
7jvaf//9y9UT22mnnZY8Vkob0/z+++0ww8/lKsnGNcdtMeoUaPM3AljQtXwxhtvlGswgTVQZqcP  
lfEOZU2bNqXFixebEvKPo446qlx9ZDKu11tuucUsnX98+OGH5eoiiu200040Z84cU0L+cfzxx69b  
Lz2V2amuMpXP9Xr99debpfOPr7/+mgoLC8vXV4EyOw1WZk3fbrvtaMqUKaaE/OPUU08tVx+ZjNsY  
esPylZ9++omqV6++Tt1ksi233DKvBdg555wTWi9hxm2sTZs2ZmlBhKoCZzHpjSWwLEIVVrt2bapT  
p86mbQXKvIWWlhKLVR04WuHSQqsEK1VXFva/N2ZLSyUrSOLrJlvVKqtlIWLsr8qKlIX9303I0ve3  
wCoQqmwI9GFI5YNNVq1atXF0ElkWowmrVUm0spLxNyqope0FZWgrb57JZwZICqquSXU6QtIYW9r83  
VkPsT0vVVymRGIIn2azmmppWKSYNVpZQFva/N0aroewBZVZCOWmrjyiW3sZUbnBLYr4hQlWBSz41  
atRYJ3hHEap5Y18pW58p7H9u7LY+07vKwv5nPlgWoSqWZhGEat7YU8rWZ6qmLOz/bsy2PtNAZWH/  
c2O2O5StxzRDpXxDhKqBe1XLXZbNII SPO+44Wr58uSkh/+jYsWO5+shkXK/vvfeeWTr/GDRoULm6  
iGKHHXZYXt9ecuONN65bL1ku/b/++utm6fWd98ChV7VcG8siVHGP4bx580wJ+cftt98e1EOU/ZLn  
eeGFF8zS+Qcu4derV2+duslkTZo0oRkz8k9gMQ888EBQDy5t7MknzRLCyJUBUEQcggeNBETi8sE  
YVNCRKogCIlgCILgHaEiVc7IBEEQBEEQhFwSiNRhQ3rRF29sR8vHH0z/6LSXyplOa1ZOCu6NaF96  
Pq1aJS9qFwRBEARBEDYcgUj9+fO/04w5s2nW3LnU44v7qP9HRTRi+Lc0dvwoeuVlM2nFkvz4QkKq  
81h6kQVBearBEHJjou0FdWnMwFNp6uzVdNPN99JPX5xCA3p1pkefeo5mz5tHy+b/l2pX34hvXS1b  
Qb90e4u++muuycjEGOqUaEYvzjajgiAlgiAlGwHDhw8PPtSRbtOmTQu+5Lchwde2Wh93/DpvqBky  
ZAgdc+yxtGDBApOTmcSiEbvRL583pN+/2YLWTtyZlk8/l0b9sBkNG9yVRo/8jsb+WJeev6+Omd1m  
LS2aOZsWL19IE0eMpHHT51IZraF5U8bRktVmlmVzaPLspWpgJU0dM5rGTJhAM6fMogUr1hKtWUnT  
J4ygUSNH0YT5K9U8a2j+5Jm0aPkCGj9iBE1dsJLWLpxClzB9xnzTy1lGS+ZMpTGjRtLYcXPVEkwZ  
LVT5K1WxRMtp9sRZtAqDaxbT9LnLaPn8OTRv6WoqW7tE+TedJk4cR6PHjKV5KzCTomwJTVT/Z/ry  
QXRD3cPp1eCDP6tpppv1OjRNGHqLLW2REtnjaWFQcGrac64qbQUppUtp+mzlyBTEARBEAQhJ0Ck  
VvQ8EcThhmbZsmW0xVZb0eDBg4Px7t27U7MDD3R65imxz95b0huPb0WrxjemJaN2oyWj9wx+l47a  
iaYM3lmmKru7S10zu80ieufCQ6hai1tp+NRhdNtpZ9Aj/ebRuK5n0eXd9Cf3vr+2mO79dRW9f+6+  
dO5zo2JasK+o9c670R0DFqqpq2nBnDm0ZNVXdGH9s2iYEpDPNN+RarX+jxK931Dr5o2o9f0/0Nzh  
P9DJzc+jD+cupwXj3qcbL3uCxs1bTuPfvJoe6LMs+D9UtobefLSU3vldKcgB/6Q9WpxFn08mWvDu  
KXSc+r8/3N6Jbv9kEq1a8D/ap3YzeqefEqnPn0GNT3pDCd2ZdNf2Nem6T+fT5G/uoS0K9qf/zSca

+OTFdPbNX9CkccPppesupH+/P56Wvns2HfbceKJZH9Ax+zSj9t8ocT30CTrszl7aD0EQBEEQhBzg  
m0gFq1atojvvuotq192MPvjgA5MbncRP/b6jPn0/oqHfNVPCTCGtnbgT0eSdieYfRa89dyQde/Ru  
dMaZ55vZbRbSy6dfTM+P0v2Zo9+5jNr8dzTR0t/o7k7/oWmjH6cTL/1BTRIHV17yBEOL7vNcSu9f  
dBk99MtiWjXqK+p83kVUen0pnbD3iTRQTXtgzXpPv8HXv+bR863OpqdHof9yMb1z0UV0X98INPPL  
f9LB+/yd2l3diTr+4xb6aBi/hLqMfv3wKXrxu9702P9dSm//+DG9/s5r1P6QNtRfTf36Xx3pjK8h  
Uj+gjsfcRdOVK2X0DbXbuxPNnNODDjnmFd3zSjPoznpH0Ouz1tJrV1xHrw2fFuTOH/QSdXruK+XW  
53TdZffTK3c/Ql+P/4KuPfkeeur+O+i9UUrVColgCIKQd1x22WXBB27C7LXXjNzrX98FKkLFy6k  
c887n9pecQVdcmkbbkxudxIBfe9l2DRvQtz/+RO99/Dmdd8mVdHjLw1Xe9nSRqvgzTj2Vxk+YYGa3  
WURvnncQHXTzN7R4xp9095md6b2JSjSWLadPX7iZTtzzMHprup7vjZOa0lkvT6R5l76j43dtThcO  
WkHj3mxH5z/Sk+aNfpEOrnmcFql7tKangn8lI5468lR69E9lX0X0+nkX0v19F9KSCa9QqwPOpQF/  
TalPU6fR/GWpC/5rp/anTsfsSoc9O0bp2pH0bPsj6NAn/IJTVtJnN3UwPanv05VH3kJTlFytW/s5  
Xb5HR5pKs+m2LQvp2h4raNYP9lG9xL70jtKcAx67gM65vSfNmjaeXrvhenr2B1Wu4pPr96Qm7V6n  
laod9L//IDr0+v/RnKXBfQaCIAiCIAg5ASK1InlhUocNG0abbV6fli3FbZ9EEydOpJ122TW4RzYq  
iT9G9qHuX39Mdz/4BH365cc04Lc/6aEH76N3d25lvX/5hi4543Qarwpel4X0ylkX0b3f/E7Dfx9G  
Y6bOobVGwM/+ /n469/EeegSsWUh//fk7jR7Vj247upTemL5E5S2licOH0NAXu2jm5Em0jMpozrhJ  
NE/fTErzJk6huStQ4FpaMHUqzQkEaRktmjmr/IAr/vsff9KUBbr/U7OaZo5Xy+P2VrX8nEmTae5y  
iMcyWjxrBs1evEoJ00U0feJsNsDYTNPHTVcSVrFiFg0bOpRGtJ5C00dMoAWB9l1N00aPoGG//05j  
Js8N7kkFy+fPpOlzlqhSlfxdPlumzliQnCYlgiAlgpALRowYQUOVLgmzvn37mrk2DC+++GLwaevV  
q/khJc2cuXPpsMMOj/yp3ETbtldQl3a1aPno7YhmHU5lkxrS0F7H0G13HUtn9tyXTjv9NJo8ebKZ  
3WYBdT3xVLq7/ylzDpZrR2f/RW3a3UODZ/B37cto3Hev0z13303/bH8VHXnRv2jiilQPqCAIgiAI  
giCkE7xbqnPb7ajbi03pykv2oI5X7EETf9mXjmu9D+2xxx7BTllgCllgCIkwldmIX4AqCllgCllg  
bKqISBUEQRAEQRC8Q0SqIAiCIAiC4B0iUgVBEARBEATvEJEqCllgCllgeleIVEEQBEEQBME7RKQK  
giAlgiAI3iEiVRAEQRAEQFAOEamCIAiCIAiCd4hlzVfWrjUDhrlyM5AD0v+3qy/r03ef6skH1uf6  
V6bsipbxaTtF8SXOdXdlffmXa9J99mkdMvmyMdY144PvPmz3jXkbeoal1LhYvpyoRw9tLkyaRPTt  
t0S9epmMmJg4kahWLaL779fj06YRNWumtrja5K+8ovMwDHv88fLjf/5JNGdOajwq8+ZFX2bCBL3e  
vXvr8ebN9XKHHKLHs4F1cPUvKo8+W7so47Sw/vuq8czsXJlalkMbwp8+WX5+oiTd95xK9uu38sv  
N5mGkSPdytQsF+DBpmMEHieW24xGRHgZW69lWRkgOPTqlVEf/2VWjYqPP/f/24yDKtXp6YhDvoE  
+/Xii+XHx43T41FBPOU40Lgx0dy5On7l7Ek0e7aeh8vm8Wzw/HvtRbRwocK08LTvvzcZGwjsQzhu  
FBZGPyahTbG/FbF0aWqeNWtMZhZWrEgtgzabiSIT9Pb44Qc93rq1Xm7PPfV4VZg1S5f73Xd6v7EZ

PZro3HNTfrZsSTRggJ7244867+WX9bhQKVQNCrEwfnyqobrwn//oZRAY4gQ7zv77a+EI/vtf/X/6  
99fjYLPNdN5LL+lx9n/ECD0OQeAitObPT5WRDYhnzFe3rh4/8UQ9fsopejwb776r569Rw2TEyDPP  
IC/7rLP0+LHH6vFMbloiFb0Crm0hKh9/rOuqpMRkZMGUx9jdd5sJijFjUvm+wX4NHWOyQsCBGOsX  
9SAONt9cl3vffSYjA+wDTkBHjUqNR4Xnh5WWmkyFzyK1Zk3t19tv63H2EyfJLqCdYbkttkgJSi7r  
s8/Kj0PARqFatdQyiH82nA+hs6GZMUPX20EHmYwsoKOB/c2EawxxEanPPqvn4zhy3nl6HCcWVQHi  
+uSTU35cdpmZoIBA3XVXnX/ddXo/uOgifdzAyQu44QaiXXbRnVFCpVC1K0Ri8WKi558nateO6Mor  
iR55JHXGj7Hk05KNeQrrtBnXAsWED34oO7xQeP997/L96TgzGyfffQyRUW6NwT/B2eFd91F1LYt  
0Y03Ev30k1nA8M03RBdcoKeHMXasLhNii4HvyLPBdOzUmB+w/xCpixbp3k2YzcCBRFdfrXc+HJC6  
dCH6xz/02Wa6SH3qKV0XN9+sz9AZ7MDOPcB8+P/33pvqUeMzYXDmmdrCgj5OChBIEZzACy8QXXyx  
/l+DBxNdeKH2BwLrjTelOnbUB9aKzmqrj/pEr/KTT2pfWKRiO2GcA08mcHsAhDaErX2rAHpu0Bbg  
Q/v2uu75BAKi5lwztC1bRvTAAzoYoo7Te94QrOfrhW66Lb76qs7HevllJixz++16fa65huh//9P5  
4Pff9QkM2g9EHdrypZcSffCB9gMnTag7tLvp0/Uyv/2me7gPO0yP4wQB7RntAGVce63+X489poM6  
w/6gl/aqq3SbXrJEtxesG0APFU5U0FYrBt8glBNxxapOMjj5I7bTLpI5f+P/RMiA+vaqZPuhccB  
EGAfRhVd9kLZd9yh6x77qt1mwRdf6HVFm37iCb2tonLTTUR77KHbJP43b2+0EUxDLxB6KI84Qu8L  
DPZDtBv8T+z3b72le1q5HbNixXZDGVg/1N/nn+vpAAdlbGOuG6xr9+5EBxygtzOD7XPJJakrK+nw  
8qj3ggJ94gsqEqnYRx96qHyb530Z28Zu81hHbvOIMTwP+Pln3WbQxrBvzJyp86OA7YQ2wqKU/bRF  
Kul0rp6gjaJNf/KJmWDA9th9d70cYgJ8RB1zWYcfrtsqthF68Lhtffih3k8wbxgsUouL9S+EDsNI  
2yIV8eLpp3Usg6+oM7tHGMLsnHN0u0W8x3EH7QhtBnB9Ir4jTml6jil//KHZberU0Z0cUYCo3XZb  
7Q94771UflCQwzDaIEQvjid2XESsxj6FdUdsQHvEMWXY5PliFTz3nK7jf/2rfl8m9n/4ivnQA4x9  
GD5g3PXKZjqIA3/7G9F22+nybjGKerX9YzCOdQI4ocG+wj35gjNptSuEMmwYUb16uvHhkjl2NjQ8  
BBn0jPz6q76swA32yCN1YMOOi/EDD9TG0xG8AH633lrnYedq00Yf6DGOS0o4YO23nx7Hzss9Lai8  
vEwYnTsT1a+veOuYrl31MpnAdBiCWNilewgfjOPAiHrYfvvUPAgatkht0oRo771TZ5qoK7589P77  
RA0a6Hylc4hL9O6kw2VFOQvlgzAOInzgRvDFtollgsCC0lIf6EFmgYjACPGA+eEvxDOEM5cVB9xr  
jHVF0+BtCl9wELIP8qgXBNymTfU4tjHEECDJBPd+owz4i2HUMXPaaToPdYB153IaNtR1DCGDcRja  
J25hgF+8DE6adtpj++4oz6wsUiHAYgqDMO3nXfW/4fb8aGHpg6G4NRTdf4OO+IyeT4uKwwcIDH9  
tttMhoUtUnFVAL9Yf4jNsJ5UiCOMQ6i0aJE6OcL+jP0DYoWXQX1AuKHng+fBQRRwbzou56EtoUcN  
466XjQHaHP9P9gcindEbD/bY3ltuqfPQbrBPcdvEgRNwW8d82NbwH+uKPKw7gFCAz8iDoX3ZlpZB  
/WB6+uV8hpfHSQfaDNoqYmOYSIU4xXh6m4f/EBUQKrwMt3kWgoit3DsJkYU8lIF15pjDQtYV/p8s  
UnFpFvs5ykW85duOjj46JTzx0rrVVjof/qNnDvDVuCsL3w1ygYnE5jOV4vSYZF6zz16X8M4n1By

2SxSv/46tZ+iLiEfPA7/AC+DdovYgPn4fzz8sJ4HnSsYR7vD+nLs45Ndpnbt6Cl1nX/+U5cJ8c37  
+5Ah+hfGxzG0c4yiftBuESN4HuzbtKjF8RA+4xfjWHfUCcBJxTbb6HyOnfPPT51gVwWcuKNMiH7E  
SQzblrVv31T9stCHFSA4ThjgP/YN+IMYKVQKVZtCVuz75iAQ0fiwE+HAgp0P2Ac7glaNQM1nsejd  
QjDEdOQzLEpxYEEvGQsP9BJi5+UDTvXquocVYD70sKDXNQzszBAPtli4/vqUbxWB6bAwkfrLLymB  
xOsM0cnzplUnAkzfMmNgylAjwjyUGZF4CwU62mvR0WwSEV9oXcVvnBAxoEFdQkRwIGd72dkEYWg  
gh4diCD0UiAvDpGKXjz8b5THbQHgoIY8CCKsH4Zh6OlKWMjzPYp8oEfdcZ0gaOJAglCKs3UcGCCg  
hg/X07E+6MnCcrrj0hYMy/y8+WDRqpMf5Eq59PzL+T0UiF6hfDB1amoe7ilBjxzqG+VjOuDbTris  
MNCusd25bBtbpGl/4Ns+cNKE3ny7blh7bEP4gPbLoMcQ8+AgbC+D+7YZzsMBsFs3PYxy0lawPVks  
QrS6YotUXObIfYXbCYtUCBGMs8CAEOf7yvkyJotU9DRyvbNAXH5ngzxYRZeJ0QuOeq+oh5iXh+jl  
doT2a0+DSEX84FjBQgLwtufYxMtA6DisNtDDhytIGMbJEEQT6gdxkMuoDPw/Ea8Rx/l+QpSL8vF/  
IHSwH9k9+fAH82FfY7gs9EqHgbAAKU6zwMFVllgqDGPfte/fhEjFfoqTTIzbV4JwZYeXQazkZXDl  
iDn9dJ2Hdop9EO0F64c4ifXl2Axha4OTHgjYysAiFeuHtoyeb9QD+4f1QZztssnHBB9PE+6SMUJ  
KIMOGOTh6giDXlBkYb90BScJ2N68/bH/oPMJeXybCJ9M2iIV64GrMPx/UZd8vGaRinaOOlg8oVJI  
zUUBgRc9kTjzRGODIZjikiQfYNLvSUXQQJc/zpDRw4FeSA6wtki170IFoOKeAlzWwSUb7HwQNBjG  
jh4FiAf8T4ZFV7b71jAPLEykVvTwDOeli1Qb3nEh8BnuXcwkUl1gkQqRzyCglA+Xx1CPMBwQUJ9Y  
H8A9VQhUDE4+kBeHSOXePpgNAjfn2wdsW5jhgIg8XHICPA9EYxi4TIbpaHM26CVBPrYDTpa4HIbb  
HIQOwEGV58kkUnEJjME+wwOwWGJxbB/s7PWuDLZlXTAOFDjhwDh62uyyeb1xsLGxD4ZhVwwA56FH  
jwUKehhxAoR2hHiAdvTmm2YBB2yRip4WhoUdi1SexxbYfH9cuki1922+pxrxxobLi3rPZDq8POID  
4H0YvY48DeOgrl0BiGw7n4e59xXwZVX0ovPVHwhz3n8RB1HvsMrA/xMiFe384IP1OE4EON7iZA9x  
wr4Sxb2iYsKVe31dYZGK/wkg6DFu93qnX2lB27HhfFxy52H7gSe+NxM9eWj3EGEQpFyfMNQl+8Dg  
MjniH65+8D24UWGRIisTDE582D/ss3wCArPhvHSRau8nfBUDJ5tM+j2pLiDu8f+BofcenRcQw7i1  
A8dxFsa77aavBNrgnly0FxxT2A/7Fg+OQ0KlKJrLBnYOnJneeWcqQGAHY8HJwitdpOJSvj2OAwOL  
tYpEKnqG+B5Vvv8R/xM9kNgJ+H4/9HShN9PudbOBUEvvSUUQZl8qAtNhYSK1X7+U/wggwH6qc32l  
VNwLCUO9ZINFKuqP4XtLEYS4LnBAwf1nCNiA7yVGjwPqGgdM7l2pSKTi0g7qHweGbKBHnYWwHdz4  
1gkIOPvSp33AThepfEDF/WRYBm0TI6HRu/fpp/p+RQgyCBfuncJDe77w3LotQsTzSxS+T7RyohU  
+yBk9+ihFwy9QHx5FQdDu6wwMtVvukgFaB98GdAuG7c24KAMHzAMUGe4lxfzoD7Te18ZzkNvGj/c  
hQMXnyhif0BPvX0RbnPf+P7uTNgi1d5H00Uq995wjxHqg3vTqyJS+V76dNBDinVALAuDI2eRiu3M  
T1Gzof2iHL48jnsDGdwriDxc1sd68zL2fcwsUtHG+WQCl+C5Jx5XCFDvuHIA0PsNn+17FDPB/5N7  
UnEpFuN8uRvtA6INJ6r2bUaZRCr2vTCwvdCOcTUlJHSRijrh/8PGPaI82w5iF8M90/DJ3mft+/pZ



pOJWDnSCQKCizfBtLGgLOAa99poeZ9AJg32KwfywinrZbaKIVGwzvr2Fn7mA3zxPuki1jwGVEanY  
BmExxd7/GMQVHE9wiR/7G25FwLEA5SOe4NgFcMKNbcf3ZgN0ZOGKC7cnIM9XNoRKITWXDTQyu2cN  
wZfPqiA+OGDbIzuxg0PU8kEHv3wwgbHoAAiInI+eIBaTEBsoHzsFxu1L5ewPglwY6FHD/7RFLHoJ  
sEwmMB0WJlIBB3Qc/LA+fM8tzFWkclCBoeeAD0I2PH2ydetARYSJVPQA4DIX8uEvbzcc/Dk4QTCw  
f5gHQZwPHgg2YXA5dpDMBlSgXGZDgOW2gHIQRHFgxDgsk0jFQYYFL/uKYfRkMVwPWAfMy5eQUS8o  
my8Rwpj1JVLxy/d3oh2jTfl6wSqC6xf7UDphIhXAZ2yv9LLZV0xDffA+id5m1Gf6ySXDebidAOCB  
FlxjeT5IsjgAxyj86K8ASKqSIUo4PnwP2F8clwVkyr2zpcxbfgkCGI8DF6eYx7ANmYfYNx+Od7A  
7DaPX6wXluPpYSKVH+jC/fUYR/vB9uNL/ywg0WYxHSdCUeD/ySdN9m1M8JP3F+xHNmEildsp2hZv  
Mxs+EUW5YaSLVlbnDYbxPal9+qSOBfi/KJPnwckp4PEwkYpbRwC/Xg9tA/WJYxrikn0rEoCatO9J  
5bJZ3GYiikgFeJAR46gHtAv7eOlqUtGRxPOi/tJPtPikCVdFKgOfMPJtYgDtFvsh8rFNOB7z/eIA  
fp99tn64TqgUqkaFSCAw42wTQgsCC1379kESBxvswLhMDxGJsyz0CCEAoReUz6rx3jvcz8U7HXbY  
r77S5eLSlcYhynAmhvtVX389FVAZ9C4iKGJ6GHxPof2UbhSRiqfsYeJ1xcEGPTjciwO/sL544ALr  
D78AyoRBpCKo2Msw8AP+2k+qozz0QmEaDpj2gYrBJXhYINscsD0g8PkMlkE9YxpEPg7e6W9KAPAb  
PqBHAesHwYxLZniqOAxchkQPUvp6ZgJlQvBgfdEm4BP+L8ABm+sNgpVBPeMsnA9UAMugNxhtDOvD  
09D+WPBwW8X64CQIBzgGbQmCBve1Mlgf1B0f3FBn9nbEE/j2Mri8Bb/sgyv85mXYD7QXHJzQm4b2  
/dFHqXvvMrVF7EMo3/abQbvh/4NhG/SWwE/71g2Ah5uwb2F/wbpCdLOQxpPd3M5sUB9YX763F6Cu  
ef9Of2oYva3ogT/hBJORGfT6ZdBGsd6lDwD7BLYlemowDT2UuOUAdccnJvAFy6Bnl6koPqAesM0g  
VFBWOtgH0Etp34tpY8cHG7QP5GN97PaLXk5u82jL2N9ZxKKNCB3Yy8A/rI99yRpxA/loQ+i15P0G  
YB/FbT18b2w2eFvbl7BxQo4HlIBfuNQfdoUKcQN+YT0YxFnsX2grYeINvmEZ7KdhYP/EdkpfFvWL  
toe2jGMBAS9xjzTaArY77pWG7wx6+LCM3QOMWAH/OF6DUaO031hfrHdYzzpEKnoRGWwnXOrGMSwb  
UeID9j/EB8RBzMdxm2MDRKq9r/P+CsLiA+bFPok2AsHKJ48MjsHHHRf+MGYU8EYQ1CP8tcG68PEb  
ftknCICPxenHJSEyGY4UwkYNLnHjEhEfUHDDPXyW3OQf5ZJNOjjw8SUPHPAhSlu10uN4KjOsJ3RT  
BcERI4LsewWFdcGBAe0DDxSgpwa3IPAT3Jva067YH9DbAiEeF3wPPO4pR/k4EPI+mC6q8xWIOPSc  
oa0JVQdiD0+oo6cVb6lh8PAwrhJUdKuIkzj5415rCECM87uy0ZtuPygVB7h1BFET+Fa1DQV6XnEC  
ZZ9sCE6oFiFskqAnF09V494ugODCl/Pss2oX0FNlv3oFwQXSXU3FfXz4BQY5LtkJ20LPAr2+B4R5c  
9DhsavWHfcD1AZNsoCcVlzT5Mi8M91DbVyTyHdxizPcmCIUDPey4nl6eVLu3HW+WsXuw4wD367Mw  
heFWCryFBldB4ga9+hs65qDXFetVmYcrhSSqBgVv4MukQuWROhQEQRcETQIRqRsa3C8E4/vpcG8N  
n0nmAtwzXg8A8A34GxsVfflDEARBEISNFjmqb2hYTPHDULhEiAca0h+O2lCwPxCreIhkYyT96XNB  
EARBEDZ65KgeFTwtiPdc4olI9H7i6en0m7v5SXLccwfDk/wMHmBCHospfH8Yrx/CPHjdCX/CkC9X

4/5P3DuKJ2PxxKI9wzceYoIPmIZ7svCkJobx7kD7vhs8XYonEu0nUhk8kYsHMDgf3PvGTxXjiUS8  
55XXA0+U2gKWn/LFPUq4oR5PBOOJYfVhBeTBJ/wfgPvo8LAHPzHPdYWnVPEkK9Yx/SIX/j940AtP  
oeLJe37KE0+Qwn/8f+IJFQRBEIRNDjmqRwE3jfp3zHFpnC+P4wll/uwflLuHjxm/k451v/O49fC8e  
4CZxfh8ID08qhNDlr6pgnOFPdfK77PjBCf4EI4tdvN8OX5eCP/x5O7yPjZ/ex+s2klcy0sHrdPgT  
hDA8aYIXm0CQ4l2EyEO5/H5D+MevO+JlsG74xcMcelgKN9tDzOlzhzwPv3qDXygPYY7X0PBL9FE+  
btTn/wlrZXAZ/MAXvvEP+GZ7+lx30/FLowGCIaICIGwSyFE9ChBWeEE+fzElgpNfBYN33QF+ATPe  
6QbwXji8SxMCyv4iCYsp/k4492aySEWPJSbxxSi8dgbgyXx+WTheaQHhh2G8BJm/oIFXXCAPxk+7  
4ilJvNcO72utCF4GPbloi4Wr/Y1o9HYir0kTPc7L4JUe6AnG/4Ngxdc+8F4+vs8Wr+S56KLUMnjv  
Hm5vgJDFS7jRAwywPL8s+fjjdR7g/4NpuKQP//gTdvgKDd5ggJ5nPBHK8wqCIAiCsEkgR/Uo4LI2  
xChEGb6KgV5BfBEKooHFKoukbK+H4fn4Mni6SD3tND1uv6MO4PYA5OOda3jACcMQyjbIg7m8koWX  
gRDG7Qs8bmN/LQsX23kYPcwMeoWRh+8cQ5zDT9yugN5l/soRenYhKvHQGAQ/eobRy4uXLPPXRsjE  
Ku7ZZfhk4IILTiy7CtKgiAlgiBs1MhRPQoQVBBAuLwOINogKpHHIhW9jBjHI50YfiEGn1/EC/QZ  
FIP8oFS6SGUxipee8wvvy0fvll7jxjXkWhNIEKr7ogXtpM70fjpeBSIXo5E/O2V/W4G9E82fueBn0  
ZNogD+uLdUGPMI4KjTyIVuShNxTg/3AZELow/nxpmEi1X4SMS/7lg4jnL1Hh85k8ryAlgiAlmwRy  
VI8CPrfG3wXHpWclMRZFd9yh58FIdVzuRh6+eYyXlmOY7yNleLnGjfxXafhzpSxS0dOlb9kjD5fe  
cT8mLv1jfM89tUDmb3NnE6l4kT/GcX9sRfAy/CURPGSFCQhyfO8Z64Jx3DPKYpuXSRepnTqlpkFg  
4z5d/sIQ6oM/f4jL9nzPLsQ9vozFy9nfOOY8W6RCsPMJApbFi+K5VxsmCllgCmlmgRzVo4JeQliw  
667TI5+//VZ/SQJPnPM3/HffJ57+x2cMO3TQT8gD+wXzeGr/6quJLrIEf1YT5Zx/PIHbntno6z4t7  
PW+8kahNG/1r92ziaXYsg/tTbdDLCuOeU3zb+cLlT58Vgflx+4L9ffzx4/U3pa+4Qi+LB7Wwbgwv  
k/6FHdzCgGn8/yBKUQ7uS8Wn72zwGTzcX4q3GuC72Oh1xbfDUWf8KVf8D1j6t8LR84r7XIE3+MX9  
vVgWJgiCIAjCJoGIVEEQBEEQBME7RKQKgiAlgiAI3iEiVRAEQRAEQfAOEambKiNGqK2rNm8uHibC  
S//xwBc+RoBXUfH7UAVBEARBECliinVTJW6Riif7+ctOFYGHvvjNA3gzwXvv6XegYhyvocKnUAVB  
EARBECIQk4LZxMHRlL76Sr+KCp/0xMvj+Vv2eN8pviOPp/Tx7Xl+zRK+f4/5+N2gAMPI469NoVy8  
HQBpugOIPHxlCk/2QxT26qWftLfBU/5ffKHff4on4vGGAP6e/dix+n/07avflpAuUvFtfqxD+jfy  
K2LpUv3eUi4nvbx0MD+/cqp/f52H11ShRxUclZ8hKwiCIAiCklUMikNlgi8nsUA74gj9O3Bg+Zfl  
s+ELS+gxHDuq9X5TvL4J7wzl93vef78ut7RUj0OQqMji9VZcjm333afnB5zXqpX+xedSAV7lxNNG  
/AUnGHPooXocr4mKCoQnXk/FHzSwy0sHr6TiefCVLqZmTZ33wgsmQxAEQRAEITMZfLeQxBap6FnE  
F5g++USPb7edfg/oZZet+/35k07S43if6jvv6Bfi48X1p56qp9etS7T99lrA4l2rKOff/9bTAAtr  
vAifwTgMlvTJJ3VvK39bH2Xx9/6vuSY1L4NeVLxLFD2/6aCn+Jxz9HtV8aUs+9v9AD2+6eWlg97l

sHIYnMNPQRAEQRCEGRQHEISFqn8WVRw++06D8IQX0li+9vfiE45Rc8DYYp5zjtP93RuvjnRZ58R  
1a+vxSWm4TOffIsALsPj5fWYvsMO+rv2mAdfmmIwDsPDSQx6RpFnf63pjz9S80bh0UdT88P44wIM  
enuzIWf3pKIHLuGeVHmAShAEQRCEiGRQHEKSMJGKnkbkQcyxyJw1S98TilsBGHwqFb2eDRsSde6s  
e02xHHpZ0bOKXlaAz53y5z1xHyfuTz37bD0eJlJxfymDL2HBN3wilL82Zff+RgG3KGBZNV6KFhNF  
pGld+DOquK8W4J5a3JNarx7RDz/oPEEQBEEQhCxEVDB5TphIBSzlchKfnzmFEMM4HmZi8AIU5MHw  
oBM+8XnAAXp8iy30A1AAD2TVrq3zjqK6OSTtYjFOF7jxHBZtkiFqOVv4eO7/+i5hTjmeRn294QT  
TIYDFYIUzkPPLeB7V/GgFD7byt/lv/JKPVOQBEEQBCECaYpDCAUCDfebHnigyTCgt/OJJ4iOPlpf  
5sctAOk9kFOMEO22m54OIYpe11tuldpnH6lbbjAzGXA/6Vln6YezXntNPYGPpy/C49xRvEQB4eh7C  
j0WhDR68OuggfX/s5Mn6u/b4dj6Db91DIF9/vclwALcX4N5Y+G3TpAlR48ZE48aZDMXIkVqgtmyp  
77/INxygvgRBEARBECIglIUQBEEQBEBHwDhGpgiAlgiAlgneISBUEQRAEQRC8Q0SqIAiCIAiC4B0i  
UgVBEBARBEATvEJEqCIIgCIIgeleIVEEQBEEQBME7RKQKgiAlgiAl3iEiVRAEQRAEQfAOEamCIAiC  
IAiCd4hIrYAYlQRhfSJtTBAEQQByPAhHRGoGGquUsJlgxE0Llew2tlwIQRCEjYkLVLLj2HiVhMzY  
9bWlSkI4orwysI9KdkMShLj5m0p2GxORKgjCkYblew4NkEIITN2fTVSSQhHIFcGRKRWjbyuXyR  
DRGpVUPamLC+kTaWHRGp7tj1JSK1Ykr5hRAEpWVEe5XtIWxESCuWrzBzCOmsWLGCSyYQYmEqilj  
YO3atcFvNtaWRZtvk0K1scPWHma1sATNXz7fBTBSWblyJS1cuHCdNrZmzZrgVwjn+OOPT9bX008/  
HeyrQjirVq0KLL2NrV69OvgV1qVsWRldvOZiK4olaOSKkWaqa7aF/oi7PpqWNZQTTAzcOUQkZrG  
OeecQwcfLlAOUMNSjahAJeQdeeSRNGDAADN3fgMxjwNey5Ytqbi4OKifggJdT2y77747HXbYYch8  
81fOpw+//5BeeK8rfdDzA5q1bDYtKltMwyb8SU8e+Rz9Pv5PWqPSKlpNS1QaO3MsDRk7mPqPGKiW  
+5je+PwN6j30Z5q6aFpQ3sbKZZddRocffriu0966fSfXGzv66KPpm2++MXPnN9yLddBBB1GNGjWC  
+klvY7vttlvQBoH0emk6d+4cxCq7nmxr1aoVffDBB2bu/lbbTPPmzalu3bpB/aS3sR122lFatGgR  
zCdtTHPLLbcEsSqoo/8qs+PYVgk65phj6KWXxJz5zd2G9tiiy10G0M9cZqUoG2qbRNMB9LGUohl  
NaA3BmLKDkyJ34Pmk0poWCZ4vfrqq8Fy+dyYvvvu3J1ksn2TxxlPyR+pHmJRTQjMZf+SoyiDxlf  
OzuJd6ln4jv6peRXeqjal3RP7fvooboP0/11H6QH6j5EL9Z5mb6t9QMtSCym1Ykymp2YR71UOS/X  
eoXeOPEtGtBv4zphOOGEE8rXzU/K7FRDmZn23HPPBcvkcXv7+eefg7rl1sYwvaSkhKZN27hPYKoC  
t5Ozzz47tI7C7OGHHw6Wyec2NmjQoKAuorQx/E6cONEsmX9wO7n88svL189ryuy0kzlz7fbbbw+W  
yec29vvvwd1Ua6N2WmKMmv6qFGjzJKCiFRDsuHYFiJSbdtvv/3M0vIHp06d1qmPiqzpnk3p4+qf  
0XuFn1DPwt40t3ABLUoso3/u9y9q174dXXT5hXTeMefT5OrTaXSNsfTF1l/Re40/oFf2fJVe3PVI  
er9xNxpSaxiNKR5Ps0vm0eqSMlpToJxQrffugx7RDm0EhNVNJpEK22mnnczS+cdtt91Wri6i2ogR  
l0Wj+UdYfWQz9OzkK0888URonWSzfL6aFIYfmUQqrHr16mbp/OONN94oVxdJs5MRqbZ9/fXXpoT8

Ju9FKp/dpTeQwLKIVO6azwvSToK7XN5Infols70TB9BDRU/QhwXdaUDhzbzSpcBpNLZxJ3bf9gq66  
rJTatWunrD21vaYtfVTvYyKITyNAbZtTez6dduHpdPFVF9NIF15O7+74fpD/VPEzaaN97D6ySZS  
GzdubJbOA9La2IM3PliuLqLajD9nmBLyJ7D6yGYNChuYpS021U6vtPV6+aGXQ+skm43olydC5SyL  
SK1Zs6ZZOv/4+OOPy9VF0uwUILL79etnSshvIAHyHjSIOEs9WURqXlkl06WXt6E+hb8kxedqY2/s  
9Ra1uaoNlbYrDezyzpfTR5spkapa5OoQm1VrLp17wbnBvFeVXkW3Hnc7LU4sozeK3gr310MLbWNZ  
RGpeWQyp3H1edvplWdj/3JjtOWV2Ksuw/tmSWnadFPY/N3aLIVVYx72Uh3PTcx41gWkZrvFlpn  
dkoTqTx/Pt8iwajDv1CrVq1yDSRpIlK1FSurZLr8witpRcHq4PI8C1TYa3u/Tm1K00XqJ5FEamIp  
Kd1y/G00P7GYXit6M9znjcVEpKZsfabuysL+58Zs1sMq6yVVUxb2fzdmW5/pZ2Vh/zMfTESqu9kp  
pCcVJiJVRGrAp59+GtpARKRaVsl0wL0t6dHiZ2h04XjqUfQd9S0cSGuVSO3RqCddcfkV5UTqh5t9  
VE6krrGGZ9aaQ2ddfDZd2elKuqzDZdR1lxeDeZ8teD7cXw9NelKzWAXJellrWP9sSXpSlyfpSQ2J  
YyJSM5prTyqsS5cuWqDkOeowL4D0BhJYFpG67777mqXzgLQTumsvuXad+qjl9t1hP3qp+DX6qqgn  
/Vk8kt4t+oi+TfxInY+4hi5vczm1u6g9dTizl31X9weaXH8ajdxmNP3e6E8a2Og3GrDtlBrcaChN  
rT6d/p+98wCQmlqj8LALSy8CYgFFxA4qKgoqClhQVFBRSliAhbsFZ8dwd57Q+yioliAhgWld0Sa  
dAXpvdF/3ZOyeyebMnd3ssLO+e7738qkzMw/f25OTm6SnrUel167vyMjSo22K1f3fvda3+gnQC/  
3ESJVNz6JmPw1Fj3m7sn5SLVWPDnAmcNmQe+f4mEzw4xJColKjllaxRXA8fzvV7t8apvTqi8uDJ  
zhoyD3z/fKlrRKRI3qysLGfpzOOzz5LzpUbevMRqQMGDHDWkNIQpDr8/vvv+YokSqRmOpUrV86X  
E98okZC22Rfj+sQm2Vpiu2xV/RVO/68rsVE25WyVDdmb5Otd+lsXU3212zfy5e5fS589P5XXa70p  
vWr2ljEVx4tkq/2LWma7SjtibmKePJX1vHzd72vn0+z4uLchSYolkZrp1KpVKykfUXH77bc7S2Ym  
c+bM8c1LWGQ69erV881LUHTs2NFZMjNzsmRJ/rxEOKmpPtSluNKKSZOkfFihN49IPffcc50lCXso  
DdybzC0Sy40IEKKXXXSRNX8mjxdxv/uDDz6Ymxdv6Efb/b/vL+P/nCCDPx0qQ3sNld/7DJbhI0bK  
yDGj5PvvBsq9hzwgX3z+tfzxySZMGGijB09Xob+MISGfT9Mxznz/hwz9dLgM/mCwjBgwRgYPHSLT  
5uGJJtf/nGPRdyOxcoPaixApJ511lnW/KwxkSeeeCK3jsKct2yxwZPfqISpYuXE7zSj+xpuxA5Y  
YylvvfRSvjz5Rb9+/az5M53ly5dLzZo1rZxY/ZhHpJbYx64xPAAh05/W5dZYr169kmopqWki9eOP  
P7bmJzYUqRpuMW3YsEEO2uMgKTujbF4RqXZ0w6Ot6SQ/F154oRxxxBFJGyHE2PDhw63pmbwj1NHZ  
cOCeB0qFcRW0CkvloUdn0BASQzp16mTd9k2vsZycHPn++++t6awxGz0PBxxwQK5gdQOVEX8wDhDC  
Ss8XHHlRp08fazprzEbPw/577S/Vvq6m9WIJqdOsjjOVeOnevbscXf9oLVsJyV6QLa89+po1nTWW  
DEVqCPVV0wuJhLny5cqkzr1ly5bOFBJEE9Xc+kLbqBoJxvtc9caNGztTSBctWrVKyhlcMBKOnq+D  
Dz7YeZUE0V41la3cNlc1Eo6er5qqEX+ovEKgSDVjxYoVSZ17UYrUJe8tkdnXz5b5Pec7r0Ssz9IOI  
9ji9IpaJ8cDWK1I3qEaC2bRpU1KNUaRG4xWpy5Ytc6aQIPR8UaRG4xWpc1Qj4ej5okgNhsorBIpU

M/5Lkfp3t79lwiETZNrZGKuaGjM7zJThieEy4eAJzivBDEkMkd8Tv8vCF9L7NCOKVDMoUs2hSDVH  
zxdFajQUqebo+aJIDYbKK4jtFKmmFIIDXA1N83eJOtGr5MNk1IXeSYidUTJETI0MVSWvLvEeSUN  
qO9CkWoGRao5FKnm6PmiSI2GltUcPV+WSOVQVF+ovEKoX1uJVK2zluEUIZMKRxMx99a5Mma3MTLx  
8lky7dxplts5ru44Zy6xBOW42uOseUdVHCUzr5gpl7JHyMickdZ0V6SO2X2MzLhshiVCBycGy9ia  
Y2Xp+0uteTYv3Cjydx1tzTe8xHD5LfGbzoO4y5qWDpo0SL41CS7al8FQpJpDkWqOni+K1GjaX6BE  
qpazObMoUqPQ81WzEp3UIKi8QqhfnyLVhKISqRCTImjMGi5ja42V6RdOI+kXTbdd0fq2K7rso2WW  
aB1TY4zlrn6at0nG7jHWXk4FcEXqqAqjZNkn9o77n//9k7t+nVwn9f00OqkK7/3zKFLDoUg1hyLV  
HD1fFKnRtG/vEalzKFKj0POF23kRf6i8QqBINaMoRSpi0SuLck+RzLh8RpJl/ePwP2RYYpjM6pLn  
ei5+c3GSAPU73b950WbLTUXouCJ10evqPdMIRaoZFKnmUKSao+eLijUailRz9HxRpAZD5RUCRaoZ  
RS1SV/640nklv0ideOREW6RelWaR+gZF6n8JRao5FKnm6PmiSI2GltUcPV8UqcFQeYVAkWpGkYvU  
74NF6tL3lloidWTZkbL88+WyZtAaGV3ZGVtaEJGaM8Ja39zb58rWZVudVwsPRaoZFKnmUKSao+eL  
ljUailRz9HxRpAZD5RUCRaoZRXnhFC5gWtF/hfOKyF/n/WUJy7F7j3VeUZ9nwAprzCrGpk4+Ybls  
eGqBJW4hNgHGslrL1MxbZvOCzfJr4lcrdJZ9tswSuXjvf+7+x3m18FCkmkGRag5Fqjl6vihSo6FI  
NUfPF0VqMFRelVCKmlEkltXvNh0+ry3utVjm3jZXVnyVJ2Rx6h8idcyuY5xXNP6j239QpJpBkWoO  
Rao5er4oUqOhSDVHzxdFajBUXiFQpJpRVE5qKmyav8k61Y/T+e7wANwNALeiWj9xvTPXfw9Fqhku  
qeZQpJqj54siNRqKVHP0fFGkBkPIFQJFqh7kkgF2zZsk/VT1svaUWtl3dh1snHmRtm+ace6YzJF  
qhkuqeZQpJqj54siNRqKVHP0fFGkBkPIFQJFqh7mkjdGaBINYMi1RyKVHP0fFGkRkORao6eL4rU  
YKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxR  
pEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1  
gyLVHlpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHlpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsor  
BlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQU  
qebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLN  
oUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHlpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1  
gyLVHlpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6e  
L4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz  
9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLN  
oUg1gyLVHlpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHlpUMyhSzaFINUfPF0VqNBSp5uj5okgN  
hsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdF  
ajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQz

KFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxC  
oEg1gyLVHIpUMyhsZaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhsZaFINYMi1RyKVHP0fFGkRkOR  
ao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUc  
iIRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQz  
KFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHIpUMyhsZaFINUfPF0VqNBSp5uj5  
okgNhsorBlpUMyhsZaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVH  
zxdFajQUqebo+aJIDYbKK4T6NyiReo4qlsJTTFUUFKnmUKSasWmLEqlqW3S3y8Z3UKRG0aqHEqla  
P7ZsNUVqFHqNHdyVIjUKiIRzEqfk1VjNKylSg6DyCqG+aaqEchsJhyLVHIpUMzaphm3RbY1VI+G0  
Uk3P2TLVSDh6vg5WjYRDkWqOW19oNVUj/IB5hUCRagZFqjUqWZQpJpDkWqOni+K1GgoUs1x6wuN  
ljUYKq8QDIUtt4y226navn279Zfk4eYEAkvqM4880zrdRLMiSeemJQzwBoLB9ui245TjYRztmq5  
GVO5W6caaywcvyYOUY2Ec+WVVyb1YwsXLnSmkCD0GgulGvGHItWH+fPnWxtaiT9L5BYRWk6pHGcO  
4uWwww6TUqVKJXVUiL59+pzpEJ3Vq1fbNVZC1ZgnZ8SfE044QUqVvZWmt98S8ubTbzpEJ2tW7fa  
NfZNcj+WqMAaC6J169ZSKstTYxMT8mi3R505iBfUWFZWVIlfhiD+XHHFFVlyUVKvME8nZA7Ot3h  
zEF0WEkaX331Ve4GViKhOnbVOSU1Z5p7tWcmuxHudz/ppJNy8xIWpXr0sObPdH755ZdcYeonUN3Y  
d999rfIZY2ING8nNTY4KvQ1X4Uy7++67rfkznVGjRuUeMFr92FdOrtxWTYWaVqtWLdm8eTNRtNGm  
TZvcOrJCb5NVOK/feOON1vyZpQpU6RcuXJWTVz6Mfe1GjVqyKpVq5yIMhO3xi655JLkPOLtngm  
9U6dOlNZ/J2qUOR6vDSSy8IFxAiQKS6G+CsWbOcpTOPpUuXJuUilcApoS1btjhryDzee+8937z4  
hZvXadOmOUtnHhjjB1hUo15ReowFe40FRABuE1Vpvl5558n5cMKr0itqkK97uZ14sSjztKZBwQU  
hFRSvhB600Qq4uSTT5aNGzc6a8g8vv/+eysPqfT97jyJR492ls481q5dK3Xr1s2Xm6SmiVREw4YN  
Zf369c4aMhuKVAe9QHijQKS6cfjhztLZx5du3bNI49U4t9//3XWkHn45SMq6tSp4yydedx33335  
cxlhUhFTp0511pB5eHNhRYBIdaNatWrO0pnHc889l5SL3NCbR6QiRowY4awh8/DmlpUoU6aMs3Tm  
EWhO6M0jUhHffeds4bMhiJVccMNN+QrECsiRCo69+uuu86y54t1dFGhtS6q1RtSTxKvqTy8bhAv  
JOTSdZdqa3LaMBVXqPB7750x2qsYp0Jr16qWeNEJxFR8aOK0IU1fV1Wu0qF33sXk+jSpYsceeSR  
+ba5VERq27ZtddZ3AN9kTcXVKSIVAgIHHT6rbM4R+fOneXYY49NykVu6M1HpJ5zzjm+6yxWcaWK  
71VoDX1R4hWVA5++Kiyy38r278duVOH33sUomjdvnp9+rNCbj0jFRbU85U+RauGORckXES11Y6Kk  
ijjbuYr83ndnji9UxNn83jMTIgWRYvBEhEhl+ITeFERqxsQTKuJsu6jwe99MCL35iFQERSpFqgWK  
wXd8DUWqHRSp5kGRGk9QpJoHRap56l0iNb5GkWo3itRAMl6kukXgVyBRlhXP23355Zfl6aeflt7x  
jAqtPadak75NJHGTysPNBnF1Qh5Y9YC2JqdNVvG4Cr/33hnjMRUzVGjtRdUS1/jkJKqPlxVXIFN  
X5fVnlLh997FJJ599llp0aJFvm0uFZF61113+a6zuAf6lm8urlgQqeXLI5dXX33Vd53FOZ555hnr

tL2ei9zQm49lvemmm3zXWaziCRWjVGjtVdUS16sc+PRVYVHqjllWsvq6rPaSCr/3LkZx6aWX5qsf  
K/Tml1IHDx5saZNMh06qQ+XKlFmVSZRlXZXE mUrPnj3z5SMqcFucxYsXO2vIPKpXr+6bl7CAUMtU  
fO+4ESFSS5YsKbNnz3bWkHnsscceSfmwlkKkYlXmpvL+++8n5SI39OYRqbgnaCbfEaF27dpJ+Ugl  
GjRo4CydeXz99de+OUlqHpGKM7tDhgxx1pDZUKRq5LuPWYhlzeRbA7msWbPGykUqtyLZZ599nKUy  
m8CLW3xi7NixzlKZC271g1zk1liASMx0qIWroKtInvfcc09uDVkRIIHDRrkLJW54LZ4yEVSP6Y3  
R6RieunSpZ2IMhs4hLm5igjcfzzTwRIb5CKwxhyR6k4neTAbHnBLkptvvtkuovFOAalWQjW89thj  
j2X0/VF1sOfT27ZNHnrolavz1jcyN3A67f7778+dn4h1arVbt25JeUK4uXv88cflzz//dObObNya  
eeCBB6RKISp2nrAtuu3XhLQ4toXce++91nysMZvevXtb251VW31zs5Xbj6HGRo4c6cyd2bg1g3zt  
ueee+WtsQkKOR3e8Jf4Ba8zmk08+ke7du9s1poXbj2Ff+euvvzpZzZuzaCf2m+//ew86TU2NyEN  
ajdgjflAkRrCYarlfHgF3R+ImxMIVr2zatWqIfU6CaZp06ZJOQOssfzoOdGfeX28aiScc1XLZzJk  
3RbVWGP5Caqx+qqRcK666qqkfz58uWssQj0GttLNeIPRWol6JzyyoipigJPCNI7KjzOkoTTpEmT  
pJxt2LDBmUL82KQatkW3NVaNhNNKNT1ny1Qj4ej5OlglEk779u2T+rE5c+Y4U0gQbn2h1VSN+EPI  
FQJFqhKqUqeZQpJpBkWoORao5er4oUqOhSDXHRs80itRgqLxCoEg1gyLVHIpUMyhSzaFINUfPF0Vq  
NBSp5rj1hUaRGgyVVwj1z1Eitb4qIUQtpioKilRzKFLN2LRZiVS1LbrbZeNLKFKjaHWdEqLuP7aX  
EqkrKVkj0Gvs4NYUqVFQpJqDbdGtsZonUaQGQeUVQv36SqRqGx4JhyLVHIpUMzZtUiJVy1fjxhSp  
UeACRj1ny5ZRpEah5+vggylSo6BINUfPFx4MRPyh8gqBlitUMilRzKFLNoEg1hyLVHD1fFKnRUKSa  
o+eLIjUYKq8QKFLNoEg1hyLVDIpUcyhSzdHzRZEaDUWqOXq+KFKDofIKgSLVDIpUcyhSzaBINYci  
1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBINYci1QyKVHMoUs3R80WRGg1Fqjl6vihSg6HyCoEi1QyK  
VHMoUs2gSDWHItUcPV8UqdFQpJqj54siNRgqrxAoUs2gSDWHItUMilRzKFLN0fNFkRoNRao5er4o  
UoOh8gqBlitUMilRzKFLNoEg1hyLVHD1fFKnRUKSao+eLIjUYKq8QKFLNoEg1hyLVDIpUcyhSzdHz  
RZEaDUWqOXq+KFKDofIKgSLVDIpUcyhSzaBINYci1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBINYci  
1QyKVHMoUs3R80WRGg1Fqjl6vihSg6HyCoEi1QyKVHMoUs2gSDWHItUcPV8UqdFQpJqj54siNRgq  
rxAoUs2gSDWHItUMilRzKFLN0fNFkRoNRao5er4oUoOh8gqBlitUMilRzKFLNoEg1hyLVHD1fFKnR  
UKSao+eLIjUYKq8QKFLNoEg1hyLVDIpUcyhSzdHzRZEaDUWqOXq+KFKDofIKgSLVDIpUcyhSzaBI  
NYci1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBINYci1QyKVHMoUs3R80WRGg1Fqjl6vihSg6HyCoEi  
1QyKVHMoUs2gSDWHItUcPV8UqdFQpJqj54siNRgqrxAoUs2gSDWHItUMilRzKFLN0fNFkRoNRao5  
er4oUoOh8gqBlitUMilRzKFLNoEg1hyLVHD1fFKnRUKSao+eLIjUYKq8QKFLNoEg1hyLVDIpUcyhS  
zdHzRZEaDUWqOXq+KFKDofIKgSLVDIpUcyhSzaBINYci1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBI  
NYci1QyKVHMoUs3R80WRGg1Fqjl6vihSg6HyCoEi1QyKVHMoUs2gSDWHItUcPV8UqdFQpJqj54si

NRgqrxAoUs2gSDWHItUMilRzKFLN0fNfRoNRao5er4oUoOh8gqBlItUMilRzKFLNoEg1hyLVHD1f  
FKnRUKSsao+eLljUYKq8QKFLNoEg1hyLVDlpUcyhSzdHzRZEaDUWqOXq+KFKDofIKgSLVDlpUcyhS  
zaBINYci1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBINYci1QyKVHMoUs3R80WRGg1Famq8+sq70uGK  
96XVmV+qPN0mpbLvkJZt0mIClfJnXe+L+PHj3XmJC5UXiFQpJpBkWoORaoZFKnmUKSsao+eLljUa  
itRonnuut8qNSKkSltnqb8kSq6V0yQWSpf47R72Gv6Wyp1dzbrEXIBZUXiFQpJpBkWoORaoZFKnm  
UKSsao+eLljUaitRo6td/yRkoZXNmyskntZMuXS6Ta67pJO3anSP71nIMcrJQd1ukX7++zhIEUHmF  
QJFqBkWqORSpZICkmkORao6eL4rUaChSw1m7dqGUKTnOEQJHNnhaCdTO0rlzXrVf51ULD/OclOv  
7tLLWYoAKq8QKFLNoEg1hyLVDlpUcyhSzdHzRZEaDUVqOJMmDVEidaWUytoqxx3bXeXrMrn88ry4  
6KJLpeYej0vJxFY59eSXnaUloPIKgSLVDlpUcyhSzaBINYci1Rw9XxSp0VCkhjNz5jgpk71YKpSb  
KeVKL5PSJUvYspOjYrm/JKfEOml91lvOUgRQeYVAkWoGRao5FKlmUKSaQ5Fqjp4vitRoKFLDWbnq  
HylTarKcfrtUq3KXPXfooTq9tyASD2iQT+pW+cT6Xjl685SBFB5hUCRagZFqjkUqWZQpJpDkWqO  
ni+K1GgoUqNYL2VyvpRjGp0rlSvMyCdSSyuRWrfOe7JXzZ7SrdtzzjIEUHmFQJFqBkWqORSpZICk  
mkORao6eL4rUaChSo6IS+WNp0eIG2aXS375O6mH1v5F99nlTnnn6fWcJAqi8QqBINYMi1RyKVDMo  
Us2hSDVHzxdFajQUqdFUq/qpNG9+t5Qvszi/k1pS5MD9f5a9an4sz7/wsbMEAVReIVCkmkGRag5F  
qhkuQeZQpJqj54siNRqK1GiqVf1EjrirfilZY02ULrVccrKXSk7JZXZkL5fddh0tu9X4Sl588SNn  
CQKovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnBrF+/Qpqd8L6Uzfld9ql9v5x//tVy1VXXSMeO  
V2vRWY499iGpUf1DqVyxj7xINzUXKq8Q6u9FkWrCiuUekdqEljWKJodRpJqQT6QeTJEaRatmHpG6  
hCl1Cj1fFKnRUKSGs3btIqlU/iupvddXcs01eNpUl6Sb+ePm/m3aXC2lsxfL3d3eVEtssxckFKmB  
bFciVTW1yeU2Es4K1fR8tVQNeSQBqNw0U0U3P2QbVSDCbVNPz1Vg11lgIKjetVNNztkw1Eo4uuChS  
o6FljaZKxT6yX903IEjtIFdf7YrULpZA7dKlk5x//iVSpuRCee7Zd50ICKDyCoEi1QxfkUpCoUg1  
w1ekklAoUs3RBRdFajQUqeGsWPG3EqBjJafkUNltt2/kpOYd5MILW8lI7c6Qs88+Vxoc/j8pV+ZH  
yUqsl6uufNFZigAqrxAoUs2gSDWHItUMilRzKFLN0QUXRWo0FKnhDB/xrZTJ3iSlS0dITvY4KVVC  
JfTJCjdKltisBOyn6vX1cvRRlzhLEUDI5cP27dtl+ujpst+G/XI7drSJf0x05iBe/vrrLxkyaYiW  
rYQct/I42bpqqzMH8Tjt9DQ5Yu0RWsYSMuqPUc5U4mXmzJkyZulYLVsJOXTtobJl2RZnDuJlypgp  
OnRVUy1jCRk0cAZaXiZO3eu/Pnnn0mCq06dOtZYaOLPpEmTpHXr1kk5GzBggDOvgBkzxyqBOT+J  
L6VMYfV5t5/CY1GzvpPSPQZLycQqaX0WnVQdilQPhx56qFStWtXe2Caq0Jt6rVKlStK/f39n7swG  
Yn7p0qWSk5MjWVlZkqik5Qqtf0lqZlW0cubOT0SOO+44qVatml1jv+dmy245CalSpYq89957ztyZ  
DWpm48aNUrJkScnOzpZESS1XaEMTUj5RXsqWLZs7PxHrzhVq1e3a+yr3GzZrUrC6uOee45PtgFu  
zaC+SpUqZefME+XKlBP6OcAas7n44ovzaswn0Mc98MADztzKpKavSJYSoTIZcyUn+2tLnFrP8M/a



lqWyX1F/h6u8LZLhw4Y6SwRxBGitmGRu+92XkiBhg1F/SYid9zhvLCTQJHqsG7dOuuIOWID8xGp  
blx77bXWcpncYT3++ONWLkqUKGHnRe38ktoAFU6+0MHjiDtTQZ0gDjnkkNycWOEVqWVUONOUvPLK  
3GUzIRdffNHKRW6NKRgf1Iap0KaPGpW5TrRbJOceeaSdKze8lrWqCmcahAbI5Brr1auXlYvcGgsI  
d/qvv/7qLJI5uHVy/PHH58tPUJxzzjnWMplcYx9//LGVl8qVm8sulZ9QovRDJUrnWUK1VFYvKVNq  
INTY9VK1n9zNmi/KCHvzTZGnnhL55RfnhRTYd18MYxGIXZwXQsB8iKITnRf+QyhSHVAY+TqpEJGK  
2Be/eoYCAeXNR5hIdWPevHnOGjIPfP98NRYiUjEvHK9M5Q51yJ+UK0SASNUjkw+G8P3z1ViISMW8  
cKkzlcceeyw5VynG0KHRbldxBd8/X42FhDtpvLWW2/lyOfJ7FZKqN6uBOqj6u/lklXCy16o+Prr  
r63lp03DcDqRbdtE/vlHZNUqkT59bKE6bJg1iwWm//svhkWJ4Fbly5eL/P23yIIF9nRXpF53ncjG  
jSKzZ4vMmmXP67J5s6h9tD0f4scfRVaudCb+R2S8SHWP7rwFYkWESD3qqKOsZTORW265JV8+UhGp  
a9ascdaQeXhzYUWISEXUrVvXWTrz6NmzZ1IurEhBpGb6gVC+CBGpCJyyzRg8Zt5rT7yWlItUY9LQ  
zD4QMo2yCXs4ThLF1Vj1fK/Pen3mm5OoGPzNYGt59Z9WXHqp/feee0T23NP+7+uvt2axOOssUQLY  
fr1iRZG99xbJyhI55BB7uitS69QR2WUX+78R2dkizz5rzzN5sv0a1uOu64kn7Gn/FeojEBSE75Fh  
hEhleCJCpJocfRfH8P3+ESKV4Yklkcoa8/n+ESI1oylNrYRq+r9z2yAVfu+5M0d3FXrbrv23afNb  
toQKv/fdmcOnBdZMQHPnh+JVq7SiRg2YQyKDIXbday/7tRtusDVM5872v888E8ab7bzC38BrBx1k  
z+OK1OOOy3NXMaYVr3IPpuA1xMKF9r//y5Ea6mOQ0qVLqx9EIYQ3KFLNigUnleEJilSzMfJZXiC  
IjUv4myDVfi9584cT6ils+2iwu99d+ZIY9NFKk7du9Subb/milS4ofh3v372v0H79vZrXpGK0/Ou  
n39uv4bQcV+D2P2v8Xy0zAT3dPMtNopUs6CTGhp0UtMQdFJDg05qRKSh0UktYKOTmnLzc1lxFtXF  
K1Ld0/8vvWT/G85nmzb2a16Rql841bev/RpCx31txgznhf8Qz0fLXFbc+Tr4CJF6wAEHOEtnHri7  
gTcfqTipa9euddaQeeD756uxiAunauAcT4Zy//33J+cKwTGpoeD756uxiAuncCYpY/CctnzhoReS  
c5ViTPwtc++Zje9fluGpsZBw583Hf3gKOVY83+vJV+wr+01j0Bf2/YzVf1qhX8DkFamu2CxVyj6F  
f/nl9il8vFYYkdqhg8jo0c6L/xGej5a54Gb0+QolRKTivnkgk2+rgSex6DmJEqm9e/d2lsw8UCeL  
Fi1KyocVISLVveo6k2usUaNGyfmKEKlPP/20s2TmgTrBrfT0fFgR4aS6y2Yqp556all+ouLee+91  
lsw83Drxy0tYgEyusbZt2/rmJSiu087Jq+NIK3Qn9fDDRapUEenWzf43ruwfpDQtruWuWdO+EAp3  
MFSryhWpRx5pXzB1++32vwFullBbTFeu7LzggKv669Wz76v62WfOi/8RFKkaGzZssIQBisQ6+gsQ  
qZncSXnp169fbl68lrXet3IH24sXL3aWyGzw5BrcrB85sWosQKRer1+2meEMHDgwt47yidThKpxp  
8+fPd5blLZs2SJ77bWXIROrxjwitUQ1e7ts166dswQZNmxYbh2FBR/3abNNqaJ6SsUgJ/mce+21  
Vq1aOUuQ8ePH58uTX0zDPacMOfdc+2KqF52HVU2ZYI/Fr1anfgP7tZ0ViLQN90gPG2CHNh2k6vyq  
uR07WpfoXazpJD8PPvigXHHjFVq2ErLbmN1k+vjp1vRMPorW0fPQvk172X367lrGEtK+S3tnKvHy  
5JNPylXXXqVlKyFVp1WVCb9PsKazxmzOPFx+3uVSe0JtLWMJufDaC52pxMsrr7wiXbp0SRINFStW

IMG4pFrBGrPR84CDHQx903Pm3sCf5AdPE7zmmmuS8oUn5n3//ffW9ILUGEY4NW5si1l39tjDPu2/  
fr0z004KRWoI9VVTJZTbSDgrVNPz1VI1Ek4T1fScbVCNBLNJNT1fjVUj4bRSTc/ZMtVIOlQAwLAm  
Ek779u2TckbHORo9XzVxjp74QuUVAkWqGRSp5lCkmkGRag5Fqjm6gKBljYYi1Rw9XxSpwVB5hUCR  
agZFqjkUqWZQpJpDkWqOLiAoUqOhSDVHzxdFajBUXiHU36xE6iZVRE6QcCyRquWr5RaK1CiabFYi  
VcsZRWo4lkjV8tV4M0VqFK22KJGq5YwiNRpdQFckRkORao6eL4rUYKi8Qqi/vxKpWiGRcFasUCJV  
y1fLkylSo2hytBKpWs5whwkSDO6OoOer8REUqVG0Ok2JVC1ny5ZSpEah54siNRqKVHP0fFGkBkPI  
FUL9+hSpJuQTqS0pUqNo0oQj1YR8lhWXtJJQcBsgPWfLLlGkRqHniyl1GopUc/R8UaQGQ+UVAkWq  
GRSp5lCkmkGRag5Fqjl6vihSo6FINUfPF0VqMFRelVckmkGRag5FqhkUqeZQpJqj54siNRqKVHP0  
fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5er4oUqOhSDVHzxdFajBUXiFQpJpBkWoORaoZFKnmUKSa  
o+eLljUailRz9HxRpAZD5RUCRaoZFKnmUKSaQZFqDkWqOXq+KFKjoUg1R88XRWowVF4hUKSaQZFq  
DkWqGRSp5lCkmqPniyl1GopUc/R8UaQGQ+UVAkWqGRSp5lCkmkGRag5Fqjl6vihSo6FINUfPF0Vq  
MFRelVckmkGRag5FqhkUqeZQpJqj54siNRqKVHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5er4o  
UqOhSDVHzxdFajBUXiFQpJpBkWoORaoZFKnmUKSao+eLljUailRz9HxRpAZD5RUCRaoZFKnmUKSa  
QZFqDkWqOXq+KFKjoUg1R88XRWowVF4hUKSaQZFqDkWqGRSp5lCkmqPniyl1GopUc/R8UaQGQ+UV  
AkWqGRSp5lCkmkGRag5Fqjl6vihSo6FINUfPF0VqMFRelVckmkGRag5FqhkUqeZQpJqj54siNRqK  
VHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5er4oUqOhSDVHzxdFajBUXiFQpJpBkWoORaoZFKnm  
UKSao+eLljUailRz9HxRpAZD5RUCRaoZFKnmUKSaQZFqDkWqOXq+KFKjoUg1R88XRWowVF4hUKSa  
QZFqDkWqGRSp5lCkmqPniyl1GopUc/R8UaQGQ+UVAkWqGRSp5lCkmkGRag5Fqjl6vihSo6FINUfP  
F0VqMFRelVckmkGRag5FqhkUqeZQpJqj54siNRqKVHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5  
er4oUqOhSDVHzxdFajBUXiFQpJpBkWoORaoZFKnmUKSao+eLljUailRz9HxRpAZD5RUCRaoZFKnm  
UKSaQZFqDkWqOXq+KFKjoUg1R88XRWowVF4hUKSaQZFqDkWqGRSp5lCkmqPniyl1GopUc/R8UaQG  
Q+UVAkWqGRSp5lCkmkGRag5Fqjl6vihSo6FINUfPF0VqMFRelVckmkGRag5FqhkUqeZQpJqj54si  
NRqKVHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5er4oUqOhSDVHzxdFajBUXiFQpJpBkWoORaoZ  
FKnmUKSao+eLljUailRz9HxRpAZD5RUCRaoZFKnmUKSaQZFqDkWqOXq+KFKjoUg1R88XRWowVF4h  
UKSaQZFqDkWqGRSp5lCkmqPniyl1GopUc/R8UaQGQ+UVAkWqGRSp5lCkmkGRag5Fqjl6vihSo6FI  
NUfPF0VqMFRelVckmkGRag5FqhkUqeZQpJqj54siNRqKVHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoO  
Rao5er4oUqOhSDVHzxdFajBUXiFQpJpBkWoORaoZFKnmUKSao+eLljUailRz9HxRpAZD5RUCRaoZ  
FKnmUKSaQZFqDkWqOXq+KFKjoUg1R88XRWowVF4hUKSaQZFqDkWqGRSp5lCkmqPniyl1GopUc/R8  
UaQGQ+UVAkWqGRSp5lCkmkGRag5Fqjl6vihSo6FINUfPF0VqMFRelVckmkGRag5FqhkUqeZQpJqj

54siNRqKVHP0fFGkBkPIFQJFqhkUqeZQpJpBkWoORao5er4oUqOhSDVHzxdFajBUXiHUf12J1J6q  
iBD3MIVRUKSa0+RdJVK1GtuwhSI1jE1blUhVeXJz1rg3RWOuRT5RIIWrsWXrKVkj0PxsitRo2n+t  
RGoPp8buUyJ1JUvqFHo/VvMFitQgqLxCqK+aKqHcRsKhSDWniWpufaFtUI0Es0k1PV+NVSPhtFJN  
z9ky1Ug4ej9GkRpNe9Xc+kKboxoJR89XTdWIP1ReIVCkmkGRag5FqhKqUqeZQpJqj92MUqdFQpJqj  
54siNRgqrxAOUy23jLbbqddq+fbv1l+Th5gR/9c69devW1uskmKaqORXGGgtBzwny5LbjVSPhnKta  
bsZU7raoxhrLT1KNaf0Yrk0g4XRSzakwq8aWq8Yay09QP7aXasQfilQPixYtkpEjR9od1IS8liqh  
Gl7buHGjFcQGG93atWtlzz33tPJTooSdJzdeeOEF+ffff525CViyZlIMnjzZztGvnd2huTWGi6fW  
r19vzbt12zZp/69Vv170+bN1mtgzZo1MuTHgbJ561bnFfVbqNiq5t+8dYs173pVpxtU6PPsrKxa  
tcpytJAfK09u+zkh3W/vLgsWLHDmJGDp0qXWxStWjX2em63cGkN9YbsleeBM0HHHHWfXmKcfu+mm  
m2ThwoXOnAQsX77cyomVozfs+kKzaiwnYfVRq1evduYmABctnnHGGXaNIU9um5OQy1tdLosXL3bm  
JC4UqRrPP/98UseUmGiVT15zXi9Tpow1fyYfKbrffddddd03OWUBcfPHF1vyZzrvvpucm99V6K2M  
CvV6haq7yGOXd5Tf9q8v8/beRxbUqi0j6h4knx96hPQ7/Cj5ruY+8luZSvLlwYflt/UOIQEq+h90  
mPx0QH2ZVOdAmVXnAJTe1+Zvk9d+X3/evJqoxPkiTu6ycBhw5xPsuPj1hiufM3NI9r5JbVhKpxp  
Z511ljv/pvPVV1/I5QvxlQq9VVOhXi9ZsqQl/tmPidSpUyc5ZwHRtGITa/5M57ffkvOTW8Vequt  
Qr2elZUI8+bNc5bKTNwacw+yc0Nv81Q4rzdS2NCaP5O3Sx2KVlerr746uYAQASLVPcr+8ssvnaUz  
j1GjRiXlIpU46aSTZN26dc4aMo9bb701f178RGqpknJB1d1kUnYFmZ5TRSaVriljcqrLyJxq8tiB  
R8hLDRtJr4PryQP7HiSvqL+PNGoi9594gtzWooVcdNjh0u2UFvK5EqfrSpWWtdmIRKkRkRz1V+0w  
hmSVlp8GfOd8oh2bP/74Q7Kzs5NrLESklo455hhLeGUq9957b1I+rPCK1Koq1OtuXt955x1n6cxj  
6tSpUr58+eR8RQRO/8OpzlQeeeQRKw9J22WASHXnee2115ylM4/Zs2f7mzl600QqAgdNPANpQ5Hq  
oBdlbgSIVDcaNGjgLJ15XN/1+nz5SCUW/Ju5p2X98uEnUg8tWU5659SSNTnl5fecqvJZp4ystQu  
0vOAw6TT9V2IS5cu1kHVXeeclFNlxaBs4+/uZFjCdPbGjaW8zt1lNZnnydv7HOAbFXzTUmUkLEf  
f+J8oh2b+++P3+++lkQqYtq0ac4aMg9vLqwIEKluVKtWzVk683ju2eeScpFqjBwx0lID5uGXjyCR  
6kbZsmWdpTOP999/PykXuaE3j0hFfp/9984aMhuKVIX3huq5ESFSy5UrZzk3Rx11VPGOY1RoraFq  
Nf6ulYnRKg8mMSIhh20+TFuT0/qrqK/C772LSRx99NH56scKH5H6eKmqMlaJ0kVKpG4oXUa2Q3hm  
l5Szzz1fuihx2rlzZyvuatVK5qlpW3xCskvJnccea80PUXt+p07yR+kKsrBUaTm/7v5ypM9n3JEC  
p7z22GOP/PIKQaTWq1fPd53FPdAXeXNhRYRlxWn/jOjH6qnrwtLOW7SWJUSoPfv1VUIxMyEFrD9LW  
4rTxKg5S4ffexSQcayCpOK0f6NGjXzXWdxjn332ScpFbujNR6TWqFGDp/wVFkKkiE1vgVgRIVlz  
JkqqiLO9q8LvFTMhfETq7zkVRUqXkq2O4MTf9aqTP6NdO7laCc4kkZqTk0+gliBq71AitbMz/5XX  
d5UPd6khK3LKyNFZpfw/y84QKYhUhiciRGpGRZxtsAq/98yEiBCpDJ/Qm49IRVCKuqRa4HSXX4FQ

pDqRrSLORpGa15RI7VpqV3m+ZC2Zn1NjNinBudkSndly/pmt8zupKYhUOKmtL28v/+aUIYWly8hR  
JbL9P8vOEBSp5kGRmhdxNorUvEaRGh16o0gNJONFqlsEfgUSJVL3228/GTNmjHWIY7GO31VobZhq  
bZ5pl4njVR5OMIjDE/LF0i+ONTItkYpfVfi9dzEJ90KzfOEjUvcPU16errSvvFRqT/k5p7pMzdlF  
hivh+vGedeSSrtdJ52uukauuu05uOau1LFdi1LowSolYyS6h/tu+QEqys+TVfQ+UC5q1IMcOPVKG  
VttDtpUpI5OVQH3vgYfUb/q77+fcUWLikCFyxRVX5M9XCiL1gw8+8F1ncY+xY8fmy4UVESK1cuXK  
Mn78eN91FqtAH6O1ward8OkNkmis8uDXxwXfKQl5bfJr2pqctkrFLyr83ruYBOpEr53ciBCppdWB  
84QJE3zXWZzd9XP3nPPPUm5yA29+YjUn3/+2dlmmQ6dVAeMI/EWSZBlda9YxJW0mQqEgJ6LVAJj  
c3AvwkwFt6/JlxcfkZpQlVPESpVxXqKUjMqqLANK1ZBxpXaRMdmV5c2qe0nvfQ+QL3fdXb6ovKu8  
X3d/ebRhI7nzbPlxtPPkHMOO0Jua3mGvL3fgSKlSsrI0hVsAZuVLfMTWdK32m4yYshQ5xPt2PTr  
18/KUVKNRYhU3K83k++Z2qJFi6R8WBEgUt284iK8TGXgwIHWmNykfEVE9erVZdasWc4aMg88pCVf  
XgJEqltj7du3d5bOPIYNG2ZdOJaUL4TePCIVB45//vmns4bMhiJVI98tgkKcVPdm65mOnpOwwIB7  
ItKjR4/k3PiJVgfaF1/0k56XtJenDj5MHt1nf7nvtFby7AMPyovde0jPZqfJGzX3ISfv/p+88uBD  
VjzT7R7p3vkaeaxNW3n1pNPkiaOOIscaNpaHz7tlnnn8cZkwcaLzKXYukvIVIIIPFAJcyKvvvpq  
cs5CnNRMFls6SfkKCRwEEZFPPvkkOTchTircV4ITXp6DIb1plhUCleRBkerhl19+kbffftsumPFW  
+VjNfVIL7PtMvkeeFwyXwGmJSpUqWfnxOqu33HKL9O/f35mbgMGDB0ufPn3sHP1i1xeaW2OYHnWP  
PDyF6q8/J8k26zITxRd3OA4cr9ynmiFPbvs5Ive1vSq3xjiGy2b48OF5N/Xvm5ut3BpDP4YnUpG8  
mvnhhx/kolMOsmvM04+df/75rDEPGF7y3Xff2TI6za4vNKvGshMyaNAgMT59ujN3ZuPWDPLI3ukl  
qR+bk5AWDVvlgAEDrPIYy3lQplZwuGq5RcTnqgei50Tv2M8++2znVRJEM9WcCmONhZBUY9ozr5uo  
RsJpo1puxlhjgQT1Y3x2fzSdVXMqzKqx1aqxxvIT1/x2f3BUKSGUF+1vDJiqqLAeFO9c2/ZsqUz  
hQQBkeXWF9oG1Ugwm1TT89VYNRJOK9X0nC1TjYsJ92N4nCUJp71qbn2hzVGNhKPNq6Zqx8B8qrxAo  
Us2gSDXnPxWp6jeywuS0b6rL/PGHyJlniqT5efoUqeZQpJqj92MUqdFQpJqj54siNRgqrxAoUs2g  
SDVnp3NS1e9qxdy5zgsB/P573rxphCLVHIpUc/R+jCl1GopUc/R8WSKVoyN8oflKof4RSqSWViWE  
SPPOtjhCkWPoK+ZKpGo1tmFDDCL1gw9ssYhn/E+ejEesieBKf7yG+OcfZ0bFt9+KHHSQSE3VaY4a  
JXLkkSL16ol8/bU93V1m9myRjh1FdtIFZP/9Rd59154Ovvgibz4ErogeM8aZWDg2bVliVa3TzVnj  
EyhSo2jVRolUrcawLaNljCLtwxAUqdG076hEKu684dTYnLkUqVFY9eVslzXr0EkNQu1BSBAYMK93  
ViQcilRzmjRRlIXLWSwiFQJSrduKUqXsv6NH570GkbpliWju+4t/16qF+xiJHH103jy9e9vrcv9d  
u7blo4+K3HGHLXrxWvfu9jxwWW+9NW9eCNw03REjV6Q60bgxRWoUrVopkarljCl1Gj1fFKnR4D6o  
es5454ho9HzVhCIAff7EBIERaoZFKnmFLlIfecd50WF+xpE6qJfImXL2v/+6y97OsRMiRLJy7nL  
zJhh/xscdZT92gUXOC8ofvopb940QpFqDkWqOXq+KFKjoUg1R88XRWow6d2DFDMoUs2gSDWnSEUq

BKcOXkNAPm6fn/dvHTxyFa95Rao+JtUVqW3bOi8oKFJ3GChSzdHzRZEaDUWqOXq+KFKDSe8epJhB  
kWoGRao5O4xIXb5cpG5d+9/33WdP79Urbx6K1J0WiIRz9HxRpEZDkWqOni+K1GDSuwcpZlCkmkGR  
ak6RiFSMJ1XrtkLHfe3vv+1/f/KJyK675r3esGHef3tFqr4TOvxw+7XzznNeUOAWVNWq2a9nZYmM  
GOFMKBwUqeZQpJqj54siNRqKVHP0fFGkBqP2lCQlilQzKFLNKRKRinVizOnixc4LDgsX2rFtm8jW  
rbbwnDpVZOVKezwqcMekvvee/W93GczvAhcW6161ynlBgSerrF0rsmSJfdGUPn8hoEg1hyLVHD1f  
FKnRUKSao+eLljUYKq8QKFLNoEg1p0hEaiqsWSNyzDG2lD3jDJHbbsMGYP97r73yLqb6j6FINYci  
1Rw9XxSp0VCkmqPniyl1GCqvEChSzaBINWeHEalg82aRRx4RadBApE4dkebNRV54wZm4Y0CRag5F  
qjl6vihSo6FINUfPF0VqMFRelVCKmkGRas4QJVJ3AihSzaFINUfPF0VqNBSp5uj5okgNhsorBlpU  
MyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo  
+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1  
R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHlIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLV  
HlIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rU  
YKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxR  
pEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1  
gyLVHlIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHlIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsor  
BlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQU  
qebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLN  
oUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHlIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1  
gyLVHlIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzTnhhGOTcrZ162ZnCvGDItUcilRz  
9HxRpEZDkWqOni+K1GCovEKgSDWDlJyU1+vT5VNq06SWHH/qmVCz7iuSUfF1ysl9UObtHmjV9QR55  
9ANZvXq5MzfRoUg1hyLVHD1fFKnRUKSao+eLljUYKq8QKFLNoEiNZvTo4flac7sAAOXqSURBVFK9  
yhlpVUKklCqpEol1UqbUJMIO/KtCJCcLr62QF18c6yxBdChSzaFINUfPF0VqNBSp5uj5okgNhsor  
BlpUMyhSo3nxxSFSttQyFZulfr0e0rHjedK1awe55pqL5ZSTL5VypRdlSSVgL2r7m2zcyPGpXihS  
zaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSw8F40xu6TpRSWZukQtkl0q7dOdKlSxfp3LmzFV27  
dpQ6+7wsJVWp1T/4V1m1ar2zJHGhSDWHItUcPV8UqdFQpJqj54siNRgqrxAoUs2gSA1nw4a1cunF  
Y5RlFSVSJ8klI5whF190oVx0UVsrLrv0fKI3yA2Sk7VAdqs+UJYvX+csSVwoUs2hSDVHzxdFajQU  
qebo+aJIDYbKKwSKVDMoUsPZtGm9dGg/UQnUGbJL5dGWY4pxqHpgTGqN6l/K3jV/V/mkk+qFltUc  
ilRz9HxRpEZDkWqOni+K1GDUpEEQZFqBkVqONu2bZbrrv1LmjftJgce8LklSEuX3O6JrdKmzTVy  
1BG9ZNUqOqleKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFKj2C7d7vpTjm54gdSp/Z6PSBXJ

yd4szZufKyc1f1bWrKFI9UKRag5Fqjl6vihSo6FINUfPF0VqMFRelVCkmkGRGs3DD09WlvRaqbPP  
V75Oak62yFlnXSutW70v69bx6n4vFKnmUKSao+eLIjUailRz9HxRpAZD5RUCRaoZFKnR9Ow5VU46  
6QapWmWI5Zx6RSpeO/GEe+Xcc/rl+vUUqV4oUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUqN5  
7LFJcnTDDpKVWVKME6SjyZ6tAn/tKJIYKPvV/VDaXvCREqk83e+FItUciLRz9HxRpEZDkWqOni+K  
1GCovEKgSDWDljWcl7/8XeWlv+xS5UNp1uxyueGGq+SaazqquNKKa6/tKJ07XywVyn8u5cp8Jqee  
co+zJHGhSDWHItUcPV8UqdFQpJqj54siNRgqrxAoUs2gSl1iu9x37ygpKripXOil0qXL1bk38nej  
S5fOsn/dl6TZCcNVPlc6yxGXfCL1ElrUKJqfd15SzpZQpEai54siNZr253tE6iyK1Cj0fNWsQJEa  
BJVXCBSpZlCkRvPol39lqexv5NJLL5HrroMo7ZlBv1+t/n31ZbLP3o/K6S1GyPr1vE+ql02qqrK  
bY1VU9qfePn8c5GXXpLtf1+MvWFyPP64yAsviEYyYm+7nQn0ovdjFKkRqPJpr5q7TaLNUY2EoHKm  
56umauzH/FE9FgmCltUMitRwtmzZKNd0mSDZJb6XnOxVctBBD8rpp10u559/obRufZEcc/RNUq7s  
YClVYrQcuN+Psmolx6R68RWpJI+//rKFaNmylhUrilx7rbRQQIXfLpfi9UqVRLKy7Hm3bXMWJi56  
vihSo6FINUfPlyVSis9UXiFQpJpBkRrOunWrpc05o5UIHSals5dLTsk/rEeklsra7vwV9fqPSsD+  
JJXK/STLIIGkeqFIDeHYy0WGDRM55pgkdzTfmNTVq+0Ja9falnX8eJFu3ezXiIWeL4rUaChSzdHz  
RZEaDJWXnatY3///feIVq1aSZ3Vu+++60wlXr766it55ZVXkvJ1yCGHyPLly63pem4zlc2bN8hV  
V46Rkon1Sox+r2KclC65LffWUzlZSryW+lv9/VL2rf2TrFypRESKFCS9O9tP8v3338tb772V27Gj  
7bNwH1k4faE1PWNrbOtWkRtvFNVBiVx0UdJZw969e0uDBg2StsvnnnvOmepwxhn2EIBHHRh/ncHb  
6q+//irvvfdeUr523313mTt3rjWd/ZiNnodeL/eSjJOaaFtlQh750Kklko9h6kDyw14fatlKSJW1  
VWT6yOnOHESHItVDtWrVkjooov+jVq5czN5kyZYpvjrxBbD768GepWGaR5GR/qcToLCVUF9oiNXur  
IFLCNSe7t2Qllss1nX+RLVs2OUv507+/bYSZpFfpvJSX2bxZZONG++9/yT///GPVkB5KvQ2TIUz  
LWPBd0eoPLI4half3H333c7cijffTL0wiiHes0BBQfJo3rx5Xm56q9BbbRXq9auvvtqZm2xUnWlu  
vhB6m6fCeZ0kw4wocFS4ZMkSqVChQnIRhUSjRo1yl81ULrzwQisXJUqUyJcfb2CegQMHOktmHqiT  
DRs2SNWqVSQrq5rU3fdcJUjfUkL1FyVQ8TjUWVKyxFeY5x5XSsWK+1s5g9Bwl/UDJvWgQSK//ea8  
kAl//JC6Fmna1J6vSRPnhf+Ajh075taPVUshlhXRt29fZ8kM4eyzRfbYQ2TlylW1e9A+0cVT3te  
Q7insq1lZ8ywf/AHHRBeyxRuuukmKxep9GOIt99+21ky83D7lrjLSXkJEKmlunXrWstk8r7yvvvu  
s3KRvGN6c0SjQ/3ZZ591liQUqQ75CigiMG/t2rWdpTOPK664wjcvUTFv3jxnDZkHvr9eY1lZNaV  
1tkqrlZxiWRnJbtfmLdKlSrWshCXn30mMnasyNSposSYyJ9/itx2m8gdd1iz5AKt8eWXlJ/9ZJ8J  
/vZbO4BXpELkfVGF/XedMwQW+5JJkzAm256vXj2R0aOL/vqa29SX0/NhRYRIRUzCh88UdtlFVEfk  
/CO5xs5X8ZwK9Y+k6KCihQp3XkQumOe885x/FH8effRRKw+mMWTIEGcNmQe+v96PWREiUt15M5U3  
3ngjL0966E1zUt3AEDqi6s35m1Fs8zRgFUaWWRx+xOHWsvq6rKb25sUutAauu/4685yp+ef9O09b

k9O2q/B7z5058J20BgpSY3Xq1rGWPeQQWz/svTecMvvC7A8/tF9DuEC04t/QHfgLo0yfRxepgrh1  
5z3iCJHFi21he+21efMhzjxTZMMG9U38vmsMAVz3ISISEKmtJ0/2XWexCpWf7epAUX1h67/tV/BP  
lQOndjAtKKapcOertqsqKAXWsW3rZmv69i/6+r/vzh5aA88+/6z6unm5SCnU/MNHDtfWpDW/9yxG  
AdztLCICRCqiLO42ofBbZ3EOuMe4IkXPRW7ozUekfus6CxmO6rEyD1wRrMogqZVQTf93ym279t9u  
8xRbsYg0tMAcv63C7z135vhMhacVtMa6y21yuCNSy5cXuf9++zaY/frZryFAnz4iJUuKwHyF4wow  
rz6PLlLd22RiKKP7Gi70djntNPu1U06x/12qVCn1b/WJ/suLEKn5HJ5iHK+pUP+RlwutL3r0QecH  
DYgN5RJyfv+8+XOX3YzpJaTl7+q/1XzFLtLQArfj31T4vWcmRIRIZfiE3nxEKiKTh0i4qB4p82iq  
miqB+JpPse3UUUVJFnO1dFX7vuzPHFyrS1B6TbnKY11lv9wpYsWAAbmaw8J1UeG2umDcqj5P0JhU  
97Vx45wXFBcneO2kk+x/V6pUSf1bfaL/MIJwUjMI3IKh/kPPRm578lbnBw2IDWUS0raPvoTdSlki  
NSHnQHAh1+q/i1XE2Qar8HvPTAiKVPPQG0VqIKpHyjyaqKZKIL7mU2w7dZRQEWd7R4Xf++7M8bmK  
NLVH5K5ckdqhg1PECq9lfewxexjAbrvhamX7NYxL1ecpjEgtX768+rf6RP9lUKTmxpsq1H/o2Uhq  
mBYU0w7Q58xrWdvs6a0HqX9lq1D/XawizkaRmtcoUqNDbz4idQA6eKKyk2FsD3j2GlqiRCL1U4WY  
t04Ne7xgEsX1wMfzvTpd0sk3L1Gx/B/7vqmZCL6/aY3tkrOLtaw7JjVMpMI13XNP+9+42Am3v6xc  
OXkeE5HaqpX9GtbRvXvRF3a3bt3U+3vykoJInTVrlrOGDGCXKi11Y/tgO/v1tjFKI5Uof6RFFep  
OEOfO29WQh3ZukjXtp/XxvlHMcRTxk93f9rKg2mM/d0ZT5OB4PvzwqnU8d53Nzf0xgunAsncytGA  
pb548WlpV65cvklIiqOOOip32UzlvPPO882NX6Cj+gEKKUNBneBZ/BUrVvTNj1/giWfO0tKunX0r  
KPd+62D4cJGGDUWcu6Gp97DvaYqr/eF+4v7uy5blahOLUaPsZfBQlp0jj7RjunY/6VWwRLp2tW9F  
1b07HuvqTChC2rdvb+Uid6cYIVI//fRTZ8kMAVe04chk9epc/bXLLrsk5QTxwpc6e99/YADDrcW  
sZadM8culHvvtV7LFLqqlkcuUh3P/CbuKZuhuPu7XXfdNTkviSJ1n332sZbJ5H3l//73PysXSTWm  
N0ekutOfeuopZ0lCkeqhatWqeUUUEJncSXn5888/fXPkDZIHbl3mlyM9HscTgAxBWeKx7LVq2XoD  
N+I/9VRbd+BWmjsreNpPbm5CRGrGgu+OmD/feUHksMMOs3ISJrzuvPNOZ24FHCiXrMiA1m6dGIS  
boKC5NFUhb0iJ5Zz7xWp+6hQ0zp16uTMTdatW5dUS0lnc1JxVwCSB7c6DfdID39xw+aaNWsmFRWf  
NBXMF198IS+88EJSvnCj8GWw8hSZfBSto+fhrbfekv33t2/c78bLL7/sTC0YzzyTd8ofgdtotm+f  
7JDurGCM1mtvv5bXsatWe0FtWTBtgTU9l2sM3xn3DLvpJltotmuXdEYb92g8/PDDk2osn0vTurU9  
oLIHD/vfGbyt/vTTT1Y/r+drt912Uwd99rPo2Y/Z6HI4/fnX5bi/jtO2yoR0f6+7M5V4GTx4sLz7  
2rtathJSZU0VmTZsmjMH0aFIDQGnW/XOioTjfbRgy5YtnSkkiCZNmiTIDE+llsFsUk1IKrfhdnLE  
4eij7fEcjVVONBHRqlWrpBpbtnq1PWH9evtIBk+F8D4RIsPR8+U+lySE0141bl9um6MaCUfPV03V  
iD9UXiFQpJpBkWoORaoZFKkRTJliC0/cUBc3zO3aVVpcfHFSjS3FIXCw2HErCMYlgcwKCT1fFKnR  
UKSao+eLijUYKq8QKFLNoEg1hyLVDlrUFMEFZM89J9vbtrWFqB49e9rjQtxbOfAUdj70bZliNRqK

VHP0fFGkBkPIFQJFqhkuqeZQpJqxabsSqRNVribb0Xg2RWOuzdu0SaqxJc44cRKmni+K1Gjaz1ci  
1dkmsX3O2UyRGoXej9X8iy11CCqvEChSzaBINYci1YxNm5RI1fLV+BiK1CjyjUmlSI1EzxdFajTt  
L7dvFeeGe6EZCUBPV809KFKDoPIKgSLVDIpuCyhSzcgUnGREAmFitUcPV8UqdG49zN2gy11Gj1f  
uJMQ8YfKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1  
R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLV  
HIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rU  
YKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxR  
pEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1  
gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsor  
BlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQU  
qebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLN  
oUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1  
gyLVHIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6e  
L4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz  
9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLN  
oUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHIpUMyhSzaFINUfPF0VqNBSp5uj5okgN  
hsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdF  
ajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQz  
KFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxC  
oEg1gyLVHIpUMyhSzaFINUfPF0VqNBSp5uj5okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkOR  
ao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVHzxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUc  
ilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUcilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQz  
KFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRgqLxCoEg1gyLVHIpUMyhSzaFINUfPF0VqNBSp5uj5  
okgNhsorBlpUMyhSzaFINYMi1RyKVHP0fFGkRkORao6eL4rUYKi8QqBINYMi1RyKVDMoUs2hSDVH  
zxdFajQUqebo+aJIDYbKKwSKVDMoUs2hSDWDItUcilRz9HxRpEZDkWqOni+K1GCovEKgSDWDItUc  
ilQzKFLNoUg1R88XRWo0FKnm6PmiSA2GyisEilQzKFLNoUg1gyLVHIpUc/R8UaRGQ5Fqjp4vitRg  
qLxCqD9UIdQfVBEh+jNVUVckmkORasambUqkqm3R3S4bj6BljaLVOCVStX5s2WaK1Cj0bZliNRqK  
VHP0fFGkBkPIFUJ91VQJ5TYSdkWqORSpZmxSDdui2xqrRsJppZqes2WqkXD0bZliNRqKVHP0fFGk  
BkPIFQJFqhkuqeZQpJpBkWoORao5+jZkRoNRao5er4oUoOh8grhcNVUCdltu52q7du3W39JHnpO



9A3v7LPPdl4lQTRr1iwpZ4A1lp+kGIPbotuaqEbCaaNabsbYjwUS1l/h2gQSTufOnZNytnr1ataY  
D0E1ttdeezmvEi8UqR5+//13ee+99+ziGa/CaSVUw2vDhw/nhQca2OiQs0qVKIn5KVHCzpMbt912  
m/zwww/O3AQMgZM+vbtm5QnhJs71NjChQuduTMbt1MfNGiQ5TZyecK26LafE9L54s7y/fffW/Nx  
x2gzevRoGTBggF1bfXOzlduPoQb//vtvZ+7Mxq2Zn3/+2XJN9W3RjbZt27LGPEyYMEEGDhyYlCeE  
m7shQ4blzJkznbkzG7dmkK9GjRol5cmN0047LXdfyRrLgyJV4/bbb08qmsREFXrTpm3cuNFZKrPR  
cxIW2DCJyCOPPOKbH7/gwZBNUl5yVOhtmApn2kEHHeQskdm89tpryTn7SoXeqqpwpvG0rE1SvkKC  
p2VtPv30U9/8+AXELBEpVaqUb368UaVKFWcJAihSHY4//vj8BRMgUt0joAcffNBZOVp4+OOPk3KR  
Suy7776yucVKZw2Zx8knn+ybF79w83rHHXc4S2ceX3/9dVlurAgRqYhatWpltAt9xhInJOXDigCR  
6ua1a9euztKZB9zTkiVLJucrImrUqCGzZ8921pB5tGnTxjcvfuHWWMeOHZ2IM48RI0ZluXL8uUm  
LCBUJ02a5Kwhs8l4kera6n6FEuaklo444ghr2YzAc/bhxqtvzJePVGLp/KXOGjIPv3xExb4193WW  
zjweeuhih/DmJEKmlWbNmOWvIPLy5sCLESUVUr17dWTrzeOnpl5JyKWqMGz7OWUPm4ZePqCibVdZZ  
OvP46J2PfHMSFT8N+MIZQ2ZDJ1Wx6667+hZJIEjNysqSchUqFP+orMLTsm0sJYk1Kg8msToh5beX  
19bitA9VIFTh9947Y2Sp+FqFp/nmJCKy1qka82vIVPi9dzGKnJycfNtcKilVroXf+jlhvLmwIkKk  
wu2qWLG7/qKVWSr8LTsm0v7bndRUXZbWW0tThumIqHC7713xiit4lkVWquoml8+lmNtwlpWX5fV  
aqrwe++dNfD7e1rpLQWrsTJby2hrsdrvRqmUaFKmKQCs+QqRmTJRUEWd7V4Xf++7M8YWK0Jvfe2ZC  
pCBSGZ6IEKkZFXG2wSr83nNnjidUxNI2UeH3vtzxNjqQZpUKQqjj32WP9io0jNizhbxbV+77kz  
x+cq4mx+75kQJZFqHhSpeRFn+12F33vuzPGoihbBRV+77szR4ztANUyDYpUB99iixCpuH/e5s2b  
rauwi31obbVqV915IX0UjB1eqpGVkEkLJ2lrctpGFUtV+L3vzhj4LptUaG2japYj7ZeXkKh9RG3Z  
qpq+LqsVp3z5BC6wy3e3DUQKItW9TVymxZYtW/LlwoolkVq1aIXZtm2b7zqLVWCb0doK1Xq+1IMS  
FZ2cpBqqBn8c86O2JqdtUVHc+rENKrS2TbVEeZ+cRESZmmVku2r6uqy2XIXfe++s4akxtFc+ecU8  
Z2US8ulPn2prsdK1TINilSHTp06JXXcVgSIVPeKRdzrMIOBENBzkUo0bdpU1q1b56wh87jpppt8  
8+IXbl4/+OADZ+nMY9y4cda476QaixCpDRs2zOg7SHTr1i0pH1YEiFQ3r2+99ZazdOYxefJk4yuv  
DznkEFmyZlmzhsyje/fuVh5S6fvdeV566SVn6cwD94rFxyne3ITFPvvsI/Pnz3fWkNIQpGo888wz  
ycUSIFJxQQfI5Bvuut+9WrVqyTkLiAsvvNCaP9Pp3bu3b368gdvibN26ITWm2H333fNyEyJS+Rhe  
my+//DlvXwivSK2mQr2OAwAletaYn57752cs4A44YQTrPkzHTxcwy8/3oBI/eeff5ylMhO3xg44  
4ADfHHnDvWtQJm+XOhSphBBCCCGEEEEIIYQQkgI0UwkhbBBCCCGEEEEISQFksZUZUsXy6+/fCH/  
TP9AZP2Hsn3Nh/Ln2PdI2NCfZPMWZyaHCX9MkpdeflWeefZlGTkq70FxpHPJLCCGEEEEIIYQQQggp  
jiSZqUvnfy+/frqPTB7/iixdLbJo+ToZNehWGFJ5FVK9ZW/5a9Ce8nWv6vLes9Xkxceby3tv3yAb  
F3eS7nfVzk5ZWX48JHOMggghBBCCCGEEEEIIaR4kRg8dKycc96F0u+tGrJ2ztmyfNVyWbR8vSxb

uUw2bdkkQ8fOlF7935RVU+vJ6J9aSr8ve8nSFStk/eatsnDpUvI9hJZ9M/bMmf8qXLheQ3lgQef  
cFZNioxta0VWfiOfP3CbtGt8nTz36zwZv9mZVigWyLLZ/eXVVjfK8499Ld+tElNLgceEEEEIIYQQ  
QgghaWfevHkyadIk60m57t+wcOeZMmVKsX5K7kcffSRHH3209O/fH5fEO6+Gs2rVSrn55puthwCm  
OzeJ9ct/kBm/7y2D+pSTAR8eKT9/302mTfplfhjwmFzUpoa88HBVmT28rqyfvo+M+P50efKJ6+Xq  
G66TPv2+k2Url8uQwW/J7PFtZOGY3eXEoxNy7TWdnVXHQTqcvOh1FNmdCiLfZ3vAZ0nIA6bjS8yT  
ZVPekNsqsHS/XtX9Vei9VxbjNmRSK973tf+f/Ls7r1v8TQgghhBBCCCGEZCa4bSZM0aITpzqvpMaW  
LVtk/PjxMmfOHOeV4gmM5osvuVSOP+EE6f5wDytXXjZv3ixff/2NdOrcWRoceZS88uqrsKW9Im4S  
S+Z9LwM/qin9e1eRUd/UktXT6sqGGXVI7V/7qqgja+e2kLXz2sja6QfJ2mn7yLrpdWXTTrLoytN9e  
8v5zu8u/Y+rItr/3k5WT68hvn1aQP4be4qw6jFUqfpS3zm8vnXZrlc9++ax0uugkOTiRkBlqKp1y  
q1zYZ4ks275E/nr3KulXHNp8Np4eX+mvbTF4sGyoV9HaXD6XXLkLQPFmrTtVxn9ens5ITtLSqv1  
lNI/Pznm6Sfl/AYXyA21L5ZHR66SSWq27Uunyvd/7SkXHJSQHjRigPOLup3DpEJ85HkZbJhdV/p  
Ua+V3NCorTzT/2lpe8zBspcz7/EP/Cwv/jxbZrzQVM4/snuOqq3ul86fL5SVm5wHMDt6nsu+UDe  
vd7+bqWs+faTCIVvl6fHzhNvmW/fpjaAx4+SF246Tjr13y4jLeN8pYof5c2z20qHmpfIY6PXyES4  
j9uWqkp6VR5/7i1peccPsnzzz/LrgzdIh9Knyf1f/i2jZK1sXvmpPHXla7mlXgt58umOst+Be9qf  
tUQJ2bPdQ3LHD+tkm+tkblso8rda33IHSm3n++x56YVyyv2PyRUIj5U7O79um6mYd/ufsmLYfdJ+  
/xpS3ZkXsXfTy6XrQJE56uuvH/+e9O+ckJue/0geHy+y1XqfaTL9l2fkOrW+G2/9RD5WK9uAI9f9  
lrOHPCWnXNNXnv1U/5EJIYQQgghhBBCij+umYoweR7Rxo0bLTN17ty5zivFm/Xr18uIESPk0MMP  
twzTkSNHyj///CMvvfSy5JQpK7fedpvMnz9ftm1LaTRggUjMmj1H/nf/A/Jwz4ek/2dXybppe8r2  
uTVl0+z9Ze1fdWXtNMdUtf57H1k7/WBZM7elyOLjZOvsWvLTh1XliXv3kEvOryWlshNyRfuLnVWH  
AUtuoLxzwVnStmlzuejl8TJgoT1FZK2MfaGD9GxZW675dJ78vOQf2Tj9RbngwG7S9dpvZZY1z2KZ  
8M2bcm/TdvLOjzNkBgZHea/KbWe2lzZnPi4DlmltYLnIvy9Jj1NbyMI7XiI9RqySCdbrm0SWfi59  
n+kqTU84QU49rYEcXH0/qZ3TQm7oO0dGyQbZsuYzefKwRnLufpfJlZ/+LePXYzn1uVe+l/cdc5Qc  
v+fZ0vXndfKn9WCudWqVn8gZrS6UU6pfiU/+sVb+Vp9qye/3yiWVq8l+ZWrfie0IkantpQzG+0r  
Jx5WQxKtXpDLPvgbC+exXf3QC96VAW/cK3uf0kveGb5GNiyfKpO6nyyv9DxH2t57nxza+j15c9Bs  
Wbp8jPS/rLHcf/9z8uClbJehsA/10rHcq4Zuo62byyJzx6UEM55/DOctVPq2X2VrzJmPWW3nJX  
g1ZyZr3b5ZV/tqssDZdJ/e6TFvteJve8MFj+sD4MGCHTv7ldzi3XSC7q+l68r/K6bf3n0ueay6VZ  
idby8C/zTzk3y8x+PeSJphXk6ld/kw/+VD/o+Aflkisflxa3DBb8vEu/6SE/3n2s3P50V2nU7lk5  
55afZObWFTL5vf/JW22byw1fzZJvcuuAEIIIIYQQQgghxZVHH31UDj30UDnuuOPk+OOPTykwL5Z5  
6KGHnLUUH2immoEcvfraa7kD/A6uV1/++usvZ2q8JF589VkZOfZHeav3y/LoU0/J2x99LB+9d6v8

2PdMmTvuLJkx8iQZ+1MjGdq/gXrtQBnw8d7yy6e7y33dGkvzszrlZdfcJkc0biJ19t03d0gxvID4  
D2+bqW+fd6ZcumtLuem7ZTJqoz1FZKr8cN/Fct3BB8uNn8+SH9aol7asFRI0pbzw9O1yWNdv5d1r  
mskLD14trT9fKeOXYZm5su2fV+SmVlfJ+Wc/lz8ut+xNhfqPRa/Jl6edLi1qtpMeozbl30t/kTkf  
Xy/7l7leOvUYK4us+ZbJ1Dful/81OFm6958pQ9XSW9b0IUcPaSGdjr9LXlBf6x/L0MabDZAXmp0t  
Vx7QSZ6atFmmWQblahU/y7sXXyodalwojwxbl9O3TZHfVz4sbUq1lHadPpKvMZsPyFJyprbLguF9  
5ZOzj5CHn+kiV77yupx95kfyfzFBZsnDRWPm87aXy8ZMd5dG+j8kR574rT3w0SzYLPsQw+eZO3Uy1  
R6b22P8U6dz0Hnnpb5H51ndYKtu3fSPPHt9KOta7Xp6eul0WbRomk768X049oKPC//JQmYzZLEbJ  
jG+7yQXIGsklHd+R95albFvXVz6+pr00zWojj/w6XyY6c1pm6pdPyNMn7SG3vPWzfGj9LotlZlNX  
ysvtm8htnz8pZ3fqLbfe+rPMV59h9GvPygcXnyvfv3u3XHD3i3LKld/Ln8stx5oQQgghhBBCCCHf  
lGjPKHXStZ4dAXwXmqnR4B6xjz32uBx9TCO5tF07WbBgAZIn3377nRzW4Ahpe+GF8vHHH8uqVda1  
1bGQ+P6nb2T46B+k5ZnN5MRmx8igld/llOnD5buBn8tttd90uDY46TI5s3EB69flAfu98hfTdq4Zc  
3Oosuer+++WVR3vKj4celC8edajccu//ZM78+c5qo8AX+IF6X9JJ2pU6Uq6+sonUO8K5BF3FIRd1  
l3uGiiyrgHPY9FPj8mH7bOk6gUPyblvzXBeddi+VWX0Ffmm2/GSlZVlrafiAfvJMU89lec3OF+u  
r32JPDpijeDOExv+/kZ6n72rnFhBvV/Jipl48hKpd+QZcu3Jp8qDP8yWYZaZ+qn0PPBk6dj4Dnl+  
tsjfuWZqf3nuhFZyRd2O8sSfm2Vqrpn6k/Rue5FcXrWN9BiyXMZadb9CZEFv6XVtMzlQfZ4c5/sl  
yISSRMtnpPMnAfez2Kpe/+tpufmopnLcfh3lkQkbZZq1vkUi816WHi1Pk0bVz5P7flsqY61bPyBR  
Q+Sr26+W9iVPkXv7/S0jLTO1j3Tft7l0bHK3vKC2qTwz9Wt5uvEZcsWBXeSpKdtkOI5eq/7/+2vl  
iua1c3+H2pe2lZPv7SGXJY6R2656TXq5l/lvmSBLf7tb2u23m1Sz5s1WUVsOb/WwvKB+lvnWaf2H  
dQNkyAud5djE0dLu/m+k7xrYrorNv8qUPjflYaWOlvOvfkfew2hi6/MRQgghhBBCCCGEZBa6mWrC  
pk2biv09Uzds2CDDhg2TAw46WI5pfKx1af+6dfYwSi+LFy+27pVatnwF6Xr99dYl/1u3WuZd2kjc  
etutctTRh0nHK0+Tj3pfKrPHnS9jBp4oQ78/Xf6ddZ+sX/Q/2bToVlk27z75+PM75IYbrpTbnp8q  
97a/Qno++5g83+sNOFWss6TarjVk9OjRzmqjsM3Ut869UDrUvFQeH79FptkTNLZb/8vjD/mt10PS  
rtll8tqvM2SyPi1pPp2/Zfl7R8lZR7WQmse/LgNXrncu/y8lwQ+D8n97+9XAJ2ZM+Puk+kZm38GL  
PVeKb2WRf96gdZisIRBCCCGEEIIlaR4sWbNGuvJ86axdOIS616ixZVJkyZJr169ZOBM1J+zg4dR  
DRw4UD766KNA47WgJjy/8vTjneSi1gl58aFdZMRXu8nsYXvK2G93k5H995DxP+wpMwfvJn/9Wk1+  
/6yKvPBQFWl4qD16sUaN3eXVV1911ilpDkWGPtIKvul2n/Rofb+8P2WdzMhnEmOY4kr56+uH5bkz  
68qRdVvKBXd/Jj9scO+Hqr/PNvW+s2Xo05fLXU1ryiEHHyR1968j++9/ghxU/zF59vtZ1j07LejZ  
EUIIIIYQQQgghhBBCCkCumbp580bZuHGdbNlsx2YV+PeGDfZf/Hv7tvVWbNiwXtauXWs5uyb3cSCE  
EEIIIIYQQQgghhJCdlVwzIRBCCCGEEIIIIYQQQkgwNFMJIYQQQgghhBBCCCEkBWimEkIIIIYQQQggh

hBBCSArQTCWEEIIIIYQQQgghhJAUoJIKCCGEEIIIIYQQQgghKUAzIRBCCCGEEIIIIYQQQIKAZioh  
hBBCCCGEEIIIIYSkAM1UQgghhBBCCCGEEIIISQGaQYQQQgghhBBCCCGEEJICNFPJjsX27c5/FBFh  
71fUnyVvNm0SGT9e5O23RZ54QuTHH0Xmz3cmOqxcKfLllyJPPiny+OMin3wiMnmyM1ExZ469bPfu  
lj17ivToITJunDNR8csvg8+KPLII/bfl14SWbbMnrZhg/3eDzwg8uij9l+8V7rZuFHknXfs9T/2  
mP23Xz9nYgGJ+k23bhX56KPk9+zTR2TbNmeGNDBpkp1X5Pzhh+0cTpvmTNyBGTPG/ryoF9QNamvu  
XGei4rPPkvP2/vsimzc7EwsBfpOPP05eN/6N18mOA2oBNeH2KaiVsWOdiTsw2PawDeLzYpvEtqn3  
lYXFW7/lzxtviEyYILJ2rTOTDytWiLz8st3/4vPh708/OROJL4MHizz0kP0b4u9zz4ksWuRMNODf  
f0WeecZeh9vfDR/uTEwjCxaIPpts3mfG+wwd6kyMCezHsT/X9+/Y36cTXT9gu8J3HDJEZOISZwYf  
sl+FTtH7+Q8/FNmyxZmB5GP6dDtXbt+FWp040Zmo+OGH5P7jtddEVq1yJsYl+i7o0rfesrcj1AH0  
wB9/iKxb58wUwldf2b+/qy+hN9evdyamAV3/4rMhb9CweD1MI44alayBnnpK5J9/nlnFFPyWv/4q  
8sordq7wF/t1Hlek8xjJTz9Ac5qC/uKDD5L7kU8/Tf/xXBzHCql8xs8/T37P996zf4v/ksLmFseU  
2LZwDIH6wnaFfqNXL7sP8x7f4jeGRsP+AX3Et98W/+2Q7NDQTCU7HjNniuy5p6pOVZ5utG/vTIwB  
dMb6eyHSfXCRLiBEv/hCpFo1+3M2aSIya5Yz0QHTDzlEJDvbnfzXnyx/boLhG5OTvJ3xk7M5bbb  
kqfh95g61Z4GcXXSSSKISonsuqv9t3Nne1o6wfs0bj8OS64wJIYCCBA9HvmZeUdwMLAxXfTp7do  
kR5T0AVCT18/om9fZ+IOzIsVJn/mMmVEfv7ZngYx1apV8nTUZphZlCoQiqeckrxu/Pu/FpAkmd9/  
F6IVS6R8eZEqVUQqVrTF8I4OBLxeWwil9HThV7+IEiXsfc/y5f4HlzNmiOy1V/lyN93kTCS+wDDS  
84V9n36SMFVwEL/vvily4rssotd088/70xMI6NH29uK/plhihWWs89OXucBB+QZmTh5gO1Un377  
7fa0dIH16et3Azru77/9T4ThApn005Pnb9bMPtAm/sB0RD+i5wwnoF2uuy55GmoahmGc4ERR3brJ  
nwtAe30gTFH8zviM+udC6PoUeIWf1qBB3sn8dDBwYH79i4Buxn4sSFvA4NHnL1dO5LffnInFFNQL  
DE5o8ZII7e+N/vCii2yNni6gJUuXTs4vNKcp6C9at7aPS2rUsP+intJ98j2uYwWceNLXiYBZ6HLu  
ucnTGjcWWb3amfgfghO/e+yR/NkQqBM/YDrPmydyxRX2b+TOj+MKHFfit6tcOa8fOell+4Scblbj  
BI37nm3a2CdB022aE5ICqgIJ+Y/xdn7YQR94YF7nirj+emdiDMBE1N8LAgkjP9NJWAdv0vID1LiC  
AwYmzth5RYL+fS65xH/9+H4tW9qiCKLyxBNFvvvOmagIM1PxfrNn2yMs8Rr+es8cpoO4zFSvWMEO  
Wx+J9r//5ZkYOPi4//7kHBf2t8RZfhwk4ndErZ12msiwYc7EHRiMZtLz5jVTYWTUqWNPq11b5K67  
0nMgjIMv/CZ7722vG+9x33326+nCZBvcMsjK74UTPTBqsC1NmWKHe7C1l+cXAv3kk+16hnmGg379  
ADnss6fyvYLMVdfQx2E7WrPGWcChKMzU4Ib3MMGPPNI+kQgTFAdy3pONqYAD5b/+yqtl/E2nmeMS  
l5l6+eXJ6zzqQlyD0CVLRC67zM4PTiQiX71729NAVE2kUjNBZioCB78w07yjVlvCTC1u9T5ihMip  
p9r9FvovmDv6QID/wkxF/ervufvutl7UwQg+fR4YJvrnxi0/fazp9WsKXLjjcmGUdjvmMpvHGSm  
lvBZ8H7of73auijM1B25RgcNso8Z8N0PPzx8pLkpGLV8xhl5xyQnnCAyYIAz0RCcsNGPT+IYtRhk

phZWl6lv1tcJo1E/PsCoXVdnQx+gr03nqG1TsJ00by5SvbptgLqGuxswS/2A7jrssGQj9YYb7N/M  
3QZgTGPfi+NYTMexGkYvu2A6RrQefLA9HTUZxxUkhESgqo+QNAHRiwMPXD6DUWoQQ96z5gh0uOh8  
MUIQOzp0nNgZYMd57LF580Ho638R2Nnik3sLCFssDNDJ+5O1wPiEiMyLr3UvoRn4ULngyow+IUf  
kYrPidDfq0IF2xDCJS3YmXXtap8h9549RWCKB77z008nG3M6yIt3OVzCkAoYvYQDEBx44zOec07w  
srqZioMmPyAEsBPEpSL33GN/Zv2gJcxMxUEl8qpPx0gYLxDQ775rG6BeUwAjAPD7nXee/X1wMIDX  
lfbd9/EzU487zh7thnXiYFCfht8bO2f8TrjcU7+cDZfD4bIhd16/3xs76rvvtnfO339vH4h8/bX9  
ObwCFzXlLodATeuXvleBWke+YdLiM+F30EdCICf6uvE98Z3xeVwhizPtOCJCZaKobQgWnIDQv48b  
2A4x/+uvB4tKmOHYrnAQ7h7EuFG1qsjNN9s1p7+um6kAwvqbb+zPiUuRUCfpODDAOmAAYOQK1o3R  
OH6j+dAn4DLRq68WadQob/S2HvjN99nHPmjHpbWoE7fuiYKPPz55fhz4Q7yedVb+7R7rQr22a2cf  
nOEzecGl3Bhdcf75lvvnzda3A38XtiWMGILB5h6HwXwmUaOtA8uMSIB8/r1PxcFir4Jvx/6CdSU  
S//++eeHSYH3xG+ov451Y6QDPjP6MmzXISolz4PPfOih9u1D0H+7oD/S50NgG3fBCTHv9DvvFLnq  
qvzbMgK5R3+PS+IQ535gu0F+8Fvi4At9trs8fh+Yafge7r7l7Wc6dXJWoMDvjz4H68AJAfQV6B91  
0Cd4ty3sG1LBz0zFd0NN4KSD+9ott9jv7dY1fsNUzFT0JRgNhr4e/Z+eAzfwux50kN23YIqITqbp  
7+PN/zHH2Jd04q/+Ogl1U7++3SdgBKc++hyGMPaVOMDDNuj2V3qgr8RBJ0wXmInpBO+PesD+Cyd/  
MPLFPbjf50Td6p+IXj27vvBbuK/hYA19jD4fAvXhBzdU4+Qcvo+3v8Fv/slLdh7wbxgv+It53T7H  
z0zFgSP6atSzd/vEvNBIqFOYD/o+GzWKbcmd161XvW6x38X2gO0W/QR+Qxhy3isIkEfkxI0OgRMO  
qel1U/HdoY9g6rqvYfQYfit3W0vVTMU+DPuqa6+1+yo//YdtHdsWTIqiVxht6BoOMJvX8lifHzWJ  
esRn8uvnYV5gn48+Tu/n8dmheaElyWr6fRb0zfjNkBPoisKaLTp4f5zAgo6AnsB/631XKmYq5kc9  
YD+HmvXbb2IfAO2F/hpXNXINRuQX+xN3fr/aQ+i1jry64b4GlwZ9HPbLCGg53CIF+3/v5dN33JG3  
HAL71lQvC/czU1FLZ56Zt4/Gbw49oZ9EScVMxbaEE+Y4vkE9ow69eUAgz3hP1DH0EkBVufgdk2A+  
5Ga33fJPw0g+HAuh34AOdX8f5Az7TpxkQr+CEcPeZZF/7Bvw+8HIC7stCq7gcvcv6TZTsZ/Evgi6  
AjocNalv+zDavJ8b/SvmdWsW2y+uRMBn0+ff8aa3ZtGPoLaQF9ecdAN5wufAPhU16fbDyD36LIDt  
xmum4n1xawtoQm9fgHpD/4tBMDhx4J5oBjhWQr/szuu3beAzYL+G/SvqC9s2rqrBb5AOnW0Kfhv0  
GTjZge+K/Sb0i2tsuhFkpqlfwTGXPm/QSGQco7rzoE6xX3d/T9QN9DE0CaZD26HvIKQIUZVHSJqA  
GaB3jtgRoBN0DxCxY4cYhbCEOINoxUElLv3RgaHgNesgJFwx3zilcuzUsYNB5w3Tzr0XE8QD7pXm  
Hry4oZ/R0sEOWZ8Py7kHeVgXRLI7DQf3EBwQPtIzLI5siyndAMMOE0LKO5IF96vCDhUH9BBsEPCp  
ntX+6y/7wADrh7EJAy3o3lf6CELS1AtCmJmKgwIccqFP1y/IQM7efDP5YAzLd+lim3aoBZyVhMDx  
HmxjFINrRvuZqdhpw1TCKFrkHwcmMIsigIPQznKgNGAx+oA70dalWTUciY+QCIsWBEv4iH9jBpwPv  
qCIETiVsL8gXxB3eHycfYNBhOqbhYA+GI34fCB3kGrnB5fbuerDtQbC4lyJhXlwqo5/EgDDFJTOu

UQaBC1EJwe7Og8C2p5up/xXYBI991R5h5X42bGPuaGv3lBrbLMQfDm7cs+fYDmG+4oQJxBIMU3cd  
COSibVu733IFK2oX/ZFeu9iuYT65IHcYPese+OBgCmb3vffmGaY48MV2hgNcdz04eHTrHweGOHhy  
+zG3z7j1VrtPBfjMOBPvNT1xcITfGXWAPkbfNhAwRmCmQliibiHkYZrr86CmcfCI7RCBPtatNzdw  
UsPNC/plrxGBvsjFe/CL+sHBOz6H23cjZa/3FHlBqC2vQfT6KORP/e74b1xMIMTD/heGMmE/QDy  
6/3+EOQmwBSHCY/tEL87fk8YKKngZ6ZiPwJg5OGA2q1HHNTByAMwPbx50M1U7BNgAOFgBtPw2VAD  
6luRG/dgA/026hXbr3tQj34U+yj0EQjvwSS+6zXX2N8boJ/F/aq9J3qwHpxwAtjf6SeZYAbi98f3  
QF+N6TgQxcGO+znwvqjnuEfLAZgc3pMICOTNHdmHbQ37DfStXlNa/72RU2zv7kE8tk1sG+hfYETi  
++JAHeatd5to2jSvT/lzU5F75AimALYt7OehkbDN6vPBaEdf5If3ZB/6Rvc9UwG5cmvW7VdwQi5V  
vGYqTjaj/0Wfhpt314l+E9sotlXUK6bpy+lmKgw1mEW6JsDvhv0bfi9d/+EkMfLobvf4jZFT5Bs5  
9d4GAX0f+jLsL9x+BsyR+mO9DrA+d+Qw5kMNUNPQv2ObwjaDz4zfDX055nd1MU5aQJMV1ZUoYWYq  
8oA6xdVO7nRoV/wbJwnwHXDVfezLcCLB/c2QK/SfrqnkBTlw14fAptDdX7lAc+jz4PeB0WwCtjW3  
P8Zf9CX6/WLD8DNT8ZnwnZEzvd+F9oFhDvyOFVwNj3pA/ejbHvp11CE0p2uWYjvESWf0RfoxCvpL  
9/YyqHP3dQT6F8yPE2GumQ8zEOv1mlctWti/G8BIUveWHtCqONmCbQXT8TmwDtxiytXzyCO2Oe8x  
mYtupuLEHbRXUQEj3/2ObuDz4nd0PxO2cdSArmcR6Ffc/SH6EWgZvR9BjUJ3u1cxuP2Uq+/dwDbg  
jnzEtuM1U7F9w5DH8Rnyi5pArXXokHciF4F9DQxAP7D/0deJPifV48SCgs+J2kc/jNrwhrtPc8G+  
Ccfg2l8jj3/+ab8ObYs+RP/8QWYq+gToN117oFYxP04MusfWON7Tjz1w4hP9uLtfwGfHCQSc3MZ0  
7JNhsBNShKjKlyRNQfZb1MKOwu348N8wyCA0McoFZgVGKOH+Kjh7hE7aC87SeUfH6ffkhAiE2ldo  
xf0nsZOF0lclwPvAuMNODztafR0Q2H5gp6bP55qp2Oni8mJ99FCqAQFVkmSLg4DZ5x40lzcwHXSw  
M4O4xIE5djw4swrDDWcwC0JhzFQYTzB99OI+95vDztA7Wi3KTA26zB8jI3RhisCIIT+QJ30+XSdt  
CHjNVlGniG+IGheIDYh+vxF9UaEfzEMU6WYeols3/9Ez+khuBLbtHcFMRS68phO2W5hYfqCO3bPY  
bsDQqb/iHX2LPsU9SNfBAaa3xpE3FwhiXTzDUAL6eBqEIMxc9GcQmO5JEv1sPALbf9jDabwHejha  
QD+G/HjNRP3+yADv672PIkZK6iAPXjMKI0GQN2BqpuJEmmvY66Bf8l4cwY1JIJp13BMabsDMckW9  
DI7DfkGf19RMLQx+Zir+7dYn9ncwn/SDCpim2Ld5a0w3U/HgCb3Pw4E7DIEgcKcVhxyiPrEOmJze  
EUswkfzWu78sDhZxm8FY8h5lpRL4TEVhLvmZqRhth20K03SwP3DNIzd0MxUnGXQzBmYZRtp4gbHn  
NUGjzFSMMvQDJ0f0+FD+Qf3vhRcmz5vvue05G4TVTdf0AzYeTN/q+HQYETiJAv+jL6WYq+kjoKn06  
tJ9+kK+Dkw343vr86DOA96QADfk/LQqdhX5KnxcnxAAMGZwQ1qelGjj5VxSEmamoeeynUI/6PKkG  
RuD5gX2vPh/6MK/JiRNe+jzQ6u5JpKLaz0zFMYkL9BZOKLv7M+x70bei/9WX0c1UaCaceNWnw9Tz  
3uLABXXtHW2Jk4EwkdAv669DawU9fNC7rWO7Q1je+nYMf9JoVQCBpYfuPoHvy/2p+7JS2gnfDZ3  
O40Lr5kKUx958o6ORd/q1Q+6mYp9jfeKD/S5ro5xwe/preUoMxVGtt89U3H85p54cwMnHfzQB8Mg

oN1wTBUn6JuhX9FP6+/tBn5vdyAABgkg99Ar3gEryln36g8MogkCmg65wSjgo49O1o84FsTlbhwj  
6vsFnIzCsbX3GAUDeNx5go4TCYkJVXWEFBLshHBGC+YXDB+c7UQHiZtmY9QBBCzEKwSr15hEx4sD  
NB0/M1XvkGHaQrToZgXOVuHg0z2DjB2m96yiiZmKM5PYSXvNVN0c8AO5wA7Cu2P2/lsnbJoL8umK  
epgG3jP9GF2EUykQZxgRip0SDhyDBG8UhTVTvaMYyfR4wfeGYaPPV1AzFYK2MGZq0CgLU1L5LaPw  
mqkw/mDs60DQ6WYqDuZxABAGPpu3NmGmeg9QcYbXK1QgEPXbJCBSNVDPDcpKOfMGQ8I6YxHfwM0Hx  
fti2vaZPkJmKg2W/gwQYdGFmKoSdPnIV2yNMJz/w+XE5IbZf1DH6T4ARmvoBH7bBoBEK+K4QpO68  
CNdMRXjNVO/BPIyHKDMVB2jpNIPd/lon6GDla6Z6D0RxNYOfOYu+0vt7m5ipha3fKDPVBWYCLoXG  
KCR9Xj10MxWjmPSTKVgOo3y9eQLYnjECsD8nYvsNMIOD7lGOy6v1+RD43PjNvGYqDr68/YgOcuc3  
rwwi1fmC8DNTkW8/osxUmlH6NoX+FwaNF+wr3StK3lgyU92Rj168Ziq2NRMz1R3RFkVh8wzCzFQX  
7Htg9nhHxOuhm6nQQN6TIngfrxEO8B1gJnkP7IPMVIwK9Ntf4DOnaqbCgloaqYd612s+LNfp+B1M  
zVT06qhLt93PH/T5CmqmuiPcU6GweYsyUwGOaTBK13u7Cz28Zqp3RDhMIK9Wd0GecVyjz++O9vQz  
U4Meiuvd1mFKuWbqIVfmmalh+sEFuQvrk3EVC0Yp4z2g1XFMh1GA6lfcPs2PVH6TKLxmKt7f26eA  
KDMVJ8/cgSluBPW53IHWBTVT0dd5+znoOz/8zFS/fYuXwmwTOOkHfQDNh4FJ0EcIGJc4HsFvjr4C  
+yD89vrn8wv3yhP3LwJ1qp8QwPYCvwd7Jb8TWQC/JXSjrk/c/telFns7k6soCEkDquoIKSQ4KIQR  
oF86hc4P5gt2lhCpECYwIPCaa4JCQMF0g7DSgdjzmiMwS92RnridgPdsK0QjDFQIEeyMcEbYvXTS  
jaDLMvV71SagXCCU0cHD2NXv4wXxhAMwiBLsBDCqDMAJe180BP4bn0EfSQi8Dz7CjibobLMXCDKM  
3sAyEPfIQZDohVHivkfQPVOjKlyZipzBsNRv1YDLQTFqGalay0Mc4EDE+zsWhZmqX5qHgOkFlx4j  
AiGQUhF+EKn6OmBI4qx9OvAzU/0O0nDPPHeEJbYIHJTjMi5sc6g9GFWobV2IYIQfTBJX5HC3F10g  
4TeB2ePWOA74YZp5zYVUzVQAA9N7cgP3eEsHqBMIQRx4u+vGdoJRE/hdYU5iHhxc4GDaOzoPAh1  
B7GNbcydhiiomQpzEidv3BGz+H1Q2xhhD1MAAtYaR+dgm3FsBIPBZcD9ndx3oL10RjnWgVIF76Otg  
gKEEY8Qr+jtduMI8Qb8KYY/fsTiaqehDcPkc6hDz4Dtiv4GaxwgKHPjh3nOof+/3Nx2Z6le/QaLe  
S6pmKsD96nBgh4NqfX43dDMVdY0DHL2fxffe/gqXzGJd2Efg6gSYR3oOcDAJ8x0nafwu8zcxU3Ei  
AKCmcYme+zpqaB8jLivF++AyV9zWAAec7jwwKrBfdi+BRU68vz22a78DVFOCzFS//j7KTEWfiPuN  
uqONsO1htBNO4kAToF+FsQ0zz61PN4rCTMX+WJ8XJiT6G+QAfX/UPg7bunffi74xvVlxUwE+C/op  
rynhhm6mYp8GnanPi7xjW4H5hP0v+nns//Db6L8fcoVtB/02vjv6LXcawsRMhanjgsus0S/jdehN  
5BnbJPoe9M84saxrBfT1OHHn3j4DeEc7Yr/il6uCEGamAuzzUNfudHwHXDaOy7zRL+A7YEQ7+kt3  
HuzH8PsGXVKfipmK9evzllc44Q69gd87FQ3mN2oQv30qpGKmuuCqJ/yG3u0UoZup+MwjRtgn+dxj  
D3wvaF98Vug11DDyj2MG7zEOTrq4V694L/M3MVNxSw/kEcA4dE+W47Ogv4lGwnaAz4l6hFbH9uku  
jxG5QVfRYH3uSWLoK+9od4xadNeDwD4Tuisd+Jmp7rGCTpSZirrGdqufaMFJSZwMwX4efQh+c2hl

7z4A/47bTEUd6vOhInCLOHyuqG0Dv6e3TnVdGjeOB+/JB2w7fqAGcayD7cOd161NaAncogL9l3uM  
iN8IGhK/kQ50IY45XX2JWg4a9UtITKjKlyRN4IAIN1LH/dlgtukjs9zACA7s7DEPDEHs2FxjxwWi  
FjtZGIHeSyMg6iCoYSZgHd5RdRCCOJsMsYazahAo2LFCLOMMKu4x5jVHYCRgdCoOHnQDFgdHMERw  
wA8DBJfd4mDMawDi33gdB1IQPF6B4VKYB1Bh54KcuPe/xPfC/TD9SOUBVFEgt+46EDCd3FFfOED3  
XhqO2zd4gdDBYGTkX88ZRCwO+N3fWF+PLpAgCL2CE8v5gYNKft5EkHmO3xNGAupQ/70xAhkiCzUZ  
hXcEAK4K4CA6HUDM6uuGyeXeZ1MHogqCGDWEAzJsc94DBAg43PMXghmjb4POAOOz4zI2HGR6TUKM  
NoMxje0K4hGXrWJ7wsE1totUwHu7B51u4Gx3OoHYhGEAIYXtQz9AcEPvf2Ds6JcL4S8u+9Xnx32h  
/MxUiFavQacbXS44wMe9E2H6IY/eZSAQkVMYKFin7K44AAAB1/4rBjFgs/u7X+QV5hj2DZgvuLg  
Tu+D/B5A5T3RgG3Cu17cS1YHJ8S8t0eA2eyKexx069MQ+n3BcOm4Pg0HiBDOXpAD7+W56Pe8ZiqA  
OYRtHwew3lFr+P3RJ8Ps85qFpmYqbpXgrV/dVAkDBqH3QU74t5+Z6olDNtyzVF8God/uxgV9AwxL  
mDTIk99IP/RP2GaRD9SDa14C3BPZewCGEYN+4OSEPh8CJx5dsK2gT0A9o5/QR84iYGphm8JBE65c  
8fZrylnXxAs6QDUfTeKtX5h1fgen6Ef0+RDekz/4TPidYKqjXvV9Cb43Tmxh1K73/n3YH7rbOcwX  
fRoCOsMP7Ju88wbpB5geOLj07qfRv+H3Qi7CwLbuHc2NfipVvCYeTCe/UeMuqEGYC/qBNQL9nbf/  
xT4POg77D/R53lFmCJge2B9iG8LJQ5ygcvt59CPYFvT5YQz5man4zF4zHN/NBTWAA3loLpyw8Pss  
ONmBkwz4HJjXW2/oZ/X5sb/33sapoOgnNxDoG7y3n8J3QH5w4guGkne/iX4P9+DGPgz7IXyHMHAS  
Tl8e+zyvTsA+DSch0F/pJ+cg9/G7oh6i8Br2MNa9RksQ6Hv0ZRE4CRcEagPbjVcflWnnCHwQyH  
AQadiGMSry7Dv7FtYpAl9sX43Lo28ztW8LuNCIBRqM+Hk4fuQAvUOrQA+iFoHhhd3isfoPeg5dC/  
wPj0ahAd9CuuTsA24x3Q4TU88XumqhGj8N4SDHXjZ+j76Qds766Z6oLjPpxUxn4KutCdF/nBfgFX  
THq3TZjDupnq3VdhJLLfvgonzVLVD9gWcdyEdel6EcdG0PFh2x9MW30/hEj1pG86gK7w9h+o8SCQ  
Q5imONmAXHr7WugS/BY4Dkcd+WkmvIzJE/cho+ifsV8lpAhR1UoliZ2ws4lh07xAgLqj3XBDyNW  
vWa0bqZihlpXREQR9nn8DA0vGJWBUTkQpxCFECzeAym8B87+eh8EAbEHkROF/hnTIVudoOXwehzv  
B6KWLCy648Tkc+NkA4wsjCby3rfzv8TkO4TNG7WeoiaVz1rYeUBYv4BpqbwHiHoffToO7GAA4CAQ  
pi1MUxjqOhgRiRNr3lG3QSMv/Uj1s/tr2OI+FGSZIEy+m8m8BUffB07uwAyE8YqHin3caZEft59  
eirfBaPMYJag7lB/2N95T8LBsMQJQ+9VHNjn+Z2g0fF+hrDPlMrnNSFqfYWZxtDPMq7vaPLZ4/ge  
Xtz1wICBVsKoZZw80EePFpSi+Pxe0vmeBV1X1Puk8zO6FHQ5P1L9fFHVmc7P5Alj2B0kA8NSN1P1  
94MZCJMS9ex39Ykp6coJwBUbGDjgHp9gJKjfiGYycrhcXO+7cVI2FcM+1c+Uyuf1w285/TUYIDj5  
AZ3td4u1OIj6LgX9rmHgpb9OKLq3BcS+1jW7CSICVPURQnYqMcrXvXQblze8YgVnofG6e5YPZ+tw  
yQpGFRQFONslcYJRI+77429QYDrOMmPkGkYk+J3ZJTs3MB8wQgMjuPEwMu+IBkJMwch1jBbyjqQO  
C1zJgAOpoKsHyH8L+n6MjsPBEa4k8Ru5vCOAUWQYiY0Re3515hcYpQXzFbfmiOPAkuycYOQYbiWA



UedBt20g5L8ClyUxilK/ugP1ilGbuGJOByMIMQ0jotE/7oj9HEyYr/ivW9yUOD4BCNE8Z1wrOU3  
OnJHAYNWMCAfV0PA0Pa7PVhxArfDcG/FgCv6/vkntQE/hKQZVYGEkJOjKjNLBTgMjO3HJJJoSBPiL  
AgZn7HBpCS7/xDy4zDqVy9fTCT4HDA9cvoz79uHSKjwkCQGhcuRcekGLoUu7MgjsuNC44DECQw4  
9CEYSYJ74aF/wcO+8EATXBqJk0+4dQgNVJJucNUHzFHsa2GEYV+L2sMJTdy3FpejY//sd5sWQgjZ  
0cEtDmAion/DPhV/sb8tDroOtwzBiX6M5MTxCfpt9N/4b7yGQSG4SmJnGOCRSTob+10cz0LbQe/h  
d+IADfIfQjOVZCb6jgf3oIfIwIEQ/ulgyL2BeyaAs8swPCEe8N1hwOLgEAeJprcIIAUHYgAmM2rQ  
NZ3TdR9WQggghBBCCCGEpAWaqaR44z5Iyb3UHA808hpUzzxjT3MDI5y7T+gs7uCyQ1yq6L1pOR42  
gSfkBj2wiKSfQKehE0IIYYQQggh5D9Hha0TUoy5+eZkcwpmKi6B18FILLi3DAI3J8dDS9J9WUe6  
L8EIW5/Je+GJw3p+cANv7z2BCvPZ0/G9U1lHOT4nFfA+BXmvVJbBw5lophJCCCGEEEEIIITs06mid  
kDQBwwiXKn/wgchll+U9dd4blSuLNGpkP60XI5Pjqbc6WA+eCP/55yLXXWcbfGXLJq8DN50+4QSR  
Bx+0n64IQ9QF98C59da8ebOykv8i9t3XfpojblaNpx26r7sBY0sH5iqenN+7t8iVV4oceqhlhQrJ  
y2D0K75zq1Yijz4qMmpU8shOPDISnx/Ro4f9/rhpuHda7dr20yRxDx9ciu815G67LXn+7GyRsWod  
iSHgMv5zzslbzi8/e+4p8r//2cYz3hf3pMEI3vPPTx9+IZOTNy8CJiBGuF56qf2U46ITk01Z783e  
q1e370v00EMiu+1mv4bPj5vXu/PgAVq416oL7quK38ydsBDPZAfF9yDVc8LPheetjxrlj0dnwm3  
M7j9dvtpN7vgcAoXXwX1DDWDXPdm3t8T30Z1DQuz+/ZM+/7oA5wCwmAz4575+EG/Rj1W65c3rJV  
q4rcd5/l/ffnHyFMM5UQQgghhBBCCNmhUEfrhKQJ3O8RT5GHKQYjCIYcniilm5XJrt8w2GA2uU+Z  
R8CAhJnoggeFPPxwssEGUxHG619/2eYeboAOMwuGnLsOmKpz5iSbXjCwwCMv8UTKadPs6e68zz6b  
PA+MLvcyf8wzeLDINDckG10wwPC0YRi3MDrxQCg8Wbp+/bZvj8BT9HHvSwCT0X0dAfPysMNsEw1G  
LUxlmIHduuU9rR+B94Uxhxzq4P319cFgxvJRuE87RH5g1urrwOfH7wXwUKt+/UTOOitvOnKNz/bq  
q7ZBie+OfOI32333vNsp4LvhaZ8wukHHjnnrCAoY53ioFp7kjH/DQlcpjvu54jPDXG/dOv9yMCLd  
B3y8+KJthOL9YZRifveJlpinbVuRihXt5VCH+FxYP3KLG5r37WvnQF8/jGfvSGU8eVefxy/atbNz  
A2MeBrz7Ogzeww8X+eQT++QD5unSxa5nN39u0EwlhBBCCCGEEEJ2KNTROiFpYuNG25Dck+NhFGFE  
4D332EYkDKIjj7WNUX0EJAL37XTBf+ujUHFZReSHH5yJGq7J+eGHtgGG0YYw97wjCGHo6e+FEane  
e6aGmal4ANMllyRPP+YY2/T1A+YiTEF9fhjMMPRwD1L9dZh5eNq0HxddlDwvRIHqT6P2fk+dsGle  
MPpSfx+YuxijCZBPfHZ9eocOedO9YP4TT8ybF8bg5Zfbn8e9d60btWqJvPWWbULr4MFFMBZdMx3z  
uWY7jHEyY4ibbhK5/np7nsaNRV55ReTff0WuuCLvPWCcwnhHXQIYsjAvMUIUo3NhVD7wgH0rCOS7  
eXORQw5JHjWKOPXUaDMVRjIMZm9t4ftgeX3es8/2f7ozthnvSFmaqYQQgghhBBCyA6FOlonJE3g  
UnAYYRgh6ZpBMNdWafPw4fY8MEC9I8frZirm1c1WjADFMI5g0GEkKkYVwhzD5dQYSZmKmeqOvHSJ  
MIO9ozdhCnvX4YLRpa7J58bBB9vmmZ+ZCrPPD6+ZilsdwJhLN1FmKt5Xn47fd/16e7oX5KRp0+T5

L7zQ30zFbQ3cEaM6GC07ZUry/C1a2LeOOPBAe1Rnmzb2KGU8sMk1PmFwY9QzjFX8G5f/45YMOqgP  
3H5Bv6UC6gHGBp/+dn5xWwaYqu50BN4/ykzFvXhxawovMMCPPz55XtwqwXtrC4CRtzRTCSGEEEl  
IYSQHRp1tE5ImsAoUd0MwshEXlo+Y4Y9KhDTMUrPe19I3UzFPVQxUhD3oMQ0zlvLtrt3zxvNh/uC  
wszCbQTcdx1l30bAfcSdhevmYqRsbiXqU7UZf4wPPURj6VK2bcLwMhbjMQFuP3AF1+IHH108kOE  
jJtO5J137HneeCPvdURhzVTcbxXGM8xp/MXI9zCWTQgzU/Gd+vQROfnkvOn4zA0binz6ad5tBxYs  
sHOInOi/LUaGwwAE3tstwNDEw76CgLG10bhuLpFz/MW9dnFvUpiu+L1hdO+6a/l8GM2KmkLd6cDs  
xb1uMY8bp59uj2qGaY6TARh5W6NG8jwwU/F+On5mqneULYDxjN9dN5lxsgGG7cCBdn3BcMWtHVCb  
vMyfEEIIYYQQQgjZoVFH64SkCZhquFQZlzG79zN1AwYV7k8JcxWKGkw53JsSl1XjNdZr031gEYwr  
jEbFw4lg5OkPJXIDhicuQceoQtwGwHs/URdcRo/3hcGmL4/3xaXhAPd11achvCYnjEWYX3goEz6T  
d31u4Hs2aWJ/LtxvFJeeu/g9gMrvFgZAv08pAkYpLIHXwUOU9HlgZKbyACqdc89NXgcMPa/Jie8A  
cxOjRTHy0/vblmDmYImYk7gHLPKHnLlgNKY+P/KHegkChu7HH+c3NmH+uoY5RprCWPUapHjAk9+t  
CDZss1T/DYYkarf2xYm8Smn2PfehQkM0xZmLkzSCy6wfyfdzPY+gAqf03uJv87MmXadIT/e/JUv  
b5v1uD8szHjcTxWmNrYPbE+EEEEIIYYQQQgjZYVBH8oQQX7y3DCgMYevSp0W9pzs91fUFker7FJa4  
3icd37EgYJ1h643rPQkhBBCCCGEELJDQDOVEEIIYYQQQgghhBBCUoBmKiGEEEEIIYYQQQgghhKQA  
zVRCdjbwZHncnxMPWMIT9/EXDzEq7uCeurg3Lu7hiu8/e7blnDn2g58WL7afkO99ABkhhBBCCCGE  
EEJIGqGZSsjOBp5iv9tuautVm68beHjRjgLMXjxQDA9ywsOV9txT5LPPnlkFYpp0kYcfth/KhCf2  
42FX+ndH4GFSISuLtGljP7jK+/R9QgghhBBCCCGEKDSQcP4SUjiK6iE5hX34T6qfMx3fJ66c/PWX  
bVDqZuJttzkTFVHvm67PtXWrPSIUT7p/+22RF1+0Td2mTUVKlkz+fL17OwsZgJGo330nct55eeup  
Xdt+Gj8MZZePPHjP2TLvPXfdVeSRR+w8EUIIYYQQQgghhKSRhPOXkMLz5pt5ppcbMNI+/VXk+OPT  
f++zj8iwYbZRhkVtV/lG5JprRA46SKRcubzIKIYUOeYYkVtuERkyRGT1audNFAMHijRsmDcv4pNP  
nlkOX30IUqJE3vRq1UTeeMOZqMAI4s8+a7+POw+MwLFjnRkUulQe36lVq/wjQRFly4ocfrhly62  
obh+fX6j8oEHkpfBZ8J37tdP5lgj7NeOPlrkzz9tcxKsWiXyxRciV14pstdeecvCQO3Z0zYPq1fP  
ex2hm6kA+cJn0+c57TRnYprAZfW//SZywgkipUuL1KghssceIrvskvy+iIKYqgOF18sUqFC3nrO  
PFnk4UJnBgfkHCNf8XtgHoxShen6wQfODIQQQgghhBBCCCHpleH8JaTwvPNOunlhneE4v772wbc  
zTfnGZkwRjGSEebm/Pn2iEKYk+3a5S135JEil7wgsnSpfQn3zz/br7nTr75aZORI+3MMHizSqZnt  
8LnTEV27isybZ88Ds/K44/LmgBH5xx/2ti0bbXNSN1BPPFFhk009Fxo0TGTVK5L33RI49Nm86Aqbi  
okX2OlzwvFR5EN6c4HNMmCAycaJtlapkcN5uT994uMGCEyaJBt3Fatmv9Sd6+ZCkO2fv3keU45  
xZkYgOmI1aD5+/fP//kKYqbi/qiNGyev5/zz/d/3l1+SzXgE6oUQQgghhBBCCCEkjSScv4QUHq+Z  
ilGYHTql/PSTfR9NsGaNfTl4gwbJ86YSGI2JUa0AhhoMRlyGxLSaNUXuvdeeBgMThiNexz00cVl4

To49UvPpp21D9qGH8tYLExSjPVessJfHQ4zwgKPhw0UGDBD58EPbDOzVS+TJJ+31YbSq91J7jK5d  
sMBeh4vXTC1TxI7e/R4u+D4dO9r3BHxnhVkmG1c3D/HfGJnrHSmbrrsv8YQbjM8P01tfvRqVKlPdd  
Zo/sDeLrr9NnpsJo1tdjYqaizgghhBBCCCGEEELSSML5S0jh8TNTcbm9DkaGwsg8+OC8+e65x77s  
3wQYalOmiHTrlnc5P0a4YITkqafa/4ZJCpMNIz5d8xGG6INPiZx7rv1vLOuasC4wfvEQo9NPt+dB  
YLQoLjF/6SX7fXFJ/+WX501H4DtFmal4SJI7glYHBq5+b1AERr7inqReRo+2R6zq83pHphYUGM0v  
vyzSrJl92Tw+Ly7bx/fHw6Qw2hVmMJ6gH0S6zFQY7+3bJ4/Uxb1R//472VDFSGX8Xu5l/jDODzkk  
/60fCCGEEIIYYQQGqpJwvILSOHxM1O//NKZ6ID7gmJU4znn5M2HUaVXXCEyaZlzkwKGIUw73DMU  
DxSCMfroo/ZtAHRw6TtGcMJA7rcWwfUqWPF7xSjG3GPTYyQ3X13e5p7D078xT1ZveYmjFJcEo95  
3MDI5kOH2tPdka0HHpg8D8xU7/08/cxUxLLvBeYg7i970kl582IUKAxdPITJBaJGzXK+75ueM3U  
detsI7JuXTs/+IvvWhSYmKm4FB+jYGFy4y+McNyrFsAkxW+De8e660EtYHQq6gPA+MzoY4x0xr1S  
MQ9GDKMWvcY2IYQQQgghhBBCSCFJOH8JKTx+D6Dq29eZ6AGjC3Hf0TPOsA1T1wjTA5fq4zJv3F8V  
D7HyAw+XwntgJKK+bJMmtukJ8KCKb79NNioRMO/c+6TqwMTD5f2472qtWvk/Gx6ihQcJnXyybajC  
uMUtCPBZsZx7uwDw4IPJy2Jd3kv8dWbOFLn7bntd+ohMN2DqXnSRbdziUv4LO7T63WK4gFUQeDh  
Wvr7lvSHf+nA4NXngxmOh3HpYHQuRpki5zCFvffCRSD/MJlhcUoetoQQQgghhBBCCCEXkHD+EI4  
wu7DGTytUymuOWEdEEIIYYQQGghpBhDM5UQQgghhBBCCCGEEESgGYqIYQQQgghhBBCCCGEpADN  
VEIIYYQQQgghhBBCCCEkBmqmEEIIYYQQQgghhBCSAjRTCSGEEIIYYQQQgghJAVophJCCCGEEII  
IYQQQkgK0EwIhBBCCCGEEIIYSQFKCZSgghhBBCCCGEEIIISIAM5UQQgghhBBCCCGEEESgGYq  
IYQQQgghhBBCCCGEpADNVFlgNqv2h2qDVRvuaXjtL9UIIWRHZptqk1QL6scmq4Z5CCGEEIIJfMxU  
zU+PDVFtnGobVSOkMMxSdFwk1xcaXhur2gbVCDGBZiopEltVq6dalqCdqxohhOzlrFXteNX0vktv  
zVSjsCKEEIIIZfLVdM1mN72UW2OaoQUhi6q6XWlt5qqTVONEBNoppICATO1vmp6J6S3NqoRQSiO  
DMzUJqrpfZfemqtGM5UQQgghJF7aq6ZrML3VUY1mKiksNfNJUqGZSgrEctUOVU3vhPR2vmqEFJYZ  
M2bluHHj5I8//rBi/PjxMnHiRFm3bp0zByEFZ5NqJ6im9116O0m1raoRUhhmz56d1I9NmDDBitWr  
VztzEFi45s6d61tjK1eudOYgpHD8888/+WoMmmz58uXOHlQUjg6q6RpMb/uqNI81QgrDNarpdaW3  
WqrhNgCEmEAzIQSyfft2579sIjOPvpokZUoJdUOqiYlp5TM7YDytU8Ssuuuu8qv3/7qLG3jXsfJ  
bLz1sGbNGmnSplmUKqVqrFo1yc7OlKRC1ZMnKIWqJBURVpSqVavKt99+6yxtU5ga27x5k/wxZYJM  
nDpRNm7kvZmKA9562Lx5s5xyiISMIFSqu9dXUoNL6X3XMItoKq1GpWk3/v9nKVt2I8RHb96OPPM  
M63+q3r16lKypNpX+vRjFSpUkCpVqlh/P/zWQ2dJG9YY0fGrh/POOy+3xrDPDKqxXXbZRcqVKydv  
vfWWs6QNa4zo+NXDZZddZtUY9HxOTO5vjZUvX97SYmXKIJEXX3zRWdKGNUZ0/OqhY8eOkp1QNVZ5  
VynzcRldgSW3mQkpWbekPHnPk86ShKRG165drRqrXqG6lOkdUmPzEpJ1UJZ0v6W7syQh0dBMJUno

O7rffvtN2rZtm084lUiUkMSu6r8nqghoJT5X83iWO/744+Wdd95x1k6Rlanov/uwYcPkkksuyVcr  
JUkrx9v+M1zbLPj5KGGHH5aPnuojA279Tn44b6D8eNJPmRDXz/JTw59loqfmv0iv7QZJL92HiSD  
uv4mg679TX6/cbB8ddE38sah78jnia9kwC4/yDcXfiu/Xz1URnUdKyOvGyUju46WEerv0C7D5dcr  
BsnAS36Sny74WQa3UfOcM0bGnDNBhrYcKT+d9Kv82GSg/NDkR/mx+c/y29ID5JeLB8mAjt/K17d/  
l1882U8+//BzGfDrt/LnnElOJkg60Wts9OjRcvnll+cz5q1+rLz6799VBLVfVJRRoS135JFHyiuv  
vOKsnf1YpqL/7hilhQNCr9mQSj+G8M532GGHyfPPP++snTWWqei/+5QpU6RLly5StmzZpFopal0d  
fPDB8tRTT8m2bfYD9lhjmYn+u+NKlJgOMOD1Wiloje2///7y6KOPyqZNm6z1s8YyE/13xwj6m266  
yTqJqNeKpcfW371VBLVZar59kmusTp068rDS/LxaLbPRa+zff/+V2267zTrJqNdKbo29qiKozVPz  
HZhcY3vttZfcf//9smrVKucd2JeRZGimkiQwaqtPnz65nUigilowUxN9VXiW0dd19913W5c4skPK  
PLZs2SJffvll7mitVIV6aGQlpFTZUtKqfGt5v8xHsiKxWraq3m1ViXXyb9Zi+SNrivyaNVS+z/5F  
vs/6Rb7N+lk+Lv2FvFvtl3l/r4+k7+5fyk/VBsmQKsPkld1ek//VuVfuqHen3ND4Ru160g1y06m3  
yM2n3yl3tLxRrj7jGrmqdSe5+PSL5dSGp0rb5hfidWfeKpC0uV++qt1fFuUslU0ltlrvvzmxTbaj  
lOWUcP5qMaTqcLnz8m4ydvw49QJF6ix7777zhqNhfoIrLEoM/VnFR4zVV/XNddcY43Ydw0JkjlS  
3bpVfvnll9yDwrT0Y07o6+rQoYMsWbKENZaBoMaGDBkiNWruYFcXhQ19XRdddJEsXLiQNZaBoMZG  
jhxpGQZ6faQj9Bo755xzLJODNZ54DfH7SD2228/39pliggzNVFbhTa/vp7TTz9d/v77b9ZYBoLf  
fNKKsVK/fn3f2kiKCDM1cYAKbX59PU2bNrVu28Qalzo4pCckFxy46Z1IYBTATPUGRt/gIJfKfTde  
e61vPRQ0srOyZL+cA+TqitfLm2XekZFZYy0TdWaJv2Vg1m/ySfaX0qfkF/JNye9leNY4+aHsL/Lq  
Pq/JDRfcKFdec6V0vrqzFZ26XCUD03eUK66+Qu5tcZ8MqPCdLMpaapmg67M3ybyqSG2VtqQ2yJme9  
rCm9XlaXXierSq2V1TnrrH8jtmeJLCy3WO4/9AFpf/oV0qltZ+l45lXsqXlnufOou+Xbqt/L/Cx1  
0OoYq38kJsmdFR+WP776w04OSQs4K+1XK/miAGaqN2rXrm2JK5JZ3Hvvvb71EEfsvvvuMnXqVOed  
SabQs2dP33qll3ArANwDk2QWzz77rG89xBG4HcCIESOcDyaZwmuvveZbD75haKZ6A1eHDBo0yHln  
kim8++67vvXgG4ZmqjdgrnpvL0cyG5qpxGLEvHny2GOPJZ3VCY00mKk4QLzzzjuty4AefPBBxs4c  
/1PxxloZKgLaw6r1UK1R/OaSuEPVwH2FjO4qHkhIq8PPlccSz8lbWe/LzyV+l/ElJsmorPEyl2uO  
rCqx1h4R6obq8fru94Wce04buaJb+lyTRfr0klE5y6drejQtYPcfcbd8kWlfrIga7G1DEaZbkkh  
MO/i8svklunvk7aXXChXX321HddcLZ2u7iRntz9bHm7UU+ZIL7Dm/TsxX3pl95aeF/eU+x6/Tx54  
8AH//DJSiu7du1sGBG4p4tfn5Is0mKkwIXDZGvpPv8/E2IniHhX3q5iolqB1V+0R1Zr93EwSd6oa  
uFeFX/+UrlDvUfHJinL9kuvlMdX0z+LbpqnAd0Gf7P1+jP8+7nVirIqA9pBqj6p2yu+nSOLuVQP3  
eGoi3aFqrNyj5eSahdf146rpn8W3zVLRXQVrbMcM1Bf6spEqAhpqDP1Jy2EtJdFN1UDcNabquHSP  
0tJpXqfUauxvFT1V3K3C7zsydoqALmrdurWvdvKNQpqpuiOIng4lef1zVmM/nYRS/wG/dpk0b33rw

jUKaqVIZWXL++edLv37Jz1IgmQsO/wmxzuS5l8SmFGkwUxnFLEqq+E5FEbdOI3SRIYkR9mhPFVtV  
bPGE+9q6xCbpfeg7ckb7M6R95/aW0emaqFZ0zjNTP6/UT/7NWmT1kn7GqV9g3sXllsktx90mbS9u  
m7RexJVdr5RuLe+WPpU/k2VZK2VZYqW8X/ljOaLEkf45ZcQbaTBTGcUwPlSxs7avVWSp8PtejB0n  
3IKxs7aBKkJU+H0vxo4TL6rYWdtgFdg/+30vRvGNQpqpDEZkFNJMdaNVq1a8VSGxwOE/lfLTtz/l  
dhCB9xnRg2Yqwxv/kZl67iVt5JHEE/Jl9jPSK/sD+S17mKzKXiuSpQq7hB3bVbiX1vfZ71Np3eZs  
ueKqK6yRqUVhpmLka6fOnaT1Va3lwSYPyazsuda9VBeUWCy9s9+Vw0s08M8pl96gmcrcwC5qpjLiD  
Zioj7qCZytjZgmYqI+4ohJmq+yPNmjWjmUoscPhPMhy9M7juuuuSOo7ASIOZuuuuu1oPcOnWrZvc  
csstjJ05uqq4TcVUFQHtdtXuVK3BTw0k0UPVwJMqnipEPKvi6YRUbl1Vjt77eLm16l3ydc63Mjgx  
XF7Mfk1eL/m2fFXyWxmVPV4mZU+TaVxzU6JeTKo/GB57oAX5OpLusjIXS+Xztd0tqLTtZ3kqmu  
kotvvFjuOONOGVD+O1mWWGX3kgaxsswaebj+l3LjqZfJpee2kxvPvEm6tbhbuifple/v/ZH8UXaS  
Ze5i3qmJv6RH2Z5y67m3yc133+yfW0bKgXu3nXXXXL00Uf79jn5lg1mauXKleWqq66yHqrn95kY  
O1Fcr+lmFWNVBLTbVLTtWOHHCuJnqoGnlDh7ZvSGeo9KrxWQTos7yB3q6Z/Ft82ScWtKtAn+31H  
xn8bN6i4UcUIFQENNdZNtRNGniCJR1QNFEGNIX2prFy++PLUauwvFberYl3tmIH6Ql82WEVAu1U1  
/NbNxzaXxKOqBh731ES6Q9Vxzgs5csmCS1Krsdkq7lJxnQq/78jY4ePWW1WNVK106qmn+mon30jD  
Zf4XXnih/O9///P9TlziFW6NtWzZ0rcefKOQl1NRY0OHDIUHcTY0VAkO/wnJBff+8+s88kUazNTm  
zZvLsmXLnHcmmQLMc796KEyUKFICylcuLxdVvVg+q/S5LM1aLtMSs6Rf1gB5qeQb8lqpt62HUH2b  
/ZP8nDVYvs7+QXqUfUpuqnO73HDcTXJHw7vk4Xo95KXaL8sHIT6SLyt+lwp2HCBvH9RbHj/mcbmn  
2T1y02k3SZfWV0vH8zvKlZdeKW3Pu1BOaHyitDq9tXS4pINces5l8sihj8qQcsOsB1H9VXKmfFb6  
C/VZVuYardtU4An/GxKbZUtIU4zYbazcf9EjMmHMn3ZySFrAfZRwXyO/WkmKNJipRx11IPUEWZJZ  
4D5dENV+NZHuOPTQQ2X69OnOO5NM4fnnn7ceqOJXE+mOAw44wHoaMskXn/9dSlbtqxvTaQ76tSp  
l2PHjnXemWQK7733nvXwMb+ayBeFNFNr1aolw4YNc96ZZAqffvqpVKpUybcm8kUhzdTddttNfv75  
Z+edCaGZSgKYOXOmdT8Qt/PlD+m/gZmqL4snXw8ePFi2bNnivBPJVOBmMsmXXHCBb50UJnavsLu8  
fscb0qt9b3mk3uPyZKVn5dnE8/KciqcTz8jj5Z+Sxw95Wp4742V56YLX5eU2r8srF74hz538mjxV  
+zV5P+tjeXvPd6V72yflkXZPyxPtnrPiqctfsP72vPAJufes++X2ZrflTUffLHc2uFMeOqyHPFPv  
dXm+9pvy7C4vy1M5z1jxTOUX5Lm6b8hTDV+Sx059Up5v/6J8dP/HMvj9wTJ96AxZPHeJrFm7RrZu  
2+pkhaStf/75R9q1a5dbG/lqzNBMDZevWrWq/PDDD7Jp0ybnnUim8u+//0qnTp3y1UhBw12+QoUK  
8vXXX8vGjRuddyKZysKFC+X666/PVymFDXd5GLV9+/aVDRs2OO9EMpXFixfL7bffnq9GChru8vj7  
4Ycfyvr16513lPKKBs/ce++9+WokKQzMVH35t99+W9atW+e8E8lUVqxYIT169PctkdwwMFP15V99  
9VVZs2aN806E5EEZlfjDlvHgRwum8VlrBdffHHeaK+yKiaojkY19V9JzXrtffVfar6Tjz1Zrr32

Wmv5yZMnW+sEHBZP3BrYunWrJbA6duwol112mZQqVSp354Xw2xnqrzVt2tS6PQWWHz9+vLXOgrBp  
60aZOmOKfP725/Jtn+9k6bllzhSys6L3MxiteuWVV0r79u2TR+L8GtKPfav+S81z/OHHWzWGp8SO  
HDnSWSP7MZJcAz179rRqBHVWsWLFvBpTEdWPNWrUSLp27SpXXHGFdcLRhTVG9Bp44oknrBqBppqS  
pUpqPfm9hIH0qDH0gfrIGtYY0Wvgueees2oM95CvXr16YD35vXbYYYdZNXb55ZfLd99956yRNUaS  
a+CVV16x+iE8BHb33XfPq6c3VD0F6bHp6r8qJuSQPQ7JrbEvv/zSWSNrjCTXwJtvvmnVCG4niBHL  
uTX2QkiNzVX/V50hB1Q/wNL8WB6jXl1Y8QPmqkkJbwdyCrVDIUtsT2vI8pt6rULVHNh50NSwVsn  
GCOTdu/LM88805mTNUYKxlbVTlQtqB87WTUX1hgpKC1atPDtwxDHHXdc7pUarDFSUM4++2zf+klc  
eeSRuaO2WGOkoGBAhV99IQ455BBrVBghheFK1YL02L6qLVSNmKJwrWpBNVZLtbmqEWICzVRSIBar  
VI81rRtKam1UI6QwQJg3bNjQV7gjcMNxjGotNkQd46brGLio3mcnYK1qTVRTFeXbmqu2QTVCgpu  
BXHKKaf49mGlxo0by+rVq525CTEHBql+WyzvNGjQgPenJ4Wmbdu2vvWFOPjgg2XBggXOnlQUjPaq  
qYrybXVUm6MaIYWhi2p6XemptpmrTVCPEBJqppEDQTCVxk3FmqSPUM6fKT4mfZFBikPV3dLXRsnB  
Zmdq+vjrrg/k18SvMjgx2Po7Zo8xsml+4e4Bunn+Zln64VJZ0nuJLHIXtTlZMP0HdeMpJlK4oZm  
KokbmqmKCCZSuKGZiqJG5qpJN3QTCUFgmYqiZtMNVOLipkdZlpG6vDEcOvvhIMnFNq0Xdl/pYwo  
OUKGJobmxqLXFzITdzoppK4oZlK4oZmKikKaKaSuKGZSuKGZipJNzRTSYFYvGKx1L+qviQaqe7n  
FE8ck5A2D9BMJYWjOJmpK75ZIZNOnCSTm06WSU0myZTTpsj6P9bL5nmbZd5982Rqi6ky54Y5snHO  
Rvn7rr9lfN3x8sfhf8i4OuPkr/P+ki3L7XsqWmiX4a8bt04WPr9QZrabKVPPmCrTL5gu8x6cZ40Q  
XTd+nczqNEtmdXBC/ffa0WudJX3M1HoTZOuKrbJpziZZ8NQCmXnFTJl21jSZetZUmdFuhvz7+L+y  
ZvAa2b45+T4AW1duldWDVsvs62bLhAMnyPASw2VYYpi1XsS4fcJ9luny+LXFsvGGTvWk8nXblgr  
TW5qYvVZfv1Y81uby4ZNNFNJwaGZSuKGZiopCmimkrhp/3h7W4+drELXY+pYs85ldWTOQppppHBO  
eaal7V14a6xxQmpeVFOm/UMzIZhBM5UUIMWLF0v9+vV9RRWiTRuaqaRwFCczddFri6xRmrkmY4nh  
MrbWWBmz6xgZkhgivyV+kz+P/tMyQKdfON16LdfkrD9BNi92RoxuF9m2dpssfHGhjD9wvLU+rHdU  
xVEy6bhJMrXIVMusHV9nvlwqP8qajsD68Hdpn6X2ehS6mYppI0qNkHG1x8nEBhMt4xfr+rPxnzJ2  
z7EyPGt47jr+bPSnLO612FmLyNZVW2X1kNXy9x1/W8vqZir+TjhkgmXk4tL/jbN2MDN17Vpp0qSJ  
b30hmjdvbj0ljZCCQjOVxA3NVFIU0EwlcYMn/PvVF6JOnToyZw7NVFI4unTp4ltfiJo1a8q0aTRT  
iRk0U0mBoJlK4qa4mqmW0Zg1XCYeMVHmPzRfNs3eJNvX5432xCjQpBGjpm6+d/NMr3tdBlddbS1  
Hhic01pPk7XD18q29duseUT92TB1g2Vuuoy3hv/veyTvANq78jU8fuPI/UT1ztT84ABilsAuPPi  
PTFq1Y+d7jJ/mqkkZmimkrihmUqKApqpJG5oppK4oZlK0g3NVFIgaKaSuCmuZir+YhToyu9WolOT  
mXF5sJkKYxOmp7suTMftAfxY9ukyax2uqRllpgbdM3Xzos3WLQD0eWe0n+FMTYZmKiHJ0EwlcUMz  
IRQFNFNJ3NBMJXFDM5WkG5qppEDQTCVxU+zN1O/NzVSMTJ1x6QwZU31M7sjUyc0mW8Yp7lOKcE/T

1b+stkaPYh2pjxNm5maPcL6XLImqvruOyo0U0nc0EwlcUMzlRQFNFNJ3NBMJXFDM5WkG5qppEDQ  
TCVxQzM1v5lqsUnk30f+IVGVRuWapbhPKUaEwsjE3zF7jLHux4rpiKlyU1f/tlrG7zvems/6XM57  
476ru00A7gm7l0EzlcQNzVQSNzRTSVFAM5XEDc1UEjc0U0m6oZIKCgTNVBI3xclM3TB9gyx5d4ks  
eU/FO0tk6UdLrVGmfqwZukYWv71Yln6w1Pq7ov8K2bYh736o29Zus2LL0i2yZtgaWTFghSz/crn1  
cKnlXy2XtaPsJ/Yv/Xhprnmba6Z+mndAjWWT3udr9T7ufVc18N4rvlmRNC+W9QWfb802WTdunbU+  
mLdYBp9v/eT1snX1jvV70UwlcUMzlcQNzVRSFNBMJXFDM5XEDc1Ukm5oppICQTOVxE1xMIPTxZbF  
W2TurXOtKZ6/J36X3xK/yYQDJ8i8h+ZZpitGhsK8nHbWNBm9i/2QKhipGKWKB1utHWSbrcSGZiqJ  
G5qpJG5oppKigGYqiRuaqSRuaKaSdEMzlRQlmaqkbbmimBgNTdeWAITKz/UzrgVS4bYBrnGIE6tg9  
xlr3UoXxuurnVbJ1ucpT/kGnGQ/NVBI3NFNJ3NBMJUUBzVQSNzRTSdzQTCXphmYqKRA0U0nc0Ewl  
cUMzlcQNzVQSNzRTSVFAM5XEDc1UEjc0U0m6oZIKCgTNVBI3NFNJ3NBMJXFDM5XEDc1UUhTQTCVx  
QzOVxA3NVJJuaKaSAkEzlcQNzVQSNzRTSdzQTCVxQzOVFAU0U0nc0EwlcUMzlaQbmqqmkQNB MJXFD  
M5XEDc1UEjc0U0nc0EwlRQHNVBI3NFNJ3NBMJemGZiopEDRTSdzQTCVxQzOVxA3NVBI3NFNJUUAz  
lcQNzVQSNzRTSbqhmUoKBM1UEjc0U0nc0EwlcUMzlcQNzVRSFNBMJXFDM5XEDc1Ukm5oppICQTOV  
xA3NVBI3NFNJ3NBMJXFDM5UUBTRTSdzQTCVxQzOVpBuaqaRA0EwlcUMzlcQNzVQSNzRTSdzQTCVF  
Ac1UEjc0U0nc0Ewl6YZmKikQNFN3NBMJXFDM5XEDc1UEjc0U0IRQDOVxA3NVBI3NFNJuqGZSgoE  
zVQSNzRTSdzQTCVxQzOVxA3NVFIU0EwlcUMzlcQNzVSSbmimkgJBM5XEDc1UEjc0U0nc0EwlcUMz  
IRQFNFNJ3NBMJXFDM5WkG5qppEDQTCVxQzOVxA3NVBI3NFNJ3NBMJUUBzVQSNzRTSdzQTCXphmYq  
KRA0U0nc0EwlcUMzlcQNzVQSNzRTSVFAM5XEDc1UEjc0U0m6oZIKCgTNVBI3NFNJ3NBMJXFDM5XE  
Dc1UUhTQTCVxQzOVxA3NVJJuaKaSAkEzlcQNzVQSNzRTSdzQTCVxQzOVFAU0U0nc0EwlcUMzlaQb  
mqmkQNB MJXFD M5XEDc1UEjc0U0nc0EwlRQHNVBI3NFNJ3NBMJemGZiopEDRTSdzQTCVxQzOVxA3N  
VBI3NFNJUUAzlcQNzVQSNzRTSbqhmUoKBM1UEjc0U0nc0EwlcUMzlcQNzVRSFNBMJXFDM5XEDc1U  
km5oppICQTOVxA3NVBI3NFNJ3NBMJXFDM5UUBTRTSdzQTCVxQzOVpBuaqaRA0EwlcUMzlcQNzVQS  
NzRTSdzQTCVFAc1UEjc0U0nc0Ewl6YZmKikQNFN3NBMJXFDM5XEDc1UEjc0U0IRQDOVxA3NVBI3  
NFNJuqGZSgoEzVQSNzRTSdzQTCVxQzOVxA3NVFIU0EwlcUMzlcQNzVSSbmimkgJBM5XEDc1UEjc0  
U0nc0EwlcUMzlRQFNFNJ3NBMJXFDM5WkG5qppEDQTCVxQzOVxA3NVBI3NFNJ3NBMJUUBzVQSNzRT  
SdzQTCXphmYqKRA0U0nc0EwlcUMzlcQNzVQSNzRTSVFAM5XEDc1UEjc0U0m6oZIKCgTNVBI3NFNJ  
3NBMJXFDM5XEDc1UUhTQTCVxQzOVxA3NVJJuaKaSAkEzlcQNzVQSNzRTSdzQTCVxQzOVFAU0U0nc  
0EwlcUMzlaQbmqqmkQNB MJXFD M5XEDc1UEjc0U0nc0EwlRQHNVBI3NFNJ3NBMJemGZiopEDRTSdzQ  
TCVxQzOVxA3NVBI3NFNJUUAzlcQNzVQSNzRTSbqhmUoKBM1UEjc0U0nc0EwlcUMzlcQNzVRSFNBM

JXFDM5XEDc1Ukm5oppICQTOVxA3NVBI3NFNJ3NBMJXFDM5UUBTRTSdzQTCVxQzOVpBuaqaRA0Ewl  
cUMzlcQNzVQSNzRTSdzQTCVFAc1UEjc0U0nc0Ewl6YZmKikQixculvp7h5ipp9FMJYWDZiqJG5qp  
JG4sM7VRiIaj2YqKRyWmdo8xEzdt4EsW0IzRSOtmGmKk1D5YF82mmksJBM5XETZd2IWZqhZoy  
bQrNVGIGzVRiznaRxaRVV011P76tjWruvIQUBJqpJG5oppJYUfu/TaqdopqqKN/WWLXVqnFfSQqE  
qpvtqrVSTa8rvTVQbZlqrDFSJIy6aauxld6O1i1BaqqxkhhoJIKYkX1T11UUXl22qqNk019mPE  
BJqppECkbKYSUKBoppK4oZIK4iZIM5WQApKymUplIUIJTCWkENBMJXGTkplKiAE0U0mBoJIK4oZm  
KokbmqqkbbmimkrihmUqKApqpJG5oppK4oZIK0g3NVJlyuC8XHjy1Zt4ambhooHy05aDcDsbWq5v  
ad+MfqOzMCEpsmTJEusegrgJ+OGHH+4rqhAwwubNm+csRUJqLF26VFauXCkzZ86UY445xre+EMcd  
d5zMnj3bWYqQ1MEDf1bMWyFzF8yVEzadO0dk9uRm46Uvxb8JbLOWZCQFEGNLZ+3XP5d8K+cuvFU  
raqSW73N9WTSgkkia50FCUmR5cuXy7J5y2TRv4uk9YbWWIUlt/237C/jFo4TWeMsSEiKYOaENBki  
7CFne++9t4wYMcJZipDUgd5fMm+JLP93ubRb107ruZLb7lt3l98X/S6yyImQkBSgmUp8gXEKINab  
NWtm7cT2228/ycrKsndsu6j4w+1+fNonKtR8NXepKfvv78Vv/76q7VO4K6fZC5uDaxZs0ZatGgh  
e+21l1Un2dnZSqliKvbYYw854IADpG7duvLtt99a6wSsMeLWAB4CdNZZZ0mtWrWsWilZsqRvLQXF  
7rvvLgceeKDss88+8sUXX1jrbKwx4tYA/p533nnW02BRKzk5OXb9ZKn4QUVQG6KidEJqIK1hLYd9  
7UcffWStE7DGiF4Dl1xyiey5555y0EEHSenSpfP6qS+tavJvY1RUSMiuObtay2Ff26tXL2eNrDGS  
XAMdOnTlrbGyZcvm1djHVjX5t0kqqiakWnY168n+2Ne+/PLLzhpZYyS5Bq699lqrxlAr5cqVy6ux  
FGKXXxaxIsO+9umnn3bWyBojyTVw8803W8eHhxyiFSoUCGvhl5WEdT+UVE7IZUTla3lUKM9evRw  
1sgal/7QTCX5gPHQu3dvOfTQQ3M7nxllSuR1RIhdVUxUEdT6qvBZtnLlylbHNH/+fOfd2DlIEu5v  
jcvz33//fTnqqKN868Q09GUhzB588EGZO3eu9V6ANZY56L/1xx9/LI0bN/atK4KEuzyMsnvuucca  
2erCGssc9N+6b9++cuKJJ+arkdzIUfGjiqA2TEVFFZ7lceLyjvvtEbou7DGMgf9t/7yyy/I5JNP  
zlcjuVFCxVcqgto4FVVV+Cx/yy23yKRJk5x3Yo1Evvp3b9/f+vWSX41kht9VAS1ySp2V+HMqy/f  
tWtXmTBhgvNOrLFMQv+tf/jhB2ndurVvjRQk9OW7dOKiY8aMcd6JNZZJ6L/1zz//LOeff75vjeTG  
qqqC2jwVB6hw5tWXv/LKK5NGRRPGiAvNVJLEZZdd5tuJ5AsDMzUoMAoHI3STzKJTp065NVBYMRUV  
GOWIG/ckM7jxxhtzayCuGnPXu9tuuyWZqiQz6NatW75a8A1DM1UPd71VqISRYZMnO+9MMoWHHnoo  
Xy34hqGZqoe7XoxAHDdunPPOJFN48skn89VCYBiYqXq468VVR8OGDXPemWQKL730Ur5aSHfo6/3l  
l1+cDyaZwttvv+1bC75hYKbqoa8XJ58lcaGZSpLorlx33XVJnUdgpMFMxahEPkCoGBJxsu62K2/z  
rYc4ot4e9WT9vPXOO4fAE4w7FxG/131d7/OthziibtW6snzmcuedQ2CNFSseeeQR33rIF4UwU93A  
JbPWPchJRvHcc8/51kO+KISZ6sauu+7Kh7tkIG+88YZvPfhGAc1UN3Bl2pQpU5x3JsWGCG3zwQsf  
+NZDHF2UVbG/ZTCSSHqsWLF559/7IsPvIFAM9UN3CZs6NChzjurUul1YyHZiqxGDhwYG5HkdKZ



w0KaqXGdnWT8x/G1irC23f5bQjXnldha0ns47xva3lDh950YO1bgfsxhbUeusfdV+H0nxk4ZKe3H  
Cmmmcl+Z2ZHS719IM5U1ltmR8u9fCDOVNVZM4yEVYU3TY3FrMmM9NkKF33di7LSRUj9TCDNVX3/T  
pk1ppBlmqnEYtCgQck3mo+KNlxMZRSzKKniOxU7a3tXhd/3YuxY8YWKnbU5D+ZjZFCkYwQqgxEa  
aRiZymBERiFHpjKKYTyhYmdto1XgYcp+34tRfKOQl1PdwENTaaYSQDOVWKxfv15Gjhwpp59+um+n  
kS84MpXhFxyZyog7wg7o0HbkGuPI1GIVKe3HODKVUYhI6ffnyFRGISLI358jUxne4MhUxg4UKfUz  
hTRTce/nRx99VP755x/HQSGZDs1UksSI16a22GEdkppGJl6yCGH8J6pxZGIE3XXX3J9bg2USMQr  
sOtUqiPr/InnvHMIPLm4cxHxe93Z6c7cGoirtzt17pGzhyyfznumZhoPPvhgXi2E7SvT8ACqqlWr  
yr///uu8M8kUUn44UBoeQFWuXDMZNWuW884kU3j55Zfz1UJgFNBMDdeLew1OmJtJeWdSbljQNr2f  
7p1XCzHrMcToH0Y77xwC9Vix4tNPP82rhah+jA+glmmGZirJx/jx462HEidSL4opJn61ltvOe9G  
MpGpU6fKnnvu6Vsb6YoXX3zReTeSieBhKnXq1PGtjXTFE0884bwbyURgcB500EG+tZEbhRyZiie6  
k8xl6dKlcvjhh/vWRm4UcmTqXXfd5bwbyURWr14tjRo18q2NpCjEyNQbb7zReTeSiWzcuNG6x6Rf  
baQrOnfu7LwbyUS2bt2a2tW1hRiZ2q5dO+fdCMmDZipJQr//x6JFi+T555/P15IYZwAjzNQSn+c/  
M3TtddaJpoL7zWSmei/+5IIS+TVV1+VnJycpFqJLOowm+ejh07Jo18YI1IjrvvmzZMunVq5eU  
L18+qVYKWmOXXaZ/PHHH87aWWOZiv67r1ixQt577z2pUqVKUq1Y+8ooM3W4mq9icp1dcMEFMm5c  
Ck8kJsUavcZWrVolffr0kRo1aiTVilVjUWbqeDVfteQaa926tYwaNcpZOyEia9assZ6KXatWraRa  
yR31F2GmltgjucZOO+00GT58OPeRJJd169bJ119/Lfvuu29SraSixxDe+Zo1ayZDhgyRbdu2Oe9A  
Mp0NGzbIt99+m+9Ed24/FmGmljgwucaOPfZY67kyW3klLQmAZioJxCuANm/ebJ3507nhyXLpadK  
xTkV8zogT6vyUxVpdVYrmTouzzwFFVEx68errvuOjnppJMsIV65cuWknZoemNayZcskYuwwoiO  
Xz3ccsst0rx5c6t+cAm1X30hKlasKkeeeqMHP182RhrjOj41QNG+zU/urmccc4ZUn1cdW3vmNzK  
TywvzVs3lyHfDXGwtGGNER2/erjvvuk2THN5MyzpzTdRuymVVVyKzetnJx4zonyc7+fnSVtWGNE  
x68eevTolc0aNZOzzjhL9vh9D62qklvZ2WXluPOOk28/+tZZ0oY1RnT86gFX+MAUbdWqVT4jX4/S  
pUtbl6j79evnLGnDGiM6fvWAgWFNj20qrU9tLbW/ra31XMktZ3GONLykofR5tY+zpA3WyTojQdBM  
JQVisWr1VdM7Ib21UY2QwoDRXg0bNvQVVQgYYTxTSArD2rVrpUmTJr71hYDhirPchBSUTaqdopqq  
KN/WWLXVqhFSUNRhnrRSTa8rvTVQbZlqhBSGtqrpdaw3g1VboBohhaF9+/a+WgyB2zbh9k2EFIYu  
qqmK8m01VZumGiEm0EwIBYJmKokbmqqkbbmimkrihmUrihmYqKQpoppK4oZIK4oZmKkk3NFNJgaCZ  
SuKGZiqJG5qpJG5oppK4oZIKigKaqSRuaKaSuKGZStINzVRSIGimkrihmUrihmYqiRuaqSRuaKaS  
ooBmKokbmqqkbbmimknRDM5UUiMWLFkv9fev77vAQbc6gmUoKB81UEjcOU0ncbNq0SU45/hTf+klO  
PqyxrF5NM5UUHDwYo9UprXzrC9HggAaybCnNVFI42rZu61tfilNrHywL/qWZSgoHzVQSN12u6OJb  
X4iau9SUaVNophIzaKaSarF48WKpXz/ETG1DM5UUDppqJG5oppK4sczUU0LM1MY0U0nhsMzUViFm

aoMG/2/vPuCcqNM/jm8v9N4WpKsoKIIFBUUCyoWVERFscAC6nn+7aeeBet5evYT2wkqggURRFQQ  
xASnAxUEC4KCwtJ75/n/fslkd5KdbZN5scrD5vH+v57VLMmWXPduZfDOZkZUrCVMRn759SwIT27WT  
P/8kTEV8CFOhbfDgUsLUvDyZP58wFRVDmApfCFOhjTAV2ghToY0wFdoiU5EIhKnQRpgKbYSpCBph  
KnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYialSp8IUwFdoiU6GNMBXaCFOhjTAViUCYCM2E  
qdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqtBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoi  
U5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYialSp8IUwFdoiU6GN  
MBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqtBGmQhthKoJGmApf  
CFOhjTAV2ghToY0wFdoiU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0  
EaYialSp8IUwFdoiU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQk  
AmEqtBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoiU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yF  
NsJUaCNMRSIQpkIbYSq0EaYialSp8IUwFdoiU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JU  
aCNMhTbCVGgjTIU2wlQkAmEqtBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoiU5EIhKnQRpgKbYSp  
CBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYialSp8IUwFdoiU6GNMBXaCFOhjTAViUCY  
CM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqtBGmQhthKoJGmApfCFOhjTAV2ghToY0w  
FdoiU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYialSp8IUwFdoi  
U6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqtBGmQhthKoJG  
mApfCFOhjTAV2ghToY0wFdoiU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIb  
YSq0EaYialSp8IUwFdoiU6GNMBXaCFOhjTAViUCYCM2EqdB22WWDPPvLVtOmjWTBgp+cKYHyIUyF  
L4Sp0EaYCM2EqdBGmApthKlIBMJUaCNMRbw2b14rs2d9K4MHfS5tWs6QvEbfSq3qsyQnc5ZkpM6R  
zNQFkp2+RLLSI4bKfp+dvtB8P1+yM2ZLjapfS8N6s6V9u6/k3H6vyLhxM5wIA94IU+ELYSq0EaZC  
G2EqtBGmQhthKhKBMBXaCFMRr29mzZP+/R+Spo3nSVqKSE6mSHbGdqmStVZq1/hVGtT7WPKavCqt  
Wj5t6klp1vQlqW9uq1l9vITLjDT7yicJyXle+nX90tZtOgn83qT1wLwRpgKXwhToY0wFdoiU6GN  
MBXaCFORCISp0EaYivjskNdenSc1qrwmqSm/SVaaDUVF9m4zQk44YYBccslAGTx4oOTnDzKVb77P  
d763t18k5/S9UvlavyxZ6T9LdrqYvhM56IBZct+9T8qSJYuddQDRCFPhC2EqtBGmQhthKrQRpkIb  
YSoSgTAV2ghT4Zd9Hly48Du5c9gXkpPxp6SmbJfMtF2m/pQju90jg/JPI6FD800NkSFDBherK67I  
l0suGSyHHTZUGjd6VHKzvpSMIB3SqN5cOf3052Tu3CXOmoBohKnwhTAV2ghToY0wFdoiU6GNMBWJ  
QJgKbYSp8Gvnzp0yd+6nctONH0lO5hpJTdkpWWnbJTvjd2nd8lnpcthf5YjDrzd1nRzetXh1O+Jq  
OezQm6R5s+ekZrXZoSNTM1JE6tacJcf3fEK+/+53Z01ANMJU+EKYCM2EqdBGmApthKnQRpiKRCBM  
hTbCVPhInwf/+OMnue/eOZKT+Zvs1fRVOajjY9Kq5ZtSq/qSwo/828pKL7lsgLpP24lmv22o1K01

