３：模擬問題

　問１：次のプログラム中の空白に入れる正しい答えを、解答群の中から選べ。

　　ある施設の入場料は、0歳から3歳までは100円、4歳から9歳までは300円、10歳以上は500円である。

　　関数feeは、年齢を表す0以上の整数を引数として受け取り、入場料を返す。

　　[プログラム]

解答群

ア　(ageが4以上)and(ageが9より小さい)

イ　(ageが4と等しい)or(ageが9と等しい)

ウ　(ageが4より大きい)and(ageが9以下)

エ　ageが4以上

オ　ageが4より大きい

カ　ageが9以下

キ　ageが9より小さい

〇整数型：fee( 整数型：age )

整数型：ret

if( ageが3以下 )

ret ← 100

elseif( )

ret ← 300

else

ret ← 500

endif

return ret

　問２：次のプログラム中の（a）と（b）に入れる正しい組み合わせを解答群の中から選べ。

　　　　ここで、配列の要素番号は１から始まる。

　　次のプログラムは整数型の配列arrayの要素の並びを逆順にする

　　[プログラム]

整数型の配列：array ← [1 , 2 , 3 , 4 , 5]

整数型：right , left

整数型：tmp

for( leftを1から(arrayの要素数÷2の商)まで1ずつ増やす )

right ← ( a )

tmp ← array[right]

array[right] ← array[left]

解答群

|  |  |  |
| --- | --- | --- |
|  | a | b |
| ア | arrayの要素数―left | array[left] |
| イ | arrayの要素数―left | array[right] |
| ウ | arrayの要素数―left＋１ | array[left] |
| エ | arrayの要素数―left＋１ | array[right] |

( b ) ← tmp

endfor

　問３：次のプログラム中の（a）と（b）に入れる正しい組み合わせを解答群の中から選べ。

　　手続きappendは、引数で与えられた文字を単方向リストに追加する手続である。単方向リストの各要

素は、クラスListElementを用いて表現する。クラスListElementの説明を図に示す。ListElement型の変数はクラスListElementのインスタンスの参照を格納するものとする。大域変数listHeadは、単方向リストの先頭の要素の参照を格納する。リストが空の時は、listHeadは未定義である。

●図　クラスListElementの説明

|  |  |  |
| --- | --- | --- |
| メンバ変数 | 型 | 説明 |
| val | 文字型 | リストに格納する文字列 |
| next | ListElement | リストの次の文字を保持するインスタンスの参照。  初期状態は未定義である。 |

|  |  |
| --- | --- |
| コンストラクタ | 説明 |
| ListElement( 文字型：qVal ) | 引数qValでメンバ変数valを初期化する。 |

[プログラム]

大域：ListElement：listHead ← 未定義の値

〇append( 文字型：qVal )

解答群

|  |  |  |
| --- | --- | --- |
|  | a | b |
| ア | 未定義 | curr |
| イ | 未定義 | curr .next |
| ウ | 未定義 | listHead |
| エ | 未定義でない | curr |
| オ | 未定義でない | curr .next |
| カ | 未定義でない | listHead |

ListElement：prev , curr

curr ← ListElement( qVal )

if( listHeadが　( a )　 )

listHead ← curr

else

prev ← listHead

while( prev.next が 未定義でない )

prev ← prev.next

endwhile

prev.next ← ( b )

endif

　問４：次の記述中( a )～( c )に入れる正しい答えの組み合わせを解答群の中から選べ。

　　　　ここで、配列の要素番号は1から始まる。

　　要素の多くが０の行列を疎行列という。次のプログラムは、二次元配列に格納された行列のデータ量を削減するために、疎行列の格納に適したデータ構造に変換する。

　　関数transformSparseMatrixは、引数matrixで二次元配列として与えられた行列を、整数型配列の配列に変換して返す。関数transformSparseMatrixを

transformSparseMatrix( { {3,0,0,0,0} , {0,2,2,0,0} , {0,0,0,1,3} , {0,0,0,2,0} , {0,0,0,0,1} } )

として呼び出したときの戻り値は、( {( a )} , {( b )} , {( c )} )である。

　[プログラム]

〇整数型配列：transformSparseMatrix(整数型の二次元配列：matrix)

整数型：i , j

整数型配列の配列：sparseMatrix

sparseMatrix ← { { } , { } , { }} /\* 要素数０の配列を三つ要素に持つ配列 \*/

for( iを１からmatrixの行数まで1ずつ増やす )

for( jを１からmatrixの列数まで1ずつ増やす )

if( matrix[i , j] が０でない )

sparseMatrix[１]の末尾に iの値 を追加する

sparseMatrix[２]の末尾に jの値 を追加する

sparseMatrix[３]の末尾に matrix[i , j]の値 を追加する

endif

endfor

endfor

return sparseMatrix

解答群

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b | c |
| ア | １,２,２,３,３,４,５ | １,２,３,４,５,４,５ | ３,２,２,１,２,３,１ |
| イ | １,２,２,３,３,４,５ | １,２,３,４,５,４,５ | ３,２,２,１,３,２,１ |
| ウ | １,２,３,４,５,４,５ | １,２,２,３,３,４,５ | ３,２,２,１,２,３,１ |
| エ | １,２,３,４,５,４,５ | １,２,２,３,３,４,５ | ３,２,２,１,３,２,１ |

問５：次の記述中( a )に入れる正しい答えを解答群の中から選べ。

　手続 Split は、要素数 num 個の配列 list を 2 つに分割して、前半部を配列 slist1 に格納し、後半部を配列 slist2 に格納する。配列の添字は、 0 から始まる。

　[プログラム]

◯Split(整数型の配列: list, 整数型: num, 整数型の配列: slist1, 整数型の配列: slist2)

整数型: num1, num2, i

num1 ← num ÷ 2 /\* slist1の要素数計算 \*/

num2 ← num - num1 /\* slist2の要素数計算 \*/

解答群

ア　list[i]

イ　list[num + i]

ウ　list[num1 + i]

エ　list[num2 + i]

for (iを0からnum1まで1ずつ増やす)

slist1[i] ← list[i]

endfor

for (iを0からnum2まで1ずつ増やす)

slist2[i] ← ( a )

endfor

　問６：次の記述中( a )に入れる正しい答えを解答群の中から選べ。

関数 MatchCounter は、長さが Textlen の文字列 SourceText の中で、長さが Patlen の文字列 Pattern と一致する部分文字列の出現回数を、以下の手順で数える。

* 1. 一致する部分文字列の出現回数を数える変数 Counter の値を 0 に初期化する。
  2. SourceText の比較開始位置を先頭から順に 1 文字ずつ後ろにずらしながら、その比較開始位置から始まる長さ Patlen の文字列と Pattern が一致するかどうかを調べ、一致したら出現回数 Counter の値に 1 を加算する。
  3. Counter の値を返す。

　ここで、 0 < Patlen ≦ Textlen であり、配列の添字は 0 から始まる。プログラム中の( a )に入る正しい答えを、解答群の中から選べ。

[プログラム]

◯整数型: MatchCounter(文字型の配列: SourceText,

整数型: Textlen,文字型の配列: Pattern, 整数型: Patlen)

　整数型: Counter, i, j, k

論理型: Matchflg

Counter ← 0

i ← 0

while ( ( a ) )

j ← i

k ← 0

Matchflg ← true

while (k が Patlen より小さく かつ Matchflg が true)

if ( SourceText[j] と Pattern[k] が等しい)

j ← j + 1

k ← k + 1

else

Matchflg ← false

endif

endwhile

解答群

ア　i と Patlen の和が Textlen 以下

イ　i と Patlen の和が Textlen より小さい

ウ　i と Textlen の和が Patlen 以下

エ　i と Textlen の和が Patlen より小さい

if (k が Patlen と等しい)

Counter ← Counter + 1

endif

i ← i + 1

endwhile

return Counter

　問７：次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。

　　※プログラム内で使われている単語は　**赤太字**　で表しています

　　　以下は、購入した商品の中から、値引きの対象となる商品を見つけて、その数量を求めるプログラムである。 購入した商品は、 **購入[ ]** という構造型（**品番**、**品名**、**数量**から構成されたデータ型）の配列に格納され、要素数は **購入行数** に格納されている。 値引きの対象となる商品は、 **対象[ ]** という構造型（**品番**、**数量**から構成されたデータ型）の配列に格納され、要素数は **対象行数** に格納されている。 ぞれぞれの配列は、 **品番** の昇順に整列されていて、 **品番** が一致した場合は、 **購入[ ]** の **数量** を、 **対象[ ]** の **数量** に格納する。 **購入[ ]** 、 **購入行数** 、 **対象[ ]** 、 **対象行数** は、以下に示したプログラムとは別の場所で宣言された大域的変数である。 **購入[ ]** と **対象[ ]** の添字は、 1 から始まる。 プログラム中の ( a ) ( b )に入る正しい答えを、解答群の中から選べ。

　　[格納されたデータの例]