U5o0/ErOPWeMzJ+3zFkTEl0wFb4QpkIbYSq0EaZCG2EqTBGmIhEIU6GNMBV+2efBbdtWyFPDF5Sc  
zl9l332ulCO7nSttWj8ptaovCIWp4QtL7Sq5bJiaasPU0XL88adly+YPyYEdnplr//KwLPiFc6bC  
G2EqfCFMhTbCVGgjiTlU2wlRoI0xFihCmQhthKuKzS14Zs1Cq5r4inTv9RU4+eYgc0GGE1K21UDJT  
yw5T7ZGpaSkiB3Ucl+eff7G0bf24dDnsGbn77uGyePEfzjqAalSp8IUwFdoiU6GNMBXaCFOhJTAV  
iUCYcm2EqYjX+PFLpE7tV83rx+ul57HXS/Nmr0uNKuGP+ZcnTE1PEdlv33fNc+pQab7XS3LUkW/I  
o4+OMdu3AmcNQDTCVPhCmApthKnQRpgKbYSp0EaYikQgTIU2wITEa8KEP6Vundekc+e/ypFH3Cw1  
q30j2RmbTYnkuM6Zaj/S767IOVPtEawtm8+Ulw6/W5o0fl2OosqGqaNI6VLCVHgjiTlUvhKnQRpgK  
bYSp0EaYcm2EqUgEwlRoI0yFH/Y5cMuWLBj8+Qp57NHvpHatN6T5XtdL2zb3SrUq86VOrWnSqOGL  
0qjBaGIY39bLHjVKgtQbI3VrT5D6dd+Rxo3elfr13pYju70qw+4Yl59/PlvWrFnjrBEoQpgKXwhT  
oY0wFdoiU6GNMBXaCFORCISp0EaYCr/Wrl0rTz89Xk447iWpmj3D9Mx8adrkDTn99DPI0oHnymWX  
DZahQ/NlyJBBMniwR5nbhwwZKFdcCbGc1OtaaZb3tFnOXKlb6zM5vOuT8vhjb8r8+fOctQFFCFPh  
S8H6Aml/V3tJucg8yV0WUwNSpM/ThKmlz+pNq+Xg+w4O9ZNXj/V6opfs2EmYcV82bN0g3R7sVmKP  
9Xi4h2zeRpgK/7bu2Co9H+1ZYo91+WcXWbeFMBX+hcLUJ3t795jZR+t4T0dZuYEwFfEhTIU2wITE  
651JS6RW9TGmZ96Wvdu+IPn5Q+TyKy6VwYMHhyo/P7/UsoFrv34DpOth10pu5nLp3HGRPPH4i7J8  
OR/zhzfCVPhSYEZ7M8xTnOf0YwYQj9VmHGyGu6/co5cZO8wA/NpgRjcz3H3IHj3M2GwG4NdWM3qa  
4e4r9+hixjoZAL92mdHbDHdfuUdHM1aaAcSDMBXaCFMRr3Fv/Co1qr5memaS1K75ohxy8GVy9lk2  
KB0iAwfmy6BBg0wNNP8Ol/3e3jZw4CC55JJBcsEFA6RLI4FSvdpdZhlLpdMB8+XBf46QP/7gav7w  
RpgKXwhToY0wFdoiU6GNMBXaCFORCISp0EaYCr927twp3377kVx/7YeSk75KUINmSfbGp5KVvkZy  
MudltdZpP6tX6RenR+kQd2vpVH9/0qjBjOkYb0vpX7tOVKnXkKpXuVXM89nkpn+nnpnvQ8lI2Sq1  
qs2W7kc9Ypb9m7MmIBphKsrFfozMbaMZB5iRsss8ycUOc1tfMyJi5wW8xPbJdjMONaOkHrMvHiPo  
MZSHV58cbUZJPWZDsAh6DOXh1ScnmFFSjx1hhg3DLHoM5eHVJ6ebUVKPdTJjixkWPyby8OqT8847  
zzPksrX//vsXnvuZH0nfAwcO9OwwW61bt5bly5eHppqPHEMuGqV9+9b7831XTJCdjK6SmLJT9Lmm  
FkpWxhTJzpxqbt9ReEX/2MpKX23qZ/O9mSdzgk6UjJT1kmN3DlyRNcnZPas3501AdEIU1Eq+4RI  
r5BnzZ8/X5o2bRp+Yqtl6ltTJY2Rpsx0r454NTSvYgLFwuCF3eP/frrr6EdplCPVTH131A3eY/X  
TZnpRv57ZGheegwlcfeY/aiOfEX6rFMUx+Fusl7vG3KTPfUg0+F5rXLoMdQEnuxKWvFihXSqVOn  
cl/ZmhqJu/xgSkzzcPDHg7NS4+hNJEes1cVPvzww4t67M1QN3mPmabSUuTeG+8NzWuXsX379tD3  
QKxIj9kLNB577LFFPVaOuuWWW0Lzbtu2jR5DiSI9Zp/vTj75ZM9eKqmuueaa0Ly2v+gxFNkIO3eu  
kRHPfy/VcqZKasoqyc6w9VboKN0stO9MzZPs9DWh8DQ7Y5dT5vs0Cd2fmf6KZGVOkayM981tH0ta  
ylbZb+/ZcsP1j8rChfGFqTb/ty9D7DVtI6XRvI7rMZtjKCIJMRRT3u30vvPCC7LvvvsWeyFJTUIwI

vvn+O1MljBgmYuZr1KiRXH/99c7So9eF5OF+3F955ZWiyMtVoR6zgf0Xpkoa75hKN+War0GDBvLX  
v/7VWTo9lqzcj/v48ePlgAMOiOoTW6Eeq2q+/9hUSWOaqRxTrvnq1asnQ4cOdZZOjyUr9+P+7rvv  
ykEHHSSpqaanXL0S6rEs8/0UUyWNz01VN+War06dOnLJJZcUhqr0WHJyP+4ffPCBHHLIIZKWlhvV  
K6EeSzXfTzBV0vjGVF1Trvlq1aol/fv3l02bNoWWT48lJ/fj/tlInn0nXrl0llyMjqldit2vIrRo1  
asg555wTCv4teiw5uR/3r776So488kjZMyM6hW/PVatWjU544wzCo9Yteiz5ON+zD/99FOpV3dv  
SU+9SnKzxkpu9gTJTv9CMtNMpY+VrLRfJCvddVRq+kZz+0xz2xwzzQJTXlt5nINalafKOlpV0qV  
KoeZnj1cFi361VmDvx776SeRnBwxfVtUAWY4dwboV/Nj5uVFryc/37kzDo8/LmYfROToo8Nf+/YV  
+Z2DdUPMfzFQxB59etttt5k/vvATVYIPcD7CVPey+vXrJzNmzAi9e43k8vPPP8vdd9/t2RdRVVaY  
OslUTJjqXlafPn1CT6qRlXKRPBYsWCAPPPBAYT+U2GM+wIT3suwRFR9++KHYo6KRXBYtWiSPPPKI  
ZGVIFeuLqPIRprqX1bNnT5k2bZps3LjRWTOsXe/mlcq///1v82KuSrG+iKryhKl1TLnmcS/rqKOO  
ksmTJ4eORERyWbJkiTzzzDOhcN2rN+lp93lOO+wwееeddpPBYDkYc+IO3LkSKlfv75nb8RT7uXY  
NzTfeuutwvAeyaOgoEBGjRpV9OlZp6pUSTG31ZTatY6QrMx8ycy4T7LSn5astImSnf6bqdWmvje3  
3S0ZaTdJdtZ10qB+V6lbt47k5qZlWprtsaLI7bfffjJu3DhZtWqVs+bdjz2tcKtW0WHqZZc5d8bh  
uuuIl9msGWFqhPnvAlpceuml5o+kaMNRyVklU2OrY8eOoY9DIrcccUVnv1QrHyEqbFlj6zmCozJ  
5zrzrO/VD8XKR5gaW/aiCPb0FEgu7jcdSy0fYWpsNWnSRObNm+esGcnivvvu8+yHYuUjTl2tunXr  
yuzZs501l1nYN4S8+kGj7JGqM2fOdNaMZPH000979oNG5ebmht7gRnJ58cUXPftBo+yR+/bTSBF9  
+kShJj07hwPNZ54R2X//8G0332yP/C9+xOiVVzoLiWFFUjz6qMixx4rUri2SISXmtYbl1VeHl33D  
DSJNm4p5jRte5skni0QOzi4pTLXvMUSWWb9+eJnVq4u0bx/+OeyvFHvaAbvbeddddIlWqRC/PXfb3  
tct1HRyedMx/AyAyd+5cOeuss0IfYfXaeBSrAMJUu2N13HHHyWmnnRY6wouq3NW7d2859dRTpUWL  
Fp79UKwCCFPtR4DsOb/oseSoU045JdRjrcyehFc/FKsAwIR71FiPHj3k9NNP9/yZqMpVtsfs9qRt  
27ae/VCsAghTc3Jy5Oijj6bHkqQiPeZ1miXPCiBMtUdY24/f0mOVoi431dPUW6ZKGfaiZfv9up+k  
jDc98JapiYr1ZopkvJshR6w+lrRe98/hOeaY6mPqGFNevyP1v60TTNnHZqypUoZ9rDss6hDePtny  
6o2gyvRx+jvp0nVl1/L12E+m+prqYcrrd6T+t3WiKfvYvGyqlGEf646LO4a3YQnosdS3U6VdQTt5  
WOW5yHfJeWdGh4vp6SK5udG3XXutyMcfhwNQ9+3uMNUetP/BByLHHFN0f2Zm+N/PPScyerSIPVOI  
DU/T0qKXs88+9gjw8HK8wlQbmtrQ057S+uWXwzVksFHYaysjQ6RTJ5EpU8LLseyHVRySEjk6Vcx+  
SfQy69UTGT5cZNascPjrnAY5KZn/DkBC7+RFPkpWrgogTKWoUiuAMJWiSq0AwlSKKrUCCFMpqtQK  
IEylKmE9bmpPHZ+Yss/PXr8XtfvUP03tqeNLU7VNef1e1O5Tw0zthqOv9BYbpl4QE6Y2ahQOPb/8  
MpyvRNhQsnnz6GndYeq4cSltWoTDWHufDUyfeClcskbs3GIPISdy2mnRy7FHqJYWprou8xDljTei  
p7M1apRzZ4yrroqezgbDP/7o3JnkzH8HIDJ9+nTJzs42fyBmE1GelkyltIsWldluwIRKuwhTkeOi

TKW8ijCV0i7CvEq79rAw9dBdwx+pj1VWmPrii+GjQyP32VD1zTedO2Ncemn0csoKU0s6Z+rYsdHT  
2SJMRTjz34Fk574q3ZAhQ8wfidIMIFUBhKn2497PPvts6lTh9qruVOWusWar/aZ5ZjxxBM9+6FY  
BRcm2nMNDh8+XCZOnOj5M1GVq2yP2av324/6e/VDsQogTG3QolE8/vjj9FiS1Ouvvy4TJkwlRbH  
qx+KVQBhau3ateWhhx6St99+2/NnoipXvbaa6H9ovPOO8+zH4pVAGFq9erV5Z///Cc9VhlqtKmX  
Tf1kqoTxmhkTzbho0kWSscrpgT6m+ipW7xTJHZAr9/x0j7xthvtn8RzLTY01NcqU1+9l/W/L9ph9  
bOaZKmG8aoZ9rPPfz5eUM0wP2PLqjaDq1BTJ6p8lt8+9XSaZ4f5ZPMcqU+NMvWTK63ek/rc1xpTt  
sTmmShiRHrvioysk5UzTA6fH9ETQZXosvV+6XDvrWpkvs0K5Sd+AwlR7TtWzzxapVSt8X2qqSPfu  
ls8+K7JjR3ia+fPtdUeKLycRYepf/xo9nT3dwPffO3cmOfPfART517/+Vb7zDfolU91XXbTnZh08  
eLCsXbvWWTOSxRNPPFG+8w36CFPdPVanTh256KKLuMhZErJXJy7X+QZ9hKnuHrNXQD7//PNI6dKI  
zpqRLF544QXZf//9o3rDs+IMU+25xc82e9j2yu5lLvYF5QEHHODZF1EVZ5hqzy1uz5W6wL7aQ1Kx  
V6a2V0EP6urqJVXVqIVD50HkQnrJZ9KkSXLIIYdlenq6Z28EVfbiU8cff7x8++23zpqRLKZOnSpd  
unQJXRzKqzeCKnv++mOOOUa+dH2G/9RT0PGjh29w9RffhFp3Dh62tiQc+NGe3FTMdvk8lWn3NP  
atBApHfv4mGp00y15y+NPTdrfn74vlivvRY9nS2za+vp+efF/B2L2SeNnr5NG5EHhHApKHAmTEl  
vwHwZsNOr41JqHwemdqpUyfb5fEf3GlctVVV3n2Sah8Hpm63377yZlIS5w1lNn97W9/K9YjheXz  
yNTWrVvLQvv2L2AMGzasWI8Uls8wNS8vT+bbwxAAwx41GtsjheUzTK1fvz7BAwrZT1jE9ki8VbNm  
Tfnvf//rrAHJ7rnnnivsjaACfBuifmyv7gMYL7/8cmFvBNVjNqSd4r4yk4u98NkmTUW1ZYv9xK9z  
p4u9bfPm6Gm3bXPuNNNavDweu9qPz9j0n+9WGsYK+3Y+O609Z+qqVSL9+kUHmgccUBSmlrUeN3vE  
q3s6W5GjYGPZ5dr77O/rXr79t73d63dOFoSp8BT56P9281c8ZswYGTlypNx3332SmZkZ3rhkmppl  
NlRmmO+iRui2EeY7M93gCwaHNmz2SDF3wOU+tQCSU6QH7Ff7scYRI0bIAw88EH0htBml9Nhr5jsz  
zUVnXySjR4+Wp59+WhYtWhRapkWPwd0Db7zxhz//PPyyCOPhF7gFfbYB6X0mL16p5nm/NPOD/XY  
U089ZXZ2zN6Ogx6DuwfsKSb+85//hl6+t0FVYY9NKqXHTP/Zac4+4ezCHnOHqPQY3D1gP4ZvAwl7  
+hp7GpvCHhtbSo+Z51E7zandTw3tj9kemzNnjrNEegzRPfDuu++GTsFI99vt6bgKe8yUVzhvu2E  
E04o7LHZs2c7S6THEN0D77//fqj/7PPI3nvvXWl/ed3Wo0cPGTVqVKjHvvrqK2eJ9Biie8BeC8b2  
mH1t2b59+xl7yeu2ww8/XF566aVQj82cOdNZom6P2QtL3X130cf8bdkjUE86SWTAAJELLxTp2VOk  
bt2i+22dfHL4QlI22MT/hnkYgLLFbkC2mXGgGSm7zIYndpjb+pkRwRMcysOrTw4zo6QeO9WMCHoM  
5eHVJ93NKKnHjjcgh5DeXj1yYlmlNRj3cyloMdQHI590seMknqssxm7zLDoMZSHV5/079/fvHg3  
PeVRNqzYYg/JMugxldXn+Tn53v2l602bdoUnhqOHkN5ePXJlVde6dlftpo1aybLli0LTfe/6jEb  
itoDrW+8UeT000UOP1ykc2eRTP1Ejj7abodF7rhDZOJEkQ0bnJnwP0WYCl8KzGhvhtn8eA67Yw/E  
Y7UZB5vh7iv36GXGDjMAvzaYYcMsd1+5Rw8zNpsB+LXVjJ5muPvKPbqYsc4MwC8bIPY2w91X7tHR

jJVmAPHo27evZwBhq127dvJn5HOMgE8DBgzW7C9bLVu25PRKiFtpzDk9ErwgzAVvhCmQhthKrQR  
pkIbYSq0EaYiEQhToY0wFdoIUxE0wIT4QpgKbYSp0EaYcm2EqdBGmlpEIEyFNsJUaCNMRdAIU+FL  
wYoCaX90e0nJNhughjGVISJ9LiFMRXxWr10tBx9/clk91uvcXrKjpMsOAuWwYeMG6XZqt/AV1z16  
rEefHrJ5C2Eq/Nu6bav0PLtniT3W5aQusm49YSr8s+d2631Bb+8eM8+fHY/tKctXEaYiPoSp0EaY  
Cm2EqQgaYSp8KSgoKHZ1PHf16UOYivisXr1aDj74YM/+stWrF2Eq4rNhwwbp1q2bZ3/ZsleN3byZ  
MBX+bd26VXr27OnZX7a6dOki69YRpsK/UJjau7dnf9nq2LGjrFxmIrl4EKZCG2EqTBGmlmiEqCF  
MBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQRpkIb  
YSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgjTEUi  
EKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRo  
I0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyF  
NsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiK  
oBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp  
0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghT  
oY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0w  
FdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGmlmiE  
qfCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQR  
pkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgj  
TEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2  
wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8  
IUyFNsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQ  
RpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFOR  
CISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV  
2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhT  
oY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbCVGgjTEUiEKZCG2EqTBGm  
lmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgjTIU2wIRoI0yFNsJUJAjh  
KrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQgaYSp8IUyFNsJUaCNMhTbC  
VGgjTEUiEKZCG2EqTBGmlmiEqCFMBXaCFOhjTAV2ghToY0wFYIAmApthKnQRpiKoBGmwhfCVGgj  
TIU2wIRoI0yFNsJUJAjhKrQRpkIbYSqCRpgKXwhToY0wFdoIU6GNMBXaCFORCISp0EaYcm2EqQga  
YSp8KVhaIO2blhKmHkeYivgQpklbhvUbpFuHUsLUzoSpiE8oTD20IDC1XRdZt5YwFf6FwtTupYSp

LTrKyuWEqYgPYSq0DTizlDC1dktZulAwFfEZ3L+UMLVKnsyfS5iKiiFMRcXtEikwo70ZZvPjOfqY  
EZkW8IMwFarMtmmdGd3MMB3lOXqYsdekMtmPwxfTNVjN6muHuK/foYsY6M+gx+GL6ZpcZvc1w95V7  
dDRjpRn0GOJBmAo1zrZpgBmmozxHSzMWmsF2DL6Z3hlshruv3CPPjPlm0GOoCMJU+FLuMBXwiTAV  
2sodpgl+ITtMBXwqd5gKxIEwFdrKFAYCcShXmApUAGEqfCFMhTbCVGgjTIU2wlRoI0xFihCmQhth  
KrQRpiJohKkokT0PI5v996effirTx0+X16a/Ji3WtyjcAMWOrn90ISmTp8jmlIdFBROwykdy8+uHz  
zz+XDz74QMaNGyd777235467rU6dOsk777wjGzdudOYMo8fg5tUPM2fOlGnjp8mEqROk/eqS3xTq  
sLyDvDXlLdnw5wZnzb4e2zj1i2yfvNmerUS++KLL2Tq+Kky6f1JctDKg1xdFT32WbWPvDHIDVm3  
ODpQpTdQlq+//lren/C+TJ4yWboUdHF1VfRouaalvPL+K7Jm0RpnTqB8Zs+eLVOmTJH3339fjj76  
aM99MVvNmjWTUaNGCaEzVNH3330nkydMlImnvTZPjlhzn2nJFj4YbG8rz056X5T8td+YEymfu3Lky  
+a3JMv3d6Xly7ye7uip61N1cV56e/rQsm7fMmRMoG2EqSvXzzz/LXnvtFdpZysjlkNTU1PDOU31T  
30U2Px5jrCkzXXpKeuirne/VV191lgoU+f3336Vt27bFe6yclZ4e7jFbI0eOdJYKFFm2bJm0bx++  
YF5mZmZRj1U19bGpksY0UzIF2zFbTz75pLNUkXUbNsje+fPI7UmTZOSLL8rwp5+RJ02NGjNGpn/8  
kSxdXiBbtm+XzVu3hk7B9NOCBfKfRx+XR1p1kCfbHypjX3tNfjM/m7Vj165QbTe1dccOWb9xoyxb  
sUIWLfoo3/3wg3z59dcyc8bn8sG0aflWxLflbFjZbTZpr4+bpxMmTpVvp41S37/4w/ZxtHa/xP2  
SPronTsX77EsU1NMlTQ+N1UtusceeughZ6IAefvG4RFHHFG8x1JNTTBV0vjGVJ1wj0Xmufvuu52l  
AkW2m+erY489tniPlbPS0tJCZb+/5ZZbnKUC0U455ZRQjxTrsRGmShoLTDU3PZaSVrjff8011zhL  
BKKdddZzHT0W2SaFaniom7zHYINtwz1mX4/a6S+77DjNiYA3wlSE2KNglkCrF27NrTxaNCgQeHG  
p9gOVTnD1NjKysqS0047Tb755pvQuiyOwEekcd606ZN8te//lUaN25c2BsV3WkvqexOlt1Rmzfj  
RmhdFj2WPCKPtX1ReN1110nTpK0Le6NYj5UzTHXP09hU/5R0+axWHfkzp4osT0mV1SmZ8ltKtsxK  
yZUvU6rKRynVZZKpcSIV5LWULHndTD/BzPeZqZ9MfZBbXYbXbyTXNWshl7XdVy7f70C5pn1nuaHD  
gabay9X7d5CrOnQ0t3eQi5u2kGvMNA+afz/del/5pFZ9WWV29LaZ9e40tc18v8HUMvP9z6Zmp2bl  
F1Wry8y8ZvLFPvvLx695aF77pU5838M/b8gfu7tyU033STNmzcv7I9iPVaeMLW6KY/5e/bsGToi  
LlLtWPJwP9a33367tGrVyrNHQIXOMNVr/u7du8vbb7/trIkeSybux/qee+4pfFM7tkf8IHv+bt26  
hT5pFEGPJQ/3Y/3AAw+ETgfh1SOFVY4wNTKte/6uXbvKK6+84qyJHksm7sf6scceKzxwlrZHCqus  
MHVvU8607vkPOeQQeeGFF5w10WMOqpiKKO5zInluhCLIM0y1FVluixYtpKCgwFkzksVFF11UrBeC  
rshyGzZsKlsXL3bWjGRh3wyK7QXPqkCYWi23ihzcah+5vWEreSenobyb2VDGm3rd1ISsBjlzq64s  
yq4tv2fXkaVZNWRuzXpy/eFHytnn95dL/vIXuchU36FD5JSLlpYzTz9b7ut4qCzIqSbr09JIRWam  
zKIWTF5bu7Z83KCBTG/WTN5v1VLeNdvINxs2kIl5TWVky9byQbPm8mW9+jKnek35vHYDmdKwqbyT  
11xmNG4q3zZoJD/WrSfLq1aTzdk5siMnRyQnW75NTZe/5tSQz1553fnfQRCuvfbawj4qtccqGKa6

K7Lc3NxcmtNnJrNmJltbb721WC94VgXDVHdFlmuP3LGnDUBye+++4r1QtDlXq49VRiSy6OPPurZ  
C55VgTDVXe7lut+ARHJ49tlnPXvBsyoQprrrLvdy33nrLWTNAmArD/e7K4MGDozYeJVYcYWqkJro  
lFm/fr2zZlQaZbxZd9WQqzz7QaPatWwnK5eU4xxevMFYqfztb3/z7ldiVc4w9dCUTLkyo5b8K6OZ  
jMhsKmMyG5lqLB9m1pNfs6rLxpwqsi0nV7baADMjSyY3ayEXHXuCXDLglskfOjS0XXWXDVf/duqp  
MrZOHVmSnS1iyoaf5SnJyJTV1WvKDYcfLmf36yf5Q4ZlvmmrUGXDpRzeveRh/fvaH6WXJGslPnF  
/J5PpmXLT6+/4fzvIjAjDhg3z7qnYiiNMjVReXp78+uuvzpqRLOyRXF79UKziCFMjVa9ePZk3b56z  
ZlQaZezbPPHge579oFE1smrlrBmznDWXgv2xSuX555/37AfP8hmmRsq+8Wivu4DkMmbMGM9+8Cyf  
YWqk7Mf/7XU9gAjCVIRMMzatcENR5rs6tulMU8u1DmrPq7dNIWOkmuH+t8aIWseuwu9KHs+Z8vqd  
qD2yyrWNNKWeYem5KrowzLwR/zKomG7JzQ+GnZGfJLIM7zPfbRhqv2ZlyZa0NBm9bz5+bz+cuGg  
QTLEBp35+VF18RVXyl29e8vrtWvLYjOPDT3tMspTkp4hK6vVkoU7dpWz+p1TGKRGln2hXfZjp8j4  
GnVlVva2rMrOkZfNz93e6/en4qpy9VicYsrPlcd5Xr84wxT6bFKWv8yVdpw9ot2y/2xD015/U5U  
5a44wls2Y1S5Ko4w1d1j9tzSfNQfFmEqQqZPn154suVyVQBHplKVRDJMvWtqTx0vmPL6vajKW+UM  
U89KqSKvZtaSnzOrS0FWVVmVXUW2ZufIuzsqLjh6ra0dBm9z75yUr9z5cKBA1XC1FWlhKn9L7tM  
bjqxl7xXrbaszcySFVm58mJWDTk4LdP7/4DSrQCOTKWoUiuAl1OpSliPm9pTxyem7POz1+9FVd6K  
88hUiiqz4jwyNVK9evUiTEUIYSoKbdiwQc4//3zPjUax4shUyqv25CNT/2PK63ei9sgq1zamnGFq  
m5R00Te9mjyekSdzM+uFzok6OrO+jMhsJGNNTTS3zcmsKVuyqohkZMjHjZrI5V2PloH9L5RBQ4dG  
Bam2gjgy9drDD5czzz1X8ocOkfzLhsqIV1wul5hIdzryWHm+SQvZZqfNypTF2bnyRGZV2cfr96fi  
qnL1GEemUnFUuR5/jkylvOohU6WN8uwXBTQqvD/2kSmv34mq3MWRqZR2BfAx/1dffdvJTgDCVMQ4  
55xzCjcYpT4xBXABqtatW8vmzZudNaPSKOONusHnFp2XNzVFZ+cnsty8anmy9re1zppLwZuLlcrV  
V19d1AulbcfKGabaaTOysqVpwy5tO5eMrpqE5mZVUc+qzwjb2XWl4lZjeWVrCYywtQrmXkyKb2+  
TMitL39td6icdVxvOe/sfnLhmWfLkN6ny7XHnSi3dj1KntynvUyu31A+bJln09q0kbf2319eO/BA  
GdW5s7xwWBcZ0a2bPGW+/qPtvvLogR3luSO6yFBDu8hbLVvLt9VqyA81asvo+k3lb03aynu1G8m8  
qjVkQXYV2WJ+ztBpCOz5VbMy5bvUdLmkZgP576T3nP8dBOGWW24p7KNSeyyAC1BVRvPvFi1a5KwZ  
ycJeYT22FzwrgAtQ2ReIP/74o7NmVBpl7Ns8MuyRol5Q3h+zNfuj2c6aS8H+WkXyzDPPFPVCadxs  
WwFcgGrGjBnOmpEsRo0a5dkLnsUFqBAwwlQU8/vvv8thhx1W+gYpjjC1WbNm8vbbbztrM/tNHcaf  
NCKPdUFBgrX55JGhKwh79Ui81aRJExk7dmxoXRY9ljwij/XatWtD5zQqtccqEKZGKs1U1/Qcmdas  
jRTkNZfVtWvL2mpVZU1ujqzOyZZVppbnVpVl1evliroNZU39hrKhXj3ZWl9+aNpfq1aTibXqyojG  
zWRM673l9bbtZFzb/WVcm/byZpt9TbWRV029aOrZ1m3kyRYt5T/NW8j4vGbyVe16srhaDVltlIFQ  
paqsqFJFVpuvBdWry5+1asmfdc39jZrKnGatZHLr/eXfBxwqj1x4qbz55nj5w/zNIRiRHRNvBp50



0kmSnp4e1SNRFUeYai8KZK9SG8F2LHIEHusdO3bl6aefXvppmOIU2ubbdLjjz8eWpdFjyUP92Pd  
r18/yczUORVMzZo1QxdTi6DHkof7sR4wYIBkZWV59khh+QxTq5t9oDvvvNNZEz2WTNyP9ZAQyQ7  
O9uzRwrLZ5hq39S++eabnTXRYyhCmIpSTZ06VaQYF+zuDUroXeYywtTUN4oHsQ8++KDs3LnTWTIQ  
9sknn4Re0LI7pcx3Fkuou+++W7Zv3+4sGQj773//Kw0bNozqldB2rKww9QNTMWHqLbf8XbZs2Spb  
tm2TTZs2ydpVq2VtwXJZu3SprP3zz1CtWbZM1q5YKevWrpX169eHyk679M+l8tm0D+Sus86TW049  
SyaOHY+LfvtNtm7eLBvMNO5av3adrF2zRtauNstftUrWrVwlg1aulPXLV5j1Fchau45QFcg6e5u5  
f83qNbJu/QbZtGWLbNuxgwn8EmjWrFmhNwrdrVLqsbLC1BmmYsLUa665xvTYFmfJQNjcuXOIbdu2  
Ub0S6rGywtRZZrq60c+pl19+uWzcuNFZMhD2888/S/v27aN6pbz7Y7HTXXLJJaHnPsDNfsrikEMO  
ieqVwqOXywhTU1tE99h5550na8x+EuC2ZMkS6datW1SvFPZYGWfQ6j7RPXbGGWfISrPvDZSEMBUI  
in3Xxf7bHsFw+//dLjc8clMOWNagaAMUM9rMbiPDhg2TZb8uc+YO450cuHn1w/Dhw+W2226Tm266  
KXSEqftJzV0tWrSQW2+9VRYvXuzMGUaPwc2rH5577jm59f9ulVvuvUX2+m0v15YrejT7qZncNOwm  
Wfj9QmfOMHoMbl79MHLkSPn71X+X2+66TVotaOXqqujReGFjuf7O6+Xnb3525gyjx+Dm1Q/2o423  
XH2L3DHsDtln/j6urooe9RfXl6vvulp+mPmDM2cYPQY3r36w5wa0pzSxR/3FhqzuqlOnjlx55ZXy  
7bffOnOG0WNw8+qHcePGyc3X3Cx33XaXdJzV0bXlih41V9aUy/5xmXw99WtnzjB6DG5e/TBx4kS5  
+dqB5e6/3y0Hf3mwq6uiR7U11WTQPwfJzHdnOnMCZSNMhS8FZrQ3w70Rco8+ZgDxWL16tRx8sHnS  
89hxt2WvpLhJxw5naqDiNpjRzQzTUZ6jhxmbzQD82mpGTzPcfeUeXcxYZwbG1y4zepvh7iv36GjG  
SjOAEPTt29dzX8xWu3bt5M8//3SmBPwZYIbpKM/R0oyFZgDxGGyGu6/cl8+M+WYAFUGYCI8IU6GN  
MBXakjZMveUWkXr1RFq1Cn899lgxr4SdOwN0223h5bdsKVK/vkj37vbzV86dPpR2BMpuenQKYSq0  
EaYiEQhToY0wFdolUxE0wIT4QpgKbYSp0Ja0Yeq4cSJXXSVy443hrw8+KKJx3rHLLjN7GWY3I1I2  
vl33qvTvvRe9TFv/+ldz5+6HMBXaCFORCISp0EaYcm2EqQiaeRUCVFzBzgJpv7K9pCw1m5+CmDK3  
9VIDmIr4EKZC24ZdG6Tb6m4lbsd6rO4hm3ftlWHqZvNzLltWVPbq/Vu32suBh4NS+2/71V4E8IsV  
RF54QWTMmPDxt98WkelINnu2idiLiNgT8NtILF8eXo690Ju9b9Wq8H2RsuuM8ApTf/stfN+6deHp  
7fJsrVghsnatIL3wUuxRpvbf9nb7s7/+ukhOTvRy77or/LPY32E3uwBdKExd3bPEHuuyqous20WY  
Cv9CYeqa3iX2WMdVHWXITvO3BsSBMBXaBqwbUOJ2rOWKlrJwB2Eq4jN43eBwjy0r3mN5K/Jk/g7C  
VFSMeRUCVFzBsgJpv2/JJ6PvcxphKuJDMaptGzZskG5doq/46a4eR/aQzTak3BOMGBEdMnrq3du+  
AhapWjX874MOEvnmg5ELL4yezH1yRsyelXlLddSL77y+SnR09va3DDxc54giRjlzo2204GxEbplar  
JnLKKeGfw317pLKyrJo1E7n+epHPP3cWYtiw9eWXRvq3Lpo2NbWoIrcNHBg9325g69atOrN7T/Pj  
efdYI4O7yDobLAM+2Qtu9O7V27O/bHV531FWriBMRXwiU6FtQP8Bnv1lq2WzlrLwV8JUxGfwwMGe  
/WUrr2GezJ9HmlqKMa8+glorKCgo9cqeffoQpil+hKnQFgpTu5USpvbYg8JUG2KanzmqqlQR8ypX

5LzzRC64QOT++8PnLL300ujp2rYV+f338HLs0aGPPCLmjoy/6/htuKPqlvj0C1E7TpUv0cmYNHbme  
xooNU2vXFrnmujA0x6d+uKL4UDXPa39eb3Yo2pzc6OnfeYZ507HbnT+1FCY2rOUMNX8HxKmlh6h  
MLV3KWFqx47mz4wwFfEhTIW2AQNKCVNbtPSCwITEZ/Bg0sJU/PyZP58wIRUjHkVAIQcYSq0EaZC  
W6UOU+0Rmzak9GJ2JqOmdYepNmW98MDo+y+5JHxfrFdfjT4y1FZpYarXEbCW+VuXzp2jp7VH1HqZ  
OrV4mGqD3d0UYSq0EaYiEQhToY0wFdoiUxE08yoEqDjCVGgjTIW2Sh+mTjpg3BmjtDDVviA2v7ek  
pRXdb4NNe7/7iE/7//Lkk9HT2SorTPW6AJU972o8Yeqjjzp37n4IU6GNMBWJQJgKbYSp0EaYiqCZ  
VyFAxRGmQhthKrRV+jB1/Hjnzhilhan2Qk+ffCLsv3/0svbaK/wR/eHDwxd8sh/xjw01bWmHqdOn  
i1SvHj3tiSeGP+r/6afh86vuRghToY0wFYIAmApthKnQRpiKoJIXIUDFEaZCG2EqfFWqMHXUqOij  
RO0V7996y7kzXuWXh6fJzAx/tReZWrzYudNh/7Z++UXklvCwWn9+uELWdIA1J7PdM6c8JX1S/uY  
/5VXRq9n331L/pi/XYedxl6Eyn6153n1smIT0cWxzHOQ1KsXXn6bNiL33isyd64z4e6BMBXaCFOR  
CISp0EaYcm2EqQiaecUCVBxhKrQRpkJbpQIT3R+/j1XafV5ssHnTSLHHCPsqVP4/KnXXivYww/O  
BMb27eHQ9Pjjw8GnLRuqNm8u8uabzkSlqOjPtAcjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOhJTAV  
2irVkalBsUGn+X+RESNEjjxSpGnT4udFjVS1auEjVe10t98ePIXAzp3OgmARpkIbYSoSgTAV2ghT  
oY0wFUEzr4aAiiNMhTbCVGgjTK0gG5QSIIYISq0EaYiEQhToY0wFdoiUxE0wIT4QpgKbYSp0EaY  
Cm2EqdBGMlpEIeyFNsJUaCNMRdAIU+ELYSq0EaZCG2EqTBGmQhthKhKBMBXaCFOhJTAVQSNMhS+E  
qdBGmApthKnQRpgKbYSpSATCVGgjTIU2wlQEjTAVvhCmQhthKrQRpkIbYSq0EaYiEQhToY0wFdoi  
UxE0wIT4QpgKbYSp0EaYcm2EqdBGMlpEIeyFNsJUaCNMRdAIU+ELYSq0EaZCG2EqTBGmQhthKhKB  
MBXaCFOhJTAVQSNMhS+EqdBGMApthKnQRpgKbYSpSATCVGgjTIU2wlQEjTAVvhCmQhthKrQRpkIb  
YSq0EaYiEQhToY0wFdoiUxE0wIT4QpgKbYSp0EaYcm2EqdBGMlpEIeyFNsJUaCNMRdAIU+ELYSq0  
EaZCG2EqTBGmQhthKhKBMBXaCFOhJTAVQSNMhS+EqdBGMApthKnQRpgKbYSpSATCVGgjTIU2wlQE  
jTAVvhCmQhthKrQRpkIbYSq0EaYiEQhToY0wFdoiUxE0wIT4QpgKbYSp0EaYcm2EqdBGMlpEIeyF  
NsJUaCNMRdAIU+ELYSq0EaZCG2EqTBGmQhthKhKBMBXaCFOhJTAVQSNMhS+EqdBGMApthKnQRpgK  
bYSpSATCVGgjTIU2wlQEjTAVvhCmQhthKrQRpkIbYSq0EaYiEQhToY0wFdoiUxE0wIT4QpgKbYSp  
0EaYcm2EqdBGMlpEIeyFNsJUaCNMRdAIU+ELYSq0EaZCG2EqTBGmQhthKhKBMBXaCFOhJTAVQSNM  
hS+EqdBGMApthKnQRpgKbYSpSATCVGgjTIU2wlQEjTAVvhCmQhthKoJiwwYbam3atEXWr98i69Zt  
kZ07Rf74Y7V07XKM01NVXJUbuu2oo46T5cvXhJaxefPWUL/ZZQHIRZgKbYSpSATCVGgjTIU2wlQE  
jTAVvhCmQhthKoKwfpICeWPsR9L100+lZrVFUrv6Mqma86fkZP4p2enLTa2R7LQnkPW2ydRG8+/1  
pLZLVnqB5GT8IdVz/5Ba1VZly73mykm9npcpU75zlgyUjTAV2ghTkQiEqdBGMApthKklGmEqfCFM  
hTbCVATHjXefSvfuj0vt6isk3Tzj5WTukqz0DVkz2k/SLG+y7NP2JTnggMekc+f7pdNB/5QO7Z+U

1q1elwb1pku13F8kJ8POI2bezabv5skN130pS5b8YHpvu7MGoGSEqdBGmIpEIEyFNsJUaCNMRdAI  
U+ELYSq0EaYiPrtk166NcustX5h+mSSZqetMiVTLXS+dO90qZ53VT4YMyTc7VvmSn1+8rrjiljn9  
9GulXp1XJDtjsWSaZ0tbR3ebIk88MUI2bNjorAcoGWEqtBGmIhEIU6GNMBXaCFMRNMJU+EKYCm2E  
qYjH1q2b5JtZn8IFF84y/bJKMtN2SXrKNqle5Xs5tfdVctllZ5salkOHDpYhQ4rXFVfky3nnXSqd  
DrpEGtb/I2RnzJSs1F3SrMIHkp//kulPwISUjTAV2ghTkQiEqdBGmApthKkIGmEqfCFMhTbCVMRj  
y5YNMnPMVLngvK9Nv2yQjJQdkpW2Qarm/CitWrwoHdo/JAce8Lgc2OExOaB98ep4wMOy797PSv26  
U8w8C8y82yQrVaRxg/fl4otGyKpVhKkoG2EqdBGmIhEIU6GNMBXaCFMRNMJU+EKYCm2EqYjH9u1b  
5ZdffpD8gT9LesOPcmCHR+Xgzg9KqxZvS/Xc1ZKVfJ4XanbGrhlr00yTnS7StcvD0vPYK6RmtW9I  
79bvY7VXT5A1azY7awJKRpGkBYSpSATCVGgTIU2wlQEjTAVvhCmQhthKuKxa9cOWb9hmVxz9VxJ  
TRknHQ/sK506XSjN8kZKtZwVkpVeVpgqocA1O32HdGh/h3TterLkNR4uRxz+sNxzz39k3TqOTEXZ  
CFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMRfx2yG23fi9Zmc/KsccMkuOOu1b23Wes1Ki6qtxH  
pmalb5djejwhZ545WPZqOkJ6nThcnnlmtGzcsMlZB1AywlRoI0xFlhCmQhthKrQRpiJohKnwhTAV  
2ghTEYR77pkn1as9L8ccc5UccfgwadzwPamavTb08f3yhakihx7ynPTs+X+S13i8nHH6KzJq1Fuy  
aRMf80fZCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMRRDuvfdHqVb1P9L96IHS8cCHJCdzkdiP  
8LvLBqZeFTIn6r57vy2dOz0gjRtOlDNOHyOjRk2QzZu3OGsASkaYCM2EqUgEwlRoI0yFNsJUBI0w  
Fb4QpkIbYSr82r59i3lht1ymTJ4v/c6ZIVVyX5fWra6X5ns9IXXrfCatWr4g++79hOzddrjs3Wa4  
tG3zZLHa235t/bS0aj5SGjcaL/XqviONGO6T7keNIhtveF3enviZ/PHHMmeNgDfCVGgJTEUiEKZC  
G2EqdBGmImiEqfCFMBXaCFMRj1WrVsvIkWPlhOPGSkbKTEINWSjt9nlOzju/twzKv0iGDBka2qnK  
z88vsYYMGWQqX7p3v1GaNHPjCjKWyl55k+X885+SDz74XNasWeGsDfBGmApthKIIBMJUaCNMhTbC  
VASNMBW+EKZCG2EqgnD/P76X9JSXJStjkhx26KPyI78MIKFDB4V2qMoKUwcPtoHqYDnzzPOI4wH3  
SXb6djnx+C9I9Muvy5YtfMwfZdu6c6v0nNxTUP42262RMWVu6zKpi6zbRpiKMuza5XwT46efRC64  
QKbWry9Pm+ffkTFlb3urc2fZ4jV/ScsEPBCmQtuATwZ4P1c+kyItx7aUhRslUxGfwZ8PDvWTV4/I  
vZYn89cSpqJiCFPhC2EqdBGmlj42KNgid9w+2/TLG6bek3p1RshBHW+R0067Wi64YKhcNOBieSS  
C2XgwAtk0KBw2e8vvniAXHjhpdkv3xA55ZSrpE3r6yUrY6RkpGySY47+WJ5+arRs4Gr+KletZvQ0  
w2y1PEcXM9aZATou+9EZs0SmTJFpGFDs+dudt0jIEhcthhMr9uXfnE/HtGTH2Slibfmq/bTUXN  
d+65It9/LzJhsjixc6KgJIRpkLbADNMR3mOlMySNAOIx2Az3H3IHnlmzDcDqAizRwVUHGEqtBGm  
Ih6bNq2XadMmSN8zvzL9slyU6dLdsbXplZITsYsyc36SarmbJlq2atM/SIVc3839Zu5ban591qp  
krVVcjKXSlb6B5KVMVEy0/8rWam7pE6NKdL37Gdk5coNzpqAkhGmokLcR4s+/HB0AGqD0/ffD9+3  
dq3ImjWhb+0cp5x6auh50asO7NBBVqK+HrtxY/jr1VdHL7tRI5EffwzfB3ggTIU2wlRoI0xF0Mwe

FFA2e04ut23btsmBBx7ouVNIq1+/fs6UxecFvHj1yWGHHebZX7ZONS8el+gxxLJh6gcfvCV9z7Jh  
6ibJSFko2ek/mvpesjPelOzMLyTHdUX/6NoIWekLTc0x0y0w9ar53ITqDqlb8wPpd87zsnKIE0oA  
LI7bohPNSNIltluxw9zWzYwltmNJyv24v/GGTa1EHnlE5M477QneRO66S+SLL5wJzOQ7dzrfFbFv  
YHs9T9rq3LlzYW+FvrrX98orltdeKzJkiMhjj4n87W/2xIU7mCMvkw6Xtui/v37e/aXLXtwReTU  
N2zHUB5efZJvRknPIW3MWGuGRY+hPLz65EozSuqxZmYsM8Oix1BehKkolT3y74cffpAFCxbIO++8  
I9WrV4/agUpNTY36d+xtDzzwgCxatEhzm54t69evd5YKFLFPWPPmzZNffvIF3n//falbt26J/eR1  
213mhWakx9bao3WAKJ2ybdtaGXb7F5KbOUWy07ZJVvoisUFqVto3puaa+k6y09cXBqihSjffp+2U  
zPQPJTnljGRlvmvmmmy4ZKXNNbZeePT6Q554dJR53xvcxf3vNocind03rhmrVKufOANnNbux6uBaN  
jp9++kl+/PFHmTFjRuhIGYXbrHfMNssM813UCN02zXxnprl28LWF2zEuFpRk7NGio0eHjxC97DKR  
b78VaddOZNgwZ4li9nnSXIdjyy+/DB0NWNhjpsp6rrziiitCF3D51ix/xYqYi+fZgPaII0Ty88Ws  
QKRxY5H27cXsANp3z52JkAzs/r7dJ/vmm2/koIMOKrGfvG679NJL5ddffw31mP0EG+DFbofsa8vv  
zM6JvQhjYT897TwwxozQbfPNd+kpcv5J54e2g3bepUuXOKsEov32228yd+7cUHXv3r2oxx4ppccW  
mu9yU+TMHmfKzz//LN9//7388ccfzhIBb4SpiOJ+J+bOO++UtLS0og1QAHXaaac5S+ddn2Tlftxt  
2J6RkeHZK37rhBNOKPz4Pz2WnKJ77B+mL2pKZsaVUjX3eamSO0Gy0z+RzLTJkpk+XrLSCiTLBqhO  
kJqV/qe5/T1z+2+SmbpAMlKflepVR0nNGo9KRvoxZlIvPvU3LrJpU/jIVL89NmFCODtx15NPOnCG  
aPJkkXTzewW5ns2bRc46K/yp4/r1w1/tmV22bnUmSALux/25556TnJwc83/rsU3KMjXFVEnjc1PV  
TcXM17VrV1kT+Rg327HKx/2Y2iNDzWMeqv/7v8I/JPej/vLLL0vVqIWjesRdqR63IXa7rU6dOhUG  
XIEdZI5EhsJcM03oj/uTT5w7UNm4ty3jxo2TGjVqFOuTeMoesWpDDSQvd4/Zg3Jq167t2SuhGmGq  
pLHAVHNTMfPsvffeoeALycvdY9OnTzf7pfWL9UIhDTdV0lHsam9TMfO0aNEiFKwCXsyeElBk4sSJ  
csYZZxRuQLzehfZb7mUdeuihMnz4cNkYOX8XksZ7770Xde4trR6zR1Q89thjHBGdhKZOnSrnnXee  
qy/CX6tWTZF69WpKldzOkplxgam7JSv935Kd/rapeaZ+MfWOue1GyUi7WnKy+0ntWi2lWrVsycwM  
LyeyLFsdOnSQhx56KHR+34qyB4Jt2hQ+KC1S27c7dwblhqmZmUVZjS2z6Y1br17Ryzz+eOeOJPHR  
Rx/JgAEDCrc5JW7HfISp7mXZow/vv//+4kcSYs83c6aYBzv8B2S2JaGNgMtnn30WOtlv8oaj7Ys0  
py8i1dLU/abumfWgqdCyTK029ZCpf5n6iyn3PLbSXD3Wpk0bueeee2TZsvDHGwuNG1f0B37ssc6N  
qEy++OILGTx4sGRnZxf2Q1DI3o7ZMOKOO+6QUUuWOGtGsrBHOF9++eVRbwaV+HxZwTDVvZymTZvK  
3//+d8L7JGSPUr7qqqukZs2anr0RVRUMU93LadSokfztb38LHRkNRJg9JKDIwIsHRm1ENougDgfJ  
6uUVDyGwZ7vysis9+OGj9m27ryxbEvMCEZXdDfc4NkPGtVqr1by24Lwzrv9NFCzZkX5gy17al/7  
MftBg8S8YA3f9p//iLz0UvR0trxCTns6gEmTRIYOFWnZsmhae1Fve3rDjz8WOeYYMS9UwteQseu4  
+GJnZsMrTL3vPpGvvhK56ClxL3KLbk9LE2nbNvxJ34kTRTZssO/4h5djg97PP7fnzSua3uxjRn21

ZT8pPHKkPUVMeL7K6vbbbze/r3dPRJXPI1PdIzeXFzqFACoR+wc1dqxltWrhP5xep4gURJ/n4/57  
7i/WC7YGmTLfRNUuj9u8bv/DVDNTscusV6uefD875sibCWbDY7Zvdj7p3j18xCrXfKxUHNvosWK9  
oFU1qtSQL2d+6awZyeLZZ5/17AfP8nFkqrtyc3PIE46kTzqjRo3y7AfP8nFkqrvm5tTpkkx1gyY  
XSTnKyq5zWacBUZ1M+wJlmPHXs6ourJq+JwhvymXWUfmn5mStzNPmpvh/lk8xyhTDUw1NNWM2u2q  
rqmWpqabKmFEeqzaqmQJ67E/MqXJjibl67GxphqZsn3m9TtS/9uq53x911QJl9J1VdXT1iPpf6R  
KtW2V5On5Z+yfqnlAftFZRehoNEdNtp6/nn7sd1weOm+3bzmKGQvvv3EEyJNmoTvs8to0EDkuuvC  
5z+1Yegtt4Rvi13O6ac7CzG8wlQbup55psj48SLz5tkj4MKfNI58steW/XRv167hUyZG2E8f26Np  
7UW/u3WLXubhh9tzhooan++STr2XffdtJnTq1iz+Oe3jttdde0rx58/J/HDaAMDU9PV0aN24cWq/X  
z0TtOdXA9E9DU2PMi37z4lbqtsfqSA1pJXlm22WmCg37nFVrba3wdmyR2c6YCr3Q25UiZ5sXg5F5  
K1o/tEmRGj+Y5Wxxlulsx9IWp0mjbY2inivrSgtpubO5fHR80c868IO6Uv0HM01rM4V93vf4Han/  
cdn9mPqmRpgqZdjHuva62qH+ClXs81uQZXvs9zRpuLVhVi+VOL4wtY+pOqa8fkdqjy7nFWnTh2z  
6TA9UJ6KM0y1RxE2MDtFPFcmT9nHOvZaG6VWnGGqLRuo2k+NcAomWGbVCMlgkxIHm2E2A3vmeMqU  
xwaN2o0qw9S7pvbU8Ylpr9+L2r1qnKndcPxD/iYbPcLULI3su+YisZ+gtUebxp7L1B2m2mvP2KNN  
IOGsDVU/+KDoSNEIe62aAw+MXs4ZZzh3GhX5mP8dd0RPV7u2/Yiec6eLDVR79oye1h4d61zMOXQB  
pqDPvbfHVgBhKIX56jlt5ptQXfmlua0co/aqFFnSJDzPrtRwRZZRntqZVvt9w391L7nkkbktRSYf  
VzTf6R+ZW78xZfva/JvajetxU3vq+MRUVVNeVxdVeSvOMJWiyqwAwlRbvXv3JkxFiNk7QjKwYeqR  
ZphNwJ457MbPY2NG7UZIXqt0WHqSFNeVxe1e9UbpnbDca/c6BmmXnKJsxGOUVaYeuON0fc1axYO  
TmPZU2nawNY9rd8w9bbboqerU8d/mFraxXKSqghTKY961pT5JlR/edTcVsZldX1/9muuP7wK1g/t  
UqTG+vBy3MssaaTvSJHJxxfNf+qH5lYbpto3T82/qd24HjO1pw4bpuaY8vq9qMpbhKmUdgUQptqL  
c1988cWEqQgxe0dIBrvM+M2MuWbM9xi/OuOsm86SlIZmY9HWlN2gaJVZR7vT2smMVTNkiRnun8Vz  
rDE1z6n51G5XP5j60dQGUyWMBWYsMqP/sP6S0sj0QJuYngi6TI+1OrGVfLLsE/nDDPfP4jnWmqLH  
dt+KPC7rTZUwfjHjdzMG3j9QUhqhbHtDuMdPHTY9pKu8vft9sZbdKwVKR/WPCVPf5S93KCIpTOutt  
KJqTE77PfrXnl33xRWcC4803w6cyjJx2MVLayaq9mr+9Ho172iOPLDr6dseOzaFzfP7www/FH8c9  
vOxVg+1FVK644grze5seKKsCCFPtxxbffddWbp0qefPRO05Nc+pNfaPZewb5o+3uiwzj/EPJ7eW  
+cs/MVuxP+RnM+xz1jVPXSMpeaYHWpmKbHPapcgg90xjRFV5T1n6p+mH5vZ5e3rWqbZVtY+tLZM  
mDdBlopX340tUx+eOsB+bFlimy08x5t/uDNz73E7kduMb+Lfb63z/vO70XtRhXZj1ltqoTxkxI/  
mnHzyJslpZnpAfM4F/aDRjVJkWqdqslrs18znbXM9ZOUMDaZ+skUPbbH1o8//hh6zrrzzjvNpsf0  
QHkqzjDVXkjNnj/TXkzP62eiKlfZHrOP9T//+U/PfvCsOMNUzpmKWGYPCSgyfvx4OfXUuws3GiVe  
Dc9HuZfVuXNneeKJJ7iafxKaNgmS9OnTx7Mv4i33sg488EB5+OGHZZ29gg+Sit3ROeccczz7lt5y

L2v//fcP7cRFruZvL0C1115R2YX06xe6q5gJE6Kns/Xvftz3OpYvF7P88JXy7blRY6e3Qeo114g0  
bRp9u/ucqTaOdd9n69FHnTtjxB4Na0PYLz2uF2IvLvXUU+FAtUaN6HnsuVTtxbXs+VUrs+nTp8v5  
559f2A8I9piPMNW9rH322Ufuvfde0wumGVC5uK/m3759sav52wupDBGwIHTO3EhfxF7Nv5Upe9V+  
e0X/8aZCyzK1xtRjTl1lyj2PrTRXj7Vu3VqGDRsWCj6iuK/mbw9FR6Uz0/SgPfef9tX87Xkn7ZXW  
Fy9e7KwZyeKrr76SwYMH55UqVaL6w7MqGKa6e8xeqNFeaX2hPeE8ksq3334beoPbfXqpEvfJKhim  
upfTsGFDue6660JvqgMRZg8JKOI+ZL3cVyuUQJ1yyinOOqPXheThftz/8Y9/IPyE57N6mhd92+2V  
kg16LDm5H/dHH320MIwlqo466qjCN4li66plq5Vn2t9/D189316t/9NPRebMKX6FfLucL74Q6dCh  
KPOwde65zgSliP0ZSvuZKvK7JQt3jz3zzDOSIZVV2B9R5fPI1EMPPbQwpGc7Vgm5H1N79TfzmlfK  
vjvi/KG7H/WXXnpJcnJyonrEXaket5V2uy37hmMkQl3qsF9/FdnPOcTeXonOboRQKbm3LWPHjg38  
9Czt2rWTRYsWOWtAmnL3mD2YombNmp69EiofR6baN4PsEYplXu4emzZtWukXPfNxZKq98Oh3333n  
rAGlZvaUgJlTMDv133//feiJ6u2335Zq1apFbWC8gjD3bffff78sWLBavv76a44QhCf7JDh37tzQ  
RzYmT55c7EmwrB6zR9REemzNmjXOUoFo9iPn8+bNCx1RaK+MXII/ed128803F/bYqIWrnCqX+uUX  
MdtPkaysopzFHp26zz7hC07ZADUvL/oj/HZae4V/G7wi8ew2zPbZp59+Ki1atAj1TqjeMf3kcYbK  
0G3TzHdmmmsGXWMe819CPbbCnggXyWP9+vDh3PaPeMiQ8Lk19t9f5K67nAmK2NNn2OdLe0ShPWq5  
sMdMlBudu+yyywp7rKCgwFmiw74YtefqyM8XswKRRo3CP4PZ/5Nt25yJkAzsUVdzzJOIPaLQBu4I  
9ZPXbfY8gnb+b0wPFzvSGXDY/Sn72nLWrFly2GGHfXTU87zYswl3TbfffJeeluf2OjfoMtTO+4f9  
OBDgwR6hbANQW0cffXRRjz1SSo8tNN/lpsgZ3c8I7c/ZHrOndQJKY/bcgLLFHhmzzexcx+5kuauf  
67OtsfMCXrz6JGonK6ZOO+00Zyp6DOXj1Sfdu3f37C9bx9vP1zv+Vz1mz09qj1B97DGRG24IZy0D  
B4oMGhQ+iM2eBuCVVwhQdxdefXKiGSm7TE/FDnNbNzMi2I4IKffj/tprImeeGT7nx513hq9gd++9  
II9/7UxgJt+50/muiPvUObF18MEHF/ZW6Kt7fWPHho+MtRuVJ58UuflmkfPOC5+zJIK+TDpe26L+  
/ft79pet9u3byxbnCoRxs1AeXn2Sb0ZJz5VtzFhrhkWPoTy8+uRKM0rqsWZmFJhh0WMoL8JU+GKP  
arA7T147Vbbsjj0QD/sRV/si0Ku/bPXq1St05DTg14YNG6Rbt26e/WWrR48estmmmYBPW83oaYbp  
KM/RxYx1ZgAh7hdw//pX0WHntuxh6NOMhe+zR7M6n/axc5zSu3fhdiu2DjzgAFnx22+haUPvzlju  
UwvYskeizp8fvg/w0Ldv38Keii37cf4///zTmRLwZ4AZpqM8R0szFpoBxGOWGe6+co88M+abAVSE  
2YMCKo4wFdoiU6GNMBXaCFMRt1mz7FVcRN57r/iV6Ow5TQ8/XH6oV08+NP/+NKY+TEuTWebdINR  
851zjsjs2fyqV+ETNANIIeyFNsJUaCNMRdDMHhVQcYSp0EaYcm2EqVC1qwJhKp8oQ2IK+sihvfDK  
eefJB/XqyXNmm/VSTP3H1NudO8sWr/n5GCMqgDAV2soVprLZQhzKFabSY6gAwIT4QpgKbYSp0EaY  
Cm1bt26Vnt17evaXrS4Hd+HijliLPbdb71l+5t+xY0dZuXKIMzXgD2EqTA3oP8Czv2y1bNZSFv7K  
kamlz+CBgz37y1ZewzyZP48jU1ExhKnwhTAV2ghToY0wFdpCYWrPUsLULoSpiA9hKhKBMBXaBgwo

JUxt2TJ0hXYgHoMHIxKm5uWFrulPVARhKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYiaISp  
8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqTBGm  
QhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNM  
RSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbC  
VGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwh  
TIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBG  
mlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EI  
hKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXa  
CFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOh  
jTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYi  
aISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEq  
tBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJU  
aCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNM  
hTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBph  
KnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2E  
qdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoI  
U5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GN  
MBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApf  
CFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yFNsJUaCNMRSIQpkIbYSq0  
EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JUaCNMhTbCVGgjTIU2wlQk  
AmEqTBGmQhthKoJGmApfCFOhjTAV2ghToY0wFdoIU5EIhKnQRpgKbYSpCBphKnwhTIU2wlRoI0yF  
NsJUaCNMRSIQpkIbYSq0EaYiaISp8IUwFdoIU6GNMBXaCFOhjTAViUCYCM2EqdBGmlqgEabCF8JU  
aCNMhTbCVGgjTIU2wlQkAmEqTBGmQhthKoJGmApfCpYVSPuWpYSpvQhTER/CVGgjTIW2UJh6eClh  
6gGEqYhPKEztWUqY2rajrFxBmIr4EKZCG2EqTBGmImiEqai4XSIFZrQ3w2x+PEcfMyLTAn4QpkIb  
YSpUmee/rWb0NMN0IOfoYsY6M3iuhC+mb3aZ0dsMd1+5R0czVppBjyEehKnQRpgKbYSpCBphKnwp  
d5gK+ESYCM2EqdBW7jAV8KncYSoQB8JUaCNMhTbCVASNMBW+EKZCG2EqTBGmQhthKrQRpiIRCFoh  
jTAV2ghTETTCVJTInofLzf57+PDhctd1d8IND98kDQsaFu6sx46237WVe++5VwoWFjhzh8UuE8nN  
qx+eeeYZueuuu+Tvf/976lnN6wnPlt2xuvPOO2XJkiXOnGH0GNy8+mHEiBGh3rn99tulefPmnv1l  
a6+99pJbb71VfvvtN2fOMHoMbI798NJLL8mw64fJnffdKa1/be16doweTRY1kZvuvUkWzFrgzBIG

j8HNqx/GjBkd9xwh9x9z92y74/7uroqejRY0kCuv+96mf9F9ItEegxuXvOwduzY0PPkvffeKx06  
dPB8nrRVt25dufrqq+X777935gyjx+Dm1Q/jx48P9dh9990nnTp18uwvW7Vq1ZK//OUvMmvWLGfO  
MHoMbl79MGnSJLntttvk/vvvI0MPPdSzv2xVr15dhg4dKI988YUzJ1A2wlSUavr06VK1atWojU1q  
Sqqk1Dfff2eqhJH6hpnGNY+thx9+2FkqUOSzzz6TOXqRPVKamrx/iIP2R1+dqwQ66uvvpJGjRpF  
9YrfHrPhKkdEI9Z3331XLJgPPVdmme+nmCpzbDBV3ZRrvuuuu062bdvmLBkls0fM7L333IG9Euqx  
VPP9BFMIjVlmurrR2zsbSmzZssVZMhD266+/ygEHHBDVK+V9roydbuDAgbJp0yZnyUDY4sWLiWVa  
fnvs/PPPI/Xr1ztLBsKWLI0qRx11VFSv+O2xM888U9asWeMsGSiOMBXF2Ce6I444QtLT06M2KFFV  
RpiaMtaU13ym7BGF77zzjrM23IVMJpHHevny5dK9e/fSeyyOatasmYwbNy60LoseSx6Rx9pelf2E  
E06QjIwMzx6Jt5o0aSKjR48Orcuix5JH5LG2YVTV3r0IMzPTs0dCVVaY+rmmpDA1UvYNGoeffz60  
LoseSx6Rx3rnzp1y1lInSVZWlmePhKqsMPUbU3VMecxbv359efLJJ0Prsuix5OF+rG0olZ2d7dkj  
8ZY9avWhhx5y1kSPJRP3Y33ppZdKbm6uZ4/EW/ao1XvuucdZEz2WTNyP9eWXXy5VqITx7JF4q0aN  
GqGDKSL0MUQQpiLKOeecU7jhKPVdnDjC1MhyW7RoIQUF0acBQOV38cUXF+uFoCuyXBtG2DcHkFzs  
DIVsLwRdkeXWrl1bfv75Z2fNSBb26NHXYvCsOMLUyHLti4M5c+Y4a0aysC/cYnvBs+IIUyPLTUtl  
k6+//tpZM5KF/Wh1bC8EXe7lfvrpp86akSweffRRz14IstzLnTp1qrNmJltnn33WsxeCLPdy33rr  
LWfNAGEqXOyFVvr37x+18Six4ghTI2XPv/TLL7+EjrywFxi9vBaY2q7qRLGRjO2mJF/XX64B2qY  
qqVYZh1tD2kr8/6cF7pAh/tn8RxbTXn9XtTuVdtMITA2mLHdjCtvTlcmNiQyqs3girTY80PbC6z  
F80ObUPdP4vnsD+71+9E7TFIP1JoT/Nwww03FD6XIVpxhKmRaty4ceE5vLx+JqpyIT2q3h71Ys/x  
5tUPxSqOMDVS9IQ7n3zyCT1WmWqLqRLGWjOs+4bfF95GVTHI9RwXVKWnSPWm1WXqV+Ggy/2zeA67  
L2n3Kb1+L2r3qXL02CMjHpGUHNMDuTE9EXRlpEhOoxyZ9OmkoHrdP4vn2GGKHtv9a7OpEsYaM6yn  
XnIKUqqaHrB95tUbQVvmimTUzZCxU8eG1uv+WWLHKjPsxUXt609UboSpCLHnRq3Qx2EDCFOpSljv  
mNpTxwhTXr8TtXuV3bbsqWOMKa/fiaq8FUCYSIGlVgBhKIUJ6xFTe+r42FSmKa/fi9p96j5Te+r4  
wpQN4Lx+L2r3qdtM7aFjbzOWmoHKjTAVIdOmTSvccJXrEHnCVCq2Mky9a2pPHS+Y8vq9qN2rxpna  
U8erprx+J6ryFmEqpV2EqZRPW5qTx2fmCLo2v3rn6b21PGlqdmvH4vavepYab20LG/GcvMQOVG  
mlqokyjn5zsfwS6rAghT7Tm6cnJyQickp/bwSjFV3dQUU2WMjG0ZkrLJ9EACKnVLquTsMj1WnjHK  
VlapVFNEvyP1vy3bY/brBFNIjIT22OYK9NhYU9mmIr8LtcDWuT/JEUCYat/gtBeH8fo5qMpbpV7Y  
zF0BhKn0WCWqdFP2OeZJU2WMzO2mxzye11Rqc4pk7zQ9Vp7xmam6pniu3D3L7ivbx+YhU2WMzB27  
aY99baqxKXps96xMU/axucdUGSNrR5Z3PyhV1s4s19q9h5ISOpIbmFr5EaYiSt++faN2rt0721HI  
M0y1y4ws117Vf+3a8DI1UImUcXqYS/tfWtQPKUonCneW26h2lyn4vRwXOeOUNnuWMh6vK/OvLNYL  
QVdkubVza8vvP/3urLkU9FilcuONNxb1QmnPIXGEqZHI2hcWP/30k7NmJlthw4YV6wXPIiNMjSzX



vrn9/fffO2tGsnjg7geKekH5udLWFx+Hz/1cKp4rK5V/P/Rvz14lstzL/WjyR86aS0GPVSojnhrh  
2QtBlnu57775rrNmngDAVDnt0auQl1TVr1sjgwYOlFv36RRuR2B15n2GqPdLiFNOKa+++iq0Lst9  
ZCwqt8hjvXHjRrnniukYcOGhb1R6ovFCIR6err06tVLPv/889C6LHoseUQe6+3bt8vVV18tTZo0  
KeyNoHrMLue4446TDz/8MLQuix5LHu7H2oaqzZo1i+oNd69UNEx1z9+jRw+ZPHmysyZ6LJm4H+u/  
//3v0qJFC88eCVUFw1T3/EceeaRMnDjRWRM9lkzcj/Wdd94prVu39uwRP+Wev2vXrj2bPiCLRY9  
ljzcj/U//vEP2WeffTx7xE+55z/00ENl9OjRzprosWTifqwffvhh2W+//Tx7xE+55+/cubOMGDHC  
WRM9hiKEqSjVjz/+KE2bNg1tSGxIFdmolDdMTUtJK5xnzJgxzKBlosWLSrciY/qsXKWPalm8v3z  
zz/vLBUosnTp0sldLNtjFd3BcvfYv//9b2epQJFVq1bJQQcdFOqRqB4rT5haLfq58sEHH3SWChRZ  
v359KJgq1mPIDFPdPXbXXXc5SwWKbN26NfQmTrEeK2e5p7/pppucpQJFbAh10kknhXrEb49F5vm/  
//s/Z6lAtD59+oR6xG+PRfb7hwwZ4iwR8CLy/1UDo3WoqR9VAAAAAEIFTkSuQmCC

"

width=500

height=300

>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>First I changed the state to valid\_states = True and the lights changed and on coming traffic moved through the scene. The states appear to be light,oncoming,right,left,right. Light appears to have the

variables red or green, oncoming appears to have the variables left or right; and right/left appear to have the variable None. There appears to be an additional right variable outside of the state array that has a value of -49 I am not sure what this realtes to.(See this code below and run the cell to get the outpu. note sim display is set to False because it crushes my kernel when the simulator rolls up and cannot find the car images,then pygame stops responding. The full game with the simulator is run through the command line with display True then display turned off to run code in the notebook here. This will be true for all the results in the notebook. Additional note: enforce\_deadline has been set to False but the environment is still obeying the deadline (see below). I could not perceive a difference between the valid states being True or false.I think the state changes might need to be instantiated through code based upon the environment states, accounting for the self current state and the destination. Not sure how to accomplish this. But I think that is the goal to reach the destination and not crash within the time allotted by coding decisions into a Q value that gets instantiated.</p>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Code for valid\_states = True

file named agentB.py</p>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[1]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="c">#Can run the cell to see the results</span>

```

import random

from environment import Agent, Environment
from planner import RoutePlanner
from simulator import Simulator

```

```

class LearningAgent(Agent):

```

```

    """An agent that learns to drive in the smartcab
    world."""

```

```

    def __init__(self, env):

```

```

        super().__init__(env)

        self.__init__()

        self.env = env
        self.state = None
        self.next_waypoint = None
        self.color = 'red'

```

```

    def reset(self):
        self.color = 'red'

```

```

        self.planner = RoutePlanner(
            self.env,
            self.state,
            self.next_waypoint)

        # simple route planner to get
        next_waypoint

```

```

        # TODO: Initialize any additional variables here

```

```

    def reset(self):
        self.destination = None

```

```

        self.planner.route_to(
            self.destination)

```

<span class="c"># TODO: Prepare for a new trip; reset any variables here, if required</span>

<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span>

<span class="c"># Gather inputs</span>

<span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span>  
<span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">(</span> <span class="c"># from route planner, also displayed by simulator</span>

<span class="n">inputs</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get\_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="c"># TODO: Update state</span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="p">(</span><span class="n">inputs</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">,</span> <span class="n">deadline</span><span class="p">)</span>

<span class="n">valid\_states</span> <span class="o">=</span> <span class="bp">True</span>

<span class="c"># TODO: Select action according to your policy</span>

<span class="n">action</span> <span class="o">=</span> <span class="s">#39;forward#39;</span>

<span class="c"># Execute action and get reward</span>

<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span>

```
<span class="c"># TODO: Learn policy based on state, action, reward</span>
```

```
<span class="k">print</span> <span class="s">&quot;LearningAgent.update(): deadline = {}, inputs
= {}, action = {}, reward = {}&quot;</span><span class="o">.</span><span><span
class="n">format</span><span class="p">{</span><span class="n">deadline</span><span
class="p">,</span><span class="n">inputs</span><span class="p">,</span><span
class="n">action</span><span class="p">,</span><span class="n">reward</span><span
class="p">}</span><span class="c"># [debug]</span>
```

```
<span class="k">def</span> <span class="nf">run</span><span class="p">():</span>
```

```
<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of
trials.&quot;&quot;&quot;</span>
```

```
<span class="c"># Set up environment and agent</span>
```

```
<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span
class="p">()</span> <span class="c"># create environment (also adds some dummy traffic)</span>
```

```
<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span><span
class="o">.</span><span class="n">create_agent</span><span class="p">(</span><span><span
class="n">LearningAgent</span><span class="p">)</span><span class="c"># create agent</span>
```

```
<span class="n">e</span><span class="o">.</span><span class="n">set_primary_agent</span><span
class="p">(</span><span class="n">a</span><span class="p">,</span><span class="n">enforce_deadline</span><span class="o">=</span><span class="bp">False</span><span><span
class="p">)</span><span class="c"># specify agent to track</span>
```

```
<span class="c"># NOTE: You can set enforce_deadline=False while debugging to allow longer
trials</span>
```

```
<span class="c"># Now simulate it</span>
```

```
<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span><span
class="p">(</span><span class="n">e</span><span class="p">,</span><span class="n">update_delay</span><span class="o">=</span><span class="mf">0.5</span><span><span
class="p">,</span><span class="n">display</span><span class="o">=</span><span class="bp">False</span><span><span
class="p">)</span><span class="c"># create simulator (uses pygame
when display=True, if available)</span>
```

<span class="c"># NOTE: To speed up simulation, reduce update\_delay and/or set display=False</span>

<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span class="p">(</span><span class="n">n\_trials</span><span class="o">=</span><span class="mi">5</span><span class="p">)</span><span class="c"># run for a specified number of trials</span>

<span class="c"># NOTE: To quit midway, press Esc or close pygame window, or hit Ctrl+C on the command-line</span>

<span class="k">if</span> <span class="n">\_\_name\_\_</span> <span class="o">==</span> <span class="s">'\_\_main\_\_'</span><span class="p">:</span>

<span class="n">run</span><span class="p">()</span>

</pre></div>

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt"></div>

<div class="output\_subarea output\_stream output\_stdout output\_text">

<pre>Simulator.run(): Trial 0

Environment.reset(): Trial set up with start = (2, 5), destination = (6, 3), deadline = 30

RoutePlanner.route\_to(): destination = (6, 3)

LearningAgent.update(): deadline = 30, inputs = {'&apos;light&apos;': '&apos;green&apos;', '&apos;oncoming&apos;': None, '&apos;right&apos;': None, '&apos;left&apos;': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0



LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (2, 4), destination = (7, 2), deadline = 35

RoutePlanner.route\_to(): destination = (7, 2)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': 'left'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -38, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -39, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -78, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -93, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (7, 1), destination = (5, 6), deadline = 35

RoutePlanner.route\_to(): destination = (5, 6)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -15, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -38, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -70, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (1, 5), destination = (8, 6), deadline = 40

RoutePlanner.route\_to(): destination = (8, 6)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5



LearningAgent.update(): deadline = -59, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'forward'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -99, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (1, 6), destination = (4, 5), deadline = 20

RoutePlanner.route\_to(): destination = (4, 5)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 3, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': 'right'},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red',  
'oncoming': 'forward', 'right': None, 'left': None},  
action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

</pre>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Now running valid\_states = False to determine the diffence in state change</p>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[4]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="kn">from</span> <span class="nn">IPython.core.display</span> <span class="kn">import</span> <span class="n">Image</span>

<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span class="o">=</span><span class="p">(</span><span class="p">(</span><span class="s">stateChange2.png</span><span class="p">))</span><span class="p">)</span>

<span class="n">Image</span><span class="p">(</span><span class="n">filename</span><span class="o">=</span><span class="s">stateChange2.png</span><span class="p">,</span><span class="n">width</span><span class="o">=</span><span class="mi">500</span><span class="p">,</span><span class="n">height</span><span class="o">=</span><span class="mi">300</span><span class="p">)</span>

</pre></div>

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt output\_prompt">Out[4]:</div>

<div class="output\_png output\_subarea output\_execute\_result">

<img

src="

jwv8YQUAAAAJcEhZcwAAFPsAABH+AQ4RXKQAAP+ISURBVHhe7J0FvBXF28cPHYIBBmma3b52YWFj  
C1iIStuJUiYWdmC32Aq28DdRkZJWQBCKFUS64z7v/HbmOWfuuXti7tnLGTjPdZ7PPbuzu3Nnn519  
9rezFSNB2EglGuucefPp19EjgwwXvl5mBgTBkd9//53Gjx9vxlJz+OGHmyFurHMm0RFH7hlkEI2i  
Y2LNqeuF59LXq9TMJ99KRTSQrqh8ln1Da4M5Zr54Hl3b+Rr6aCXRIz2uoX9U3hWxcjQAE6d9g7+C  
kDMTJ06k9evWmTHTWAVhY0Aaq7DRil1V2GiltrGuXk20dCnRBRfo8b//JioqlrruOqJ//03kgZj6  
19tvT7RmDdGPPxKNHavzFyzQv2D9eqLttiO68EJdNnjwQaJ/IEqeN49o7lyd999/urwttdDjoEMH  
opVKVKM++J9YBvAyXJ/584lWKXGO//v223o+ngfDmP7YYxD2Og//h8E0gPmwPNZ18WKdB/bZJ+ED  
gHJXrCAaAnkz//vQT0cCBehrAsrvskvgfWMe16jyB/w/GbdiXWH9w3HG6Hlx/BuVxmTxt4UL9y36w  
16t6db1OS5boccyDbQGwva69lqhXLz3OdOxoBhTwJ3w2eLDJiAarhhEwfLhe6csvJzr//IRze/bU  
v1On6l/wzDNEtWrpYTTWHj2lzlHjzMoq149M2JRswLC2cuXEw0aVNzZTKNGRNdck5jGOWJvXOTv  
v7/+te3DD/V0gP9lw/MABDgMV6mSyF+0SE9jeF40TnDvvURDh+r8CRN0HjdmBuiYVr68Hp49W//+  
73/6d/p0/WuDhjVpEtGpp5oMBcrkcgHKrFRJD3PDu/hiPc+ffxldfbTOYzA//le/fnqc/y/yu3bV  
w8w36qSaGz3z/PN63giItjQXbryR6JZbzlh2bLcVxAbLSpQl1mzzlgDTZuaAQs0CpSHSOkjaMhh  
9cbZOOorNDdzmhBPMwIzhwzRWPlz22Hu9KxHvscLGQbRbHY2IzaZbN6J33yWqWIVP69MnvLHysn37  
6vkrV07k4VADBgzQ40LBIVtd2GjYMI01LlpeemNj/Nq1zYDi00+Jmjc3I4IQdWPF4fnEE82IAuM4  
c8TZL3c9AeS3bGIGFG3aEH31FS4EmwwFhP0225gRQdhQkVUQIkAaq7DRil1V2GiltLHWVUnYyFHN  
E/vTKbSidQt9khsBu6kUBZE01mbNmql1jFFsTkyd5Kuz/BwJyjKWK1KWAwcfrMv5Q1k1UxYunZeS  
l488UpC3MUbXX3+9yS09uXtKUa5cOV0p1VjZaaWyVCl3kxWpCwshc2byQ5RFpZaKwubP5OISmHz  
ZrJUKWzeLE39CYyHk6c7m2qsaCO5EkjLSoq0pVSjRXDuclrGQVVqlQJylrBN5LkQOXKIYOydtst



98NarVq1grKi2liTJ08OyqpWrZrJKSXG57GByv9Vc/d/vF2oxhpJuzC/kRCIZsVKRoWU5QjKevud  
8Is5pcArzcrUrSuN1QVfy2rSpEkkRylmiiMRiG4NFdJY3ZDG6kZ0a6iQxuqGNFY3oltDhTRWN6Sx  
uhHdGiqlksbohjdWN6NZQIY3VDWmsbkS3hgpprG5IY3UjujVUSGN1QxqrG9GtoUlaqxsWN2lbg0V  
0ljdkMbqRnRrqPD2cmuEq1kQjVWIFSpFhXeXW+Es3MgSBXwX17zk1+WUgqBeKvXAG19yJCjLWK5E  
WVanTp0iK2vSpEn6RpZqEd6UNDGadhFJKewor24RtC3XMgrsFkFurKHTSmOqsel3VyJprIMHD9aV  
Uo111KhRjrf08EpewC94y4GgLLPxcgFRJn7fbo5lAS4HZeYawbisKOp1zDHHxBtrrnz//fe6Xqqx  
ZvOW60xE0lgZ0axu+FrWJq9ZgfQGuOfT5WuKzekgbkhjdURaaxuSGN1I7o1VEhjdUMaqqvRraFC  
Gqsb0ljdiG4NFdJY3ZDG6kZ0a6iQxuqGNFY3oltDhTRWN6SxuhHdGiqlksbohjdWN6NZQIY3VDWms  
bks3hoq610fYWC8ogAZ2WYRINY+urCb3qca6ZhNurBVjFSk2K5ri0CCiup7/zDPP+NtYI4wVKOvr  
T782Y7IR7sdyFKsSTd0qVaoUmc8iKSVoXDBzi+DNN99caqtQoYluCymXsopupsv/vZxiD6lyjKlc  
uvlKa55s7cmbqZtKJcp615rH0Xgdaz/yLDVSPmzfjv3tpH4x3KljR7qia7fQ5cLM9tc111wTOk+2  
FpRI7rrafPPNQ+fJylSKPaDKYZ/dq32WC2XSWHNhu+2202UZ5+fChAkTqFz5CvTK2VvmXBbA7Xx9  
K9eMpCxex8rqd9uXXqejbu1Cp3TtShe1vox2VNNnzA750F0KbH8tSv4cpyNBWaaX7rvvvia3dARl  
WZYruZdgEektghFV7e3m9amoRwVa2bWqySk9gyrVpjWqwX6tfqMgqnUEUZyV+S2C0huQHU+12o9u  
bNOcfmlf0+SUnn41t6YbmzenXvUampzcSLWON5uP+h40blzwG8PN7UOGBMOpiMpfQLquHlnK+fc3  
34PW3VGV3jm/uskpPaceeAitrVKFTtvvAJOTG5kaa2Pz1EVs0KCgwaZDGqsjPjbWSTduEZY8jGhb  
2eTkQMwKQVkrK1cxGbkRZQOTxuql9xcFuqrF5fOlepjxO9UvzIF4WW+/jRGiv/7Sv7bhG/34qHIG  
pLG6Ed0aKjaKxjr9R6KPW+vxu8onGm6WFGusAl0Vn+1EPuyoo3Q+PgOaAWmsbkS3hgrvG2sEFEJZ  
OlgtkQbmhjRWR6SxuiGN1Y3o1lAhjdUNAaxuRLeGCmmsbkjhjSO6NVRIY3VDGqsb0a2hor5KURHp  
hoxoNYPPO0boMpQVxZv6AL5VGxXHHXecGYqGPVWKgkjWEA0rsAjuuuJvrbLlwm+//abLQcqXLLAx  
ILVw4UKTWzqCsoxtv/32Jrd0xMvy6S2C8UqXztqnT5/SmUoVW1ek2EWqnBbKLIZIIXNm4U9+OCD  
ul5mQ4bNk6317ds3srJgdIkffPBB6DzZml3W888/HzpPVqYSfM5W+9raOfk/qBfMp8bKRCUD/u68  
BdFdVahPt5YmJzeUq8xQbogMKB1eyQCm7inRnGC173hVcMPlc7ddbnJyI3ZQhA3M17J2ja6syE+w  
ztuUewPurBjc1jf8wydNRm5EcQhiCqEs6bpy4PwTm1K7tm2pT48LTU5uSGN1QxqrAx8/cxc127mc  
HrnDVHH8h0TL5hInuCMrlyXSWN2Qxupl3PlomLiNb8V/enz2cKLb3aotjdUNAayOSANzQxqrI9JY  
3ZDG6kZ0a6iQxuqGNFY3oltDhTRWN6SxuhHdGiqlksbohjdWN6NZQIY3VDWmsbkS3hoplG2uEVYu0  
URRAvTbp1wfVrKlFvgbbeutTW7pwBvogrKQctyYc+fOjdcLL1XLlajqBbheUZQVfFPW1CvXiBJU  
ybyY7d577zW5paNy5crxdWzQoIHJLT05eypwUlgyISy15VpGkbKwFDavi0VVRlgKmdby3V526L4  
KnaKlAu579aK4NY5U8kobnldOeVUiol2bdoE5UXBkCFDAof/9NNPJqf0fPvtt0G9fvnlF5NTetas

WRP4685b7zQ5uRH7Vm/LXIm8XZjfSBDN6oav9ZJXXjoS6YaUspyQ3gBHplG5IY3VEWmsbkhjdSO6  
NVRlY3VDGqsb0a2hQhqrG9JY3YhuDRXSWN2QxupGdGuokMbqhjRWN6JbQ4U0VjeksboR3RoqpLG6  
lY3VjejWUCGN1Q1prG5Et4aKSL8wGOWGjHA1vb3cGmVjjfpyq0pRENkawll4MVsUBGUpe/nll01O  
6QnKQopgY0ZelrFcibKs2267LX7XVRQE9ZoYUVnmNyfizjJvESy1pUph87pYFGWwRVWfsBQ2byZL  
lclmzdaiuEXQnt/elhhUKuLI+vPPP5uc0hOUpVL9+rm/4ZDLwm+u8DpGUdaWW+qvfdR1htvvBFv  
rFHd7ulVZBWEDUHsnptiNHfsflT0n+O7UFcnvtLcctuLzJAgZOb3338PLB2dOnUyQwllt3W7http3  
upCann56kHH/aPVnwcfcz49UoeLxNOXdW2j4WkwZRSe8PBsDdN93K9XfSXTvwHm0dtGbtM2+16vx  
8XTp2QfSQfeMceYRhFRkaqhgyZlI9PXXX5sxjcgAYaNBGquw0SCNVdhoiLax/vOP/t15Z6J//yVa  
v56odm2dt2CB/k3miCNULVQ1+vYIWrpUL4fxuXPxeCSUtpLHk4j+/lvP//rrRCuVZl63jmjZMp2X  
zK+/6l+uD+Zdq4Q3jJfB/1m0iGjWLD3M2GWuWkVkX5jo35/o7LN1/QCWW748sTzqnAx8gDqvWUP0  
33+JdQMYvvrqxLpxuQDzYv3nzCGaP1/n2Z8NataseD3A4YfrvKZNE+vO86AcBAP4AMPz5ul8/J8w  
4GP2NcpHOVxmrVpEr7yih5m99tLzwLAdJ040E6LDrEkEwBGffKlbu67E511lq44f4sf2P12cBqD  
6Xg5xgkn6GEYNjLo3Fn/2rz4op5ntjrhq1KFaPVqnW+Xf9tt+ndP86UQLrd9e/2La99XXKGnJcPz  
9uypx7lcnFbA01HHN97QeYsX69+wBsvzb765/mXsYYBxfhkHdiRejg2+tUEeGglzDE6zwbj++yj  
GygaEJf1f/+X+F8g7JFwnpeNwTa1+f77RH0nTCBq0cJMiBarBhsY7H3JK12+PO56MCN5pnFjomrV  
zEgEYGdMXt/S0KGDGYiQdO8uqFmTqDQfb7Mbd0SUbWNFhX/7zYwo7MjHlMFKCZsm0baU4cP1oQqN  
En1kalilJj8gQ8+0L92A8UwH44GDtS/991HtNVWevILL4j+9z89X4UKOk8oSKJtrIJQhkTXWMMO  
8enYe28zYHj2WX1SBrcEq8T1MOCYlg+ss6YoX9x2D7vPN3tgcM8xu3Dv91Y+ZDPjbVSJaKpU/Ww  
lBib6yCUEZlYxU2GqSxChsNkTXWmLT7TYMLLTDbMooLGlo020UkJQWPLiDZJ1ClhB/P8Lms8ePH  
m9zS8cUXX8TLqIGjhsktPXbdcgVI9D0nurKCFEFZIOdSF9eXKxSOVmqFDZvJkuVwubNZKIS2LyZ  
LFUKmzeTpUph82awfuZX/QIsXzOes5n6REEkpURZqc6dOwfl4AswUYCyuoegwgl1jOCdQRRlhV8  
rSWKsqZODSI9tuWlflNQDgTrGFG7ANGUokCloiKqIQNRltW/f3965JFHfJfueLmeuFUTZU2fbjJy  
l9J2YX5zxtcGJo3VHW/LMr85UwjOksbqTqRlmd+cKQRnSWN1J9KyzG/OFIKzpLG6E2lZ5jdnCsFZ  
0ljdbQs85szheAsaazuRFqW+c2ZQnCWNFZ3li3L/OZMlThLGqs7kZZlfnOmEJwljdWdSMsyvzIT  
CM6SxupOpGWZ35zx9nJrhPUqmMYa5baMtKwlgKOCFIHD9txzT11eBGW988478XpF9RZn3DQSBVHd  
fil1Yn8NGDDA5JaeoCyVjj32WJNTERisKNYT5FwK3qNpVyonS5XC5s3Wcl0eliqFzZvJUqWweTNZ  
qhQ2r4tFUQabKSsKlikliBlqRRF17BXNNRpyvaJw1qOPPhqv11r7PV2lYNGiRfGyXnvtNZNberis  
rfGKohyIR+mIfBZluwDRNHkFKhUVUTiKibJeolndibasiPDW8RGWJb0B7kRalvnNmUJwljRWdyIt  
y/zmTCE4SxqrO5GWZX5zphCcJY3VnUjLMr85UwjOksbqTqRlmd+cKQRnSWN1J9KyzG/OFIKzpLG6

E2IZ5jdnCsFZ0ljdbQs85szheAsaazuRFqW+c2ZQnCWNFZ3li3L/OZM7LIIK9XKT2dF3IhbRLie  
Ufo/yrK8bKzRFRVtWcpZxx13nBnLjYoVK0bmfJRjr+ekP/4wQ+40atQoMp8Fd9FFVBbwqrGuWrUq  
7nj83nzzzTmZXVb37t1D58nKVlo9pMoxduSgl4O80HkzWVJZsXvUepa2LGXBOPr1rD9uPJ1/1llo  
Ys+e1Ej58+Q27aj+/AWWhy4XZHnvsUcxnYfNka127do2mrOtupltUsn2mcuJIIXIhkmZvr2CuRF1W  
t6O3pi2rV6FevXqZ3NKBss6vuBkdXbFaznUL1hGmUsPFS6jOR5/STI98SRdffjntpKbv4HALYocO  
HeJI5VovEHIzqxlgmIIUWMGoiKqs+87fg+iuijS8Q273eYKV+DZrxQrOV8UaNDfVR5Md8dFnINKy  
ImqolILKS1q1URGW9dPHOdGOB5tT58jNpzrwUX3/OkttOO5tuatGCbrzgAvr6rXdNbm6kWs96I0dS  
7JdfKDZ4MB0/fjydN3kynZbhK9M++h9EWpb5zRkfV7DHD1o3R1VaU7n3F9MvHL5ClpXtSPR5cq0  
hr/YnSOp1jNorD//TBWGDjU5RIfb38ANwdsGFmVZ5jdnfFzBBztfF2i6LhcebXJKz8rly+mKa6+l  
cy65IFbwJ+NzpCAaWJRIld+c2SicNWMQOZ1q/O3T1XFd/d5bnejLK83EzPQfMCDRz4pyYfj8+S67  
JPI+/1z/ZsFG4bMcibQs85szG4Wz0Fjvrkj02PZEo17VeX98on+zoNhFAZQL69SJaN99dR5o1kzn  
Z8FG4bMcibQs85szheAsudzqTqRIld+cKQRnSWN1J9KyzG/OFIKzpLG6E2IZ5jdnCsFZ0ljdbQs  
85szheAsaazuRFqW+c2ZQnCWNFZ3li3L/OaMv9eToyvrJ5WeUSkqYkV+rmdU/g/enRVlvxcvTmDI  
ghTBSgZIGcsVu1533XWXyS0dQVknK7tRI5cLQVIW3XLh4osvjqwsEK9bVGVFVC8QSSI2pfr06VN6  
Uyl2sSquTL1+7FKQW7YvFmYXa/zzjsvdJ5sLSjLaqxh82RrQVIW3cLmydaOPPLlaMqa2Yc+VCnw  
vzGVGz5vFvb2228Xq1cURFJKIOF+aJuKtP6OqjSkfW2TkxtR1QuIDHDDSxkAYgdFU9SgKxtQ+/bt  
6fwLLzY5uRHbLzpn9R+sTrD6RHiCtVuEG/LACMuKqLGCqNoFiKyqFaw9xUHBrf19Wm3l8nJjSgd  
L7OB7kRalvnNmagqNaRNFWrbti1NvKaGycmNKJ0ljdWdSMsyvzkTVaXmL1xE5+xehT557TmTo+A7  
pLqr/9GvFdGK+Xo8C6J0ljRWdyIty/zmTJmu4KqFRNO+l+qi8n93e6QkynpJY3Un0rLMb84UgrOk  
sboTaVnmN2cKwVnSWN2JtCzzmzOF4CxprO5EWpb5zZlCcJY0VnciLcv85kwhOEsaqzuRIlmV+c6YQ  
nCWN1Z1lyzK/ORNtpfx0Vn+VHIEpKrz1maf+j6QkVlgtV6L83qpdr++++87klo6gHHPXVTM8bp0D  
RxxxRLG65cL777+vy4nSZypF/R3eKMi5FL1qSWmCMquizoYUlu9iqVLYvNmadYtgqS1VCpvXxalo  
gy2q+iSgl1XKhUia/DfffBNU8OuvvzY5pWfy5MnBin377bcmJzdQrwYNGpix3lgdrmp2UURRQtUr  
qq9FV6pUKfBZFPTt2zco6++//zY5pQdHM6zngAEDTE5uRLOGCIQqKqJyPliyXqJZ3YI2HSMi0kp5  
Wpb0BrgTaVnmN2cKwVnSWN2JtCzzmzOF4CxprO5EWpb5zZlCcJY0VnciLcv85kwhOEsaqzuRIlmV+  
c6YQnCWN1Z1lyzK/OVMlzpLG6k6kZZnfnCkEZ0ljdSfSssxvzhSCs6SxuhNpWeY3ZwrBWdJY3Ym0  
LPObM/5eouuLK8vt3rq52jLigC+rc+328q23XbboF5RlDV79uz4XVdRwOulDzXnCpd18MG53dUE  
grli8lmUt3uCnEsJPvltrWBOliqFzZut5bq8bb7dlpgqhc3rYIGUwWbKioJISomyUuXLI4+vaK6c  
fPLJ8XrhjXa5EjTWmyJyfETrGLypz5TVrl07k1t6grJUwm2HucJIRbGeIF7K2mk7029jB9MXn39O

y5YtM7mCIAiCK0FgbX1ejFZOaUzLJu9B80ztQ/TPzsFEzXA6/dYnabd9etBpNz9Ka+eOp0lrp9PF  
NzxD9WOH0qLfP6e7urZV831PF93emxqc2pGe6XQYrVA51/Z8iTo/83FQyt49vqAv7zgrGP56+mB6  
dyXR7hd2o+OPOlvWrf6VbrnzKTpww2up76X/R73vbU3D5y+kUXOJ9v+/drT48wuC5QRBEERL77//  
boZyp/VII9F1119vxkoSgzyvu22M1k1vTEVzm9CpLc6m2rXtF6QPp69N38qyCa/S8mBoPc2et5Bq  
X/B2MCYIgiAkiKZDQRAEQYgJgVUQBCFiCi+wfv890RdfEC1cSHT00UQrVxJVq0Y0dCjRVlsRPfQQ  
OT77mJmT2HZbMxACuk8GDjQjFqeeSuRyBbRuXf1bq5b+tenbNzG9rHH5P1i/004zIxY//FDcZ9ts  
YwYioHp1M2BRs6YZSGKLLYhGjTljSbz1FtH225sRi6VLibbc0owoUpVdGIDWxInh68CsXk20ww76  
F8yfn3p+u+sO8/zjxlxZMwYomOOMSNJbL21GTDMMkW02WZmJlIU+eCMM4guv9yMpODWW4n239+M  
GA44gKhLFz3csKH+9Rh/Aytue1i8mOj884nWryfq1llor72lrr6a6O23ifBGLb41YrvtiB54Ql8/  
/jjRhRcSHX+8ntYWF9Ys7r5b/9arp3+ZH3/Uy/fooRvRK68Q4e1Ya9cm/g9+8X8R4DgP82AZZsKE  
8IZTsSJR1apEN92kl33vPaLKlYmGDCE66yyili3xurBEuenAsjXMF8L23FM3RCz32WdEP/2kh3Fn  
B5eF/ztnvh4fPZroq6/08KWX6qAyc2Zi3rBfLHvzzbq+nlcdfd999XrZtGpIBhT9++t5Dz2U6N13  
i5eJuv72mx5++eVEORjHQQ+/ffoQNW5MdN99eloYmk9RI6IZM/TwmjX6F+A2nHvv1eNt2uhgydPw  
Fi8eZq64wgwo0PYw/ZJL9P/neRE0Hn6Y6L//dN60aYlp+OV1fvFF3QYvukhPC4OXa9qUqGNHogcf  
TASwM88kOuEEPczzMe3bmwEDpmpZFfi308Jw5RHvvTdS6dWK6/bvTTnoaxu+/X+9bfGMpz5MKTD/y  
SN2OMDxuXGIZ/E6frn8fe0zvCzytZ8/0wqR3b6Lnnfy69VfcVUrUq1di2V12IWrsRP9iGyEecPv3  
FFNzD4EKwo5xxx16fPZsvfP9+y9RISpEU6YkGg5vZDgcDRMqkeGNwyAlg2SVYgdWVgCHHEJO+umJ  
MvgX/4eHx44t/j9SBcfy5YleeEEPYx6s2xNP6HHctwiFiMaIRm4TVhZ49FGiChV0wAP2fDzcr5/2  
FVQ4WLJEH1D22EP/TwRzBEfAy6T6/fDDxPCIEfp/P/lkycDK8wAOMgB3l/Mw/ledGwFx+XJdDt+L  
iHmwjSdN0mcSfOUVQf3pp/Wwjf3/eBgHT7SDzTdPBFackQCeB0oQ0zkf2GVxYGXsYQQt+BFtcvz4  
xDT8fvIJ0YIFumwc4AECEZRaMnaZOEaICnnKLPqAAEpQ1Yy8H7HEMI7ACLA/VjWAFed6owebN  
9XCdOsVVKs+DX2yHZHg6B1CAgA7Rg3E7H/AwDs7Yhlwm2s2zz+rhZA4/vPgZEw78EAdXXqnH4V8c  
SDzG8sBGCgJE165En39uMkrB//6XULJlxbXXmgFBEDZ1/A2s3E+DI966dVol4LQep/onnZSYBtB3  
162bVpvcNwmFiL4mzINTknPPTfTx2X1ngMuBaklZv/yi81hx4UiLX1Zd6KeFEgTc94Z8KFFWrJh+  
zz1EO+9M9OWXOk8QhILA370dp6g4bccpEfrj5s7VwRYBC6cC4NhjdR4C4o036jz05yGlod8RcEBD  
Px4PJwc5Hp88WZ9+oo8HXQPvv5+YZvfd4fdj/eAD4XstOK1B4EbQ5/5LgGVQX3QncH+ZIAibPCYC  
FBDom0W/aGnBIVMoYEEQhBT4G1ih+mCvvqrVKG6XwQUJXNRApzyuMmI6AiU693GnAK7Q45tZrBi5  
DFaeUKFQt+hW4OmpQAc6rtbb8Py48JFuWUEQChp/owMHLtx6g9N9jOMWKIBbWHC7Bd8NAKAkMYz+  
V1zptK/cA766f9hhJftow9hvp7xySA/zfHwrE65Q4nYjQRCEENJEFKEQBKE0SGAVBEGIGC8D6yKV

lqgkCIInLFqkzUMQL5aq5BPeBdazTz5bVSpGleebyxc94YQTTiC8XbwO+m8VUbzIPVdQh++++y6o  
F6x79+5mih8ccsgbQb1O4j5tTzjttNOCeh133HEmxw/atm0b35aTJk3KbxvD01vKT1yD2r/VDvbL  
k84y2xKPoOKJqzzvB2c1OyuoV7Xp1UyOH3gRWNeVXx9vUIEhzUmMjxkzZsObSZUPrEyxvU9Qmya  
SvE5w8ooCzOp7599Q+sE6/Rsp/h8Y0YqCysnapusTKWpKoXVKbAd1LbkNENZWDIRG9bfpNj2IXUy  
Nkml+Jxh5ZSFmXT6baeH1gn28+Kf4/ONGa0srJylbfBB9HIhQtp8uTJwf6n/hQz5NWtW1fVR188  
DiujLG3cuHFBHeKGNDEx7gN+1EJRwIFWYMOxneplWbly7l+jy4XfR/8eWi9Yl1vMW4DyBHwTVi9Y  
PgmrD1s+OafZOaf1gi3+b7GZawOD2w0V+Mre7qoeqjKB7W/qtQXeGKYosl+6swEp5icJrNkBR9VV  
yTd82na2vtYLH2FGvaL8EHNUyLbMArxhC08yKprQ0cF+iU8uBS9awRvePAH12k0ln/AzsKqGhVMN  
35Cd0Q0JrO74Wq8mRzcJ6rViRRBavQL12m1XCawZgaMksGaPr/WSwOqOr/Vq0sTzwLqbBNaMwFES  
WLP13pJYHXH13pJYHVDAqsDvjZ6X+slgdUdX+slgdUNCawO+Nrofa2XBFZ3fK2XBFY3JLA64Guj  
97VeEljd8bVeEljdkMDqgK+N3td6SWB1x9d6SWB1QwKrA742el/rJYHVHV/rJYHVDQmsDvja6H2t  
lwRWd3ytlwRWNYswOuBro/e1XhJY3fG1XhJY3ZDA6oCvjd7XeklgdcfXeklgdcPPwKpSPZV8w9dG  
D38h+cbXX38d+Ouxxx4zOf7g7bb0tF5NVEIbW62Sb6Be8q6AFHCDCgzJk7db4e0+xepmWT4p9uo0  
Tma8R48eZq78EK9XiOWTsPqw5ZMWLVqE1gm2gD/1nifidRmoDKmaHuf3EueLeL1gSPJ2q3BKOMoK  
rE2bNt2wdolyk8p/W55iX6t6hNgpKsXnDCunDO2www4r7i8kM96oUaPQZcrUrlKmEnwS5qvAvlLb  
ktODysLKidoOU2ZS7MuQOhk7Wax4nGHlIIWZVGdcndA6wY5Ze0x8vqbHKAAsrp4zs5JNPTsXpMBa  
tWrV0GU2IMXrBUOSwJqeJXOXBI7abv12JscP/v33X682HLNy5UrdsFSi/0ymJ7z33nuBv+644w6T  
4werV6/2clvOnz/fy3qBw9ccHrSx6fOmmxw/mD9X+Uylxmsbmwx/8G8LKMJPxqjuy3LxKltij6I6  
KZ/5hly8csfXejX5sEnQxlas9fDilarXbm9IH2tG0LDkroDs8bVeEljd8bVecleAGxJYHfC10fta  
Lwms7vhaLwmsbkhgdcDXRu9rvSSwuuNrvSSwuiGB1QFfG72v9ZLA6o6v9ZLA6oYEVgd8bfS+1ksC  
qzu+1ksCqxsSWB3wtDh7Wi8JrO74Wi8JrG5IYHXA10bva70ksLrja70ksLohgdUBXxu9r/WSwOqO  
r/WSwOqGBFYHfG30vtZLAqs7vtZLAqsbeIgd8LXR+1ovCazu+FovCaxu+BIYVaqrkm+gXki+4evO  
6HVg9XVbelovfm3gCpV8QwJrGnbddddd4gAiSebvVwQcfbObID2PGjEm8OpCTqWc+WbNmTbwetqGu  
S5cuNXPIh7B6wV555RUzR3648847E/UxqalabrB8OTUY/RvVHfAd7fPBR7R3j9vptJs701nt29MZ  
N95EO65ftZup5esrmzlzdlBWIODVgKna2Hr1v/NJ3F/W261Q16ITp5o58gNeWxivm2V485UPeBFY  
lyxZUvy9p0jWawO9MU5h0zaUFSIzSWFIIUdoswltVYVWVs6GMpNq39uLdli1Ogicz84k9qrYHri  
PT2DcBaT2rSIDh06UL2586hig4bh5UVInMKm5dOSXhuYN3NIh6qUL7xRrEVFRXTuuecmnDcrFowj  
P9/w6++QyqnUqlUrMyW/vNnnbRravjatvaM60V1V6K8btqAXmzc0U/PH2rVr6d9YVfqk0ja0tkoV  
GI95S6Jq1czU/IG21Lp162Lb8qKLLjTE3zxww9maMMBcYF6oU6oW5ODm5gp+ad58+YU+077DHX8

+eefzZT8gW153nnn6W1pzJd4AfzrzFFgA0ofa2Yeuv9BurBly0Blwdp16ER9zqpgpuaPVatW0cN7  
7ReoPNQLv/2qqeDqEa7b8ubp06neyJFmrCSxQYNoldqpY7/8YnJKh2u9NhTSx+qGn4FVOUruCsjM  
c08+Qe9dtT+tu6NqYAu7bkYvn1XJTM0fq1Vg7dLkWFpXtWqgWKlyZbrmxFPMVD9w3ZYcWHv/8w/t  
MWYM3T97Nj00Z46ZqspTgTX4zTWwetbGGLkrwA0JrA741ugfaHMKjblmC1ravWZg87psTv/dUtVM  
zR8rly6l4TU2pyWb1aQl1WvQ8s1q0PzKVWjRsmVmjvzj27ZkfK2XBFY3JLA64FujX756PdH9FYnu  
qExLuihl2LMK3X/rVXpiV1XPGVpFleC3d4nuLKeHbyub9ZkTKx8o1XIVVKCvon5r1jJTLd7+mNSe  
akaSCPNzhL73bVsyvtZLAqsb/m1BBRwlgTV7QuvFgfXNk4meUo3umX20gVGvEt2IAh/oUXbrE3of  
69tvo8J6GL9//ZUYB/Y0cNRRRM2a6WF7vhzZqLalB0hgdcO/LaiAoySwZo+v9cr6AYH/+z+ilUOI  
nn2WqMKGufgm29INCaxuSGB1wNdG72u95JFWd3ytlwRWNYswOuBro/e1XhJY3fG1XhJY3ZDA6oCv  
jd7XeklgdcfXeklgdUMCqwO+Nnpf6yWB1R1f6yWB1Q0JrA742uh9rZcEVnd8rZcEVjf8DKwqefll  
q6eNHv5C8o3+KqFej6jkG776zNd6eR1YVdpNJZ/wagty4AqSebuVD1SqVCIRN2NDcHtQnjnzDN1  
fTip4UaNGnnxloqgXiebet2k6+YDQb2SfJZvsL222267EvW66irzsEceeeeddxl+M1atWjV/2hgM  
aale9gUvasJv9inmqHy/NtA1hZWxocyHOoQZB9YbzbhPxilsWj7Nh3r1U+aSwsrYklZkAivMB/wJ  
8YpiL/tVgRXjPoA6vXBmQ/rv1s3o96u3pJeb16Nnn3/JTM0ftWvXTvgLyZOGNW3aNF2XpMDqA+wj  
TmhjOLDnm+R6YfiAAw4wU/PHTTfdlKibZT5QLF6owOpLvABeBVYGjvKpj3XQ9Tvtw5cdQWddeAWd  
27IV9W23E/3w+n1mav4xTcuM+YP0sbrja72kj9UNPwOr2oA+3RXwyIVHE91VmdbfUTWw6TdWp/H9  
869YGZ9Uhl3cFeCoR/WSuwlckMCaBR9dsaMKrBXj7z19u/1e9PNLXc3U/OPrziiB1R1f6yWB1Q0J  
rFnwffdjauGXmnTVFRdSuwvPUqq1Go38/gS98dY0LpwygOgWNB1oPdHCaSYzenzdGSWwuNrvSSw  
uuHfFITAURvNAwJ3mPHu6veFg4ieaETUe0+d10XI/TOGaMJHRLexnatD6+UBEljd8bVeEljd8G8L  
KuCojSawvnacGVB80FzJ29vNiOKlw4nm/0G0fg3R282IliY+5RElofXyAAms7vhaLwmsbkhgdcDX  
Ru9rvSSwuNrvSSwuiGB1QFfg72v9ZLA6o6v9ZLA6oYEVgd8bfS+1ksCqzu+1ksCqxsSWB3wtdH7  
Wi8JrO74Wi8JrG5IYHXA10bva70ksLrja70ksLohgdWB2Heq0X8jO2O29P9PBdbvVWCdIYE1W2Jf  
q3opn/lGk1EqsKp6rVgvgTUbvNmCeA3ZL7/8Em/wsEGDBnnxerL4S0VUKqcS6ukLHTt2jPtrhx12  
MLn5Z+zYsRQ7XPssdlGMhg8fbqbkF7uN4aUdo0ePNIPyy9q1a6ly5cpB+4LP7uvhz7sovvzyS4p9  
a7al8tu8efPMIPwydOjQ+LaE/fTTT2ZK/sI7YG1gUmyGcs7MEFP5PE+DycoabFiLv0EHxsmMb7/9  
9qHLIKkVKVNpuzXbhftL2eaLN2ePUYPNIIWVU0YGn8T9FfLawLBlytRqKjMpNI3VlcrfsIYqxecM  
K6cMrXr16gmfcTLj9erVC11mQ1jDhg0T9RoY1Ipi1fR4hQoVQpcpU9tamUkp44XaxjzPGSrlC28U  
62abbZbYiJZttdVWZo780KITp+KvJ0Mydcsnf//9d7wetqGuAwYMMHPlh3h9kgLrCsecYOblDwce  
eGCibkmWT958882UbWz58uVmrVzA9bADK+ravXt3M0d+CHv5PKxOnTpmjvziTWC1gYO87GM1yTe4

UfmG168N9NRnuoX5V68mKqFeK1TyDWxH6WPNAjhK7grIHl/rJXcFuONrveSuADcksDrga6P3tV4S  
WN3xtV4SWN2QwOqAr43e13pJYHXH13pJYHVDAqsDvjZ6X+slgdUdX+slgdUNCawO+Nrofa2XBFZ3  
fk2XBFY3JLA64Guj97VeEljd8bVeEljdkMDqgK+N3td6SWB1x9d6SWB1QwKrA742el/rJYHVHV/r  
JYHVDQmsDvja6H2tlwRWd3ytlwRWNYswOuBro/e1XhJY3fG1XhJY3fA3sG4jgTVbfK1X/69MYH1I  
Amu2+FqvJkd4HlgbS2BNCb8iUDUtqq+Sb/ja6OEvJJ/AtvxZJdTrWZV8g7elD6+ltPG1jR2vko9t  
DKBee6rkE954ihtUYEhzEuP5pEqsSvG6WRYnD/vmlCITEnXhZMbvuy+/7/KM1yvktYH5JF6vEluz  
lbel+V+tz28dWifYqgWr9Ex5Il6XpNcG5vvdv/F6wZAmJsZ9wl9aKEo4yqqsS5cu3fBmUpVaKrDW  
UPUIMRCfM6yMMrRhw4YV9xeSGe/WrVvoMhvcQLXeIYE1bJkyN5OCOoRsR9h6leJzhpVRfMZSyyta  
htYJNmvrPh8oWWUoRXblkmBFe9qDVtmQxhepRivFwxJAqsgCMKmSxBUr2p/JH3yytbqtGS9N9Fe  
EARhYyQ2ffoUmjx5IL3z/pv01gdf0px/l9HmWzQwk8uGy/e72AzZfELj15lBQRCEiPn999/jNnjw  
YJPrxquvvkpr1qwJhmfNmkX//vtvMGwTmzNnKsOesSMt+3NvWjmlMY3uX49efOJUM5moeaw6PXvX  
OfTbpG9p1er1dOXVV9Fzp+5Jzz52CX2ty71IVfozt7D6LTYVtS7zUF0z2O9qephL9KQB7rRm089  
SkvXFNHawY+qklZTkzu+po96X0VtYvtR3ytOoqcevIM+nLKYbjvpOhr0nnNpJq2h09s+TUcf0YPu  
6N6e1o+5g8Yulzrr8T91ZQRBEcjg1KhRZsidJUuW0Jlnn23GShL786/JdOLpzejV1x6ivQ+sT2e3  
usxM0lwaq6r+DqSh6u+RJ91Gv6ggd175ukTfdqLhsybQdJV/WdfP6fKtZlFDg2ngaQK2saPo6aN2  
VuMJjmh2Js17pz1d+d48urLC4fThFa1V7iq6/aO/6Jz9Lq15LxxHc1VpD6sCz9vsPFozog8d2+sP  
euzmM3UBgiAIEZFLUK25+RbB79777hv8JhObN+9fmvhDXVo543B67sIL6aZbbzWTNJfHmgS/69as  
pCYnvxIML/p7On1+XSOKwVBEAQbuSolCIQIRJUBUEQIkSCqiAIQoRIUBUEQYgQCqCIAGrsNEF  
1bC3C7m+cahePf17wAFE//xDVk0a0bp1RFvo2yRou+30bzKPP060++5mxGD/7y23NANJ8P/LhmOO  
lbrhBiK8X/bjj02mRa1aZqCMue46ouOOMyNZkGod4ZNvv9XDzz1HtHPx2+xKzZllervZbL450bhx  
ZiQJTEvF1lubgSSqVyd68009/PvvRDVr6uEoQFILI5ZcB5s2bYjatzcjCtQnjK22lurWzYwoNtvM  
DJSCvfYiWrzYjFi8/TZRw4ZmxJDq/2CfSIVXkM3+sMceRE89ZUYUeEcv73t9+xINH66HPcXfoPrf  
f0Tz5unAtXq1qqmq6tq1RKtWEU2fTrRwoZ4PARHTli8nwrfsVGRB7A8dkDms8+lpk3Tw/xibATX  
9et1GeCtt/QvmDtX/088NdGzpw4S8+eX/N9g2231799/61+GpyN/0SI9vGCBruhD/4H/hcZ0+eXa  
Ro82Ey1q1DADctSDy1i5UpePesJvTPL/WbZM+we+AhhGHq/nzJk6H0+bfPGFLg/+w3Tbj9jxsC78  
NAMv45w5uhwG+TDUDzsG74i8nRBYAOqO5exth/+B/xsGlscBCOOCy2K+N97Q6wvgGzz1gl+AOqD+  
7AtuT/j/HABsvwBML1eOaPZsvS0wDtD+8P84+GCd0UaxPMoF7FO0qzB23FEvz2Xy9uf/j7pjGvyG  
3yFDdNsAaHv2NuZ6Yh4eB1wmt1XUE75FPfkdqSgf4+YJoYCwm9rRHq66Sm9z/G/4/PXXzUQF/Aof  
oD3g/6AO+L/sf/iM68W/qcB0IDViBNHYsUQfflhy2Ux15Bk/a7fLLkQnnqgb5SGH6Dx25NFH61+M

cyPGMBopjqYIijxv5856ozBVqyYaeoOkR3F5GfsXCgU7L4YRVLE8T+OGyfMzrfFQQwiY77zziC6+mOjSS4keeyyhfrgMKCoE1Exgfqzn1Kl6+MsvE2Xg988/9cEC64+bnOGXHj30zgxq1yZ69109L3YS/N57L9E99+gdFOsA1YL5k/2P1wvus09iOZtWrfROxfB0/D77rA6qL72k1RVo3Fj7FdOhaAcM0MMmfEB0wgl63uT/YcPzX3ih/h05UivsK6/E84SJZfGLwPF//0d0/PGJPBsEje++MyMKe1n4E78//URUoYLOP/xwoiuu0Pn4X7/9pofHjNH+QdDhMsLgoMoB5/339e+UKXo6L4tf+JrPinr3JnrgAT0MMN00+vhFQOPxiy4iOulIPX7bbYlPOKDgzOeddxLzgjPOMAMhtGih1/+SSxLLbLMN0YsvEnXqpPM4qILy5Ylee00P2/8jmV9+IXr+eb2/Vayo9xGUi7wZM/SyTzyhf7Edb7rJLOgnadY0j+AlBUV5wQWJRswbBb+8Q9tBFUGCGwQ3zGS4jKFdiZo21cOMXT7/oiHjdAPDrFR5GqsKnt+GdwAbnu/IlxP1mzRJ//K0nXbKLqgi8EFiACz79deJMvCLctE4oejBNdfoHYsD/v776yBiL4Mdkw9IdlC9+urEPABPKSTnpYKnQ/ViGEG1bVuiAw/U+dh5eldnMlz6fvSRHp88Wf+GwUEV4BdBFTsylvHPIvj3zDP1/wOclwqezgELhgDBB1Z0B9WpU7wcDEOR8wM0CMZMcpvgolqzAy4Dw3ab5t+//tLDYWD6jz9qVYhbmB3YELAqV9bjCFAAwziIoR0AHJwZbPew9gsQVHnbcfn4RVuEKsaw/b+x7+LACDgPJJeP7YUDGtZ/4kSdd/DBROPH6+APNQ24jLvu0r+eYq2pRyBoosFCyfBpJxTlaadptYcd4/rr9dEdoL8FR2H0LaGfaslEnY9T/Jtv1sPgzsTChPI2/AG4180NOyg33yj86DiuG8O4xxUcVpm/w9ePhk7H4qJVSPAzoqA0727Po1sAxOP8OAmScLg3FKBmXH5eOXgzXUan3rawo4wiPYpvmkHreXQfDo00cPQ0ljvVA+gi/Pw2DHQ+C18wDGoTYEzrUJPchQu1ih4N6BvZ8PNysmVbUyUHGJiy04rQXB7/vvy8+DeCge9hhehj/mxUzQOCBGmLs/wdlyOMvv6x9iIMDsOfjYfSfoqsmORjY2Kf/n3yiy0TZDE9jBcvgwHDooWZEgWklqoAPXgC+wQHgVPMeD+TbQRU8+CBRIspaZDB8+m//T6Z5c92mAE9HV0elSomDgx1UcfCHigXYnhAUTfj5AIIKvrBful7thmse3LXE1z88JcWabclwxsQv9yGGgaM45sH0luoCSFmDU2DfQL8Z/AJL8exzmdC/vxnIAKttWKqumA3NK/rxbu+BzzglpjrBswYX1NjfuBiZLQj6pQH95kmP0vuG8sQmAj+OlpyuXRIXcoC9A3ZKkgQhE2WjTeoQrng6Ahw6lpa7FMmV1L1PSVz+umJC2yCIGzS+BIUzz1X9y9y4LT7afCLzuvkK978i1P1jh31LU7oT8S9keiL4XlwMYj7FAFus8K0n3/WfTkafta4yop8nP7jim6TJvoCCvLQd4dTX5SDC17ow+Ty8Ys+JvRzYRh1RX+pBFVBKAhMJPAMXFA56CB9kQqBCffJcdDCLVK42ISrw3Ygs3/5Pj/cNnXLLTqPpyXDFwJwsalRI52HC1olyPYyGIZBdeJCDi6gvPde4j5Hnjd5GYDbeCSocKJBUEUAj+B7/vBEC1/pRL8p8vC0D5g1S4/jCisHr3799DDucYO6xdXm5KvX+EXQZPiKN7jsMj1sXzFIWrbUV8NxZwHn8//GOx+4bxHYy+DWD4zjfkbcRsaIwiaPFQEEQRCEXcm8oGrfk+cKbufAc8d8470gCEISG1dQzfZqezrs0/MwcGEJfbrJ8+ECGG6UBpjGNyILgiBY+BIUt99eP82E4AXDI3B4mxCeY8c4v2gCF7D4zU0Y50cK8YYnvJABwzBcjMIVeNwvinGeP51q5fkYPOHBT3JgmV3yDUEQBENS5PAEBFVc2LFvhwJwYemFF4geeqj4s9clml276nG8B4CDKsDvp58mhsPAY4X2I53J8+GIGfYLUKQLQBCEEFJEmDwDVYq3



3gC8YARBjN+niFuZOODhF28/ArhvFON4Zv3hh3XfJ8+HZfAcMt4FwM/cY9qwYXo4DF4WL6TgYX50  
FS8CEQRBCMFECEQEBCEKJKgKgiBEiARVQRCECJGgKgiCECESVAVBECLEu6C6UKVFKgmC4AI4QRG+  
MMEfEfQIH+OFd0E1ZplgCHkETy/i9ZsnnUT0et/Efml/9M8D4vXyCK9q8/TTT8ed9N6z76ntGsFj  
qRHRuHFj1ZZidDx/jdMTzjrrrKBezFANII8444wzgnpVqVKF1q9f78W2RB1mz54d1At2CQKER5x4  
4on+bEvc640PSiqaHto0vl/Gwffg8sz9998fr9eXb3/pTbzwJqhyQ48nM55v4vVKsnX8ffo8ceed  
d4bWa98N+d2oFITVC5ZvwuoEyzdbb711aL3efvttM8cGxnzgcubMmfG6rKmkfpP3S/7NA1yPeLLr  
lWe8qMXdd9+d0km9e/emMWPGbFibpUyl6SrF9IX1CLPdY8E8QfpDWVg5UdtlZSbFdggpE2yvGP2m  
UjBXkbKwcsrCTGp0dqPweimbqFJ8zrAyySJKW1Pp19W/htYJ1uTqJlwr7eOwcqK235SpBJ/E9gmv  
V2wHq43NVxZWTsQ2ynzFYszo0FF9EONsw03eX3/9ReMmTqQp6uwyrJyytFtwTWoQ2BJ8eJVPE6e  
Z7wlqnEHHTgpL9ZRmUvCDhBWTImYS1qjLKyMsjDXFFZGWdhUZS4prlyysM2VuaTeySLKKQNTf+LD  
l5lxtn+U8bSayppa43kxTlZevsl/DRQvvfRSSid9hm9I5ZFysXKJuiVZPgmrD1s+abBdg9A6wY/5  
ZN2qdSm35bFHHWvmyg9hdWLLC/gum4LrcHc5VQ/URdmOysrhPRp6Bv27gXnooYfidYsnM/4jvg2X  
Z/K7ByZRSLFuJMqxSqZ3PxTo0aNxEZU1rx5czMlf6BT3q4Tmw+d9R988EGxOm2GN4h5QvXq1YvV  
bQTeZuYBdp1gefeZqgM46KCDdJ1Mqlq1apBPPXvqFXTlkQoVKsTrVTmmDwQ+4FVQBewk3+ipGhEa  
1w8//GBY/IF3RN/wtV61a9f2sl58Z0JlFA/NB158Mf5i+GCvLDI+g+8wzQOCeqnkE961LB+dBCSo  
uuNrvSSoOvDNN1TUuhU2pjZ1FuITPsYL71qWrzuiBFV3fK2XBFV3fN2WPtbLOy/5uvEkqLrja70k  
qLrj67b0sV7eeecnXjSdB1R1f6yVB1R1ft6WP9fLOS75uPAmq7vhaLwmq7vi6LX2sl3de8nXjSVB1  
x9d6SVB1x9dt6WO9vPOSrxtPgqo7vtZLgqo7vm5LH+vlnZd83XgSVN3xtV4SVN3xdVv6WC/vvOTr  
xpOg6o6v9ZKg6o6v29LHennnJV83ngRvD3ytlwRvD3zdlj7Wyzsv+brxJKi642u9JKi64+u29LFe  
3nnJ140nQdUdX+slQdUdX7elj/Xyzku+bjwJqu74Wi8Jqu74ui19rJd3XvJ140IQdcfXeklQdcfX  
beljvzbzqk8bT4KqO77WS4KqO75uSx/r5Z2XfN14EITd8bVeEITd8XVb+lgv77zk68aToOqOr/WS  
oOqOr9vSx3p55yVfN54EVXd8rZcEVXd83ZY+1ss7L/m68SSouuNrvSSouuPrtvSxXt55ydeNJ0HV  
HV/rJUHVHV+3pY/18s5Lvm48Caru+FovCaru+LotfayXd17ydeNJUHxH13pJUHXH123pY72885Jy  
UZB8w+ug6qnPfN0RJai6420b87Be3nnJRyBe++9N2jwP/30k8nxB1995mtQ3Xrrrb2s1z//BPU  
64ILLjA5/uBtG/OwXl7UpqioiB599FG9E3JSw88880wwLZ+sXLIS18tYuXLlqE6dOmZqfonXy/JZ  
vsH2OvXUUXN1MzZ27FgzR/4YOHBgiXq1bt06720M2HVCg8OvD1SuXJnKxVR9PGtjLHLser366qtm  
jvzixZZ77bXXtIOSnAT77LPPzFz5IV6vEMsnxeqS5LN8svPOOxevm2X5ZNWqVaF1gp188slmrwQ  
Vie2fFKsLh61sbgAC6nXoEGDzFz5I7/eMcQdFOKkvFhHZS5pX2Vh5Wwo4xQ2bUOZaworoyxsqjKX  
FFZGWdjmylxSb2Vh5Wwo4xQ2LZ8WUq9840VQ/fjjj1M6af/996emTZtuWHtcmUqnqhT7WtUjzAbE

gnmC1E5ZWDllaHF/hfjsxBNPDF2mTM2kzX7ZLNxfyk5SKT7nCcrCyona/lam0vHrjw+tE2zrUVtz  
rajp4crCyonazlSmEnwSVqfAvrTa2CfKwsopQ0vXxk455ZTQZTaE7brrrinrNX78eBNV8ocXQRWk  
cli+Qd/Ntddem6ibZflmypQp8b43Thjv37+/mSN/JPsKdtBBB5mp+aNhw4ahdcs377zzTmJbWvbv  
v/+aOfJHvD6c1DD6M/PdDx1WL5gPeBNuWZlIS+JOoiUm0xO6dOkSbLS+ffuaHD9Y88+ahM88Yt68  
eV41dGbdunVUq1atoF7Lly83uX6Ai3mo11InnWVvy/GDZ38u8bGOLFi1K1GuZyfQAv7ykiDvJM+Q+  
VXd8DKpA7IN1x9s25mG9vPNSrJdy0ZP+bTyvg6ryV+xxDxu8BFUnvA6qaGNq3/QNH+OFd17ydUeU  
x1Td8bVeElTd8XVb+lgv77zk68aToOqOr/WSoOqOr9vSx3p55yVfN54EVXd8rZcEVXd83ZY+1ss7  
L/m68SSouuNrvSSouuPrtvSxXt55ydeNJ0HVHV/rJUHVVH+3pY/18s5Lvm48Caru+FovCaru+Lot  
fayXd17ydeNJUHxH13pJUHxH123pY72885KvG0+Cqju+1kuCqju+bksf6+Wdl3zdeBJU3fG1XhJU  
3fF1W/pYL++85OvGk6Dqjq/1kqDqjq/b0sd6eeclXzeeBFV3fK2XBFV3fN2WPtbLOy/5uvEkqLrj  
a70kqLrj67b0sV7eeenXjSdB1R1f6yVB1R1ft6WP9fLOS75uPAmq7vhaLwmq7vi6LX2sl3de8nXj  
SVB1x9d6SVB1x9dt6WO9vPOSrxtPgqo7vtZLgqo7vm5LH+vlnZd83XgSVN3xtV4SVN3xdVv6WC/v  
vOTrxpOg6o6v9ZKg6o6v29LHennnJV83ngRVd3ytlwRVd3zdlj7Wyzsv+brxJKi642u9JKi64+u2  
9LFe3nkp1lw56TL/NI7PT1VQVfX6YalE1WzxtV61r1ZB1cM2NnuhCqqqXi2f8TCoqnrFzpc2lg3e  
eUm5KEi+OVMI1OsHIXzD1+Dla71qq+RjG5utEurVUiXfQL189JmPbcyr2lx22WXxjXdzm5tNrH9U  
uL2Crtn+nm1A06h8a1gVK1aM1+uff/4xufln6NChFJsfbEmqU7OOyU2wdMkSM7ThqbiD8hnSqx62  
MZN84qyzzoq3sfvuu8/k5h8vvLRy5cq4c+LjK9atcrMIR/i9eoW1lpix8SoXLlyNHDgQDNHfjj3  
3HMTdbOsZs2aZo78MGPgJMA/YXXLJOVFRYm6mKCK4W3atqMdFiykut8OpD1eeZ12H/A/OvaWztTs  
hhvowFNOob1+GEg7rl9PO6ky9vvqf7qwilnXq56p17t6/KabbjJz5lf33nsvUTdOZjyflF++PFGv  
JFu9erWZK394EVSPpfYhGOSnt55551HN998c16sR48eiXpZQZXzwpYpU7tSmUmxnqoOD4XYgzHq  
rFlw1zplYeWUhZIU+xV1ah1WL2W3qRSfM6yMMrQbVJCMb0sTVMvFqIG9ob/SjurAvYNqhx07dKAO  
7dvTjkuXUiM1DtteWaeOHemSs8+m7Sb+EVp2Lnbbbbbcl6pUUVGFhy2woi9cLxsmMd+/ePXSZMjWT  
jh9xfGj7gl00+6L4fPnCi6CabuN5YyFBNS/mktYoCyujLMw1hZWxocwKqtv2eT9QoduvW0cdEFTb  
tQuCKvJgDZUh2LY+40xq+Ne08PKispCg6o1xCpu2ocwx5Qsvgur7779f0nFm/OuvvzZz5Yd4vZKC  
arVq1cwc+YHrddj2teiTizanaw+rRzFz2p1PDj/88KAOVStWoicrbUV3VdqSKpn+1XyC00L2mX36  
3+Dqa2nb9/tRnf99S/tfcwPt9ezzdOiLL9I5SrEeeuRRtN+TT1G9z78IAuyB35RNI0+8XiFBNZ/E  
6wXj5EG9nn/++eJ1s2zYsGFmrzhRVBl6tWrR+VUUu6hnbZAM/aD4ALaLaiVSvvG6PHHHzdT8suy  
26rQk6fVpXV3VKX/Xbot/X1TTVq2co2Zmj8Gt7+S3qm0HRVVqUILqISnDyvVpR8efNRMzS9Nmzal  
2L9mW6qdMPki2m9jx5ihDQf6ey+44AKKbWnq9Zo6jb3ooiA/33Tp0kUHLJWwb7Zu3dpMyT+435j7

73fffXeTm3+8CqrANCsz5g8+3lK17s7qtLx7dVrbrQLR3RVpalet6cHHnjFT80ffipXp88p1qahc  
eaLyFeifKlvRkGOPN1PzT2lvqcoU4nINgXJLITusUH1CgmqW+BhUI/eoTh06dKT26lQV9sQVh9L8  
RSvM1PzR/8ST4nWCtbnYShpygz+3yJUmqMYGDzZDJYkNGUJ95s+nDxcsSDtfJiSouiNBNQt8dBLw  
7THV/p98RG+0qk83tGIBN7ZpHlj3M3en164+0cyRP97Yaju69qxz6Mbmql7Kbt//MPq8Yn5v9blp  
zWOqHCxvnD6d9hozhuqOGEHDly2jCbgdUE3befRo2n7UqCDA/lnK2wDIMVV3fKyXBNU8fHZ/37t  
dyS6q2rQpwq7q/UxtDz/Xar00pln0/qqVWltISqBDVRBbNT1N5ip+SeXoIrf+1Xwe0DZQ3Pm6LxB  
g2jI0qX0Qa5KVYKqMz7WS4JqlvgYVB9v3qBYUG1x/P40fMzvZmr+uPflJrSiarV4UH2zXj0aekV7  
MzX/5BJU206dSlv/+itVGTaMxiuVCjiofr5wYaBU/zD5rkhQdcfHekIQzRlfg2rRnVVpxe2b09Lu  
NQObcmMNeu3dj83U/PHZtnVpZbXqtKR6jcDG1NyChp95rpfaf+QtVe74ul/6WC8Jqlni5av/7q5E  
17e9SF8Q6tBBVblyDR/7h5qQ31tx3tl9L3q53o5BvXBD/ZwttqSfLvbrVhwhf25gEVXd8rJcE1Szx  
Maj2PqmyOteuQv/eVpXoHjXcw/JblzQ+vF1N++MTorvKEf38gMmMjl9efJWoahVaVqkyralYKRj+  
KsxvH6dQ1dj+f+DgYKFOsYP8CJCg6o6v+6WP9fLOS75uPC+VqiHUZxxUXzpUB9s7VQAFP9yuhtU4  
gjB+0wXfHAMtF8aPO45o1Cg9DLbbTg+XL4/HZfTwgQfq3ypV9DwYhv38sx7PAQmq7oRuSw/wsV7e  
ecnXjbdRBtWidUTd1e+z+yjbl2j4s3oa8sC9mylZ+YgeLgNC62WPY3ju3ERexYqJoDppks7jabhQ  
ZC+bAxJU3Qndlh7gY72885KvG2+jVap3q0D11XV6fPFMncdB9cGtiF48WA+XAaH1sscxcvGaN/u3c  
Wf9yUOXTf55/+HA9PH26Hs8BCaruhG5LD/CxXt55ydeNt9EFVQ/Iql6LFhHVMS+LLleO6MMP9XAZ  
lkHVnay2ZR7wsV7eeecnXjSdB1R1f6yVB1R1ft6WP9fLOS75uPAmq7vhaLwmq7vi6LX2sl3de8nXj  
SVB1x9d6SVB1x9dt6WO9vPOSrxtPgqo7vtZLgqo7vm5LH+vlnZd83XgSVN3xtV4SVN3xdVv6WC/v  
vOTrxpOg6o6v9ZKg6o6v29LHennnJV83ngRVd3ytlwRVd3zdlj7Wyzsv+brxJKi642u9JKi64+u2  
9LFe3nnJ140nQdUdX+slQdUdX7elj/Xyzku+bjwJqu74Wi8Jqu74ui19rJd3XvJ140lQdcfXekIQ  
dcfXbeljvzbzknJRkHxD66Dqqc98rVdpP1Fd1nj9NVVfg6pJPuGdl2JF/jmpqKilHnrooaBRjRgx  
wuT6Azcs1NMnfNyWYBuVfKsXtt0SIVCvVir5BgDvaWOZ8al22FD777+/3nCc1PDhxx+e9404c+bM  
eINiK4+XKXtAvE6Wz/INTleNGjVK1Ot5vNlvz3Tp0kXXa36iXugK8CFQBPWqZ+r1rq6bDwT1SrJ1  
69aZqfKB22v33XfX9eGkhps2bWrmyC9ebLk333wzsdEsJ8G++OILM1d+iNcrxPJJsbok+Syf7Lnn  
nl7Wa9WqVYl6WUEV1qxZMzNXfojXKymowvJJvF4hlk+eeOKJRF04mfHBOXwiPCry6x1D3EEhTsqL  
dVTmkvZVFlbOhjJOYdPyab7WKymoemMhQTUv5prCythQFIKHfONFUP3ggw9SOumWW26hPn36bFgb  
qkylj1WKXazqAbtAWXNIF5rxFrFgniB9riysnDK0uL9CfPbOO++ELrMhrH79+inr9d5774UusyHs  
tddeS9QrKagecMABoctsCHv//fcT9QoJqmHLIKm9rsyk2DmqDtz+k6yfSvE5w8opQ2vVqlXCZ5zM  
+OjRo01UyR9eBFWQykn5pn///nTPadvQiE61ad0dVanfhXXo2B0qmqn5pVy5csV8hnEfCNUWbdq0

MVPzx7HHHqvrIRRU8w26JoJtuZH0qf7+++9mav6l14eTGq5WrZqZml+8CaqMb1fz/hw7nj4+fXua  
fO0WQVD99rJtaX7nzWjNmrVmrvxjmpUXF1xs5Op/9mDbLVUJ9ZKr/9kjV/+zQLvln2pNmTyZXjmn  
Dl18QUs6+5K2dN3Fp9OnLbag9evXmznyj28+Y3ytl9yn6g4Hvd/QLcyvennnpdiiuykUH+VOtT/t9  
TI9evAe179CB2rdvTx07dqQ/O8Vo+fLIzo78A3/FDvCwwe+l6rWff/WqfYIKqh61MWb2PBVUVb1a  
dpGgmi2xXfyKF8A7L/m28XCf6pNX/B8V3VmV1qvTf7qrGrVvdqCZ6gfeNnhP6yWPqbrj67b0sV7e  
eck3J/01dSpdcccFtP7OakGfKt1Vma455xCv+pZ8bFjA13pJUHxH123pY72885JvTprx5x908QUX  
qKBaNR5UH7v0/2idT32qHjYs4Gu9JKi64+u29LFe3nnJNydNHjuc+l24FbVr357atm1L517aju2  
SrRmrUdX/z1sWMDXeklQdcfXbeljvbzzkm9Omn3fqBQbzzvCLqm3SX0+RUN6LcrlWIFn3ckemBL  
PRzGQ9sSDX2CaOfJqNs8LFhAV/rJUHVVHV+3pY/18s5LPjrp4w8+pAevOJZa7FWFnr/tEvP58K96  
Qv/riR5rqIf/nagC6JNEqxbp8VGvEHVX6zHmDale6nfsWzq/DPDRZ8DXeklQdcfXbeljvbzzkq8b  
L/R9qhxUZw8nurc60StH6QDKdDHD3cp2fXz1ma/1kqDqjq/b0sd6eeclXzde2qA66XMdUMefn+hf  
IEHVy3pJUHxH123pY72885KvGy80qA57hujDC9RAEdH4D4jeOI2d/i/U08BLh+nfoU8RfXK5Hi4D  
fPWZr/WSoOqOr9vSx3p55yVfN558o8odX+slQdUdX7elj/Xyzku+bjwJqu74Wi8Jqu74ui19rJd3  
XvJ140lQdcfXeklQdcfXbeljvbzzkq8bT4KqO77WS4KqO75uSx/r5Z2XfN14ElTd8bVeElTd8XVb  
+lgv77zk68aToOqOr/WSoOqOr9vSx3p55yVfN54EVXd8rZcEVXd83ZY+1ss7L/m68SSouuNrvSSo  
uuPrtvSxxt55ydeNJ0HVHV/rJUHVVHV+3pY/18s5Lvm48Caru+FovCaru+LotfayXd17ydeNJUHxH  
13pJUHxH123pY72885KvG6/nNBVUv1dBdaEE1WyJfapQ9Z1/9fl2qK5SQVW1sZa/exhUVb1ix3vY  
xjxs+955ydcA0VMIVTP6QSXf8NVn8BeSb3gbVH3+RLVJvuFj2/eqNkceeSSVK1cucNJpp51mcpPP  
yJEjKXajaVY7xWjChAlmSn756aef4o0KNmzYMDMlvYxYsCCoTzmV4LOPPvjITMk/Tz/9dDGfrVy5  
0kzJL6NGjaJYdXhLpedj9PPPP5sp+SWoF3ylErbn2LFjzT8s88++8TjxYUXXmhy848XQXX06NHF  
Grptf/zxh5krP8Tr0k0Z0jF6HN+ryieNGzdO1M0yNLJ88sgjj8QbejyZuuWTtWvXxuthG+r6xhtv  
mLnyQ7w+9ZQhvavHDzwww59CP++88xJ142TG88ngwYMT9UqyGTNmmLnyR169s4VKDVSqtbwWxWYq  
p4TY1iu3DuYJ0ubKGmw423777RMbLCmowsKWKVPbTJlJsemqDiH+is2IUUOVgrnWKgsrpwyUqVK  
CZ9xMuMNG6p6hSxTpjZDmUr1i+qH+0tZ1X+ran8hbaEsrJwyMvgk7q+koAoLW2ZDWbxME5mHPtG  
2DJlaiZtuXTL000I23b1tvH58kVeg6pyg1uyN/KGtpCgmhdzSWuUhZWxoYxT2LQNZVOVuaSwMjaU  
hQRVb4xT2LQNZY4pX+RXxxt+/fXXcCcqGz9+vJkrP8TrkhRUL7roljNHfiimoi3L9+k/bj3b2E7/  
X3jhBTNXfojXjymoos8wn+C6RrxunMx4Pvnxxx8T9Uqyv/4q2y8XZ4MXQZU59NBD4zvKSSedZHLz  
z6BBg4pdqBozZoyZkj+KiopowIABxRoU7qFFfr6ZP39+UB++UPXWm2X3JVIX0Odr+2zFihVmSn4J  
2lg1eEul52P09ddfmyn5Jd5/qRK255AhQ8yU/LPnnnvG40Xz5s1Nbv7xKqgCbuy+lbdUuaN3Rf/q

JbdUuePrtvSx7XvnJV8DhARVd+AvJN+QoOqOr9vSx7bvnZd8DRDymKo7vtZLHIN1x9dt6WO9vPOS  
rxtPgqo7vtZLgqo7vm5LH+vlnZd83XgSVN3xtV4SVN3xdVv6WC/vvOTrxpOg6o6v9ZKg6o6v29LH  
ennnJV83ngRVd3ytlwRVd3zdlj7Wyzsv+brxJKi642u9JKi64+u29LFe3nnJ140nQdUdX+slQdUd  
X7elj/Xyzku+bjwJqu74Wi8Jqu74ui19rJd3XvJ140IQdcfXekIQdcfXbeljvbzzkq8bT4KqO77W  
S4KqO75uSx/r5Z2XfN14EITd8bVeITd8XVb+lgv77zk68aToOqOr/WSoOqOr9vSx3p55yVfN54E  
VXd8rZcEVXd83ZY+1ss7L/m68SSouuNrvSSouuPrtvSxXt55ydeNJ0HVHV/rJUHVVHV+3pY/18s5L  
vm48Caru+FovCaru+LotfayXd17ydeNJUHXH13pJUHXXH123pY72885KvG0+Cqju+1kuCqju+bksf  
6+Wdl3zdeBU3fG1XhJU3fF1W/pYL++85OvGk6Dqjq/1kqDqjq/b0sd6eeclXzeeBFV3fK2XBFV3  
fN2WPtbLOy8pFwXJN7wOqp76zNd6SVB1x9dt6WO9vPNSrMg/JxUVFdHDDz8cNPiRl0eaXH/ghoV6  
+oSP2xJss8023gVVbLtly5YF9WrVqpXJ9YdgS6rtKW0sM97UBoOpME5quFy5cmZq/vj2228TdbPM  
B+L1MckHf4HkemG4Y8eOZmr+aNq0aaJuluWb9evXe1kvEK+PtS0nT55spuaPsHrVqFHDTM0vXmy5  
999/P9RJMAS1fBKvV4jlk2J1SfJZPtI77729rBetSr0tzzrxLDNTfgirE1s+KVYXj7bls88+m7Je  
o0aNmMnPljzy3dE3cQSFOyot1VOaS9IUWVs6GMk5h0/JpPtRrjqKXFFZGWdjmylxSb2Vh5Wwo4xQ2  
LZ8WUq9840VQ7dOnT0onffTRR7R06dINa6uUqQRiNVQ9wqxSLJgnSMuVhZVThhb3V4jPVq5cGbrM  
hrDdd989Zb3Wrl0bukyZ2nplKv236r/w7ajslPNP4S0ZXkZZ2DJlkq1VKbZZeL3gMz2XSmuUhZVT  
hhbfjrCkbQnCltkQ1qtXr5T1Gjx4cFC3fOJFUEXnd3xntJx0wAEH5L1jfNq0afENZpsPxOtg+Szf  
YHtVqVKIRL2efvppM0f+uPHGGxM+M1azZk0vLr4k1wvmA/H6WNtyzZo1Zmp+wPbaccddS9Tr6KOP  
NnPkf2vL+XVVtXAEQRAEQRDiyNB31/SpHHP0D//LaMx44bR3HEH0FuPV6ZxY7+isWP70U1t9R07  
PpwtlYZErV3qLwJeEARBEAQhnrWjtdPrt6y1p/rhd6dv3dqQUw6nVdMOoP5v7UijJ06hld+0  
p+WTG9O93ZsGC0WHr2LwK7pi13b0xzozKgiCIAiCsBHw+++m6GS5OOO/qITp1KVatVp0aJFwfgf  
f0ykHXZqRAsXLgzGsyEQq39O+ZM+6f8dzRI/Ps0e3oBW/rmTska0aurOtHja6fT5m3vQFrW2pvPP  
D3vjw3BqU+EkGmTGBrSP0YPKT59fciM9M30FTR/em158ayrRPY/Q0ae/Q1r/jaeryzWhAWul1vR/  
gLZs2IC2q9WQLnjgf2raQLqidgsaG8z3K3WqfAr9HCz0G10VO5y+XryQuh/3f1R3hwbUoF492uas  
HrRg+dpgblo3llqf0JVGr15Ab915KZ3X7EH6be1w6tGyM/1H0+jKCoFTryuJPrijcnp0KJy2iv53  
57XU46PptGr0g9Su+0AjnwdSp9060t9LB9JR5a8ydSEa2X0zuvYbtW6P96RP/5xLb7Y8gW5/7ixq  
O+wH6n7gyfTNajOjIAiCiAgFxWabbUYNGihtkmT1IFa5/fbbzVxlj29iFaxatYpu69KV6tarT6+9  
9rrJzZ5ArJ5+5qnU77M36lv/9adHnnqaej32OH367Tf08GGH0qvtLqV3r2IHbdq3CxYoyXC6ItaE  
ur3wKF195VX05Rzkack38sYYNe2zKhhGztpj/eiqTldS729epIti59IQITf3+5folhtvoLuf60sD  
R+OIdDpok17v0Zz1WGoWfdrzTzoWiNW/6bO7X6XJgS79h17v3pmuu/466vj+7RstdUFunYivXzX

m2pJRdEUerHbSzQtqM58+vKel2m8Wn78px/TT9OXY2aa/P1X9MOE+ZiBJqh6XdOpEz09ZCR9/9zn  
9G+w3AR65JqrqNOVd9AP//L/mU/9HnmVxqAIWkRfPP06jdAnDIlgCIlgCHnjt99+oxkzZpSwWbNm  
0ZAUF4bjv79+9Opp51Gy5cHginOmDFj6Ljjo/3tmYifs/qu2/1ol63Val+j29GV1xYg56+uzrN  
HVWLYpWL0V133W3mCmMlzZkwjZZpfRqwdsV/NHHM7/RfsYfb1tE/f4ym0WNgM/AOIUEQBEEQBFI  
S1ysColgCIlgCIJviFgVBEEQBEEQvEXEqiAlgiAlguAtIIYFQRAEQRAEbxGxKgiCIAiCIHiLiNWo  
adlSeVW5tV49k5EIIsvr5W66yWREAF64+3//R3TkkSZDcfjhRJUqET34IL7sTrRihf6/sHXriH78  
MTEOvv+e6M47iR5+WI9nQ8+eevktztZadhsMz3vVfpcf7f/8M7d7PgoIP0/OefbzliZl89dNmX  
X67HuW7vvKPH0/HWW4n5NxU+/pjorruInn3WZETleedpXx1yiMnlQP/+Cf9iX5s2zUxQPPKIzkfb  
8gm8ooXrnl5u3bT9/bfJyAlud+hQk5ECvH/RrsOLL+rhihX1eCYWL04sD/vlFzNBMXp0It83uF7j  
xhVfB1cuu0wvt912RK+bd0VyWZMnE82b5142zw/jMhnOnxO8E3LDgf/Xtav+31dcYTlz8NJLev4K  
FUxGCNgHunTRli2oC/shEzfeqOfD8RTgelfxs87S465MnEi0+ea6jHPPJercmahmTaKzzybiVzH9  
+2/iOHTYYUQ9ehDVr0/UqBHR7NI6HsS1k0/Wx1uhVGSx9YU4P/+sgxQaJvV16kS9eunp119ffNo+  
+xANGkS09dbF87EjXXCBFofYeffv/j01q2JPvkkleTYUJ79rrJtt9X5qQTuGWcQHx880WrrawXb  
b09Up44ZMSAY/POPHk4Wq9276+GtttLj4I47iKpVS8x33HHFI2GxCIF8xBGJaVjvTp30euOAcfDB  
iWmwFi30zvznn0TrgxftEn39dWJ6GktWEY0dS7TGvCeN523aVNe5QQOiKVN0EOFpsNq1ie67Ty8D  
EHCOPZaofHk9HT7idWSxCnH/xx+JuqUD64hAt2yZyVA88ID+v3Y9atTQ2xu8914i324TEBLwDf4/  
GDOGaN999UGB56Ivi+jxx/V0AD9zkGXDyQO/mBriHnkoA22Y5zngAKJddkmMlyuXOLC0aqXzdt9d  
j8O3GK9bt3hbrVJFiyCbq69OnJChTAR3/GJegDaK7Wi3VS7vt99MhoUtViHs+BeEiVWsn9oDLwOD  
7zt21NPhO87fa6/EMHyPgZxGezKK3UeT0f9ceDKBnyqetYsLWiAvQ6XXqp/lf44b8QIPR/aLx8M  
YTgBrVpVD3Mb5mknnZQYhiFGfPutnuezz3Qe/A7D8Oefa5FlxxVuV3Z7Ymyhx35G28Q+ESZW3IF  
tw/Oh6G9QTxgGVs82/so6nDMMQnB/uST2tc8HfWHEMmWBQul/vpLb4MwsQof88kpG2LAyy/r6WvX  
lvQtxwfbnzgWzJ+vjTn9dD0N8S4MLg9CCL/Yv2bOLD6NxeqvxxLtvXdiX4LBL2gTiDeA8yGQeBiG  
/ZVPZtAZsMUWxafvtltCXDHlz7Z9I+ZNmJBoS3wCipjHotFu30zfvtrXnH/qqYnh6dOLi9VzzkkM  
o40ceqjergCdKjwNhvWdOrX48cGVDh10WTj2MigPeeiQAB9+mPifDGIFxhEv0HZwHMD2uPBCM4Pg  
iuVdISOvvqoblAcmHDDat08EFgBhg2l8QAdvvKEPNti5bJFnH5i32UbnQeg995we5vnZ8P8gQLIF  
ZeBsF+AAiWCMg9d33+m8MJLFKs4SMQzBDU44QY83a6bHwQ8/FF8mrGcVByaexz4wolcMeTffbDJy  
hiUEDqQABw0EcuTBf7Y/MS8O1oDrZvfQIXgjj8VqLqCdoCxsA5uHHtL5OAmCqOB62HDepELE//2X  
GEdADGOnnfT05JOYo4/W+RdfrAUyhiGKGS4XYhgk9w5DVGMYyg7suKMehxBlwgl3TpAwjp4aBuID  
edgGpcEWemDkSN3ecADmqxssViGMMN6kiR5n0OaQj+385pt6mMUzgzzykiUJsQ6hYLcjGHzyqoAB

ifJtOA/rhH2Fx+0THxYa996rx3meYcP0ODjzTJ2HE0YGYoLnLQ3JQg9iEic4GOcTW5721FN6GP6x  
+eornY9tn6rXjPNwssrl4uTX9jn2Z5xYuZK8DhA0O++sx5NjBGi9xAnDQgpCmuGylHhLay+P2At4  
34Tw5Wnw04wZiXH7hNlulxCMPGwflzgGXnKJvkKC9cKJub2uWHf42ObRR/VyaN8o2wWcxGBZnFgz  
EJ9cP/Duu8XHAU4oOC9ZrNrwsRUf4wD3BvPxYoUbbkj8H/iHjyHwI2IHjr2IAXxyDJHNoE2hAwD7  
5Smn6CuRmAdl8rbiPKFUiOeyBT1q2KHR2NCziB2edwwctLIBtmun8yAKhw9PnFHiAA/RyoEdxj1l  
gHsfTjwR30Mj2mEHPY7L49ihuRwcABiuD19CT6ZxY91zhDM7BuVCGKUik1jIM2PszDjY45lwLovY  
y7iIVRYy6J21D7QMH9hg2cBi1e7Ze+wxnQcf9+mjewoxjm3CPQl3363zEJBwKQ5BHeMw9HqFwdPt  
g0lqEHB5XBGN339fi0mM4wCBHni0KS7ThvMgVrEtOahC6GN9uFfuqKP0AQVik3tecGaP/8WCDT0Y  
qAsOPhgPE6voIQNRiVWcEGAc2wYHYARtHBg5LxVcji0WmGSxyqD3mvNZrOKAxyeD6JmBP+AXjGMf  
wv6GdozxVGIVJz0QejhA4iD/9NN6X8S2w3RsP3t+nIRklhuxiu2JfRjj2N64BQU9dXzCnCxW7dsA  
wsQq2hDPI3bOvbw2PJ2vGNmE9UoCW3TwNPRmCu8e2tnbbyf2M9innxavj409D07KWJzjNhTUG5dY  
MY6rEQCCCOM4lctE2Drwicuuu2ofX3edHocotb9fnk6sovcevdTJ4OoWpu+5p8llgpdnsQpw7OB9  
BIZ9Fr2DfPUOMRcx7plnEjEPxyXAY0DcMixWISABemcxjjiA/YHXC72TNsjD5W8G47C5c01GGlis  
HnigyVAKi1Wc7PCVDFzZQhvhbQtzFavYBsjDvo8YsXKlmWDgciDCs+Gjj/T8aH+lh9zpAJ/wySnu  
LcFVGuQ//7zeDhjGiRQfX7DtEKfhd6FUKI8KGyU4+8zE+PE64L32mslQQDBBSGdD2P/ApUFcDrPv  
OWJhh8s9yWRTz2zJVFaU/ysb+PKsfckvKjbkuuB/ufo2m/rxPGgr6JFHbyrAFQXcpoE2gx6tMLlp  
n0k3r0s5TGmWsYE4hMhygf9n8v+Gr9BjA8GDE2aAS7h8SRxiJRuyXadc150pTTmpfJAtp52WEGwb  
glzWkSnturqQy/9AG7PFK41w00MEPwupKoDbmvAyZd9Dyj3QOKkMPm2BCaXdcL+yVfUNhQ4scWx  
GLE4CaVctQhkhWY9cQgs3LuKnRw9ujgrLO0Oj15SLI9LozAEGddLREJhgfaBNoPeNrRH3JtsX8oU  
0oPeRexr8B/231S3gAhCFGB/RXuzbz0pK3AcQjxAXEAvNv5nae8x9RGsC3wpx8icELGaD8Lu08Fl  
fvQyhT3UUNZ8803ikgouo+Eero0RfpAhm0uBgiAlgiBsFlhYdQFPjOPBIhaaMNwHCYHJvZS4BwtP  
/Nn3G+GVFfxE8wsv6Dzcc8b3neGhHn6a+pZb9HwA91riQS0uB/ch4gl/XDoBvAyeiOR7kGBYxr53  
jfNxx1cyt92mp3FdYHgKeeBAXW8WsTA8Ecn3B33wQSKfn6Tm+3GvuUbPM3hwYh6Ae49wHy3GcV8l  
entxvy3fewvDvVh4kh1vSQD2vYm9e+tfvsR6//2JJ7xxzxruN8OwiFVBEARB2GRQR3YhayCulJRw  
ozRulscDRCyk0MVv5bDvsQOoYp7VfjSBt+0DWP4ARB+/Q7fk4Z5bVjYQSZyq4PsJ+nxwA3yYNmC  
G+ztZfjhAghLG7ybFfl4hxzELC9jP2CEd7rihn+ITQhJCF7cal4Hz/DgAuZHzzKAj+AzCFbcB2W/  
4ovfs4peX86zn9Tnhwrs97HiwSXkiVgVBEEQH0GdWQXsgY9ixBDeDgET9HipnAWUhCyyOOnX3FJ  
GoKNX1KMHll+ITBeechL4VVEIfmsci8iRCteNYWHePjpXjzRj3vXlBoxbvfGholVFrj2e0VtksUq  
nr5GDyd6W/HUNO4jwlPe3PuKtxqkekofT3fyO/PQ84unRvn1JzC8t47v3eG3KeAdobhxn5/Sh4WJ

VRusM/Lwv/CgE57UZIEvYIUQBEEQNhmSFICw0ZDNw1GYJ5v5bFznB+mWKU15qXApK8r/KwiCIAhC  
3hCxKgICIAiCiHiLiNVNHX4HKr/IPVfwomncInDRRSYjBfhMIb8wHYZbCPClIDyAJgiCIAiCkCui  
VI3gB6ogAFkE4v1p+PIIvtLBwgxf7uCvh+DNAMhr00aP4yEhng8v/MZ9rfyNfH5RMe4P5a8DwXBv  
Kl4qzPd64ktPPA1fucEvf6cYDyzt5/Owz2u+PQbhu0vZ/Cy9hsD0oFvL+OeWnxZhO8LxReRUoE3  
HuAtAvanYfEQFpbDfazJXxURBEEQBEFlgVIPQtbw5zLxAA9eYwXw+UYWf8mGVysB9GpCpP3zj36t  
FMbxhPy11yaeqseT9gAPXOEb8a+8ol9dxd85h/FT9Hi1FOfZbwvvgp+lffdVkkLh+LFYz3cuZ/Gqu  
5LcR8Gdf04lVrDfmwSfqGAhzLpMfNBMEQRAEQciAUg5C1rBYxaudAIQfv84Jn5PEOHOQ8R1sCFK8  
ZxTgyX30MkKwoscRn0HFE/K4NI73sUlgcm8jl4dL7RB19gcE+DvYtli1uewynYf/j69KodeV3zJg  
96zusYc2fOvclTCxim8jozz+9jV6lfHpVdwG8NxzOo/fZIDXVMmXPARBEARByJlktSMlgiAlgiAl  
/iBiVRAEQRAEQfAWEauCIAiCiAiCt4hYFQRBEARBELxFxKogCllgCllgLSJBUEQBEEQB8RsZoF  
MSvVVkkQBEEQBAW+ZnjYYVRUp446WCpJYdvBh+gP5nz6qZ4303u+CxBbX+ykkhCOiNUssBtTXZWE  
9BSpgLRu3brA1q9fb3KFdLC/YEImpl25I23MHfFZCqpV0x+54XeOG5qoZB8vV6gUBx+JmT8/8TVH  
IcD2124qCeGIWA3hm2++USeFqumw2WmOMpP/888/myUKG4iFY445JvBJuXLIEn4zxnm77rorrZRP  
rQaMHTu2hJ/C7P333w/mhzgrVHjdr7zyyIAfwbiNVahQQR0P1QFRoJkzZ5bwT5jdd999wFzSxoie  
fPLJUB/BbB9OnTo1mL9g4Lah1p2qV9fDikWLFIGNGjW0f2LlqNxASR8rlaumfdaqVSuzhKJvX100  
rAD57LPPAp/EzU4TlZn8X3/91SwhABGrFq+88krXsRmJ0ussn3yySemhMID65/uQJhsPG+hkq1I  
TbYuXbqYEgqP7bffPtQnqazQ29iqVatK+CQbO/74400Jhce5554b6pNMtgBfJ9zUYaFapYr+XLgB  
68/72uvKBpczAtSyV5V1V8bz1qpVyytWdzt2pmRTZ+HHnoo8EMJs5Mlvtl++uknU0Jho1qLAJYs  
WVKikcTNTiFitWCsg7KyTHsrC/u/G7MVKsurtEZZ2P/cmO0gZWWZLIUW9n83ZpuqrCxT2P/cmK2G  
srJMTysL+78bsfVTpgZK5MN+UqYG0lPbZcnLbaYM04638grS7BQiVmGCairmV1B89NFHoQ2IWAoR  
q7Vr16b69etv2lZP2W3KrNRQpXKz1Zn1LOUHF5sZC5a1ywpSU2Vh/3tjterKipRZabs121FsRohP  
Opny1+ZLNqcGKtl1V+rDP8j7H9vrHa6MiuHnVT8p2K4X9KZ8nEJfyFdrwxtOex/b4xWQ9lOZVaq  
V1TPvY0pqzq/arjPNlCW9r83VttNmZWwztUXVA/1SVpTPq67vq5VkkmvKdtGWdj/3gitlRL+DRsS  
lStHm1WuTPXVclMGDYodA9WfIDZSGc+H23QaQuVRxq6YXr06XaDKCvu/m5ptscUWcT8UMzuFiFVc  
jRNUUzK/guH1118PGkixS9t2MmKVp3/44YdmycKD7/M68cQTE77KYAcffHCwTCHfH/fjjz8GvijW  
xkKMpz/88MNmycKD28mll15awj+pDLcNLF++vKDb2lwZM2jLLbcM9Y9t3MauMw/KFKLPeJ1vvfXW  
Yj5JZ9WVYPrjjz+C5QqG668n2mYbolmzCB7Dswq77LJLcd9MjtEznWLU9kU1jFTN5Ctr0aJFUEzg  
7c8+02L23XeDvEKhd+/egS+KtTE7GbHK0/v372+WFESspmHy5Mk0bOCwRENSqfbq2rTgzWk4T6mU  
Qli1a9cusSMqe/TRR81UIZI58+bRqFGjivkLNn36dDOHkAzu4XrkkUeK+atDhw5mqscwCMNDjSNH



jizRszNmzJhgulCS4cOHU79+/Yr569hjj6U1a9YE0wv2RAhvRVAnjjR0KNExx5hMzZBvhtB+i/ez  
jpYx+vrnr81UwwEHED3wANEbb+jxAj6hndBhAg36aZDlrRg1XN6QFs9YbOYQbESsZoHdmOTVVZnp  
2bNnsSD/ww8/mClCKmx/wYT0oMfB9hfEq5Ae3K5k+0xlz+zZs4v5q2XLlmaKEDBliu4dZbviOmpC  
J1K5olhgSCuqWtNhQgm0p3SSV1eIRlpPFsS+Uc3oB211fxGxmkgkRq+7Y/oIJ6RGx6o6IVTdErLqz  
/8wTKDZO+Qs2WonV9dZ7VoVQbH2x2zARq6mQiJUFdsCqW1fEaiZErLpj+wsmPEfEqjsiVt0QsepO  
kyZNivlsxQoRq5mw/bXbbiJWUyERKwvsxiRiNTMiVt2x/QUT0iNi1R0Rq26IWHVHxKo7tr9ErKZG  
IlyW2I1JxGpmRky6Y/sLJqRHxKo7IlbdELHqjohVd2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xlj4hV  
d0SsuiFi1R0Rq+7Y/hKxmhqJWFlgNyYRq5kRseqO7S+YkB4Rq+6IWHVDxKo7Ilbdsf0IYjU1ErGy  
wG5MIlyZl2LVHdtfMCE9IlbdEbHqhohVd0SsumP7S8RqaiRiZYHdmESsZkbEqju2v2BCekSsuiNi  
1Q0Rq+6IWHXH9pe1dRixMoCuzGJWM2MiFV3bH/BhPSIWHVHxKobIlbdEbHqju0vEaupkYiVBXZj  
ErGaGRGr7tj+ggnpEbHqjohVN0SsuiNi1R3bXyJWUyMRKwvsxiRiNTMiVt2x/QUT0iNi1R0Rq26I  
WHVHxKo7tr9ErKZGIIyW2I1JxGpmRky6Y/sLJqRHxKo7IlbdELHqjohVd2x/iVhNjUSsLLAbk4jV  
zlhYdcf2F0xlj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWFlgNyYRq5kRseqO7S+YkB4Rq+6IWHVDxKo7  
Ilbdsf0IYjU1ErGywG5MIlyZl2LVHdtfMCE9IlbdEbHqhohVd0SsumP7S8RqaiRiZYHdmESsZkbE  
qju2v2BCekSsuiNi1Q0Rq+6IWHXH9pe1dRixMoCuzGJWM2MiFV3bH/BhPSIWHVHxKobIlbdEbHq  
ju0vEaupkYiVBXZjErGaGRGr7tj+ggnpEbHqjohVN0SsuiNi1R3bXyJWUyMRKwvsxiRiNTMiVt2x  
/QUT0iNi1R0Rq26IWHVHxKo7tr9ErKZGIIyW2I1JxGpmRky6Y/sLJqRHxKo7IlbdELHqjohVd2x/  
iVhNjUSsLLAbk4jVzlhYdcf2F0xlj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWFkQ20U1pAO11T1ZxGom  
Rky6E9s30cZi+8tumQkRq+6IWHVDxKo7TdoqscpxbGclVleJWM0E/MQ+2+1cEaupkliVBaoZxVNd  
IYT0iFh1h9sXJyE9IlbdEbHqhohVd5qopLwVTytUETj+2s3IYRwJGKI4bbbbqPG9RvHGxJSub/L  
0cPdHjZzCMIss802VKtWrWJBvlq1amYqUVFRkRkqbNgP77//Pu24+Y5WC9Op1WmtgulCSRo1ahTc  
jmO3sQoVKsQvOUob07Af/vjjj8Bf8JHts0MOOSSYLpQEI7N32GGHYv4qX748jRgxlpqubUxj+2Hr  
KltT1WFVrSgWo/o71zdThWQ6duxljXcsri/KTypPLz/ycjBd2lhxRKwmceihhxYLUiHZaY4ya9op  
p5xilixMli9fHvihXLlyxfySztBjUchceeWVJf2SnKxp9erVM0sWLVCDsXv7/vvvzZKFySuvvBLq  
IzBjvxY68INLG3vsscfMkoXJr7/+WtlvA5XZqZoylS9tTLPrrrsW9xfMThOVWdMuuugis6Qgrcew  
du1aatOmTbGGEjc7JYIV2C233EJvfxWpm/DIFnpE5UqXlqRYhpcP7hY81iwrF1WkL5QFvZ/NyF7  
8MEHS7SfwJIT0vTzzjuP3n333dAyNynrr8xK/VSqf0v98HaUzs6M0fsq2WUFabCysP+7sdqrypYq  
s9Lra18P1j/UL2nsglcOoA9VsssK0mvKwv73xmrV7K7PSByod2ftlil0Y7peUpnz8wriXrJMMmqTs  
TWVh/3sTsY8++qhEjAoshVhl23rrrenjjz8OLXNTtnfeeYcuvfTSYr6Im52SxCqsa9euRqUUNiJW

DYsXLw4aRuiZtZ1CxGrBWEdlZZnwKFHY/y0ES05h8xSCHaKsLFNrZWH/d2O2qcrKMoX9z43ZNldW  
lqm3srD/WwiWQayKhZidksSq9EgnEC9YvP7668UaStzsFCJWP//8c1NC4eFy2cy2QmXcuHGh/iiR  
kqb36NHDIFB47LLzLiX8kY0VKqtWrQr1RyY7+eSTTQmFBx6eCvNJluwYIEpofAI80cmsVrlr358  
9NFHi/kibnYK6VkdNGiQKaGwEbGaBN/UvOOOO1LNmjVLNqbZMdoqtIUwHchN0BrSiPvvv3+JHY1t  
v/32o+uvv97MXdhwm9I5552pTp062kdJqUqsCu20007BfNLGNN999x3tvffeJdoW2x577FHQgsuG  
28yBBx5IjRs3DvUXDO91nDJIjSjBvoQOfzZ07I3bZZZeUJ+GI+/vss098foHo3HPPpb322kv76Adl  
diofC/zVr18/M3dhw20G7ah69eraZ3YaH6NtKm0TPEQKpIOELGaBVZTkldXZYG8usodbl+chPTI  
q6vckVdXuSGvrnJHXl3Iju0veXVVaiRiZYHdmESsZkbEqjvcvjgJ6RGx6o6IVTdErLojYtUd218i  
VIMjESsL7IAIn1vNjlhVd2x/wYT0iFh1R8SqGyJW3ZHPrbpj+0s+t5oaiVhZYDcmEauZEbHqju0v  
mJAeEavuiFh1Q8SqOyJW3bH9JWI1NRKxssBuTCJWMMyNi1R3bXzAhPSJW3RGx6oaIVXdErLpj+0vE  
amokYmWB3ZhErGZGxKo7tr9gQnpErLojYtUNEavuiFh1x/aXiNXUSMTKArsxiVjNjlhVd2x/wYT0  
iFh1R8SqGyJW3RGx6o7tLxGrqZGIIQV2YxKxmhkRq+7Y/oIJ6RGx6o6IVTdErLojYtUd218iVIMj  
ESsL7MYkYjUzIlbdsf0FE9IjYtUdEatuiFh1R8SqO7a/RKymRiJWFtiNScRqZkSsumP7CyakR8Sq  
OyJW3RCx6o6IVXdsf4IYTY1ErCywG5OI1cyIWHXH9hdMSI+IVXdErLohYtUdEavu2P4SsZoaiVhZ  
YDcmEauZEbHqju0vmJAeEavuiFh1Q8SqOyJW3bH9JWI1NRKxssBuTCJWMMyNi1R3bXzAhPSJW3RGx  
6oaIVXdErLpj+0vEamokYmWB3ZhErGZGxKo7tr9gQnpErLojYtUNEavuiFh1x/aXiNXUSMTKArsx  
iVjNjlhVd2x/wYT0iFh1R8SqGyJW3RGx6o7tLxGrqZGIIQV2YxKxmhkRq+7Y/oIJ6RGx6o6IVTDE  
rLojYtUd218iVIMjESsL7MYkYjUzIlbdsf0FE9IjYtUdEatuiFh1R8SqO7a/RKymRiJWFtiNScRq  
ZkSsumP7CyakR8SqOyJW3RCx6o6IVXdsf4IYTY1ErCywG5OI1cyIWHXH9hdMSI+IVXdErLohYtUd  
Eavu2P4SsZoaiVhZYDcmEauZEbHqju0vmJAeEavuiFh1Q8SqOyJW3bH9JWI1NRKxssBuTCJWMMyNi  
1R3bXzAhPSJW3RGx6oaIVXdErLpj+0vEamokYmWB3ZhErGZGxKo7tr9gQnpErLojYtUNEavuiFh1  
x/aXiNXUSMTKArsxiVjNjlhVd2x/wYT0iFh1R8SqGyJW3RGx6o7tLxGrqZGIIQWqGcVTXZWE9PRU  
yfbZDyoJ6bEDFkxIj4hVd2qrpLwVT0J6Zqtk+6ulSkJ6mqhk+2yFSkJ67DgmYjU1ErGyIFbEu16M  
6qkkoCBIrIei9EQIYT0AELJoRTVFQU/H7//ffF/PXkk08G+UJqtlZJeUsnFdMA+1NIWd75TyU7  
9l+kkpCeY1WKe0zaWfBYcWz33Xc3uUlyclS04J2qVatWxRpQsTRHmclv3759MH+h74wLFy6kbbfd  
NvBJuVg5inUzvjKp3LEqT02rUaMGTZo0ySxV2HTp0iXeJtLZmWeeGcxfyG2M132PPfYI9RGsXDnd  
xmBffvllMH+h89JLLyX8g/1yvhq2k5I23BFH0p1vvEPn/PQzNR05io5XdsGgX6i9OiHo+MNA2v+d  
D2nfb76jtiqv3eAh1PaXwdRq4I/U4qdBdO7Y3+jM0ePolF9H0OnDR9CFP/xIN3zQI15/+12aNFIP  
UxP/4TZW7DJ2vcBL8VTuvUQbk5MjzcCBA6I8+fKBT4I2NjDhryBVU6amNW7cOJhf4hhR8+bN4+0o

nV1//fXB/IJGxKrFm2++GdpoiVLrLJ98cUXpoTCo3LLyiX8kSxWY8coS5qnUPntt99K+Clbu/PO  
O00JhUc6kZrOCpVVq1aF+qOkWC1HNf7vEKrz9ge04+rVVOd/39N27/ej+sNG0SEPP0qd2rahDuqE  
vEPbttT0/vtpx6VLqdHatdRozRpqpObfSf0f/O6g/ufRV19LF192OZ3c+nJq/P0PtJPK2+bHn+j3  
2XOCOvnORRddVNJfSWI19q6ypHkWLFhgSig8kn0RWAqxytawYUOzdOHx+OOPF/NftvLL7+YEgob  
EauGJUuWhDaUwOwUilbFkiwLsVowZl1GjDytURb2PzdmO0RZWabWysL+byFYklgtH6tBDSZMCISI  
bQ0XLabLTj2VOnToEFw96tCuHZ14T89ArCbPG8yv7KQ2bamjmf/KNm1ohy++ooZz5IDF7bcPr8vG  
YFmIVbEkyyBWC8rKMB2qUqEhYtWiX79+mRtdiFg999xz6aabbio4696901WoUKGEP7IRq926dQst  
c5OyTsqqKlFnpsvmXUexe5YOHLeul7AFI9xt7UNIDyqzP/xyBN2ikl3WTeuU4X+E/e+N1Z5QZqWu  
KtV+tXZxf8Hgn2Szp/eM0a0q2WUF6R1IYf93EzJcPkze3wlr0bNanrY45mSq/91Aqv3EM1RnwLfU  
YOIkavjXDDrswV505WWXUcdAfF5BTe++h7Y3vajbK4NAXTB+MX5E59vonFaX0eldu9EeH/UNRGy9  
YcOpzc2qzYbU0Se79dZbw3vvsxCrV199dWiZm7qlvI0pg1jdfPPNCyP2X6vMSjerdNyl43Rst+NU  
JrsvRhfnvsgqSacnVCo0RKwmMW/evGCnsu+BK5aMWOXpuF+z0HnmmWcSvoKIEatdu3Y1SxUuy5Yt  
o5o1N6edt65G3U9sSH9cX5tePrsu0T1VaHn36vT2+XWpT8v6dMMx+j7gadOmmSULI2+++SbwRast  
t6K7qjeiCZVq0cTKm1NRIsoUdiW9FbFevRkrYZ0YqXqdGbz881Shcu6deto3333je93gYXcs1rO  
TLv3o3504ONPUe3nXqRaen9u8uTz1OqVPnTp62/Rrk++Tge//AZd/tgzdPmTz9Fljz9L599zHx13  
axc65IFHaM/n36Ltn3mOdn7uFTr+yRfp2ldepb4jRtDMefNp/fr1pkb+w7foxGN/GrG600470Zo1  
a8yShUuJ2yfSiNVPPvnELFW4zJkzJ/BFMX0RYjwdV3wFjYjVLFDNJ57kbQCZuVcl22c/qSQU595W  
TWi1EqafXlyPBrbZhoruqErrjBXdWZxeb1mP/rhuK1rUuSo99+q7ZqnCZeWSpfRZ7W3o7Ur16N8q  
W9DgylspoVqZ1iqxul7ZiMq16ZvKW9O8SjVo2OlnOoKIEuSTKfY2AJWE9Pyjku2vC1QSOnOMSrbP  
1qgkpMcWqfI2gNRlXMoC3vGQ5D2rmZH3rGbmnpuuopVKrL7WcnvqfOFxdNb5F9CpF1xBp7e4jFqe  
dw71vmx/+uaKOvRDq6o0bPR4s1ThsnL5Cnp3933ofxVr0ZuN96BLTzuTTmt1KZ3epg2dceHFDNWx  
J9LAOg3ok/Lb0k+XtKYVq1ebJQVmQ7xn9cq//qLYTz9RtWHDTE72vDI/PtUcPpxiP/5ID82ZQ4vX  
raOvFi6k/osWmTk2LPKeVXfkPavu2GJV3rOaGhGrWRBroRrS5drq3iBiNRM9P1Vi9VLjswuVWJ0o  
YtVm4ZKldMe119KSbpXp6kuaUYeOHYMHU2y7rG0H6tN+LxpwUSX65MuBZsnCZeasmfTxvgfQ69vv  
FDz4ww//sOHhnnPatqUJsUr0eYuLaIVcoi1B7WuUWDVxLNaybEL/zdOnU2zwYKo3cqTJOQxeupRu  
nTGD2k+dSg8qlfqPtX2mr1pFN0ybFiwHazZxlnWfOZOOGT9e5w0ZQrepZWdt4BOQ2QuVWLVif8tn  
Raxmosn9SqxyG2uuxOoaEauZELGaHSJWs8BuTPK51czl51Yz89Sjj9JnHRRSD9fuSGtur0oruhe3  
ld2r0K0t/49eOqsSLVqxzixVuOA+3+fPOJue22MvWIVV+UjZ8ipV4rZSjS+oWJFaHX0c/XrtdVS4  
b3NMzYb43GqyWP1RiVSMVxg6lGr9+ittM2IE1UDvqrcbcdSoYB4mNmhQYMPUMuDDBQso9ssvwzb5

QD636o58btUd218iVIMjYjUL7MYkYjUzIlyZ88j991Gfy3egV688kujOKvH7Vdn+7bIF3XxJU3rp  
zGo0d8Eys1ThsnrVKnrm9GZ0lbLV1ZSPIDjF/apsVKkSfda4MbVteiL9es0NZinBJh9i9Tu8EICN  
HzhuHP1telOnrV5Nz82dS58kPZzKYnWIEasfiFjd6BCx6o7tLxGrqRGxmgV2YxKxmhkRq5I5vHN7  
6t+plT3X6ehQsTqv65bU7dKmNLrjZvTz8LFmqcJI9fLI9Mpue9HVZ5wV9KyGidVPdtmVuh57Ao1r  
cSEtX73KLCKwG0KsfqwE5jV//UU9Zs4MxrmHe6ASrbi0f/OMGfTSvHm0LOQtAR2mTg1uE5imTkzA  
uqli6qvK66yWuVcJx/lr1wb5GwoRq+6IWHXH9pel1dSIWM0CuzGJWM2MiNXM3HvJEbT+9qp0X7NG  
1Oqii+yyyy6LW+vLLqebWp1CK7tWpqW3VaHeL/cxSxUuK5XY6bdllfq1Sg26ZL8j6bK2baj15cpX  
l19Olyk7p00bGrzINIRUuTINPvV0+k9e+VKCDSFWNyVERLojYtUd218iVIMjESsL7MYkYjUzIlyZ  
88yLr9OKLPWJ7q5KL1zQmM5reQF17NCeLm/Tli4+82SaeM0WypFV6N6m6temi9pLYTMGmYwsWbea  
aMANRPdvTvTURkQLpxEVrde2EdzhuXrNGvrkoktoXYVKtKJqdep00JHUSonUDu3a08UXX0KP77wn  
raxajRYo+/ye+81SaVDtMrC//jIZWcDL/PGHyUgDeg49e8eoiFU3Rky6I2LVHdtflIZTlxErC+zG  
JGI1MyJWs+PnX4bS492up7duO4+uPbQK3XD4ZnTDEdWo08FV6OXbLqHHHrifli5Lul+1q9plk8Xq  
+rVEc8cRTRtINP0nogV/mgmGRUqYDriJqIda7q5yRP+Mlfp7hNpQVYluV3mftS+5jlfMnT+fenW7  
gx4+5Qx69sDD6OpyVejGitXo6vJVqefu+9ADZ55HDz3+pJnb8M8/RF9/TfTtt/hoPtH33xMtX67W  
+3Zt9muRFi8mwv2RPykfwu94/dKQlfhSiJ6u2nJgU6ZowYp2PUhth7lz9XTw77/4CHhi3I9/Lf4/  
8oiIVTdErLojYtUd218iVIMjESsL7MYkYjUzIlyZ88j991Gfy3egV688kujOKvH7Vdn+7bIF3XxJU3rp  
g+juCkSjXjELKca9S9RdzXdXeZOhuG8zLWCHPW0yNh769+9fzF+PPPKImZLE229r0QhTB1I1oxaP  
nleeVdwysPfeevyhh/RyH3xAVF75CnnPP6/zeJlevfQ4xO8+++i8c87ReWDs2MS8HiFi1Q0Rq+6I  
WHXH9pel1dRlxMoCuzGJWM2MiFV3bH/BUIJMrK4n+uhios5qfMoAM4Pi3wIEM9X0pXNMhmLUqyXF  
6r1GrA5+1GRsPJRKrNpwHsQqekMrVdLjEYfq6ehlrVhR5yWLVfs2gKOO0nnNmpkMBXpneV6PELHq  
hohVd0SsumP7S8RqaiRiZYHdmESsZkbEqju2v2ApeWArZVuql+lwK6HAJfwPLyB6sjHRCwcR/fqM  
mWDx23tEVBmerShyVD8N5notWOIHqIHNRXk7lXkLVY/fhjQDSiekmfSa5RQ9uMGXocvdQPP0y0  
555EBykf/ql8mixWa9bUhtsAGPSO1q9P1Lq1yVDg5fUPPki0xx5E//d/ROP9+AKZiFU3Rky6I2LV  
HdtflIZTlxErC+zGJGI1MyJW3bH9BRPSk7VYzQYI1X79iHbZRYtTNghXCFhM3wQQseqGiFV3Rky6  
Y/tLxGpqJGJgd2YRKxmRsSqO7a/YEJ6lhWrBYKIVTdErLojYtUd218iVIMjESsL7MYkYjUzIlyZ  
sf0FE9IjYtUdEatuiFh1R8SqO7a/RKymRiJWFtiNScRqZkSsumP7CyakR8SqOyJW3RCx6o6IVXds  
f4IYTY1ErCywG5OI1cyIWHXH9hdMSI+IVXdErLohYtUdEavu2P4SsZoaiVhZYDcmEauZebHqju0v  
mJAeEavuiFh1Q8SqOyJW3bH9JWI1NRKxssBuTCJWMYni1R3bXzAhPSJW3RGx6oaIVXdErLpj+0vE  
amokYmWB3ZhErGZGxKo7tr9gQnpErLojYtUNEavuiFh1x/aXiNXUSMTKArsxiVjNjIhVd2x/wYTO

iFh1R8SqGyJW3RGx6o7tLxGrqZGllQV2YxKxmhkRq+7Y/oIJ6RGx6o6lVTdErLojYtUd218iVIMj  
ESsL7MYkYjUzllbdsf0FE9IjYtUdEatuiFh1R8SqO7a/RKymRiJWFsQeVA3pCW11XxKxmgkRq+7E  
Hk20sdhjsltmov80JVYfNv5SvntkpljVTNR+XoIVbmO9pl1lYvYyJVat2N/ySxGrmRCx6o7dxnZ7  
Q8RqKiRiZyFqRvFUVyUhPSJW3eH2xUlIT3+VbH89opKQntoq2T4T0jNbJdtfLVUS0iNi1R1uX0i7  
qSSElxErBYsXL6Yzzjg2OHKqRRvTjNi1PqM1rRw4UlzpwCWLvsWBCY7UAW+K6d8p37/+++wKdC  
gn///Zfuuusu7auktE+9fWj58uVmTgGsXr2a1q1bp/3VJGm/vDhGo0eODtqZkABt7LPPPtM++zfu  
rcB3lWKVaO3atYFPBc369etpzZo1tOWWW1Jsy6Q29nyM+rzYh+bNm2fmFsCCBQto8uTJuo1ZxrEf  
bWzlypVmbgEsXbqUjjrqKO0nu42NjdENl98gcSwEEasWs2bNojp16hTb4QKz0xxl1rTGjRvT3Llz  
TQmFx5tvvhn4gQNTOU5nnrqKbN04bFkyRI6/PDDs/imRDL58FmlSpXojz/+MCUUhGMDlj7lu6v  
kwMvJdKNyniasltuucUsXXhAcLVq1aqYPwKbr8xOJp/9OnjwYFNC4TFixlhivgisXuClRHpXGU9T  
du655wYnUIXKHxfUcwf6Yz9ihOnQmXq1KlUs2bNEr4pliYqs6btt99+wcmAlGI1DnoFt99++2IN  
JW52ShKrsAMOOIBOPPHETduaKntCmZVOU6n8N+Up9j/IBxfrH6NTVbLLClJ7ZWH/exOyUKEKS05J  
03FSdNJJJ4WWuSnbqaeeSptttlkJf2QSq7CTTz45tMxN3U444YQSvggshVhl23rrremUU04JLXNT  
NrSTOE6KDGIVduyxx4aWuUnZ0coGKbPSKSrFvIQ+Clvvaazqj1XDY3/Y/93EDA9PJbefwOyUJFZh  
u+++u1EphY2IVQN6vJlbSdzsFCJWC8Y6KivLtK+ysP9bCJacwuYRS1gWYlUusyTKIVbEky0KsFoz1  
U1aWKex/ForZKUSswgQRq8XA5cYKFSqUbCx2ShKruERbyJfP+L4bWLnYFanuVjWp48F16LOltqTv  
L69Jj5yyHTXZaSuqXLV6fL59993XLF144FJQ9eoJX8QtOSVnf/XVV00JhcfFF19cwh+ZxGrDhg3N  
0oUH7rIMfk1VYBnEaufOnU0Jhcfdd99dwh+ZxCrua0UnR6Gy0047FfNHNNbOOeeYpQuPdZ74oPht  
Jmx2ShKrVapUod9++82UUNilWE2iqKgo+H3ggQeC7nc0mGI3QM+M0YE7HUj3339/MB/PX8jMmjWH  
hnQ/hP68bgt6p/l2NLhdLVp/Z1Vad0dVmnVTTZp6/eb0Qcu69F/nKjTo41cL3md2G8NI7hIBS6Xa  
sdr08MMPB/NJG9P89ddf9PTTT2t/HZG0X14ao+5dutOLL75o5i5suM089thj1KITJ+2zeXFvad+p  
vCeffJIWLVoUzCtQ8CDQ448/rkXFZklT7MUytW7RWvbLJF5//XW65557dBuzjIVZ7969afz48Wbu  
wobbTK9evYIT6sBPdhsbF6Mm+zYRfRGciNUsiDckleTVVSWZPPoXGtw6Rv0vrUM/XLEdUc9KtLpb  
RVrRtQLRnRXp96u2pLm31aahbbeg124+I+j5EYpjtzEkIT3y6ip35NVVbsirq9yRV1e5w+0LSV5d  
lRqJWFlgNyYRqyX5a9p0GtV1T7q+xTHUoWNH6tChA7Vv3z5uGO+orF/LyJT4+fkBDEEU40hCekR  
seqOiFU3Rky6l2LVHW5fSCJWUyMRKwvsuU8+t1qSB665iF5ttRN92npb6tr6JLr9ipOpx+Unxa17  
m1Op98V70mPn1qe+7fekX4cNNUsKjN3GYEJ65HOr7sjnVt2Qz626l2LVHdtf8rnV1EjEygK7MYlY  
LcmsmbPozWuPousuOZXonkq0/o5qwf2qbHRXVfqsft26+fxDaMKAl81Sgo3dxmBCekSsuiNi1Q0R  
q+6lWHXH9pel1dRlXMoCuzGJWC3J5Inj6b7zdqGjDjtCCdPKtP7OJLF6dxV6okUDuqTJLvT1Cz2C