![テーブル

自動的に生成された説明](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDkRXhpZgAATU0AKgAAAAgABAE7AAIAAAAOAAAISodpAAQAAAABAAAIWJydAAEAAAAMAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAOW6g+eArCDkv4rlk4kAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzk3AACSkgACAAAAAzk3AADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIwOjA2OjA4IDE1OjE0OjU4ADIwMjA6MDY6MDggMTU6MTQ6NTgAAACDXixwIADKT8lUAAD/4QsgaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyIvPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIj48eG1wOkNyZWF0ZURhdGU+MjAyMC0wNi0wOFQxNToxNDo1OC45NzQ8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIj48ZGM6Y3JlYXRvcj48cmRmOlNlcSB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6bGk+5bqD54CsIOS/iuWTiTwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCAEdAsEDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD6RooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCK7u7ews5bq+uIra2hUvLNM4RI1HUljwB7msD/hY/gj/AKHLw/8A+DSD/wCKpvxHsrnUvhl4isrCCS4ubjT5o4oY1yzsVIAA7mvPdB1TR9P8OabZah8Hdfnu7a0iinlGgW7B3VAGbJfJyQTk80AesaTr+j6/FJLoWrWOpxxMFkezuUmCE9iVJwas3F9aWYzd3UMGBn97IF49ea84+EFhc2+q+ML5/D95oNnqGopNZ2t3bCBgnl4+6CQOfQ1n/GHQL3XtJu7i88PeG7VbFke116/1BQ8ao24B0a3OUJJBj3EHNAHoF7468J6cD9u8T6PbkdRJfRA/luzU/h/xXofiqGeXw7qcGoR27hJHhJIBIyOe/wBRxwfSvmmz8d3ll48vfEtl8PrOyibQkHlZIhgjM2BelBDvKZ6gLu285xXsvwr/AOEms/D+nwXkOn6rZXBeafWYtca5MjNk5SPyQoXOBtDDA9SOQD0iiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA56f4geDba4kgufFuhQzRMUkjk1KFWRgcEEFsgg9qsaZ4x8M61eiz0fxFpOoXJUsILW+ilcgdTtVicV5B4VEPhuXX7fxJ8MNb1m4uNbu7mG7h0eKdWidvlwzsD2J9Oa1NBg/tT41aNq2keBtU8N6da6dcQ3D3enJbK7tgr9wkH8aAPZKyLTxTpF94pvvD1ncmbUbCJJbmNI2KxBvugvjaGPXbnNZXje78YFbbS/A9hAs96GEur3Ui+VYAY52cl2OeBjGRzXjNnB4K8Lade6P8RPBpuvGkOXj83dOdbkkYhZIpPdjyCPlHqQQAD6SorhfhJ4FPgbweYrqKOLUtRmN5eRwjEcTt0jUf3VGB+dd1QAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFYM3jXQIr6C0GoJJPNqB0wJEpbbchdxjbH3TgZyeK3icDNfK3im21fVfGdld+D9S1DVbe71dYn1K6itrVby8WF1xEVhAwqKylyCuWHHGaAPoeXx74ai0+wvv7UjktdRvxp1tLCjSCS4JYbPlBPVSM9Peuir5dOoPqHjTwP4OsdT1fToNKuz5jXun2ySaddFWEMW1UEbHMbMCQdwkzn09P8Ah3rHii81HxZeaprt74hsNGvZdPtbSOztonuXj2lmBUIM5yoywU556UAdte+OfCWm3stnqPijRbS6hbbJBPqESOh9CpbINQf8LH8Ef9Dl4f8A/BpB/wDFV4XqWtWmneNPEF6mp3umzahd+bPZTXsUDwMBt2kQatDk8fxLmqGs6/aa3o11ptz4juFiuUKOf7T8zA/3ZNbZD9GUj2oA+pYpY5oUlhdZI3UMjochgehB7inVzngbWBq3hi0MWn3lrBBBFHDNcmErcqFADoYpJARx610dABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAeWeObW81z4yeHNAXXdZ0uxuNNuZpV0u+e3LsrDBO3g/lXA+KL7W/Cul/FnTbDxPr066Quj/Yri61KSSaHzWVpNr5yudxBxjI4Nex+K/hj4b8ba9Y6p4kt5bw2ULwx2xk2xMGOcsAN2R2wwHtWQ/wO8Hpomv6Tp8d5p1nrwtxcx20w/d+Q+9fL3q2MnrnPU9KAOZ8aeG7vwVceD77S/F/iq4a98SWVnPDe6xLLG8TliyleM52gc9s1W+MnhqaPxf4avjqOs6m+oaoY47EeRKkCiMkCGKVRGWzzl8npzxXZj4HeAoL/T73TtHNhdafdx3cUtvM2WZDkK24kFc4JHHTrXQeKvBOk+Mn01tZN0P7NnM8P2a4aEliMcsuGHHoQfegDxeC98UH4vXluP8AhKvtP9ipuK6fpP2wx+bjDf8ALPyvTB3Z7YrU+DHhhj408TXv23WNObTtSCNp5MMEcwaPOJoYR5W4HBBTFegj4S+EFv2vha6kLto/KNwNbvfMKZzt3ednGecVp+FfBOk+DptSfRjdH+0phNN9puGmIYDHDNljx6kmgDoaKKKACiiigAooooAKKKKACiiigAooooAKKKKACvPPjZeXtl4AiOm6hd6fLNqVrC09nO0UgVpAGAZTkcGvQ6wPGfg7TfHXh/8AsbWnnW0M0czeQwVmKHIBJB4Pfv70Aee3mi3fg34y+A7Ky8UeJL+01Vr/AO1W+papJPG/lwAr8p46sTznoPSuY8LaJqOs/s+XPjS78aeLhqsWn31woTWpRHvhMuz5c5x8gzzXp+l/BzwXoXifTte0LTG06808yFBBKxSTehQ7w2c4BOMY61k2f7PPw9tNCOnSaXNdy7HX7dPOfP8AmJwcrhcjPHy9hnNAEmmaz46n+HXhFPC2m2uoXV/pEEtzq2qXh2wP5aZLr9+RmznIPY5rkPBvgSbxP8TfFj+M/EGqX+saHLbRQX9lcvaFBJGWZUCH5V7YH9a9t0XSbfQdCsdJsS5trG3S3iMhy2xFCjJ7nArlLn4R+F77xNquu341Ge61R0eVV1CWBEKLtG0RFCeP7xb2xQBy/jDQp/B3iPwVLpXiTxJKL7X4LW4iu9YnmjkjIJKlWbBzgV7BXCN8HPCg1bS9RtRqVvcaZdpdxf8AEymmV2XoGWVnGPpg+9d3QAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFfMvjDWNU8OeMNT1bxPr2vWs2kQyweGLm40m3CXkpX97yIdgBbbhsZ25Oe1fTVcnafDPwtBqE9/d6e2q3k4ZWn1WZ7tgjZyg8wkKuCRgDpQB5r49sJdK+F9qdX1ZtR8b6pdWd7p/2a0gEz3kYXG1YkXcigsNzZwCOTkCu0+B76a3wr08addNcXO531EyDEq3TMTKrjqCCcDPUAGtXw38MfCfhPV5dS0XTPLumUJG8sjS/Z0/uR7idqkkk47k1ZTwFoMGuarq9lBPY3urQ+VeSWdy8Pmc53/KRh+fvDn8SaAPI49R1vQPEniRtJ1TUIrbUdWmvFS3srlAN2F536XPk/L1V9vTA7nk/Aesa8fhHDpFlqV/BaTw3EQjSCV4gHdwfuaZLkZJziYnryp4Hv/8AwrrTf+gz4n/8KK8/+OVW074T+H9IsY7LSb3X7G0jzst7bXbqONMnJwqyADJJP40AXvhra/YfhnoNmZPMa2s0hZ/Kkiyy/KflkVXAyD1UV1FZeh6BBoMMsdtealdCVgxN/fy3RXH90yMSB7CtSgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiimmRBIIy67yMhc8keuKAHUU1JElQPG6up6MpyDTqACiiigAooooAKKKKAAkDqaTcPUfnXmFv4K8N+KPiz4zk8Q6Naag8IsRG08e4rmE5/kPyrd/4VB8Pv+hS0z/vzQB2W4eo/OjcPUfnXG/8ACoPh9/0KWmf9+aP+FQfD7/oUtM/780AdluHqPzo3D1H51xv/AAqD4ff9Clpn/fmj/hUHw+/6FLTP+/NAHZbh6j86Nw9R+dcb/wAKg+H3/QpaZ/35o/4VB8Pv+hS0z/vzQB2W4eo/OjcPUfnXG/8ACoPh9/0KWmf9+a5TW/hj4Kg+J/hWwh8NaelrdWuoPPCIvlkKCHYT9NzY+tAHru4eo/OjcPUfnXG/8Kg+H3/QpaZ/35o/4VB8Pv8AoUtM/wC/NAHZbh6j86Nw9R+dcb/wqD4ff9Clpn/fmmr8Jfh27MqeFdKZk4YCIEr9aAO03D1H50bh6j8643/hUHw+/wChS0z/AL80f8Kg+H3/AEKWmf8AfmgDstw9R+dG4eo/OuLf4SfDyNC8nhTSkUdWaIACnD4Q/D1lBXwnpZB5BEPWgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuN/4VB8Pv+hS0z/vzR/wqD4ff9Clpn/fmgDstw9R+dG4eo/OuC1X4S+AYdGvZYvCmmq6W8jKwh5BCnBrK8BfC3wPqPw58OXt94Y06e5udLtpZpXiyzu0Sksfck0Aepbh6j86Nw9R+dcb/AMKg+H3/AEKWmf8Afmj/AIVB8Pv+hS0z/vzQB2W4eo/OjcPUfnXG/wDCoPh9/wBClpn/AH5o/wCFQfD7/oUtM/780AdluHqPzo3D1H51xv8AwqD4ff8AQpaZ/wB+aP8AhUHw+/6FLTP+/NAHZbh6j86Nw9R+dcb/AMKg+H3/AEKWmf8Afmj/AIVB8Pv+hS0z/vzQB2YIPQ0V5v4Z8NaL4Y+NWqWnh/TbfT4G8P20jRwJtBY3EwJ/EKv5CvSKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD5n8W312NE+It+LfxJLeWOtzRWmp2upultZoDF8hjE6/3j0jb7w544Nf8AFcK/GCC6t/E2pSxRWdxGtzDrOkjaDIp2o7/Iqcfdk/ecD0NN8XaWX0r4i2Mlj4oOrXmtySWUFpDefZZ0YxbWZUHlMThsFuThfaofH+papqR8TR+bqEtlY63bQW/2WSySCMboiAQw80tk8EfLnGeM0Adf+zprMU+jjTm1a5kkRZ3SylvrN4xmdiWWKP8AfA88l/lO4kcEV6d4o8JP4jnhnXxNr2iiBCpTSrtIVfnOW3I2SK4z4V6nqM/jrxTpeqy3ZSxis2totQa0eePeshc77cBCDhenTjNbPxc8Uy6J4TOkaOfM1/Xm+wadAh+cs/ytJ7BQc56A4oAyPgV4lbX9M8SQ/wBuXutwWesOtndXzlpTbFF8vOQP7rdhzngV3ni6WSDwTrksLtHJHp1wyOhwVIjYgg9jXlWnaTB8DvGujSSOE8O67Yw6dqFyRhYb2IfJK56AOCw9ByTXqfjP/kQ9f/7Btz/6KagDD+HukGz+Gun3+lzzvqup6TbzvLqN5Pcxmcwghirudq7mJITbkfQYyPh74g8WXXxO8YeHvFup2t6NKjtGhW0thFGhlQudv8R4IHzMeldX8Pf+SY+F/wDsD2n/AKJSuQ8Gf8nFfEj/AK46b/6TigC14002LS/H/hHUrC4vobjUdZEN0ov5jFInkSHb5RfYBlQeFHSvR64P4if8jR4C/wCw9/7by13lAHF+Gf8Akqnjj6af/wCiWrtK4vwz/wAlU8cfTT//AES1dpQAj7tjeXjdj5d3TNeXeF9f8Zr8c7zwz4q1WxurRdCN/HBYWvlRo/nIgOWy5OC3ViOelepV5dbf8nX3n/YpD/0pSgDp/iFN4ntPCt3eeEr6wspLW3mmmku7dpWwqbh5YztB4P3gR04p3w11i+8QfDXQtV1ef7Re3dqsk0uxV3tk84UAD8BV7xn/AMiHr/8A2Dbn/wBFNWJ8Hf8Akjnhn/rxX+ZoAwdd1/xro3xm8K6XearYf2HrVxdqlra2mH8uJNy+Y77ju+ZeV2jg8V6nXl3xE/5Ld8Lf+u2o/wDomOvUaACuL8Qf8lg8Gf8AXlqf8reu0ri/EH/JYPBn/Xlqf8regDtK8n+LnjXxNo+rwad4JuEjuNO06XWdTVolk326OqhPmBxn5+Rg8DmvWK8P8MXniLxD4s8Y+KNJ8M2uuabqs50y3kuNVFuBbwgoQq+W2QxJOc9aAPWzrE194Vi1jw5ax6k9zbx3FrDJP5Kyq4BGXw23g56GvJtG1vxD4N+Ll/8A2/4fs4j4wmS4fytVD/2fBbx4klc+WAVwc5O3pirfwx8Qa7onwr1HQo9Hk1XxD4Zvm077Cs6jcrNmNyxx8gDYyB0TOK5zwhp3je6+Kniu91QaXr91E9tp+rafMgWNraZCxWF2zhUzypBDjOecUAdP428TaDP8aPhtdQ63p0lvbNqXnzJdxlIt0ChdxzgZPAz1rv8AXPGuiaL4Ov8AxJ9vtrqxs42JeCZXDv2jBBxuJIGPevOfGfgvwta/GX4c2Vr4a0eG0vG1L7Tbx2ESxz7YFK71C4bBORnpXbeI/B2PDMOmeCtE8MW4W9S5a31CxP2cEA/OqRbcSZCc+mRQB5DqFx8QtO+Beu2/iDws0seoxy3l1qU+sKWiErBsCAqSABgbd3rXoOheKPiHH4d01Lf4cW80S2sQSQ+IYl3rsGDjy+MjtWJ8TU+Jw+GOvHXJ/CTaf9kb7QLOC5EpXI+6WcjP1rb0KP4s/wDCO6b9lufBgg+yxeWJLe73bdgxnD4zigC748lv/wC3vDUT6nqWlWF+ZoLx7G5SMQP5e5GYlCSucrkMuMjrxXnmg/aJ9a8MRyvc3PiV4Lq71WW81a4nitoUDJiJfN2lnBBAO7aDyO1eieJb+2hsdLtPHfgqXxVqHlNIzaXov2y2ifOCFMmdhIA6mvL/AIapBqE114uh+HtxJfTtLbxS2VtbWljYRglSUXeGLgZ3MQW7A0AeofA2WSb4KeHZJnaRzFJlnOSf3z96vfE278U6b4OvtS8JX9hZfYLO4ubh7q3MshCJuAj52g8N94EdKyPgre22nfAPQry/njtraC3meWaVgqoomfJJPQe9dF8RHWT4UeKXjYOjaJdlWU5BHkPyKAGeDfEE138K9J8Qa7P5sz6Wl3dzbFXcdm5mwAAO/AxXAw+KPiJf/Dif4jW+o2UFuiyXsWgNaKyNaITndN9/ftUtkce3PG9oNnNqH7McFlagtNc+GGijAHJZrcgfqap6Bq9ov7KyXvmAww+HZYmOf41iZCv13DFAFzxL8Qr6/tvB2n+CjFBqPi9fOhuLlN4tIFjEkjlOhYA8A8cGptD8Q+IND+Jw8G+KtQj1eK/sjeabqAt1hkJQ4kidV+XIHIIA/wAOF8MWs2i+Kfg5LqCvGlxotxbqWXAWRot6qfQkMBj1rrfE4+2/tI+B4bc7n0/T766uAP4Y3QxqT/wKgCLTtc8beP7vxDf+FdYtdG03SrySx0+F7RJvtssf3mkZuVQkjG3B59jnb8L/ABO03UvhhZ+LPELjTVLi2u1EbOI5/M8sgBQTgtjHoCM1k/Ad1tfA+q6fO2y403WryC5D8bGDbufbB/nXmV2pl/ZR1y+w3kajrzXMG4dUNyq/zU0Ae4WvxS8FXviJdDtfENrJfuxVFAby3YdQsuNjH2DZp2k/E7wbrniEaJpWvW9zqDAmONVcLLjOdjkBX6H7pPQ1x/xz0XTI/Avh2yjsLdLWHXbSGOFYwFRGLBlAHQEdRU/xatbey1n4bvaQRQND4mtoIvLQLsjbhkGOgOBx7CgCeeWT/hqe2i8xvL/4RIts3Hbn7U3OPWvTa8wn/wCTrbb/ALFE/wDpU1en0AFFFFABRRRQAUUUUAFFFFABRRRQBS1r/kA6h/17Sf8AoJrG+Gv/ACSrwr/2B7T/ANErWzrX/IB1D/r2k/8AQTWN8Nf+SVeFf+wPaf8AolaAOmooooAKKKKACs7X21hNCuT4ajtJNU2gW63jssQORyxUE8DJx3xjI61o0E4GTwKAPKPE+q+N/hzb6ZrureJbfXtPkvYrbUbNtPSDy1kON0LJ83HoxOf0r1evJdRuT8WviLaaXpnz+FfDF4t1f3g+5eXa/chQ/wASrnLHp/46T61QBxdt/wAl21L/ALFu0/8ASm4rtK4u2/5LtqX/AGLdp/6U3FdpQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAeep8LY9R0XxPH4jvhc6p4huvPe9t02m2EfFuIwTkeWAD16kjNV/DPwf0fSJ9fk1XQPD2oi7u/N08S2MbeWgjVdrDy8JlgSQuRzXpVFAHnPgj4d3Ph7xNr+vz6Z4e0u61CGGGyt9MiMkVpsVgxyUjOHJUlQBnb1q54R+Htxp+vy+KPGOpjXPEkqlEmCbYbKPn93Cp6Dnk9Tz6nPdUUAZ+u6Hp3iXRLnSdbtUurK6TZJE/8weoIPII5BritP8IeIfDPgDxLol1rUmvWH2GZNJR4SbqNTGw8tmH3+cBeM/hgD0WigDhfBDtrnwx0/Qiuu6FeWWm2ttPM9k9rLG6xgHy2lTa3KEEgEc+4qDTvhHBpniK61y28YeKPt96YjdyNcwf6QIxhVb9z0wMcY4r0GigDzrxleS6t4+8J2FjpeqyHTdYE9zcf2dMLdE8iQbvO27CMsBwetei0UUAcX4Z/5Kp44+mn/wDolq7SuL8M/wDJVPHH00//ANEtXaUAQXtubywuLZLia1aaJoxPAQJIiRjcpIIDDORkEZ7VwK/B6BfETa6vjTxUNUe2+ytdfabfcYtwbZ/qOmQDXotFAHN+J/Bp8UW628niLWtOg8hoJYrCaJVnVhgl90bZOOOMVD4M8CReCbdLWy17Wb+yih8mG0v5YnjhGc5ULGpz25J611VFAHnmo/CGDVNZsdVvfGXimS809pGs5PtNvmAuAG2/ue4AHPpXc6bZNp2mwWj3lzetCu03F0ytLJ7sVAGfoBVqigAri/EH/JYPBn/Xlqf8reu0ri/EH/JYPBn/AF5an/K3oA668t/tdjPbedJB50bR+bEQHTIxuXIIyOoyDWb4T8MWHg3wvZ6DpHmm0s1Ko0zAuxLFiWIABJJPYVsUUAYGneDrDSvGur+JrOa5W61iKKO6gLL5LGMYVwNuQ2Mjrjk8Vj2fgnVtP13xlqema1DZXPiCW3ktpvs3nfZfLTadyEgNn6967eigDyvUfhf411XxBpGt3nxHha+0czGzceH4wE81Qr5Hm4OQB16V3XhnTde02ymj8S6/Hrk7Sbo5ksVtdi4Hy7VZs85OfetqigDA8deH5/FXgTV9Ds5Y4Z763MSSS52qT3OOa1NJtH0/RbKzlZWe3t44mZehKqASPyq3RQBn62usPpjDw5NYw324bWv4nkixnnIRlOfxrE8A+DJPB/gVNAvLxbyQvNJLNFHsXMjsxCgk9N2Oa6uigDyK1+GXja0+HbeBIvEGif2I0b25uPsMv2ryncs3PmbM8kdOn512XiTwFH4k09dPbxBrWm2P2M2clpp8sSRzRkFTuDRsckHHBHFdXRQBzPg7wTH4MtVtLXXdY1C0jhWGC3v5Y3SBV6bdkan25JrGm+DXhqaSaHz9UTSZ7n7VNoqXhFnJJnOSmM4zztDAe3Arv6KAMPxR4Q0nxdpMVhqsUiLbyrPbTW0hiltpF+66MPukVW8NeBdM8M6neapHcX2papeqsc9/qM/mzMi9EBAAVfYAfoK6WigDitZ+Fmiaxq1/fx3uq6Y2qKF1GLT7sxR3gAx864PbjK4J59Tm7r3w90TXvASeD2Wex0qMRLGtmwVkEbBlALBu45yCTzXUUUAYXizwlYeMdPtLPU5biKO0vYr1DbsqkvGSQDkHjnnv70eJfCNh4quNGm1Ca5jbR9Qj1C3EDKA0idA2VOV55Awfet2igDndQ8Faff8AjnTvFn2i8ttTsYGtv9HlCpcQkk+XIpByASSMY5+groqKKACiiigAooooAKKKKACiiigAooooApa1/wAgHUP+vaT/ANBNY3w1/wCSVeFf+wPaf+iVrZ1r/kA6h/17Sf8AoJrG+Gv/ACSrwr/2B7T/ANErQB01FFFABRRRQAVkeKPD0Xirw3d6Lc317Yw3ahJJrGQJLtzyoJBGCODxyCa16KAPNdL+C1tomnR2Gj+N/GFlaRZ2QQX8SIuTk8CL1NejQReRbxxGR5fLQLvkOWbAxkn1qSigDi7b/ku2pf8AYt2n/pTcV2lcXbf8l21L/sW7T/0puK7SgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5m9+I/gzTb6ay1DxRpdtdQOY5YZbpVZGHUEE8Gr03ivRoPEUGhy30Yvp4HuETPy7FIBy3QHJHHWvnfxdrEtt4W+KGmrr2h20c+vzM2mXIP22fmH5oj5o+Xj+433W59HeMLKWx8b/8JFeeFLO10sK8LrcaBZTEyyyLsLxi7HmMSf8AWEqRn7vJIAPonw74m0rxVpMeo6JdCe3kZ1XIKt8jlGyp5HKmptQ13SdJkRNV1SysnkGUW5uEjLD1G4jNeTfAnQ9R8PK9pq3h6K0nMcm68isLZWwZSwV50nZ3BGMKUAG3GTgE+neIdB8N6lbtf+JtG0y/SziZ/NvrSOUxIBlsFgcDjNAGhYanYarbtPpd7bXsKtsaS2lWRQ2AcZUnnBHHvU1xcRWlrLcXMixQwoZJJHOAigZJJ9AK8X/Zy1y31K38WQW+nJpSyakNRisUGFiinQbQowPlwnGBgDFepeM/+RD1/wD7Btz/AOimoAk0PxToPiYTnw9q9nqQt9vnfZZhJ5e7OM46ZwfyNa1cRoOhnxF8FvDmmf2je6ck2k2XmTWMgjkKCJCyBsHAYZGRzzXHeM/DGmfDvxJ4Lv8AwNHNYX97rcVjcWsc7st7A4PmGQMTuIwvzH1yegwAenHxj4cXxD/YLa3YjVt+z7EZ183djONvXOOa2q4P4if8jR4C/wCw9/7by13lAHl8HhqbXfi14zeLxFrOkiMWOV06aNA+YT97cjdMfqa3P+FeXX/Q+eLf/AuD/wCM0eGf+SqeOPpp/wD6Jau0oA4v/hXl1/0Pni3/AMC4P/jNH/CvLr/ofPFv/gXB/wDGa6bW72807Q7u70zTpNTu4Yy0NnHIqNM3Ybm4H+evSvPdd8beOfBFnaa34w03Q5NFluI4rtNPkl86xVzgMWb5ZME84A/rQBuf8K8uv+h88W/+BcH/AMZo/wCFeXX/AEPni3/wLg/+M1c8Q6h4xbXItN8J6VYJb+T5s2q6nIxiUkkCNY0IZm4yclRzWf4N8barqmua/wCHPEumwW+uaJsdhYyFobqN1yjJvwR2GCeMj3wASf8ACvLr/ofPFv8A4Fwf/GaP+FeXX/Q+eLf/AALg/wDjNYureKviXpXhm68UXPh/RIbC1jNxLpLzyNdrCvLEyj5AwUE7dp+ueK9A0PV7fX/D9hq9lu+z31ulxGG6hXUMAffmgDmf+FeXX/Q+eLf/AALg/wDjNcrrXgi4i+J/hW1PjLxLI09rfsLh7mEyRbRDwh8rADZ5yD0HSvXq4vxB/wAlg8Gf9eWp/wAregA/4V5df9D54t/8C4P/AIzR/wAK8uv+h88W/wDgXB/8ZrtK8r0jxf4+8VeK/FWnaC/hu1t9Cvzax/bbSd3kHOCSsoHbnigDoP8AhXl1/wBD54t/8C4P/jNH/CvLr/ofPFv/AIFwf/GaZ4N8cajrGqeItC8Q6ZHba34eaPz0sZDJHcpIhdGj3YIJA6H1HPJA4rWPiDqmhfE+HxTq/hrxRZ+HZNKXTFtJEiy9405ZWEYl2nK8Zzu9sUAdx/wry6/6Hzxb/wCBcH/xmj/hXl1/0Pni3/wLg/8AjNUvi9qd9pugaBLpt5c2bzeILKKRoJWjLxsx3IcHkHuOhrq/FGunw14avNYGn3Woi0TzGt7RQZGXIyQCewyT7CgDB/4V5df9D54t/wDAuD/4zR/wry6/6Hzxb/4Fwf8Axmuc+F/irVfENtdWJuZNSWS1a+vtVguQ4tLmcl0tYNwKny0IyOingjtWP4o1LxTo/hTU9RtZviJBNa2zypLejSjChAyC4T5tvrjmgDu/+FeXX/Q+eLf/AALg/wDjNH/CvLr/AKHzxb/4Fwf/ABmrVlc+JtT+Hmg3mhXenLqlxaW81xNqUDyI4aLLELGykMWIPpjPFeWXni3x74M8a+KLvxNrOm2trI9nHa/arC8ltZm8olvskYkzx/y0wT83JxQB6T/wry6/6Hzxb/4Fwf8Axmj/AIV5df8AQ+eLf/AuD/4zXE+FL/4la5431nUNM1jSLrTbi0tntrmS0um0+QFTnyF80bXHR+Sc9QK6bxz4s8WeGNN8L2dgNFn13WdSSxlaWGUW67gcuqh9wA46k96AND/hXl1/0Pni3/wLg/8AjNH/AAry6/6Hzxb/AOBcH/xmrfhxfH66of8AhLZvDb2HlnA0yKdZd+Rjl2IxjPv0rmU8feMPETa7qHgjR9Mm0jRrh7ZPtzyCbUXj/wBZ5W3hB2BOc+3OADb/AOFeXX/Q+eLf/AuD/wCM0f8ACvLr/ofPFv8A4Fwf/GaqX3xX0+L4T2fjTT7SS7bUNkNpYhsNJcsxTyifZg2T6KcVFB4z8U6B4q0TSvHlhpaW2ukw213pjyYguAMiGQPnOegYYye1AGh/wry6/wCh88W/+BcH/wAZo/4V5df9D54t/wDAuD/4zXSXHiDRrSS7S71axgeyVXulluUU26t90uCflB7E4zTD4l0IaMNXOtad/ZhOBe/ak8k/8Dzt/WgDnv8AhXl1/wBD54t/8C4P/jNH/CvLr/ofPFv/AIFwf/Ga37/UoLvwleajpN5FPE1pLJDc20gdThTgqw4PI7Vg/CLUr3WPhNoN/ql1Ld3c8DNLNK25nO9hkn6CgBP+FeXX/Q+eLf8AwLg/+M0f8K8uv+h88W/+BcH/AMZrtKKAOL/4V5df9D54t/8AAuD/AOM0f8K8uv8AofPFv/gXB/8AGa7SigDi/wDhXl1/0Pni3/wLg/8AjNH/AAry6/6Hzxb/AOBcH/xmu0ooA4v/AIV5df8AQ+eLf/AuD/4zR/wry6/6Hzxb/wCBcH/xmu0ooA4v/hXl1/0Pni3/AMC4P/jNH/CvLr/ofPFv/gXB/wDGa7SigDi/+FeXX/Q+eLf/AALg/wDjNH/CvLr/AKHzxb/4Fwf/ABmu0ooA4DVfh/cx6NeufHXit9tvIdr3cOG+U8H9z0rL8BeBbi8+HPhy5Xxp4nthNpdtIIILqERx5iU7VBiJAHQZJr0bWv8AkA6h/wBe0n/oJrG+Gv8AySrwr/2B7T/0StAFH/hXl1/0Pni3/wAC4P8A4zR/wry6/wCh88W/+BcH/wAZrtKKAOL/AOFeXX/Q+eLf/AuD/wCM0f8ACvLr/ofPFv8A4Fwf/Ga7SigDi/8AhXl1/wBD54t/8C4P/jNH/CvLr/ofPFv/AIFwf/Ga7Ss3xJqb6L4V1bVY9m+xsprlfMBK5RCwzjBxxQBzv/CvLr/ofPFv/gXB/wDGaP8AhXl1/wBD54t/8C4P/jNYfhbWPiv4l8O6ZraP4Ogtr+FJxG1vdb1Rue0hGcV6hQB5v4Y0WTQ/jTqlvLrGpaqW8P2z+bqMiO4zcTDaCqrwMZ/4EfbHpFcXbf8AJdtS/wCxbtP/AEpuK7SgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA+cfFttfQ6X8SdER9EazuNZSe51CaS43WX2gxlFO2Bl4CjcQ3y559+c1+zk8T6j4tuNL0W4uppNbhZUfwrPcXCIBGWzLwYxgEmMjLD/AHq+ktO8E6FpuialpSWhntdVmmmvhcOZGuGlPzbmPJ44HcACl0Pwhp/h1tXbS5biP+1Z/PkG5cQt5aoBHheAAoxnPNAHmPwbkhsvij4xsY7Mwebb2TqINCk02JNqyZzE2fLJ3cZPzYJHQ1tfFHXZvEl3F8NfC0hk1LVMDVJ4jlbC0z85c9mYcBT1B9xXWaD4HtNDvNXvX1PU9RvtYSOO6u7uZFk2xqyoFMSoFwGPIGfepvCXgjQ/BVlLBodsyyXD77i5mcyTXDf3nc8nr9KAOE8Y6NJ8NNY0bxv4ZspJtM0yyXTNYs4Blms1xtlA7lCMn2xyBk12uu6vYa78LtY1LR7uK8srjSrhopomyrDy2/XPBHUHiulZQylWAIIwQR1rjoPhtpGiaJ4ltPC0Jsm1yCRTA0zfZ45GjKgqnIQHIzgdAOwAoAb4d8QaZ4X+C3h/V9du0tLK30W0LyP3/cpgAdST0AHJri/B+taZ4y8Zr8QPFusabYQ2yPDoGlTXsQe3ibhp5Bu4kcduw+gNdt4X8L3r+DbLw7490fQr62062t4IBG7XSSmNNhdkliUKeBjGep/G9/wrjwR/0Jvh/wD8FcH/AMTQBj/ET/kaPAX/AGHv/beWu8rifEOgeJde8aaFcGLSodH0fUReLL9qkNxKPKZMeX5W0HL/AN/oK7agDi/DP/JVPHH00/8A9EtXaVxfhn/kqnjj6af/AOiWrtKACvJ/iDc/8LF8YWPw70j99Z2txHe+IbgcpDEpykGf77nt2wP9rHoviPT9R1Xw7eWOi6qdIvZ02R3wh80w8jJC5HOMgHPBOe1edeGPhR4w8H6bJZaF8QreFJZWmmkfw+kkkzsclndpSWP1NAHpOt63p/hzRLrVtZuVtrK1QvLI3YegHck8Adya4X4U6bf6lqGu+P8AW7ZrW58SSRmztZPvQWca4iz6Fhgn6A96i8cfCzXvG2paVPc+M0httOSNhZSaUJYZp1HzSuvmANk9FIIA47nPQaL4f8a2NxM2r+N7fU4mt3SKJdFSDy5D91yRIcgf3eM+ooAw/i3r095ZJ4A8N4n8QeIYzCyjlbW1PEs0noNu4D1ycdMHvNE0m30HQLDSLLP2exto7ePPUqihQT78V5joPwh8X+HdT1DUrD4ixPqGpPvury50FJZZPRdzSnCj+6MD8hXpmh2mpWOjQW+uamuq3ybvNvEthAJMsSP3YJAwCB15xnvQBoVxfiD/AJLB4M/68tT/AJW9dpXF+IP+SweDP+vLU/5W9AHaV4X4E0zxJf8AxF+I58N+IrbR0XWSJVm00XRc/PggmRcY/GvdK82svht4k0TxL4g1Xw54ytrJNcvDdSwTaMJ/LPOAG84Z6+lAGL4IvJ/AXjrxPo/jDdqes3Vj/bkmsWyMz3kCEoE8kD5SuCAq5HX2rmNU8U+MfEnxg0u8uPC5ksNK05tZs9BmkIuJIfM8rzto48/qyoegGBycn1vwr4CGg+IL7xDq+r3Oua7fRiB7yeNY1jiByI40XhRnBPqR9ar3PhjVz8a28VWq25sh4bOnIZJOftHnmQAgc7cY5FAHnfxR+GvhKw8PaDqNhpF1aTahrdnFOk99OzBJSd6FTIQp7ZHI7Gu/vPC2k/DjwjrOoeEZYtDLRCS6ur43F8iIm7kRtJnIDHGOvcHpXPeLfC/xW8X2Nja3g8HQLZX8N8jRXF1lnjJIBzH0Oea2tQ8N+OPGXhvUNC8Yz6DYWl15X7zSWnld1WRWdCJAuAyjGRnGelAHkfgrSNRk1iLR9O1Gycw3F7pptEF1Yf2jHGkc5naZHJEgkeP5QFO3cM7eBLrmoaJYeI/DcXinS/ENnpmqWF1Fqejy6pezN9pRiqgDzSWDMNq9iHBI9PRLT4YancNrcpu/7Fvj4ln1HTb6HEjxwSRJG20ZwGZQw+bOODjpWaPhDHe+LjYSaPe22hxadcQy6pcXsck93dPJG63CkMW3goDlgPugYxQB0Ph+28V6H8MLNPDfh6ytdRnkZxZX2ozypZQkHZkuWZiFVFKKQMk4xXmuvavrHhjwzqni/UNH8SXPi6MIkesarZ24tLFWlQNHHEsrBAVJXOCSW5Iya9hgsfHth4CnsRqelah4gjbyra+lieNGjyAHkXnMmMnA+XOOvJPK+JfhRdw/BLXPDuiSNquv6tLFcXd5cuEa7mE8buxJOAAFOBn9aAMiO21mx8Ri6+GugeIvDM99KGurDULSB9LlZuPMZVmLxe7RjovSu71nwVceOdO0C48UXE2kalpk5uWTSbgEByCvyyMuR2ORyPyNJ4y8DXWqXFt4h8J3i6T4qsYwkVyeYrqPvBMo+8h7HqDyKva74W1fVryx1XTPEt1omqW0BhdYUE9rLnBbdC/XkcNkHFAHJ2UmqeBvjRo/htNc1LWNF8Q2k7rDqdwbiS0liUvuEjfNtIGMHufpUv7PZEXwgt452AuILy6S5J6hxKScn1wRW94f8AyWHiyTxR4k1qXXdbNv9mhlaBYIbaLOSscYJwSepJJ/XObe/C27S81oeGvFV1omm69K01/ZR2ySfOww7ROSDGWHXg/ywAeSaONvwj+Hl0/Gm/8ACcxyAEfKsfnSAHHpw3516d8bf3n/AAgsEH/H0/iyyaLHUAb8nPbkrXT6h8O9Dv8A4dJ4M8uWDTYoUjheNsSRMhysgb+9uGc98n1qhpnw8vT4m03W/FniafX5tIjZLCJrVIEjZhhpGCk73xxnge1AHL2XhzSfEH7SXittbso75LOxs5IYZxujDlMbih4YgZxkHGTiqngDwV4dm+Lnj6yuNKgmsdLu7eSyspV328DzRMZGWI/KCdoGccAYHFei6Z4O/s74j634r+3eZ/attBB9l8nHleWMZ37uc+mBijw74N/sDxp4o1/7f9o/4SCS3fyPJ2+R5SFcbtx3Z3Z6DHvQBwfw4t4tM0n4naRZL5VhY6rdrbQD7sSmL7q+g4HFdP8ABH/kivhv/r2b/wBGNWh4d8Cx6Hc+KXmvjdR+Ir2S6dBF5ZhDrtKZyd314+lT+APCc/gjwlDoM2q/2nDayP8AZpDbiJo4ichDgncQSfm4znpQB0tFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUta/5AOof9e0n/AKCaxvhr/wAkq8K/9ge0/wDRK1s61/yAdQ/69pP/AEE1jfDX/klXhX/sD2n/AKJWgDpqKKKACiiigAqhr2kxa/4d1HR7iSSKHULWS1kePG5VdSpIyCM4NX6y/EeiDxFoM+mm/vdPMpVlubGXy5Y2VgwIP1AyO4oA8s8aeFNQ+F3w9TX/AAp4r1oy6CkKm0vrvzLW4iDKhQxY2rweNoB/HmvYLK5F7p9vdBCgniWQI3VcjOD+dcHffDTV/EaW1l418Yzaxo8EqytYxWEdsbkqcqJnUncM9gF/PmvQwAqgKMAcADtQBxdt/wAl21L/ALFu0/8ASm4rtK4u2/5LtqX/AGLdp/6U3FdpQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFNMiCQRl13kZC55I9cUAOopqSJKgeN1dT0ZTkGnUAFFFFABRRRQAUUUUAcX4Z/5Kp44+mn/APolq7SvL4I/Fj/Frxn/AMIvdaNDHix80ajbSyE/uTjbsdffr7VueR8T/wDoI+Ev/AC5/wDj1AHaUVxfkfE//oI+Ev8AwAuf/j1HkfE//oI+Ev8AwAuf/j1AHaUVxfkfE/8A6CPhL/wAuf8A49R5HxP/AOgj4S/8ALn/AOPUAdpRXF+R8T/+gj4S/wDAC5/+PUeR8T/+gj4S/wDAC5/+PUAdpXF+IP8AksHgz/ry1P8Alb0eR8T/APoI+Ev/AAAuf/j1crrUPxBHxP8AConvvDRvTa3/ANnZLK4EariHfvBlyT93GCMc5zQB69RXF+R8T/8AoI+Ev/AC5/8Aj1HkfE//AKCPhL/wAuf/AI9QB2lFcX5HxP8A+gj4S/8AAC5/+PUeR8T/APoI+Ev/AAAuf/j1AHaUVxfkfE//AKCPhL/wAuf/AI9R5HxP/wCgj4S/8ALn/wCPUAdpRXF+R8T/APoI+Ev/AAAuf/j1HkfE/wD6CPhL/wAALn/49QB2lFcX5HxP/wCgj4S/8ALn/wCPUeR8T/8AoI+Ev/AC5/8Aj1AHaUVxfkfE/wD6CPhL/wAALn/49R5HxP8A+gj4S/8AAC5/+PUAdpRXF+R8T/8AoI+Ev/AC5/8Aj1HkfE//AKCPhL/wAuf/AI9QB2lFcX5HxP8A+gj4S/8AAC5/+PUeR8T/APoI+Ev/AAAuf/j1AHaUVxfkfE//AKCPhL/wAuf/AI9R5HxP/wCgj4S/8ALn/wCPUAdpRXF+R8T/APoI+Ev/AAAuf/j1HkfE/wD6CPhL/wAALn/49QB2lFcX5HxP/wCgj4S/8ALn/wCPUeR8T/8AoI+Ev/AC5/8Aj1AHaUVxfkfE/wD6CPhL/wAALn/49R5HxP8A+gj4S/8AAC5/+PUAdpRXF+R8T/8AoI+Ev/AC5/8Aj1HkfE//AKCPhL/wAuf/AI9QB2lFcX5HxP8A+gj4S/8AAC5/+PUeR8T/APoI+Ev/AAAuf/j1AHaUVxfkfE//AKCPhL/wAuf/AI9R5HxP/wCgj4S/8ALn/wCPUAdpRXF+R8T/APoI+Ev/AAAuf/j1HkfE/wD6CPhL/wAALn/49QB0+tf8gHUP+vaT/wBBNY3w1/5JV4V/7A9p/wCiVrG1WD4l/wBjXvnah4UMf2eTeEsLkHG05x++61l+AofiKfhz4cOnX/hhLQ6XbeQs9lcNIE8pdoYiUAnGMkACgD1OiuL8j4n/APQR8Jf+AFz/APHqPI+J/wD0EfCX/gBc/wDx6gDtKK4vyPif/wBBHwl/4AXP/wAeo8j4n/8AQR8Jf+AFz/8AHqAO0ori/I+J/wD0EfCX/gBc/wDx6jyPif8A9BHwl/4AXP8A8eoA7SiuL8j4n/8AQR8Jf+AFz/8AHqPI+J//AEEfCX/gBc//AB6gAtv+S7al/wBi3af+lNxXaV5v4YTX0+NOqf8ACTz6bNc/8I/bbTp0EkabPtE2Mh3Y5zu/MenPpFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB8z+Lb67GifEW/Fv4klvLHW5orTU7XU3S2s0Bi+Qxidf7x6Rt94c8cGv+K4V+MEF1b+JtSliis7iNbmHWdJG0GRTtR3+RU4+7J+84Hoab4u0svpXxFsZLHxQdWvNbkksoLSG8+yzoxi2syoPKYnDYLcnC+1Q+P8AUtU1I+Jo/N1CWysdbtoLf7LJZJBGN0RAIYeaWyeCPlzjPGaAOv8A2dNZin0cac2rXMkiLO6WUt9ZvGMzsSyxR/vgeeS/yncSOCK9O8UeEn8Rzwzr4m17RRAhUppV2kKvznLbkbJFcZ8K9T1Gfx14p0vVZbspYxWbW0WoNaPPHvWQud9uAhBwvTpxmtn4ueKZdE8JnSNHPma/rzfYNOgQ/OWf5Wk9goOc9AcUAZHwK8Str+meJIf7cvdbgs9YdbO6vnLSm2KL5ecgf3W7DnPArvPF0skHgnXJYXaOSPTrhkdDgqRGxBB7GvKtO0mD4HeNdGkkcJ4d12xh07ULkjCw3sQ+SVz0AcFh6DkmvU/Gf/Ih6/8A9g25/wDRTUAcv4S0fU7T4Uade+Epw2uapp1nNLPrF5cXMW9kUu+1nO0/MxwuATgHjGMzXtX8Z/D3WvDc2q+IofEdhrGqRabcWrafHbyRtJnDxFDkgY6Nn0zzx2Xw/dU+GPhbewXOkWgGTjJ8leK818f6DcfDrUrL4gS6zd+JI7K6SJ7LWm80wpIwUtb7dqq47fKeO/HIB03jTTYtL8f+EdSsLi+huNR1kQ3Si/mMUieRIdvlF9gGVB4UdK9Hrg/iJ/yNHgL/ALD3/tvLXeUAcX4Z/wCSqeOPpp//AKJau0ri/DP/ACVTxx9NP/8ARLV2lAFLWNXsNA0e51TV7lbWytU8yaV84UfhyT2wOtchF8XNGW8sY9T0rXNItdRlWGzvtQsfKgmZvujOSVz23AevSu6dFkQpIqurDBVhkEV5b8b5V13T9I8B6f8AvNW12+iYBPvW8EbbnmPoBj8fm9KAOt8Q+PNN0DWIdHS01DVtWmi84WGmW/myLHnG9iSFUZGMsRUvhPxvpPjFb1dNFzb3eny+VeWV5CYp7du25T64OCMjg1vsI0LTMFUhcM544HPJ9OteV/Du8h1rx7468fo6waFMYrS2uG+VJ0t0xJNnuvHB9MjtQB1nj/4i6H8ONIgv9f8AtEn2iXyobe1RWlkIGSQGYDAHU57j1rqIZVngjlQELIoYA9cEZr5t+IMM/jX4Z+LPiPqcbpbOsFn4ft5AQYrYXUQabHZpD+OOOQRX0bp3/ILtf+uKf+gigCxXF+IP+SweDP8Ary1P+VvXaVxfiD/ksHgz/ry1P+VvQB2lcD/wtm1n1vV9M0rwt4k1WXR7g291JZW8LIHGemZQTnBxxXfV4X4E1rXNK+IvxHXRPC1zrqyayS7Q3kEAjI34B8xgTn2zQB6l4Y8caL4rtb6Wwkmt5dNkMd9a3sRgltWwT86t04B56cHng1yp+L2jf8LaXRV8R6CfDx0X7Ubz7ZHj7V523y/M37c7OdvXvWd8Lb1brx541v8AxPB/ZHiHUPJludInXCwWsabEcOflkBzyw4/OuQ8YeOfBfib4haXoMdnHH4WsJReajqdtZApOVbYoLAcQCQhWfkE4HQZoA9e+Iniy98JaTpN1pkVvK97q9tYyCdWYCORiGIwRzxweR7V1N3LJb2U80Fu91LHGzJBGyq0rAZCgsQASeMkgc8mvCvipovi9dB0O5uvHVtqVnca5Zi0WHR44xGzMfLkDhzvA646GvT9M0fxtp+m363vi2w1a9l8v7JJcaP5McGCd+5Y5QX3A8cjBA68igCjrHjfxNp+h394ngDU4mt7aSUSTXlmyKVUnLBZ9xAxyBzjpVXwr8QPE+ueE9M1OXwJf3L3dskrTWl3aJE5IzlFkn3Aezc15P4/j0vRfEsHhu7i+HzajqbMt7dDSJIfsCMDmVpDOQH5yF6k49Rm/ofh7wjDa2Ok2/iD4Y6jOoSCOW40pnmnboCxFwMsT7cmgD2698R3un+H7bUZvDGrzXE0mx9OtPJlmh+98zHzAmPlHRj94e+OE0z4x6pceM/Eem3ng7WDaaaLbyYoLeIXMPmR7m8/dPt5PK7e3XmurutfufA3hWOXxPGNQm83yLaDQdNlwwC5VNhd8HCtyWAxXjPibxD4f16x1Hxf4vm0B7YGNo/C2lanCbq/ORGklzLGcuyKxIUZCgduaAO/074vavdeOtZ0mXwVrclrZQwPFbwW0X2qMuCSZcz7dp/h2/jXS+IviPZ+F9C0jUdS0TWTLq1ytrBYRQRtcrKwOFZd+MnHYnqK850fxPa/DzWCmh6novjbTLvZFGbO+t21mJAMJDgNm5A4CjO4Zx0AFdb4v0rVvH8Pg/XPCSRQiwv8A7cU1ZZICuFIG5Mbjz24+vOQAdB4c8byeIdUNm/hPxJpAEZk+0anZpFEcEfLkO3Jz0x2NUNW+LOg6XqWo2sdpquox6SdupXdhaGWGzPcO2RyO4UHHOehqto3jTxFp/wARYPB/jm103ztRtnuNNv8ATN6xzbOXjZHJIYAE9cce/Gb8AIUvfhH9ru4labVL27nuww/1jtIVOfwAHNAHdX/i3Q9N8InxPd6hEuj+Qs63QyQ6NjbgdSTkADrk4rI0P4laTrWuW2jzWOqaRe3kJns49UtfJ+1IBklDkgkDkg4OO1eJaXJJc/B3wBosxaSxk8ax2b7ud8Qmc7T2Iyx/KvTPjUfss3gTUYBi7g8U2sUZHUo4bevrg7QKAPUaK8hkt9d8S/HLxRoUfijU9L0i3s7SV4bOUh8lOkbHIiySSxUZOBzVHwjpfiXW/FHirwXqHjbWP7J8P3MZjuYpAt9OsyFlRrjk4Xb2GST1xgAA9i1O6ax0i8u41DPBA8qhuhKqTz+VY3w+8SXHi/wDpWvXsMUE99EXeOLO1SGI4zz2rjPAWqarN4R8caFrWpTaq+g3l1ZQXlwcyyRCPK7z3brz7+1bXwR/5Ir4b/69m/8ARjUAd5RRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFLWv+QDqH/XtJ/6Caxvhr/ySrwr/wBge0/9ErWzrX/IB1D/AK9pP/QTWN8Nf+SVeFf+wPaf+iVoA6aiiigAooooAKoa5reneHNFudW1q5W1srZd0srAnHOAABySSQAB1Jq/TXRZEKyKrqeoYZBoA4i2+LGkNqen2mqaVreipqcgisrrU7LyoZ3PRQdxKk9gwFdzXlnxjmXxBe+HPAunfvNT1HUYruTZ1tbaIktKfT0Hrg16nQBxdt/yXbUv+xbtP/Sm4rtK4u2/5LtqX/Yt2n/pTcV2lABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB56nwtj1HRfE8fiO+FzqniG689723TabYR8W4jBOR5YAPXqSM1X8M/B/R9In1+TVdA8PaiLu783TxLYxt5aCNV2sPLwmWBJC5HNelUUAec+CPh3c+HvE2v6/Ppnh7S7rUIYYbK30yIyRWmxWDHJSM4clSVAGdvWrnhH4e3Gn6/L4o8Y6mNc8SSqUSYJthso+f3cKnoOeT1PPqc91RQBn67oeneJdEudJ1u1S6srpNkkT/AMweoIPII5BritP8IeIfDPgDxLol1rUmvWH2GZNJR4SbqNTGw8tmH3+cBeM/hgD0WigDznwxpOl+NvhfpHh7XtL1a0l0qys1c3NrPZSQzpGAGidguSpU8qSOmetXo/hVpcupWd3rusa54gFjIJba21W9EsMbjo2wKu4j1bJruKKAPOvGV5Lq3j7wnYWOl6rIdN1gT3Nx/Z0wt0TyJBu87bsIywHB616LRRQBxfhn/kqnjj6af/6Jau0ri/DP/JVPHH00/wD9EtXaUAZniO81Ww8O3lz4e0warqaJ/o1oZliEjEgcsxAAGcnnkDA5ryLwZa/EHQNV1DX9e+HdxrHiPUmxPfHWLRFiiB+WKJNx2IAB35I56V7hRQB5J8Uk+IviWGz0TQ/DEy6Jcwo+rPBqNvHPJnlrdWZvlA6FsHOfTOd/wwup3+m/8Ivrfw6/sLQFszAA+pQToy8DyyqHdyCcn65613lFAHkPxK+Cmg3Xw81OHwP4Usl11vK+ymIrG3+tQvhnYAfJu6mvSfDWgaf4a0GDTtJsorGBRuaGIcByPmP51q0UAFcX4g/5LB4M/wCvLU/5W9dpXF+IP+SweDP+vLU/5W9AHaV5D4bsvGXg3xt4zvIvBdzqtprOpG5tpodQtY8oN2Mh5ARnPpXr1FAHmuieDNY8Q+PNU8V+PNOs7OK50v8Asi30mOYT/uC25zK4ABJJOMdj7VUk0W2sPj+I4tIzo8PgprcQRW26Jh9qJMQXGCSP4e+a9VooA8O+Ivi6bxLomj2ejeB/GCGx1e1u2V9CkRRFETkLjvjoK7aXxvp3iHw5rCalp3irw3aQ226a9uNOltpEBOMxMASWHXgHj2ru6KAPApNNtfPsLr4KeGdbttZtXIOr3Ns0FtdxswMiXDTkNKDjPAyCBgirlpda3pvjX+3vi/4f1a8urRj/AGdNpNr9p06xTAy4VCXDn+8wzXuNFAHLTePNOn8E3PiLRLLUNYhjYxLawWciSySZA27HUEDJ5bGBz1xXl/jzw3rUX7PfiO/8T2lmNXvJoJorOztIwLCM3EWIlZV3MQOpJPf8feaKAPOPFum6x4Z1C08UeGdLtdW0+2jUX2ipZxLNtHPnQOF3bx/dJOQOMGt7XdR8X215Y3/hzSLXVNNeA/adPmm+zXSucEFWbKcDgqcc966migDznS/D3iPxJ8T7Lxh4q06LRrXR7WSDTdPFws8xeQYeSRl+UfKcbQT/AI5unaF448Bwa9ofhTR7bU9Pv7qW50u8e9WL7CZeqyIeWCnkbck/jx6xRQB5bd/COSH4LaZ4V0m6iXVtHljv7W5kz5Zu1YuSe+0lnA9AR6VJLonizx14u8OXfirRIND0vQZvtzxC8Wd7q6AwhXb91FPOTyemK9OooA4XQ/Dmq2fxq8Ua9c2uzTL+ytYrafzFPmMi4YbQdwx7gUng3w5quk/E7x5q2oWvlWOrTWbWUvmI3miONlfgElcEjqB7V3dFAHnfg/wlq9jdeP11G3Fsmt6lNLZSGRWDxsm0NhSSOexwfatD4SaTrHh/4b2GieItO+w3enM8AAnSVZk3FhICp4B3dDzxXaUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUta/wCQDqH/AF7Sf+gmsb4a/wDJKvCv/YHtP/RK1s61/wAgHUP+vaT/ANBNY3w1/wCSVeFf+wPaf+iVoA6aiiigAooooAKyPFN7rVh4bu7jwvpa6rqqri2tXmWJWYnGSzEDA64yM4xWvRQB4h4JtfH3hm81DWdX+HV1q/iLVHzd6g2s2iAID8sUa7jsQccZ5wPQAe1wPJJbxvNF5MjIC8ZYNsOORkcHHrUlFAHF23/JdtS/7Fu0/wDSm4rtK4u2/wCS7al/2Ldp/wClNxXaUAFFFFABRRRQAUUUUAFFZ3iDXtP8MaDdazrErQ2VooaV1QuVBIHQcnkiufj+KnhYX0lrqNzd6RJHb/amOrWMtmvl71TcDKqg/MwFAHY0V5knx98