I5YLxbHbGExlj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWFlgNyYRqyX5c9wIGtKuOo28sibN67IF/dl5  
W5rSeZvA/IQ2Vdm4a7agZV0q0rt3tQ4+vycUx25jMCE9lIbdEbHqhohVd0SsumP7S8RqaiRiZYHd  
mESslmTK5En04zU7EfWsSk9dsge16XhN4uGqjh3pjHMvpLV3VqP5N1ek717tGd6zmvNDVxv3Q1t2  
G4MJ6RGx6o6IVTdErLojYtUd218iVIMjESsL7MYkYjWcDx66hdZ0r0h0Dy77V6XVt1ellT2qEd1Z  
xeRVpjG37kZjx08ySyj6X090h2qCj5XiBe53ISPqqpYd+gTRivIEv79H1EWN99g4m7TdxmBCekSs  
uiNi1Q0Rq+6IWHXH9pel1dRlxMoCuzGJWM1Mz549i/nshx9+MFOSCBOrP92nReh9NYke2DJ4YCsQ  
pgv+1NP/GU30+I5KBKt5YPdWJ3rpMPVPISjmvN57Ec2foOffSLD9BRPSI2LVHRGrbohYdUfEqju2  
v0SspkYiVhbYjUnEamZKLVAHPU10uxp/tAHRhxdq++gior6XaLNBlyps9nA9/vu7RN3UOJbfCLH9  
BRPSI2LVHRGrbohYdUfEqju2v0SspkYiVhbYjUnEamZy6ll95wyiW1XeM3sT9TmVqNfWWoRO/9HM  
YEgWqxP6EXVX4yiv78VEi6br/I0E218wIT0iVt0RseqGiFV3RKy6Y/tLxGpqJGJlgd2YRKxmJmux  
KsSx/QUT0iNi1R0Rq26IWHVHxKo7tr9ErKZGIlyW2I1JxGpmRKy6Y/sLjQRHxKo7lIbdELHqjohV  
d2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xIj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWFlgNyYRq5kRseqO  
7S+YkB4Rq+6IWHVDxKo7lIbdsf0IYjU1ErGywG5MIlyZl2LVHdtfMCE9lIbdEbHqhohVd0SsumP7  
S8RqaiRiZYHdmESsZkbEqju2v2BCekSsuiNi1Q0Rq+6IWHXH9pel1dRlxMoCuzGJWM2MiFV3bH/B  
hPSIWHVHxKobllbEbHqju0vEaupkYiVBXZjErGaGRGr7tj+ggnpEbHqjohVN0SsuiNi1R3bXyJW  
UyMRKwvsxiRiNTMiVt2x/QUT0iNi1R0Rq26IWHVHxKo7tr9ErKZGIlyW2I1JxGpmRKy6Y/sLjQRH  
xKo7lIbdELHqjohVd2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xIj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJ  
WFlgNyYRq5npeY+IVVdsf8GE9IhYdUfEqhsiVt1pcrSIVVdsf4IYTY1ErCywG5OI1cz0VEI5K55+  
UElIj93GYEJ6+qukW5dOj6gkpefEqhuzVeL2hdRSJSE9TVSyfbZCJSE99j4pYjU1ErFSUFRUZlaK  
N6b69eubXCEZ9hmEAwJVklpiNFolIT12G4MBuw0KGvbJLyqhbXF6TiUhPdtss420sSxgn0Bo2W3s  
UpWE9ByvUtxjyndA2lhJbJ/Y++See+5pcoVkrKyGULNmzWINKJVtu+22ZonCZty4ccV9002ZnY5R  
ZqZ9/vnnZqnCZu+99y7usxRWrlY5YP5CDvhY9zVr1hT3zcnK7HSjMjPtjjvuMEsWNhdfHfxn6Ux  
UOhtDBTzSz1ldnpXmZnWrFmzYP5C57HHHivus4HK7FRNmZm2aNGigm5jTOXKIYv7LIXttNNOZgkB  
iFi1ePuNt0MbTSbr/3l/U0LhUaVKIZI+SSNW2QqVieMmlvBFNnbP7feYEgqPvfbaq6RP0ohVtkJl  
3ap1JXyRjZ1x6hmmhMLjksuKemTNGKVbcmSJaaEwiPZF4GIEauwHXbYwSxdeDz12FPFFJGtDR00  
1JRQ2BS0WfVNoezSBGUhDa8gLAuxWjBmXUaMPK1RFvY/C8GyEKsFY1OVIWUK+5+FYFmIVbEkyyBW  
C8rKMB2qUqFR0GJ1pEq4nxJpjEr3fn4vxXZXTWE/B9s1Rk//9HS8nHhapWyEstGbrk2bNi38kkYW  
YvWwv/4KLXOTMmz/pPTRnx9RrJHyQVhbSmV7x6jTc51onEp2WaOLIG3ibWzKlCnUqFGjEu0nG7E6  
YcKE0DI3KcP2X63MSsNXD6fYDsoHYW0pjTW5pgn9oZJdVpA28TaGdnLaaaeVaD/ZiNWffvoptMxN

3SZNmITCF4FIEKt4QHnq1KmhZW5SNkqZlcaq1PmtzhTbU/khZN9LaTvH6I0xb1gl6TRJpUJDbgMI  
oWrVqsV2sFS21VZbmSUKm1GjRhX3TRqx+vHHH5ulChs89VnMZ2IMIFq5cmVxv6QRq127djVLFTYt  
WrQo7rM0JmiK+SWNWD355JPNEoVNr169ivssjVidP3++WaqwwXMlxXyWwho2bGiWEIBEqRTYN4Lb  
DUjeBpAa9pm8DcAdu43BgDyMUBL2ibwNwB15G0B2sE/kbQDuyNsAssP2ib1PytsAUiNiNQvsxiTv  
Wc2MvGfVHbuNwYT0yHtW3ZH3rLoh71l1R96z6o69T8p7VIMjESsL7MYkYjUzPR9SYnUz5a8tIVVU  
YvVnEauZsNsYTEhP/2+VWK1k2phqa488LWl1EyJW3Zg9x3zBCm1sCyVWLxexmokmpymxCn/BIO9W  
rBSxmg17nxSxmhqJWFlgNyYRq5np2VM+t+qK7S+YkB753Ko7IlbdkM+tutOkiXxu1RXbXyJWUyMR  
KwvsxiRiNTMiVt2x/QUt0iNi1R0Rq26IWHVHxKo7tr9ErKZGIIYW2I1JxGpmRKy6Y/sLJqRHxKo7  
IlbdELHqjohVd2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xlj4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWFlg  
NyYRq5kRseqO7S+YkB4Rq+6IWHVDxKo7Ilbdsf0IYjU1ErGywG5MIlyZl2LVHdtfMCE9IlbdEbHq  
hohVd0SsumP7S8RqaiRiZYHdmESsZkbEqju2v2BCekSsuiNi1Q0Rq+6IWHXH9pel1dRlxMoCuzGJ  
WM2MiFV3bH/BhPSIWHVHxKobIlbdEbHqju0vEaupkYiVBXZjErGaGRGr7tj+ggnpEbHqjohVN0Ss  
uiNi1R3bXyJWUyMRKwvsxiRiNTMiVt2x/QUt0iNi1R0Rq26IWHVHxKo7tr9ErKZGIIYW2I1JxGpm  
RKy6Y/sLJqRHxKo7IlbdELHqjohVd2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xlj4hVd0SsuiFi1R0R  
q+7Y/hKxmhqJWFlgNyYRq5kRseqO7S+YkB4Rq+6IWHVDxKo7Ilbdsf0IYjU1ErGywG5MIlyZl2LV  
HdtfMCE9IlbdEbHqhohVd0SsumP7S8RqaiRiZYHdmESsZkbEqju2v2BCekSsuiNi1Q0Rq+6IWHXH  
9pel1dRlxMoCuzGJWM2MiFV3bH/BhPSIWHVHxKobIlbdEbHqju0vEaupkYiVBXZjErGaGRGr7tj+  
ggnpEbHqjohVN0SsuiNi1R3bXyJWUyMRKwvsxiRiNTMiVt2x/QUt0iNi1R0Rq26IWHVHxKo7tr9E  
rKZGIIYW2I1JxGpmRKy6Y/sLJqRHxKo7IlbdELHqjohVd2x/iVhNjUSsLLAbk4jVzlhYdcf2F0xl  
j4hVd0SsuiFi1R0Rq+7Y/hKxmhqJWfKqM60a0t/a6s4VsZoJEavuxOYk2hiGhfTOX6XEKvtM/T6y  
TMRqJkSsuiFi1Z0m/ymxynFMHTdXFIIYzYStL3abJ2I1FRKxskA1o3iqq5KQHhGr7nD74iSkp79K  
tr8eUUIlj4hVN0SsutNEJeyPnFaoJkTH9tduKgnhSMRkoqioKPh96qmnaK+99gqCVDmV4s1pZowO  
2eUQeuKJJ4L5eP5CZ8iQIfTyty8XC+6B78op36nfZ599Nrh0KyTaDnrQmWeeqX2VILYuvzU988wz  
wXzSxjTTP0+nV199VfvriKT9sIWMet7Vh958800zd2HDbea5556j66+/Pr4/svF++dJLL9GSJUuC  
eQWiVatW0QsvvBD3D/+ytWvXjnr37h3MK/ul5r333qOHHnpl++hbZSYF+6fKwz47adIkM3dhw23m  
6aefph122CHwT7E4Ni5Gxx94vOiLEESsWnz77bdUsWLFEGCKm51w6dGaVqVKFRo2bJgpofBo2rRp  
MX9kYwcfLBZuvCYNm0a1axZs6RfklPS9EIWYVdccUXgg2LC4WRldrpRGU9T1rhxY7N04bFmzZrg  
3nrbH+mM/dq9e3dTQuHxwAMPFPNFNrbNNtvQ0qVLTQmFx+67717SLwOV2amaMpXPfr3gggvM0oVH  
v379wtuXnSYqs6ZVr16dxo8fb0oobESsGtC7YDeSYmanJLFaUNZRWWmmfZWF/d9CsOQUNo9YwjKI

1YKyqcrKMoX9z43ZNldWlqm3srD/WwiWQqyKpTE7JYIVNkHEapzly5dTnTp1QhtKsVTIYrWtMjsV  
WcOuKWzZ3ZWF/d9CsOQUNo9YwkSsJmyKsqRU7NJirinsf27MVkVZBCmlj59UFvZ/C8FErLqbnULE  
6o477mhUSmEjYtUCI2hxaSe5sRRLSWIVDWnOnDmmhMID93fBD9lcPuN5evXqZZYuPBYtWhTcBpHs  
mxLJ5MNnuDXlt99+MyUUHh9//HHcF3F/ZRCr11xziVm68Fi3bh21aNGimD/SGfu1kB+E/OWXX4r5  
lhtr1qwZrV692pRQeHTu3LmkXzLcBvD++++bpQsP3Ldbo0aN4v6C2SIrOK5mfz55sSChsRq4lg  
CllgCIK3pBSreArNfhJNnkoTBEEQBEEQNjRxsfrf/L/p6Z41iebsTJe23MnkUtB936TJOWZMEARB  
EARBEDYcMe4xHf7tpTRz5kj6d9Ea+vKTLvTnD5Xo+y+vp99++5n6vtuGxv7aJ5hvk8fqQJa+ZEEQ  
BEEQBHeyuSKf7VX7oGf1hON2psWTDqHpf6+gr7//mT7teydN+fUM+uSrH+iff+fS4oU/0zGHxmj+  
/P+ChTZGxvd7gLo++S2tzuix+fTQ/x1PPceuMeOCIAiCIAj+s379eho6dGjwftbff/+9ml0dO5Zm  
zpxp5twwNGzYMHihfTLPv/ACNWnSxIxIjHCrq6YfTd+/X58m/7wrTR+6PS2Z2Zym/bldfT2gF81b  
sliGfroVvdxrM1q+fFGwUFkRTU+mKiVeUGIKXEJPH3sWPfj7mFiNpoaCIAiCIAhRA7H6xx9/mLHi  
rFixYoOL1bVr19JNN91MTz71VHz8+BOa0ldffRWMZ0sgVtu02kuJ0S1p3YxdaPmkRrRs6mG0fHIj  
Wj2IPo36amuaM2JH+vy1LWjlimSxuoZ+eOhy2vPINvT4w72o/fmn0UNf/Umjb9ubjnrQTDHrMd3  
osN7/UITP+9Ozc7rSo8/+gBdfMxB1ObFX9XUIhr6yVP05tu306E7tqUJavzDK06i3c+4mXo/eCed  
e9qr1LrL/fRlr/vp3GNPpGeHzVPLjKfHmrelXi++Ra9d05buHr0q+D9gcM8D6aZvlMhc/BYdXnNf  
eupPITn6LtrzhkE058eHqNMN/dR/+J5Oi9WiC+99nnq3O4C2PKGPWosi+vHWk+n4Sx6lZ57sRkfW  
OYAe+n0trfjrQ2rT7BK698HH6Y5OJ9EtvctP6+cSDs/Mk0VPJGuPmBfuvCLxer/jaRLuvamFavX  
B/UQBEEQBKEwiPJyd674JlaZZ597js4973xq0fICGjToF5ObPbEIS5fSDz9/RwO+6Uv//b47LZmw  
A638s5GyHWnt36fQxJ/3oWvbNaLKVWvR4sVKmBVjDX3b6xbq+aF2zPxRb9G1j72rpN+v1O2AzjSd  
htGlp75Oq2g5PX37/TRoFsTuOhry3O3U/aURRKvm0avtm9AWdbajrSofRJ+uJPrgisvp0V/xrepV  
NOD2G+jeT6cEywx9+Xa67dmRSiM+QjvEqPd7RtQ/YY70lkvJj5FNn/UM9T1wQHU7+a29OrQPnTz  
Ta/Rs+3b0OuziP769hG6/rZPVd0GUpvazWIMsMQI6ljpeBq6cApdcO5jNDfiW0wvnnAePTJ+PU18  
oyt1fuxHtZaqNv9No3vvuZYW/fMFnXHUvTR40Bf07hcP0ikH3ks/DX2LHnr1W1onHa+CIAiCUHC0  
bduW6tevTw0aNChh+MT2hsJXsTpo0CDafY896cSTtg7eae9KbKESoOMm/EJ77LULPf/KC/TLkM+p  
S7dOdOpZR9He+9SmM5q3oFNatKD/ffONWcRmDf346MV0SKePgrEhT95G3V8fQuhfHNvNjprn2Oo  
+yC80HYffXTjKdTuFV3Be0/flVr3mUTrP7uC9rrye8xNJ1U6jPopsfrhFa3p3oHoQV1FX3a/mm7/  
aJlaXke/PN+NbntGCdx1/ej4GgfSIKuTI09Y1i37h564bFfa87rP1dhK+qTzoXRg589pjZr+x9cj  
sXr5VufQ8LVYYghdXqkpDaVldMeudajrcJW1bAQd02BX6vX7OprxfXc6r1VvzEgLf36l2nR6U62J  
WqrbPrRny240Wq3amHsOo7Nvfpi+mrA8mE8QBEEQBCEf+ChW77//Aereo0fwwZKFCxfS4UccSSNH



jjRTsyP22OOP0+DhA+iOe7rSR599Rk89/wzdc/8j9Mlzvenhffak/l/3o3uPPYp+HDLELGKzhr5/  
tBt17nIX3XjDDdTz2bdp4To9ZcX0IXT+ddfR0IVFSiAq1syiJx/oTjffegvdcXUXuv/DsSpzKX32  
4FV05X0v0+evvkTjVxON//Rj+mk6hN9amvzdV/TDBDzUtZ5mjviOvhmunbxg4hfU49pr6cbrr6WH  
vp0V5GnW0rj+H9P3ZpkpP39B/YfNCP7//CmDacA32IDT6JNe79Gc4Ir9LPqk55s0DcPLRtFtN19P  
XR97nPo+9R4NnofMtfTrR8/SLTdcR7ff/wHpGxtU7twh9M4bA5XEVSWaQa8//QUI9zkLgiAlgiBs  
SCBWR4wYeyJSGTNmFLMpU6ZscLF6+unN6lknjRjCdQ0bUfXKh2XLcE9q3vssQ9deVktohn1aPWc  
U2jF5Lo0eURL6nD9hXTCqadR2/Zt6bXXXgsWKM4a+vbBG6nr66PNOCiipfNmU/9Xbqfen0zWQlWx  
ZvlCmj1zBv31xzhqccwe9O7v8gkxQRAEQRCETZHX10FzjopRu88WY06d9qMrr68Bg3/rCa1b7VF  
8H1afM88nCJaPn8uzVuUeMgJPZrp4yl8ZP/Du71ZFYunE2/jxtDY8eOo2l/L4uLWEEQBEEQBEFI  
RSBWMyndfUUmYAlgiAlgiDYxHtWBUEQBEEQBME3RKwKgiAlgiAl3iJiVRAEQRAEQfAWEauCIAiC  
IAiCt4hYFQRBEARBELxFxKogCllgCllgLSJWBUEQBEEQBG8RsSolgiAlgiB4i4hVIYF8/EEQhEOZ  
iXGCsFEiYtU3og6mHToQXXAB0cqVROvXEz31FNG22xI9+qiefu65RPXqEV18sR5v3Jhom22IXnhB  
j9esqW38eD2eDZtvm3MGJORJW++SbT11kQ77GAyMrB4MVH16kTVqpmMCJk7V5cNA59/TISrFtF2  
2+nxTGC+rbYi+vJLk7EJwP7491+TESHYhih7yRKTkYHtt9ftuF07orVrTaZhyy2JttiCaNgwk+EJ  
p52m97W2bU1GCNgv0Xb22stkZAHv07vtZjJSEBZbatfW/vr+e5ORgdNP1+tw/PF6H7HZe29d94cf  
Nhme8OCDul777KPHer3atNHjLvz4I9Ghxl1aED0v/+ZTA4pW5domuvNRkZQJxFvEXMGzHCZBpu  
vFGX1aSJydiA3Hab9InfviYjCxC70Za+/dZkhHDEEXqdunQxGVmA9ccy8Ec6ktv3smU6puR6fMD6  
nHgiUaNGRC1ahMe/P//U0zAP5v31V52/dCnRmWcSde6sx4VSI2I1amLKpbBZs0xGFgwnlguSvr1  
02X276/Hp0/X4zho/vCDzjvmGJ136ql6HDs3xh97TI9360bUtSvRP//o8WzA8jCsVzrGjk3MC557  
Tg9XqQTHM7FwYfHlo+Tvv4uX/cEHbv+L5/3oI5OxCYADDAwHgahhfy1aZDIyULFiYhk+8WI4/+ef  
TYynHHKlRhdOEFMxcCDRnXcSPfSQyciC++/X5UKgZwlnnpi3Y0c9zr766is9ngkINV6mfXuTaYDw  
Rf4dd5gMT+jeXdcLgh7wOpxzjh7PlnXriMqX18tC6P7xh/YBxnFiCiDYMd6qlR7PBOIs5oftvz/R  
f/+ZCYorrtD5u+9uMjYgn3yS8FO28Hp88YXJCAEnVnfdlX6eZLD+KBf+yATXydQofeLL46WIZ0+9  
PIT77bfrkxGM8/ETfPihzsOxE6L0kkv0+GuvFZ/+yy96XCgVyoNC1uDMDTvAvHlavCGwrFmjp6F3  
B2dcaJSw0aOJVqzQy+AAjF6IGTP0ssuX62XAqlVEX3+dWA7TUCZ6QRcs0MthGZyh2eD/zZmjLRU4  
MOFMIpkYRf+PiRNNhuLxx3Xeiy/q8WSx+vrri2iAMGdQP64q6YRh+mD9fry/gdcHJzfF68DTwerV  
eofneSGA0LuDs+BrrjEzKeBnCEfbZwzKw8Hh2GP1OHyeWHwH/4v/g/A8qgzyoNfceBJBnXAMphv  
9uxE3QC2J3qL0/WK2Vx6qT6AjRtnMhSoE9oCfIH/A79x/QCvK3rB0WuMccwXJg6Rx+0wbH2wXbj9  
wLD97Hm47WA+IIOTK8wPUPbMmXo5e5sh+MJQP5TFZaCNY/ujLvi114nBtsH/gWFZ+BiG/w/QTg84

QJcN0N7hC8wfBovVcuX074ABZoKct5stVvE/4QOUh/XCuvL/BtxuUHdsl6wL5k1ud5huryvWK1sg  
QFEvPoihDPxP+Bj/E/7ACdxbb5U8yYE/uN4ca2AgWayiXrzN0eYAthF8gB4qzMtC86KLdM8exwSs  
H/siDBYw7PeXXjITFGFiFf8XbRm+DPMZ8vG/sH68jrDkHnasM/YXTMN6h8WDVEAYVaigT7pBKrHK  
+1Sy7wDqbMcEnPhjm7GgRE8ruPtUPf7OO3oc2PtJMrZYhV11VWK+VGIV9YQP4M9kX3DcQD62Jfsd  
62P/f6wb8rCumCfsikafPvr/ZwuuzqF3dclEPc7rjf+F7Y59+9NPid5+u3hHBuqFuqCeaANYP8yP  
PMBiFScImJe3EeZl0LZtcfrrdd7q97Lcf0dFHm5lKAZc3cqTJUKAXfOedzYgCV9wwzzPPmAwF4j9O  
bBj06u+4oxkRSoPysJAV2NHQGwHBwg0YttNORFOnalGCcTRQPvvGwQlnY1WqFF8GBwf01AHu/UTw  
5wMAAt0JJxRfBsKHgy1AQOBpYeDyDaYh4DAsVu1L+ghoOEDygTtZrPL/4Ev6uNxx4IGJfAxXraqH  
cbkN8LSTT04Mw3B2iiACELSQZ683LplPmlQ8+LlvcUaeDAIXRBaCF8Dy/L8QLPALAYzgCBHM02AQ  
Rvbl40ceSfQ6wex1BAiG6ElBIM0GBE6sCx/wcPBALzb7F4Z1w2Uj7tXi/D32SAzDUC/OdDC4xGjX  
Ff6DYOf1gUhDgEcPNc+DbYSD9LRpeh7ORwDI4cqV9QHc7rXcc0+93vYyOKhwW4ldfnhiGKIAwhNi  
lxk0SAd3nue44xLbnEUmDj4QA8ytt+rpEEBhcB1bt9b1Rq8Z2jHg/8NiFet82GGJdgrD8rhs/tNP  
eh7Ox8GNh2HY7/hy75Ah+v/Y0+G/5Mu3qYDgsfc1LgP7Ov5Pw4a6ZwZ5di8pelrtdsO9OzDAYhXb  
+8gjE9PgF8QstAu0RxZpbOefr/e1yZMT7RT7J08Pg8vA/sG9iLhFbiSLVZR51lIEm22WKBNWpw7R  
q6/qeSBukAexZ++jijk8D9pa8j6Bml55RU/PBNb/t98SJ0JhYvXJJ4vHabRPtGuG46kd31mksOEw  
KrR7HAvs2MLTwwQhi1VsK9zSgmG+UhamVq+7ruQxCGOFI+0Bl4G2gGMTz4P2jn0KQNCed15iPWA1  
apQ8EcdtWZiWLVg/uy1x2VwntG20cQzj9jTmwgsT+zPwbZdddBzZdVc9ncUq/MsnWzDs71763lw  
soU8+3hyzz1a8PLxoTTw/+IYCOBX3O6AkxWAFrfz2O2Rtx3H2+ef1/62T6oFJ5Q3haywD858HyLO  
pCBAevTQ44DnQWDEmR8OkjgoMBBJml4DDgOxy8sB3JuDYezE772nAykovjC+HwiCAZeaUl1uuuEG  
XQbOLhkcVJHHZ75hpBORELZ82wB6AMDLLyDE0QMP6DxeBsEPB2b4AZe4klf7eRgITZ43FVdeqXf8  
bO6pg+jj8iACcTkGvSo8/vTTWrzjMizy+NYHPmFA0Gf/tmyZKCsKTjIfI3XSSYmDJg4wyMP9lziA  
8P9r3lz7DWK8WTOdx4Gbe+ewPrwdb7pJ56GdoOcNwR7j3HuGctAGkQehCDAM4x51bCuMY/tzrxHP  
w/et8TiEpd1m+f5mPmDA3n9f56GdstBArxPg3icY36KSDLYJ6nTzzSYjCT64oRcSvXsY5nv7uGyl  
VRw40bOCcdw7hpMOcPXVOg8HQvial0FbA8hj4cdXJyDGclDEZeV3300laohe+NgV3m8gktAmcTKK  
226QByEEIngwXr++FkGoP05MkAcDLFZRDh8MuU3D7BMstDXk2Se+Njgo4wTgsstMRHls9BCbcBKC  
YQgliAIWnixW+Xlobn/AyRpAXIEQgSjB/2LxcdRRiVjF2wYiFzRtqschvNGucFKMcFw/tENXksUq  
YhvG0fuFtowrSXxCgJ5nG+TBGLRPjEPApAI9jjDe721YrGK/w4kD9muM46pTp056mMXqvfqcqQF  
nDgBzldOAORDLF1/vR5G++GTKOwjyIMB3FOM4bPP1h0jzz6rfYI9CgKVYbHKJ1eu8P+EsENbwEkN  
C2isG+DjFI4raNullei5RB7EKWCxihPJv/7SebzvoS3ZIA+Gq5gg2/0S7fONN7SohD/4ygZODIAe

7nFGhwb2U/xPHCv46gN0AOZBm4ZARmcCn/jwITXETIyj80ooFcp7QIZATKAHBA0VjQ72f/+XuKTH  
8DRc/gAlprh8gR4ZnPHx8iiLwf01vBzAwRHDODCit4ENBzcEnmzAzd5cHuADGs5y011CSydWITp5  
3L48zL18yWKVbzlHeKgBefYlGfR28LxRYItVhsUd/G77Er5FgAZ85o9eRAZBMbmsXOCykg+unl/e  
bh6GEGI4mPNIRm4bELRh2OXYQAJY+TzMAgFBFON2TybPw4Gbx5PFqg3noTcbJN+XzHBeKrGaCRar  
OLgAFv7oheSylVbRw8XjuLRow/kQ1DxsxnRBkCEPvYi4blneJ+zDye0I83BvkgssVu37U3E/H/JY  
rOLkDuN8YgXwkA/yYCDsnlUcpHke+3lp2hHy0K5KAws9vu8W/sc4xAj3oLJYxTAs+UEkPpBDeEIE  
YJh7BgHElVlga8Rd7pVPjoeYzr3pLiSLVb6nFW3KLh//Dyd+DMQU5oNx/GNhne1DocnYYhWgZxux  
Cv+XT7JYrHJPdnIPKMdWnNjecoseZqEHcAKLPBjAMwsYxnZlXl8+oQTYx/fdV+e73GPK8P9EbyuD  
eiGPxSq3BTvmcTxOFqt8lgm4EwJmw3mu7QKX+e2rJuhUwj6NbQ5xj5Mq1Oe++3SMRG+qffLx2We6  
naIMtGWO03avLsaxbkKpUN4TsgK9AHi6GL2BOKvFAYYve9k7ETd2HABxUGchh50HBxDuFUknVrkn  
we4BxVkcndr7wMOEnt2iR4DLA6gPxCTy0vVGpBOR6KHHs0foaQbYmVOJVfu+pNKIVdfejqCxp5D  
jPPbDhjUn8/SOZONAx8OEFgfKO5yaHK9sqkn+8i+DMSCDz7F7RX8/+z73ZLFKve04peBHyFecZBH  
rygL0d5GACIMeTjwAAzDylqs/v57lo//F06WOC8bsRrm32SxCvGQfNsJxCq2Jw7EGLcfcGBRD2GA  
XhVehm9TASxWcVKK9cA2xMHIvucSbcg+GDPZtAkWq7wOIFmsco844gaXafeagtKI1UxPVoOwdUgW  
q4AFEhuLVe415Z53gLaDS95o8+hpY7HKl6iBLVaxXfFmBlwjhjG43I7bbHC1x5Vksco9jfYT22g3  
2CexPzHlw3ywwshKrgO8VZWoxyscF+J5Pjrd+fAUAV9m4p9cWq8knqng7DIbt/QH7ANox37cOcJUJ  
+w78k3yil0375v/J956CZLGK2zkwpfzAWI18nIRq4iJ6UiuP/YRxAvs/1hvPnbhuM/HCYBYiP0W  
J7AM4qwdA6ARIGZ5HwZ80pztWyKEEiRtaSElFAkGBw1cmsclAw789gMFvLPgKUI0fgRcjCMY40DE  
Z5L260j4MhQMgapXLz0M4YD/w0ICl1D4HhbbBITBwhQPBJFYFnnpXkOVTqxih+OnX3EpCqIKPT68  
jq5i1b7vFusZdm8RT8clmEyEiVXAARGXyyGi+FYMPoiiHnwfGAI+DpY48leVxbCQgXjJBhbNMNyT  
hv/BB3PuJeXp6cQqtgMEJ/JwiQo97dwOhw7V24h7JhAscZsG7vflbZR8X3FZi1Xcx8YiEqIDPSgH  
HZSYL5VY5cuZEDZhJltVAPHC2xqG/Q8HJbt3Hb6AT3BpEnncS8XLhIIV3MIC+NIwbuXAPed80MeB  
F//HflAyG7IRq4MH63FsP7RfvDGDt3z4/7iIVb5HGZdd0V6SybQOYWIVvZ/cCwjjeGgLCIwerZvg  
3kHcD4o68j6QSqwCFmC4FQb7Bl+axTpACNknP9n0cCeLVZx88P4O/+P/8+V4xHomTKxyO+V7m8PE  
My8DPyUTJIYB7/cwFqul2xyfcbUObZBvV8J+i/iO27Mwnk6sop4YRIm4hYZvq8A+YYs8xGReBuC2  
AYzjvtNswLwwWwAni1Xe1ohpaNu4PYX3i1zEKtqTfZ8pSHV8SAeOb5gfJ2TwB9/GYL8NAAIUeWhP  
2Ca8L9hXXfkWPPshLMEJ5T0hKxAE0aOBAy0uB+KAgd5RvneIQUDGA1RomAh0uL8V94khGCIw4b4i  
iAk0fpyBAVxOwKUDIIL0xAkONDiXi/suAgO9qVCYN8fGAZ6BxBwtu/54/tu04IVFkJ8kz//D9QJ  
QAzhUivWB+uJV+3wwTOdWOVLT+gFYHCQwD16EI04OIe9F5MDF+43zUSqYISDGfyA/4EAjXsX7V5H

gHs00VsJn+HeKPR2Izji/r0wcLDA/8H9mNmCMs84Q/sL9YAvwp5it8UqHwxcsDgQllgj7aC7QUx  
yE9zM+i1wqUonBxhvff/eRsC/l/cY8JilUUS4HIKI1bt21WwnVFfrDN6oOy2m0qs4hVzmJ7qZIB7  
+HGAs8F+w2VjmlHog59wqRr7L0467QMOL2OLVRajuBTKQFhif0a7RCywtX/3XsOyAW0N89rrwLcx  
QPww6NmBwMI6Yz9K7vXHpUkMo10xqcQq7qHD7S7lh/hNJtM6HHywnsZCj0HPet8GgLEoJ2gpwzr  
g7alHil+uBTw/ZZ2ryYO8sl1wAOGqDdOUAFBCccjKtY5XWA+GlgqHAijliE9oETfbsnF4SJVVzx  
wskL2gOOB/zQjQ0vE/aAFelSpvEVDwbloEMD0/h+dYD/i2MFTIDRfnC/NHo+UTeAB5ewjH0PbbjY  
BbjsDbGPk0Gc4OEEA23GJvkBK+7xRHZMBv6ftljkWB+dRpA3EZ9URccD5JvA8D6Y/zyy/U4SCVW  
0U74hJUf0GO++SZ8mXTA39gnEWtRP/Ru2z2tAH5DJxNiG9onOkPs2AKwTuilxv4glAqHrSZ4S3KQ  
YRBwEdT4/lkWq3gwAiQvl2kcogOXJdHDhF4sgIM19xbBPtPSFXfZLKdL4xcls0WiJmo7kHKpr4b  
Yp0A/k9p/lfYmJghwxPhEloQuYB7bGC4/zITyeVG7Ydsysv2f+JVcPbDk6nltjw8NacxZPc8ovcf  
vrNPLJgofZncVpRlg2zKy/Z/4q0OEPSZ5o96HalkXd1KU+/Srisvhw4CtDMbXAXCQ6+llaw+OlnF  
Q598AoMTBH4jBz8MmgmX9cTJgX0L3oYAPd7o+cZzJEKpSWqJwiYFnuLEGTvODLFD474kvqRi97hm  
C8rgr0yhDDb0sOLpYfS6Fgq4LQTCREgPerrw0AZ61uw2g3sV8WAL9whtCuBSHw6+2ftuZQt6DdFT  
w72wMPTwoLepNA8XbYrgJBpXDvhkSMgdfuYBva8Mel/DXiGYC7gCyLcXsaFnG2/Rse91jQJcuYPQ  
dhG3uYJYgFs6cBzm3nihVKiWIWzSYAfBAY9BYMe9U/gtLVgWI7RQDi4x5IKWUBjgVhe0GVwSRJsp  
zYMxhQoOeNiHcfDG5WEMb0oiX/AP7K+I7xuiAwL/Aw/vljYgRuCYtSEFZVmC9cC6hb22THBCxOqG  
JmwXnXGv63C/F78Xc0CA44AEmPNnK7xnd2IBfCR8pHpgRISQlgiAlmwwiVvMBX+7gTxXiqUVcFrUf  
otmQ4AESvvySzYNMPoLeJ/ar3ZMsCllgCMJGjTqyCxscFIUffqjH0bOJpzPt7w+XBakurXB9+JOe  
mS7BlGZ6pmXCcF2G10PEqiAlgiBsMqgu5AVuEcMr1Thbxuz4RU4eOcpG/dp8qthYHiNBr+zFN8+  
5/d58nsv8aoR9Kzy/Awe1sDrYfiF5jC8sgWvKQH2e1b5Cyww3Mh9zTWJ19XYr/JJBq+w4lcAcX3w  
5DbAQ1n2d+P5dS787jrOxwMI/O48PPGIXzzYhfUQ+NOO/OIG/iwdxuFP3PCe/C12+M7+4g3n4xVS  
eE0UXo+DXIT4hz+wAONPjsJErAqCIAjCJoM6sgtZgVfw4Cl4PLmIl9ejN5Rfcl33dgl8HY5xCFql  
OjwMwS+gxvtlGRZV/P305E9S4sXMeMleApLfLYd3U+Kdk5gHL67Gu954GbyLE6BnlPPwfleAT55C  
wEL0poKX+fhjPW6/+B9fclE4xNO2yMOL3XFPKC8Dw5PeeKcnv8garybidcD/5ner4wvfeKoZ7wAF  
+Dwd3tfr9LilyvjfZwMvxYLhpdWQ9xCqEIYoyw8lQ/4Ky4wEauCIAiCsMmgjuxC1uCFvhCKeDEw  
vrzCX9HBi7oB97riay0MXroO4Wa/VoVFFUQdSBareEk4hiH0bFiMoncRL6O2l2E4r29fk5EFvAx/  
n5nH8RJ7BpfkOR/rw8Po1eUnRrE8Xo6N90LyJyHxYmaITAhM+AsvsWZf4OnPZ5/Vvbl4oTL3lqMM  
hsUqPg+JHlvAL9JGLzUDPyMPJmJVEARBEDYZ1JFdyArOpriYxbvm8Lom/qwai1UilozjU6sMvuyC  
Fx7bX41iUYV3loJksYoHrzAMsWrft4mvYyG/tGI11T2gvAyLVRbh9ucYcVsBz2eLVV4HgNeO4LYH

fEkHtwzgXXl4bQfmw8vz0VPMtwQAvMMP0/D5xBkztHDFeJhYtT9TFyZW0fONPJiIVUEQBEHYZFBH  
diEr7M9LoocT39DmF53j+8EAn1LFOHoS8dlCCFX+VJz9KVG+PxSfScTtBPgEIsZhAF+84PtF0eul  
XkN8YhEiDnkQa5k+d8li9Zdfwuez4eksVnv00OMQg/hkKl6txd9zx6cKcb8pL8O9wwzEKE9DPdHr  
ik8sch4+e8lwHr5NDbHK9/qGidXevU2GAg+i4RYD3G+LHlf4h2/JgIIYFQRBEIRNBnVkF7IC92ni  
Szz4egwEEUQZHmTCQz54+AdfhwL4zjELO9j//V/x740D9CCi1xTT8Q30CRMS8zPoKUT5EL48DT25  
+CYsB+wsuE8Fp74DGHYfDY8nYUnBCYeJttnn8Q03KuLHmN8ZxpWPh4os7HrhUv/uN+v70VN/s47  
BDV/DQv39uJdr/gUHj5Xid5bwGVhmg1ENH/jGz3B6l2Gf5AnHykQBEEQhE0GdaQXvCKXVzxIWjZs  
ei7/rzS4LAuhKwiCIAhCQSNiVRAEQRAEQfAWEauCIAiCIAiCt4hY3dTBO0xh/NqnDQUewvrzT/2e  
VzwQhbcXCllgCllgOCJidVOHH1DCq7eiAA+Zobx04A0HJ56YeOsBW7duZgZBEARBEITsyKA6hDII  
6oebwqaxUAWtQ9n8f8yD103hM6/NmiXKS8ftt+t5zj9fj+Mds/zpVf6yliAlgiAIQhZkUB1CnEmT  
EkINL/HHpz7xpSm8LB9fccKL8DFtp530i/Lx+id8qQmfDkU+g1dZYfySS/Q4Xt+EcXz1CT2SeDXU  
FVfo8pEPg0jEq6wYnoaX6Z99th7Gsnhl06236h5NvKv0008TZbBYRV0xjneUZgPWA6/XwqdX+aMH  
sHRgXsxjC1O8jgp5eM2XIAiCIAhClmRQHUIc3H/JQo0vb+NTohBm9erpjwDg5fm9eulp+E4+wJeb  
MI6vWOHF9xCzGMe7WEHz5nr8hRf0OL/b9IkniCZOJOrUSY/jAwRMrVo6D4ITvxCvU6YQ7b+/HkeZ  
eE8pPm2KcVhpxard+8pfo4Klg+exv4CFDycgD6JVEARBEAQhSzKoDiGOLVbx2VPw2WeJvDAD77+v  
heHpp+sHnZB/111afOLy+BZb6E+UoucWoNcWwhYvub/IFqK99ipeHmCxesopRIsW6bw+fRI9qvXS  
flhnXjab2wD4e/5shx9uJhjwOVWelg6eJ0ysNm1qMgRBEARBEDKTQXUlcWyxyz+uh4/7jiTEcl/  
/+jL5/hiFe7lbNQolfrwqVB8TtRe/qqr9DR8iemHH4jeeqvK/2Wxes89JkOBL0khD1/YYlauTCyb  
zQNWAwbor0xBSNeoQXTuuWaClvux2rixnufjj/U4RDG+9IU89BQLgiAlgiBkSQbVlcQJE6vo1cQ4  
ekdxfyh45hmi2rX1/aTMBRfo+dDDevPNOg89pvzp1o8+0nmA/8fLL+tx9J5yHhMmVocM0SIT/wO3  
lgA84MTLslhFubhdIJ3ATkUqsXrMMbpMfKIVwAeY55BD9PjPPyfQPHq0zhMEQRAEQciJNUhpASX  
6cOE2v/+R7Tzzolp6Cm98ELdq8n88ktiOnovAe5pxXjyvaN33KHLwLRq1YjOPJNoxx2JGjYkmjIT  
z8Mit0cPPc7ggS9+uAnWooX+pj9E7Pz5eh5+Uh+3C7iSSqxiHZCH11oBvGMVD3ptuWVifvQo26Jc  
EARBEAQhC5JUhyAlgiAlgiD4g4hVQRAEQRAEWtErAqCIAiCIAjelmJVEARBEARB8BYRq4lgCllg  
CIK3iFgVBEEQBEEQvEXEQiAlgiAlguAtIIYFQRAEQRAEbxGxKgICIAiCIHiLiFVBEARBEATBW0Ss  
ColgCllgCN4iYjUDK1RaZXK1KgmCIAiCYLFoUXFbtsxMEFKxXCXWFgtVEn2RGhGrGWinUsxK41US  
BEEQHiklQej/rItHtMceREcfrdSEkhPfrCoVHXUk0X776XmFEpynkq0vZqkkhCNiNQOdVLb0wSV  
BEEQBKfGmTSJqFw5LUovv5yoWTNaobLtY+X+s7clOrsV0aWX6vmOPVaLXBa6ArVUyfbZbJWEcEsS  
ZkDEqjvr1Nk2m5CZlHw82V/r18tloGyQNuaG3cbEZ9lh+ww+EwzDhxM9/DDRNtuYDA1umbOPIU1U

ivP770Q9exLVqWMyBCBiNXtErCZhB6Xjjzyedvxyx3hDQjq41cFmavF5Cxn2w/vvv0/nnHOOOoIW  
njJ29tln02OPPRZMF0py8skn0z777BP31xZbbEEnnHCCmSptjGE/jB07Is4444xibQw+vPLKK4Pp  
QklatmxJTZo0KeazU089lebPnx9MlzamYT+sWLEiaFMVK1aM++vwww8PYlnBwm3k/POJPviA6Jpr  
9LjhlItuodPPP906Usao5siaNOy7YcH0YOmpU4IOO43oyy+JnnsuyC807H3t+MOPp/o/1bc8FqMj  
WxxppgrJiFhNYtWqVbTXXnsFAapcrBzFnrGbksrbQ+WZ4CU9FJohQ4bEg3o6GzBggFICOF8Fffik  
XDndnmzjvNq1a9OSJUvMEoXN5MmTS/gnzF544YVg/kiWYLzut99+e6iPYLYP58yZE8xf6MybNy/U  
P8l24403BvMXXBs791x9Of+SS7T4VPTp0yfhm2rKrFTux4QPf0fPKrPTTrqcYVrIFhLLly+n7bff  
PvBJoC/eTfgLqVx97TOcLBVYDaTDxGoSvHPFLUmsxnZXZk0vdNq1a1fMH5nsxBNPNEsWLMF+SWf/  
/vuvWblwefHFF0P9ksrq169vlixclSpEuqbVPb999+bJQuTkSNHhvoliUHMfHtffacFphWldt11  
1+J+UWL1rJ/UPBhg+SxKg+vVq5dZUvH447q8AgJC1fZHYELiINVZPmTVdSCDesEDjKHFGnUas8ryF  
Sv/+RN+crDevXubEgqP6tWrp+21SbZCb2Po9Uv2STZ2ySWXmBIKj0MPPTTUJ6mM21gh3y9t+yFb  
2wNPwRcKuE9VrTMtXhyMXnvtXE/NFX2LKYZKyqnfxcpe10NV1e/PO8kPJgFhnlEzw8KpAcR61+i  
jaURq4Ue+5MRTxjsS0DFLEPPavny5Wn69Ok0ZsyYTdtGKktKLwx5gWI7Kz/s62DKf3d8dAeNVcku  
a8xaZfgfYf97Y7W/IFlpmkq1jq0V7pd01ihGU1SyywrSVGVh/3djNWz/dcqs9N387yi2Q4hP0tk+  
MTrrvVOChyHtsok0qbWxCcqsNFmlQ9oeEu6XdLZ9JEvH22VZNLfysL+78Zqo5QtV2a131VybmPK  
9rxgz/D9crSysP+9EZs62NGMp56iMePGBZf0L7vssvgx8GplmJ7KaijeT///POgvFGzZv0/e+cB  
JzX1ffGhdwRREEXACmLv7S+KDRULFkRAEARE7NhFVOWCNIRQQcWG4E/sWFERC3ZBBBG12EB67+3+  
30nzfZPNTHLZzQo75/s+18VJm7I57+Xk5CWRfQceKBvNZz/+9VeB7ZW0mDFjRI4O0iLCWYW5scQ7  
Qch1TG0i4JBDTAdvVZK8iBCrORVJlRkmwra5JcepJplsX5sl2+6WHItNJFnCtrklhxGZiZabTIRt  
d0uOr00kWck2uYWH+Y+0C3yGqGMC07LFchPB5RBfmVgT+CznIkKslTq2beuplNzG1CYCmjRpUqCS  
OEGxmh9Jlnkmwra5JcfpJplsJ5oI2+6WHBSrumholSnS00TYdrfkoFhVh/mPnB/4DFHFxMcmMD1T  
9DERXA5BsWoihlg944wzPJWS25jaRPY77oKVxIkYynXHHXd0buoo0VHVRKBss3obSf1tcvCPIsz8  
NZbXsNbILQUmqpgl2/aWGp1MWGVHUOrPKh2el2zx16ljptjrckobE2Hb3VID+3+pCavU3VDX+f2h  
eckSVRZVxqm2OtySkmrY4eZsAp+c4V5FUJzkjVMju315JU+JrYzEbbtLTG2MvGjCasgZ5tSx8rP  
KR/eLmuZCNv2FhyyzTZyuTnWbb/99IKvXj2pWrVq3vHvKhPmHxmjpgl/3jp16jir26pGDfnB/P9a  
s67KgW2VxIBG8HOQFhFitVq1ao42IaYqEX+JAZWjwABO3mCVkVHvjsrPkylGPjLQW0PuUaFMBfeR  
JSF5CQt/3lxI7qwMY8kj4vzWuXuD1cH7Hxyak0yRV8dy+H0Udh7ixu477+4tnQP4TwNYifdU4TGr  
I+flAWK0LqZ5saG0+3eyiQalUILG/PXnnTZtmrO885xV8/+5BH4/b7DadJgJD99dPeyww/IrEiKL  
WG3WrJmzTC4/D+3NN990clGgEQbCn/7UU095S+Yefj0JvjghW2B4CsJlOoYXAZQvXz40P3b4daxX

r17ekrmHX0/wkgQ7J9lim222kdmzZzvL5SqLFy92XMOW/Njh57Nz587OcjnTLn//3X1j1X33eR+I  
3HPPPXk5KYX8VDYCFkFKXmpj/o0yzkzzpuNGZDyPO4+TTxbJoccY+vVkv/32S6tP2cTqKXiBgiGX  
+34bitUQ8Ngb2O9OpQmK1d1TziUQnFmS/IaEXGUSFP6bYAAbnsVDDz8sISpVKpArP1DHjrrqKG9u  
AvDs0IoVK4bmyxcR//zzjzc3+eKLLxyREMyVH7jTGJdIST477bSTVKISJTRfiHLLysk777zjZ1j  
dO3quqH77ut9ILJo0SInL049q2hyZJcxKamQquBMA3k9P9aByMGXAoAWLVrk64ugWK2bcupf7969  
vbmJD8VqFr74+AtpOatlfkUy5dmvn/WmkiB///13gYdrQ2DkumuTjTFjxkjPnj3TcnbzzTd7U4mP  
f5IDB2zcuHFSpkyZtJxNmTLFmU4KMnbsWBkyZEhavs4+++y8nPIE0sXPA543+9133+W9acgP5DHn  
+f571xG98ko85d77UJzHM3385cfWkTlley/aW1bOdYcNOPTv7w4nOOggPCvS+zB3+fKjL+W4ecdZ  
GUvJiLEjvKkkCMVqBN1NsSsTnt1IsmN38HAPSXaCQmLYsGHeFJIJ+73tCJkd8ePHp+Wrr48e3hSS  
iUaNGqXlJbgg6OfPF4Hzh5xAfHqsNMVkkKq80lWO8KYY77xQ5/XSR5s3d/+cJkkNrU+yczTSFhMMW  
GAHFqh67g6dYjYZiVQ/Fqg6KVT0UqxEOaCBSqpQrWk2sIPWS2piSUI40/dT9PC8gWEkaFKvxMTWI  
ZINiVY/dwVOsRkOxqodiVQfFqh6K1SzYzj6q1GjPFFa2vtrYpsmsvG9d0VeftmbkQShWI0PW2AE  
3X8zYtWclTrxkRGrKyhWo7A7eIrVaChW9VCs6qBY1UOxqsCIV4xOTf1ocvWzG/tNb+pOlxlP/YsR  
q5a+mLmGYjUTbIERdO9uxKrVYU2eTLEahZ0vitVoKFb1UKzqoFjVQ7GqY+XKIWn5anoUxWoUrc81  
YtXK2cyZFKuZYAuMgGJVj50vitVoKFb1UKzqoFjVQ7Gqo4BYbUqxGkXr1hSrcWELjIBiVY+dL4rV  
aChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb1  
2PmiWi2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRsr8WEL  
jIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDS  
RoftsAIKFb12PmiWi2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8Wq  
DopVPRsr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0Uqzoo  
VvVQrOqgWNVDSRoftsAIKFb12PmiWi2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjV  
Q7Gqg2JVD8WqDopVPRsr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54ti  
NRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWi2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopV  
PXa+KFajoVjVQ7Gqg2JVD8WqDopVPRsr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzY  
AiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWi2GYIUPxaoOilU9FKs6KFb1  
UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRsr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOx  
qoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWi2GYIUPxaoO  
ilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRsr8WELjIBiVY+dL4rVaChW  
9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12Pmi

WI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRSr8WELjBi  
VY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSrof  
tsAIKFb12PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopV  
PRSr8WELjBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQ  
rOqgWNVDSRoftsAIKFb12PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gq  
g2JVD8WqDopVPRSr8WELjBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqK  
VT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wgu73GrG6  
v6lIB5rY3YjVGRSRUdiNj2I1miFvGbG6t1fH9jI9QOK1SgoVnVQrOqhWNWxcrURq7uaXKEfM9G0  
C8VqFK1vNGL1AC9nuxmxOpdiNRNsgRF0N8VUo7wy2RSSHbuDp1iNZogpfv1CGWYKYQ7Fqg6KVT0U  
qzpWmmlylVeamkKy09oUO2czTSHhsAVGQLGqx+7gKVajoVjVQ7Gqg2JVD8WqDopVPRSr8WELjOAS  
U+zK9JspJdt2B0+xGs1zpvj1C+VIU0h2KFZ1/Pzzz2n5uuaaa7wpJBMUqzrWmGlylVeONoVk5zxT  
7JzNNNoWEwxYYYOPGjd6/RGpVrSXVX6yeV5FQqh9S3ZtKfPyc9e/fX3beeee0Dr5UqVJy7bXXOtNJ  
Qbavu73Uvr62VcNSUr5DeW9qen3MZfw8fPbZZ7Ljjjs69cquZ82bN3emk4Lss88+Ts7sfJUuXVr+  
+usvZzrrmlufhwULFkjt2rWIXLlyaNbffffdnemklGeddZbsvKfp+61S+vPS8u7L7zrTWcdc7DzU  
qlxLqo6samUsJbX2ruVNJUEoVgOgo6pXr57TOZVKmQPILsqmc/2cA+S6LiWL1/uLZXbDBo0KK1T  
zxR9+/Z15s/ljsv/7YcffiTE6eOdTT5sUqpNm4dq1y5ssyYMcOZP9cZOXJkXj0KCIU7rrzySmd+  
1jGRNm3ahOYIYedw7Nixzvy5zo8//hian2CcccYzzyvYl33HBDfm4qmbBKqc/yczhixAhn/lwH  
d/twvfXWTk6cvn94fr5QSu3g5my//faTdevWeUsRQLFqMWfOnLzGIRcBsZpqbmKavnTpUm/p3AQd  
t52PqEAjzGU2bNjg5CHtYBgQq6nzTPjTTEycONFbOje566670vKRLZDXChUqeEvmLn4ugvnJFMOH  
D/eWzE3ee++90LyEhZ/XXKdWrVrpdSwgVINjTPjTTOT6FTYYD3Y+nAi1dT2Jsznfl7XrFnjLU3Y  
4izSKpEfEWIVkat8+OGHBXIRJ5544glvDbIHjRo1CuYkQqwicpV///23QC7iRKdOnbw15B5HHXVU  
aE6iAidSuUpYPqICwytyFdygVyAnEWIV8fvvv3tryD2CuXAig1j1o3z5/CFhuQ7FqseXX36ZVknY  
lkKsYuzXm2++KUOHDI3ZMcREoPSc2FNSZ5k8tFVEq5R0+aSLvGSKva6hq01gG2HbLiGBelK9evW0  
+uNEDLH62muvha6zRAX2/1oTVhm4eKCKTjc5CKtLmaJNSpoNbib/M8Vel1NKWh17w4RVXjVlz7v2  
DM9LtljA5fnHji9aavDlexismwra9JcazJuabsMpwU9R1zMTON+8sr5lir8spz5s123YJCbjpg556  
aoE+Ko5YffDBB0PXWdLjJTfeKJALjLEasWKFeWnn37yVEpuQ7Hqcdhhh6Vvkryl4azmTCRZ5pkl  
22YuRAYxmjOx2ESSJWybW3I0NJfk6WkibLtbcnxtlSkSts1ciBhilRGICLGKaNeunadSchuKVY/G  
jRsXqCROUKzmR5KFYjW/UKwmV8K2uSUHxao+KFATCypVfcQQq6eddpqnUnlbiLWdf2djsJl4ESFW  
MRD65JNPluOPP75kx6EmAuWABQdl6n2Th1GK+CAle/yzh7UWryw2gW2EbbuExEknneSMQbLrjxMx



xCoezRS2zhIVh5hYbslqTdc0ldR7JgdhdSIL7PDLDnKiKfa6nIJthG17S412JqzS3JSaP9QMzUnW  
eDclx208zlgTV54x0cxE2La3xDjCxGQTVjnBFPz+0LxkierVHFyba/LKUeZCnT2CYkTTjhBGjRo  
UKCPiINWjzjiiNB1lvRA3x/MhRMRYhXDxkAuP3nCh2LVwq4kecEbrDly6sNRBXIRJ5584klvDbIH  
zZpGSARzwhusMslbrPTwBis9Yfmlin332ddbOve4+uqrC+aEN1hIJZgLIjLEKp9skg/Fqod/5nL+  
+eenVZZsYrV79+7OMrl81jNhwgQnF1GPYfGnf/HFF96SuYdfT2699db0/GQRqyeeeKKzTC7XMQjW  
0BvTAuHXsVx+pqNfTwYPHpyWk2yx6667yrJly5zlcpxVq1c7d/eH5ccOP5+PPvqos1wut8u33347  
LSeZxCqm44rSn3/+6S2Ze/j1pGXLLm6u/MgiVq+77jpnmVyuYzYUqyFcf/31zvNAnUrzmFOF8ktD  
91mh99xjzd3buM3pJ122sl57l5el7SiatWq0rBhQ2c+NjwX3CGad3Bsb8luZ6Zk7732lrZt23pz  
5zZ+nWnSpInUr1+/QP1C4KkcEF1LlixhHfPA83kzsU3sdtuuzk3lgLmzKVZs2YFXrNqB+rgV199  
5c2d26DO4MH1eGuh87avsiZHdvkkJTtU3yHvzV+sYy7dunXL7/tfysuWW2qlZN9995XHH3/cm5v4  
UKxmYeG/C6Xjyo75FcmUL+blrjMYBd7otWrVqrTOHY/elJmZO2euPLL0EauGpeTxReyoMoGHZK9d  
u1bKli2bVs/Wr1/vzUGCzJs3T0aPHp2Wr4suusibSoJgOAT6MZz82DnD2w1JOAsXlpS/5v5l9Wlp  
OWyNORFa5c1ACrD438XScnVLK2MpGT9vvDeVBKFYjaC7KXZlmmwKyY7dwVeqVMn7lGRiiCl+/UIZ  
ZgrJTlCskuyMHZ8+LV94qDvJTtBhJdlZaYrJVf5pagrJTmtT7JzNNIWewxYYAcWqHruDp1iNhmJV  
D8WqDopVPRSrOihW9VCsxoctMAKKVT12B0+xGg3Fqh6KVR0Uq3ooVnVQrOqhWlOPW2AE3Ucbsdrf  
VKNHTPQzYnUhxWoUdgdPsRrNkF+MWH3lq2MPGrE6hWI1CopVHRSreihWdaxcZ8RqX5Mr9GMmmo6g  
WI2i9XtGrHr5Qu5mLqdYzQRbYAR4PJXdYU2eTLEahZ0vitVohgwxYtXK2bBhFKtRUKzqoFjVQ7Gq  
Y+VKI1atfDVtSrEaRevWRqxaOZs5k2l1E2yBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs6qFYjQ9b  
YAQUq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVBsaqHYlUHxae  
itX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2l1PmyBEVCs6rHzRbEaDcWqHopVHRSreihW  
dVCs6qFYjQ9bYAQUq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVB  
saqHYlUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2l1PmyBEVCs6rHzRbEaDcWq  
HopVHRSreihWdVCs6qFYjQ9bYAQUq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18U  
q9FQrOqhWNVBsaqHYlUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2l1PmyBEVCs  
6rHzRbEaDcWqHopVHRSreihWdVCs6qFYjQ9bYAQUq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPD  
FhgBxaeO18Uq9FQrOqhWNVBsaqHYlUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGq  
h2l1PmyBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs6qFYjQ9bYAQUq3rsfFGsRkOxqodiVQfFqh6K  
VR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVBsaqHYlUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1  
UKzqoVjVQbGqh2l1PmyBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs6qFYjQ9bYAQUq3rsfFGsRkOx

qodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVBsaqHYIUHxaeitX4sAVGQLGqx84X  
xWo0FKt6KFZ1UKzqoVjVQbGqh2I1PmyBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs6qFYjQ9bYAUQ  
q3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVBsaqHYIUHxaeitX4  
sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2I1PmyBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs  
6qFYjQ9bYAUQq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQrOqhWNVBsaqH  
YIUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2I1PmyBEVCs6rHzRbEaDcWqHopV  
HRSreihWdVCs6qFYjQ9bYAUQq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgBxaeO18Uq9FQ  
rOqhWNVBsaqHYIUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2I1PmyBEVCs6rHz  
RbEaDcWqHopVHRSreihWdVCs6qFYjQ9bYAUQq3rsfFGsRkOxqodiVQfFqh6KVR0Uq3ooVuPDFhgB  
xaeO18Uq9FQrOqhWNVBsaqHYIUHxaeitX4sAVGQLGqx84XxWo0FKt6KFZ1UKzqoVjVQbGqh2I1  
PmyBEXR/3ojVC0xF6mjiXCNWZ1GsRpE6z8uXiUpdKVajoFjVQ7GqY/xfRqy2dttkqoMRq8MoVqNo  
dKMRq14/ljqHdSyKIWuNWDV58nPW9B6K1SgoVuPDFhBd1NMNcork00h2bHzVckUkh2KVT0UqzrG  
m4L26JceppDsNDLFzhnJzktP7Hw1NYVkh2I1PmyBEVCs6rHzRbEaDcWqHopVHRSreihWdVCs6qFY  
jQ9bYASXm+I3vtTGLMwwhWQnL1+mUKxmZuPGjc7foUOHpnVYr732mvM5yUyYWPXzSfLxc/KrKei/  
/HK9KSQ7jU3Jy5jJHWAdK4idE7uOHWMKyU7btm3T+rEFCxawjmWAYjULqDylnyid1/icUo8py0Sv  
Xr2kdCqQr5Upadu8rTcHCYI6VqZMmbQOC7F69WpvDmLzxhtvFMgVokmTJt4cJjNmjWl7IFG3Nvl  
jpRM+GGCN0d2vvn6a1m/fr33fyWfGTNmOHWq1JRSdsakYrmK3hwkyGGHHSblqpezsmXKqJQ8P+B5  
bw4SBHWsdGlzLT6MQQJh5kJMH36dKlevbpTaUqITGc10Gl2eaXUHuYzM22bbbaRuXPnekvIjv4Z  
4CWXXJLW2NKKkEav+5ziLBLI85ojfvmHDBmnQoEF6zqwoVcqtY3APv/nmG2/J3KZv374F8hQWxx57  
rDN/rtcxOCBB+bnZl8TVin1oFvHEFdf5Nc9O77cvJ3P8ix48Zji2+/ly6jP5Wun46Rlm+8LTVH  
fSLtPhglF5m62NUI1wu/GCttPv1Mzv7uezlz0hRp8f2PctL3P8hZ3/4gF458V+4dOkw+efsd5zts  
Sbz++ut5OXH6/l/9bHnFm7b33ns787OOiZx88sl5eUIVcrKUXz4z4U3r3bu3M3+u8/PPPztPyEFO  
/H7eDv+zunXryvLly3O6jgWhWLX48ssv0yqMEwGxmmpswprnxx9/9JbOPdasWSNHHnlkfq78slsl  
VhENGzZ0HnGSq/z+++9OHsl6qkzx5ptvekvHnD0go+PyxZ+XhctWuStlfeYM2dOWi6cCljV1IMm  
zOdVt6ou27/3gTRYtVp2/HumbP/ZV7Ldp1/ljs8PI/2ff0GajPpYgVw9Xw665VZpfs210uK66+TU  
Sy6V4887T05s30GOvuseabBwoTQwJ2A7mW0jGpjY2sTVve82/90yGDhwYH6u/MggVv285vL4wqVL  
l0qFChXS61hQrI4x4U8z0a5dO1m3bp23htxj1KhRTh7i9P3+PJMmTfKWJhSrFsEK40QGseoHbPxc  
5aOPPkrLRV7YJSBWEYMGDFLWkHtsu+22BfIRJ3KV2bNnh+YjKrp06eKtIfdo1qxZwZwExeoDpu9K  
lZc61/WSHSb/KvV+nSp1/veGbPfeR9LwzXekozlBuLhbN+nWtatcfOGFsuMno2Vns+6d1q6Vnc1J

al6Y/280ZYqc1a69nGHE6xE9e0n9+fMd0bpfn37O99kSKJAvRAax6gec61zlOnPSEsxHlFhFTJ06  
1VtD7hHMRZyoUqWKtzShWPUYPHhwaGWJEqs4A7rlllvk2muvzbk499xz03KRF3YJEavNmzd3Oruw  
dZaoGGLCKr1MqTygsqT6mTxo4q6U3GyKvS6nDDIRtt0tNS4xscqEVS5Zfomk7gzJSbbom5J9PthH  
bjTFXpdTsl2wbZeQ6Nmzp9SvX79AmysgVu9NSfXWXWxbof+THX75VXac9a8jRhuaOOKxx6SrEfsX  
XXSRdDNxCcTqh6NcsWoCf+3YY8pv0r5tO7n44ovlljPv4U886ayn3uNPys09eoR+z80pr/++oL5  
QkSI1e22205uvtm0y5B1luS44YYb5OCDDy6QjzhiFSeRYessUdHDxFgTVkHfn7rH5CCsv8oS5R8u  
L7eYYq/rElNWm5JrUKx6YIB4sGE5ESFWGSFhlxCxmjNxuokky4kmwra7JcdiE0mWsG3mQgTFqjkB  
qtmzd96lez/qmzj2vvvyxCqiuydWg/P60XjKFDm/bVtX3HbtKifcfqc0XL1athn4ePh32VliQqwy  
AhFDroZMBHVDZclpuQaFKsee+65Z7xKR7EaHXahWE2uUKzqS9g2cyHCnNXzL5RafR+R7X8Ynyc8  
Md708P79N1msdu3USY58bKCznloPPxL+XbaUoFjVBcVqfCsFjkUqwb/jrtYIS5kGAAe2XTNNdfk  
XLRq1SotF3lhlxCxeuKJJzqXS8LWWaLiGRNWwaX8ygMrS+p+kwc/cLmnjwkjHpy4z0RfE/Y8d6ek  
pyn2upzyplmw7W6p0d3EKhNW6b68u+MCpuUDEbhcfpyGYQB4lqi9LqdgG2HbLiF0003xRsGYPJU  
pmpd2fG2O6XOR6NI22eHyg4/TZR602blzu9/KJ2MQL24mxGqXbvKpW3byA6ffe4lUDivwdj9zz/l  
lDbt5LRrrpVmffrKTjNmOCJ25953yE1XXx36PTenCB1/iYgQq3Xq1HGGXYStsyQHhk1s6jCAzp07  
h66zRMVVJr40YRX0/U7/bvdTMaj8//LOEAJ7XXhREYcB5DhVq1Yt0LiixGqtWrW8pXOPMWPGpOUi  
L+wSIaffz53n723++675+UBJzpVK1aQ8/etLUNa1pTX21ST+07YVg6rv1XaQ+9z+SY+/872vJyV  
KSMHICKv3cvVklfLV5dB5beSK8pWk63LlpOU9czCq666yltD7nHGGWek5cyJoFj1ngZQY9ttpeEn  
n0rd8T/LNo8+IXVGvOk8DaDuLXfJsZdfKQfffa/sOPx/svflkdLs3nvlMuvkFZGyJ7e8kw5/exz  
5LT2HaT+hAIS7/sfpMHatY5lbbhhg9Qzcdm9/d0vtAVQIF+ICLHqPyYtF7ntttsK5COOWMWjIXOV  
8uXLF8hHVOy4447e0oRiNQCeB5dWYbKI1YcfftHbKndZvHixk4u0x3HYxROr/nSIj1znVdecXLR  
7f9qSZ+TdpIfL6klf19dTTb2riiTLqshw1ptLw+fWV+OalBBul58mbdU7oIXJNqznXZd1J71dxB  
xlXYVp4qt71lhfkysnwl+V+5uvJSxe3l2hp1nLz++uuv3pK5C57Pa7e7TGIVsevBh8j5zwyRBo8/  
JVs98YTs+shAOW/wc3L+s0Pl+IHPyTbDX5GO/R6TTv2fkE6PDpLz+/WXFr1ukSNvuUX2f2yo7PLk  
ENnx8UFywmDn5dxHB8oDI9+R76bNkJXLl3vfZvMHV9eOP/74vJw4ESJW/XziRD3XgfC0c5JJrGI6  
nl2ey48s9MFJtJOrGPHMM894SxFAsRrCgAEDpEOHDM6l6W/CK6VMSdVjYQUXXCAvvPCCN3du4w+h  
OOuss2S33XZzcubkyS9GrNYuU9uZDviQYyPwlyyV0Xe0kLnXVZWh52wnH3ao5QjV9bdVINnXVpVp  
V20lr7SuKyturiifvDTA5MxbMlf5bOxXsqRMJfmxXE151ljTleUry7oKFRzB+lG5beWf8tXktzLV  
5dV6O8saYcLAn3/+6Tzb0unHdgu0yztTctKxJOU+4muNEZyPvjhUVi1f5n1Ssrn00kvltNNOc3P2  
S1623NyZz9q3by+//PKLN3du4/flGA6W96B7u46NSsIBjQ5ypgP2/S59+vTJb5dW+KK/Y8eOzpv6

SDoUqxFcaorT8LzymykkO3a+KplC0pn913T54PwK8nTLHWX+jVvJuO4188Sq9K4gY7tul6M7bSML  
b6wqb/Q6XZYU23lccqR4qkNnI04rykvl68nY8rVkrRGqEKuuYK0gw8tvL5Mq1JCFqdLyxnvveUsR  
n59NsdsIxr6R7DQyxc4Zyc5aU+x8HW0Kyc55552XJlJxXGkSDltgBBjM7Dc+IMmmkOzY+aJYLci/  
f/wm75+bko871pYPOmwnck9ZkVvKyLpepUVuLyNTLt9K5t20tXx2YU157bpmjhOb6zxxaktZX7Gi  
DCu3gywoV1WkdBlZW6q0bDAhZcrI2HLbyl8VasmSVBI58nle9Qgy3hS7XfYwpagpab4ZxaqOlaby  
+WpqCsIO69at08RqLr8VLQq2wAgoVvXY+aJYLcicf2fJylt3k7fabiPPXrintDunpZzS6gJp0aaz  
nH5OW7ni3OPky8u3l7fOqyUfP3q5rF6z1lsyd3mk5y2yulwVebrGjtLr8KZyZqvWcvKFF0qLCzpj  
q5Zny4N7HyijK9eRj1NIZfyvbKNBikOsgtSnnrxji3J84t9r1sixkydLauxYqTdunMxZu1Y+WLxY  
3lu0SGaZf/8XUKZqoFjVQ7EaH7bACChW9dj5olgtYlX58+TTGw6SW847VC7CKy1N+M+0ROD/u17U  
TV5vU0XGPneHbNjAsV7PDXIW1ISoKOe1PEe64U1Jvr6cMJ/1PfJoed50+Gu8ZUG+xSZWv/zSiSmr  
VnmfxOOBf/+v1FdfSZVv5XF69c7n6W+/toRry+a9vJfQLGqg2JVD8VqfNgCl+j+vBGrHU1F6mSi  
tRGrSyhWo0i18fJlotJFFKtBvhr7IXzQvZEMbN/liKzuBcTqxRd3k7PbdZJfu5eWobdcIEuX5vaY  
1XUmBtxyuywvV06uOL65XNS9e1q+EJ1NDgcecJB8aDr8sT/84C5I8hj/lxGr53nt8gljVocVn1hd  
asRn/9mzpfluMGXLvN3/KGwsXeINcRixYIPtMmOCIO3LffCPXmnmu/OMPV6waAXvYxlkyrBMcdDo  
JiNWvX4s1YqHyihWrjVi1eTJz1nTeylWo6BYjQ9bYATdzYHRrkyTJ1OsRmHnC3eJknRuvqCFvNC2  
rrzefhu5rMPZck2Xc+Xqzq3yo+t5clurg6X/2fXkrfa1ZPQHI70lc5MvP/xlBtTbRZ6tW08G7bKH  
XHmuyVcrkycrrmt5ptxipn1eqZo83ugA+WveXG9pAsaPN2LVapc9ehSPWB2/YoVU++47qWNOIA7+  
+WfZG6LUCNAjJ02SNd7d4Y8YIbv199874rS0EavHmT626S+/uGLVxHZm2Uf/gxtPGjUyYtXKGckO  
Hk1I56tpU4rVKChW48MWGAHFqh47XxSrBzk7d648ftkJ8kTnfWT9rRVI1S2VZGWvinmx+tZKMve6  
CtLptCPkh2F3yAY+8kWGPfWMvFV3R3Inl12dJwGsCMRKE081biKPIskvugvQucF/IVbXmnrbYdq0  
vMv7tY3orGVEaWkjQMsaUTps/nxvKZE+5iCN+aobYevjiNWxY2WoNV9xQrGqg2JVD8VqfNgCl6BY  
1WPni2K1lAsXLJCnrjhBrjv/BJE7KsiG3pWcx1b5lbdXIFEX15arWjeT8S/fxaeGGv737LPyxvb1  
ZfAhh+JVMHmPrfJjWeXKct+hh8vAsuWZrxD+C7G6csMG+b9Jk5z/7/nXX3KfORDfY6LfrFnS18S4  
FSu8pcSZlkms/mdjViIWWVCs6qFYjQ9bYAUQ3rsfFGsFmTe3LnycJf/k4vani1yW/kQsVpeRnTb  
Xbqf3VS+fvZWWc8brGTy00Nk2A47yIPHHSdSrlwBsbqkalXpecwx8liZSsJnJxTkvxCruMzfBupU  
SX3xhzbzouaOoyS/jZM20gT/X5N8KI02svkCxukVAsaqHYjU+bIERUKzqsfNFsVqQf2f+l7edtYcc  
939HGHEa4qzeUUEGta0nZx21t4zqf6ms4aOrZNB998tjdbexa/faN9RZXVilinTYZx8ZXKqC/MVX  
+hbgvxqz+u3y5VLVCFAlz53GjZO6P/zgTD9tyhTZ4MzhEiZWa2lczqeEoO+//3qfFh8UqzooVvVQ  
rMaHLTACilU9dr4oVgsyf86/MqTDrtK5XSvZ6Dir+ULVEau3l5dXlt5D7mx/lHwz4GI+Z9XwwrU3

ysd16sq9J5wY6qwurlZNeh/dTMaWqyl/TPzZW4r4FJdY7Tp9uhN4TqrPvHXrnBukrvnzT7n9n39k  
7LKCr24dtXixXDpjlhv5vHBOu4yB+/L//hDRi8t/hdjUKzqoFjVQ7EaH7bACChW9dj5olgtYoy/  
Z8gHF1SXL7vVkdW3VZFt1SXZb2q5cXKW6vJuMtqyllbKsjl28+RZcvzx/blKk937iybjSj9ttpW  
srJKVvmKqJwfc6pUk9/MZ/NSpeXdjz/xlil+XSVWSxIUqzooVvVQrMaHLTACilU9dr4oVgsy68+p  
8l7bcvLFhdWlzfFNpeOFnaVjx45edJKWbTvLBCNWpXcFebVnC1m6LLefswqebNdB1hux+mDtBtLu  
7NZyQadOedHRRlfTz5BlpcvJMIPnXhmZ24/6CoNiVQ/Fqg6KVT0Uq/FhC4yAYIWpN+K1YLMmz1b  
Rl68u8jdFeTx83eTzhdfnvdwe7wg4PRzpzclt1ST5TeVl9GPXZVhzGpu3XQ14KZesrF8efm7cjW5  
5qzx0l6kgDdaXXLMcSJly8o4U+fG//qrt1QMNUwXyFvgo8QoVvVQrOqgWNVDsRoftsAlKFb12Pmi  
WC3losVLZeTNJ8mG2yrl4p7VpOvpR0vHTHca4dVv2RtK4Pb7S7rbq0kf/eoIKP/96S3IOHLfiK9  
TJO9u7L3gYL5v4gM3FPkFrP8Vw+4n23c4MYWwMvvvStLypq6VKGCDK3bUM45p5V0M0K1U9cuct6J  
p8hPIWs4N149hL9RXHopKqll/freBzG4+mp3me239z7lwgaTU8RmJGopVvVQrOqgWNVDsRoftsAl  
KFb12PmiWC0IJMz7g+6WJdeXfmrsgsidFWRZzwoy/8ZKsv6W8t5n5WTGDXVlZJffuguBsfe7YvOe  
qt4HCu7bSuRms+xbnUXmGeH656ci15r/v27L6ALwytX+5c1vKFdOpKLJT9lyMq+8EfvLzL/9z0x9  
e7RzV3eBbEyaJPLOOyKjR3sfxOCaa1yxusMO3gdZMKLZmff2270P/nsoVvVQrOqgWNVDsRoftsAl  
KFb12PmiWA1n7vyF8vSagTLwmvPkkQ4HyOWHVpJrjqgklx1SQfq000ie7H21DH/ITW9ujOxidcU8  
kb+/EplhxNes70XWWHdbr10hMnuCuxxc2c/vEfnnG5GvHnTFKz6b9aPI6iXeApsv7330sdxYtT5  
rkVLuaJcJbmqbBXpUbaSdCtVTp5ocabcdOFFMnNW4BFHH30kMmqUylFIhMninrxP/774vcfbfl  
gAHeTB6//+4K2J9+EsHd51+ZnH79tTvNF6v16rn/j88/NYIf64SLCtYZSY0hCFWquPN26uSuczOA  
YlUPxaoOilU9FKvxYQuMgGJVj50vitVohgwZkpazYcOGeVMChInVn4e6n91qww97b3WRhZ5lgova  
dxuR3uZzBMTpo7u5f/3PMKzgh0+UbSGULVs2LWcZwTTEEue4Dux++4l07ep+1rChN5OhQ4f8eRGN  
G+f/G/hitUIFkZ13zp+GaNvWveS/aJHIIYekT7voInf5/xiKVT0UqzooVvVQrMaHLTACilU9dr4o  
VqPZZLG6aLrlbeb/760m8qoRTO9dKfLCCa4QHdjEncchY/U0Mf1j9/9nfityk/fZFoharNas6V6W  
f+01Ngr3M1+swmXF/++1l8mLOVjgGaEtW+YvC3yxivX4N3B9bHLpz+O7q6BGDFezB7yxwZsBFKt6  
KFZ1UKzqoViND1tgBBSreux8UaxGs8li9acX8p3RgXvnx+P7iAwIEau47D/lbff///jUFaq5IlZf  
fNH7wHDZZe5nvlG98ED3/9u3d/8fYEyrvywlG7OKd9v789hidaut3M/uucf74L+HYlUPxaoOilU9  
FKvxYQuMgGJVj50vitVoNlmsTn3fFatP7Cuy1hunisvR/3zIOqc2uS5WR4zwPjAExWqLFu7/n3qq  
O+4UDByYvywIE6vLI+fPEyZW773X++C/h2JVD8WqDopVPRsr8WELjBiVY+dL4rVaAo1ZvW5Zu7l  
/GePEfn4RpHHjXDFHf4/DvZm8AiK1b++cIUqPv+kl8jSv93PtxCKVKz+/HP+fOed544zRb31PwMa  
sVq7twvZAQelvP669+F/C8WqHopVHRSreihW48MWGAHFqh47XxSr0cQWq98/ldKvjsiju3sfelx7

RmTQgSKP7CLyahuRhVO9CRb9jIDqs7XItFHu/69f4z4Z4LHGloMPFpk/xf18CyG2WN1uO5FatdzL  
+j69jDivW1fksMO8DwyzZrnOaoMGIpdC4j4xAOv1133HHe4yBx3k/j/AMADc+V+5crpYnTZNpJk5  
icAzWe83JxibARSreihWdVCs6qFYjQ9bYAUq3rsfFGsRhNbrJI8YovVOFx7rStS/TGr5qBrJrSu  
UMWd/yUAiIU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVU6RidfVq9+5/PNoK60LgJv7TTnOfDIAC  
oFjVQ7Gqg2JVD8VqfNgCl6BY1WPni2I1GopVPUUqVnMAiIU9FKs6KFb1UKzGhy0wAopVPXa+KFaj  
oVjVQ7Gqg2JVD8WqDopVPRSr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj  
54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0w  
AopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRSr8WELjIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/F  
anzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWI2GYIUPxaoOilU9FKs6  
KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRSr8WELjIBiVY+dL4rVaChW9VCs6qBY  
1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb12PmiWI2GYIUP  
xaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRSr8WELjIBiVY+dL4rV  
aChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDSRoftsAIKFb1  
2PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+KFajoVjVQ7Gqg2JVD8WqDopVPRSr8WEL  
jIBiVY+dL4rVaChW9VCs6qBY1UOxqoNiVQ/FanzYAiOgWNVj54tiNRqKVT0UqzooVvVQrOqgWNVDS  
sRoftsAIKFb12PmiWI2GYIUPxaoOilU9FKs6KFb1UKzGhy0wgu6fGLH6sKII/U30NWJ1IcVqFHbj  
o1iNZsgvRqw+4NUx83fYfIrVKMo+asQq8oW4n91YFOPnGbFq+i8nX6Y/6/EZxWoUjZ43YtWvY31Y  
x6KgWNXT+l0jVq06NnM5xWom2Alj6G6KqUZ5ZbIpJd2h0WxGs0QU/z6hTLMFJKdsqbYOSPZGW+K  
na8eppDsNDLFzhnJDsWqntam+PULZaYpJBy2wAgoVvXYHRbFajQUq3ooVnVQrOqhWNVBsaqHYjU+  
bIERXGIXIXa6FamjRs3eINJGHaH5YtV5qwgfk5eNsWrYU4de8sUkp00sWpyBljHCuLnZJopyJNf  
bjKFZGcPU/lyxjqWETsndt9/7LHHep+STJxvilfDnDq2zBTWsXAoVKNYunSpLFq0yG10GEvilVKm  
pKqnZMWKFbJs2TJvbgIWLlwo3bp1c3JWqpTJk9VpnXDCCU5OST5wldauXevm6Dy3fuWVpin5448/  
nDpl8pk7d64MHjzYzdm6vGw57bJOhtQyZs0advQW69atk/Xr17v52s3rv/xyc0pGvTtK5s+f781N  
APLx9ddfuzn7JS9bbu7MZ8gn6hnJZ9WqVbLbbrs5+Qn2/f369ZN58+Z5cxKwePFimTNNpujF936  
heLUsdlpWb16taMxSDoUqwF+/PFHKV26dH6DG+hXJa80NmE+x93I06ZN85bKTXxhgMs9efnKENvv  
vbczfY4LCvx2dEQVK1bMz01HE3aBePWmvfnmm96Suc2FF16Yny+EJVad4n2+/fbbO/Pneh0DNWvW  
zM/Xvk6W8suDJrxdp9xxhzN/rvPwww/n5wvxqwm7eJ9Xr17dmZ91TKRBgwbpOcsQuOudiHzyySfp  
gn64CbvsYmJ8Xr58eefkPJfrWBCKVYvHH3/cqShpISmDWPXnefbZZ72lcw+cMTdu3Dg/VzGiVq1a  
8ueff3pryD1ee+01Jw9pdSyLWEXcfff3tK5B1ybFi1apOXDiQxi1c8rHtWUq3z++edpuXAii1hF

dOnSJacPjFdccUVaPpzIIb9vH788cfe0rnHL7/8kpaLOHHMMcfl8uXLvTXkHg8++KCTh7ScBcXq  
9iaseYYPH+4tTShWlflqkB0ZxKofZcqU8ZbOPT766K00XMqNXMrNVWrXrl0wJxFiFZGrzJ49u0Au  
nMggVv3o2rWrt4bcA2MFg/mIEquIDRs2eGvIPYK5cCKDWPXjoIMO8pbOPa677roC+YgTU6dO9daQ  
e4TII5NY9aNatWre0oRi1ePqq69OqyR5ESFWcQZ0yimnOOMyS3QcZiJQDlxwoKTeM3n4UBHvp6TJ  
P02stXhlsQlsl2zbJSROPvlk5/KOXX+ciCFWsWzYOKt6ZBxiEiFWd9hhB2nevHnoOktyntHTSSbL1  
1lsXyEccsXr88ceHrrOkx4knnlggF05EiFUMB0C+w9ZZ4slqJ5rS8LeG4f17tng3JUeuPNJak1e+  
NHGUibDtlpBA/x2sP05EiFWYYffff7+nUnbilWPww47LK2S5EWEWM2pSLLMMxG2zVyIGGKVEYgl  
scoIRAyxyghEhFjNqUiyyG4ibJu5EBFiFdGuXTtPpeQ2FKseBx98clFK4gTFan4kWSHW8wvFanRQ  
rOqCYiUfFKv5kWSHWm0vIWk1TZs2nrJbShWDf6NBcFK4kSEWMWTA9544w158cUXS3Y8YyJQbpp4  
k6TONnloZ0UbE+eaaOVFaxNtTfjTzbQuo7viUFPsdb242gS2EbbtEhKoJ7h0aNcfJ2KI1VdffTV0  
nSU9BgwYUCAXTkSI1WbNmsnLL78cus6SHCNGjJAmTZoUyEccsfrCCy+ErrOkx7BhwwrkwoklsbrT  
TjvlTru0CI5a0uKlFm5fb/f9UdEyJQ/OftBak1f+NvGcibDtlpB4/fXXC9QfJzhmNTYUqxb7779/  
WkVxlkKsHn744d7Sucf3339vjcEsJeXKV5QWjWvLnc1qy5ddqsnQc2rK1UfulHVRVJFU6TLOfBD3  
b7/9treG3APjm+3640SEWN111129pXOPBQsWhAv8CLHat29fbw25R+id7RFiFTf+5fiNVnjkmZ0P  
JyLEaufOnb2lc4+BAwcWyEdUVKISRf755x9vDbIH6ElkhFjlixXyoVgNMGTikLTKkk2s8hmYLuUr  
VJDDG1aRfi12lNk3bCUvn1tH5I4Ksu62Skaw1pXX220nt5+4rZMzlvLDDz/k1y9EFrF66623ekvl  
NgUekRYiVv3HveBh+LkORH5avrKIVZxAEZG2bdum5yyLWJ05k6/FBMhFnMdX1atXz1sit3nggQfS  
c5NFrH766afeUgRQPYTw7rvv5j8g+iGnGjnFecPEViI59NFHZfTo0d7c5LGH7pfVN5WTTzttl4+f  
ur3I7ZVk/W0Vzd8K8tmF28i47jVl8Y1V5bk29WXBslXeUrnNTz/9IH+Ju5VXt/w61iwl99xzj7z0  
0kve3LmNP0wHbinGbzk5s8QqclYuVU4eeugh5w1DfJC2C56DjL7KyddO+XXMKbek5IruV8gjzzi  
zMucueBZ29dff72bs0l52XJzZz577LHHctodtPHrDASyF/UjKFzPOOMMue+++5z5WMdcMHQEd/g7  
OXrerV8oTh0r7dYxvEWNpEOxGsFVpuRVp43mjNoUNrp0erdsIktuqiofXrCdfN5lW/NBBUesltbd  
WlFGnLe9/HaVEaxmnmdet1bivj16GVT7Dr2likkO2VNsXMG2C4L4udkminlk19uMoVkZw9T8jLG  
OpYROye2UOUl7GjON8WrYU4dW24K61g4FKsRdDclrzKZMtkUkk7vU3aQ1bdWkZdb13Mu/ctt5WTl  
TWVlba+yIreXI58u3UY+MyJ2Za+q8thTL3pLEZ8hphth1DDcwkOykiVVTSHbGm2Lnq4cpJDunTLFz  
RrKzcuXKNLKGZyST7LQ2xa9fKDDSDshgRFQrEZz2xUXyeIbKsvDrRrLZW1PlbPOay9ntOsiLdtc  
KG1bt5IHOh0mb5y/g3zQvqpMns7GGIRiVQ/Fqg6KVT0UqzooVvVQrMaHLTACitVo7rntFpl3U1Vp  
17addOvWTS666KK06NLtUhnUeV8ZelZIwbSCN78EoVjVQ7Gqg2JVD8WqDnOsMHIAAOguSURBVlpV  
PRSr8WELjKD7fUasHmCq0UEmGhmXoOoNi1ebvWbOlZ8e2MueGinJFh5ZGrF5cQKy269RN3r54JxnV

oYq8/OoH3pLEZ8jbRqzu49WxvY1Y/YBiNYqyhqxinwh9mQ3FsX434xYxZNMkC/Tn/V4kGI1ikZn  
G7Hq17HdWMeioFjV0/oml1YP9OrY7kaszqVYzQRbYATduxuxajXAYZMpVm3efe0VGdTpABnWYXsZ  
eME+csuFzeWWTiemxb0dm8qtZzSSzzpXIVfuvUwWLLnmLU1A8HFpeEg5yU7ZskasWjkj2Rk/3ohV  
K189elCsRtGokRGrVs5IdihW9bRubcSqlTM+EI0zbIERUKxG8+A9d8rrF9WXb6+uL3J7RdlwmxW9  
K8q6W8rLbecfKS+cU1FWrvCWInlQrOqhWNVBsaqHYIUHxaoeitX4sAVGQLEaTf8+98jLXXeS965s  
4ohV/7FVfsy+oar0vOB4edGI1dV8KkcBKfb1UKzqoFjVQ7Gqg2JVD8VqfNgCl6BYjeaunjfUx13  
IdeuOFA29i4oVv++oZZc3u5keeHsKjJ38QpvKeJDSaqHYIUHxaoeilUdFKt6KFbjwxYYAcVqNNd1  
OU/uPbO+PNlu51CxOu2a6nJa0/3lzbVZNwvv3tLER+KVT0UqzooVvVQrOqgWNVDSRoftsAIKFaj  
6Xd5W3nn4oYy/NLMzmrP9sfJr5dXk2/GT/GWlj4Uq3ooVnVQrOqhWNVBsaqHYjU+bIERUKxGc3vr  
g41lrSBTrqouM27YVqZet61Mu96NqSamXF1LZl9fWebdUFveeu19byniQ7Gqh2JVB8WqHopVHRsr  
eihW48MWGAHFajS9z9pHVt1cSa49obFc2LGjdOnSJS16dZpOVveqIAtvrCLPvcz33gehWNVDSaqD  
YIUPxaoOilU9FKvxYQuMgGI1mt6XdRa5s6xMuHwrueyiTnlvsepmoutF3aRvh4NF7iovn3eqKhN/  
/8tbio8F8KFY1UOxqoNiVQ/Fqg6KVT0Uq/FhC4yAYjWaxwcNkWXXlzGCTYI802YnubBtS7mia3vp  
fmFb6XH2/8nMq6sasVpBbj2pvreEx1Wm+iHmK8exrloo8vKZlreaZR/dXWT5bJFFf5iYIbJ2uTfT  
lgPFqh6KVR0Uq3ooVnVQrOqhWlOPW2AEFKvRrF2/UQacUF7knnJGsFYUua2CrLqloqy91fz7jgqO  
iJWeKelzZ29vCY8bTfVDLFA+IWDsA2YbZrm7q5hIPaF7i/I/sw35+SX3/7cgKFb1UKzqoFjVQ7Gq  
g2JVD8VqfNgCl6BYjce4nybKy4P6yydP9ZbTdi0I5+xZ0UQFOWvPyvL+M/3k5ZeGy/r1gddXhYnV  
datEfn1T5NsBlj8MEvnna2+Cx19fiLxwoitW76ok8tMLZuNPu/8Pp3VEGzPPI97MWwYUq3ooVnVQ  
rOqhWNVBsaqHYjU+bIERUKzqsfNVqZIRIjkiIw4pPfXEbmzvMh9NUTureYK0JdOcaeD0beK9Daf  
OVFKpP/Olg/Wy//sdvPZZ3d6M28ZUKzqoVjVQbGqh2JVB8WqHorV+LAFrkCxqsfOV2yxun6tyFud  
RXqZ/3+4oVFwpqMbflDIHWWNeC0nMvIVbyHDF/e5IhaC1gfO6s0mJo3wPthyoFjVQ7Gqg2JVD8Wq  
DopVPRsr8WELjIBiVY+dr9hiFTdGPX2kK1aHnynyalv3kv5r57v/nvGJt5Dhs7vDxSqW/XnLE3oU  
q3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfMVW6yuWy3y8tnu///+jjeD4Y/RIIPeFFk2y/vAQLGa  
81Cs6qBY1UOxqoNiVQ/FanzYAiOgWNVj50s1ZhXuKS7731FG5IUTRAYf4k7HkAD7kVRhYvVusx08  
EaD/LubIPMT7cMuAYIUPxaoOilU9FKs6KFb1UKzGhy0wAopVPXa+sorVwYe6sfQf7wPDkj9F3jch  
OhdPENm1nchvb3sTLCYMFxn2aJFhp3sfGBb/Ifj6B5Hnmon8Yo1v3QKgWNVDSaqDYIUPxaoOilU9  
FKvxYQuMgGJVj52vrGKVOFCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqh  
WNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JV  
B8WqHopVHRsreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQU



q3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8  
UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZA  
sarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViN  
D1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfF  
qh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6  
KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY  
1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoN  
xaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47  
XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIER  
UKzqsfNVqSLFahQUq3rKlqFY1UCxqodiVUCBsXoUxWoUFKvxYQuMgGJVT+ojk6vRblQaQ7EaBcWq  
nrKfGrHq1THUN5IdilU9jb4xYtWvY6NYx6JYucGIVZMnP2dNx1GsRkGxGh+2wAgoVvWYTOWVSqaQ  
7FCs6ilril/HUEh2KFb1NDLFr18oJDsrTbHz1dQUkh2K1fiwBUZAsaoHHZVfKFajoVjVQ7Gqg2JV  
D8WqDopVPRsr8WELjACdul2ZZs+eLRs3bvSmkjBSG9FVucUXq8xZQfycvPLKK2I17J133nE+J5IJ  
E6umvgHWsYL4OZkxY0ZaHbv55pudz0lmpji1TDWsSzYObH7/uNMldlp3759WrtctWoV61gGKFZD  
mDJlinz99ddpIQhRqlQp5+8PP/wg06dP9+bObfyG9cUXX8hZZ53I5skUkzG3rEjJQTsdJF999ZUz  
Hxuiy7///is//fRTgbrl/x01apQzneTz+eefy2233ebmbJ1Xv0xBfauaqio//vijbNiwwZubrFix  
lrSO+TF48OC8dklcvv32W3nppZfcHE1C7XKL06eZzyZMmCBLlzx5ibg+++/l+22287JT1rf/15K  
enTp4bRbks+kSZOcnPjt0A+/felqyN9//+3NTXwoVgN8+OGHBSprpkAjsI77LBDem7sstKE93nl  
ypW9JXlbuPNp+coSgwYN8pbKbY499tj03Fhi1SnWNJDLJ0X+b7dzki26du3qzJ/rXHfddem5+dWE  
XaxpgHUsUMcqmbDLGBPtiMPptiZP9d5+eWX03OWJaZOneotRQDFqsXFF1/sVJKgAxEW/jxXXXWV  
t3TuMW3aNKITp06B3KQVS6wilFhzWeT37t3byUOcOubHeed5y2deyxbtkz22WefgnnJIFb9vL7x  
xhveGnKpp556Ki0XceLoo4/OafHVvHnzgnnJIFb9vD766KPe0rnH+++/n5YLJ7KIVQQeBbZgwQJv  
DbnHBRdckJaPbOHn9aabbvKWJhSrFsKEyFkPmp4S+ceo0ePDs1JWgmIVcTTTz/trSH3qFMrRNzH  
iFxl3rx5ofnl5qwiunXr5q0h9zju6OMK5CNOSA6PoAjLRzZnFZHLbuENN9xQIB9RYhXxxx9/eGvl  
PYK5iBPVKIXzliY5exTsZ0pdU+p5pb4pqb9MBflbH1jWX09eOdBEvZld2267bWgDSyshYrVvmzZqh  
6ytx0cmEVVBPSs8sHVqHssafGepYWxNh2y1Bs322xeoP05EiNUqVarljjvuGLrOEhWHm7DKjqZU  
mFshvB5lC1PH7PXklT4m6pol23YJCdSTYP1xIkKsl9fXurXN+0yZJlOZCvatWqFchHHLGKsa1h  
6yxRUdvE8yassqn6otQ/pQr0/VVNWWZKrpGzYrWXKaY6JFfqmgg01JwJu4SI1ZyJ000kWU40Ebbd  
XlglSzoZ0dBEkqWnibDt5kJEiFVGIGKI1ZyJgSYSLetMyTUoVpMq25klq8S5EHbJZbF6mokky/Em

wrabC0Gx6kYDE0mWm0yEbTcXgmJVfXsr+THARIKFYjWH+NeU8VaZYUpqT1MN9tVF6f1Lyx+m2Oty  
yiQT40t24E710IZqlxCxeuuttzqP1AlbZ4mK6Sasgjq2dbOtQ+tR1tg1JVNNsdflIGkmwrZbguLj  
jz8uUH+ciBCrelzaL7/8ErrOEhW/mLDKb6Yc0uWQ8HqULYzoHbdhnLUmr/xrlmy7JSgmTpxYoP44  
ESFWmzRp4tyxHbbOkhzIV8eOHQvki45YHTlyZOG6S1SMMzHfhFWmm5La3eQgrO1licqHV3aOG/a6  
fjBlvSm5Rs6K1TA6dOhQoHFICv9uvYsuushbOvf47bffZJttimQm7QSEKuVKIXK6Wc7Xn/99U4e  
NHdqn3766d7SucfSpUulcePGBfOSQaz6ecUjYnIV3KVu5yJOHHbYYTn9NAA8DaFAXjKIVT+v/fr1  
85bOPd566620XDgRIVZ33nln54bJXKVvq1Zp+cgWfl6vuOIKb2ICsRrg3XffLVBxMgVeHEBEateu  
nZ4bu1hitVy5ct4SuQ1eqZeWrywxYMAAb6nc5qijkrPTRZnlbjYOckWcMmIOI8hTMtNFmd1/frc  
c7bCsHOSTazut99+3hK5zYsvvpiesyzx66+/eksRwJ49hJ9//jn0sUz+2Q7ePoG3XJH8h0PjZQpw  
AJ082W8xWWE6qvr7OZdzQS67NzZ4Qwkc5mDd8v++/fbbzt0SD6oYzfeeKObM0usor5VTIV22iXf  
YJUPnIEbVsf8gAP7ySefeHMTMGbMGHnmmWfcHIW8wQoGxaJFi7y5Cfjss8/yDlvgG6wuveBS+eCD  
D5z52Pe74E17YS8f8tvn2LFj+YbMEChWI+jRo0dahcLbh9josmO/H7qSKYA5K4ifk1deeSWtjr3z  
zjvO5yQzZU3x6xf254ZPyd4vqVdx26++Wbnc5KZJqZ4NYx1LAT2Tuy+/zhTSHbat2+f1i5XrVrF  
OpYBitUlunfvnlaZJk+e7E0hmUBH5RdfrJLMDbkyJK2ODRs2zJtCMpEmVk0h2cGNH3Ydw0k4yU4j  
U/z6hUKys9IUO19NTSHZad26dVq7xBAXEg5bYAUQq3rQUfmFYjUailU9FKs6KFb1UKzqoFjVQ7Ea  
H7bACChW9aCj8gvFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqSZUxuarhRqXtKFajoFjVU7aW  
EateHUuVYzcWBcWqnkYHGbHq1zGTM5KdlauMWEX98nLW9BSK1SgoVuPDFhgBxaeO194rirJDsWq  
nrJlJVi1ckayQ7Gqp1Ejl1atnJHsrFzpiVUvmjalWI2CYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6  
KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY  
1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoN  
xaeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47  
XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIER  
UKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW  
48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVB  
saqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8Wq  
HopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3oo  
VnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxG  
Q7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarH  
zhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tg

BBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K  
1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1  
UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGx  
qodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzqsfNFsRoNxaoe  
ilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MWGAHFqh47XxSr  
0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqHYjU+bIERUKzq  
sfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopVHRSreihW48MW  
GAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUZAsarHzhfFajQUq3ooVnVQrOqhWNVBsaqH  
YjU+bIERUKzqsfNFsRoNxaoeilUdFKt6KFZ1UKzqoViND1tgBBSreux8UaxGQ7Gqh2JVB8WqHopV  
HRSreihW48MWGAHFqh47XxSr0VCs6qFY1UGxqodiVQfFqh6K1fiwBUbQ/aKAWP2FYjUKO18Uq9EM  
eTogVodSrEZRnkWxqoFiVQ/Fqo4CYvUlitUoWp8TEkv/UKxmgioWgu5LjFj911QkxEwjVtdRrEZh  
Nz6K1WiGrDRidZZXx8zfYasoVqMo+68Rq367NDkj2aFY1UOxqmPIRiNWzTHSb5dNF1CsRtF6kRGr  
fj9mcjdzPcVqJtgCI+huiqlGeWWyKSQ7dgdPsRrNEFP8+oUyzBSSnbKm2Dkj2aFY1UOxqmOlKWIL  
fmlqCslOa1PsnM00hYTDfhgBxaoeu4OnWi2GYlUPxaoOilU9FKs6KFb1UKzGhy0wgqtMyatKG1My  
y5SNGzd6U0kYdgfvi1XmrCB+Tv5nCuqWX8feNoVkJ02smpwB1rGC+DmZPn16Wrvs2bOn8znJTJMm  
TdJyBljHCmLnJK8fM+VYU0h22puSlzGTuxWmsl6FQ7EawqhRo+TJJ590O6mH8qqSIDlIVT0ItZ31  
IHxxxRfe3LmN37D69+8vRx11lJOzUqVMnqxOfrrddpNHH33UmY8N0WXixIn5TwFo5dUtv44dk5KH  
HnxlXn31VW/u3MavM6hDF1xwgZuzdciWW5CzcqlyTptdv34965jHggUL5Omn85rh8F2ed1118ng  
wYO9uQlAm7z11lvT8oTwc/fMM8/lrFmzvLlzG7+dPf7441KjRg03T+i//PJ+Slqd1EoeeeQRZz62  
S5e3335bBgwY4Nat5/Oy5eaulFvHfvjhB29u4kOxGuDFF1/M66CcGOhWpLzS2IQ37d133/WWym3K  
lStX4EAYDH86Efn555/T89PRhF3OM+FNU/POO72lcps999wzPWeWWHWKNY2ILFu2LC0n2eL000/3  
lsptOnToEJqfsJg3b563VG6DXOT1/ZVM2GWMCS9fDRo08JblbSDc/Zw4MdyEXbY34U0bO3astxQB  
7NktDjvssPSKhMgiVhHNmjXzls49xo0bJxUrVklR1SUKVNG3nvvpW8NucdZZ51VMC9ZxCpir732  
8pbOPRYuXCi1atVKy4cTwcQq4qGHHvLWkHvAMUuOok4g7ahXr55s2LDBW0Pu0bBhw9C8hIWf10su  
ucRbOveAlx/MSzaxithqq61y+jmi+++f1o+nMgiVhGnnHKKtzShWDX4lyfsSpIXEWIV4itX+eLD  
L9JyETeee/w5bw25R506dQrmJEKsInKVJUuWFMiFExFiFS/zyBkCV1ePP+L4AvmlE5K7WjU0H1Fx  
yF6HeEtb5MiVbox5LpCTCLGKyGWxGsyFExFitVq1at7ShGLVI9RVRUSI1ZyKJMs8E2HbzIWIIVYZ  
gYgQqzkTDU0kWXqaCNvulhxfm0iyhG0zFyKGWGUEIkKsItq1a+epINyGYtWDYjVGJFkoVvMLxWp0  
UKy6QbGqD4rVZIJiVR8Uq7GhWPW49NjLC1QSJyhW8yPJMtdE2DZzIShW9UGx6gYObkVQ0u7itssN

JsK2uyXHVyYCJePv35QSts1cClpVfUSI1dKIS8s999zjqZTchmLVwq4keRFzzCruvi3xESgJR490  
81BVEeVS8tizjnPk7PXtWyjibBtlqAA2267bVr9cSLmmNWwdZb0mDZtWoFcOBEhVjt27Chr164N  
XWeJihUmrAKOPuXo8LaXLuzOlmxYYq3JK2tMhG13S471JqyyyhSn3oTlJUvsvf9T+Tr7tdTkIbJsl  
KFavXi1XXXVWVWntzloZYnTBhQug6S3qAYC6c4JjV2FCsWjz44INOBUm7izaDWPXnwTPmcpXZs2fL  
rrvump+rGFGzZk1HgOQqQ4cOdfKQVscixGqvXr28pXOPIStXygknnJCWDycyiFU/r99++623htzj  
o48+SstFnGjfvn1OPwezW7duoXkJCz+v77zzjrd07uG/ES2tjkWI1SOPPDJPuOUid999t5OhtJxl  
EKv+PHjuL3GhWA2Ag1xeRUJkEau//fabt1Ru4h/cDjnkkPScZQi8HADk8kERvx0ddtmyZfNzk0Ws  
jhgxwlsyt8G4rbx8ITKIVTzmCuR6HQNvqlRJz1mGyOWTIZu+ffuG5icYeFwfYB0LPN0ki1g944wz  
nPlznQ8//DA/X4gMYhWX/2EG5XldC0KxSgghhBBCCCGEEEEIIUpCjVW8Xnv27L9l0fzfRdZOOFvk3  
VZYt/l1mzZohq9es9eZyWbFipfz66xSZMGgi/Pnn396nhBBCCCGEEEEIIYQQUnIJGkvuUN61q5fl  
uy/sLmNGtpT583+VRYv/lHff3SFvDq4i/47bW+b90lS+fOtgEXNIY3nsvgPkputOl7EfnibDnz5l  
ttl2Oxn81EvOegghhBBCCCGEEEEIIaQkkmesLly0VAY+OUyGDz5Kpn6xk8yfP0kWLBWZM3+OrFyz  
Uib9PkdGvPW+zPr5WPn9y93kpSEXyrTpk2QdlI2yVH6a/K/88N1w+WfC0fLCoLbyxJMvyqpVq92V  
k+Jj3e/yz3dvYz3DpCX358kP5tdsK7Qj75YZWKafD90hLzz5LvzybwNsoCP0yCEEEEEIIYQQQggp  
cvAM0wULFsg///yjjpkzZ5boFxx8/vnnzvOXf/75Z++TaPBms+eff1769OlT5LnJM1ZXr/hDxo06  
WD4ZWkbeG9piXhtxr/w08XuZ8tsPcvwJR8g5p5aXr9+sK8t+rStTvm4uD/e/V1p36Ci97rhLZs+b  
KxN/HS+Txt0tq6fvLueclJlmex4gCxcu9NZO/juKwgHffnxXHj3mDLlw967ywKS1MmW904UQQggh  
hBBCCCGEFB0bNmyQSZMmqV9ECQNX3Lhx8vffJfdRnWvXrpUff/xR9t5nXzn08MPI448/lkWLFnlT  
84E5PX36dOnTt69U26qG3HjTTbJw4QInt0VJnrG6ZuUfMuWLQ+SLEVXly9d3kiUTG8jSXxvlnHE7  
yorf6suav4+Sff+eLyumHyiLJ9WTuT+Z6VMPk1/HniDDBx8jEz7ew8xn5v19F/nurWry2VvNZe2a  
xd7aM4HxrpPk2+cHyB0HdJGX33pGHnvYJjnzXj3ZrebWskeLi6Tj6wtk2to1snryMHm2TRO5a8Cr  
8sJ08339PKyeK/Lzo3LHLffKwd1ek7HzMGGurJv6IPQ6/iA5sFo1adh4Dznh0QFy2U0PywMHdZFB  
b/4iX8Nv3Lhe/hh2gdzfqrss1LCBbFertuxwyKmyy51fyttTMNp2jYnv5d3e98hdR18ir38xVO69  
pYuctE1tqV9nOznphufkyV9E5n9wqwY5dH/ZbvsdZdva28l+Z18p3UculKlpvvlEmf7BbXLB7jvL  
XrVrSu3tD5C69XrK7f/7Wf40U4M+5d+j+so7tx4kfd+fIR/OwieY43f5cfgAuWvvc+XJtybLt45n  
av6z8jP54fNXpGPvMfl9jO9k/k/D5N4Dze96Zlx8ZH7CRvMbRt3VR+4+rLO8+fxtcnH306V27bpS  
t+4OcnDHO+Taj1bl3OVYl8/3Mu6l6+S8hvVk9xo1pckZp0iroU9L1z3OkOv27uYYq786+v9pUvS+  
vHzjuXLittvKLvXry7bbbiN7Hd5crrlgqYzHV175r4x/8Ch5+tGb5eHxlv+aRUQWybr1X8hzZ18m  
97R7UF6dlzlbH280P3T2COn35FvS84kJsMzt0VZ2QgghhBBCCCGEkM0d31idMmWK90k8Vq5cWeKN

VRsYp4MGDZa9991PzjzTOd3w3gdMeJVabDTTtLhgo4yatQoWbFihbdE0ZNnrM6fv1DO79hNWp17  
sgx8oIXM+nEXkb/qyerpu8qK33eV5VN2MtFAlv+2sxe7yLjpB8u6fw4WmbWTFp9OHXnlie3k5Sd3  
luZHp+SkEw+SxYsLOsbp4EVYP8rnD3eR9qltpdHx/eSqt+c6dibM0Rkjbpdl628nne//SIb9uVRk  
Wj/p1ulOOaTVazJtjWtFLvpjjDx99v7S5/5n5OW55oON78n/rjIHmm91INz27nyZ4MxlmHKfDOp0  
INQvc4xc9MyP8nHeUwpmY5LpX8nLQ4fK8FfvkT7XtJVmqX3kguvfkuErYUp+Jm9fcoq0TDWU3du8  
KA987728a+UwebbzIVInVVMovv1LecExPw0bXpN3enWWnVlt5ephE2W8mJ235Hm55+BDpGn5g6TV  
vc9Jn5fcllcH3ysvX3ugHNmlt+zb93eZtxwms8X8UTL3/etlI3Mfl06Dp5rvsVbWvNIDXu/VVE6/  
uJ3s3uE56fHCH7Jclsn0ge3liSvbSOsRc+WHNVNk5pgH5Zlyh8tVN74lr67yfkO3k+SMsrVJbjeO  
lmeme9tY+YI8deH55ru2kds/nSnmW8iamU/KIU0OILobXS+D/hF8e8NUWf5hZ2nTaH9pstcV0m/S  
epkjb8j0d26UM1PbS7tLh8qLeflcLmt/GSo3Nakn7S68W/pM3Cjy88PyWN/bpEbr1+Wzv8z/z/pM  
/nq4ufTs0VwO7nijHHDZp/LxH6tk1fQP5LNL95PL+j0vN47dKKs5KpYQQgghhBBCCcnxjBw5Uvr2  
7SuPPPJI7Ojfv7/069dPxowZ462I5EBjVc/Axx+XVCrxF577yPTpk71piSLY6y+PGKEXHXtVfLJ  
Z2+aeFdeeX2EfPb1Z/LWq9fLi/3ryF/f7OTc4r/k191k8eRdnVj0yy6ybkZ9+fSN/aVI6zZyYfdr  
pMXppzs/4KmnnnJWHg1Myu/kk77XyIVljpbabhsvH+Jxng7LZNqoPvLAoaWly53D5cFf3TGbG0bf  
IG/3O0eOuH+SPNnvOvms3yly9IM/yFPjscxSkbnPyMBbb5djjntCPpy6QNYBmDATv5VPH7pOOpVq  
KjcNmShjls+RjZP7ycV7XCinHP6ovG/mWCsbZOGPr8qA/ztC7uz/hvxvGUzJMTLiwgukU80z5e4x  
c+U7x/+Ei/iVvNvrMrmw/Aly66u/yRjHDcY3HCdn7xZuqT+T258Ypx8snyByNT+cs0BnaXFQffL  
W2aWeA9IWCPL530rQ888Rfrd3EX6TBgpnVs+Lo/eM1Kmy+8y7KLb5Znu3eWtyQ/KmR37SYfu78rv  
GzCEdar8PuqBgLGa6Td8Le/ecrl0Kmd+w+t/ynfL/pA1P98v7U+/Q7pc+65Zk7uHRObLxvVvS/+j  
T5eOjbvLA7+sl39XfilTht4szUp3lGv6fGyy689rGvLsX+TTHjWI961XybVfiiww++X3YffLQ0cf  
Lg9+8oRc/9Qr0q3pAPnit5/l+7Hvy6P/11nefP0Wuf/1B+Ww/3tEnnnvd+xJ870JIYQQQgghhBBC  
cgsaq/HA81LHfPaZ7NFkTzn40MNk/PjxMnv2bHn88celQqXKcvXV18gff/wh69cnN3LPMVZn/jtL  
fvjpO/l58ljp0q2NbFunhVR5oLf57Av55bfx8tBjA+WkU1vKro0byVW33SmDr7hKHqldS1occJA0  
u6Cz9LztNnnmuKYyYN8mcvHFF8nn38FmiwOsuO9ITP9rpWNqdzn3tBbSvN3pcsAB+8t+u1SRg/7v  
VDn1likyauo6y2RbL/MnvinPXFbDjj6VNn7ylfkp7l5wyUNA2T12AEy+rL9ZOfG+0ulHfaVQ484  
VM7sfZmceXprObXcUXLtkAkyeuV62Th7pDzW+Wg5rmFN2fOAA2SvM9rIYSecLSfXaSzD+7wpw5bD  
3PtIXmp9trRKHS+9Pp4jXzvuldzfz+WNqzvJualD5bphk51b7l1j9Tv5rH8PaZPaR6548At5y5l/  
vqz4+gEZeuXxsmetPaRRw73koAP3kwP320d2a3uXtH52iixaGRixCtYtE5l2vzzZ/kjZNXWAnDXg  
e3nVcWXNdv4ZKG/fdLlcWO4wOe2uj+WZmeZjJ0mT5Jd375ILUk3koitelZccYzXwG5xN4Td8IW9e

c6H5DYeY3/CrfIKP1/0tS5/vJPe0bCKldzhQdtx1Hzn2nOZy7l2XydHVD5BTGlwgd01YK85TPpb8  
IDNe6irXn3KUNNluPzlg/wNk30b7yuEht5IO93whr/28Ag8LcFk6Vua+21VOq7mr7HPI1XL9jxvl  
D6Rr9QRZ8cXlcvHeB8oBe18iV3yxQiakPZaAEEIIYYQQQgghJHfY1Ges4jb4XDBWZ8yYlae3bClH  
/N9R8tBDD8u0adO8KfkgFx9++KFccumljvH66GOPyZo1a5xnrxYlJrHav/8jcsQRh8uNPa+SZ194  
0Gz4CflyTD8Z9nx3ebjfufLNV4/lovnvydoVo2XOrFdI2Fv3S7tLOkvfffeRV08/WV5960V58NIB  
0vbii2WvAw6QEa++6qw8GriO7ojVzmWaSa/hU+VLd0IWNsrCn9+QAec1kt6PPCvP/yWy0jaeM+Xn  
z3vkmYvPkQbVb5CHP50mZrF4FFjfRuejgvvB/SDj585/s5O+bKblIj4PeyxpgZkz/QZ34XhPNvXW  
4f5PBvyp7rwF0X5OCCGEEEEIIYYQQUrKB+Yc3/E+ePFkdv/76qyxYsMBbU8lj2LBh8n//93/Os1Pj  
smTJErn22mvlzDPPLPLc5D1j1Wf40Pul7ZnV5aHbtpHv3tIR1kzf1XnO6sqpiF1k7Yxd5a9vd5IP  
h+4gbz2zg/TosrWULl3aeQQAftjSpbiJOy4wVr+Vj/v0kl6po5xHAYyyB5/msVJm/fCUDDimrOyW  
qipb73mx3PH9cvnVGQ1qA0twvwx44Qrp1SQL25Z3n62QSpU3cbw06zhUPjRzLHHmJYQQQgghhBBC  
CCGEkE2jgLH65x+/yBdjhsuvE16R6ZNfkck/vSKffzpCPvxghHzY0QgZ/90rMu/vV2XD8jfk8zGv  
y9NDXpLnn39e5syZ463BddZjU2DWkGU1wxc51JEQQgghhBBCCCGEEJlwBYxVQgghhBBCCCGEEEEII  
IdmhsUoIIYYQQQgghhBBCCCFKaKwSQgghhBBCCCGEEEEKIEhqrhBBCCCGEEEEIIYYQQooTGKiGEEEEII  
IYYQQgghhCihsUoIIYYQQQgghhBBCCCFKaKwSQgghhBBCCCGEEEEKIEhqrhBBCCCGEEEEIIYYQQooTG  
KiGEEEEIIYYQQQgghhCihsUrllszGjSLr14usW+f+xf8H2bDBne7Pg/+38Ze35/HB+uxpCHt5e91+  
hH2HwlJc27Epjm1my/3mTPB723UirM4UVd6SXDcpOoJtZ0ut10X9vcPqL7YRVYf/i/5vS6eo9mVx  
5f6/2MfFsc2w/YDtZgPfwV4GEbVMrhO1L4uqPWjBd8C27O3j33HrWdLfOyxv+Czq+wWXK+rvtTmC  
nNi/W7MfNUTVZQ3B+oN1J0FxbccmuM3/sg76dSPsO2lyEbWesHqAz/z5M81DSDFBY5Vs3jz0kMhR  
R4mcfLL796KLRBYs8CYWMDomibRpl3L00SLNm7t/n3vOm7gZsny5yJVXiSoYFqyaco33igyf37+  
QQx5uusukQMOcKjatUSueYakb//dudZuVLk0EPzpyOOPdY9SIFPPxUpVSp9er9+7jSA7dvTEN9/  
700sQm67LX0b+M3ffONN3ET+/FPk/PPd/XzSSSJNm4o89ZQ30fDYY+nbRLz/vjexCFi0SGSvvdLX  
f9pp3sTNmFmzRHbaKf17t2vnTTQMH54+DfHii97EQvL88wXX/b//eRPJZgPqsb2PttnGrTebO6ee  
mv6999IHZPFib2IR8O676etHVKniHnf+/debKYQ+fdKXKV1a5LPPvImkANhn2Hd2zlq08CYqwfHS  
Xg+isMeeMO64I30b5culFPWVN3ETwXG+fXv32OYf4wYN8iYaHn88fZulkSO9iUXAKiUi++2Xvn7o  
if33Fkx1ypspB9/dNuFvVzPnt5EEsr116fnq2pVkZ9+cqdhP+y7b/r0U05xpyUFdCj0M9pdjRrp  
24aWf/llV2c+80x6/ezYMB8vnDNHZNdd05c991x3WIEBPWuvH4Hj1Q03iKxe7c0UARsJvUzDhilz  
Z3oTSyhhfily3nn5v7lcOZFmzdx9WJQ88UR6bhFvveVNVDB9esF+BMfaogb7Hfvf3g7qR2FB+znm  
mPzzUeR+Xgx32tKlbj9qbxNtqLiZMkXksstEDjrIPb+0v48f9euLnH66yGuveQuFsGqVyCOPiBx/

vDu/f15rB/qRww93j5U4fwLoZ9BfXHKJOw++wz33uNMI+Q8wtZCQzZiuXdM71j32EFm40JtYxPzy  
i0jt2unb693bm1hEFNWVtJ9/FjnzTPfka/vtRQYOFpnrL2+iB0TFzju7v2PbbV2TNAG0ZhCqvnla  
tqxlp075xipO3itXTs8JzO7iBgax/R1wgP3uO2/ijJ1qsgOO6SvF+a0z7BhrsD2p0Hgf/GFN9GQ  
bV/G2c842bENKOzLSy/1Jm7GzJ7ttkP/eyNQZ3xgHtl5rVdP5O23vYmF5M03RerWzV83BFhRmt2A  
V7uTY3PP7cUXu30g6hbMy5Yt3ROYOMT5bair/vrtKFPG7V8GD86/MGbwAPp8+Nk0e6LioKSVO+x  
z3Ai5+cLfWv37t7EzZR7703fx1ttVXgDFyfhO+6Yvt7rrvMmGnBRCTRAn7bLLiJxngTDdnqRJz6  
gmNc8MKtHzgB7tZNZN48b2aL8eMLnqTj4mpRUPLqO7j/ftdM9fN14IGu6QHC9gP0Y9IEL7DBOF2x  
wpvocdVV6fPYBiUukJx4Yv40mC240FGUQM/a27e3hRx+8IE3Y4ALL0yfv3HjLePi4aYSbC/2wAMc  
J4sSmO72uRjOY0aP9iZuhmC/Y//73xeB+IFYggNK0Ff/9ps7Dab/WWfIT4PJjYFHxUn//ukXMPH9  
YLL6F3TA55+7uUB7wjHtggvCQS02mAeDp6pVy18X/h+DNHyQY5wH4hiF6bg4g/MBH5zv0i3XrOkG  
jvfB7RBSDJjaSUhCzJ0r8t57lpdf7l5lgtmCjrN6dVfs47NrrxX55BNX+PIMmuQeUOXONhi4SorR  
hRxiStXI0a4VyJh+ECQV6zonkxhHeilfD69nXNNBtsF98RV/rCtoPA+mB4ff21u8zvv4sMGOBe  
8d99d5Gtt3Z/E0ye444T6dXLHXGRCYhd/+qmf5J9003exBhg+8gPTSaxbeQXo1eDYAQuvhPWv912  
rnEcBAbquHH5BjZE66+/5ouoKGMVxjM+841ZmF4TJngTA0AYYf177+0eYOvUcUUz9j9M0v/7P1ec  
4ECNv00GeAsagsYqlsflWHxXHMhX5Rb7CYrRINC6D35ZP7IXJ9ly0Q+/LDgyDQ7cFBGTiHuYK5C  
4OE7Yzm7ngJ8d+x7LOfnAFfb44DbVvC7MdoTy91yi7tvfSDog1fbMTLgjz/cq+H4rRCc+A7+7S8T  
J7omzAknuEYNfgtyhfqGq9640htWD4Kg/t55p8hhh+XnFTmAmMYoquAoEttYxahptMcGDdyA+Cmq  
iyE4Ccd+QV+C74DRVf6V6yDI5dNPi7Rt654gweDFvLrCEYAoP955x23Xti8+qorVP3fh3aK0bLo  
myASmzRx84oTWdRj1F20hWztHmDEB/YB+gnsO3wXrAftFPsMOcc+zMSaNW6bxLbQdtDfQcyin0M9  
3XNPt+6jjaJdB/FH2aBvxF+ONVxwQftCPIHf8Fvxm9C/YT9CwKOeol/D+vGbMQ/6HvTDGGUQNB/9  
k06/f8NoKYw+Ajb70G/icz/HaMM4mcbvOuQQ94IGvgu2gXqH/heGfZQZgn4Q+w6jntDvY3+jDuLY  
MHmyW3cwShzrRB9z8MEiY8d6CvxQNmA84TvhN6AvQzu1QTu1fxv2XZw2BcKMVex/30RCn466gTZu  
ozFWcdEN7fTss93RgrjwhvqBXGlf+8eojz5y65MNLtBh/X5fhuWQ92+/dUfLYL/hM9QBrBfruu8+  
tx8OAxoAbRS5RB+N4xCWR53FnRRdurh1LGi2FBbsMxwfMFoTv+Pmm9P7VoyW9E0o/7dCT+BErEMH  
t13utpv7u7EpsNox1+0jbC2BWBKok9Bu8I6MC/aNXKN74OTQf8Y16hR+h0SQWMVdRfbx4k0jkfl  
I98Xoz8/4wz3WBO8oIq+DL/PNpaDgXWgTeHEFcc47Be0bRxzgs4/KbgMQ76Jw5hhh60GfoqtH2s  
D+0HbdZGY6ziQh/qEExaaEr07fh9+M7o89GfQN/gdwaNL/QJ6B+wfr9dYuQndANGyPqjsNDe8Bft  
CXfpZBptCz2KfXbrra6JiP2E9u3XJ/TNDz5YcJ8VBai7MDrQt+Oul/QP+D4gbD9kMIzxsR2/AfnW  
/ga0O/QX0DT28dMOFd/0Q6hv0JZ+nQoGRurh92B9/nELx0cYKEHOOced7o9yw/cOhtMzETRW8X2w

r/Gb8f+or+i/gn2Uf4zzA+06zFiFoZmwig6HFPxu1H/0Q/iOldjOPQozk+Q9yA4NmL9fj7Rb+C4  
hBGcGA2M9oP66R/rsH8wEj3THRCoJ7iDCDmDjvH1A7T7EUe4Rje0uj+4lhMvwJD/27HNogRteuhQ  
N0/ID46b9p0j0HTYrq9joFOhH/C7cEzHcRT9Ni5Moc5iHnyGv8hnGNh36I/Rb0Jflp/Yp9ADWDc0  
EtaFPgD1F/P5OQ4zVnFsQ/vDeQzOP7FvUKf8Yx/OhXBcCF5IxTkq2heOHfb67EA/B00A7YqLwfgM  
g0JwrChO/HMXP5CbsDaAvsLfD2hbWysV2NfBfgBaLQx/ZCoC583Isa8LcZdm587uvkPfAm36zz/u  
NEKKCVMzCSlicLDBia0tjHFidbGJ29Bw8MLJKQSCPx1GKw5cOMhA2OGKF4wMHED8eRA4AcCVerww  
cUIMEWCPyIClhvjCdnBy2aOHK6796RBeMIHWrnW/K/5CROPkASebEBj+vAjfXPjhB5FHH3WFnj7H  
SSpO1vEZbm/AQQAnG/jMF+gQBDB2gh07zJmgWakZqYhbl3xjAlflcBISPPkHEFD+SQMEStyTf5so  
Y9U3GfyAUPSvVmJf4iALAYCTcHsenARh/2Bf4WQT4tVeDwlnmj5BYxUnuVgnhOXtt7uGAIQUBLV/  
EohAfUEd8G/pwv7G/sDvwomDL579wEkW9iWEcCZhagNT3l4egd9bFGDdQRMG+8LOFeohxBI+v3/I  
GL8Z5gDENeomTp4hMCB+/eVgNEKE+mYXgGkJsWy3S2wPBg1OcLDfcTIGQylogmxj9b8E+wzmM4Sv  
/f1Qx7C/Ybajzak92SNfldAhEn3D5PXX05dHoN7ihBXGKPoX3J4NcW2vByefMKD8No92CbGLuolp  
/nxYBm0AfQbaQevW6f3UkUfmm7RYB8wyCEqcQPnz0J5jP0OE4/ugvaC/8k8gEDhBwL73H5/in5D4  
UamSa3hB5ON7oD/BiZ89D7aJkwYIVtQzzIsTGXsemJi22YBRCfZ0rNOvazgx9i8s+YF+F6lc60a9  
xXfBOuyTFewj5CU4cgVt2r8dzTfLEPhOyDvWhZMOflde0I/oh3xHBxhhWNKpotJQcKMVZhj2L84  
IfTbFdow+nn/5Bb72F7GNIZxUoETF7RPe6Q4AhdT0Heifjz8sGsG4GKLvR60Bd8YDd4aju+D74J6  
hfqFEbXos7Fe+/iOk0Wc4PmmHC7qoU77xw6YJDj+oy+Cifbss+5JkX/hD33a1Ve75mOUeV4U4MQu  
aPyE9a0YSYNbgu358LnfNv1jHHIfPMZh9JB/jMNFHLQ1ez0I7BufoLHqH+OQe1xwQNuDRsG67OMW  
TsBh9vkmGvoLGAH47jg22v0OAmY49gP0U5xjHNq1vTwCJlAcwGw9jLJCzlDv/PqBvOPOGWgxgPaE  
HNrL2cYqLqbh+IgLWvY80FwwQpAr1GW/3dvzQCPgmAnQN9v9NwL9KnQeDFa0TfSfODb4xgACbR51  
2r/YjpzffXe+3sK8aA84FuFY8tJLbnvGvPXgTYEcyxoZCdBHGO1KH4D2gP6ltQt6Pxx7qFbkVnc  
bMU6oTdwTLXnQb8CExL7CJrV1+rZCK4DhmHcvAaNVbRTXKxG/flv+ElgH6+84i1kwPHQXs42VvGd  
YdrjlovdVIHfoQ+QR/TJaEf+nWf+PNDq6AtgLgK0E38aAscPHD9xruEf+9FHoS347QmBebA/fXC8  
aNbM7dOxPWgXaBic3+B3oW/H8d/Xz+hX8D0zmXW2sVoco59t0Ff420ZA89i6H4E+Eb85qDXsOxfQ  
Z+JcEsanf1xGv4vzSFwwhO7F/Diu+hfw/MAFH9EJPZ70FjFsRUXcLE86hj6EVzos/U1vje2hbrm  
g4vUOEfE3QTBAT/QN+gH0SfjAny2R1VsCtpjL3KENU3rFtQZ9Jv2hUfkFxcG/eMQ2gN0uH3sgaZD  
v2LrW9QpaAX0SwD6EfXdfqQJ2iDy4PcRuFCMvgdtA9NxThR3oAshRYSpeYQUMTjYQLza5gtMAxwM  
ORHjKhvMNxz0cDUaAguf4eQy2LHDsPHXgbDFCwxYnDyg84UwwO3GMMQgxtCZY4QCTLXgrXAQYcGR  
OgAHA5wg2vPCRAFvwoGe4NjT4gQO9hCkPtkOXHEPajBT/PXjwG9f8cSJOAwyGlwPXfyArGF0S3B



UVBxKlyxiiv8+K62WY16ASPYFsu4Eo2ruvZBFYG64hM0VnFwtkea+SAXMFvteSE4w0ZE4YBsnwwj  
IITjUhT7MhthxioMQpys+7dPYn+jnuMkz54vTqCuo935wISC6eK3W5xUwICwcwch9/HHBZ+xurkY  
qziRRpvzfWNEndpu2EkW+inbJIPRApMRxl+YmZDpOWLoY+z5IML9kxEYABCltmGDW5xgBAZHhKD/  
QR8J4wMCHPsZYhInqjAUbSMIBinqr932AfYP+kT72Vtof/iOINifQqQGzRWYMFy8ONENPnYDxow9  
D8IedaY1VmFChY2QwUmfPR/C7k8BTMHgxRmMUMP+tfOD34n6Ya8L7UtjrBa2zYcZqzgxBviuMND8  
ExDUYYwgkKJU9RexjZWkTf8XrufhkmNdhrs97AN1E1/JCcCJhFOuGG+B59hjBPwTM/c9E9e/ECd  
Q1+OkaEwt4InunEC3ytYp5MgzFjFcQIXXTDK1gb7xJ7PNIzXyO6LyPYxDuvBdFtjoP+BmWwb/wjb  
+Agaq8ifv49tUM+Co1FxATpshB7MHfuCDcLvC+JQ2PoeZujBWPXBfrCNBpj4aPPIn23cl2xjFbnE  
sds3O1CHYVz4ZpQN7qBAX+qbVzAzYB7i+IoLYEENkCk/eN6hPR8C3x/72b6wqQn8dujepAnbD7YZ  
Bk1dmN8QdmEJ/RKOfda8ME/8CwA+/ohMP7Bv0OfIaoOxqmjQWMV/SiOoQDHBxzLUL8wDX0GRs9C  
f2FEor2cfW6Cv7hIbXumaLOo175R5IP6g5Gq9kUC9O8wjdAX4mKj/zkC2itskAT6b/t9CgjoBIA7  
ZfA77HOyOIFjIX3xxwaGuq8DcaEe53HYb2F3zxU1QWMV5x24UG0/xgRAA/i3jvthG6vQv6i/9rEA  
ZjL6DB/UY+g1DDCW14P1ZjNWke8wsE374jcCpnYYGDVvz4ffGfci7qYCrYcLKnbdTQO/Gxc+/OO0  
b9jffCKobRDox3EB1c8x9A36b/9ivw/aFAxYfwR2cD3Qd+gb7AsV+J4Y4RtsU/b5YqYRyoQkhKl1  
hCQABA1GH0C84uCAB1IHb22AKEZHjBMliG2MFAyeVGUzVnHSiCvfuKUFwgfrw0kPTjJgDuAgi4MQ  
TAI7HTAu4hqrviA6LRHaUEk4aoiTsjwgMBBAf/GX/w/bn32o6ivKtoH+KDZg21jplb/XZFfGI0w  
QuJewbcpjLGKEz0ccO3puDlcNglBRnhwZE02YxXzBs0egPoTNFZhuiRhrCZNMlHqmA+EIAYdWAb  
aRD+ME+CdRPz2vUSYe+L4K1tMMP8EUQ2MOiDInJzMVYhru3vhRFMvvgNAyLanh/9B9o1TCn7cwQM  
7DCCxiqMDN9YRc6DozsxYkVjHuEkyh45g7CfkxgE+xx13p4f2wTBUTYw4DC/DUYW2vNAGONiiA1u  
3bLnQRTGWM0kfuMYq+hvguYR+qUw0H7s+dC+tCNWC0OYsYrRGzbY37gA5c+HEUz2yQTCNIZxUGGT  
xJ4OA8O+dTII+jl7fpwswXjGKHb7cXXMXorjKCxipNrjLZBX44Rfv7JEb4/Rs5j/6M9ZDpW4t9x  
b98tLGHGaqaRLcELDbaxiuMKtI09HRdgw475OF5hWXvebMYq9jlyGQTaKmiswoBPwlgTLFHGKSaF  
cnwn34jHX9SZ4Agx21iFUWqbRBhliD47EziW+uaYH9BE0JBDYDRf2EEjVVSH6Mq0c4w+to2cPD/  
OO74x92wYzDqO4y7sAtKRU2UsYo+OfgbcBGqML+hOlzVoiDMWIXutUF/Ac3jn7PgAkDQ/LHPTaCb  
giY1Ln5IAv1IULditCL6dv/RNH5gH4U9hgH712isQgsC9L/YF36bgV7EwATsT+zDbH1ycH/5oD/H  
8Rn9nW/CoS+ENki6TgeNVRx78F2DRBmr0HTB4wB+UxjBkcNYbzZjFXo6jDBj1X4kjM1/YazirhH8  
1uBFQD/wO3FuFtSwOKdGP+P3FegXcByEfsCFc1/DQE9meykj1gtzHvUP68H6sG4YqDB07Rf+4tgc  
Bu5q9ecp7ufOkpzH1DpCihB0+hiuD/MPIgoHdogS+0QDHS5ELUwPv/NDYFRI8ICM0SD2PLi66IsK  
HNIdb1XHFVp0xgAdM0xG/7Z5P2CsBq9wgbBREv4JOg6g9rNdYHTgZMAebYADAA7KOOHE8w/xFx18

8PlzEDA4QGC0HEQ8/mLkblxwVc8/6YbhiHyGXZnHSZU/WhemdhV7igKY6zihAknQ/aIFFzhxshl  
+3l3vjkevEJaHMYqBLI9L25H1YCDPW7PxHIQwDAssokGDWHGavDKNgQH2hyMM38eCB+cFNk5xkge  
mCatWrl1E+0So539N4wCmE64LckX3zABcDKA0Q4+qA9ok0ETXGOswoDDbVr4jviLNrlppn8Y6Fdw  
e6L/vfyLNjixsPsgnLjhRNK+VREn2DjJQ7+B3Puf+7EpxirMDtwKbPdTuKUQ5qXdL2Af4hYq/2QN  
V+0xShF9HL4P6ph9Kyr6NlxkQR32Qd3H98Y+tvsxXGjCCAEQNDxLgrEKox8nMbZBgnqM+o2+EUCo  
w8CEYWmvC+1La6xifrvvhjnm/7Yo4hirAMdBj7EcSZoMCFsYxX7Hf2CPeIJFwhhbuK50PYJEI5X  
OH7YJ5v4N07g0D7w6Br/c4TGWMXJOuoq+iTUN/9ECscP/7Y+//iOvziBQ3+J9ooLbqj/qL/+sQxt  
FMcXv29FfQ+7S2FTCDNWw/YDyGaslmeYbp9Uo24jn34fADC6C+0yaO4Vh7EaNA7R52ol2w/2MSEb  
cYxVH6wTx+qwRyYgbGMVlwpRv/2Tf7QpXMSF3kMd9EHfhn4U+fHNDLQdHC/Rd6LvKlyx6us21Eu7  
b8FjPXDB0+4XsC20aehc1HmMeIshAt0lcJET5r5/bLziCrcdhWk7LVHGKvRRpt9gjwLO9Bt8zW2D  
Nh7HWA0aVjBW/ZduxQUj7ZEvX1NgII9vM9GHGMVlleo/0HN6ldtrOK4A81q6wtoXxz8FgxGzzj  
HHXZHngCo9B/jrF9Vw0ik7GKuh40Vv0Lqtg/0CH+57hwgWom/TtxjMSFAuxTaES0U/RtYX0QsDUA  
6oLd7gAGytg6D+2wqEazhhmrwe2DKGMVfQ6OPfaFS38gin/cRK7vuMN9vIS9nulwVqFR7flwfmXf  
Zp8NnJvgXMs+N/EfXVJU4G4w6Amc7yOPO06hj/DBsQN13j/uQdeiLgQfj4e+GLoT5+943jDudLQv  
IOGCLfpfXzfiGIHtQd/YYJ/h/Nq/clw7SbBeQooRU/MIKWJg4OF5jhAg9u2ZwcDBBR0kbp3FQR/L  
BUUKTjbQGQdPSGAWQLigA4UYD56sliC6YepBsODfvkGI26axraCji4MAhCBMI3s0BAJXvXDSCbED  
MwLbCzvhReAgDYGdg3GYuIOhEhRn+B1xgdnmg4cwiWE0hI3Axe9B7jAftrepxmow9zDOFD42NNw  
Oukbqz64IRTPLENew3IGwwi3JAfFhm2sIpf2NBwwMxmrQTGPecthxpEEQ7O9u3vfkDoZnpBhQ3m  
CdY9jKYpCnDSFvxewRGrPti3aCsw3JDH4HJ+4LfCdMJzYG3DxQejNXHSjufOhbUpmGwQmFgP2q7f  
pnBLTlyCi2PxnF23ABcFELkQIBDb+H6ZbmICXUTdxok1TkDwKBefPN8qOH/Q0PMJnvghfbapAiDg  
sQ2cNAYNHT+wz1Cv8agS9DPBiz+oa6iXuFiC752p/8Hvxe9GP4WTGhiZPsETEjwDzD+x9wleLMG+  
DhqrwZGNCHukh32rOQInHf4JOr6P3y/5gYtWYeARL/Z8CJzQhYETN/Ql9m3ZdqCPsc1pBOq41lhF  
Tu11wMREu4kDjMNg28z2DDDUZZgYfjvzA8fV4G3iqHMw+WHio21mqmfoH7A8TILQFnFi7wMz2p4X  
+clkrOIZqva8qEu+wYITaNwaihMufPewvgSBugrzGG0jeNzAc9eCvwGmTIGAfR5sP5n2Aww9ez4c  
x3xj1Qd9NYwotJWwdolnPP+BfMADEmdK9WeBs2SyVgNvnwRplmYsYoTWNsfsGMcjPHgiWkYYfsB  
9SQOMKOQF3vZqBfIdCfBu5sQ/kVuH2ge3P4PQxWaK1M/j0B/iRxfFCCOob72Q7uFyWTPC4M/DJic  
9nwIGAs+uJ0bpiz6/6CW8QP7ANuDBg32Y9AoMIDt+aHzwo7TWraF/PcD+IF6EUT7G3AcyARyBf/k  
RECrB41V9NswHYPnCTAZMD3DTNsg0HH2stA4wRcqZiL4YkDUITBj1Qc5wvqD9Q3HlqCOQR+Ni874  
fVhvyppiyfaHPhhELI8zW6jj3sOfFujlZq/bzJxEYsOADkx+P3IJ5ju+e6bugv0buYZiFPVrDJ+rI

VcHnudvHh8Liv6zMD2idTMYq8mXPi7uabNA2YMrjgjj2gT0vAvlAHxZ8FrNtrGK/B7UFBiCEgfMh  
ez5EpkcB4HMcO4LfC9vG+ZFv5IcBkzy4j3HBoShBW8agBZxno/1m0qQ4L0SbsV9IbAOti4uAvl5A  
WwlbD/KAPKNuho1QRh+KbeCcD98F68v0KCNCESLUVEISAp0uDqS4LQYn07iChlF0+IurWjgY4cpu  
2K3hPhCV6CwxL9aB5XECif/HiQROMjAyEglAo+8wHYYiTihwwMTyOMnzb0/wb2WC6AoaF1gXPkOH  
jQMmvie2hfVhGd+8xF8cxPG7MM3/Tdg+DnTYbtC0tcHy+C0wWLF+/NUKDIz19EelQSgFBR0IGqvB  
kbNxd7E78OyGEWAv/4oMIC82L8D2/TNZOQTuURO8P0wHQFRiN+PfPkjOnB1MmiYlSED9aDZf3v  
gHxjv4eBbeEgje+Ev9hPQcMe+Psb+xl7G9/d39/YH3FGPKBu2vIBZLSFV8OmrBvfGblCjoN1E7f6  
+m0iCrQZ7De/zWEdWCeW93Pm3zqGv2En9ZnAb4ApDoMeghgnA9kE4qaC74nvi9+N/ga/wc4H+gjk  
Kmw/4/egLvH5x78znajh92Mf+W0AOcvUp/I9ot8G7O8Tp+8AqPf43vj+/joQ2N+ox/i9Yf0bwDS7  
bWD+4Pb8No3fjb92m/bB+v3c+Pmx6wD6Y3s7yL+/HeQGv9fOGXIYRpz9gN+J5ZE/tHWYJcgHcozc  
4jfiNwGcxNt9DAQ8Ri9qCOvzwi5shaFt0+ijsG70k/78WBa/K2hQ+CC/WCfyEaz3/nEX7TfsO6Nv  
t7eD5TJtJ05dww9jH2Je7HP/uyBn2FflZaZ+Ft8P3xfmOk5SYYz5o7gKi2Y/BI89dnvA/kEukWv/  
GldchB3jYFDal6MQuHPFB+uxt4NcxT3GYXuZ+vVsx7hM67cJagCErQGyge+E72Z/16i+Ht8X+wK/  
386H34aDYBuoY8iz/xvxfRHYB6hn6B/wO4I6QKPF0Df7vx+B+cNG4CGnWlF9Xfy2h+9o95M2+G5Y  
Dhc6MFIQF+FgWMU5Xkeh3Q+b+huCxNFi+G5hGh/HN/T1cX6/pj0EQV226xm2n6nPA/j+yA++n11v  
kJuwYy5Af4E2iGWC/SCWw77wtVUQ5NveDpYPmw99LfaXnYewfey3r6B+wL/xWdQ5mQ8uNPv9WJhJ  
j/XgwhDuSMLFPpiTmdqWFvwu+3fid4ftb/zWoNbw+2OA/Kit4h+fMHISecD6sl/8Ooj9F3yMEgYE  
YD6A7WA/+tvBd8pkSqPP8L+L35dg34eB34S6iN9rtw20I3yv4PHWBn0bfhMGCEAOllwQTOpFTvge  
2N/4XsE6hbaJ+hZs92FgHvQFYXoBbRr1J0y3+CCXOKfA/qldu+CLTgkpBkztI4RsceBAjNFPuAql  
gwiebeS/6dYHB3r/1iAYCBihggMrxA4OXkkDMYJbXOXnTGIUDkbN4YUgMNZwmwZGPvi39CEgAHB7  
GL4nKXng0Q8YxYHRObitFLd1ZjJXCikCjZAY5WY/XgCmBEZjwrhCP4M7AmDg26OMMOoDoxEh4Mnm  
B/YLjg+4KwMj0jDCDye6mxM4ycNzzu2RhrjVHLf9o26h7uEvRIxAFw5hsOKW1KK+NZNsucDMwl1R  
GI2Hx3qg78Jtv3EMCUKKA1ygwHkG7o6yHxuBxwcEwaNfcEs3+kaMgsZdO9lMsf8CPBoHo1XtO0Lw  
SBvcvt63r/sMfowYxqhXezQmzmlwkS+TIbo5AFMTdxTi+AnjG4+y2tyOn0UJ7pDx36eCOpfpbbWE  
JlypgYSQLRbcpokrQThRw+gGPMvNH4WCq4i4ggjTABdbQjzgdjDcQolrgsUFTDPc2oTHPeA2Z5ge  
GH2EE038xW1HOPDjBBTP+drcxBchZPMHo6jQf+BECSc+uJ0VF2lw0QajF3CRCbcn4tZPjBaJO6KJ  
FD9bpmpeYxZGKeFYixNZmPxhxzgYxZujwUAIIVHgESh43Az6N1xAwiNIYGBIG+G7JYC7EnDuBCMY  
F2ChF3BOhf4bj6qDYQfzGBc+so303lzlPyx0H24q+XEE91jLR71kO1RHoQkDI1VQkC2AxFHDlgw  
D8kTIWPuA0IIIIYQQgghZLOBxirJDTByE7dC4GH7CNxWGLwFFG/r9G/18CPT26pLGhjBimcm4WUJ

9nPg8EByvJkSo3FI8YCRdPZtVghcRS+OxzcQQgghhBBCCCEkNuaMnZAcAM/7sY0qxPPpexOJk5/g  
W07xohc8aD3Og+xJ0QFjFQa3vS/wNI0aq4QQQgghhBBCyGaFOWMnJCFgBH3/vftWSDz3pHvR9yUU  
MI3OOst9Zg0e0I833Ec9ZBoPGR8yxH0JCZ6Dc/bZ7kuQunZ13x6PB6oHDUC8pXDMGJGrnLfSmkb  
VYg993SXh8GKN3birYMnneR+jmeR4u/jj3srM9i3YWNbeN7QoEEi113nvigFL6zA98L3u/pqkYED  
3RcwBV/MgweI4y3HeBbbqae6f/Ed8GxUjKrFM4yQqzPPdP9efrn7ggw8p9TH/i54XiDmwwwusTjIF  
5IQTXKM0CjwraMQIkUsuCc8PRkni+a2PPuq+idIH+wovHHroITe3eOMkfjd+P/Jw440izz3n7rMg  
eFnH6aeLNG/u5hrfF8+cw1sj8exD5A45wfrwzFjG3UGb7a0wTORKCe8+ArPrsN3wEhkH9Q97J/r  
r3ef64o6h7qG7dv1BG/1HD7cfYkSto1tYI58x3bt3PqGepcp9z/95H5X5Pzkk93fhboIQxp5w37B  
d8N3sZ+rh9+Lt5Hfcou7HbywAnXx5ZfdFwRgXfa+oLFKCCGEEEEIIIRsdpgzdkKKGJhGeBv8gQfm  
G0N4GDjeYA8jDLc577OPSKVK+dN33NE1mfwXisC8mjdPpGdPke22y5+venX3rY1HHy1y+OHug8X9  
abiF/eab841MGKsff+waVngDtD+fH/gMb8ofPNj9zjAo8T3tebA+H3wnGIww4PBSFH8efCeYkPhd  
CHw/ezq+V/v2lr//7q4H68CLVfzpCOQCnyEveFA63gq7++7pt+VjnTDpYObZwJgMjja9+25vYhZg  
8OL2f/yeXXdNXx6x226uMQhjHMYiTHI8INzeFvKF/Yz9AfMPBi0eH+BPx2/CowTmz3e3iQfE+9P8  
qFdPpH79/P/H/oahDcPS/+y880TGj3f3AQzhJ59Mfws4Agasv50pU9y3LmPdmlY3eultnj54yQ3e  
9ukvW66ca6Rj/6GO4qH8dt3Cb4J5CoPUBiZqhQr58yHwm+3vtu22lu++6xqj2PeoC3b9wPx4iyX2  
e+PG7ksB7DeQImisEkIIIIYQQQgghmx3mjJ2QlgYjGjECs29fkceE3nzzfTnmU6d6o7mDJp5GL2I  
N9kDmGjNmrlvdPanwwjDKEh75B/+je1ss437RkCMTuzfv+Bb7197LX1biOCjACZMcE0wex7bWMWb  
B/feO99Ilg8F4ww0i48alj2LEvydNct8A7JuQWGa//dy3+M+cmW/4+YHRuxipGHxTNQxEe75SpUR  
f92b6IFIMNoVb+b0Y906b2JMMCLU3g4Clz9kGO8zb9MGXcaze/kedo0bwYPmNpffSXSqVP+evBm  
TexL1IHRo/M/9wMGmt7kDdMSxih+C8AbIv387byzO3oUb4D89FPXyMU0mJ8YVYq5srueRx5xR4vi  
rZCoD1i2Th2RSy9NN0W/+ELkwQfdQD3Ad/Zz5o8mhZlrf0+Yrx9+6M7jE2as4rtelLHlt9+6Fwd8  
QxQjamGc+78JBipGW6M9+OBiWvVuSOm7XXSWCWEEEEIIIIYQQQjY7zBk7IUUMzE4Yi/fd5xpCMJpg  
WGJUIm7/xq3zMM3wuW0e2cYqbjXfeuv06bhVXINTdsZ5xmqUsQqT2J6GUaYyEZsJmKgw/OxlyK7N  
nl3QWMUt62HgJVv2fAiYxEVNILGK0Z/2NjiYuI0+E8FcYVQvzHIYnvbniGef9RYKgJeJoa74o3Z3  
2sn9DGYo/h+pJGiGaNTMRoan+Hv00+7dQ8mND7DSNDgi8qwD954wx2R26SJSN26+aNvMSOYhidu  
z8fyfsQ1VvEIgDDuvTd9PvwuPCoiCB5VgLZgz0tjIRBCCCCGEEEEII2ewwZ+yEFDHTp7sjO21jCMac  
D4xXmi4YvWfPAzPJH7GJkakYiWjfCt+2rWvM+QYTjFOMTMWt5jA5y5Z1DagnnhCZM8edxyfMWMVz  
QG2ijNWRI931+48wgNGGkYkYQRkcRTtqlPt8UsyDeXErOW73hmGMW/CDxioMxDBgxNrziYLGKvIB  
IxxGJR4TgL8Y/aohyljFbfTYX/4t6sjTbbe5o3VtMNoTzwnFs1H99WAFyI/iGbbY7/7nfgQNbhvc

Ou/XJeQSxiceG4FRn8gvRoXCjMdZyJyFvPBKN1jD/ffuKUfRmfw+bswZP3tl7A/kDefP/5wn5Nr  
z4PtY7/ahBmr9iMHbGCS47f4l1ZhumNUs/0sWjzmAM/TDT7zlsYqIYQQQgghhBCy2WHO2AkpYmB2  
4uVMtkkJMwkGG0ahljCC0TaOELi9G8aSD24Jx8hW+/mbME9xizdGF+L2f/t5njCjMpla779f8Pmp  
uK0dRtedd7ojGGHiZXvGKoBheeWV6cao/Z0QGE mJz/zpeN4mbmP3zU6MsAw+Y7UwxipuY7cfmYDA  
iE0NUcYqmDjRfWGY/XxQmlrYn/jd2N/Yr/5IUQTMWLzkyjee8Vglf5of2YxVGLV4HAJGndrLwETF  
PgUwVrFe+5msfvimqj16GeAZwPhu/nfFX+QQ+xG/B399A9QOvFzMRmOsAhjgeI6rPT8MvTlWrxC  
7Ya1DTxPmMYqIYQQQgghhBCyWWHO2AlJCNx+jdutcVs4boPGS5Awag8j9GCGwWDEW/sxChDPtMS/  
MQ3PaA2CF0thxChMNqwHb3jHcz/xkiN8DIM0G9ge3iSPW7lhfoFW7zvucP+NN8XDxMXzPfG4AZip  
t9/u/sUzQcPA8zixTbyUCCYdlsN3wu/Ecz7xJnlsC7fLB009bAdmo70dGHRhIB+9e7sjGxH4zsiF  
DUbA4vf06uXOI5eA4YVWGVByKqzb3w6+F0xAH/s34Bmm+F541itGB/fr574sC98B+xpmMEbmwqwO  
gn0OlxvbwIhXfF+MFM4Eto vntiKX+F1YBjnGaGP/RVWYB6YjnpuK34D5YCxjH2R7XAGeRySROBgh  
it+AZfB7sK2IS936AtMaj3tA/cQ+wgjZv//2VmDA6Gzsd/+74Xd99503MQt4zi7qLeoflve3jd+A  
34KRx9imv208AsF/TAYhhBBCCCGEEEI2C2isEkIIYQQQgghhBBCiBlaq4QQQgghhBBCCCGEEKKE  
xiohJZHg4wclIYQQQgghhBBCSJFCY5WQLZnzzjOt2DRjP3bbTWTOHG/iZgAMXry4atky97mleJZt  
YcGzX/GM2hNPFKITR6R0afe344VW++0ncs017nN78WxWQgghhBBCCCGEKISgsUrllkz79unG6l57  
icyd6038j8ELmLp1E6IXL//7nXKKN1EJzFm8kOuww/LXVamSyNlni7z2mvuCJ7wEbLvt8qfjTft4  
mdaiRRzBSwghhBBCCCGEKln5f0lpOjA6MR33hF56SWR//3P/Yu3yMPcwmhDvA391VfdN6pv2OAu  
s3at+3Z7vIX9/fdF3njDNczeftt9M//EiSJLlrjz+mD049ix7t vpsZ2XX3bNPBusd8wYkaFDRV55  
ReStt0TGjcs32vB2fxiReIs9vifWgTfF//WXO80Hb8L/7Tf3u+D7Y11YJ+Z/8033jfHjx2c2NWHu  
YT4/J8OGuW/lx+/Hcvhe+M1447y9Xfwbb7bH2+lxD3KC3E6e7P6288/PNxIRYcYq3viP7fnbxfDY  
uNCbWATgO/7xh8iNN4pUrpz+fYJxxhneQkqwJ48+On/9VauKPPWU++Z+H78+HH54/vYaNXLf1M/R  
q4QQQgghhBBCCCLiUt5fQoQqQVNF6tfPN7cQu+wicsIJjVquP9foYLIiy+6BuHgwa4haM8fFhUr  
ipx8srt+ALNzyBCRY47Jn+ess1wDFMCIhYm4zz7p6znuOJHFi915Zs4UueMOkSZN8qfDIFy50p0O  
s/fxx0Xq1k1fR6aoXVvkgtEfVzRXd7nhx9Ett46fd499nC/S5Uq7v/XrOman+vXu0Zgv34iBxyQ  
vowfyB9ug8eotfvzMGp1rrvS58F2vv/em1iEwGCFuenf7j9vnsihh6Zve1ON1fvvz7/IH7HVVq4Z  
HgSPG2jePH2bp56avz8JIYQQQgghhBBCioiU95eQoiPMWMIQ5iEt90m8sILisOHi7z3nshVV6XP  
s/feli1auCMx27UTO fNM15zzDcSyZV0TFOvw6dMnfx14xihuCcelTlySbd3aNeGwPJ6/Wa6ca/LC  
zF2xwh3N2bixuywMztNOE/nwQ2/FBvz7ootEOnYUufpqkWeecUfk+mDUKZ7p6W/fD/w+mzBjtXp1  
99b2u+8WefZZdzTq/Pnu6Ff8bjwzFPOVKePmAyN5/ZG2s2aJPPqoyL77pq8zzFiFeY3vg1G2+luR

rxhBGwXM1wcfFDn3XNcQxW33CPy7bVuR228X+e47b+YQkCd79ChiU41V5MheD/Zp2LZhSuNxA/a8  
MN6xrwkhhBBCCCGEEEEKKkJT3I5CiI8xYvfRSb6IHRjfCpMSITX8ejGY95BCRpk0LBm4DR+DfRx3I  
3ILv8803It27uyMxsZ6ddhKZMME1KsuXdz/DS55gxDVR5v7/sce664Ch6Zu2uG0cjwGwbxvHrfq4  
PR/f9cILRQ46yB2VCIUMUI0ZhbMKoxfJ2xDFWcYt6GIMGpc+H3wATOgyYz/a8RfmMVRiwmFLt9fsB  
sxeGJW7RzwRGjxaVsYqXVWHEsr8eGKt4xEMQjEQOjliFSc0Rq4QQQgghhBBCCCLiUt5fQoqOMGP1  
yiu9iR4wVmEWtmmTP8+BB7qfBZ+ICqMTxuQnn4h8/LH7vFHcwm/z++/uaFesB6NaTzrJHelaqpQ7  
AhbPagX4Hr7ZitGiMFgxT8OGltfdX3DbAwa4hi3mwTlw9PDGeYDRkXipEI6Q5P8GP+IYq3fe6U0M  
gMcBNGjgmpcYD6NsMXLUfts/jMKvvxY5/vj0dWqM1agXOvmPJMAJF4KBEaC45R/zZCKusRrnXVKo  
F6ef7o5qxnowurh/f9f09oGpilG9ML/97eHfyB32EyGEEElIIYQQQkgRkvL+EIJ0xDFWfTDqECMM  
YR5iPpiJMDw7dBdp0sU103bdNX89eC4pjEc8v9MGRt8DD4gcfHD+vP768JljGLkAo1hbtsw3Lf24  
7jrXrA0ahVjWfqs9loMBfM457mhRjFj1n5Fqx9NPeyvw0BirulUeJnLwt+AxB7gFv1Mn11CFyWtP  
R+y+e7oBC265JX0eGM/ffutNTBDNiFW88d83r/3AowZs8NgFGPEYLezPAYPZrysYjeyPPsa6UI8w  
6jaOcUsIIYQQQgghhBCiJOX9JaTowBv1cYs/DEk8vxR/773Xm2hhG14zZrgvs8Lt9rjFfP/93Wey  
YkRpt24izz/vvkgqE1gXDEmMMIURu/POrukJAXTPWvXBs1fxNn+80ArmL567ihde4W3/mcDzSJ98  
0jVSYRTie+ElWjAsMVIW3x1GKI42BbP0nntEnnvOfUapb9ROmuQui23C/MSIVLwUK4qJE0V693Zf  
wlScILBtmMh4ziqMQ7ycqm9f9yVceGRB8HmiTzzhbg/bxfZhOOL7JA1GtsII3X5797m4+HvZZd7E  
AD/95BrJ/vfECGI8B9fHrivYhyNHivToIXLiia7Rjf2IR0TAZMULzfwXnAEaq4QQQgghhBBCCEka  
GquEEEEIIYQQQgghhBCihMYqIYQQQgghhBBCCCGEKKGxSgghhBBCCCGEEEEIIUporBJCCCGEEEEII  
IYQQQogSGquEEEEIIYQQQgghhBCihMYqIYQQQgghhBBCCCGEKKGxSgghhBBCCCGEEEEIIUporBJC  
CCGEEEEIIYQQQogSGquEEEEIIYQQQgghhBCihMYqIYQQQgghhBBCCCGEKKGxSgghhBBCCCGEEEEII  
IUporJJC08OUIClVTaLufXy2uyITTCGEEEEIIYQQkuNs3Oj9I8CwYSIVKoiULy9SvXrBSKVETjnF  
mzlApnUSEKJHU+BVVDMI6F8cYMosUwjRQGOVFJrupqATCit1TZIsCiGEEEEIIYSQHMM3PdetE7nj  
DpHLLxfp2NE1Su3YbS9Z07C/9B/YWLR0T8kVgejyTGnpf1FK1gaX228/keuuE+nWTeSzz9xtAZqt  
JAOTbE9C7s0MmWmKYRoMD0RIYWDxiohhBBCCCGEKAL89ZfllVeKVK6cb4YefbTlvHneDPmsNkWp  
HGwdTVplY8pMO9bMEeChh0Rq1Mhf9/77i3zwgcqVd4MhKRDY5UUNTRWSaG50hQc6AoU89kOpvxj  
Ctjlq4YkJsG68uSTT0rNmjWlbt26suOOO0q9evWkatWqcuONN3pz5MN6RuIQRcdvv/221KITR2rX  
ru3UMUSVKIWkU6dO3hz5sl6ROATryXfffSc77bSTbL311nl1DP3YKaecluvXr/fmcmEdl3EI1pM/  
//xT9t13X9lqq63y6lj16tXI0EMPIblz53pzubCOkTgE68mKFSvk2GOPIWrVquXVMeiz3XffXSZO  
nOjN5cl6lqPY+717d5E99hD5x5wLXn+9SMOGII9+6U10CasnzUzJdG55rCk+BZZdvNgdtbrzziK/

/SZy771iTh5Epk3zZiC5SFgdO9+UTHVsD1OWmgLYj5G40FglWbE7kzfffFN69+4t9913nyOgUinT  
+SAeSkkpU7zuKK84n/1t/IUjJfvV30/69esnt912m4waNcpblzsrkl4HRO8e7dSR+++Xw455JD8  
OmaiVCITn7L8PwyLe42Auv322+XVV1/11sg6RtLrwI8//uj0Yw888lCccMIJaXUIYderYB2rVauW  
Uz/vvPNOefbZZ701so6R9DowzZzAoY6gH2vVqIVaHUJkq2PlypUz557Xyz333COPPfaYrMNTkwbW  
MWLXAZik0GJ9+/aVLI26pNUhRLY6hujevbn06dPH0WWLFi3y1sp6RvLrwKpVq6R///5OPbvqqqsK  
1KOoOtauXTunD7z77rvIL4xW9GAdywE++kjkvPPckaPHHSemsxHTUeXdom/XgMGDBzt1pFevXo5Z  
n1eHPjb1KtO55QfmX2aeFke3cLQcdL9t6jvrx8XKW28VufZakZtvFINJRerVE1OpRWbPduYjJRe7  
nxk+fLjcccdctdddzkDc/Lq2AtZ6tjP5I9lUnLknkc6dQza/0vrggD7MRIGjVUSCQ56ficUJp5S  
A01kKrNMNDbhZwsvP2LECG8LJnfBaEG/boTWMUXYy8OYIAR8/fXXzqiaYB3ZILCX79mzp2zYsMHb  
Csllfv/9d9lItt93S6sqmhl3HOnbsKCTxFrjkeQgc+bMkaOOOiq0nmjDXrZ58+ayYMECbyskl1m2  
bJmce+65ofWkMIGR1BhRTUo4CxeKnHWWa6pWqQJXy5uQz9q1a+XKK6/MqxsF6lglE1+kpLw5j6w9  
JyXb/ZuSrRabz/zyuQnMYy9jAne1jR8/3tuKxerVlscem/+dRo70JpCSDC4e+nUjtB8bbiJT+dXE  
9ia8ef3lK1SolB/hwgeHldBYJaG89957sueee0rp0qXzO6BMoTBWg1GmTBk58MAD064Ckdzg22+/  
dUalli1bNrRuFFXgYNi4cWN54403vC2TXAEjBzEqFUlorG4UZTRo0ECeeuopb8skV5g/f74zKhWP  
jQirFOUZeFQFRrKS3GLNmjXOCFPc3h9WL4oycPGpR48evFiUg9x6662y7bbbhtaLogw8/qRDhw6y  
dKI7myOpQXz4oTsy1OxnOfxw19C0ePTRR9NHDJooZf0bcaKJSaVTMsGcO/65k1mP+X/EisopGb9v  
SqYdkplndzXzmnns5ez1VKxYUVq0aCF///23t2WPQYPy1ueMqCUIDoxO3XXXXfPqQtZQGKvBwN1F  
//d//ycTJkzwtkyl6Vq8v4SkDWvHaNKwjiQ0CmGslmCufvLJJ96WSa4w9oOxUiGVvOHLxwsDX/C2  
nAXe2bHlKWWfzZgwQ+pUqRNaH5Kle3ve6205C6xjWx5Z9tmyf5fjXg32Cq0PScQl7S/xtkxKFNN6  
hdUiJxxW8LEISUXL41qKxPFV2ZdtWUTsrwwOviC0PiQRhZQ5RNYuXOttOQusY1sO330nGy++WKR0  
aZGDDsLLEbwJ+dx8882h9eEGE+YfabEx5LOWz383sbWJ4DobNmwo//77r7dlj59+EuncWaRSJdm4  
554izz+PidreRFiSeOKJJwrUhYxRCGMVgUdX/IQ6RYiH6ZUIyQdXkDGCNKwDyRiFNfYZJTC6mthS  
y1wTe5gl+12MzSdOMRH20Pktoaw1cayJsN/F2HxifxP27YdbWmlnlux3MTaf2MHENBNbarnBRNjv  
Ymw+UdHEVya21PKlibDfxdis4lUT/mjV86zP48S1Jsw/Nil+NFHHRNh6w6KCibEm8F3XmL/HeJ8z  
cjAKaaz60bp1a89FIbmO6VklYQfG6mGHHrbacWQMGquMYFxsYkst80zsYyLsdzE2nzjdxJZsrJ5o  
lux3MTafOMTElmysXmAi7HcxNp9oaGK6iS219DQR9rsYm09UN/G1iS21DDAR9rsYm1W8bsL8w4nz  
vc+iop6JxSbM/zgjUTONUsOUG6x/32libBvBqGbiKxPmfxj9XgTYfMxcICKyFjFi/olAaZnISQf  
vCDjxhtvIF122SW08wgNGquMYHQ2ka38V4ZYnO1ixOruJsJ+F2PzieYmsu1Pa1rYWz+LuuRtI04d  
g7F6tlmw38XYfAIXWLIZq/9VHYtb2pgl+12MzSe2MxFzxOpmWceuNxH2uxibT5QzMdZEJfLsdSzO

8bK/ibDfxdisYoQJ8w8nzvVe9BMW/kuA7GeidjZh/rFJ8bWJvHX7f7Nsv6yJL72RtTBWj/l+Z+Rg  
FNJYxaMM99InH+nfv7/nopBcx/RKhBTkhRdeCO1EQqMlnrH61ltvOdttdt24doyTEBhNZCvjg4w+k  
QhXvGatlEwyvvnj35IPu8p/Wm2N8ltKw3Efa7GJtPYB9lKeCnST9J7e1rF1sdu+2O25ztAvu7hBbW  
sc0/YtSxf2b/I3vsvUfydayMCbONzt06O9sF9ncJLeiHw34XY/MJ1LGNJlUsHz1cmI2QrPk6xjC  
bKPFGS1kzYY1zrbt7xJaWMe2jlioY6BN+zZuHUNfE1Y3iirMNg489ECZt2ies137u4QW1rEtIpzH  
4b7/vkj58rLe7ON1hx4q6xYtMnswN2uuucatY4HoYcL8Y5NinImaJoLrrF+/vkyZMsXZ7roNG5zv  
se6xx2SdmeaMjG3b1pm2Hp+H/B7GlhX+CxcfeuihAnUhYxTBM1bHjh3rbJcQYHoWQgqCB34/88wz  
sv322+d1lBmvACqNVXs9u+++uwwbNkzmzXMMFFskdFixYIP/73/+kSZMmoXWjMGGvB2+5HTRokPzz  
zz/elkmusGzZMnnnnXfkqKOOcQ0bhQ1/XeXNiUS/fv1k2rRp3pZJrrB69WoZPXq0nHHGGQXqRVGE  
vS68+GPSPeNelkmugBPGb775Ri688MLQelHYsNfVvXt3+eGHH9JeZkpyA7zd+tprrw2tF4UNE114  
HuHnn38ua9fGeHkV2bJ4991807NpU+/DfH777Te5++678+oD/tojVxF4PED78ik5s29KnrjGW5eJ  
aTubz59PSbv3U3JsLzOvmcderrRVx44//nj54IMPZMWKFd6WPV54lf/7mWM2KXn89ddf8thjjOn1  
6tXz6kPGvkxprNrrOeCAA+T111+XxYsXe1smxHQt3l9C0rBF9fjx4+WQQw6RsmXLpnUweaE0VmFC  
HH300TJjxgvc+nbI7mBvc9nzpzcKEKfBwRrIUMjilGQe/bb7/1tsA6lqv4+33VqlVyzjnnSJUq  
VULrzKZE48aNZeTlkc76AetYbmLv94svvlhq1KgRWl82JR0aCBPP/20t3bWsVzF3u+33nqrc8Ew  
rL5sSmy33XZy7733emtnHctV7P0OY6JevXqh9WVTYuutt5YePXp4a2cdK5HY+7Rjx3wD86KLRLyL  
zvZef+WVV2TXXXdNqyd5USklpT41f0NKKfx3jAkzj70MRg+ef/75eSMX87a1fLnI66+L1Kzpfp+K  
FUV++cWbaGBdLFHYfucYMWOCw/VLly6dVlfyIsPyxcslrfkrmpzyimnyPz58531sx8jQWisktgs  
NwcnXP3DKDB7dA7e2Bn2XCbns7/Nv7ZKSaeWnWTNmjXOsgWulBLigWf84gVqGMIlwySWXpB3Qwq44  
2p+ddNJJzpVDrAN1lZAwMMIQ9QSC6K677kqrT1Gx7777OiofUT9RTwkJA/Vj4cKFzr9x50dYXcoU  
devWIYkTJzonh349JSQlbn1ctGiR8+933323wPEx2/ESFzAxyhpgHevX42ZYQtKx+6DvvvvOMUfD  
6lOmz3A3GkbfyNGpOcgbb7hGJp4/+dtvllttJXL33d5EF9StJUuWOP3Z9OnTpVGjRnn1JzXa1KdM  
55bvm3+ZeR6840FnPejHcl6ZxpW5ljvsgOcP4BY5kTp1RE4+GSLQm4HkCvAeoP1xpyQGduXVsaFZ  
6tgE86+yKbmmzyXOMRL1FAM0CMkGjVVSaK40JfQB9OazHUz5xxTAE0QSI2BdwW3WeQfCQFSqVEm+  
//57b07WMxKPYD0ZPnx4aP3yw38ONGAdl3EI1hPc/prxzg8TAwcO9OZKHSPxCNaTqVOnZh3NipeT  
+rCOKtgE6wkuKu6xh/dc6ZDAqEEf1rEcxd7vo0aJYLTYfVeI3HEHBL3I5ZeLeBd3wEZvIKINM1My  
nVsea4qPU8f87cEOvf12kZ49RR54wDVVL7tMpG9fjNxx5yE5SVhfdL4pmerYHqYsMwWwHyNxbFK  
Ck13U6zuKK3UNWWyKYQUhmwjC2Gsfrpp96chGwaQ4YMCa1ffvijbwjZVN5///2sxuoDOBEkpBDg  
0U21atUKrV8I+3ZsQjYFPLopbWRhIPAMVULysE2pPn1ESrlv5HfCHA+d57Li1uqpU0XwsqI5q2WI



rJMj5RjrbDK9HLnmYFk5baLib/+K/PmnyKxZlueck79exIEHuiNVAY0xEk JrU+x6ZZdGpsw0hRAN  
puchpHDQWCVJQ2OVJA2NVZIOFZJ0tBYJUIDY5UUCpicf//tmqlffSWmw0o3REuVky07NZR/d6ok  
0xqmZEYgpu1a5v7dxnuzvx3XX4/nTojg/R1LlIngblyQzNFZJUWN6IkIKgTk+xjJWebGQFAlaqyRp  
aKySpKGxSpKGxipJGhqrEjINlr0599k3dWxy0u9tpPb705J37vS4/aeKXnppWNInTd7GhyZSuLg  
VZNYxiqrFFFAY5UUmU6du4eKK0TdrerK5EkcsUoKB41VkjQ0VknS0FglSUNjISQNjVWSNHgJbdOD  
m4bWLwSmYR5CCkPrLq1D6xeiUf1GMvMfjglOmiskkLTvXsWY7VuXZk8mcYqKRw0VknS0FglSUNj  
ISQNjVWSNDRWSdI4xmrTLMaqmUZjIRQW9FVh9QuBPg59HSEaaKySQkNjISQNjVWSNDRWSdLQWCVJ  
Q2OVJA2NVZIOFZJcUBjIRQ1NFZJoaGxSpKGxipJGhqrJGlorJKkobFKkobGKkkaGqukOKCXSooa  
Gquk0NBYJUIDY5UkDY1VkjQ0VknS0FglSUNjISQNjVWSHNBYJUUNjVVSaGiskqShsUqShsYqSRoa  
qyRpaKySpKGxSpKGxiopDmiskqKGxiopNDRWSdLQWCVJQ2OVJA2NVZIOFZJ0tBYJUIDY5UUBzRW  
SVFDY5UUGhqrJGlorJKkobFKkobGKkkaGqskawiskqShsUqKaxqrpKihsUoKDY1VkjQ0VknS0Fgl  
SUNjISQNjVWSNDRWSdLQWCXFAY1VUtTQWCWFhsYqSRoaqyRpaKySpKGxSpKGxipJGhqrJGlorJLi  
gMYqKWporJJCQ2OVJA2NVZIOFZJ0tBYJUIDY5UkDY1VkjQ0VklxQGOVFDU0VkmhobFKkobGKkka  
GqskawiskqShsUqShsYqSRoaq6Q4oLFKihoa6TQ0FglSUNjISQNjVWSNDRWSdLQWCVJQ2OVJA2N  
VVlc0FglRQ2NVVJoaKySpKGxSpKGxipJGhqrJGlorJKkobFKkobGKikOaKySoobGKik0NFZJ0tBY  
JUIDY5UkDY1VkjQ0VknS0FglSUNjIRQHNFZJUUNjIRQaGqskawiskqShsUqShsYqSRoaqyRpaKyS  
pKGxSooDGqukqKGxSgoNjVWSNDRWSdLQWCVJQ2OVJA2NVZIOFZJ0tBYJcUBjVVS1NBYJYWGxipJ  
GhqrJGlorJKkobFKkobGKkkaGqskawiskuKAXiopamiskkJDY5UkDY1VkjQ0VknS0FglSUNjISQN  
jVWSNDRWSXFAY5UUNTRWSaGhsUqShsYqSRoaqyRpaKySpKGxSpKGxipJGhqrDigsUqKGhqrPNDQ  
WCVJQ2OVJA2NVZIOFZJ0tBYJUIDY5UkDY1VUhzQWCVFDY1VUmhorJKkobFKkobGKkkaGqskawis  
kqShsUqShsYqKQ5orJKihsYqKTQ0VknS0FglSUNjISQNjVWSNDRWSdLQWCVJQ2OVFAc0VkiRQ2OV  
FBoaqyRpaKySpKGxSpKGxipJGhqrJGlorJKkobFKigMaq6SoobFKCg2NVZIOFZJ0tBYJUIDY5Uk  
DY1VkjQ0VknS0FglxQGNVVLU0FglhYbGKkkaGqskawiskqShsUqShsYqSRoaqyRpaKyS4oDGKilq  
aKySQkNjISQNjVWSNDRWSdLQWCVJQ2OVJA2NVZIOFZJcUBjIRQ1NFZJoaGxSpKGxipJGhqrJGlo  
rJKkobFKkobGKkkaGqukOKCXSooaGquk0NBYJUIDY5UkDY1VkjQ0VknS0FglSUNjISQNjVWSHNBY  
JUUNjVVSaGiskqShsUqShsYqSRoaqyRpaKySpKGxSpKGxiopDmiskqKGxiopNDRWSdLQWCVJQ2OV  
JA2NVZIOFZJ0tBYJUIDY5UUBzRWSVFDY5UUGhqrJGlorJKkobFKkobGKkkaGqskawiskqShsUqK  
AxqrpKihsUoKDY1VkjQ0VknS0FglSUNjISQNjVWSNDRWSdLQWCXFAY1VUtTQWCWFhsYqSRoaqyRp  
aKySpKGxSpKGxipJGhqrJGlorJLigMYqKWporJJCQ2OVJA2NVZIOFZJ0tBYJUIDY5UkDY1VkjQ0