Ftpdrdm9RWuNQFk1uZo/MiXeV89hu4j43Z9CK9D03U7HWdOiv9Ku4by0mBMc8LhkcAkHBHXkEUAWqK4W4+NXw8tLqW3uPE9uksLlHUxS/KwOCPu+td0DkZHIoAKKyfE/ibTPCOgXGr61OIYIV+Vf4pX7Ig7sTwBWZc/EHSNL0nTbzX4NR0yTUIfOS1awmnkj6ZD+SrhTyOCR/OgDqaK462+KvhK7uoreC7vzJM4jQNo94oyTgZJiAH1PFT+KviHo/hKS9g1Bbh7m105tREKJjzolcKwRjgFgSMjOcHNAHVUVyNz8SdEt9L8QXqi5lTw/Ast5sj4LMm8RqxO0vggEZ4JGa6TTL9NU0m01CKKSKO6gSZY5gA6hlBAYAkZ59TQBaooooAKKKKACiiigAooooAKK+d/FPjzWbPT/HN2nirXbXUdK1aW3061t7BXtBEDHgPJ5DAH5m6yA/d9edHXfFHim1+MVtbx6zehEsrgKkXhi6lTHmLj90sn77H/PVcAenNAHu9FeSfAXW9a1bwyo1bUbq8hTzyouNLmjO7z25Fy7FZOMjYBlehPy13HibR/FGp3ED+GfFUWhxIhEsb6Wl15hzwcs64+lAHR0V558IfFepeJtP8QRaxq9vrMumavJaw3ttCsaTQhV2MFXjk7j39MnFdf4ouprHwhrF3aSGOe3sZ5YnAztZY2IPPuKANSiuU+HMN+/g3TdV1TXNQ1WfVLC2uXW7EQWFmjDME2IpwS38RboPfPKXXxdu5vjZpvhDRbW1m0iSeS0ur6RWLNcRxl3SMhgPkygOQeWoA9WorgPFkmraJ468MXFnr+oNa6tqotp9PkEJgVPJdsLiMOOUB5Y9TXf0AeXwa1rWmfFrxmuj+F7rWlcWO94buCER4gOP9Y4Jzk9PStz/AIS7xf8A9E21H/waWX/x2jwz/wAlU8cfTT//AES1dpQBxf8Awl3i/wD6JtqP/g0sv/jtH/CXeL/+ibaj/wCDSy/+O1113d21hZy3d9PFbW8Kl5JpXCoijqSTwBXLWHxW8E6nqkOn2evwNPcPsg8yOSNJmzjCSMoVznjgnt60ARf8Jd4v/wCibaj/AODSy/8AjtH/AAl3i/8A6JtqP/g0sv8A47Wt4h8beHPCssMOu6rFbXE4zFbqrSzSD1WNAWI+gqbw94t0LxXYSXvh/U4L2CFtspQlTEfRlbBX8QKAMP8A4S7xf/0TbUf/AAaWX/x2j/hLvF//AETbUf8AwaWX/wAdpZvi74DgvDbSeJLbKyeW0yo7QK3oZgvlg/8AAq7JHWRFdGDKwyGByCPWgDjP+Eu8X/8ARNtR/wDBpZf/AB2uV1rxL4nk+J/hWeTwHfRXEVrfiK2Oo2paYEQ7iGEm0bcDqRndxnmvXq4vxB/yWDwZ/wBeWp/yt6AD/hLvF/8A0TbUf/BpZf8Ax2j/AIS7xf8A9E21H/waWX/x2u0rA8S+OfDXhAxL4j1eCykmGYoSGeSQeoRQWP5UAZX/AAl3i/8A6JtqP/g0sv8A47R/wl3i/wD6JtqP/g0sv/jtbPhrxj4e8YWrz+G9VgvliOJEQlZI/wDeRgGX8QKqWXi17v4map4VNoqpY2MV2LnzMly5I27ccYx1zQBR/wCEu8X/APRNtR/8Gll/8do/4S7xf/0TbUf/AAaWX/x2r/i7xf8A8Ite+Hrf7D9q/trVI9P3eds8neCd/wB07sY6cfWuloA4v/hLvF//AETbUf8AwaWX/wAdo/4S7xf/ANE21H/waWX/AMdq94k8bweGdSitbvQ9evIpIRKbzT9Oa4hTkjaxXJDcZxjoRR4e+IPh/wAT6kdO0ua8F6sTTNBc6fPAQgYKTl0C9SOM5/WgCj/wl3i//om2o/8Ag0sv/jtH/CXeL/8Aom2o/wDg0sv/AI7XZSOI42d87VBJwCTgew5NZnhrxLpfi3QYNZ0Kc3FlOXCOUKnKsVOVPI5B60AYH/CXeL/+ibaj/wCDSy/+O0f8Jd4v/wCibaj/AODSy/8AjtXfF/ji18I+XHPZXV3c3EEs1tFCoAnMYDPGGPG/ZuYKfvbSBzxW9p1/b6rpltf2T77e6iWWJiMEqwyMjseelAHKf8Jd4v8A+ibaj/4NLL/47R/wl3i//om2o/8Ag0sv/jtauqeMtM0nxdpHhuZbiXUdWV3iSFAwjRBy8hz8qnkA88g1v0AcX/wl3i//AKJtqP8A4NLL/wCO0f8ACXeL/wDom2o/+DSy/wDjtaGv/EHwt4ZvxY6zq8cN5s8w28UbzSIn95ljViq+5wK1dF1vTPEWlRalod7DfWc2dk0LZBx1HsR6HmgDmv8AhLvF/wD0TbUf/BpZf/HaP+Eu8X/9E21H/wAGll/8dq1468d2XgvTowI2v9YvCY9P0yE5luX+nZB1LdAPfArP8IfEddW1ZvD/AIot4NI18DfDEk4kgvo+cPBJ0fGCCOowfQ4AJv8AhLvF/wD0TbUf/BpZf/HaP+Eu8X/9E21H/wAGll/8drtK4jX/AIo2fhy8vYr7wz4okgsQzS3sGku9vtUZLiTONoHfpwaAH/8ACXeL/wDom2o/+DSy/wDjtH/CXeL/APom2o/+DSy/+O10+jarb67oVjq1jv8As19bx3MXmDDbXUMMjscGrtAHF/8ACXeL/wDom2o/+DSy/wDjtH/CXeL/APom2o/+DSy/+O0yz+LXhiTw/LrGp3LaXarey2UIuBue5aM4JjRNzMM+gzXQ+HPFGi+LdM/tDw7qEV9bBtjMmQUbGdrKQCpwRwQDzQBg/wDCXeL/APom2o/+DSy/+O0f8Jd4v/6JtqP/AINLL/47XaUUAcX/AMJd4v8A+ibaj/4NLL/47R/wl3i//om2o/8Ag0sv/jtdpRQBxf8Awl3i/wD6JtqP/g0sv/jtH/CXeL/+ibaj/wCDSy/+O12lFAHF/wDCXeL/APom2o/+DSy/+O0f8Jd4v/6JtqP/AINLL/47XaUUAcBqvizxa+jXqyfDrUY1a3kDOdTsztG084ElZfgLxR4pt/hz4chtfAF/dwR6XbLHcJqNoolURKAwDSAgEc4IzXo2tf8AIB1D/r2k/wDQTWN8Nf8AklXhX/sD2n/olaAKP/CXeL/+ibaj/wCDSy/+O0f8Jd4v/wCibaj/AODSy/8AjtdpRQBxf/CXeL/+ibaj/wCDSy/+O0f8Jd4v/wCibaj/AODSy/8AjtdpRQBxf/CXeL/+ibaj/wCDSy/+O0f8Jd4v/wCibaj/AODSy/8AjtdpUN5eW2n2ct3f3EdtbQqXkmlcKqKO5J4FAHI/8Jd4v/6JtqP/AINLL/47R/wl3i//AKJtqP8A4NLL/wCO1Y0r4o+Dda1WDTtO1yJ7q5JFuskUkSz4/wCebOoV/wDgJNdZQB5v4Y1LUtT+NOqS6voU+izDw/bKIZriKYlRcTYbMbEckkY6/KfbPpFcXbf8l21L/sW7T/0puK7SgAooooAKKKKACiiigDgPjl/yRPxJ/wBcE/8ARqV5rcazej4nTXWneINQvni8MyCS91aJdIe0T7TGSVZrXaR05MZ6nnivVfi/b2d18IfEUepXv2G2+y7mmEe8gqwYKFyMliAo5HLCvG/iFcazGb+88RtLdX994SMcEMVglqkcLSrJIWLTvuaMhQyqM/PkDAJoA5eC9udtv4e/4S/T1Ww1BNYW4OvQC2ZjKzbUP2XPnDOSSSvOdmDgfUng24vbrwlYT6lIs0skYdZ0u1uRMh5V/MWOMHIIPCD+teF33hvxpEdY1yaz8VCym8PC3MzX2mFyFVmIcD/llg/wjf156V6l4F8R2WhfA/w/qviNhpNpbaXApe4dTvURgKy7Sc7gMhfvc4xnigDn/wBovV7pfA9v4Z0qBLq+1yVswFdzeTCvmuyj1BVK9F8IazF4i8F6Pq8Dh1vLOOUkHoxUbh9Qcg/SuG8Baff+NfGV18RvEFm9ratB9k0GzmHzJb9WmYdi+fyJ7YNUdG1H/hTHiqfw9r37nwdqty02j6if9XZSuctbyHoq5JIJ44J55KgF74u6r4T8HTW3i3X0Ooa1bQmLSNNlmJjMmc+aI+gIyMv6ADrirWl6v8QdB0Swt5vDc/i64mgFzPqCajb2yK7ksYlR8HCjAB7/AKVV+OWgaNN8M/EPiH7Bbyap9gjgS825cR+cpAB7dTyK9I0r/kD2f/XBP/QRQB5xD8U/Flx4kudBh+G1y2p2sCXE0H9s2w2xscKd33Tn0BzXGfHXTZ9SbVZdNtdWtms4c3Wo3t9Mtu28Li2tot+12chd2Bjju3TvNG/5OP8AEv8A2BLX/wBDNcj8ZbRdN8a6Zq2uDVZ/DMTCe5t7fXCkktx92MwW7SDBQgN8mCSx54NAHE+OrDXPDPg2bw1Np+ozXd9H9suLqLU55bW5tI8SSyNFIx8qUMsYPXqcHBAr0Pw3pf2j4zWtjoviTxFcaHYaLDqMkc2sTSo8sj5jUgtgqY+cdD6Vl+DfE/h7T/h7qWveLNf1HVLzUJbq0j0KfVJrmZEdtqWohdifMwANxGfmOTirPwDWPwhfan4Y8Vwzad4pu/JnjW8lDG4tViCxIjdygDAqOnboQAD3SiiigAooooAKKKKACiiigD5b8Yajaw+G/idZSeLfsNxLr0zJomIP9LGYvn+ZDJ2P3WA+T65qePrTT2m8Q29noWm2UenavBbRvaeHJJS6loyS1wrbQTuI2EZYcd69oHgfxHrGk+KrjVdWk07WNWuxLp/2W5dorFIcCDpgHdty/HOeRkVj+Hfg5ZeZ4lHirTLy6ilvxNZIusXAW4VY0w2BNnO8HBkO4cc4AoAZ8H4LH/hOPElsmiadazaXDbCG6h0N9MmYTBy6tG7FgPkXHrjNdj8UvFZ8J+BLua0+fVL3/QtNhX70lxJ8q4HfGd34Vz3w/wDBd/pHjTxL4ik0W901Lq3gi0+11DV2uHYqjb/MZZJBgnbgtuKjOAOQbnhzwRrer+LY/GPxIkt31C1BXTNLtm3Qaep6tn+OQ+v/ANbAByfgTRj8GfHWm6LqDhdN8UWEMfnknYmoxDDJ7Bw3HqSB2r1jxn/yIev/APYNuf8A0U1N8YeEdL8b+G59G1uNmhk+aOVDh4ZB910PZh/9Y8GuTtbXxnpnw58U6X41uLTUUtNPnWy1SJyJbqPym/1iY4YcDOeffG4gGfq3iu+8Kfs9eHJdGgml1K90uztbZ4oWkEBaBcysFBOFAJ6HJwMVwUus+EfDfi74YWuj3N7La6Q9899cS6ZdJJLJJGmZNrR7nLMD90HAxnAxXsXwq8QaRrPw70S20rUra8nsNLtIrqKGQM0D+UBtYDocqw59DVrxF4L/ALf8b+FvEP2/7P8A8I+9w32fyd32jzUVfvbhtxtz0Oc9qAMnx7Mlx4h+H00RJSTXA6kqQSDbykcHkV39eZ+O/EGkXnxD8F6Pa6lbTalaa4GuLRJAZIh9nk5Zeo6j8xXplAHF+Gf+SqeOPpp//olq7SuL8M/8lU8cfTT/AP0S1dpQBDeWdtqNlNZ39vFc206FJYZUDI6nqCDwRXivxb8QWPiZbTwFFp11ou7UY0Gr6nB9ktYFjbGYHb77EDC7Rgg9cV674hvdT07Q57rQtMGq3sWCloZhF5gyNwDHjOM4z3rzXx7rupeP/BN14X0jwRry3+o7I2fU7LyLe0+YHe0hJBxjI2Z/pQB6F4gutO8Pxv4hn0e4vruKMQb7CyM9z5ZOdo2jO3PPoK8++E9xB4l8f+MfFsax2C6gYIP7Jdx9ojEakeZPGPuM2OB/vfU9NN4k1jwjcWumaj4b1TVdNhsoUXVdMT7S7yquH8yIfOvQEEbs5/Ln9As9e1r4rav46ttAuNJs10f7BbWmoYgl1CYPuDuozsHAXJ5xt98AEPxW8S6X4e8G3/gyw8N3MMV5bG1iuDZ+Rp1t5g++ZeFG0tnjJ3CvQvBmnLpHgbRNOS8S+W1sIYRdRtuSbagG5T3U9vbFcfqXj/U77wxdadd/DzxC+r3Fu0D2JtBJbMzLg5nB2FMnr1x2roPhl4cv/CXw10bRNXmEt7aQkSlW3BSzs2wHuFDBfwoA6quL8Qf8lg8Gf9eWp/yt67SuL8Qf8lg8Gf8AXlqf8regDtK8l+C1tF4hu/EvjjUkE+qX2rTW8Mkgybe3jwEjXP3euOPQV61Xkml2+u/CbxNrkUOgahr3hXVrxr+3k0tBLPZyv99GiyCy8DBHQDuSQAD0ufQNJuf7QM2nWzNqUP2e9bywGuI8EbXI5IwxH418+6rafDPV/jBo3hfwxb6VpsdhP9oub6MlVurhf9Xao/TryT3xgcivXNP1C/8AiFZ6tp2seGdS0PQbiARRTXc32e7mYk7v3a8ouMck+xBBwOIg8L+ErD4oeLdI1PTLWPw5Z+HLaSaAodkaIzMX45yME5HOaAKfxIf4kHVfBn9sweFlkGvwfYvsk1wVM2G2iTcownXJGTXr3hlvFTW0/wDwmUejxz7x5P8AZUkrKVxzu8wA5z6V5B8Rfin4H1XUvBTaTr0M8Wm69Bc3JWOT91CoYFjlegyOleteF/HXhvxmboeGNVi1A2mzz/LRl2bs7fvAddp/KgDBMfxV1W8mH2jw74fs1ciJhDJeTsoPDEblUZHOKzLnxD4y8A+LNAs/FuqWPiDRtcuxYLdxWX2We3nb7mVDMpU9P8Mc9Z4s8e6N4Na3i1Vb6W5uwxt7ezspJmm24yAVG3IyOCR1FcbZaZ4j+JfjbSfEHiLSZdA8O6JKbiwsLoj7TdTYG2SRR9wDsOufXNAHq9eD+FL3xHH4w8VaR8JorO78PXV+J11S73fZ7CZh+/SMD/W84IC8DA7GvYPF1hfar4L1nT9Il8m+urGaG3fOMOyELz25PXt1rzbwF8R/D3gvwPpnh7xRp+peHL/ToRBNDNpkzrK6/ekVo1YMGOWz7/jQB2fhXwGNEklvde1e78R6rPIkz3N8AUidQQDDH0iwGbpzyea6m7uoLGzmu7yVYbe3jaWWRzgIqjJJ9gBXK6B8TdB8T61Fp2iRanOZFZhcvp8sUICgnl3A644rD+Ms95qdjongrTfOjk8TXy29xPGpxFbJh5ecYBI7dwGoA4nwB8RtDuPGWo+NPEkOpLNrl2NP0+6NlI1tZW4bbHEZMYDMRk7cjPJxzj6BrzH4x6Atr8D7nTtB09mg042zx21umSsUcqkkAeigk/iaz7/xo/xW1bRdJ+Hsl8umRXcd7qmsCGSBY44znyVLAEux4OPTuM0AeoRabpem3d7qUNrbWs91iS7uQgVpNq4Bdu+AO9eX/COzvNS8LeMtS8N3A0m01nWbqbRpTAHWNfuiURnjBI+7/s4ql8VfEeva7r//AAiMHhzxNF4aVh/amo6dpksr3i4yYYiBjYehbPPPGBhvQfBXiGw1GwbT9N8OazoNnpsKJGmp6c1qmzkAJu64A5oA8btNHl0D4xeJbS58QeJLu7js7fdqNppCahO/mIS6n9zIYk7ADAwMc4rO8ZaNZad4D0nRLG/8VPZWt/DFbR6p4fFssAeXLFbhrdWVsnj5+enPSugn1fwJf/G3xdd+IPEcMFm1paJbXFpq8kAdwhDgGFxvIOODnFcrd2avZWOn6zfy/wBiXF20lt42nv764hkRZPkHk7vLSUEEZcBeMgHqQD6L8JeH9Q8N6ZJY6j4hvNdQSZtpb1F82KPAARnHLnOTuPPPtXNfGrWruw8DJo+jzeXq3iG6j021wMnDn94cemzIJ7bhWxa+KfDHhnwFFqFx4pj1DTLNBE2pTXa3DzNjOCy/fc+g5rl/B+n6n4+8cJ8QPENpJZaZZxND4f0+dcSKrfeuHHZmHAHp9ASAafwT1Q3vwusLC4+S/wBFd9MvIc8xSRMVAP8AwHafxrvLiXyLaWYjd5aFseuBmvLfFVpqfw08bXXjrQrOXUNB1JVGv6dbjMkbKMLcxjvj+Ie5J65X0bQ9d0zxNolvquiXcd7Y3K7o5U6H1BB5BHQg8g9aAPIP2bfCsZ8FxeKtUxdXty8sVm0gB+zwiRtwX0LSbyT9K3LBE0T9qHULPT02W+teH0vruNBhTOkxQOR67QfqW961L7xX4W+GFvb+FdF0nULi6jjMttpGmWcszlXYnIY/Lgtu/i9ai+H3hnXJfE2q+OfGkMdrq2pxLb21gjbvsNspyELd2JwT7j3wAD0SiiigAooooAKKKKACiiigClrX/IB1D/r2k/8AQTWN8Nf+SVeFf+wPaf8Aola2da/5AOof9e0n/oJrG+Gv/JKvCv8A2B7T/wBErQB01FFFABRRRQAVV1PS7HWdPksdWs4b20lxvgnQOjYIIyD6EA/hVqsjxTrkvh3w3d6na6Xe6tPCv7qysYmkklYnAAABOMnk44GTQBwPxuWC9s/C3h7T40OtXmswNYBFG63WM5eUeiqMZ+vtXqteC+C/EV3beI73xh428IeMLvxHeKYYY4NClaHT7cE4iizzz1LY/mSfdoJfPt45gjx+YgbZIu1lyM4I7H2oA4+2/wCS7al/2Ldp/wClNxXaVxdt/wAl21L/ALFu0/8ASm4rtKACiiigAoornPF3j/wz4FjtX8VaoLAXhYQfuZJC+3G7hFJ43Dr60AdHRXm3/DQfwx/6Gb/yQuf/AI3R/wANB/DH/oZv/JC5/wDjdAHearo2na5apbavZxXkMcyTKkq5AdDlW/A1HregaX4i02aw1mzS6t54mhcElW2NjcAykMM4GcEdBXD/APDQfwx/6Gb/AMkLn/43R/w0H8Mf+hm/8kLn/wCN0AdjrfhTSfEOmxWGqRXD2sQ2rHDeTQZXG3axjdSwx2ORTLvwZ4c1Cx0yzvtGtLi20kr9hilj3LBtXaAAe2AOD6D0rkf+Gg/hj/0M3/khc/8Axuj/AIaD+GP/AEM3/khc/wDxugD0kAKoCjAHAA7VT1XSbDXNMm07V7SG8s512yQzLuVh/nvXBf8ADQfwx/6Gb/yQuf8A43R/w0H8Mf8AoZv/ACQuf/jdAHaXXhnRr3wx/wAI7cafE2keUsP2RcqgRcYXjB4wKsX+j2GqaLNpOoWyT2E0XkyQNnDJjGPWuD/4aD+GP/Qzf+SFz/8AG6P+Gg/hj/0M3/khc/8AxugDqvDPgbwz4O84+GtHt9PacASvHks4HQEkk49qsHwpoB8QvrraNZNqrgA3jQKZOBgYY9OBjiuN/wCGg/hj/wBDN/5IXP8A8bo/4aD+GP8A0M3/AJIXP/xugDs28K6C/iFddbR7I6qoIF55C+Z0Azn1wAM9ccVPfaHpep3tneahp9vc3Ni/mW00kYLQtjGVPauF/wCGg/hj/wBDN/5IXP8A8bo/4aD+GP8A0M3/AJIXP/xugD0mivNv+Gg/hj/0M3/khc//ABuj/hoP4Y/9DN/5IXP/AMboA9JorgdL+OHw71rVrXTNN8RCW7vJVhgjNnOm92OFXLIAMk45Nd9QAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcX4Z/5Kp44+mn/APolq7SvL4PGWieGPi14zj1m4nhaYWJQRWc02cQHPMaNjqOtbn/C2/B3/P8A3v8A4Kbv/wCNUAdpRXF/8Lb8Hf8AP/e/+Cm7/wDjVH/C2/B3/P8A3v8A4Kbv/wCNUAdpRXF/8Lb8Hf8AP/e/+Cm7/wDjVH/C2/B3/P8A3v8A4Kbv/wCNUAdpRXF/8Lb8Hf8AP/e/+Cm7/wDjVH/C2/B3/P8A3v8A4Kbv/wCNUAdpXF+IP+SweDP+vLU/5W9H/C2/B3/P/e/+Cm7/APjVcrrXxL8Lz/E/wrfRXl0be1tb9JWOm3IILiHbhTHk/dPQHHfGRQB69RXF/wDC2/B3/P8A3v8A4Kbv/wCNUf8AC2/B3/P/AHv/AIKbv/41QB2lRi2gW5a4WGMTsoVpQg3EehPXFcf/AMLb8Hf8/wDe/wDgpu//AI1R/wALb8Hf8/8Ae/8Agpu//jVAHaUVxf8Awtvwd/z/AN7/AOCm7/8AjVH/AAtvwd/z/wB7/wCCm7/+NUAdpRXF/wDC2/B3/P8A3v8A4Kbv/wCNUf8AC2/B3/P/AHv/AIKbv/41QB2lFcX/AMLb8Hf8/wDe/wDgpu//AI1R/wALb8Hf8/8Ae/8Agpu//jVAHaUVxf8Awtvwd/z/AN7/AOCm7/8AjVH/AAtvwd/z/wB7/wCCm7/+NUAdpRXF/wDC2/B3/P8A3v8A4Kbv/wCNUf8AC2/B3/P/AHv/AIKbv/41QB2lBGRg8iuL/wCFt+Dv+f8Avf8AwU3f/wAao/4W34O/5/73/wAFN3/8aoA6e10XSrDH2HTLO224x5NuqYx06CrckaSxtHKiujDDKwyCPpXG/wDC2/B3/P8A3v8A4Kbv/wCNUf8AC2/B3/P/AHv/AIKbv/41QB08Wi6Vb2LWcGmWcVq0nmNAluoQvkHdtAxnIBz7Vdri/wDhbfg7/n/vf/BTd/8Axqj/AIW34O/5/wC9/wDBTd//ABqgDtCMjB5FR29vDaQLBawxwxJ92ONQqr9AK4//AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDtKK4v/AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDtKK4v/AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDtKK4v/AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDtKK4v/AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDtKK4v/AIW34O/5/wC9/wDBTd//ABqj/hbfg7/n/vf/AAU3f/xqgDp9a/5AOof9e0n/AKCaxvhr/wAkq8K/9ge0/wDRK1j6r8VvCE2jXscd9eF3t5FUHSrsclT3MVZXgL4n+FNP+HPhyyu727We30u2ikVdMunAZYlBAZYyDyOoJFAHqdFcX/wtvwd/z/3v/gpu/wD41R/wtvwd/wA/97/4Kbv/AONUAdpRXF/8Lb8Hf8/97/4Kbv8A+NUf8Lb8Hf8AP/e/+Cm7/wDjVAHaUVxf/C2/B3/P/e/+Cm7/APjVH/C2/B3/AD/3v/gpu/8A41QB2lFcX/wtvwd/z/3v/gpu/wD41R/wtvwd/wA/97/4Kbv/AONUAFt/yXbUv+xbtP8A0puK7SvN/DHiPTPEvxp1S70eWaWFPD9tGzS20sJDC4mJGJFU9GHP19DXpFABRRRQAV5t42/5Lt8MP+4t/wCky16TXm3jb/ku3ww/7i3/AKTLQB6TRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUV85PH8Orn4rePrz4kQLN5d9DHbnyp5BEqx4Zj5QO0E7eW449jXuXg6y8PWHhW0j8GCAaM4MlubeQuh3EkkEknrn6GgDkfjJ/zIX/Y56d/7Ur0mvNvjJ/zIX/Y56d/7Ur0mgAooooAKKKKAPKdY+KniGw0vxVrFloOmTaZ4b1B7GUy6hIk0pUp8yqIiuP3i9W9an1H4m69Z/Ei20SLwvO9q9pNKYvtdqJJSjqA6sZgqrg/dbDcjjrXlXjDR/tPhv4nal/wjWl3nka9Mv9rzXG25tuYvlRPLORz/AHx948esfjrT7aLWdV1Wyt7rUL/Tb6Oxubq60/SfIdpXQnMZQM7EMMOynB6kDNAHtPwp8dav428Px3OsaSbd8zZu45YvKk2zMgUIJGcEAckgAkHB5FbnibXPEek3ECeHvCMuvxyITJImoQ2/lHPAxIQTn1FeffBfRbDRfEesaZp6X9lPpsUf2m3vLewYyCYs6nz7dSzY2nguQAQMcDHpPjLxJb+EfBuqa7dMAtnbs6Kf436Iv4sQPxoAoeA/GkvjOy1R7vSJNIu9L1CSwuLWSZZSHRVJO5RgjLY4yDjIJrc1vUDpHh/UNSWMSmztZZxGTjdsQtjPbOK8Z+Bg1Pwl4lvfDnimWVrzX7KDWrZ5s7ncqRMhJ6uDjj0UmvWvGf8AyIev/wDYNuf/AEU1AGNpXjTUk+H0/i/xRptpbWA09NQhi065aeVoym8hg6IAwBXgEjOeRjmppPxWGsmye08EeLxbXvlmK6fT4xEEfGHLCU/Lg5z6VB/Yd74l/ZvsdG0ry/td74etoovNfauTCnU4NZd54l8ZfC7R9Fl8U2uiXnhyIwWNxJYGVZ7MYCLIxc7XAxzgL9BmgDqdb8U65oXjDSLC50zT5NK1a++xwXEd4/nqfLZ8tGY9o+6Rwx7V2FcH8RP+Ro8Bf9h7/wBt5a7ygDi/DP8AyVTxx9NP/wDRLV2lcX4Z/wCSqeOPpp//AKJau0oAKKp6vpx1bRrvTxd3NkbmJo/tNpJsliyPvK3YivEfiF4e8NeFodOj+HN0yeOft8UdulvfPNcXB3fP567jlMZJLDA5HTIoA96orkvFvg7w9rmoRap4uvJmsLaHy/sVxeGKz3ZJ8xlyAWxxycYHSuT+EZSPxt4ttvC91Nc+C