VklxQGOVFDU0VkmhobFKkobGKkkaGqskaWiskqShsUqShsYqSRoaq6Q4oLFKihoa6TQ0FglSUNj  
ISQnjVWSNDRWSdLQWCVJQ2OVJA2NVVlc0FglRQ2NVVJoaKySpKGxSpKGxipJGhqrJGlorJKkobFK  
kobGKikOaKySoobGKik0NFZJ0tBYJUIDY5UkDY1VkjQ0VknS0FglSUNjIRQHNFZJUUNjIRQaGqsk  
aWiskqShsUqShsYqSRoaqyRpaKySpKGxSooDGqukqKGxSgoNjVWSNDRWSdLQWCVJQ2OVJA2NVZIO  
NFZJ0tBYJcUBjVVS1NBYJYWme6csxm1ujJ5Eo1VUjhorJKkGTlowlh9gcYqKRw0VknS0FglSUNj  
ISSNY6welMVYNdNorJLC0vr0LMZqvUYy828aq0QHjVVSaLqbYrqh0FLXIMmmyEZvZkI2ARqrJFFM  
/zTEFFOjMpZhprAfl4WBxipJGhqrJGlorJKkWWIKU1NMjQotmIZ5CCkMrU2x65VdGpky0xRCNNBY  
JYUmlrFKSCGgsUqSjpaxSkghoLFKkobGKkkaGqskaWiskuAXiopamiskkJDY5UkDY1VkjQ0VknS  
0FglSUNjISQnjVWSNDRWSXFAY5UUNTRWSaGJMIZ/NYWQwhBlrI4ZM8abk5BN41ITTI3KWlabQkhh  
+OCDD2iskkt56aefshqrV199tTcnIZvGrFmzaKySRFIISpSxinkIKQxRxuosUwjRQGOVZXjxvyH  
Cvbb319OOeUUOffcc2W77bbLF1L9U1LKFL8z8ovz2T/mX3VTUr9afTnvvPOc5Z955hlvjenrJ7mJ  
XQdefPFFp46gruy8885pYr1UKVOfrP8PfrbNNts4dRPL9+vXz1sj6xhJrwMYNXjqjac6dWyvfbK  
r09tTGQrp6SkSqKKnNnyTGf5m266yVsj6xhJrwM//PCDnHnmmU5/dMghh+TXsRhRunRpadGihZxx  
xhnSvXt3Wb16tbNO1jFi14EZM2Y4fdjZZ58tRx99dIF6FWW8PPbYY+Wss86S9u3by7///uutlFWM  
5NeBJUuWSJcuXaRly5bSvHnztLqEiKpjhx9+uLRq1cqpoXmNtTnTWCvJHiF0HrrnmGjnttNOcY17F  
ihXz69NoU58ynVt+ZP5l5tlvl/0clx/L2wMsWMelXQcwOAe66pxzpzGaNWvm17EXs9SxSeZfVVKy  
27a75fkXI0aM8NbIOkbCobFKCmB3FrjI5+abb87rhMKEVGqgiUxllonGJrx57eX79u0rc+f09bbe  
TiqXsPf1vHnz5OGHHw6tl5sS9vLXXed/P33396WSK6CE8QXXnhBypcvX6CO5EVHE9nKeSa8ee3l  
L7jgAvn999+9LZFcBW8oHjlypNSpUye0nmjDXvakk06SCRMmeFsiucratWvls88+kz333DO0nmjD  
XvbAAw+Ur776StatW+dtjeQiGzZscC4MHXPMMaH1RBv2sg0aNJAPP/ww72IRyV0mT57sXHgMqydO  
VDlxxkSmgmmYJ7B8tWrV5H//+58sW7bM2xLJVf744w+59NJLC9SRtBhulIP51cT2Jrx57eWfeOIJ  
Wbhwobcl+hckHxqrJJThw4enXznMFgpjNSxq1Kjh3CJJcgs8FzXbLYtFGeXKIZPnnnvO2zLJFSD  
d9lll9A6USAUxmquO+++7wtk1xhzpw5cvDBB4fWhyTiiuu8LZMcgUYURglH1YfkggYHjDYS6B  
0alh9SGJwCjrRYSWeVsmucltt9wSWWh8KhNJYDYsmTZrI9OnTvS2TXOHxxx937vwJqxMFQmGshkXd  
unXlm2++8bZMCI1VYmFfcXnjTdcO5HQKKSxWqZMGWcUBilhRFzA++b9b6RCqkJonUgiXnrsJW/L  
WeBFxxlFRiunPbYkWxSBsdqnTx9vy6REkaVfWD1rtey9w96h9SGJuKztZd6Ws8B+bMsj2z5bLXli  
wSeG1ock4syjzSJ46uynm1ZROyvTi07hdaHJOLQ3Q8VWeBtOBUSyYWK4jRWGzZs6NwRR3KLwYMH  
h9aH0CiksVq9enWZNGmSt2XTXXHkas5DY5WkgVtmcZXP7zRCh84HoxDGaqz1M7a86GAiW9mY/++w  
59skVqztZixzTDQ0Efa7GFtkxOpnCmmssi8rgbGHiUUmMpVi7sfythGnH0NpZSLsdzE2n9jaxFQT

mcp/VcdQ4tSzq02E/S7G5hVfmshUNvc69qCJsN/E2CijllyqplFKPZbbEXv/F8JYtbeB5wMTAmis  
kjSWLI0qhx12WFrnERmFHLHKKIFxsYkttcwzsY+JsN/FKLIRBCNWGSUsDjGx2MSWWi4wEfa7GJtP  
4CLedBNbaulplux3MTafqG7iaxNbahlglux3MUUpuFMGIVQYjMgo5YtWPdu3aeS4KyXVorJI01qxZ  
4zyL8oQTTgtPEKDI1YZwcAJfbZSzCMk8kqc0REYsbqTibDfxdgiI1Y/wxGrjGA0MZHNWP0vR3rF  
KRyxuvlHLRMcscplOjhilbGZRCytxBGrjEJE7P1fSGMVL8Rt3bq1vP32256LQnIdGqsklOeffz60  
EwmNQo5YRQf48ccfe1smucKX738p5VLIQutEEvH8gOe9LWeBj8cpUeDFBbVr1w6tDwWiCEas3nPP  
Pd6WSYkiS7+wbNYyaVI///E5Uf387t7WYylimzHntUixx16XGh9SCJOP/Z0PmO1JBKxvzqc1SG0  
PiQRBzc+WNYuWOttOQusYyWKnj17htaHAIEEI1br168vs2bN8rZMcgW8vCqsPoRGIY3VKIWqyNix  
Y70tE0JlWQA6bH99ttvpXHjxnkdSMYrQEj1V7PoYceKj///LOsXRtDYJESBfb5L7/8IkcdvRo  
3ShM2OvZeednQMf6jTJLdatWyczZsyQVq1ahdaNtNgEY9VfFx5g/84778jKISu9LZNcyf369fLv  
v//KZZddVqBeFEXY68IFz2XLInlbJrkCXoixYMECueuuu0LrRWHDxtcDDzzAt7XnKHjHwqBBg0Lr  
RWHDxtf1118vc+bMkQ0b4rj3pCSxYsUKee2116RixYoF6kVabIKxaq+rQ4cOzstLcXwmucWqVavk  
008/dd7YH1Y30kJprNrrad68ufz222/OeQYhPjRWSSj2m+3+/PNP6dKli9SoUSO9g0I5HUyEsVpq  
j/QODSPIr/8cpk/f763hfTtkdzA3ueLFy+WHj16pB0IERkPhoElzletWjXp1KmTTJ061dsC61iu  
4u93nMT17t1bdtlll7S6khcRxmnpNul1rFy5cnLOOefl999/76wfsI7IjvZ+f/jhh2WfffZJqytx  
l6y/O/nkk+XDDz/01s46lqvY+x0Ge9iz8OMcL8PmwcXNI19+2Vs761iuYu/3kSNHyvHHH1+grmxq  
HTvggAOckWQ+rGO5ib3fv/zyS2nZsqWULI06ra4455ZRxupnJgLGaqNGjeTee+/11s46lqvY+x1v  
7D///POLcuXKaXUiz7+IMFZL7ZDel+24445yww035A2iYB0jQWiskk2ic+fOTidTIVVByjxZjr8j  
CpZ/TNRPYq1db/CWJCQet956q1PHYGCVLVs27eAWFu3bt/eWJCQeTz75pFN3SqdkS7mu5eyeq2A5  
JSXH7388R9cTFXj2VoUKFZx6hudx2X1WWOy9994yd+5cb2lCosHFnVq1ajn1B3UsvyzaYYcdnBNO  
QuKCARa77babU3/Qn0XVMdwiy0d8EQ0YzXrEEUc49adijYpS+vPStgJLLx+lpFTIUvLiEy96SxMS  
j7POOsutY+UqSplXsvgXk0xsk5I+N/XxliQkGhqrpNBcbkroA+jNZ9ubMsMUQgrDfffdlyba7ahU  
qZJ89dVX3pyExCN4pflFU0yNylheNYUQDcE69sknn2S9SPTII494c3IkBIHsJ5MnjxZttlmm9D6  
hbjuuuu8OVnHSDyC9QR3m9mPCQtG27ZtvTlZx8imsdGUY0zJdG6JaZjHmZd1jGwibU3JVMcam7LA  
FEI00Fglhaa7KVZ3lFbqmqjLZFEIKg/1suWDAWMXzdAgpDENMMTUqYxlmCiGF4f33389qrOL5loQU  
hvHjx+eNXg0LPHKHkMIwc+ZM57brsPqFwFuyCSkMK01paoqpUaEF0zAPIYWhtSl2vbJLI1NmmkKI  
BhqrpNDQWCVJQ2OVJA2NVZIOFZJ0tBYJUIDY5UkDY1VUhzQWCVFDY1VUjg2xjRWeacGKQqQ0Vkm  
mP4plrHKfowUAhqrJGlorJKkobFKEsXorNjGKjUZ2RS8ehPLWGUDlwporJJC071D91Bxhahbqa5M  
nsgRq6Rw0FglSTPKiSGh9cuPYc9xxCopHDRWSdLQWCVJQ2OVJA3eut50v6ah9QuBaf6b2QnZVFq3

aB1avxCN6jSSmX9zxCrRQWOVFJru3bMYq3XrOi9TIKQw0FglSTNkSISxOozGKikcNFZJ0tBYJUID  
Y5UkjWOsNs1irJppNFZJYUFFVa/EOjj0NcRooHGKik0NFZJ0tBYJUIDY5UkDY1VkjQ0VknS0Fgl  
SUNjlRQHNFZJUUNjlRQaGqskaWiskqShsUqShsYqSRoaqyRpaKySpKGxSooDGquqKKGxSgoNjVWS  
NDRWSdLQWCVJQ2OVJA2NVZIO NFZJ0tBYJcUBjVVS1NBYJYWGxipJGhqrJGlorJKkobFKkobGKkka  
GqskaWiskuKAxiopamiskkJDY5UkDY1VkjQ0VknS0FglSUNjlSQNjVWSNDRWSXFAY5UUNTRWSaGh  
sUqShsYqSRoaqyRpaKySpKGxSpKGxipJGhqrpDigsUqKGhqrpNDQWCVJQ2OVJA2NVZIO NFZJ0tBY  
JUIDY5UkDY1VUhzQWCVFDY1VUmhorJKkobFKkobGKkkaGqskaWiskqShsUqShsYqKQ5orJKihsYq  
KTQ0VknS0FglSUNjlSQNjVWSNDRWSdLQWCVJQ2OVFAc0Vv+/vfsAj6rK+zgeICSE3nu3YC+76trX  
7toLlojSSwhiL9hBRcHuKuoqKmIXO1gAUZGqllqiggVBeu8dWu89Z+YmuRNuCTw580rm+7nP/wky  
5YadszeTb25mEG+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVr  
hFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+EVYRGWIVr  
hFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFv  
hFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhG  
WEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhG  
WIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhG  
WIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+EVYRG  
WIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQ  
VhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuE  
VbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuE  
VbhGWIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuE  
VbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+E  
VYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZY  
RSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZY  
hWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZY  
hWuEVbhGWIVrhFW4RliFiHBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZY  
hWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RliFiHBW  
EW+EVYSW1aGQsJpRTzN+IqwinH63FJWUzI0ZjRhFeEQVuEaYRWuTZs6TTXKFBJWMwmrCGfBvAVq  
UbeQsHoOYRXhRMLqYYWEVXMZYRVhEVYRb4RVhLNDyjKbOQwFbvXMNsNs9nrAbjFrp5/Z/OvKv2WY  
bYzZWGMlg7AK1wircMp8DZxmthpmMysqclvObHytxG4za2eB2VqYzb+u/Ftrs+VcF9hIZt1sNNuJ

ZvOvK/9mL7PXYY0hDMLq4o2witCKFVaBEIoVVoEQCKtwjbAK14oVVoEQih1Wgd1U7LAKhEBYRbwR  
VhEaYRWuEVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWsVtWrVqIJXOXaMX8FWq/vn3uF7v8W+3t  
tfXloi/NDbwbAsW0du1aLZy3UGvnrdXNa272rarYrdyOchq6ZKi03LshUEwbNmyIPHFav369Hnvs  
scAnVzkzcOBAbd++3bslUDybN2+OrDG71l577bVCw+odd9yhTZs2ebcEimfbtm1auHCh1v21Tp8t  
/UzVsqv5vklGbl3XdtWahWvMjwbw8OOHdq2dau2m88DJVN2drYWL16sNX+t0dRFU7XXtr18qyp2  
O2/DeVq8YLE5+Hk3Bopp6dKIWvnXSs1aOEv/2v1v36qK3f61+V/6Y8Ef5kmcd0OgmFasWKFly5Zp  
0aJFOv/88wOfi9lp3ry5v32W+9WQPEQVlGoHeYJc47rr79e9evX14EHHqiKFSvmHYAGmiloW2Cm  
WYqqpFSJ3K5Bgwa69957vXuMvX8kJ/8aePDBByNrxK6VatXMN4c5a+zWyGoK3jaaOTpFFVlq6lAD  
DojcvlevXt49ssYQuwZefvllNWnSRPvvv79q166dt8aKMWlpadpvv/3UsGFDXXrppd49ssYQuwY+  
++wz7bPPPTp3330jx6OgtVTQIC5dOnLbxo0b69RTT9Xq1asj98kag38N/PjjjzrkKEO01157RY5n  
uWvoQDPLzBS0PWDGXK9Zg2aRbxyPOOII/fbbb969SmuXLNFPX3+jEUPf0dAXXtSr/3tGbzz3nL4c  
Nlx/Tv9J2es3SFu3SlvMZGfrh4nf6t7Hn9Ph/31at330qVbOmWd/mhC9M3O5+aSl7ebj5i3atHKl  
ls+bq7kzZ+qPadP0y+RvNW38ZE0Y+YVGv/+hPn7jDTNv6vMPhuv7cZM0b+Zv5jb8VD7RctaZDanH  
H3+8mjZtGllnpUqViq6x2mZmRFZT8PaaGXO9RjUbRY5l9mvmhAkTlvdpcSxDzhrYao4lNm41atQo  
coZg2bJlo2ss1cyYyGoK3r40UyZFdSvUjayvZs2a6d13343cp8Uag38NdOnSjfJczH6PmJGREV1j  
xZwaNWpEbmeF99sTLHKwxhCEslqd+A8WY8eOVZs2bXIPMLlPrPzztJmCtoVm9jPjXdd/+65du2rK  
lCnenjhIIRP/Y/3dd98pKysrc13kzu1mCtpsWP23Ge+6/tu3bNISX375pbcn1lgy8T/WNhzceuu  
gWtkd8Z/+1NOOUXDhg3z9sQaSyb+x3r+/PkaMGCAKIwqFLhOdnX8tz388MP1+uuvR85MRHLxr7GV  
K1fqaaeeinyDF7ROInOomeVmCtoeNeO/fikzlauq7lUt1eS9D1Vr2o+qtmChaq1dr9rLVqj2vIWq  
OXuuqk39QRVHfaG0oe8p/fW3lPHmUKW/+74qTJykGr/MUKPxE9RwyMuq26evGt1+h5re1Ufn7u2n  
ZvfdR6bmY5O+96iZmUZ39VvdM82fekb7Dn1bzUZ+pnq/z1KtDRtVx/xb65h/bu3NW1RzxUpV/e13  
VZkwUdWHfay9335XJ5i5xPz5IXObKZMna+P69d7/MogXe4b9kCFDIj/czlkjO62x+mZmmiloe9NM  
wG1toLAvd7JkyRJvb3y9TCb+x/q9997TcccdF7hOlPnh5iszBW32MnudfLe3YbZPnz76888/vT2x  
xpKJ/7EeNWqUzj333J3WyO6O//bXXHONpk+f7u2JNYy8hFUEeumllwIPJoGzC2HVP/77HT58uLdn  
JAv7RS9oLQTOLOrV//jv95lInnvH2jGRhn/hUqVilZk3Ee/xrzP4qN5KL/XUyezaXf024nl4dO3p7  
RrKwscsf1QqdXQir1SpV0v5HHKGmJzymesM+Ue23P1Ctoe+rzgcfq/4XY9XozzlqsmqVGi9brkab  
NmvfaT/oxO7d1bZtW13ZrZsyO3dWp1atdOn5F+i0tlfo4JdfVcMVK82sUpM//tBe48erxaiROmjY  
cB3+7ns6cuhQHfbyy9r3ued0qLnuP99+R4cMG6Z9JoxX059/VpM/Z6vJzzPU5JcZam722XzDBjXf  
ulXNtm9XM/ONa3Pzv0UTM83HfqNzBjyl3+yvmyNuWpnHMmYtFTS7EFYLmoMPPjygywkl2uvvTZ3

DRT6vH8Xw2rQ1KlTR7///ru3ZyQL+8ObnDVQ5PeWuzk592tD/njzdQ7lQVhFLv9PXD744lOdDiQF  
zm6G1ZwpU6ZM5MxYJfJkycrPT09cE3sNLsZVv1jz/hCCVTID4rn/zBfdcvWDVwPLubB3g96ey4E  
P9je8xTymG1ZuEUH1z84cD24mF5t817mpECssRLnzDPPDFwPO01RYfVB801hSpoyDj1ZVbrfoJr/  
fUa1Xn5Dtd8brrqjvID9b6ep4aw5arx0uZllarxqtZqs36BDBw/Rfzp3Ubfu3ZVlJjMzMzo9MtXD  
fOzVpYvO6NdPjUeOUrN16yIRNGea5Zv8f9fYzP6//aYzOnRW+0taKatjR3Vt315dzLTt1Fkn9bxS  
B40aFYmq9vpVvpmsA54bpLk5LzmAuOjcuXPwmso/cQirRx99tLdXJJO77rorcD3sNHElq/YlJYv  
580XSpwint8MemBQ4HpwmZVTkuvnsT97ey6eHTxBK9Elq4ixZs2ayOvV5Bw0ivXTnhBh1dVPk5g9  
Z4q1BkKEVXv/rLMSOCeb2WGmoM13mVkBuX92teXuo7DPKWfbauZoM0H/LubvMy3MrDJT0Pb/ucaK  
s85amgn6dzF75BT61hRYfU+cz8pFVT1mptVd/jHavDDT2q6aVNgALVjY2aj9Rv07wcf0hVnna1u  
XbtGgmr37t1zJ7NbN2V17qTT7+2nRjaArlsXeF8FTSMz+/36q87o0lVXtG0bCbW59232l3XF5Tp5  
wINqMOKbNdm2TY3mz1f1/z6plMqVg/83YHZ7irXGQoZVno8l9xTr8Q8ZVlJjXTsb1wUtnnPixL6  
fMxuxXg+tp/ZVpsNJRthFTHsO7HbnyQHHTAKmpBnrDJMkROHM1aZEjbnmylOXPo7bjasnmEm6N/F  
/H3mKDOrzeypWwczQf8upuROccPqjberwXfT1NQ874vMjh2B0TMSVtes1Un9B6jdf/5TSFjt7IXV  
z9Q0nmHV3nenTjrlvv5q8PKYNdm6VY3mzVPNJ55UaqNGwf8bMG4nDmesMkyhE4czVpkSOIX1hr/5  
9i+ZrTEbSjbCKmLYN8ewv6JtnzgHHtSCHjNWmRBTDXAGatM/uGMVcb1cMYq8zeaYn0dKyqsPmDu  
x74UwBGnqnLWjarxzltq8OMvavjLTNX5eITqjh6jeuMnqf6U79R42YpldLVx9fAhL+usy9upa7fu  
6uGLqjnxS2dOWB0xUk3X7npY3X/mTP2nfUd1aNNGV2Z2V8+uXXVp06RlwQ49Yp2Ovc99yPXs59P  
w5m/qsq99ymIfIXg/w2Y3Z5irTHOWGVCTLEef85ZYKGM1bxN0dYRaBEv3nVmDFjvD0jWXz99deB  
ayFw4vDmVa+88oq3Z5Qohbxc0dxpc1UtpVrMmgiajJQyqpFSVg1TUtXMzN6RKaPm5mNj8/d1blgw  
/x10W//cf8P93p4Lwcsr7XkKecw2L9isfWvuG7ge/FPOTNWU0mYt2TWVqr28NdbUTEMzdcxI5VOK  
OA6ayWyV6e25EKyxEufkk08OXA87zS68eVXISpXV7KijVa/37ao95DXVGf6p6owcrXqTvlGjmb+p  
3mdfqPab76nu+x+r9tij2ufdD3RWpy5q2+YydezUWe27dFW7rCy1veoqtex5pY554kk1nThj+86Z  
o/OXLNCBf/6pg3//XYfMnKlDf/pZh/3wow76dqr2nfS1Dpr6XeS/D54+XfssWqRmf83VPp+OOP63  
99EBva7Tvm+9raZffKmG5jPnVLNp7csV2LDaeMxYndHvAS1et97+T4M4ueKKK2LXUkETHzNWDz/8  
cG+vSCa33npr4HrYaeJwxmr9+vW1bNkyb88oMYp4fvO/+6XuWZKFeM5VZhJM98bTB8z3dtz8fAa  
qyUbYRUFWrFihY499tjAg0nMhDhj9ZxztHGjRu9PSLZbNmyRRdffHHg2oiZEGes/vOf/9SSJUu8  
PSIZXWW+8Y9ZF6VKq1z5imrSoKFOP3w/3fefpnrg9Ma67cTG+qVXFenedG2+o5wmdq2qIRfX1RPn  
NFbfUxqr3b+aa7/991f5qtVj7q9hw4aaOXOmtzcko/79+8esiZRSpZRaPkP1atXViY33012VmurR  
Mg11Y2oTfV22qpSWpq3p6ZqaVlmvIk2jgamNdG/ZxupSuakO2mc/VaxTO3lfofXkZGhcePGeXtD

MhoyZEjueijwh5G7EFbtIC6TqrSGjVTpmutU58/ZarRtu/fSANKxLw9g/9xk6zY12rBZjTdvUVNz  
vSbbs9Vk02Y1mfOX9nr7He37xpva/+NPtf9Hn2j/YR/rgA8/0gHDhuuA99/XAW++pQPN57/fs89q  
74FP6YBXxtEBI0eq+fTparp2TWSfkf1EZkfk0z1T1r6xVcMtW1V71p+q8cll1XnwSXUc+JzGfPGI  
1q9day5FPI0aNSryhrJ2bRS4xnYjrPrv64knnvD2hmQ0bdo01apVK2Z97DS7EVb9a6x3797e3pCM  
5s6dq/3Nc3X/+oj3tGvXTtu3b/f2CEQRVhFoh3kSnWPlypUaOHCgGuV7PavcnwQVEVZL7R/75My+  
Odbzzz+vDRs2eHtAstu8eXPKLOmDD459d+3cNVZEWc11Uuwaq1evnh5//PGYdwT1r2kkj5zHfdPm  
Lbq+9x3q8Z9D9Oy5GfquZ2V90am2Bp3XQCPa1dTzGr6pVc1rbutvNQ3Xdv7llsE1nk3VNIbrepp  
Ws/q+uPaqvqgbV1N6lpV77WpoFP3q6n33nojcv8Wayw5+R/3x599VledcJleL1Nek9IraHxGTT1b  
poGGp9bSN2nVNN3MyvTy2pGepm3p6ZL5uDytgoaWNeuqbA3NSaui4Wl19aW53mflK+qSmnX1whMD  
vXtnjSUR/+M+YclEtW/fPuZrnp3l18siwmqpx2K/VtrJ7N5doz/9VHNn/hp5KajhY77Sm+Mmasj4  
CRo8frxeGjdOb48dr5Hm776eNfITvp6iKd98q++mTNW7w0bopicH6cC77laHgc9o5MjPNXnKd/pu  
2k/69vsfzUzXt9/9qG++/V7jv5msLydM1Khx4/Wp+Td8bO5n+Dc/6r1JUzV0/Nd6w/z96+Mm6G3z  
548mfKvR4yZpovnvN6Z8q0WzZmnLGI6fziX/Gvvhhx909dVX77RWImusiLBa6q2d19gl1yiL774  
wrt3jmPJyv+4zzL/n77zzjuVlpYWs1Yia6ylsFpqrHcd3+1OPfVUDRs2zLt31liy8j/uiXyT0kMP  
PaQaNWRerJUCf2iUb/Jfz56o89pr0VeNhEIQlhFkYK+ON1xxx2qV6OeDmh2gKq9US33i13+rfTi  
0qp5XE093udx75ZAsPzr7KmnnlLDug21f/39Veu/tXyrKnYrtamUKp9XWbd2v9W7ZR6eWCHH4oUL  
dGevjqvqW0OpT7pGta+lJ66oo0nda2jopfU0sWuNSFDNiap2dvQtP413ZETi6reZ1TSiXW29d1k9  
zb6+qjbenq6pndM04pEeGvvNd5Fwi+S2acNGPX3X3Xr3oEO0o2yavkqtrvftzBpLq653y9bV52k1  
c2Oq/Wgn28ym9HJamFZR36dV1ShznXfT6uqX9GraVDZd35on8xPad9GM2bO1iSfzMPJ/Xfv99991  
xBFHqGnlpmp2YTOVWVVG9xUydiv7ZFkdd9hxmj97vnfrKL5Wwi//eli1aUkXDWq1Ej7/Hsfpf2Z  
5ltVsVvQO6k6aP+D9P3E771bR7HG4Be0Hlq3bq36letrv8P2U/kp5X2rKnYrM76Mmh3QTCpfGend  
Moo1Br+g9WB/g82efGPPaK1cuXJMOPWPPXO/bt26kd8U8WONoTCEVYTW02zmMBS41Tfbr2YDwrjP  
bP515d8yzDbWbEBhFs/5XQMuPUiftUnR1rvK653WdTTk4noa26Wm5t5QVRvuqKDspuWU3deM/eiN  
Da3247gutfrRu3paeHNV/XhINU3NrKKNt5XT+71P0yNPPqc1a9Z5e0Ky2rRmrZ67oKU+TM/Q9vRy  
+jitiL5Mq6/PzcdZ6dW0Nr1iJKZuzzfZ6WmR2DolrYbeN9efm15Vv5j5umwVrUkprW9OOU0vvv6G  
lq5a5e0JCPaj2Wqazf810r9dbzYgjEVma2E2/7ryb23MBoSxxWwnms2/rvybvcxeBwijTZs2gVHV  
TosWLSJnvAK7grCK0LLMZg5DgVs9s80wGxBGP7P515V/s2F1jNmAwiyaO1v3tD9FH11eQdv7lNfE  
bjU1oI0tfdW5pr7Mmq9Psprpze77aXC3gzWo26F6vvuhkT+/3v0AfZC5t97t2FDjutfUutsr6dse  
NTSyXQ3NuzZDw/tepMGvv6t163lpk2S3ad06PXFZe71ZpYa2pWdoamo1fVKmlsak1tCY6vX0aeOm

emufFhp8wEEadMghet7Miwcdolf3O0DvNTNrrHYjja5QS6vLVtL01Or6KLWW/kwpqwlInnaO3hg/T  
irX8KjQKN81sNczm/xrp364z257is9Wrdfvcubp7/vzlx8fMN7nLE3DW9prt2/XZmjV6dskSPbhw  
YWT/zy9dqqr12tdla+plyznMS0wW2FhtbXZgDA2mq2osGqvA4RhZ5AOiqp2bFhdsGCBd02geAir  
Cl2wCtclqwJLvjsQwMe1Oe9j9aaW8ur33+aqvWZp6l1+27q3D1LmT16KDMzM3C6dc9U96wr1bFz  
Z7W54Cw9csmB+vDSqhpyXqomvHyvVqzdpGx+PSjp2deKfvGFF/XRaWdqfdnyeqJCA7X5x7/VunMX  
dcwya8xOjwLWmJnuV16pzt266dKzzlW/5ofqw9l19Zl5gj/huhs4NwfFUpLC6o1//aWUCROUMmIS  
5GP9777Tr5s2eZe68ery5dr3hx9U+ptvIDJxolk+/lpp5s8n/fKLnIuyRJvNcf79lSujn9e4cdGx  
r007dKl3DyUfYRWuEVaRCIRVxBthFeHsKGZYpTlgd5m1U6ywyhpDIvAtWq1HHnpEY249UpvuKqf7  
LvunOr5SRN0zbVCNRtXu3bsXOPbyrplZ6tK9h57tdoRGXpGul88vra9f66+1m7NZfoi8ocErr7yq  
UWf+J/KO/0/ud6Dan3dRdP144T7/uvJP5HJzvU5XXqnHjz1eo8qW18vmCf43N93i7QEohDkIFSus  
7iEHq/+PsPrEokUx+6w8ZYp+2hgbcD5dvToSXCPH1Y657gcrV3qXRpXYrwmH1bssMoXRewOs26K  
HVZZYwiBslp4l6witKz2WYEHJTv1Muppxk+csYpw+t3eL3B92cllydCY0ZyxioJNGDdWD956tR68  
5B96//JamnJlFb12RX3dfsXJatf1SnXq3ktdMq9U5+49C5yuPa7SFZ2664ILLtYTbffTlO7l9X2P  
Cnq+dVMNuPIKff/tZK3b6Pabfvx9/TZjpob0f1BPHHmChlYxa6xSZQ2tXkd3H3ms2vflUserrlIX  
e0ZqT7OeCpiuV/VSh8weOv/iS3X/AYdoYrkK+q50Ob3XoLmeat1eP076Whu2cu4qCjZt6jTVKBP7  
Dsj+uS6zZl2xumXHDv1m/u7RRYt09syZqj11avT6OWeTmttVmJfX/78s3rPnRv5Ff++tX674Ot1  
69T9zz+j+xo7VqW//lqlzNiPkYBq/s5eVu3bbyNnp0bu17tOzqR89VVkv1mzZ+u7DSX7ZWEWzFug  
FnVbBK4vO63P4YxVhLNx40adeNiIgevljr3MXgclg7CKeCOslrSsrELCar16mjGDslpw+vUrJKxm  
ZGjMGMIqCrZq9RpNMt/83tf3Hj3R41R90qOBpu7RWKMz62hst8oa0bmyhnesrI86VvLG/jlnov/9  
QYfKGpdZSYtvqqBn2u2rrP8coBcuSNPHj1+tb6f/ptVr1mpHdra3RySbpcuX64svvtD9t92hB08+  
VcPrN9LHdRtpVJ16GlverLGKITWsilnlSppmBn70T/27z4wl39eqblWIS2v1xo3V/eDD9P/ypbT  
OPMv0Ceff66FS5cqmwWGQkybNk01ahQSVq8rOWHVvtqqDaop9tf2zeX7mn973/nz9cPGjZHgut5G  
182bl6+TGomi5jpVzcf2s2ZpwrrgNxt8yFzXv097xup369d7l0Z9sGrVTmesvmX+/58sbGyw0SFo  
fdmxsQlIlxJWtywkrJrLCKsli7CKeCOsljTCKlwjrCKsVStX6un/Pq7nrjpDH3Rvp7tj9fHV+0v  
3Zsu3VNOurucdvQteOx1Nt6Roe97Vdb9HU5QVstT9cKFZTXtrX7asJXfSIO0betWvT1kiAadf7E+  
qNdM/Y88Wm8eeqhUzqyftDTtSE9XdiEjM5vNdX+uWFGPH36kup16tp5KTde063gndxRPsoTVTdnZ  
GrJsmU4zzy8jl9vQaT/mnFFqzzLNGe91UHMvmzhRTyxeHLmf/AaYb6Tzh1V7Rqv2/Y1VvOF1VfN  
55IsCKtwjbCKRCCsl4lqwiNsArXCKsla8WKFXr68cf0XK+T9U63Rrry8nP0cs8jpbvTzKQu2+G  
tvcpV+DY6yy/tZKGZTbShe1P0jWXnqAXzk/Td2/cq3X2NVYpq0lvy+bNeuulIRp03vl6r359XXva



6Xri+BO03YbVsmUjH7elpxc4Nr6uqVBBIxo00D3HHatep5+mp1PNGrv2Osl9iiWZwuqLy5bplJyw  
On68zv/1Vy3dZs9j3X27G1Zf44zV3CGslizCKhKBslp4I6wiNMIqXCOSlqzFixbqnluu132XHaZB  
7Zur0+WtNajnCdp2Z5rUtzhhNU1Lbq2m1zMP1FWX/UfdLjpJgy8qry+evk4zZ83V1m3bvT0hWW0y  
3+i99OTTEvi0M/RCo8bqdsFFeuj0M7Qho5hh1VxndaVKes88ob/xtNPU8fQz9XSZ8voms5dWb9oo  
XgQARUmWsGp/0LBtxw49bF8Kwluc9t37D/zxx8iZrCu2bdMmc7n9If+r5sxR+uTJkfuwr8F6tfnv  
qfl+vT9HccLqOwFh9RHzeWzlzo68PEFJR1iFa4RVJAJhFfFGWEVohFW4RIhFWEsXLVC/azvrgVb7  
6oX2TXxpBWfriU5Havtd6VLfcsUKq4tvqalhnfZR94tOVI8LjtHo9hU17sksff3ddG3estXbE5LV  
pg0b9Oydd+uBfxypIQ0b6YrTTfdxx2vjeUylmejFn3GalmtqlhRbzZrrquOPU4dTzhRw1LL67vL  
O+inWbO0aRtrDIUrSWG15+zZSjFf2yO/vm8+ZkyerF/yxRR75qr9uwcWLTsZ5rlm5M2rbOzM+fX/  
SZPU6LvvdNGvv0Z+/f978//Rwn5A0WfevOgbUdnbe29INX7tWu/SqNne67Ye/fPPKmuDrd3f2LGR  
z8++QdbPJTz4EFbhGmEViUBYRbwRVhEaYRWuEVYR1rJFC/TkjZ30UZfG+v36arr1suP1aveDpXvS  
zKRHftXfnrla4Jjrrb69kkZkNdELXQ7Wm9321Z/XZGjcY5316ReTtHHTZm9PSFZb1q/X0F7XaXij  
ZppTobLuPeJfGnj4PyOvrWrPRIWa/WjWW0GTWIYbypfX5w0aavB+B+rlFgdqRql0Tb/4Uk2cPFnr  
+EYSRSgpYbWwl75w9bIY8dynq8/x74CwCtclq0gEwirjbCK0AircI2wirAWzvld913+Lw1vkxp9  
s6p70/XLtTU0sM2+uvqSE9Sp5Wnq1vLknaZry1PMnKwrzj9V97T+h2ZdV0Vb7iynHX3StOG2snrv  
trM1cNArWrM2+FdLkTw2rV2r565or3eq1Yi8EZWnqbMrVdWghnvr+qOPVcfTT1PXk82aCppTzBo7  
6VT1Pvwo/VsXuJaVTdf2tDStN8e4r88+R29//LFW5DtZdsivJJ2xir8nwipcl6wiEQiriDfCKklj  
rMI1wirCWrpwnh7q2UojO9eOnIE66+qKerVtl117wdHq0bm9umf1UmZmj52nR/TjhZd3043t/6PZ  
N1RR9l1pkddmXXxdukyOaKf3P/ICGzZGX/sPyWvTuvV6tlUW3q3bIPlu/3PTyuvd6vV0wwH/VI/W  
bZV5lVldj0VMC27d1eP8y7QT1VraluZVG1LS9filFKadGFLjfqK63eQLxH4QircI2wCtclq0gE  
wirjbCK0AircI2wirCWLfmiu+/qow+uOjzyeqn21/8X3JihYZ3rqnen89WhU3d16tJNXbtlqlv3  
6HQ107mr+ftOndWx1Xl6ot0h2tyngnR/mpbdmKY3LkrXqGfv0rzFy7V9e75X7tu0Ulo+08yv0Y8r  
/pC2JSC+Zm8335Usl1b/Ja2cFd3/6jnm81llLuM1Ol2yb1418Mmn9MYJJyk78qv9qVqaXk4jKtfU  
XaedqQ5du6lTt27qkmnWlzdZdXTubtdeV3U870Ldd8g/tLJSJalsqtampeudlFladeU1WrZ2rUK9  
PdrSpdJMsw5/+y36cfZsaauD9bBsWex+/vxT2rLFuxCuEVbhGmEVrhFWkQiEVcQbYRWWhEVbhGmEV  
YW3cvEUTJk/TqEe66dvOqVp/m33d1Axt75uhdXdU0Pybq+j7q2vq3c4NNKhdEz3fvqne6dRQE7Jq  
a17vqlp7e0VtvTMj8iZXujdVc2+upjsu00GffffSht4d8Jjwo3Wq+xN5h5hYz/czt5k7wLnRk7njp  
zYukh+xZuWafdm4y81At6au+0oYl0uJp0sD9pAdqSI/Ukx6sKU19zrsDhLE9O1vT58zRmNvv0qSU  
clpZqoxUrlzkTavWly+vhRUR68cqNfRhrrfp6vkETDWrUVG/Waagx1WprduWqWI2+gjZ711eZMlph

rt//qGP1/kuveHsloXNn84zPrIWcqVZNmjXLuzCORr8+dj/mOYB+/dW7cDdNnSoddJBuVbpUv75U  
w6zdgQO9C+FWWIVrhFW4RlhFihBWEW/mWS8QDmEVrhFWES/ffDIKj1/TWh/0PESjOtfQyLap+vKy  
FE3tXEp/XF1ef15bUXOuq6C515XXbDN/XFNBM3qV14T2pTSqbRmN7IJLH3Zpqvduv1CjP/1I8xcv  
8+45n4kPS7eZL7F3mrGB9f6K7sPqz0Olu8y+bMy1+767tDRrIHehZ+EU87lUiH5O9nOz15vytHch  
4uHHCRN13wWXaFD9xno3pYyGm+PUaDPfmPk1LUOz0itodloF/WX+/KeZ39PLa4aZSSml9Km5zgdM  
Xi6bocGHHalPX39Dsxcv9u7ZZ8cuvj1Odnb0DNvt2/l+7up9FMcNN8SG1QYNwofV6dOlqIVj7/fr  
R70L4UdYhWuEVbhGWEUiEFYRb+bZKRAOYRWuEVbh2uBXXg9cXznz5jvvedcspuKG1e2bpYXfRs9w  
HXxC9OxSe6apvd29ZaUn95Xeu0L6+S1pw1JpR75fCF/6s/TF7VL/ytLN5jY5Z6r2LRX9aD8H+/Hx  
ptl9ZaKfi/9y+9GeUxtPqvRRpjRvknfHiLdRX41RSmpq4Pqy88ITT3rXLKaXX46NjXaef96sJbPO  
Tjop+t977y19953UsWPs9eyZpPbX9PNbvlx65RXp0kulWmYt2uuWNevwyCOlYcOkd96RmjWTateW  
6taV6tSRXnvNu7GRP6za/Uwya2rEConCC6O3ybnMxtKTT5YGDli+bIDfypVm3Zv/zx1+eN71S5m1  
6v9YurTUpYs0bpx3IxBW4RphFa4RVpElhFXEm3lmCoRDWIVrhFW4Nnjw4MD1ITnvPGGd81iKk5Y  
/f1jaWCL6HUerCF91EP69UNp5e/Sit+iv9o/7v5obLX3Za/z2U3REJvDvm7r+sXSmrnSI32iZ6za  
+7NnrdozVr9+PPoaq8vMcXjVn9J3g6lvS3C7uTzn7NYR15rLZktrZPIrRu8O0a8jRgxQqmFhdVH  
HvGuWUyvv54XHXPGRID/fzdVlk2ZlnXtGvv3TztGX2fVsmew2rNCr7IGSk+PXm7j5T77SE88IY03  
6/Dtt6Uzz4y8tEHM/dh59tno/Vj5w6r592qvvaSePaUPzdqeOFF64YVoZLUvR5BzPfvna806tGfW  
Wtu3m3W72qz1hdKLL0oVzf9//Pdr78++lIH9xmc9b+qVg7AK1wircl2wikQgrCleZLNTIBzCKlwj  
rMK1hIbVHdnSbx9J75hvQO1ZozZw2rFnnN5g5jpvrdzoxl7ezs59zemb/R+8vt+cF5UteHUhtVf  
h3sXeuaY/6/cVyHvvuzHibsY9LBbEhJWL7tM+sisrc2bvSt5evWKvZ4/rNo3nDr77Nga8PnX3/F  
vlyA/UZ20KDI67/G3FdhYdW+FED+s1Ete1/HHx973XPOyQurfvaM1ypVYq97//3ehfAjrMI1wipc  
l6wiEQiriDfz7BQlh7AK1wirc2xYXW79ONr0qv/iYZV+wZTLxwjrfg9evnulqz+rSUkrL77rndh  
PoWF1SVLor/q77/8iuiCl6f9lf6cs1pzpqiWgVQaq/Ys0/xh9dxzCashEVbhGmEVrhFWkQIEVcSb  
eXYKhENYhWuEVbiW2JcC2CFtWmGu81D0TFX7GqcDqkqvnSX95guh9jVVf3hFeunf0ddlfe4laeR1  
0ZclCFLcsNqvXPRzste11xvf37sQLiUkrNrXQA1SWFhds0a6557oO+/nXN6okdSjR97ZpvYMWpuy  
AP/6V95rnOZMIsJqpUqx1+1bwFnbsY6wCtclq3CNslpElKwi3syZuyAcwipcl6zCtbiH1fEDomei  
2oBpP95dRvor35vsbFoZfUmAdy+THm1gFnpGNMbaX/+3Y4OrjaOPNY5G10mPSFvWeDcOMPX56G16  
m7EvK2AD68wPvQs9y2dK710evU973zbs2pcfsB+Hd+PNqxyKe1i1bzJlbhczb73IXZhPZmbs9eyb  
SOV/86rPPpMuuij6uqxpabHXtzH19NOIM84oPKxefXXsZTVqBL8UgA2r+c+StfcfFFbta6m2by81  
aRJ9wyr/bTp14s2rfAircl2wCtclq0gEwirizTwrbclhrMI1wipci3tYta+jmr0tduyZqoHM39vr

25clyH8b+3f2sgJv62NfD3On2wfdzttfdr79FXc/2C1xD6v2sd1mHjf/BD7ehg2W/uvZN4fKYf88  
d67MF2vpl1+is2hR9Db557XXom9IZT7f3Hn+ee+ODHsd/37sFPQ52f36r+f/nPKLrG1z3/lvY/+u  
oPtPQoRVuEZYhWuEVSQCXRXxRlhFaIRVuEZYhWtxD6tAPnEPq/Fiv0F9/33ptNPYqk5ruqYY6SW  
LaXLL5cuuCD67v7+N65q3Fjmk46eUYq/BclqXCOSwjXCKhKBslp4I6wiNMIqXCOSwjXCKlz724bV  
HPbszzlpKeekjp0kE45RTqKOKf/5COPjoaV22Ys2eo/vGHdyP8nRBW4RphFa4RVpElhFXEG2EV  
oRFW4RphFa4RVuHa3z6sYo9HWIVrhFW4RlhFihBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhG  
WEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhG  
WIVrhFW4RlhFihBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhG  
WIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RlhFihBWEW+EVYRG  
WIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQ  
VhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RlhFihBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuE  
VbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuE  
VbhGWIVrhFW4RlhFihBWEW+EVYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuE  
VbhGWIVrhFW4RliFa4RVuEZYRSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RlhFihBWEW+E  
VYRGWIVrhFW4RliFa4RVuEZYhWuEVbhGWEUiEFYRb4RVhEZYhWuEVbhGWIVrhFW4RliFa4RVuEZY  
RSIQVhFvhFWERliFa4RVuEZYhWuEVbhGWIVrhFW4RlhFihBWEW+EVYRGWIVrhFW4RliFa4RVuEZY  
hWuEVbhGWEUiEFYRb4RVhJbVvpCwmlFPM34irClcwpG/y/IsLqEMlqwhkxfIRSUwoJq/cRVhHO  
tKnTVKNMIWE1k7CKcBbMW6AWdQoJq+cQVhEOYRWJYI9VQevLjj3G2WMdsCslqwhnh5RINnMYCtzq  
mW2G2ez1gN1FWIVT5vg02GxmRRW4vWE2jmPYbWbtjDBbqtn868q/PWI21hh2m1k708xWw2z+deXf  
rjMbawy7zaydBWZrYtb/uvJvrc2Wc11gdxBW4ZR3bLLHKrOiAjd7jLPHOo5j2BWEVYRWrLAKhEBY  
hWvFCqtACMUKq0AlxQqrQAJFDqvAbiKslhGKFVaBXUBYRWiEVbhGWIVrhFW4RliFa4RVuEZYhWuE  
VSQCXRXxRljFbpbk5c6Y+/-Rzjf9svM6ff37ugSj/Vn1zdQ0aN0hLf13q3Rlonjlz5uizzz7TpEmT  
1Llz58AnV3bS0tL0wAMPaP78+d4tgeJZtGiRPhv1mb759Bvd/MvNviPXzlvv73vrj8l/eLcEgu3Y  
Eft7Y6tWrdLnn3+ubz75Rg9Me0BldpTxrYrYresvXfXT+J+8W+bJf59IbvnXgw0MX331ISZ9PEmD  
vhmkylsr+1ZV7Hbhnxdqypgpyt6c7d06ijUGv/zrlTs7WxMnTtT4j8dr6PiharShkW9VxW4nLDhB  
Yz4foy2rt3i3jmKNoSjffvt5Fj26aef6pBDDgl8zm/noIMO0scff6z169d7t4xijaEo06dP15ef  
fKkvP/9SJy05yXfkit0arm+o18e8rpWzV3q3BlpGWEWxtWnTJvIFzZ4hWKaM+eYw54vc0zmHoYBt  
oZkWKZE37ChXrlzk+j179vTuEYh1/fXXR9Zlenp6oe+eHTR2Tdq1af980UUXefclxLLvG7XSNmy  
ZSOTu4Y6milsa52i0imlc49jxxxzjNatW+fdK5DnzTffjByPSpcuHfnBT+4aO8PMtshqCt6uS1Gp

IFlqVza6xvbee2/99ddf3r0CeWx8qFy5cmSd2K+XuWvsEDPLI6speHvEjL1N6ehtatSooSITpnj3  
CuT59ddf1ahRo8g6sV/3SpUqFV1j9c3MjKym4O1NM+Z6aSnRY59dn8OGDfPuFcizYsUKHXbYYTuv  
sWKOfQ6X873C//73P+9egTz2h0JnnnlmZl3Y7xHt87LcNWSPVQVt9hhXL9ovcr7G3nHHHd69AsEI  
q9iJ/yd+zz//vI444ojcg1DgF72iwup+ZrZr+m9/wgknRL4BzcFPGpOH/7F+9913ddJJjWwUkd0Z  
/+0PP/xwPfPMM96eWGPJxP9Yf/nllZr33HMD10juFBVW25jxruu//b777qv777/f2xNrLjn4H+tp  
06ZFvgYE1Lzz5lmCgur15vxrutfYw0bNITv3r1zz85hjSUP/2Ntf4uja9euuUE1/zqJzKfMCGur  
j5oJuG316tWVlZWlhQsXentjnSUL/+Nsz7C/7rrrVLdu3cB1EplhtX8Y6NGu3btNGNG3suDscas  
R85jvX37dvXt21fNmzfPXR5r7bFdnJzb24/2xloJEyZE9mWxxpKH/7F+/PHHI2c2518jMVNUWLXh  
Ou+6/tuffvrp+uijj7w9scaQh7CKQDaoBh1MAmcXwqp//Pf7wQcfeHtGsrC/6h00Ful5/vt98skn  
vT0jWfzwww+5oavINbYLYdU//vu18QvJxYao+vXr77QWAmcXwqp//Pd72WWXeXtGsti8eXPRP+DO  
mV0lq/7x36d9/UJ7lg+SywUXXBC4Hnaa3Qyr/vvcZ599ImcrlrlceeWVgeshnpNzv/YHUPasaySX  
AQMG7LQWCpxdCKv+8d/v2LFjvT0DhFX4+H/i8s4778QcRAqd3QyrOWN/ZfKLL77w9oxkMXHkRKWn  
+H6F0fG88vQr3p4LwQ8dS5TZs2erTp06gethp9nNsOqf/v37e3tGsrAvB+E/K6LQ2c2w6h9eSic5  
2TNkgtbDTrObYdU/F154obdXJJMOHToEroedZjfDqn+OOuoobd261dszSowinkPffvXtgevBxTSt  
0VSLflvk7bkQPO8vUexLQgStH8DZzbCaM5UqVYqcwAHkIKwixtq1a3XAAQfkHjSK9RPFEGHV1U8s  
mf/n6WCmsG1H3p9LmS3vxxvvv0WuCo08xM0L+L2SONWMeZkGGVY1lyT7Ee/5BhITWW3FOsxz9k  
WGWNJfcU6/EPGVZZYyV0hpopbPOefyiOX/MPorzvN9+7kH/JmaPnGlFY0KEVf8++GEKchBWEcOG  
VfuTZP/Bo8gJecYqUwKnh5k9dVtmxr4BSNC/iym5E4czVhmm0InDGasMU+jE4YxVhil04nDGKIMC  
530ze+pmP/egfxNTsifkGas5w0s0lQdhFTHsywHYF6+3v9ladPAIHM5YZfJPezOFbX/3M1abmgn6  
dzF75BTrOMMZq0yIKdbjzxmrtlgp1uPPGatMiCnW488Zq0zQcMYq8zeZYh9jQobVChUq6NVXX9WG  
DRu8ioJkR1hFoBdffDH3wFHkASoOb171+eefe3tGiVHE6xZN+GhC3lPCV5jpcxUT0nTXinl9M/0  
dB1fMV0nVUnTv80cZ/78z3lZ2rd0eVVMSS3g9nn3+9KjL3l7LgSvtVSi2HfRtk98lmuhqONYHN68  
6p577vH2jGSxceNGNW3adKe1EDhxePOqLl26eHtGMjnuuOMC18NOE4c3rzrjjDO8vSKZtG7dOnA9  
7DRxOGPVvi4176RdAhXxkn7U5abcNVDQ8/6wk3O/NUrX0JIZS7w9F4JIWKIMHDgwbY0U9ZxsN8Oq  
/37HjRvn7RkgrCKA/8mOfd0Q/8EkEkcSdq5c2dvT7H7Rcnmf6yzsrJi1kRa2VTvqlZFh+zdVG2O  
3V/PXdRYj5zZWPed2kiLb64o9U/TtjvLaVLXanq9VV09dW5jDTijiboEv48OO/gA1aheVWVK5d3f  
WWedlbs/1ljy8D/W/fr1i1ljgRPijNV//OMfvMNxEvKvMfvDyCKfxlc4Y7Vx48a8w3GSGzFiRO4P  
igqcEGesVq1alXc4TnLTpk1T3bp1A9dH7oQlq2XLitVbb73l7Q3JaOHChTHv5eFiHn30UW9vPO9P  
JjmP9aZNm/Tvf/87cG3ETlgzVm+88cblvizWGHlQVIEk+86d9sn2wQcfHHNQyf1pYxHfhdT+sd9s

Hn/88Zo8ebKys7O9PSDZLV62Qvfc/6Ae7XCUPrs8XXNuqKTPO9bSs+fV19DWtTSpW3XNv7GKNt+Z  
oR1907W9TznpnnTNvKqKXm9VT7Ovr6I/rq2qd1rX08xelfTJFeV16TH7aNJXX3h7AKQZM2aoTZs2  
Mcej3ONYEWG11GWxx7FatWrpgw8+4FeAEMOeJX311VfHrBU7kXVWRFgtdUPsGitTpowGDx4ceXme  
HDyBT07+x33RokW69957Y9aKncgaKyKslnosdo3Zeeihh7R06VLv3lljycr/uNsfFD7zzDM7rZXI  
GisirJZ6a+c1dvPNN2vevHnevbPGIK1bt05Dhw5V5cqVY9ZKkT+gNBN0nU6dOun333/37h2IBtaR  
lOfm/lZRzuQ+7y8irJZqELvOzjnnHP3444/evQM7l6yiSEFPgP73v//pP6f8R5eed6n2GrFX3oEo  
35a+PF3Hdj5W7wx6x7tllL1Pnlghx++//KRb2p+tiT1ravMd5fR263oa1aG2vsuqHvnztKxq0t3p  
2tG3XCSq2sk2f950R4YW3VhRE7vW0LC2dTts8jpaeFMVrbulrH7IrKDhj9+gMV9/r+3bt3t7QjLL  
f8wZM2aMzjv3PLU6tZWOGHiE78i187b/3fvr/hvv926Zh+MY/PKvh59++kktW7ZUy5Na6oR7T1Cp  
7eaJegFb84HNdVO3m7Rje+x9sMbgl3892NDar107XXjihTrtthOUTjbNt6pit/pv1le3tt20amle  
rLdYY/DLvx5soOjZs6fOO+E8ndPtHFVaUMm3qmK3mqNrqvUlTX7l9neraNY/ALWg+33Xabzj77  
bF100UWqWbNmTNTyT5UqVXTuuedq6tSp3i2jWGPwC1oPDz/8sM466Sy1uriVGk1o5DtyxW7l55bX  
yZefrJFDR3q3jGKNoTCEVYSWZTb/wci/1TPbDLMBhflj2kTde04DTe6Yog13VNAbrerqlZb19FWX  
WlrWu7K29SmvbC+o+kd907XlzyN7lhbX3Suo+W3VtHkzOr6PrOSNpr7efnm8/XMi69HzroGCjPY  
bP5jV/7tDbMBYYwwW6rZ/OvKvz1iNiCMaWarYTb/uvJv15kNCGOB2VqYzb+u/FtrswFh2NcuP/HE  
EwOjqh17mb0OEIY9VpkVFbjZY5w91gG7grCK0AirCOv3n77X7a2P09hu9tf9y2tMp5r6vEN1jelc  
U292bq6nOxysAe2OVN92x+uuDieqT8cTdXf74zWg/VEa2P5gvdC2qSZm1tSmOytoYteaGt2umv68  
tpLe7d9N73/6pbZt44xVFI6wCtclq3CNsArXCKtwjbCKRCCslt4lqwiNslqwZs/5S/f37atvbztE  
i28srxtOP0CXnHu+WnfKUpfMnsrskWWmh3pkZsZMpplumT3ULauXrujYRW0uOldPtd5f719SXi9f  
VF5Thz2nNZuyxW9uoCiEVbhGWIVrhFW4RliFa4RVJAJHfFGWEVoxQqrhC0U4q+583T/3X31/e0H  
asWt5XX7pf9Sp7aXqntmViSe2unevXuBYy/v3L2nunbrqle7HaBRbUvr1YszNHX4IK3dzOJDEcwS  
KVZYZSlhd5m1Q1iFU2aNEVbhGmEVrhFWkQiEVcQbYRWWhZXXKCvzCZ6depXqa8TNnrKJg77/1uu7q  
flGeuqipPryimkZ3q66X29ZX73ZnqE2Xq9Sx+1XqnNlNbpFWeB06XGNLmvXVZecd46ebLOXxnWt  
qGk9K+m5C2vr8V6tNHXKZK1Zz7u3o2CDnxsceAzLmTde4YxVhDPi4xFKTUkNXF92HulPWEU4076b  
phrpNQLXI53rehJWEc6CeQvUomGLwPVlp/X5hFWEQ1hFlthjVdD6smOPcfZYB+wKwipCy8oqJKzW  
q6cZMwirKNja9es16espuvvOPno260R92qO+PuzRTGOzaun7Hhka1628xnS1k1Hgfn6lvH6+qpyW  
3ZyhBy47WD3PbKEhF2do9MsPaebcZdq0ebO3NyDY4MFFhNU3CKsIZ8SIEUpNLSSsPkjYRTjTpk1T  
jRqFhNXrCKsIZ8GCBWrRopCw2pqwinAlq0gEe6wKWl927DHOHuuAXUFYRWiEVYQ1f948Pd7/Pr3U  
6996u+te6tP+JI25dh/p3jTp7nLK7mumT8Gje8pp1a0ZGpdVQ7e3O13XXHysXrmknH4Z8bw28b5V

KAbCKlwjrMI1wipcl6zCNclqEoGwingjrCI0wirCmjd3rh65714N7nm0XunSTN3bXqRhVx8i3V1W  
slG1b4a29ylX4OjudC3oXUVDuzXXTe1O1/Utj9ArLcvrp08Gaf1WXhoTRSOswjXCKlwjrMI1wipc  
l6wiEQiriDfCKkljrCKs33+dqVuu7KIh2xygJ69oobZt2ujNK/8h9UnVjmKF1TTN6V1DL3Q5SF1b  
/UdZ5x2pN1pV1Jcv9dNvcxYqOzvb2xMQjLAK1wircl2wCtclq3CNslpElKwi3girCI2wirBm/zZD  
fbq30pOX7aVBHffRZa0u0RtZh0t9ynphNTio5kwkrN5cQy90PICZI56h21odqUldK2vs87dr6k+/  
EVZRJMIqXCOSwjXCKlwjrMI1wioSgbCKeCOsljTCKsKa8+vPeijzAo3NrKPfbquvpcdpdG9mkn9  
UqOvs3qPfa3VQqZfWa24rZJG9Gikt7rvq8+zGmjeDZU08r/XaNTYb7V9Oy+0isIRVuEaYRWuEVbh  
GmEVrhFWkQIEVcQbYRWWhEVYR1m8/fK0+F+6rrzrYkJou+4ZV47PqqO/Fh6jdRWfokovOVZsLz1Kb  
i3ae1ubvzzv7bN3X5h9acENFbb+rXORM1zW3ZeJl21rphdfe09at27w9AcElq3CNsArXCKtwjbAK  
1wirSATCKuKNslrQCKsla/avv6h/1/M1qWcdqW9ZLb6xot7t0FC9LzlaPbt1UvceV6p7ZuZO9mj  
h7p3z9R5rTupX4cTtfyWCub26dp8W6r+urairPn3ien0+YSpnrKJlhFW4RliFa4RVuEZYhWuEVSQC  
YRXxRlhFaIRVhDXrzzm669Zb9OX1B0l3p0bOWF14Y4Y+7dZAN7Y7V5e366S2nbqrU7ce6pqZpS7d  
s9SxW5auMH/X7vIrdNUIp+idrnuZ22VI96Vr9lWpev6Cyhr3zv+0Yt0m7dixw9uTZ8mP0s9DpRnv  
RT/+MULast670KFt5ongkunSn59Lv34U3f+cMdKq2eayTd6VguT7/BF3hFW4RliFa4RVuEZYhWuE  
VSQCYRXxRlhFaIRVhLVuw2ZNnpNo+6/TF93KK3Nd5aT7slQdt8M7bg7Q+tuL69Z9gzULtX1evva  
Gtqhtj7vUk2/XVPJXNdcR4835vq6p4ymX19Xt3a6SJmTfT2kM+la6VbzOGvjxn78bFG0orfVQsd  
+ekt6dl/Sv3Mv+0us8++Zm4388wh0uQnotH11w+jf3+HN/bzm07QSwTCKlwjrMI1wipcl6zCNclq  
EoGwingz37UD4RBWES8Txilp+7oqRe6Ha0XLqmjweeW0VvnpWh021KamllRU3tU0vdZFVDVgV9  
16OCppi/G9+5vN6+MEVDziurI9s20VPtD9er/a/WtO+/19r1BTzx+v8lq5Mek241+7JR1X4cUE1a  
PtO70PP7x9HPyQZXO/a6v33kXZiDs1ddIKzCNclqXCOSwjXCKlwjrCIRCKuIN/NdOxAOYRWu3dP/  
4cD1FZnUDI3/Zqp3zWlqTljdKs2tXxx9qYDh3aQn9o6cDRuJor3N2Oj5aAPpjQukb/8nLTfrfEe+  
13Kd/7X0UWb07NMbzdxdKnpGqv1o923/7k4zD9WK3u/NZuzlfe31vOteb8bu99OrpMU/eHeMeCOs  
wjXCKlwjrMI1wipcl6wiEQiriDfzHTsQDmEVrvXr1y9wfdnJyMjQmDFjvGsWU3HC6pSnpPvLR88c  
fXlfaUxfack0afsmacvaaEgd318aUFW6zVzn4TrSp1dLi77z7sCwoTV7W/TPX9wZvV7OGav9q0Rf  
X9Xa6r2+67TB0c/J/1IA378YvWz75mjshROEVbhGWIVrhFW4RliFa4RVJAjHfFmvmshHwiGswrWE  
hIX7Wqc/viK9ckY0guaEzpvMXGfmWu+jHXvGqT171Y4929SefTrlmeJ95Df2vnwvBVDVflcyxbvQ  
8/Ob0f3ZmGv3aa/La6wmBGEVrhFW4RphFa4RVuEaYRWJQFhFvJnv2oFwCKtwLaFhdesGaepz0uB/  
R8OmDaZvXmSe6a2IXl4sAa+DSlj9WyOswjXCKlwjrMI1wipcl6wiEQiriDfzXTsQDmEVriU0rO7Y  
Ef1Vf/ur/zlvtW/kjT4BOM34Xmvo7r0Z+nLPtKj9aV7UqOvwTrSfNPqfykAv+KE1Z/ejF6eE1bt

52cjL5wjrMI1wipcl6zCNclqXCOslhElq4g38107EA5hFa7FPax+1CP6K/32V/jtRxs5l//qXejZ  
vFr66yvpkyul/x0q3V8pGkRv8Ma+Xqq93QvHSqOul37/pPDXQLWvsWpfPsCeAWs/2jewsm9u5bf0  
J+nTXtLAFtHLbVy1n9+9adJnN0YvhxOEVbhGWIVrhFW4RliFa4RVJAJhFfFmvmMHwiGswrX4htWA  
X9PPVdhlyfx/7BO7grAK1wircl2wCtclq3CNslpElKwi3girCI2wCtfifsYqkA9hFa4RVuEaYRWu  
EVbhGmEViUBYRbwRVhEaYRWuEVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWERphFa4RVuEaYRWu  
EVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbhGmEViUBYRbwRVhEaYRWu  
EVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWERphFa4RVuEaYRWuEVbhGmEVrhFW4RphFYIAWEW8  
EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbhGmEViUBYRbwRVhEaYRWuEVbhGmEVrhFW4RphFa4RVuEa  
YRWJQFhFvBFWERphFa4RVuEaYRWuEVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4RVuEa  
YRWuEVbhGmEViUBYRbwRVhEaYRWuEVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWERphFa4RVuEa  
YRWuEVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbhGmEViUBYRbwRVhEa  
YRWuEVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWERphFa4RVuEaYRWuEVbhGmEVrhFW4RphFYIA  
WEW8EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbhGmEViUBYRbwRVhEaYRWuEVbhGmEVrhFW4RphFa4R  
VuEaYRWJQFhFvBFWERphFa4RVuEaYRWuEVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4R  
VuEaYRWuEVbhGmEViUBYRbwRVhEaYRWuEVbhGmEVrhFW4RphFa4RVuEaYRWJQFhFvBFWERphFa4R  
VuEaYRWuEVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbhGmEViUBYRbwR  
VhEaYRWu9burkLCakqExnxNWEQ5hFa6N+HiEUIMKCasDCKslh7AK1xbMX6AWjQoJqxcQVhFOJKwe  
WUhYNZcRVhEWYRXxRIhFaIRVOLVD6mc2s6lCtwyzjTGbvR6wuwircMocn0aYLdVsZkUFbo+YjeMY  
wiCswilzfFpgthZmMysqcGtttpzrArvMrJuNZjvRbP515d/sZfY6rDGEQVhFvBFWERphFa4VK6wC  
IRBW4VqxwioQAmEVrhU7rAK7qdhfFQiBslp4l6wiNMIqXCOswjXCKlwjrMI1wipcl6zCNclqEoGw  
ingjrKJIO3bs/LsWo0eP1oABA/TEE0/o2GOPDTwo2alUqZKuueYaff/9994tgWD519k333yjAfcN  
OMD7B+qsCWflPqHKv5XdVladX+qscR+M826ZJ2jtlnnlXw/2hz4PPPCAnnrqKbVp0ybwGJYzl1xy  
iT755BPvlnlYY/DLvx7mz58feVOqJ+9+Ut3f7a7S2aV9R6/Y7YwRZ+itZ9/ybpmHNQa//Oth9erV  
GjhwoP773//qxhtvVIUKFQKPYXaOPvpovfDCC9q8ebN36yjWGPzyr4dt27bp+eef16N9HIWfZ/uo  
9sraviNX7Hbg9AP130f/q5ULVnq3jmKNwS9oPdgyFD/c52E98MQDaj6/uW9VxW5N5jTRfY/ep3kz  
5nm3jGKNwS9oPQwfpJzyvP+xxx7TYYcdFvh10k7NmjV1yy237HRyGGsMhSGsolhWrVqlU089Neag  
U6pUqZj/LmjyX69Vq1batGmTd89A1JYtW9SuXbuYtVlqxVs7t5spaNtorfv2DV2wgknaPny5d49  
A3l69+4ds1aKO/mPY/vss49mzZrl3SuQxwYu/1rJnTPNbDNTwFbqhtg1VrVqVU2ePNm7VyDP0KFD  
Y9aKnel8Jwu6zkcffeTdK5Dnq6++Uvny5WPWSuQ5WX3z55lmCtreMuO7jZ1BgwZ59wrk+eWXX9So

UaOYtRJZYxnmz1+ZKWgba8Zex3e7e+65x7tXIM/ChQv1z3/+M2at7G6/yMzM1Pbt2717BnZGWMVO  
/D+Nufnmwv9tbiwU7duXd1///3envgpUDLxP9YPP/yw6tevH7hGcqelsJrybzMBt6tevbquvfZa  
b0+ssWTif6xfeeUVNW/ePHCNhJ2KFSvq8ssvj5zRY7HGkof/sR45cqQOPPDawDWSO0WE1ZTrzQTc  
rly5cjrrrLO0bNmyyL7sflInych/ONvf/DnqqKNUpkyZwHUSZsqWLavjjjsu5uwc1lhy8D/O9gx7  
exJFenp64DqJTFFh9U0zAbcrXbq0Dj30UI0dO9bbG2ssmeQ81hs2blj8BIBhZ9YXGVbtZfnCas7Y  
H3q//fbbkX1ZrLHk4X+sbQStUqVK4BoJO/aHAFY33XKwxcDslpA9ld+cg4gxf3Jzq6O/37ff/99  
b89IFp9++mngWgic3Qyr/vt98sknvT0jWfzwww9KTU3daS3Ec/z3a38QheRiz4awryWefy0Ezm6G  
Vf/92pesQHKxv7bvP+PGxbHMf5/2Nz6ys7O9vSNZnH/++YHrYafZzbDqv8+9995bK1as8PaMZNGz  
Z8/A9bDThAirOfdrX4ru119/9faMZGfPjD/Woj3+O/X/4MigLCahLLNtr2AzV5mDXl1SN5BplzD  
MfdfJq2Mhn8yPLJf/+dS4Lad2dPHGjVqVOFnRfhnN8Oqf5599tnlfu03jEGfE7OHTbaZQjbrx59/  
VO16taNrIoj4E6/x1life/pE9mv5P5fAzX7+Qf8uZo8Zy765wf7775+7Bgqd3Qyr/unatWtkv1bQ  
58TsgVPIZm3YvEEnn3ZydA0EHX/iOWYf55x/jrZmb43s2/+5BG4cx/b4ydG2bduYY02Bs5th1T/2  
BwU5L9cU9DkxJWty2NeADloPO02IsJozRs3zg2rQZ8TswfODjMFbDvMZj32xGN56yDoa1y8xtx/  
paqVNH7S+Mh+/Z9L0JbTV1CyEVaT0C1mM4eEgrcdvj8ncivOfn8xU81MzkGTSY6JQ1hIstjYSFY  
McN3WSmz5f2Xmy13H8U5jm01w5otUVOsMyNChlVXZ18w/49T18wsM8XYEnocK+52k5mgfxyzR06x  
jjEhwyrHseSeYj3+lcMqa6yEzhNmCtv+xx3iULOtMRtKNsJqErrDbP7/s+9R2wwz9cwEHXCZkjuE  
VSb/nG/m/+tJVNjNhtUzzAT9u5iSO3E4Y5UpYdPUzJ9m9tTtNjNB/y6m5E4czlhmElndmesMiVw  
njazh27/MhthteQjrCahW83m/z/7HrXZM1Zrmgk64DlIdwirTP4528yeHFZPNhP072JK7hBWmfzT  
wEwxz1j9W269zQT9u5iSO4RVxvUQVpmgedLMHrr9w2yE1ZKPsJpkcl6DpCivPvNq7oGsVlqbX6nw  
3++ld0Z4ey4Eb7pXoowZMyZvLRT1aztxePOqQYMGExTGiVLlceH3739XWkpadC0k4Dh257V3ensu  
BMexEmXNmjVq2LBhdC0UdRzbzbDqv9/27dt7e0aJUthxYZN01P5H5a0HB8cy/32efOTJ9sX4i8ax  
rES5+OKL89ZDYceyOLx5VYsWLBrlYxZvz0gW1157beB62GlChNWc+7XvCD9//nxvzygxivi689g9  
ea+xmojn/d9++a235+lpbofBnomwip3s2JH3f/prrrlGlStXjjmgxGuqV6+uPn3y3uzFv1+UbP7H  
+v7771eNGjUC10ju7GZYrVixorKysrw9scaSif+xfvHFF1W/fv3ANRJ2ypUrp0suuURbt0bf7IU1  
ljz8j/XHH3+svfbaK3CN5M5uhtWyZcvqlFNO0eLFiyP7Yo0ID/9jPWxKFB1yyCEqXbp04DoJM/Y+  
7Rsk/fTTT97eWGFJwv84//XXXzruuOOumpoauE4iE+KMVRtUv/jiC29vrLFkkvNYr1u3Tuedd17h  
b14blqzaN6164403lvuyWGPJw/9Yd+zYUeXllw9cl2Gndu3aeuyxx7w9scaQh7CKYlm6dKlOOumk  
mANLoT9t9E3+61100UWRs3wAv/Xr1+uyyy6LWSu5PxUslqyWOil2jR1zzDFatGiRd89A1Lzt23T9  
9dfHrJXiTv7jWPPmzTVjxgzvnoE8Dz/8cMxayZ0iwmqpG2LXmP2h5oQJE7x7BfK89tprMWvFTnGe



kwVd57333vPuFcgzevToneJX5DIZEWG11Fs7r7Gnn37au1cgz7Rp03b6oXdkjRUVVseayRdW77zz  
TmVn887riGV/WHTYYYfFrJXd7RedO3fWpk2bvHsGdkZYRZGCFhLzySef6K677tIDDzygf/3rXzEH  
Hv/YMwYzMzM1efJk75ZAsPzrzAaFu26/Sw/d8ZBOG3Na3hOqfFvqtlS1fbatRr812rtlHn6KCL/8  
62H69Onq27evHnrolbVs2TLwGJYz9gyL999/37tIHtYY/PKvh7lz56pfv3564OYH1PGNjiqdXdp3  
9lrdThp2kob8d4h3yzysMfjIXw+rVq2KHMP69++vXr16FXqWjj0rdeDAgTt9c8gag1/+9WB/KGnX  
Tb8b+6n3f3ur5vKaviNX7Nbi+xYacN8ALZ+33Lt1FGsMfkHr4aWXXtLdN92tvG/0VdO/mvpWVezW  
cFZD3XHfHZozfy53yyjWGPYc1sO7774b+W1Z+9uSBx98cODXSTv2t2rtS1fY7xP8WGMoDGEVodlf  
tQ46KNmpV68eZ3UhtH5mMysqcMsw2xizAWEMHjw48BiWM/5fLQN2xwizpZrNrKjA7RGzAWHYM8AK  
e2md6667zrsmSsHsWmK2F2cyKCtxamw0IY6PZTjSbf135N3uZvQ4QRuvWrQO/TtqxL1uyYMEC75pA  
8RBWERphFa4RVuEaYRWuEVbhGmEVrhFW4RphFYIAWEW8EVYRGmEVrhFW4RphFa4RVuEaYRWuEVbh  
GmEViUBYRbwRVhEaYRWu9dvSTylrzZoKmlx1GRqznbCKcAircG3EthFKXZcaeByz88gWwirClazC  
tQU7FqjF+hZKWWPWVP7jmPm71pslqwgnEIY3nFjgGrOXEVYRfmEV8UZYRWiEVbjWr2+/wPVIJ6NO  
hsZ8SVhFOIRVuDbikxFKTUKNXF92HnmQslpwCKtwbcH8BWRrEXg+rLT+mLCKsLZuHGjTjz6xMD1  
ZcdeZq8DhEFYRbwRVhEaYRWu2XfVDlpfdjlyMjRmDGEV4RBW4dqIESOUmIpIWH2EsIpwCKtwzcYG  
Gx2C1pcdGyuAMCJh9cRCwqq5jLCKsAiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwjrCIRCKul  
N8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwjrMI1wipc  
l6wiEQiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipc  
l6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipc  
l6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhElq4g3wipC  
l6zCNclqXCoswjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwjrCIR  
CKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwjrMI1  
wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1  
wipcl6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1  
wipcl6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhElq4g3  
wipCl6zCNclqXCoswjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwj  
rCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwj  
rMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwj  
rMI1wipcl6zCNclqXCoslhElq4g3wipCl6zCNclqXCoswjXCKlwjrMI1wipcl6wiEQiriDfCKklj  
rMI1wipcl6zCNclqXCoswjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCoslhEl

q4g3wipCl6zCNclqXCOSwjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCOSwjXC  
KlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCOSlhElq4g3wipCl6zCNclqXCOSwjXC  
KlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCOSwjXCKlwjrCIRCKuIN8lqQiOswjXC  
KlwjrMI1wipcl6zCNclqXCOSlhElq4g3wipCl6zCNclqXCOSwjXCKlwjrMI1wipcl6wiEQiriDfC  
KkljrMI1wipcl6zCNclqXCOSwjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCOS  
lhElq4g3wipCl6zCNclqXCOSwjXCKlwjrMI1wipcl6wiEQiriDfCKkljrMI1wipcl6zCNclqXCOS  
wjXCKlwjrCIRCKuIN8lqQiOswjXCKlwjrMI1wipcl6zCNclqXCOSlhElq4g3wipCl6zCNclqXCOS  
wjXCKlwjrMI1wipcl6wiEQiriDfCKkLLusoLq6lm0nxj/q5ek3qaMZOwinAlq3Bt8BBfWM13HEsp  
k6l33iKslpwRo0YotbwXVv1rrJQZ83ePPEZRTiEVbhGWIVrkbB6ihdW/V8r7Zi/s5cRVhFW68u9  
sFp25zXW4iDCKnYdYRWWhZZnNHIYct3pmm2E27fCuDOwGwiqcMsenwWYzK6rA7Q2zcRzDbjNrZ4TZ  
Us3mX1f+7RGzscYQBmEVrhFW4dpGs51oNrOiAjd7mb0OEEZrs/nXlX9rYbYFZuM5GXIFYRWWhFSus  
AiEQVuFasclqEEKxwioQAmEVrhFW4RphFYlQrLAK7ALCKkljrMI1wipcl6zCNclqXCOSwjXCKlwj  
rCIRCKuIN8lqirRjx87nwb/44ou6+NyL1bF1R+07et/ca1H+rdyKcjr1yIM1bMgw75Z5gu4XySv/  
evjwww/VsmVLdezYUYcddljgE3g7ZcqU0THHHK0XNrJu2Ue1hj88q+H8ePH69JWl6rDeR107HPH  
+o5cO2+H9T9Mj97+qHfLPKwx+OVfD/bNG6+44gq1O7udTn3wVJXKLvVbVbFbi2db6M6r7pSyvRt7  
WGPwy78elixZom7duqlt27Y699xzlZ6eHvi10k6TJk109dVXa82aNd6to1hj8Mu/HjZv3qzrr78+  
Ek3t87LKIssHri87devWjTxv++uvv7xbR7HG4Be0Hu6++261+k8rte3cVrVn1vZ9dYzdvq1YTa3a  
t9IP43/wbhFGonf0Hp44okn1PLslmp/eXs1/bqpb1XFbhXmVdDZXc/WF+9/4d0yijWGwhBWUaTt  
27drypQpOvzww2OePJVKMd8g2j8/HT0IBW4LzfX2967nzUknaTp06dzcEKMX375RWeeeWbMWilV  
KnbtFDT5r3fggQfq66+/1rZt27x7B6TZs2erXbt2MWsl9zjW0UwhW6nLYtdYnTp1NHLkyMg3nECO  
RYsW6aabbopZK3Yi6+xM8+dtZgrYSt0Qu8bKli2rN998U+vXr/fuHZBWrFihBx98MGat2CnO18ug  
6zz55JNavXq1d++AtG7dusgPq/Ovld1dY3feeaeWLI3q3Tsgbdq0ScOGDVPVqIVj1krka2WG+fNX  
ZgrYSo31ruO7XWZmpubNm+fdOyBt3bpVY8eO1V577RWzVnKf979ppqBtprleg9hj2YUXXqjffvN  
u3dgZ4RV7MQfPFu1ahVzUAmclsJqyn5mgm5nxn4hzEFoTR7+x9qePRO0Nu1559/vrcn1lgy8T/W  
DzzwQODaiJkiwmpKGzNBtzNz9NFH54YJ1ljy8D/WL7/8cqFnCkamiLCacr2ZoNuZ2XvvvTVr1ixv  
b0gW/jU2evRoVatWLXB9xGNq166tiRMnentDMvrpp5/UqFGjwPURjylfvrzeffddb298vUwmOY+1  
PcO+sN9Ci0wRYTVyWb6wmjP2t9jsD4tysMaSR85jbYPqaaedFrg+YqalsJpS30zQ7czceutkX1Z  
rDHkIKwi0AsvvJB78CjyJ9S7GVb99/vBBx94e0ayGDFiROBaiOf473fgwIHenpEsfvjhB5UrV26n  
tRA4uxlW/fd7yy23eHtGsli4cKEaNmy401oInN0Mq/77vfzyy709I1nYs+KPOuqowPUQr/Hfp/2t

ouzs fK9HgRLPno0VtB5cjH2NVnvmNZJlR169ctdAoWssRFjNuV97JixnFyYf/29zFHkc282w6r/f  
cePGeXsGCKvIx/7UZcOGDXr44YdjDiKFTogzVu3Yny7a12xduXKl/vzzT2ZPnz/MLDNTyLbSbk9O  
eFVpLdKUUsesg2YOp0F0+r/ZXyvMNTts/s9lp227mflm7L8j6N/H7BFjv2mzv6pf2Bu5xEyIM1Zz  
5tprr438uqN9bbmgz4kpWbNs2TJNnjw5cjZp0HrYaUKcsZozbdq0ibx5zPz58wM/J2YPGvs1Zq6Z  
bWYK2BabbeaWmTqm7TFKqWbWQNDXuHiO2ccpXU7RH9l/aJHZ/J9L4LbSjP13zDIT9G9k/n/HPjZz  
zGw2U8Bm36BlrtkuuPaC6Bprmm9NxHvMPg4+/2BNWzNNS83m/1wCtzVm7Ppije2xM3fuXC1evFjd  
u3cP/Lq204QlqzITv359ffnll5HngkGfE1OyZs6cOVq+fLnuuuuuwPUQOCHOWLVTouIFDR06NHKG  
LGARVhFj7dq1OvLIlWMPIAVOyLDKIMDpbmZP3ZaaOdBM0L+LKbkTh7DKMIVOHMlqU8KmkZlZzbU  
7VYzQf8u5u8z5c18bWZP3Z40E/TvYkruxCGsMkyREzKs5oz9gTdgEVYRw4ZV+8Y/QQeOAoewyuSf  
Hmb21G2ZmUPMBP27mJI7hFXG9RBWmfxfjzw7808yeut1mJufxfx9prKZPTmsPmUm6N/FINwhrDKJ  
mBBh1f9yAPZIVAClsipc/hdffvtt2MOIIVOHMJqxYoVVb169cgbMzB78FQxk2LmejOFbNXNVnFb  
RaWsMo//ysRM+S3li/v1fx6B23lzx5mx/46gfyOzR4w9nISuXLno11jKmTiEVft6rkGfC1Myx66x  
KIWqRF7OJmg97DRxCKv2DbKCPHdmD5xSZg41M8dMIVtVs6WuS1XKCrMGAr62xXXMPsquKxvZp/9z  
KHC710wZM5XMBP0bmf/fSTXTyMy3ZorY0tanJWyNIVlbRIV2VPHtvZDteTPlzJQ3E/RvZPaYyXnN  
+ylnDmG1dOnSkeeAfG+ZPGMf64yMjMD1EDghz1itVKmSvv/+e6+cAIRVFOD555/PPXAUGSbi8OZV  
H330kbnJlvRH4/OWwspXyxfuzj++33m8We8PSNZ/PLLLyptmx0LRR1HlvDm1fddttt3p6RLOxr  
ejVo0GCntRA4uxlW/ffLm1clnx2bdujlQ/JeosnF10v/fZ50zEkS712VdC482/fmVY7X2D5N9tG6  
Feu8PSNZXHXVVXnrobCvlyHCas792qg6e/Zsb89IFv73iCnyOVkc3rxq7Nix3p4BwioC+M9cte+k  
fvDBBwceTHJnF8Kq//ZHHHGEhgwZ4u0pdr8o2fyP9euvv66jjz46cl3szvhvf8ABB+jRRx/19sQa  
Syb+x9q+idUpp5wSuEZyZxfCqv/2TZs21Z133untiTWWTpyP9ZQpU3T++ecrNTU1d23sNLsQVv1r  
rE6dOpFvSNeti4Ylu1/WWXLwP85//PGH2rZtG/kNn6B1sqvjv60987pDhw6aN2+etzeOZcnC/zjb  
N/rJzMxUzZo1A9dJmLFn27ds2VI//vijtzwWDLJey3bdum3r17q1GjRlrY6c1tpth1d7PmWee  
qTFjxkT2ZbHGkof/se7fv7/23XffmLXhXyuR2YWw6r/98ccfr/fee8/bE2sMeQirKLZLLrkckCx  
3zjGHKCKEVZLp5TO/XVJ+66QQJCrr746skZKly4dmdw1Voyxazlnapx33nnePQKxBgwYkLtmYtZY  
UWG1tVljKWaNIYmuMfsmf6tXr/buFcjz6quv5q6rmJcJGZYT50VXWNNmjSjvNstkN/nn3+e+yuP  
xX4pCt/k3MYG1UmTJnn3CuT5+eefVbdu3Zj1siuTcxv7ddYfiYAcS5Ys0UEHHRRZJzhfWxYzrNrv  
LXPWmzORCMjPvmP/qaeuvmMas1OMsGrXWM5tbrnIFu9egSDS/wFLUcL7Hu8BPgAAAABJR5ErkJg  
gg==

width=500

height=300

>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Running valid\_states = False

file named agentC.py</p>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;  [2]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class="highlight hl-ipython2"><pre><span class="kn">import</span> <span class="nn">random</span>

```
<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>
<span class="n">Agent</span><span class="p">,</span></span> <span class="n">Environment</span>
```

```
<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>
<span class="n">RoutePlanner</span>
```

```
<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>
<span class="n">Simulator</span>
```

```
<span class="k">class</span> <span class="nc">LearningAgent</span><span class="p">(</span><span class="n">Agent</span><span class="p">):</span></span>
```

```
<span class="sd">"""&quot;&quot;&quot;An agent that learns to drive in the smartcab
world.&quot;&quot;&quot;</span>
```

```
<span class="k">def</span> <span class="nf">__init__</span><span class="p">(</span><span class="n">self</span><span class="p">,</span></span> <span class="n">env</span><span class="p">)</span></span>
class="bp">self</span><span class="p">,</span></span> <span class="n">env</span><span class="p">)</span></span>
```

```
<span class="nb">super</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">,</span></span> <span class="bp">self</span><span class="p">)</span></span>
class="p">)</span><span class="o">.</span></span><span class="n">__init__</span><span class="p">(</span><span class="n">env</span><span class="p">)</span></span>
class="p">)</span><span class="n">env</span><span class="p">)</span></span> <span class="c"># sets
self.env = env, state = None, next_waypoint = None, and a default color</span>
```

```
<span class="bp">self</span><span class="o">.</span></span><span class="n">color</span> <span class="n">self</span><span class="o">.</span></span>
class="o">=</span><span class="s">'red'</span></span> <span class="c"># override color</span>
```

```
<span class="bp">self</span><span class="o">.</span></span><span class="n">planner</span> <span class="n">self</span><span class="o">.</span></span>
class="o">=</span><span class="n">RoutePlanner</span><span class="p">(</span><span class="n">self</span><span class="o">.</span></span>
class="bp">self</span><span class="o">.</span></span><span class="n">env</span><span class="p">)</span></span>
<span class="bp">self</span><span class="p">)</span></span> <span class="c"># simple route planner to get
next_waypoint</span>
```

```
<span class="c"># TODO: Initialize any additional variables here</span>
```

```
<span class="k">def</span> <span class="nf">reset</span><span class="p">(</span><span class="n">self</span><span class="p">,</span></span> <span class="n">destination</span><span class="p">=</span></span>
class="o">=</span><span class="bp">None</span><span class="p">)</span></span>
```

```
<span class="bp">self</span><span class="o">.</span></span><span class="n">planner</span><span class="o">.</span></span>
class="o">.</span><span class="n">route_to</span><span class="p">(</span><span class="n">self</span><span class="o">.</span></span>
class="n">destination</span><span class="p">)</span></span>
```

```
<span class="c"># TODO: Prepare for a new trip; reset any variables here, if required</span>
```

```
<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span></pre>
```

```
<span class="c"># Gather inputs</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span><span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span><span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span><span class="o">=</span> <span class="n">planner</span><span class="o">.</span><span class="n">next_waypoint</span><span class="o">=</span> <span class="p">)</span><span class="c"># from route planner, also displayed by simulator</span></pre>
```

```
<span class="n">inputs</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span><span class="o">=</span></pre>
```

```
<span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span><span class="o">=</span></pre>
```

```
<span class="c"># TODO: Update state</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="p">(</span><span class="n">inputs</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span><span class="p">,</span> <span class="n">deadline</span><span class="p">)</span></pre>
```

```
<span class="n">valid_states</span> <span class="o">=</span> <span class="bp">False</span></pre>
```

```
<span class="c"># TODO: Select action according to your policy</span></pre>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="s">'</span>forward<span class="s">'</span></pre>
```

```
<span class="c"># Execute action and get reward</span></pre>
```

```
<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span><span class="o">=</span></pre>
```

```
<span class="c"># TODO: Learn policy based on state, action, reward</span>
```

```
<span class="k">print</span> <span class="s">&quot;LearningAgent.update(): deadline = {}, inputs  
= {}, action = {}, reward = {}&quot;</span><span class="o">.</span><span class="n">format</span><span class="p">(</span><span class="n">deadline</span><span class="p">,</span><span class="n">inputs</span><span class="p">,</span><span class="n">action</span><span class="p">,</span><span class="n">reward</span><span class="p">)</span> <span class="c"># [debug]</span>
```

```
<span class="k">def</span> <span class="nf">run</span><span class="p">():</span>
```

```
<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of  
trials.&quot;&quot;&quot;</span>
```

```
<span class="c"># Set up environment and agent</span>
```

```
<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span class="p">()</span> <span class="c"># create environment (also adds some dummy traffic)</span>
```

```
<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span class="o">.</span><span class="n">create_agent</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">)</span> <span class="c"># create agent</span>
```

```
<span class="n">e</span><span class="o">.</span><span class="n">set_primary_agent</span><span class="p">(</span><span class="n">a</span><span class="p">,</span><span class="n">enforce_deadline</span><span class="o">=</span><span class="bp">False</span><span class="p">)</span> <span class="c"># specify agent to track</span>
```

```
<span class="c"># NOTE: You can set enforce_deadline=False while debugging to allow longer  
trials</span>
```

```
<span class="c"># Now simulate it</span>
```

```
<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span class="p">(</span><span class="n">e</span><span class="p">,</span><span class="n">update_delay</span><span class="o">=</span><span class="mf">0.5</span><span class="p">,</span><span class="n">display</span><span class="o">=</span><span class="bp">False</span><span class="p">)</span> <span class="c"># create simulator (uses pygame  
when display=True, if available)</span>
```

<span class="c"># NOTE: To speed up simulation, reduce update\_delay and/or set display=False</span>

<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span class="p">(</span><span class="n">n\_trials</span><span class="o">=</span><span class="mi">5</span><span class="p">)</span><span class="c"># run for a specified number of trials</span>

<span class="c"># NOTE: To quit midway, press Esc or close pygame window, or hit Ctrl+C on the command-line</span>

<span class="k">if</span> <span class="n">\_\_name\_\_</span> <span class="o">==</span> <span class="s">'\_\_main\_\_'</span><span class="p">:</span>

<span class="n">run</span><span class="p">()</span>

</pre></div>

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt"></div>

<div class="output\_subarea output\_stream output\_stdout output\_text">

<pre>Simulator.run(): Trial 0

Environment.reset(): Trial set up with start = (5, 2), destination = (8, 6), deadline = 35

RoutePlanner.route\_to(): destination = (8, 6)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -10, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': 'forward', 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -37, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5



LearningAgent.update(): deadline = -62, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -63, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -64, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -98, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -99, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (7, 3), destination = (4, 6), deadline = 30

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'right'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5



LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -73, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (7, 3), destination = (1, 6), deadline = 45

RoutePlanner.route\_to(): destination = (1, 6)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None, 'forward': True}, action = forward, reward = 2.0



LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'left'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -22, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -41, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -42, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': 'left', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': 'forward', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -64, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0



LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -82, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (3, 1), destination = (6, 3), deadline = 25

RoutePlanner.route\_to(): destination = (6, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -1, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -26, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -28, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -31, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'red', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': 'left', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -51, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -57, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -71, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -86, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -90, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -91, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'green', 'oncoming': None, 'right': 'right', 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (8, 6), destination = (2, 4), deadline = 40

RoutePlanner.route\_to(): destination = (2, 4)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red',  
'oncoming': None, 'right': None, 'left': None}, action = forward,  
reward = -1.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0



LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': 'right', 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -11, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -14, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -21, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -22, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -23, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -24, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -25, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -26, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -27, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -28, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -29, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -30, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -31, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -32, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -33, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -34, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -35, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -36, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -37, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -38, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -39, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -40, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -41, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -42, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -43, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -44, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -45, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -46, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -47, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -48, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -49, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -50, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -51, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -52, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -53, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -54, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -55, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -56, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -57, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -58, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -59, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -60, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -61, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -62, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -63, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -64, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -65, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -66, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -67, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -68, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -69, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -70, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -71, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -72, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -73, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -74, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -75, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -76, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -77, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -78, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -79, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -80, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -81, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -82, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -83, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -84, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -85, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -86, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -87, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -88, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = -89, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = -90, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -91, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -92, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -93, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -94, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': 'right'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -95, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = -96, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -97, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -98, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -99, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = -100, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

Environment.step(): Primary agent hit hard time limit (-100)! Trial aborted.

</pre>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Each action taken by the smartcab will produce a reward which depends on the state of the environment. The Q-Learning driving agent will need to consider these rewards when updating the Q-values. Once implemented, set the simulation deadline enforcement `enforce_deadline` to `True`. Run the simulation and observe how the smartcab moves about the environment in each trial.</p>

<p>The formulas for updating Q-values can be found in this video.</p>

<p>QUESTION: What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?</p>

<p>I found the car did not correctly show where the light was positioned and randomly crashed into oncoming traffic at a higher rate than previously observed. I am not sure why this behavior has occurred by changing the `enforce_deadline` to `True`. It possibly maybe due to the program only using the deadline for reward justification and not other variables such as arriving at destination, no crashes etc.</p>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[1]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class="highlight hl-ipython2"><pre><span class="c">#cell for code above file agentD.py  
enforce\_deadline=True</span>

```

import random

from environment import Agent, Environment
from planner import RoutePlanner
from simulator import Simulator

```

```

class LearningAgent(Agent):

```

```

    """An agent that learns to drive in the smartcab
    world."""

```

```

    def __init__(self, env):

```

```

        super(LearningAgent, self).__init__(
            env)
        self.env = env
        self.state = None
        self.next_waypoint = None
        self.color = 'red'

```

```

        self.color = 'red'

```

```

        self.planner = RoutePlanner(
            self.env)
        self.planner.route_to(
            self.destination)

```

```

        # TODO: Initialize any additional variables here

```

```

    def reset(self):
        self.destination = None

```

```

        self.planner.route_to(
            self.destination)

```

<span class="c"># TODO: Prepare for a new trip; reset any variables here, if required</span>

<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span></span>

<span class="c"># Gather inputs</span>

<span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span>  
<span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">(</span>  
<span class="c"># from route planner, also displayed by simulator</span>

<span class="n">inputs</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="bp">self</span><span class="p">)</span></span>

<span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get\_deadline</span><span class="p">(</span><span class="bp">self</span><span class="bp">self</span><span class="p">)</span></span>

<span class="c"># TODO: Update state</span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="p">(</span><span class="n">inputs</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">,</span> <span class="n">deadline</span><span class="p">)</span></span>

<span class="c"># TODO: Select action according to your policy</span>

<span class="n">action</span> <span class="o">=</span> <span class="s">"&#39;forward&#39;</span>

<span class="c"># Execute action and get reward</span>

<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span></span>

```
<span class="c"># TODO: Learn policy based on state, action, reward</span>
```

```
<span class="k">print</span> <span class="s">&quot;LearningAgent.update(): deadline = {}, inputs
= {}, action = {}, reward = {}&quot;</span><span class="o">.</span><span><span
class="n">format</span><span class="p">{</span><span class="n">deadline</span><span
class="p">,</span><span class="n">inputs</span><span class="p">,</span><span
class="n">action</span><span class="p">,</span><span class="n">reward</span><span
class="p">}</span><span class="c"># [debug]</span>
```

```
<span class="k">def</span> <span class="nf">run</span><span class="p">():</span>
```

```
<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of
trials.&quot;&quot;&quot;</span>
```

```
<span class="c"># Set up environment and agent</span>
```

```
<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span
class="p">()</span> <span class="c"># create environment (also adds some dummy traffic)</span>
```

```
<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span
class="o">.</span><span class="n">create_agent</span><span class="p">(</span><span
class="n">LearningAgent</span><span class="p">)</span> <span class="c"># create agent</span>
```

```
<span class="n">e</span><span class="o">.</span><span class="n">set_primary_agent</span><span
class="p">(</span><span class="n">a</span><span class="p">,</span><span
class="n">enforce_deadline</span><span class="o">=</span><span class="bp">True</span><span
class="p">)</span> <span class="c"># specify agent to track</span>
```

```
<span class="c"># NOTE: You can set enforce_deadline=False while debugging to allow longer
trials</span>
```

```
<span class="c"># Now simulate it</span>
```

```
<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span
class="p">(</span><span class="n">e</span><span class="p">,</span><span
class="n">update_delay</span><span class="o">=</span><span class="mf">0.5</span><span
class="p">,</span><span class="n">display</span><span class="o">=</span><span
class="bp">False</span><span class="p">)</span> <span class="c"># create simulator (uses pygame
when display=True, if available)</span>
```

<span class="c"># NOTE: To speed up simulation, reduce update\_delay and/or set display=False</span>

<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span class="p">(</span><span class="n">n\_trials</span><span class="o">=</span><span class="mi">2</span></span><span class="p">)</span> <span class="c"># run for a specified number of trials</span>

<span class="c"># NOTE: To quit midway, press Esc or close pygame window, or hit Ctrl+C on the command-line</span>

<span class="k">if</span> <span class="n">\_\_name\_\_</span> <span class="o">==</span> <span class="s">'\_\_main\_\_'</span><span class="p">:</span>

<span class="n">run</span><span class="p">()</span>

</pre></div>

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt"></div>

<div class="output\_subarea output\_stream output\_stdout output\_text">

<pre>Simulator.run(): Trial 0

Environment.reset(): Trial set up with start = (3, 6), destination = (4, 3), deadline = 20

RoutePlanner.route\_to(): destination = (4, 3)

LearningAgent.update(): deadline = 20, inputs = {'&apos;light&apos;': '&apos;red&apos;', '&apos;oncoming&apos;': None, '&apos;right&apos;': None, '&apos;left&apos;': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None, 'forward': True}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent ran out of time! Trial aborted.

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (4, 6), destination = (1, 5), deadline = 20

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0



LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'red', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'oncoming': None, 'right': None, 'left': None}, action = forward, reward = -0.5

Environment.step(): Primary agent ran out of time! Trial aborted.

</pre>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Improve the Q-Learning Driving Agent</p>

<p>Your final task for this project is to enhance your driving agent so that, after sufficient training, the smartcab is able to reach the destination within the allotted time safely and efficiently. Parameters in the Q-Learning algorithm, such as the learning rate (alpha), the discount factor (gamma) and the exploration rate (epsilon) all contribute to the driving agent's ability to learn the best action for each state. To improve on the success of your smartcab:</p>

<p>Set the number of trials, n\_trials, in the simulation to 100.

Run the simulation with the deadline enforcement enforce\_deadline set to True (you will need to reduce the update delay update\_delay and set the display to False).

Observe the driving agent's learning and smartcab's success rate, particularly during the later trials.

Adjust one or several of the above parameters and iterate this process.

This task is complete once you have arrived at what you determine is the best combination of parameters required for your driving agent to learn successfully.</p>

<p>QUESTION: Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform? Best run successfully trained on trial 5 Simulator.run(): Trial 5

Environment.reset(): Trial set up with start = (6, 2), destination = (1, 5), deadline = 40

RoutePlanner.route\_to(): destination = (1, 5);policy choice

['forward', 'left', 'right', None]self.alpha = 0.9

self.epsilon = 0.0

self.gamma = 0.35

Tried a Boosted approach with the first code running a random based Q value policy and following with a best Q value implementation. The Boosted Q value resulted in success in first trial with a 0.5 penalty rate:

Environment.act(): Primary agent has reached destination!

success/total = 1/1 of 1 trials (net reward: 15.0)

penalties/moves (penalty rate): 9/18 (0.5)

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0</p>

<p>After 100 trials of the Boosted Q value the results were:

Environment.act(): Primary agent has reached destination!

success/total = 97/100 of 1 trials (net reward: 2184.0)

penalties/moves (penalty rate): 111/1382 (0.08)</p>

**QUESTION:** Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?

**Answer:** The agent does reach the destination before the deadline. The agent did not find a zero penalty policy. Even after extending the trial rate to 400 the penalty rate increased to 0.09 in 5,801 moves based upon a randomly changing environment

Environment.act(): Primary agent has reached destination!

success/total = 389/400 of 1 trials (net reward: 9216.0)

penalties/moves (penalty rate): 551/5801 (0.09)

The optimal policy is defined as  $Q'(S,A) \leftarrow r + \gamma \max_a Q(s',a')$

where  $S$ =state  $s'$ =state you end up in

$A$ =action we just experienced  $a'$ =all the actions you could take

$r$ =reward

$Q'$ =estimate of  $Q$

and consider trying simulated annealing when stuck

$T'(S) = \operatorname{argmax}_A Q'(S,A)$  w.p  $1-\epsilon$  where epsilon is as small as possible

lowering epsilon from 0.35 to 0.15 increased the success rate to 99/100 however the penalties increased from 111 to 130/1,283 moves=0.1

Environment.act(): Primary agent has reached destination!

success/total = 99/100 of 1 trials (net reward: 2230.0)

penalties/moves (penalty rate): 130/1283 (0.1)

I was able to lower the penalty to 0.06 while maintaining a 99/100 success rate by setting the discount to 0.15 and the gamma to 0.15

Environment.act(): Primary agent has reached destination!

success/total = 99/100 of 1 trials (net reward: 2264.5)

penalties/moves (penalty rate): 95/1474 (0.06)

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;[48]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="c">#file agentE.py implimenting above changes.  
run code to see results</span>

<span class="kn">import</span> <span class="nn">random</span>

<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>  
<span class="n">Agent</span><span class="p">,</span> <span class="n">Environment</span>

<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>  
<span class="n">RoutePlanner</span>

<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>  
<span class="n">Simulator</span>

<span class="kn">import</span> <span class="nn">random</span>

<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>  
<span class="n">Agent</span><span class="p">,</span> <span class="n">Environment</span>

<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>  
<span class="n">RoutePlanner</span>

<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>  
<span class="n">Simulator</span>

<span class="kn">import</span> <span class="nn">math</span>

<span class="kn">from</span> <span class="nn">collections</span> <span class="kn">import</span>  
<span class="n">namedtuple</span>

<span class="kn">import</span> <span class="nn">pprint</span>

<span class="kn">from</span> <span class="nn">scipy</span> <span class="kn">import</span> <span class="n">constants</span> <span class="k">as</span> <span class="n">sc</span>

<span class="k">class</span> <span class="nc">LearningAgent</span><span class="p">(</span><span class="n">Agent</span><span class="p">):</span>

<span class="sd">&quot;&quot;&quot;An agent that learns to drive in the smartcab world using Q  
learning&quot;&quot;&quot;</span>

```
<span class="k">def</span> <span class="nf">__init__</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">env</span><span class="p">):</span></span>
```

```
<span class="nb">super</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="p">.</span><span class="n">__init__</span><span class="p">(</span><span class="n">env</span><span class="p">)</span> <span class="c"># sets self.env = env, state = None, next_waypoint = None, and a default color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">color</span> <span class="o">=</span><span class="s">'red'</span> <span class="c"># OverflowError('ide color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span> <span class="o">=</span><span class="n">RoutePlanner</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span> <span class="c"># simple route planner to get next_waypoint</span>
```

```
<span class="c">##initialize q table here</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span> <span class="o">=</span><span class="nb">dict</span><span class="p">(</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">alpha</span> <span class="o">=</span><span class="mf">0.9</span> <span class="c">#alpha is the learning rate</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span> <span class="o">=</span><span class="mf">0.0</span> <span class="c">#epsilon is the exploratory rate</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">gamma</span> <span class="o">=</span><span class="mf">0.35</span> <span class="c"># gamma is the discount factor</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">discount</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">gamma</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_state</span> <span class="o">=</span><span class="bp">None</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span><span class="bp">None</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_action</span> <span class="o">=</span><span class="bp">None</span>
```

<span class="bp">self</span><span class="o">.</span><span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get\_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_reward</span><span class="o">=</span> <span class="bp">None</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">cumulativeRewards</span> <span class="o">=</span> <span class="mi">0</span></span>

<span class="k">def</span> <span class="nf">Toss</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">p</span><span class="p">):</span></span>

<span class="n">r</span> <span class="o">=</span> <span class="n">random</span><span class="o">.</span><span class="n">random</span><span class="p">()</span></span>

<span class="k">return</span> <span class="n">r</span> <span class="o">&lt;</span> <span class="n">p</span></span>

<span class="k">def</span> <span class="nf">reset</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">destination</span><span class="o">=</span> <span class="bp">None</span><span class="p">)</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">route\_to</span><span class="p">(</span><span class="n">destination</span><span class="p">)</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_state</span><span class="o">=</span> <span class="bp">None</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="bp">None</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_action</span><span class="o">=</span> <span class="bp">None</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span> <span class="o">=</span> <span class="mf">0.0</span> <span class="c"># we want epilon as samll as possible  
<https://www.udacity.com/course/viewer#!c-ud728-nd/l-5446820041/m-634899065></span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">cumulativeRewards</span> <span class="o">=</span> <span class="mi">0</span></span>

```
<span class="k">def</span> <span class="nf">actionsCanTake</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">state</span><span class="p">):</span></pre>
```

```
<span class="k">return</span> <span class="p">[</span><span class="s">&#39;forward&#39;</span><span class="p">,</span><span class="s">&#39;left&#39;</span><span class="p">,</span><span class="s">&#39;right&#39;</span><span class="p">,</span><span class="bp">None</span><span class="p">]</span></pre>
```

```
<span class="k">def</span> <span class="nf">getQValue</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">state</span><span class="p">,</span><span class="p">,</span><span class="n">action</span><span class="p">):</span></pre>
```

```
<span class="k">return</span> <span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="o">.</span><span class="n">get</span><span class="p">((</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">),</span><span class="mf">20.0</span><span class="p">)</span></pre>
```

```
<span class="k">def</span> <span class="nf">getValue</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">state</span><span class="p">):</span></pre>
```

```
<span class="n">legalActions</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">actionsCanTake</span><span class="p">(</span><span class="n">state</span><span class="p">)</span></pre>
```

```
<span class="n">bestQValue</span> <span class="o">=</span> <span class="o">-</span> <span class="mi">99999999</span></pre>
```

```
<span class="k">for</span> <span class="n">action</span> <span class="ow">in</span> <span class="n">legalActions</span><span class="p">:</span></pre>
```

```
<span class="k">if</span> <span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span><span class="o">&gt;</span> <span class="n">bestQValue</span><span class="p">)</span></pre>
```

```
<span class="n">bestQValue</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span></pre>
```



<span class="k">return</span> <span class="n">bestQValue</span>

<span class="k">print</span> <span class="s">&#39;bestQValue&#39;</span>

<span class="k">def</span> <span class="nf">getPolicy</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">state</span><span class="p">):</span>

<span class="n">legalActions</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">actionsCanTake</span><span class="p">(</span><span class="n">state</span><span class="p">)</span>

<span class="n">bestAction</span> <span class="o">=</span> <span class="bp">None</span>

<span class="n">bestQValue</span> <span class="o">=</span> <span class="o">-</span> <span class="mi">999999999</span>

<span class="k">for</span> <span class="n">action</span> <span class="ow">in</span> <span class="n">legalActions</span><span class="p">:</span>

<span class="k">if</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span> <span class="o">&gt;</span> <span class="n">bestQValue</span><span class="p">):</span>

<span class="n">bestQValue</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span>

<span class="n">bestAction</span> <span class="o">=</span> <span class="n">action</span>

<span class="k">if</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span> <span class="o">==</span> <span class="n">bestQValue</span><span class="p">):</span>

<span class="k">if</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">Toss</span><span class="p">(</span><span class="o">.</span><span class="mi">5</span><span class="p">))</span>

<span class="n">bestQValue</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span>

<span class="n">bestAction</span> <span class="o">=</span> <span class="n">action</span>

<span class="k">return</span> <span class="n">bestAction</span>

<span class="k">def</span> <span class="nf">makeState</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">):</span>

<span class="n">State</span> <span class="o">=</span> <span class="n">namedtuple</span><span class="p">(</span><span class="s">"State"</span><span class="p">,</span> <span class="p">[</span><span class="s">"<span class="p">light</span>"</span><span class="p">,</span> <span class="s">"<span class="p">next\_waypoint</span>"</span><span class="p">])</span>

<span class="k">return</span> <span class="n">State</span><span class="p">(</span><span class="n">light</span> <span class="o">=</span> <span class="n">state</span><span class="p">[</span><span class="s">"<span class="p">light</span>"</span><span class="p">],</span>

<span class="n">next\_waypoint</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">())</span>

<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span>

<span class="c"># Gather inputs</span>

<span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">()</span>

<span class="n">inputs</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get\_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="c">#current state<span class="p">S</span>"</span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="p">

```
class="n">makeState</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">))</span>
```

```
<span class="c"># TODO: Select action according to your policy</span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getAction</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">state</span><span class="p">)</span>
```

```
<span class="c">#execute action and get reward</span>
```

```
<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">action</span><span class="p">)</span>
```

```
<span class="k">if</span> <span class="bp">self</span><span class="o">.</span><span class="n">previous_reward</span><span class="o">!=</span> <span class="bp">None</span><span class="p">:</span><span class="n">previous_reward</span> = <span class="n">reward</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">updateQTable</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">previous_state</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">previous_action</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">state</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">previous_reward</span><span class="p">)</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_action</span> = <span class="n">action</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_state</span> = <span class="n">state</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_reward</span> = <span class="n">reward</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">cumulativeRewards</span> += <span class="n">reward</span>
```

```
<span class="k">def</span> <span class="nf">getAction</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">)</span></span>
```

```
<span class="n">legalActions</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">actionsCanTake</span><span class="p">(</span><span class="n">state</span><span class="p">)</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="bp">None</span></span>
```

```
<span class="k">if</span> <span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">Toss</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span><span class="p">))</span></span>
```

```
<span class="k">print</span> <span class="s">&quot;random choice&quot;</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="n">random</span><span class="o">.</span><span class="n">choice</span><span class="p">(</span><span class="n">actionsCanTake</span><span class="p">)</span></span>
```

```
<span class="k">else</span><span class="p">:</span></span>
```

```
<span class="k">print</span> <span class="s">&quot;Running policy choice. Policy=alpha=0.9;epsilon=0;gamma=0.35&quot;</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getPolicy</span><span class="p">(</span><span class="n">state</span><span class="p">)</span></span>
```

```
<span class="k">return</span> <span class="n">action</span></span>
```

```
<span class="k">def</span> <span class="nf">updateQTable</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">,</span> <span class="n">nextState</span><span class="p">,</span> <span class="n">reward</span><span class="p">)</span></span>
```

```
<span class="k">if</span><span class="p">(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span> <span class="p">(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span></span>
```

```
class="ow">not</span> <span class="ow">in</span> <span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">):</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)]</span> <span class="o">=</span><span class="o">=</span> <span class="mf">20.0</span>
```

```
<span class="k">else</span><span class="p">:</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)]</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)]</span> <span class="o">+</span> <span class="bp">self</span><span class="o">.</span><span class="n">alpha</span><span class="o">*</span><span class="p">(</span><span class="n">reward</span> <span class="o">+</span> <span class="bp">self</span><span class="o">.</span><span class="n">discount</span><span class="o">*</span><span class="bp">self</span><span class="o">.</span><span class="n">getValue</span><span class="p">(</span><span class="n">nextState</span><span class="p">)</span><span class="o">-</span> <span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)]</span>
```

```
<span class="c">#print &quot;LearningAgent.updateQTable(): self = {}, state = {}, action = {}, reward = {}&quot;.format(self, state, action,nextState, reward)</span>
```

```
<span class="k">def</span> <span class="nf">run</span><span class="p">():</span>
```

```
<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of trials.&quot;&quot;&quot;</span>
```

```
<span class="c"># Set up environment and agent</span>
```

```
<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span class="p">()</span> <span class="c"># create environment (also adds some dummy traffic)</span>
```

```
<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span class="o">.</span><span class="n">create_agent</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">)</span> <span class="c"># create agent</span>
```

```
<span class="n">e</span><span class="o">.</span><span class="n">set_primary_agent</span><span
class="p">(</span><span class="n">a</span><span class="p">,</span> <span
class="n">enforce_deadline</span><span class="o">=</span><span class="bp">True</span><span
class="p">)</span> <span class="c"># set agent to track</span>
```

```
<span class="c"># Now simulate it</span>
```

```
<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span
class="p">(</span><span class="n">e</span><span class="p">,</span> <span
class="n">update_delay</span><span class="o">=</span><span class="mi">0</span><span
class="p">,</span> <span class="n">display</span><span class="o">=</span><span
class="bp">False</span><span class="p">)</span> <span class="c"># reduce update_delay to speed up
simulation</span>
```

```
<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span
class="p">(</span><span class="n">n_trials</span><span class="o">=</span><span
class="mi">100</span><span class="p">)</span> <span class="c"># press Esc or close pygame window
to quit</span>
```

```
<span class="c"># code assistance from rahulravindran and jaycode</span>
```

```
<span class="k">if</span> <span class="n">__name__</span> <span class="o">==</span> <span
class="s">"&#39;__main__&#39;"</span><span class="p">:</span><span>
```

```
<span class="n">run</span><span class="p">()</span>
```

```
</pre></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="output_wrapper">
```

```
<div class="output">
```

```
<div class="output_area"><div class="prompt"></div>
```

```
<div class="output_subarea output_stream output_stdout output_text">
```

```
<pre>bestQValue
```

```
Simulator.run(): Trial 0
```

```
Environment.reset(): Trial set up with start = (2, 6), destination = (1, 1), deadline = 30
```

```
RoutePlanner.route_to(): destination = (1, 1)
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
```

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.step(): Primary agent ran out of time! Trial aborted.

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (8, 5), destination = (4, 5), deadline = 20

RoutePlanner.route\_to(): destination = (4, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$



Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.step(): Primary agent ran out of time! Trial aborted.

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (2, 2), destination = (3, 6), deadline = 25

RoutePlanner.route\_to(): destination = (3, 6)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.step(): Primary agent ran out of time! Trial aborted.

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (6, 1), destination = (7, 4), deadline = 20

RoutePlanner.route\_to(): destination = (7, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (3, 4), destination = (8, 2), deadline = 35

RoutePlanner.route\_to(): destination = (8, 2)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 5

Environment.reset(): Trial set up with start = (7, 2), destination = (2, 4), deadline = 35

RoutePlanner.route\_to(): destination = (2, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 6

Environment.reset(): Trial set up with start = (2, 3), destination = (6, 6), deadline = 35

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 7

Environment.reset(): Trial set up with start = (7, 3), destination = (3, 6), deadline = 35

RoutePlanner.route\_to(): destination = (3, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 8

Environment.reset(): Trial set up with start = (5, 5), destination = (7, 1), deadline = 30

RoutePlanner.route\_to(): destination = (7, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 9

Environment.reset(): Trial set up with start = (5, 6), destination = (8, 5), deadline = 20

RoutePlanner.route\_to(): destination = (8, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 10

Environment.reset(): Trial set up with start = (5, 1), destination = (6, 4), deadline = 20

RoutePlanner.route\_to(): destination = (6, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 11

Environment.reset(): Trial set up with start = (1, 6), destination = (1, 1), deadline = 25

RoutePlanner.route\_to(): destination = (1, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 12

Environment.reset(): Trial set up with start = (5, 2), destination = (3, 4), deadline = 20

RoutePlanner.route\_to(): destination = (3, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 13

Environment.reset(): Trial set up with start = (8, 2), destination = (3, 1), deadline = 30

RoutePlanner.route\_to(): destination = (3, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$



Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 14

Environment.reset(): Trial set up with start = (7, 2), destination = (6, 6), deadline = 25

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 15

Environment.reset(): Trial set up with start = (8, 2), destination = (2, 5), deadline = 45

RoutePlanner.route\_to(): destination = (2, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 16

Environment.reset(): Trial set up with start = (4, 2), destination = (7, 3), deadline = 20

RoutePlanner.route\_to(): destination = (7, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 17

Environment.reset(): Trial set up with start = (6, 6), destination = (4, 2), deadline = 30

RoutePlanner.route\_to(): destination = (4, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 18

Environment.reset(): Trial set up with start = (6, 5), destination = (4, 2), deadline = 25

RoutePlanner.route\_to(): destination = (4, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 19

Environment.reset(): Trial set up with start = (6, 5), destination = (6, 1), deadline = 20

RoutePlanner.route\_to(): destination = (6, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 20

Environment.reset(): Trial set up with start = (7, 3), destination = (1, 2), deadline = 35

RoutePlanner.route\_to(): destination = (1, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 21

Environment.reset(): Trial set up with start = (4, 2), destination = (8, 2), deadline = 20

RoutePlanner.route\_to(): destination = (8, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 22

Environment.reset(): Trial set up with start = (2, 5), destination = (8, 1), deadline = 50

RoutePlanner.route\_to(): destination = (8, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 23

Environment.reset(): Trial set up with start = (5, 1), destination = (6, 6), deadline = 30

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 24

Environment.reset(): Trial set up with start = (4, 3), destination = (6, 6), deadline = 25

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 25

Environment.reset(): Trial set up with start = (6, 1), destination = (2, 2), deadline = 25

RoutePlanner.route\_to(): destination = (2, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 26

Environment.reset(): Trial set up with start = (5, 2), destination = (4, 6), deadline = 25

RoutePlanner.route\_to(): destination = (4, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$



Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 27

Environment.reset(): Trial set up with start = (6, 5), destination = (4, 3), deadline = 20

RoutePlanner.route\_to(): destination = (4, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 28

Environment.reset(): Trial set up with start = (5, 2), destination = (2, 1), deadline = 20

RoutePlanner.route\_to(): destination = (2, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 29

Environment.reset(): Trial set up with start = (1, 3), destination = (7, 3), deadline = 30

RoutePlanner.route\_to(): destination = (7, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 30

Environment.reset(): Trial set up with start = (1, 6), destination = (4, 2), deadline = 35

RoutePlanner.route\_to(): destination = (4, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 31

Environment.reset(): Trial set up with start = (6, 6), destination = (5, 2), deadline = 25

RoutePlanner.route\_to(): destination = (5, 2)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 32

Environment.reset(): Trial set up with start = (7, 5), destination = (2, 1), deadline = 45

RoutePlanner.route\_to(): destination = (2, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 33

Environment.reset(): Trial set up with start = (5, 3), destination = (3, 5), deadline = 20

RoutePlanner.route\_to(): destination = (3, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 34

Environment.reset(): Trial set up with start = (6, 3), destination = (3, 2), deadline = 20

RoutePlanner.route\_to(): destination = (3, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 35

Environment.reset(): Trial set up with start = (1, 2), destination = (7, 4), deadline = 40

RoutePlanner.route\_to(): destination = (7, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 36

Environment.reset(): Trial set up with start = (3, 6), destination = (7, 6), deadline = 20

RoutePlanner.route\_to(): destination = (7, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 37

Environment.reset(): Trial set up with start = (5, 4), destination = (7, 1), deadline = 25

RoutePlanner.route\_to(): destination = (7, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 38

Environment.reset(): Trial set up with start = (3, 6), destination = (6, 3), deadline = 30

RoutePlanner.route\_to(): destination = (6, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 39

Environment.reset(): Trial set up with start = (6, 4), destination = (1, 4), deadline = 25

RoutePlanner.route\_to(): destination = (1, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 40

Environment.reset(): Trial set up with start = (1, 3), destination = (8, 6), deadline = 50

RoutePlanner.route\_to(): destination = (8, 6)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35



Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 41

Environment.reset(): Trial set up with start = (4, 3), destination = (2, 5), deadline = 20

RoutePlanner.route\_to(): destination = (2, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 42

Environment.reset(): Trial set up with start = (7, 6), destination = (1, 1), deadline = 55

RoutePlanner.route\_to(): destination = (1, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 43

Environment.reset(): Trial set up with start = (5, 5), destination = (6, 2), deadline = 20

RoutePlanner.route\_to(): destination = (6, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 44

Environment.reset(): Trial set up with start = (8, 4), destination = (1, 3), deadline = 40

RoutePlanner.route\_to(): destination = (1, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 45

Environment.reset(): Trial set up with start = (7, 2), destination = (5, 5), deadline = 25

RoutePlanner.route\_to(): destination = (5, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 46

Environment.reset(): Trial set up with start = (4, 1), destination = (8, 5), deadline = 40

RoutePlanner.route\_to(): destination = (8, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 47

Environment.reset(): Trial set up with start = (6, 4), destination = (2, 3), deadline = 25

RoutePlanner.route\_to(): destination = (2, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 48

Environment.reset(): Trial set up with start = (1, 4), destination = (7, 6), deadline = 40

RoutePlanner.route\_to(): destination = (7, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 49

Environment.reset(): Trial set up with start = (8, 3), destination = (6, 6), deadline = 25

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 50

Environment.reset(): Trial set up with start = (3, 3), destination = (1, 6), deadline = 25

RoutePlanner.route\_to(): destination = (1, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 51

Environment.reset(): Trial set up with start = (5, 5), destination = (1, 2), deadline = 35

RoutePlanner.route\_to(): destination = (1, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 52

Environment.reset(): Trial set up with start = (7, 2), destination = (4, 3), deadline = 20

RoutePlanner.route\_to(): destination = (4, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 53

Environment.reset(): Trial set up with start = (2, 5), destination = (5, 6), deadline = 20

RoutePlanner.route\_to(): destination = (5, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 54

Environment.reset(): Trial set up with start = (4, 5), destination = (1, 2), deadline = 30

RoutePlanner.route\_to(): destination = (1, 2)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 55

Environment.reset(): Trial set up with start = (1, 3), destination = (6, 4), deadline = 30

RoutePlanner.route\_to(): destination = (6, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35



Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 56

Environment.reset(): Trial set up with start = (3, 1), destination = (7, 3), deadline = 30

RoutePlanner.route\_to(): destination = (7, 3)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 57

Environment.reset(): Trial set up with start = (1, 6), destination = (4, 4), deadline = 25

RoutePlanner.route\_to(): destination = (4, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 58

Environment.reset(): Trial set up with start = (8, 4), destination = (1, 4), deadline = 35

RoutePlanner.route\_to(): destination = (1, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 59

Environment.reset(): Trial set up with start = (8, 2), destination = (7, 6), deadline = 25

RoutePlanner.route\_to(): destination = (7, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 60

Environment.reset(): Trial set up with start = (5, 5), destination = (3, 2), deadline = 25

RoutePlanner.route\_to(): destination = (3, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 61

Environment.reset(): Trial set up with start = (6, 3), destination = (8, 5), deadline = 20

RoutePlanner.route\_to(): destination = (8, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 62

Environment.reset(): Trial set up with start = (1, 4), destination = (7, 3), deadline = 35

RoutePlanner.route\_to(): destination = (7, 3)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 63

Environment.reset(): Trial set up with start = (1, 1), destination = (6, 5), deadline = 45

RoutePlanner.route\_to(): destination = (6, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 64

Environment.reset(): Trial set up with start = (2, 6), destination = (6, 1), deadline = 45

RoutePlanner.route\_to(): destination = (6, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 65

Environment.reset(): Trial set up with start = (5, 2), destination = (4, 6), deadline = 25

RoutePlanner.route\_to(): destination = (4, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 66

Environment.reset(): Trial set up with start = (7, 1), destination = (2, 2), deadline = 30

RoutePlanner.route\_to(): destination = (2, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 67

Environment.reset(): Trial set up with start = (8, 5), destination = (3, 6), deadline = 30

RoutePlanner.route\_to(): destination = (3, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 68

Environment.reset(): Trial set up with start = (3, 5), destination = (7, 1), deadline = 40

RoutePlanner.route\_to(): destination = (7, 1)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$



Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 69

Environment.reset(): Trial set up with start = (7, 1), destination = (2, 2), deadline = 30

RoutePlanner.route\_to(): destination = (2, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 70

Environment.reset(): Trial set up with start = (5, 6), destination = (8, 4), deadline = 25

RoutePlanner.route\_to(): destination = (8, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 71

Environment.reset(): Trial set up with start = (8, 6), destination = (3, 2), deadline = 45

RoutePlanner.route\_to(): destination = (3, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 72

Environment.reset(): Trial set up with start = (5, 2), destination = (2, 1), deadline = 20

RoutePlanner.route\_to(): destination = (2, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 73

Environment.reset(): Trial set up with start = (6, 4), destination = (8, 2), deadline = 20

RoutePlanner.route\_to(): destination = (8, 2)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 74

Environment.reset(): Trial set up with start = (1, 2), destination = (7, 3), deadline = 35

RoutePlanner.route\_to(): destination = (7, 3)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 75

Environment.reset(): Trial set up with start = (8, 6), destination = (6, 2), deadline = 30

RoutePlanner.route\_to(): destination = (6, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 76

Environment.reset(): Trial set up with start = (6, 2), destination = (4, 5), deadline = 25

RoutePlanner.route\_to(): destination = (4, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 77

Environment.reset(): Trial set up with start = (3, 3), destination = (2, 6), deadline = 20

RoutePlanner.route\_to(): destination = (2, 6)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 78

Environment.reset(): Trial set up with start = (3, 2), destination = (8, 6), deadline = 45

RoutePlanner.route\_to(): destination = (8, 6)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 79

Environment.reset(): Trial set up with start = (7, 3), destination = (2, 5), deadline = 35

RoutePlanner.route\_to(): destination = (2, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 80

Environment.reset(): Trial set up with start = (4, 3), destination = (2, 1), deadline = 20

RoutePlanner.route\_to(): destination = (2, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 81

Environment.reset(): Trial set up with start = (2, 3), destination = (6, 1), deadline = 30

RoutePlanner.route\_to(): destination = (6, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 82

Environment.reset(): Trial set up with start = (8, 6), destination = (4, 5), deadline = 25

RoutePlanner.route\_to(): destination = (4, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 83

Environment.reset(): Trial set up with start = (1, 2), destination = (1, 6), deadline = 20



RoutePlanner.route\_to(): destination = (1, 6)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 84

Environment.reset(): Trial set up with start = (3, 2), destination = (8, 1), deadline = 30

RoutePlanner.route\_to(): destination = (8, 1)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 85

Environment.reset(): Trial set up with start = (3, 6), destination = (5, 4), deadline = 20

RoutePlanner.route\_to(): destination = (5, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 86

Environment.reset(): Trial set up with start = (2, 2), destination = (7, 5), deadline = 40

RoutePlanner.route\_to(): destination = (7, 5)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 87

Environment.reset(): Trial set up with start = (7, 3), destination = (6, 6), deadline = 20

RoutePlanner.route\_to(): destination = (6, 6)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 88

Environment.reset(): Trial set up with start = (8, 2), destination = (4, 4), deadline = 30

RoutePlanner.route\_to(): destination = (4, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 89

Environment.reset(): Trial set up with start = (3, 4), destination = (7, 5), deadline = 25

RoutePlanner.route\_to(): destination = (7, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 90

Environment.reset(): Trial set up with start = (5, 3), destination = (1, 2), deadline = 25

RoutePlanner.route\_to(): destination = (1, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 91

Environment.reset(): Trial set up with start = (1, 4), destination = (6, 4), deadline = 25

RoutePlanner.route\_to(): destination = (6, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 92

Environment.reset(): Trial set up with start = (8, 5), destination = (4, 2), deadline = 35

RoutePlanner.route\_to(): destination = (4, 2)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 93

Environment.reset(): Trial set up with start = (2, 2), destination = (7, 2), deadline = 25

RoutePlanner.route\_to(): destination = (7, 2)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 94

Environment.reset(): Trial set up with start = (2, 1), destination = (8, 4), deadline = 45

RoutePlanner.route\_to(): destination = (8, 4)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 95

Environment.reset(): Trial set up with start = (2, 1), destination = (1, 5), deadline = 25

RoutePlanner.route\_to(): destination = (1, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 96

Environment.reset(): Trial set up with start = (2, 4), destination = (8, 5), deadline = 35

RoutePlanner.route\_to(): destination = (8, 5)

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 97

Environment.reset(): Trial set up with start = (4, 5), destination = (8, 4), deadline = 25

RoutePlanner.route\_to(): destination = (8, 4)

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Running policy choice. Policy=  $\alpha=0.9$ ;  $\epsilon=0$ ;  $\gamma=0.35$

Environment.act(): Primary agent has reached destination!

Simulator.run(): Trial 98

Environment.reset(): Trial set up with start = (7, 4), destination = (4, 1), deadline = 30

RoutePlanner.route\_to(): destination = (4, 1)



```
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Environment.act(): Primary agent has reached destination!
Simulator.run(): Trial 99
Environment.reset(): Trial set up with start = (4, 3), destination = (7, 5), deadline = 25
RoutePlanner.route_to(): destination = (7, 5)
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Running policy choice. Policy= alpha=0.9;epsilon=0;gamma=0.35
Environment.act(): Primary agent has reached destination!
```

```
</pre>
```

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Refined code to print wanted parameters obtained</p>

</div>

</div>

</div>

<div class="cell border-box-sizing text\_cell rendered">

<div class="prompt input\_prompt">

</div>

<div class="inner\_cell">

<div class="text\_cell\_render border-box-sizing rendered\_html">

<p>Attempted a Boost approach random based Q policy followed by find best Q policy</p>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

<div class="input">

<div class="prompt input\_prompt">In&nbsp;  [3]:</div>

<div class="inner\_cell">

<div class="input\_area">

<div class=" highlight hl-ipython2"><pre><span class="c">#file agentE.py implimenting above changes.  
run code to see results</span>

<span class="kn">import</span> <span class="nn">random</span>

<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>  
<span class="n">Agent</span><span class="p">,</span> <span class="n">Environment</span>

<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>  
<span class="n">RoutePlanner</span>

<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>  
<span class="n">Simulator</span>

<span class="kn">import</span> <span class="nn">random</span>

<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span>  
<span class="n">Agent</span><span class="p">,</span> <span class="n">Environment</span>

<span class="kn">from</span> <span class="nn">planner</span> <span class="kn">import</span>  
<span class="n">RoutePlanner</span>

<span class="kn">from</span> <span class="nn">simulator</span> <span class="kn">import</span>  
<span class="n">Simulator</span>

<span class="kn">import</span> <span class="nn">math</span>

<span class="kn">from</span> <span class="nn">collections</span> <span class="kn">import</span>  
<span class="n">namedtuple</span>

<span class="kn">import</span> <span class="nn">pprint</span>

<span class="kn">from</span> <span class="nn">scipy</span> <span class="kn">import</span> <span class="n">constants</span> <span class="k">as</span> <span class="n">sc</span>

<span class="k">class</span> <span class="nc">LearningAgent</span><span class="p">(</span><span class="n">Agent</span><span class="p">):</span>

<span class="sd">&quot;&quot;&quot;An agent that learns to drive in the smartcab world using Q  
learning&quot;&quot;&quot;</span>

```
<span class="k">def</span> <span class="nf">__init__</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">env</span><span class="p">):</span></pre>
```

```
<span class="nb">super</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="o">.</span><span class="n">__init__</span><span class="p">(</span><span class="n">env</span><span class="p">)</span> <span class="c"># sets self.env = env, state = None, next_waypoint = None, and a default color</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">color</span> <span class="o">=</span><span class="s">'red'</span> <span class="c"># OverflowError('ide color</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span> <span class="o">=</span><span class="n">RoutePlanner</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span> <span class="c"># simple route planner to get next_waypoint</span></pre>
```

```
<span class="c">##initialize q table here</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span> <span class="o">=</span><span class="nb">dict</span><span class="p">(</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">alpha</span> <span class="o">=</span><span class="mf">0.9</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span> <span class="o">=</span><span class="mf">0.0</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">gamma</span> <span class="o">=</span><span class="mf">0.15</span> <span class="c">#tried changing gamma</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">discount</span> <span class="o">=</span><span class="mf">0.15</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_state</span> <span class="o">=</span><span class="bp">None</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span><span class="bp">None</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">previous_action</span> <span class="o">=</span><span class="bp">None</span></pre>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">deadline</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">get_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span></pre>
```

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_reward</span>  
<span class="o">=</span> <span class="bp">None</span>

<span class="bp">self</span><span class="o">.</span><span  
class="n">cumulativeRewards</span> <span class="o">=</span> <span class="mi">0</span>

<span class="k">def</span> <span class="nf">Toss</span><span class="p">(</span><span  
class="bp">self</span><span class="p">,</span> <span class="n">p</span><span class="p">):</span>

<span class="n">r</span><span class="o">=</span> <span class="n">random</span><span  
class="o">.</span><span class="n">random</span><span class="p">(</span>

<span class="k">return</span> <span class="n">r</span> <span class="o">&lt;</span> <span  
class="n">p</span>

<span class="k">def</span> <span class="nf">reset</span><span class="p">(</span><span  
class="bp">self</span><span class="p">,</span> <span class="n">destination</span><span  
class="o">=</span><span class="bp">None</span><span class="p">):</span>

<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span  
class="o">.</span><span class="n">route\_to</span><span class="p">(</span><span  
class="n">destination</span><span class="p">)</span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_state</span>  
<span class="o">=</span> <span class="bp">None</span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span  
class="o">=</span> <span class="bp">None</span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_action</span>  
<span class="o">=</span> <span class="bp">None</span>

<span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span> <span  
class="o">=</span> <span class="mf">0.0</span> <span class="c"># we want epilon as samll as possible  
<https://www.udacity.com/course/viewer#!c-ud728-nd/l-5446820041/m-634899065></span>

<span class="bp">self</span><span class="o">.</span><span  
class="n">cumulativeRewards</span> <span class="o">=</span> <span class="mi">0</span>

<span class="k">def</span> <span class="nf">actionsCanTake</span><span class="p">(</span><span  
class="bp">self</span><span class="p">,</span> <span class="n">state</span><span  
class="p">):</span>

<span class="k">return</span> <span class="p">[</span><span class="s">&#39;forward&#39;</span><span class="p">,</span> <span class="s">&#39;left&#39;</span><span class="p">,</span> <span class="s">&#39;right&#39;</span><span class="p">,</span> <span class="bp">None</span><span class="p">]</span>

<span class="k">def</span> <span class="nf">getQValue</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">):</span>

<span class="k">return</span> <span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="o">.</span><span class="n">get</span><span class="p">((</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">),</span> <span class="mf">20.0</span><span class="p">)</span>

<span class="k">def</span> <span class="nf">getValue</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">)</span>

<span class="n">legalActions</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">actionsCanTake</span><span class="p">(</span><span class="n">state</span><span class="p">)</span>

<span class="n">bestQValue</span> <span class="o">=</span> <span class="o">-</span> <span class="mi">999999999</span>

<span class="k">for</span> <span class="n">action</span> <span class="ow">in</span> <span class="n">legalActions</span><span class="p">:</span>

<span class="k">if</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span> <span class="o">&gt;</span> <span class="n">bestQValue</span><span class="p">)</span>

<span class="n">bestQValue</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getQValue</span><span class="p">(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span>

<span class="k">return</span> <span class="n">bestQValue</span>



<span class="k">return</span> <span class="n">bestAction</span>

<span class="k">def</span> <span class="nf">makeState</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">):</span>

<span class="n">State</span> <span class="o">=</span> <span class="n">namedtuple</span><span class="p">(</span><span class="s">"&quot;State&quot;</span><span class="p">,</span> <span class="p">[</span><span class="s">"&quot;light&quot;</span><span class="p">,</span> <span class="s">"&quot;next\_waypoint&quot;</span><span class="p">])</span>

<span class="k">return</span> <span class="n">State</span><span class="p">(</span><span class="n">light</span> <span class="o">=</span> <span class="n">state</span><span class="p">[</span><span class="s">"&#39;light&#39;</span><span class="p">],</span>

<span class="n">next\_waypoint</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">())</span>

<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span>

<span class="c"># Gather inputs</span>

<span class="bp">self</span><span class="o">.</span><span class="n">next\_waypoint</span>  
<span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next\_waypoint</span><span class="p">())</span>

<span class="n">inputs</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="n">deadline</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get\_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span>

<span class="c">#current state&#39;</span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">makeState</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="p">)</span>



class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">))</span>

<span class="c"># TODO: Select action according to your policy</span>

<span class="n">action</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">getAction</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">state</span><span class="p">)</span>

<span class="c">#execute action and get reward</span>

<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">action</span><span class="p">)</span>

<span class="k">if</span> <span class="bp">self</span><span class="o">.</span><span class="n">previous\_reward</span><span class="o">!=</span> <span class="bp">None</span><span class="p">:</span>

<span class="bp">self</span><span class="o">.</span><span class="n">updateQTable</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">previous\_state</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">previous\_action</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">state</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">previous\_reward</span><span class="p">)</span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_action</span><span class="o">=</span> <span class="n">action</span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_state</span><span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">state</span>

<span class="bp">self</span><span class="o">.</span><span class="n">previous\_reward</span><span class="o">=</span> <span class="n">reward</span>

<span class="bp">self</span><span class="o">.</span><span class="n">cumulativeRewards</span> <span class="o">+=</span> <span class="n">reward</span>

```
<span class="k">def</span> <span class="nf">getAction</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">):</span></span>
```

```
<span class="n">legalActions</span> <span class="o">=</span> <span class="bp">self</span><span class="p">.</span><span class="n">actionsCanTake</span><span class="p">(</span><span class="n">state</span><span class="p">)</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="bp">None</span></span>
```

```
<span class="k">if</span> <span class="p">(</span><span class="bp">self</span><span class="p">.</span><span class="n">Toss</span><span class="p">(</span><span class="bp">self</span><span class="p">.</span><span class="n">epsilon</span><span class="p">))</span></span>
```

```
<span class="k">print</span> <span class="s">&quot;random choice&quot;</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="n">random</span><span class="p">.</span><span class="n">choice</span><span class="p">(</span><span class="n">actionsCanTake</span><span class="p">)</span></span>
```

```
<span class="k">else</span><span class="p">:</span></span>
```

```
<span class="k">print</span> <span class="s">&quot;Running policy choice. Policy=alpha=0.9;epsilon=0;gamma=0.35&quot;</span></span>
```

```
<span class="n">action</span> <span class="o">=</span> <span class="bp">self</span><span class="p">.</span><span class="n">getPolicy</span><span class="p">(</span><span class="n">state</span><span class="p">)</span></span>
```

```
<span class="k">return</span> <span class="n">action</span></span>
```

```
<span class="k">def</span> <span class="nf">updateQTable</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">,</span> <span class="n">nextState</span><span class="p">,</span> <span class="n">reward</span><span class="p">)</span></span>
```

```
<span class="k">if</span><span class="p">(</span><span class="n">state</span><span class="p">,</span> <span class="n">action</span><span class="p">)</span> <span class="ow">not</span> <span class="ow">in</span> <span class="bp">self</span><span class="p">.</span><span class="n">qDict</span><span class="p">:</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)]</span><span class="o">=</span><span class="o">=</span><span class="mf">20.0</span></span>
```

```
<span class="k">else</span><span class="p">:</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)]</span><span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)]</span><span class="o">+</span><span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)]</span><span class="o">+</span><span class="bp">self</span><span class="o">.</span><span class="n">alpha</span><span class="o">*</span><span class="p">(</span><span class="n">reward</span><span class="o">+</span><span class="bp">self</span><span class="o">.</span><span class="n">discount</span><span class="o">*</span><span class="bp">self</span><span class="o">.</span><span class="n">getValue</span><span class="p">(</span><span class="n">nextState</span><span class="p">)</span><span class="o">-</span><span class="bp">self</span><span class="o">.</span><span class="n">qDict</span><span class="p">[(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)]</span></span>
```

```
<span class="c">#print &quot;LearningAgent.updateQTable(): self = {}, state = {}, action = {}, reward = {}&quot;.</span>format(self, state, action,nextState, reward) </span>
```

```
<span class="kn">import</span> <span class="nn">pdb</span></span>
```

```
<span class="kn">from</span> <span class="nn">environment</span> <span class="kn">import</span> <span class="n">Agent</span><span class="p">,</span><span class="n">TrafficLight</span></span>
```

```
<span class="kn">import</span> <span class="nn">operator</span></span>
```

```
<span class="k">class</span> <span class="nc">LearningAgent</span><span class="p">(</span><span class="n">Agent</span><span class="p">):</span></span>
```

```
<span class="sd">&quot;&quot;&quot;An agent that learns to drive in the smartcab world.&quot;&quot;&quot;</span>
```

```
<span class="k">def</span> <span class="nf">__init__</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">env</span><span class="p">,</span><span class="n">trials</span><span class="p">,</span><span class="o">=</span><span class="mi">1</span><span class="p">):</span></span>
```

```
<span class="nb">super</span><span class="p">(</span><span class="n">LearningAgent</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="o">.</span><span class="n">__init__</span><span class="p">(</span><span class="n">env</span><span class="p">)</span><span class="c"># sets self.env = env, state = None, next_waypoint = None, and a default color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">color</span><span class="o">=</span><span class="s">'red'</span><span class="c"># override color</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">=</span><span class="n">RoutePlanner</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="p">,</span><span class="bp">self</span><span class="p">)</span><span class="c"># simple route planner to get next_waypoint</span>
```

```
<span class="c"># TODO: Initialize any additional variables here</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">learning_rate</span><span class="o">=</span><span class="mf">0.8</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="o">=</span><span class="p">{}</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">default_Q</span><span class="o">=</span><span class="mi">1</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">discount_factor</span><span class="o">=</span><span class="mf">0.15</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">epsilon</span><span class="o">=</span><span class="mf">0.1</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">success</span><span class="o">=</span><span class="mi">0</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">total</span><span class="o">=</span><span class="mi">0</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">trials</span><span class="o">=</span><span class="n">trials</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">penalties</span><span class="o">=</span><span class="mi">0</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">moves</span><span class="o">=</span><span class="mi">0</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">net_reward</span><span class="o">=</span><span class="mi">0</span>
```

```
<span class="k">def</span> <span class="nf">reset</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">destination</span><span class="p">=</span><span class="bp">None</span><span class="p">):</span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="p">(</span><span class="o">.</span><span class="n">route_to</span><span class="p">(</span><span class="p">(</span><span class="n">destination</span><span class="p">,</span><span class="n">p</span><span class="p">)</span></span></span>
```

```
<span class="c"># TODO: Prepare for a new trip; reset any variables here, if required</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_state</span> <span class="p">=</span><span class="bp">None</span><span class="p"></span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_action</span> <span class="p">=</span><span class="bp">None</span><span class="p"></span></span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_reward</span> <span class="p">=</span><span class="bp">None</span><span class="p"></span></span>
```

```
<span class="k">def</span> <span class="nf">update</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span> <span class="n">t</span><span class="p">):</span></span>
```

```
<span class="c"># Gather inputs</span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span>  
<span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">planner</span><span class="o">.</span><span class="n">next_waypoint</span><span class="p">(</span></span>  
<span class="c"># from route planner, also displayed by simulator</span>
```

```
<span class="n">inputs</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">sense</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span></span>
```

```
<span class="n">deadline</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">get_deadline</span><span class="p">(</span><span class="bp">self</span><span class="p">)</span></span>
```

```
<span class="c"># TODO: Update state</span>
```

```
<span class="n">inputs</span><span class="p">[</span><span class="s">'</span>waypoint<span class="s">'</span><span class="p">]</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">next_waypoint</span>
```

```
<span class="k">del</span> <span class="n">inputs</span><span class="p">[</span><span class="s">'</span>oncoming<span class="s">'</span><span class="p">]</span></span>
```

<span class="k">del</span> <span class="n">inputs</span><span class="p">[</span><span class="s">&#39;left&#39;</span><span class="p">]</span></span>

<span class="k">del</span> <span class="n">inputs</span><span class="p">[</span><span class="s">&#39;right&#39;</span><span class="p">]</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">state</span> <span class="o">=</span><span class="nb">tuple</span><span class="p">(</span><span class="nb">sorted</span><span class="p">(</span><span class="n">inputs</span><span class="o">.</span><span class="n">items</span><span class="p">()))</span></span>

<span class="n">\_Q</span><span class="p">,</span> <span class="n">action</span> <span class="o">=</span><span class="bp">self</span><span class="o">.</span><span class="n">\_select\_Q\_action</span><span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">state</span><span class="p">)</span></span>

<span class="c"># Execute action and get reward</span>

<span class="n">reward</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">env</span><span class="o">.</span><span class="n">act</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">action</span><span class="p">)</span></span>

<span class="c"># Some stats</span>

<span class="bp">self</span><span class="o">.</span><span class="n">net\_reward</span> <span class="o">+=</span> <span class="n">reward</span>

<span class="bp">self</span><span class="o">.</span><span class="n">moves</span> <span class="o">+=</span> <span class="mi">1</span>

<span class="k">if</span> <span class="n">reward</span> <span class="o">&lt;</span> <span class="mi">0</span><span class="p">:</span></span>

<span class="bp">self</span><span class="o">.</span><span class="n">penalties</span> <span class="o">+=</span> <span class="mi">1</span>

<span class="n">add\_total</span> <span class="o">=</span> <span class="bp">False</span>

<span class="k">if</span> <span class="n">deadline</span> <span class="o">==</span> <span class="mi">0</span><span class="p">:</span></span>

<span class="n">add\_total</span> <span class="o">=</span> <span class="bp">True</span>

<span class="k">if</span> <span class="n">reward</span> <span class="o">&gt;</span> <span class="mi">5</span><span class="p">:</span></p>

<span class="bp">self</span><span class="o">.</span><span class="n">success</span> <span class="o">+=</span> <span class="mi">1</span></p>

<span class="n">add\_total</span> <span class="o">=</span> <span class="bp">True</span></p>

<span class="k">if</span> <span class="n">add\_total</span><span class="p">:</span></p>

<span class="bp">self</span><span class="o">.</span><span class="n">total</span> <span class="o">+=</span> <span class="mi">1</span></p>

<span class="k">print</span> <span class="bp">self</span><span class="o">.</span><span class="n">\_more\_stats</span><span class="p">()</span></p>

<span class="c"># TODO: Learn policy based on state, action, reward</span></p>

<span class="k">if</span> <span class="bp">self</span><span class="o">.</span><span class="n">prev\_state</span> <span class="o">!=</span> <span class="bp">None</span><span class="p">:</span></p>

<span class="k">if</span> <span class="p">(</span><span class="bp">self</span><span class="o">.</span><span class="n">prev\_state</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">prev\_action</span><span class="p">)</span> <span class="ow">not</span> <span class="ow">in</span> <span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="p">:</span></p>

<span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="p">[</span><span class="bp">self</span><span class="o">.</span><span class="n">prev\_state</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">prev\_action</span><span class="p">)]</span> <span class="o">=</span> <span class="bp">self</span><span class="o">.</span><span class="n">default\_Q</span></p>

<span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="p">[</span><span class="bp">self</span><span class="o">.</span><span class="n">prev\_state</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">prev\_action</span><span class="p">)]</span> <span class="o">=</span> <span class="p">(</span><span class="mi">1</span><span class="o">-</span><span class="bp">self</span><span class="o">.</span><span class="n">learning\_rate</span><span class="p">)</span> <span class="o">\*</span> <span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="p">[</span><span class="bp">self</span><span class="o">.</span><span class="n">prev\_state</span><span class="p">,</span> <span class="bp">self</span><span class="o">.</span><span class="n">prev\_action</span><span class="p">)]</span> <span class="o">+</span> \

```
<span class="bp">self</span><span class="o">.</span><span class="n">learning_rate</span>
<span class="o">*</span> <span class="p">(</span><span class="bp">self</span><span
class="o">.</span><span class="n">prev_reward</span> <span class="o">+</span> <span
class="bp">self</span><span class="o">.</span><span class="n">discount_factor</span> <span
class="o">*</span><span> \
```

```
<span class="bp">self</span><span class="o">.</span><span
class="n">_select_Q_action</span><span class="p">(</span><span class="bp">self</span><span
class="o">.</span><span class="n">state</span><span class="p">)</span><span
class="mi">0</span><span class="p">]</span><span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_state</span> <span
class="o">=</span> <span class="bp">self</span><span class="o">.</span><span
class="n">state</span><span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_action</span> <span
class="o">=</span> <span class="n">action</span><span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">prev_reward</span> <span
class="o">=</span> <span class="n">reward</span><span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">env</span><span
class="o">.</span><span class="n">status_text</span> <span class="o">+</span> <span
class="s">&#39; &#39;</span> <span class="o">+</span> <span class="bp">self</span><span
class="o">.</span><span class="n">_more_stats</span><span class="p">)</span><span>
```

```
<span class="k">print</span> <span class="s">&quot;LearningAgent.update(): deadline = {}, inputs
= {}, action = {}, reward = {}&quot;</span><span class="o">.</span><span
class="n">format</span><span class="p">(</span><span class="n">deadline</span><span
class="p">,</span> <span class="n">inputs</span><span class="p">,</span> <span
class="n">action</span><span class="p">,</span> <span class="n">reward</span><span
class="p">)</span> <span class="c"># [debug]</span><span>
```

```
<span class="k">def</span> <span class="nf">_more_stats</span><span class="p">(</span><span
class="bp">self</span><span class="p">):</span><span>
```

```
<span class="sd">&quot;&quot;&quot;Get additional stats&quot;&quot;&quot;</span><span>
```

```
<span class="k">return</span> <span class="s">&quot;success/total = {}/{} of {} trials (net reward:
{})</span><span class="se">\n</span><span class="s">&quot;penalties/moves (penalty rate): {}/{}
({})&quot;</span><span class="o">.</span><span class="n">format</span><span class="p">(</span><span>
```

```
<span class="bp">self</span><span class="o">.</span><span class="n">success</span><span
class="p">,</span> <span class="bp">self</span><span class="o">.</span><span
class="n">penalties</span><span>
```



```

class="n">total</span><span class="p">,</span> <span class="bp">self</span><span
class="o">.</span><span class="n">trials</span><span class="p">,</span> <span
class="bp">self</span><span class="o">.</span><span class="n">net_reward</span><span
class="p">,</span> <span class="bp">self</span><span class="o">.</span><span
class="n">penalties</span><span class="p">,</span> <span class="bp">self</span><span
class="o">.</span><span class="n">moves</span><span class="p">,</span> <span
class="nb">round</span><span class="p"></span><span class="nb">float</span><span
class="p"></span><span class="bp">self</span><span class="o">.</span><span
class="n">penalties</span><span class="p"></span><span class="o"></span><span
class="nb">float</span><span class="p"></span><span class="bp">self</span><span
class="o">.</span><span class="n">moves</span><span class="p">,</span> <span
class="mi">2</span><span class="p">)</span></span>

```

```

<span class="k">def</span> <span class="nf">_select_Q_action</span><span
class="p">(</span><span class="bp">self</span><span class="p">,</span> <span
class="n">state</span><span class="p">):</span></span>

```

```

<span class="n">best_action</span> <span class="o">=</span> <span
class="n">random</span><span class="o">.</span><span class="n">choice</span><span
class="p">(</span><span class="n">Environment</span><span class="o">.</span><span
class="n">valid_actions</span><span class="p">)</span></span>

```

```

<span class="k">if</span> <span class="bp">self</span><span class="o">.</span><span
class="n">_random_pick</span><span class="p">(</span><span class="bp">self</span><span
class="o">.</span><span class="n">epsilon</span><span class="p">):</span></span>

```

```

<span class="n">max_Q</span> <span class="o">=</span> <span class="bp">self</span><span
class="o">.</span><span class="n">_get_Q</span><span class="p">(</span><span
class="n">state</span><span class="p">,</span> <span class="n">best_action</span><span
class="p">)</span></span>

```

```

<span class="k">else</span><span class="p">:</span></span>

```

```

<span class="n">max_Q</span> <span class="o">=</span> <span class="o">-</span><span
class="mi">999999</span></span>

```

```

<span class="k">for</span> <span class="n">action</span> <span class="ow">in</span> <span
class="n">Environment</span><span class="o">.</span><span class="n">valid_actions</span><span
class="p">:</span></span>

```

```

<span class="n">Q</span> <span class="o">=</span> <span class="bp">self</span><span
class="o">.</span><span class="n">_get_Q</span><span class="p">(</span><span
class="n">state</span><span class="p">,</span> <span class="n">action</span><span
class="p">)</span></span>

```

<span class="k">if</span> <span class="n">Q</span> <span class="o">&gt;</span> <span class="n">max\_Q</span><span class="p">:</span></p>

<span class="n">max\_Q</span> <span class="o">=</span> <span class="n">Q</span></p>

<span class="n">best\_action</span> <span class="o">=</span> <span class="n">action</span></p>

<span class="k">elif</span> <span class="n">Q</span> <span class="o">==</span> <span class="n">max\_Q</span><span class="p">:</span></p>

<span class="k">if</span> <span class="bp">self</span><span class="o">.</span><span class="n">\_random\_pick</span><span class="p">(</span><span class="mf">0.5</span><span class="p">)</span></p>

<span class="n">best\_action</span> <span class="o">=</span> <span class="n">action</span></p>

<span class="k">return</span> <span class="p">(</span><span class="n">max\_Q</span><span class="p">,</span><span class="n">best\_action</span><span class="p">)</span></p>

<span class="k">def</span> <span class="nf">\_get\_Q</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">)</span></p>

<span class="k">return</span> <span class="bp">self</span><span class="o">.</span><span class="n">Q</span><span class="o">.</span><span class="n">get</span><span class="p">(</span><span class="n">state</span><span class="p">,</span><span class="n">action</span><span class="p">,</span><span class="bp">self</span><span class="o">.</span><span class="n">default\_Q</span><span class="p">)</span></p>

<span class="k">def</span> <span class="nf">\_random\_pick</span><span class="p">(</span><span class="bp">self</span><span class="p">,</span><span class="n">epsilon</span><span class="p">,</span><span class="o">=</span><span class="mf">0.5</span><span class="p">)</span></p>

<span class="k">return</span> <span class="n">random</span><span class="o">.</span><span class="n">random</span><span class="p">(</span><span class="o">&lt;</span> <span class="n">epsilon</span></p>

<span class="k">def</span> <span class="nf">run</span><span class="p">(</span><span class="p">)</span></p>

```
<span class="sd">&quot;&quot;&quot;Run the agent for a finite number of
trials.&quot;&quot;&quot;</span>
```

```
<span class="c"># Set up environment and agent</span>
```

```
<span class="n">e</span> <span class="o">=</span> <span class="n">Environment</span><span
class="p">(</span> <span class="c"># create environment (also adds some dummy traffic)</span>
```

```
<span class="n">a</span> <span class="o">=</span> <span class="n">e</span><span
class="o">.</span><span class="n">create_agent</span><span class="p">(</span><span
class="n">LearningAgent</span><span class="p">)</span> <span class="c"># create agent</span>
```

```
<span class="n">e</span><span class="o">.</span><span class="n">set_primary_agent</span><span
class="p">(</span><span class="n">a</span><span class="p">,</span> <span
class="n">enforce_deadline</span><span class="o">=</span><span class="bp">True</span><span
class="p">)</span> <span class="c"># set agent to track</span>
```

```
<span class="c"># Now simulate it</span>
```

```
<span class="n">sim</span> <span class="o">=</span> <span class="n">Simulator</span><span
class="p">(</span><span class="n">e</span><span class="p">,</span> <span
class="n">update_delay</span><span class="o">=</span><span class="mf">0.00000001</span><span
class="p">,</span> <span class="n">display</span><span class="o">=</span><span
class="bp">False</span><span class="p">)</span> <span class="c"># reduce update_delay to speed up
simulation</span>
```

```
<span class="n">sim</span><span class="o">.</span><span class="n">run</span><span
class="p">(</span><span class="n">n_trials</span><span class="o">=</span><span
class="mi">100</span><span class="p">)</span> <span class="c"># press Esc or close pygame window
to quit</span>
```

```
<span class="k">if</span> <span class="n">__name__</span> <span class="o">==</span> <span
class="s">&#39;__main__&#39;</span><span class="p">:</span>
```

```
<span class="n">run</span><span class="p">(</span></span>
```

```
</pre></div>
```

</div>

</div>

</div>

<div class="output\_wrapper">

<div class="output">

<div class="output\_area"><div class="prompt"></div>

<div class="output\_subarea output\_stream output\_stdout output\_text">

<pre>bestQValue

Simulator.run(): Trial 0

Environment.reset(): Trial set up with start = (6, 5), destination = (5, 1), deadline = 25

RoutePlanner.route\_to(): destination = (5, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'waypoint': 'right'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'right'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'green',  
'waypoint': 'right'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'right'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 5, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 1/1 of 1 trials (net reward: 25.0)

penalties/moves (penalty rate): 10/25 (0.4)

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 1

Environment.reset(): Trial set up with start = (7, 3), destination = (6, 6), deadline = 20

RoutePlanner.route\_to(): destination = (6, 6)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 2/2 of 1 trials (net reward: 47.5)

penalties/moves (penalty rate): 12/41 (0.29)

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 2

Environment.reset(): Trial set up with start = (1, 6), destination = (6, 5), deadline = 30

RoutePlanner.route\_to(): destination = (6, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 3/3 of 1 trials (net reward: 69.5)

penalties/moves (penalty rate): 12/55 (0.22)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 3

Environment.reset(): Trial set up with start = (3, 1), destination = (6, 6), deadline = 40

RoutePlanner.route\_to(): destination = (6, 6)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0



LearningAgent.update(): deadline = 28, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = forward, reward = -1.0

Environment.act(): Primary agent has reached destination!

success/total = 4/4 of 1 trials (net reward: 100.0)

penalties/moves (penalty rate): 14/74 (0.19)

LearningAgent.update(): deadline = 22, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 4

Environment.reset(): Trial set up with start = (7, 1), destination = (3, 1), deadline = 20

RoutePlanner.route\_to(): destination = (3, 1)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 5/5 of 1 trials (net reward: 118.0)

penalties/moves (penalty rate): 14/81 (0.17)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 5

Environment.reset(): Trial set up with start = (8, 1), destination = (3, 3), deadline = 35

RoutePlanner.route\_to(): destination = (3, 3)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 6/6 of 1 trials (net reward: 142.5)

penalties/moves (penalty rate): 16/98 (0.16)

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 6

Environment.reset(): Trial set up with start = (8, 3), destination = (2, 6), deadline = 45

RoutePlanner.route\_to(): destination = (2, 6)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 7/7 of 1 trials (net reward: 168.0)

penalties/moves (penalty rate): 17/114 (0.15)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 7

Environment.reset(): Trial set up with start = (4, 1), destination = (1, 5), deadline = 35

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 8/8 of 1 trials (net reward: 192.0)

penalties/moves (penalty rate): 17/125 (0.14)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 8

Environment.reset(): Trial set up with start = (2, 4), destination = (6, 5), deadline = 25

RoutePlanner.route\_to(): destination = (6, 5)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 9/9 of 1 trials (net reward: 217.5)

penalties/moves (penalty rate): 18/144 (0.13)

LearningAgent.update(): deadline = 7, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 9

Environment.reset(): Trial set up with start = (6, 4), destination = (1, 5), deadline = 30

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 10/10 of 1 trials (net reward: 239.5)

penalties/moves (penalty rate): 18/158 (0.11)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 10

Environment.reset(): Trial set up with start = (4, 5), destination = (1, 4), deadline = 20

RoutePlanner.route\_to(): destination = (1, 4)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 11/11 of 1 trials (net reward: 257.5)

penalties/moves (penalty rate): 18/162 (0.11)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 11

Environment.reset(): Trial set up with start = (1, 3), destination = (8, 3), deadline = 35

RoutePlanner.route\_to(): destination = (8, 3)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 12/12 of 1 trials (net reward: 281.5)

penalties/moves (penalty rate): 18/181 (0.1)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 12

Environment.reset(): Trial set up with start = (5, 4), destination = (2, 5), deadline = 20

RoutePlanner.route\_to(): destination = (2, 5)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 13/13 of 1 trials (net reward: 298.5)

penalties/moves (penalty rate): 19/189 (0.1)

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 13

Environment.reset(): Trial set up with start = (8, 2), destination = (4, 6), deadline = 40

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 14/14 of 1 trials (net reward: 324.5)

penalties/moves (penalty rate): 19/210 (0.09)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 14

Environment.reset(): Trial set up with start = (4, 1), destination = (2, 3), deadline = 20

RoutePlanner.route\_to(): destination = (2, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 15/15 of 1 trials (net reward: 347.0)

penalties/moves (penalty rate): 21/221 (0.1)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 15

Environment.reset(): Trial set up with start = (8, 2), destination = (2, 5), deadline = 45

RoutePlanner.route\_to(): destination = (2, 5)

LearningAgent.update(): deadline = 45, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 16/16 of 1 trials (net reward: 375.5)

penalties/moves (penalty rate): 23/240 (0.1)

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 16

Environment.reset(): Trial set up with start = (3, 5), destination = (6, 6), deadline = 20

RoutePlanner.route\_to(): destination = (6, 6)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 17/17 of 1 trials (net reward: 392.5)

penalties/moves (penalty rate): 24/252 (0.1)

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 17

Environment.reset(): Trial set up with start = (8, 4), destination = (4, 3), deadline = 25

RoutePlanner.route\_to(): destination = (4, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 18/18 of 1 trials (net reward: 412.5)

penalties/moves (penalty rate): 24/263 (0.09)

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 18

Environment.reset(): Trial set up with start = (1, 2), destination = (6, 1), deadline = 30

RoutePlanner.route\_to(): destination = (6, 1)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 19/19 of 1 trials (net reward: 429.5)

penalties/moves (penalty rate): 26/280 (0.09)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 19

Environment.reset(): Trial set up with start = (3, 1), destination = (5, 3), deadline = 20

RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0



Environment.act(): Primary agent has reached destination!

success/total = 20/20 of 1 trials (net reward: 447.5)

penalties/moves (penalty rate): 26/289 (0.09)

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 20

Environment.reset(): Trial set up with start = (2, 2), destination = (8, 4), deadline = 40

RoutePlanner.route\_to(): destination = (8, 4)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 21/21 of 1 trials (net reward: 475.0)

penalties/moves (penalty rate): 27/307 (0.09)

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 21

Environment.reset(): Trial set up with start = (2, 3), destination = (6, 1), deadline = 30

RoutePlanner.route\_to(): destination = (6, 1)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 22/22 of 1 trials (net reward: 497.0)

penalties/moves (penalty rate): 27/324 (0.08)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 22

Environment.reset(): Trial set up with start = (6, 1), destination = (3, 5), deadline = 35

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 23/23 of 1 trials (net reward: 532.0)

penalties/moves (penalty rate): 29/349 (0.08)

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 23

Environment.reset(): Trial set up with start = (5, 1), destination = (3, 5), deadline = 30

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 24/24 of 1 trials (net reward: 555.5)

penalties/moves (penalty rate): 30/360 (0.08)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 24

Environment.reset(): Trial set up with start = (2, 3), destination = (8, 4), deadline = 35

RoutePlanner.route\_to(): destination = (8, 4)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 25/25 of 1 trials (net reward: 582.5)

penalties/moves (penalty rate): 31/376 (0.08)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 25

Environment.reset(): Trial set up with start = (1, 3), destination = (6, 2), deadline = 30

RoutePlanner.route\_to(): destination = (6, 2)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 26/26 of 1 trials (net reward: 605.0)

penalties/moves (penalty rate): 33/397 (0.08)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 26

Environment.reset(): Trial set up with start = (4, 2), destination = (8, 3), deadline = 25

RoutePlanner.route\_to(): destination = (8, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0



LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 27/27 of 1 trials (net reward: 624.0)

penalties/moves (penalty rate): 34/405 (0.08)

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 27

Environment.reset(): Trial set up with start = (1, 4), destination = (8, 6), deadline = 45

RoutePlanner.route\_to(): destination = (8, 6)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 43, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 28/28 of 1 trials (net reward: 649.5)

penalties/moves (penalty rate): 35/423 (0.08)

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 28

Environment.reset(): Trial set up with start = (4, 5), destination = (8, 2), deadline = 35

RoutePlanner.route\_to(): destination = (8, 2)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 29/29 of 1 trials (net reward: 673.5)

penalties/moves (penalty rate): 35/434 (0.08)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 29

Environment.reset(): Trial set up with start = (2, 4), destination = (6, 1), deadline = 35

RoutePlanner.route\_to(): destination = (6, 1)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red',  
'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 30/30 of 1 trials (net reward: 697.5)

penalties/moves (penalty rate): 35/447 (0.08)

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 30

Environment.reset(): Trial set up with start = (2, 3), destination = (7, 6), deadline = 40

RoutePlanner.route\_to(): destination = (7, 6)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 31/31 of 1 trials (net reward: 734.5)

penalties/moves (penalty rate): 37/467 (0.08)

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 31

Environment.reset(): Trial set up with start = (7, 6), destination = (2, 1), deadline = 50

RoutePlanner.route\_to(): destination = (2, 1)

LearningAgent.update(): deadline = 50, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 49, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 48, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 47, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 46, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 32/32 of 1 trials (net reward: 764.5)

penalties/moves (penalty rate): 37/483 (0.08)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 32

Environment.reset(): Trial set up with start = (6, 2), destination = (2, 3), deadline = 25

RoutePlanner.route\_to(): destination = (2, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 33/33 of 1 trials (net reward: 789.0)

penalties/moves (penalty rate): 39/500 (0.08)

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 33

Environment.reset(): Trial set up with start = (8, 4), destination = (5, 3), deadline = 20

RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 34/34 of 1 trials (net reward: 806.0)

penalties/moves (penalty rate): 40/508 (0.08)

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 34

Environment.reset(): Trial set up with start = (8, 1), destination = (4, 6), deadline = 45

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0



LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 35/35 of 1 trials (net reward: 823.5)

penalties/moves (penalty rate): 41/521 (0.08)

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 35

Environment.reset(): Trial set up with start = (5, 6), destination = (5, 2), deadline = 20

RoutePlanner.route\_to(): destination = (5, 2)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 36/36 of 1 trials (net reward: 838.0)

penalties/moves (penalty rate): 43/531 (0.08)

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 36

Environment.reset(): Trial set up with start = (2, 4), destination = (7, 3), deadline = 30

RoutePlanner.route\_to(): destination = (7, 3)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 37/37 of 1 trials (net reward: 859.0)

penalties/moves (penalty rate): 45/548 (0.08)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 37

Environment.reset(): Trial set up with start = (3, 6), destination = (6, 2), deadline = 35

RoutePlanner.route\_to(): destination = (6, 2)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 38/38 of 1 trials (net reward: 883.0)

penalties/moves (penalty rate): 45/569 (0.08)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 38

Environment.reset(): Trial set up with start = (8, 1), destination = (7, 6), deadline = 30

RoutePlanner.route\_to(): destination = (7, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 39/39 of 1 trials (net reward: 905.0)

penalties/moves (penalty rate): 45/586 (0.08)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 39

Environment.reset(): Trial set up with start = (8, 1), destination = (4, 2), deadline = 25

RoutePlanner.route\_to(): destination = (4, 2)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 40/40 of 1 trials (net reward: 924.0)

penalties/moves (penalty rate): 46/596 (0.08)

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 40

Environment.reset(): Trial set up with start = (1, 6), destination = (8, 1), deadline = 60

RoutePlanner.route\_to(): destination = (8, 1)

LearningAgent.update(): deadline = 60, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 59, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 58, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 57, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 56, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 55, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 54, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 53, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 52, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 51, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 50, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 49, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 48, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 47, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 46, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 25, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 41/41 of 1 trials (net reward: 973.0)

penalties/moves (penalty rate): 48/638 (0.08)

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 41

Environment.reset(): Trial set up with start = (2, 2), destination = (3, 6), deadline = 25

RoutePlanner.route\_to(): destination = (3, 6)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 42/42 of 1 trials (net reward: 992.0)

penalties/moves (penalty rate): 49/648 (0.08)

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 42

Environment.reset(): Trial set up with start = (3, 4), destination = (8, 3), deadline = 30

RoutePlanner.route\_to(): destination = (8, 3)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 43/43 of 1 trials (net reward: 1013.0)

penalties/moves (penalty rate): 50/660 (0.08)

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 43

Environment.reset(): Trial set up with start = (6, 1), destination = (2, 1), deadline = 20

RoutePlanner.route\_to(): destination = (2, 1)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 44/44 of 1 trials (net reward: 1031.0)

penalties/moves (penalty rate): 50/665 (0.08)

LearningAgent.update(): deadline = 16, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 44

Environment.reset(): Trial set up with start = (2, 6), destination = (5, 1), deadline = 40

RoutePlanner.route\_to(): destination = (5, 1)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red',  
'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 45/45 of 1 trials (net reward: 1057.0)

penalties/moves (penalty rate): 50/686 (0.07)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 45

Environment.reset(): Trial set up with start = (6, 2), destination = (3, 5), deadline = 30

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 46/46 of 1 trials (net reward: 1083.5)

penalties/moves (penalty rate): 52/703 (0.07)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 46

Environment.reset(): Trial set up with start = (3, 4), destination = (4, 1), deadline = 20

RoutePlanner.route\_to(): destination = (4, 1)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 47/47 of 1 trials (net reward: 1107.0)

penalties/moves (penalty rate): 53/720 (0.07)

LearningAgent.update(): deadline = 4, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 47

Environment.reset(): Trial set up with start = (1, 6), destination = (5, 3), deadline = 35

RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 48/48 of 1 trials (net reward: 1131.0)

penalties/moves (penalty rate): 53/733 (0.07)

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 48

Environment.reset(): Trial set up with start = (1, 6), destination = (3, 4), deadline = 20

RoutePlanner.route\_to(): destination = (3, 4)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 49/49 of 1 trials (net reward: 1148.0)

penalties/moves (penalty rate): 54/740 (0.07)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0



Simulator.run(): Trial 49

Environment.reset(): Trial set up with start = (5, 2), destination = (5, 6), deadline = 20

RoutePlanner.route\_to(): destination = (5, 6)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 50/50 of 1 trials (net reward: 1171.5)

penalties/moves (penalty rate): 55/755 (0.07)

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 50

Environment.reset(): Trial set up with start = (3, 3), destination = (7, 5), deadline = 30

RoutePlanner.route\_to(): destination = (7, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 51/51 of 1 trials (net reward: 1196.5)

penalties/moves (penalty rate): 56/771 (0.07)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 51

Environment.reset(): Trial set up with start = (4, 6), destination = (3, 1), deadline = 30

RoutePlanner.route\_to(): destination = (3, 1)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 52/52 of 1 trials (net reward: 1222.0)

penalties/moves (penalty rate): 59/791 (0.07)

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 52

Environment.reset(): Trial set up with start = (3, 2), destination = (4, 6), deadline = 25

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 53/53 of 1 trials (net reward: 1246.0)

penalties/moves (penalty rate): 59/800 (0.07)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 53

Environment.reset(): Trial set up with start = (1, 3), destination = (8, 1), deadline = 45

RoutePlanner.route\_to(): destination = (8, 1)

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 54/54 of 1 trials (net reward: 1269.0)

penalties/moves (penalty rate): 61/821 (0.07)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 54

Environment.reset(): Trial set up with start = (4, 1), destination = (1, 4), deadline = 30

RoutePlanner.route\_to(): destination = (1, 4)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 55/55 of 1 trials (net reward: 1296.5)

penalties/moves (penalty rate): 62/834 (0.07)

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 55

Environment.reset(): Trial set up with start = (7, 3), destination = (4, 6), deadline = 30

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 56/56 of 1 trials (net reward: 1316.5)

penalties/moves (penalty rate): 64/859 (0.07)

LearningAgent.update(): deadline = 6, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 56

Environment.reset(): Trial set up with start = (5, 3), destination = (3, 5), deadline = 20

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 57/57 of 1 trials (net reward: 1334.5)

penalties/moves (penalty rate): 64/866 (0.07)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 57

Environment.reset(): Trial set up with start = (2, 2), destination = (5, 3), deadline = 20

RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 58/58 of 1 trials (net reward: 1356.5)

penalties/moves (penalty rate): 64/873 (0.07)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 58

Environment.reset(): Trial set up with start = (7, 2), destination = (5, 4), deadline = 20

RoutePlanner.route\_to(): destination = (5, 4)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 59/59 of 1 trials (net reward: 1374.5)

penalties/moves (penalty rate): 64/881 (0.07)

LearningAgent.update(): deadline = 13, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 59

Environment.reset(): Trial set up with start = (8, 5), destination = (2, 4), deadline = 35

RoutePlanner.route\_to(): destination = (2, 4)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 60/60 of 1 trials (net reward: 1397.5)

penalties/moves (penalty rate): 65/901 (0.07)

LearningAgent.update(): deadline = 16, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 60

Environment.reset(): Trial set up with start = (3, 5), destination = (1, 3), deadline = 20

RoutePlanner.route\_to(): destination = (1, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 61/61 of 1 trials (net reward: 1419.5)

penalties/moves (penalty rate): 65/912 (0.07)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 61

Environment.reset(): Trial set up with start = (5, 3), destination = (3, 5), deadline = 20

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'right'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 62/62 of 1 trials (net reward: 1435.0)

penalties/moves (penalty rate): 66/916 (0.07)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 62

Environment.reset(): Trial set up with start = (6, 2), destination = (1, 4), deadline = 35

RoutePlanner.route\_to(): destination = (1, 4)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red',  
'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 63/63 of 1 trials (net reward: 1459.0)

penalties/moves (penalty rate): 66/933 (0.07)

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 63

Environment.reset(): Trial set up with start = (8, 4), destination = (1, 1), deadline = 50

RoutePlanner.route\_to(): destination = (1, 1)

LearningAgent.update(): deadline = 50, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 49, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 48, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 47, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 46, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 45, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 43, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 41, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 40, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 64/64 of 1 trials (net reward: 1490.0)

penalties/moves (penalty rate): 70/974 (0.07)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 64

Environment.reset(): Trial set up with start = (8, 2), destination = (5, 5), deadline = 30

RoutePlanner.route\_to(): destination = (5, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 65/65 of 1 trials (net reward: 1512.0)

penalties/moves (penalty rate): 70/990 (0.07)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 65

Environment.reset(): Trial set up with start = (6, 1), destination = (7, 6), deadline = 30

RoutePlanner.route\_to(): destination = (7, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 66/66 of 1 trials (net reward: 1533.0)

penalties/moves (penalty rate): 71/1002 (0.07)

LearningAgent.update(): deadline = 19, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 66

Environment.reset(): Trial set up with start = (7, 6), destination = (3, 4), deadline = 30

RoutePlanner.route\_to(): destination = (3, 4)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 67/67 of 1 trials (net reward: 1552.5)

penalties/moves (penalty rate): 72/1020 (0.07)

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 67

Environment.reset(): Trial set up with start = (1, 1), destination = (3, 6), deadline = 35

RoutePlanner.route\_to(): destination = (3, 6)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 68/68 of 1 trials (net reward: 1568.5)

penalties/moves (penalty rate): 72/1027 (0.07)

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 68

Environment.reset(): Trial set up with start = (7, 5), destination = (3, 3), deadline = 30

RoutePlanner.route\_to(): destination = (3, 3)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 69/69 of 1 trials (net reward: 1589.5)

penalties/moves (penalty rate): 73/1041 (0.07)

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 69

Environment.reset(): Trial set up with start = (8, 1), destination = (3, 1), deadline = 25

RoutePlanner.route\_to(): destination = (3, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 70/70 of 1 trials (net reward: 1608.5)

penalties/moves (penalty rate): 74/1049 (0.07)

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 70

Environment.reset(): Trial set up with start = (8, 1), destination = (8, 6), deadline = 25

RoutePlanner.route\_to(): destination = (8, 6)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 71/71 of 1 trials (net reward: 1627.5)

penalties/moves (penalty rate): 75/1065 (0.07)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 71

Environment.reset(): Trial set up with start = (3, 6), destination = (7, 4), deadline = 30

RoutePlanner.route\_to(): destination = (7, 4)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 72/72 of 1 trials (net reward: 1649.5)

penalties/moves (penalty rate): 75/1074 (0.07)

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 72

Environment.reset(): Trial set up with start = (1, 5), destination = (5, 3), deadline = 30



RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 73/73 of 1 trials (net reward: 1671.5)

penalties/moves (penalty rate): 75/1084 (0.07)

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 73

Environment.reset(): Trial set up with start = (8, 4), destination = (1, 5), deadline = 40

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 74/74 of 1 trials (net reward: 1697.5)

penalties/moves (penalty rate): 75/1111 (0.07)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 74

Environment.reset(): Trial set up with start = (6, 4), destination = (1, 6), deadline = 35

RoutePlanner.route\_to(): destination = (1, 6)

LearningAgent.update(): deadline = 35, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 75/75 of 1 trials (net reward: 1721.5)

penalties/moves (penalty rate): 75/1127 (0.07)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 75

Environment.reset(): Trial set up with start = (6, 2), destination = (4, 6), deadline = 30

RoutePlanner.route\_to(): destination = (4, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 76/76 of 1 trials (net reward: 1743.5)

penalties/moves (penalty rate): 75/1143 (0.07)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 76

Environment.reset(): Trial set up with start = (1, 1), destination = (2, 6), deadline = 30

RoutePlanner.route\_to(): destination = (2, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 77/77 of 1 trials (net reward: 1757.5)

penalties/moves (penalty rate): 75/1145 (0.07)

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 77

Environment.reset(): Trial set up with start = (6, 5), destination = (1, 5), deadline = 25

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 78/78 of 1 trials (net reward: 1776.5)

penalties/moves (penalty rate): 76/1156 (0.07)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 78

Environment.reset(): Trial set up with start = (2, 5), destination = (5, 1), deadline = 35

RoutePlanner.route\_to(): destination = (5, 1)

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 79/79 of 1 trials (net reward: 1802.0)

penalties/moves (penalty rate): 77/1174 (0.07)

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 79

Environment.reset(): Trial set up with start = (7, 6), destination = (7, 2), deadline = 20

RoutePlanner.route\_to(): destination = (7, 2)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'left'}, action = forward, reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'right'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 80/80 of 1 trials (net reward: 1828.5)

penalties/moves (penalty rate): 79/1190 (0.07)

LearningAgent.update(): deadline = 5, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 80

Environment.reset(): Trial set up with start = (8, 5), destination = (5, 3), deadline = 25

RoutePlanner.route\_to(): destination = (5, 3)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 81/81 of 1 trials (net reward: 1848.5)

penalties/moves (penalty rate): 79/1195 (0.07)

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 81

Environment.reset(): Trial set up with start = (8, 6), destination = (2, 6), deadline = 30

RoutePlanner.route\_to(): destination = (2, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 82/82 of 1 trials (net reward: 1870.5)

penalties/moves (penalty rate): 79/1206 (0.07)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 82

Environment.reset(): Trial set up with start = (6, 4), destination = (1, 5), deadline = 30

RoutePlanner.route\_to(): destination = (1, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 83/83 of 1 trials (net reward: 1894.0)

penalties/moves (penalty rate): 80/1222 (0.07)

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 83

Environment.reset(): Trial set up with start = (8, 4), destination = (3, 5), deadline = 30

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 84/84 of 1 trials (net reward: 1916.0)

penalties/moves (penalty rate): 80/1239 (0.06)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 84

Environment.reset(): Trial set up with start = (2, 1), destination = (8, 5), deadline = 50

RoutePlanner.route\_to(): destination = (8, 5)

LearningAgent.update(): deadline = 50, inputs = {'light': 'red', 'waypoint': 'right'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 49, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 48, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 47, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 46, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 45, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 44, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 43, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 42, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 85/85 of 1 trials (net reward: 1941.0)

penalties/moves (penalty rate): 81/1249 (0.06)

LearningAgent.update(): deadline = 41, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 85

Environment.reset(): Trial set up with start = (3, 3), destination = (6, 6), deadline = 30

RoutePlanner.route\_to(): destination = (6, 6)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'right'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 86/86 of 1 trials (net reward: 1967.0)

penalties/moves (penalty rate): 81/1268 (0.06)

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 86

Environment.reset(): Trial set up with start = (7, 1), destination = (5, 4), deadline = 25

RoutePlanner.route\_to(): destination = (5, 4)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'right'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 10, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 9, inputs = {'light': 'green', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 8, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 7, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 6, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 5, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 4, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 3, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 2, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 1, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

success/total = 86/87 of 1 trials (net reward: 1983.5)

penalties/moves (penalty rate): 84/1294 (0.06)

LearningAgent.update(): deadline = 0, inputs = {'light': 'green', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.step(): Primary agent ran out of time! Trial aborted.

Simulator.run(): Trial 87

Environment.reset(): Trial set up with start = (8, 6), destination = (7, 2), deadline = 25

RoutePlanner.route\_to(): destination = (7, 2)

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 87/88 of 1 trials (net reward: 2003.5)

penalties/moves (penalty rate): 84/1310 (0.06)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 88

Environment.reset(): Trial set up with start = (8, 5), destination = (4, 1), deadline = 40

RoutePlanner.route\_to(): destination = (4, 1)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 88/89 of 1 trials (net reward: 2029.5)

penalties/moves (penalty rate): 84/1327 (0.06)

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 89

Environment.reset(): Trial set up with start = (8, 6), destination = (3, 3), deadline = 40

RoutePlanner.route\_to(): destination = (3, 3)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 89/90 of 1 trials (net reward: 2056.0)

penalties/moves (penalty rate): 86/1344 (0.06)

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 90

Environment.reset(): Trial set up with start = (7, 3), destination = (1, 3), deadline = 30

RoutePlanner.route\_to(): destination = (1, 3)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 90/91 of 1 trials (net reward: 2078.5)

penalties/moves (penalty rate): 88/1359 (0.06)

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 91

Environment.reset(): Trial set up with start = (8, 5), destination = (7, 1), deadline = 25

RoutePlanner.route\_to(): destination = (7, 1)

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 91/92 of 1 trials (net reward: 2098.5)

penalties/moves (penalty rate): 88/1372 (0.06)

LearningAgent.update(): deadline = 13, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 92

Environment.reset(): Trial set up with start = (5, 6), destination = (5, 2), deadline = 20

RoutePlanner.route\_to(): destination = (5, 2)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 92/93 of 1 trials (net reward: 2116.5)

penalties/moves (penalty rate): 88/1379 (0.06)

LearningAgent.update(): deadline = 14, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 93

Environment.reset(): Trial set up with start = (8, 2), destination = (3, 5), deadline = 40

RoutePlanner.route\_to(): destination = (3, 5)

LearningAgent.update(): deadline = 40, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red',  
'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red', 'waypoint': 'left'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red', 'waypoint': 'left'}, action = right, reward = -0.5

LearningAgent.update(): deadline = 25, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 24, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 23, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 22, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 21, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 93/94 of 1 trials (net reward: 2146.0)

penalties/moves (penalty rate): 91/1408 (0.06)

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 94

Environment.reset(): Trial set up with start = (8, 3), destination = (6, 1), deadline = 20

RoutePlanner.route\_to(): destination = (6, 1)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = left, reward = -1.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0



LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 94/95 of 1 trials (net reward: 2162.0)

penalties/moves (penalty rate): 93/1417 (0.07)

LearningAgent.update(): deadline = 12, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 95

Environment.reset(): Trial set up with start = (8, 5), destination = (5, 6), deadline = 20

RoutePlanner.route\_to(): destination = (5, 6)

LearningAgent.update(): deadline = 20, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 11, inputs = {'light': 'red', 'waypoint': 'left'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 95/96 of 1 trials (net reward: 2180.0)

penalties/moves (penalty rate): 93/1428 (0.07)

LearningAgent.update(): deadline = 10, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 12.0

Simulator.run(): Trial 96

Environment.reset(): Trial set up with start = (2, 1), destination = (4, 3), deadline = 20

RoutePlanner.route\_to(): destination = (4, 3)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = forward, reward = -1.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 13, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 12, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 96/97 of 1 trials (net reward: 2197.0)

penalties/moves (penalty rate): 94/1438 (0.07)

LearningAgent.update(): deadline = 11, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 97

Environment.reset(): Trial set up with start = (5, 3), destination = (8, 4), deadline = 20

RoutePlanner.route\_to(): destination = (8, 4)

LearningAgent.update(): deadline = 20, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 19, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 18, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 17, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 16, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 15, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 14, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 97/98 of 1 trials (net reward: 2215.0)

penalties/moves (penalty rate): 94/1446 (0.07)

LearningAgent.update(): deadline = 13, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 12.0

Simulator.run(): Trial 98

Environment.reset(): Trial set up with start = (8, 6), destination = (2, 4), deadline = 40

RoutePlanner.route\_to(): destination = (2, 4)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 36, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red', 'waypoint': 'right'}, action = right, reward = 2.0

Environment.act(): Primary agent has reached destination!

success/total = 98/99 of 1 trials (net reward: 2241.0)

penalties/moves (penalty rate): 94/1457 (0.06)

LearningAgent.update(): deadline = 30, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 12.0

Simulator.run(): Trial 99

Environment.reset(): Trial set up with start = (1, 6), destination = (7, 4), deadline = 40

RoutePlanner.route\_to(): destination = (7, 4)

LearningAgent.update(): deadline = 40, inputs = {'light': 'green', 'waypoint': 'left'}, action = left, reward = 2.0

LearningAgent.update(): deadline = 39, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 38, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 37, inputs = {'light': 'green', 'waypoint': 'forward'}, action = left, reward = -0.5

LearningAgent.update(): deadline = 36, inputs = {'light': 'green', 'waypoint': 'right'}, action = right, reward = 2.0

LearningAgent.update(): deadline = 35, inputs = {'light': 'red', 'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 34, inputs = {'light': 'green', 'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 33, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 32, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 31, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 30, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 29, inputs = {'light': 'red',  
'waypoint': 'forward'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 28, inputs = {'light': 'green',  
'waypoint': 'forward'}, action = forward, reward = 2.0

LearningAgent.update(): deadline = 27, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 26, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

LearningAgent.update(): deadline = 25, inputs = {'light': 'red',  
'waypoint': 'left'}, action = None, reward = 0.0

Environment.act(): Primary agent has reached destination!

success/total = 99/100 of 1 trials (net reward: 2264.5)

penalties/moves (penalty rate): 95/1474 (0.06)

LearningAgent.update(): deadline = 24, inputs = {'light': 'green',  
'waypoint': 'left'}, action = left, reward = 12.0

</pre>

</div>

</div>

</div>

</div>

</div>

<div class="cell border-box-sizing code\_cell rendered">

```
<div class="input">
<div class="prompt input_prompt">In&nbsp;&nbsp;[&nbsp;&nbsp;]:</div>
<div class="inner_cell">
  <div class="input_area">
<div class=" highlight hl-ipython2"><pre>
</pre></div>

</div>
</div>
</div>

</div>
  </div>
</div>
</body>
</html>
```