4WgGns0rSwrNt/erC7E5UHrg46fiAes0V8yeDofhTcQ63e/EKCJ9QudduiJpI7gxxoXG1S6DYvOT8xB5r6O0a00+w0SytdFWNdOihRbURPuTy8fLtOTkYxzmgC7XF+IP+SweDP+vLU/5W9dpXF+IP8AksHgz/ry1P8Alb0AdpRRXjfhzS1+Mmu65rXime4m8PWN/Jp+maVFO0ULhPvSyBSN5bI6n1HSgD2SsuDxHptz4ouvD0UzHUbS3S4mj2EBUY4U7uh+lZGg+ALPwpbanb+GNR1CwgvIdkFu85nisXww8yJJM4OSDg5B2jjFebavpHiDRPi5p8WgeMrzU/EWpRo2ql7C32w6fEc5KqmAxPyrjBJPJ6UAexav4h0vQp9Ph1W68iTUrpbO0Hls3mStnC/KDjoeTge9aVeDfEj4hx6nqvgyRfC3im1+x6/BcFbvSHiabAb5Iwfvuc8KK9c8OeKovEdnc3J0rV9IS3bDDV7JrUsMZ3AN1A9aAN2qtvqdhd3tzZ2l7bT3VmVFzBFKrPAWGQHUHK5HIz1rwbwP4s8Qaj4p8VaL4L1SxtNJa8m1eHUdYjkkf7OxCsYUyA0YdT8xOMe9ReBdX8aeF/C2ofEeaK21zSNXvHutSh8jyLsQISgnTB24ABOw9u/UgA+iKr3uoWemwrNqN3BaRM6xq88oRSzHAUEnqT0FLY3tvqOn299ZSCW2uYlmhkXo6MMqR9QRXk3iS/sPHHxitNF1K8toPD3heRZ7jz5Qi3d+w/dxDJ+bYpyQO5IPWgD2CiiuX8Y+J9b0GSzt/DvhO88QXN4H2mKVYoYSuP8AWOfu53ccc4NAHUUV55FpXxU1lfM1HxHo/hpSciDTbD7W4HoXlO3P0GKqweI/Ffgnxzo+g+M9Rttd0vXHaCz1SO1FtNFOORHIinaQcgAjGT9KAPTarXOpWNnd2trd3tvBcXjMttDLKqvOQMkIpOWIHJxXmfxN8WReHPiX4Rlvr24g0+0tr27uLeBzm6baiRR7B99t54B7n2rl/iGPHWpXPg3W9ZubTw6JvEFva2FlbQC4ntDMrASSO/yswAwUAxz7UAe+0VwPhjxL4h0/x1J4L8bPa3l1JaG907VLWIxLdRqQro8eTtkB54OMfrmeLfBPw+0y31LVPHGqzSXVwZJo7q/1JhNADnasChhjb0AAJOMHNAHqNFeGyeJPF2jfsw6ZJqE9xB4i1F0sLeefImVZJSEZied3lDg9ehPNWvGHhiz+FEHh3xN4XmuYZLfUYLTVTJcO4v4ZPld5ATy+cEHtn0AwAe0UV5M9hD8R/jZ4g0zXmludD8MWtvHHYCRkikuJlLmRwCCxABUZ4GPzn+HOqt4a1rxv4T1S+lk0/wAMyx3VrNcyF2itZYzJsLHJIQDvzz+QB6lRXkmrfGfXbHwivi218BSy+HZGXZdz6msUmxm2rI0SxsQpJHcnmvTNEu76/wBGt7rVbKKxupQS9vDceeqjJxiTau7Iweg64oAv0UUUAFFFFABRRRQAUUUUAUta/wCQDqH/AF7Sf+gmsb4a/wDJKvCv/YHtP/RK1s61/wAgHUP+vaT/ANBNY3w1/wCSVeFf+wPaf+iVoA6aiiigAooooAKKKZNH50DxFmQOpXcjYYZHUHsaAH0V454J0G38O/tG+ILK1ub26VtDilaW+uWnkZmkXOXbk17HQBxdt/yXbUv+xbtP/Sm4rtK4u2/5LtqX/Yt2n/pTcV2lABRRRQAV5t42/wCS7fDD/uLf+ky16TXm3jb/AJLt8MP+4t/6TLQB6TRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBydj8QvBV7p9/dx61p8EcEjpeJcsIZFZTtO9Gw3OMDI54rnfgHazQfD27uBDJb6dfatc3WmQuu3y7ViNgA7DIYj65713V94X0DU75bzUtD028uk+7PcWkcjr9GIzWoAFUBRgDgAdqAPNvjJ/zIX/Y56d/7Ur0mvNvjJ/zIX/Y56d/7Ur0mgAooooAKKKKAPBtQ8GSeK9O8c61pmgytBJqW+z0+a6vIf7SMZH2hmjEqgFypCfLxjp0xl6b8N7vxveeMX1jUdHtZrfVke4up9BEkwKxxuQG87KJ2ZMsDzzzX0bRQB4l8I4YrT4h+M73Sb/SbvRY7a03yaLpXkwzOEc/u1R2AK87gNxYsOnQ6KQ6l8ZPElleXtjc6b4I0ucTwwXSFJNWmU/KxU8iMeh6/j8vrlFAHGfEXwRP4r06zvdCuxp3iLR5TcaXeY4V+8bf7DYAPX6EZByrXxnd+Kfhz4ptdc0S70XWtN0+eO9tpYz5RYxNho5OjKcZ65HuME+kVDe2cGoWFxZXieZb3MTRSpkjcjDBGRyOD2oA4zRD4gX4M+F38JLYvqCaZYsYr4sEljESbkDL91iOAcECsTxTpHjH4m2tnoGr+HI/Dujfao59Snmv47h50Q58uJY/U4+ZsdOnY934c8J6R4Ttnt9DjuYYGVEEc17NOqKgIUKJHbYAD0XHb0FbNAHB/ET/kaPAX/Ye/9t5a7yufuPA+g3fiSHXrm3uZdQgmE8TPfzmON9u3csW/yxwSOF7mugoA4vwz/wAlU8cfTT//AES1dpXF+Gf+SqeOPpp//olq7SgDN1/X9P8ADOizatrMrQ2cBXzZFjZ9oJAyQoJxzyewryX4pt4Dh8J3HiLwpeafbeJ7iRZNOu9ElUXN1O0gyMRnMm4k7sg/nXtbKrqVcBlYYIIyCKyrTwp4d0++N7YaDplrdk5NxDZxpIT/ALwGaAMGfxF4W1UQeFvHLad/ajWcFzcWGoxqI2dl52bxtYgg/dyRXH+DL/TtJ+MGvWXgu5afwfaaQLi8jt3aa2tLsP8AdixkAlMkqvU59MV6tqmiaVrcIh1nTLPUIl6Jd26SqPwYGpbDTbHSrUW2l2VvZQA5EVvEsaj8FAFAHFav4/8AAd98NdR1A6rp02mXFpJvtt6h3ZlOYzH13knoRnNWPg3p+o6X8H/D1prKyJdpbFikgwyIzsyKR2whUY7YroW8LeH31Qam+haa1+DkXZs4zLn134z+tatABXF+IP8AksHgz/ry1P8Alb12lcX4g/5LB4M/68tT/lb0AdpXjXw31qw+G+veIPA/iy5h0pjqUl9pdxdOI4rqCXGArnjcCORnOTgdDXstU9S0jTdZtvs+safa38Gc+VdQLKv5MCKAOf8A+EwsfFa6vo3gXXLZ9XtbcMLxYTPbwuxIALD5WIx0yevQ4IryfR/hiP8Ahc2t2cfiHUk1u1023vY9bL7pftDMd5KH5WQj5dh4xxXvWn6ZYaTai20qyt7K3XpFbRLGg/BQBWXP4M0e61rVdUuIpXn1axFhdgTMqvDgjAxgqeTyCDQByXxYBXV/h4Gbcw8T2wLYxk7W5re+KdvqV38KvEVvoaPJeyWTqiRglnU/fAA5JK7gBWXJ8Cvh5KVMuhzOUbcu7U7s7T6j971rofC/gfw/4NN0fDtnLbfa9nneZdzTbtudv+sdsfePTHWgDwbxd5Hj7SfD9v8ACTTdRuLvTdMazv5rdPs8aWrIFe1ZnADSZzgDOMk8549M8deKdD8M/DlvC2krHLq17p/9nadosLLJMC8ewBlUnAUHkk4ODg16aRkYPIrD0TwX4a8OXUtzoWh2NjcSkl5oYQHOeSN3XHPTpQBzt1qX/CqPgXbzXzLLdaTpkUCqTkSXG0Kq/Tefyrih8ItI0f4H+INT8R6fFqHie60u51C6vbhA8kc/ltIAhP3cNjJGM8+uK9h1zw9pXiWyjtNdso7yCKZJ0RyQFkX7rcH3NaEsUc0LxTIskbqVdHGQwPUEdxQB5ZJ4/wD+Eb+APh++trhLvXb/AEu2t9OgBDvPctGq/d77WyT9MdTXptg10+m2zaisa3ZhQzrFnYJMDcFzzjOcVzOgfCvwT4X1Y6pofh62tr3JKzFnkMZPXZuJCf8AAcV11AHL658S/BfhyaaHWfEunwTwHEsCzCSVD6FFywPtiuJtZtQ+LfxD0TWbfTbnT/Cfh2VrmCe9j8uTULgj5SiHkIMA5/qcD00+HtFOpSaidIsDfSkGS6+zJ5rkAAEvjJ4AH4CtGgDzXxt4Pn1r4zeBtZNi91p9iLn7TIACsLhQ0RYHsW9O4rhPF+s+K4vGHhvwh4i06bxBPpOrrraXWlhHmurZCwTdFkeWwLbSSQpxxnGT9C1iaP4Q0XQta1LV9PtWGoapJvubmWVpXbphQWJIXj7o4H4CgDlPDWk+IPEXxGbxt4m0w6LbWli1lpmnSSLJNhmy8sm3hScABeePpkv1DVfhb4602a/1qfRrj7HuiklvSsNza7WIxlsSJzkjpzyK9DrJvvCvh7U70XmpaDpl3dDkT3FnHI4/4ERmgDwtk13Xf2cbnUG+16imi639t0uS7LGW5soZBtYk8nAL/gv0rqPib4j0nx7o/hbw94YvoNQuNc1O3n2QuGaK2QlnkcDlcYxg88H0r2BY0SMRoiqijaFAwAPTFUbDQdH0u5luNM0qxs55jmWW3tkjaT6kAE9O9AHnGlXlr4N/aA8VQ61cRWdv4ltLW8sp532I7QoUdNx43ZYnHpiqXhCOy8S6t8UfFlyr3Gg6qosYnhHNxDBA0crJ/eBGAD6g16xqWkabrNuINX0+1v4VO4R3UCyqD64YGqev6Vdz+Db/AEvwzPHpV3JavDZyxrsWBiMAjH3fqOnUUAeHeHTqHi34SaT4W1Pxh4Vs9CmiiWWQXf8Ap4gVgwgZDhVcYVS2e3TufoW3jihtYorfHlIgVMHI2gcfpXjMPg26HhsaP/wprQv7QW2EH9pT3Fq0TPjHmlwDMefm6bvfvXpXgTw1J4O8C6VoE1215JYw7HmP8RJLED/ZGcD2AoA6CiiigAooooAKKKKACiiigClrX/IB1D/r2k/9BNY3w1/5JV4V/wCwPaf+iVrZ1r/kA6h/17Sf+gmsb4a/8kq8K/8AYHtP/RK0AdNRRRQAUUUUAFMlljghaWeRY40GWd2wFHqSafUV3aW9/Zy2t9bxXNtMpSWGZA6SKeoKngj2NAHlGlaxpi/tLa9cNqNoIG0GFVlM67SfMHAOcZr1yub/AOFceCP+hN8P/wDgrg/+JrpKAOLtv+S7al/2Ldp/6U3FdpXF23/JdtS/7Fu0/wDSm4rtKACiiigArzj4k6V4mPjfwb4l8K6CNdOiG9E9p9tjtifOiVFO5+OxPAPT3zXo9FAHm3/CbfE7/okf/ly23/xNH/CbfE7/AKJH/wCXLbf/ABNek0UAebf8Jt8Tv+iR/wDly23/AMTR/wAJt8Tv+iR/+XLbf/E16TRQB5t/wm3xO/6JH/5ctt/8TR/wm3xO/wCiR/8Aly23/wATXpNFAHm3/CbfE7/okf8A5ctt/wDE0f8ACbfE7/okf/ly23/xNek0UAebf8Jt8Tv+iR/+XLbf/E0f8Jt8Tv8Aokf/AJctt/8AE16TRQB5t/wm3xO/6JH/AOXLbf8AxNH/AAm3xO/6JH/5ctt/8TXpNFAHm3/CbfE7/okf/ly23/xNH/CbfE7/AKJH/wCXLbf/ABNek0UAebf8Jt8Tv+iR/wDly23/AMTR/wAJt8Tv+iR/+XLbf/E16TRQB47rw+InjnVvC1vqXw+Gh2mma/a6nPdnW4LjCRFtw2Lg9Gzxnp05r2KiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4TQb+0s/it42F3dQQFhYbRLIFz+5b1rrf7b0r/oJ2f8A4EL/AI1R1LwT4V1i/e+1bw1pF9dyYDz3NjHJI2BgZYqScAAfhVX/AIVt4H/6E7Qf/BZD/wDE0AbH9t6V/wBBOz/8CF/xo/tvSv8AoJ2f/gQv+NY//CtvA/8A0J2g/wDgsh/+Jo/4Vt4H/wChO0H/AMFkP/xNAGx/belf9BOz/wDAhf8AGj+29K/6Cdn/AOBC/wCNY/8AwrbwP/0J2g/+CyH/AOJo/wCFbeB/+hO0H/wWQ/8AxNAGx/belf8AQTs//Ahf8aP7b0r/AKCdn/4EL/jWP/wrbwP/ANCdoP8A4LIf/iaP+FbeB/8AoTtB/wDBZD/8TQBsf23pX/QTs/8AwIX/ABrjtf1bTm+Lfg6RdQtSiWepBmEy4XIgxk59q2f+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaANj+29K/6Cdn/AOBC/wCNH9t6V/0E7P8A8CF/xrH/AOFbeB/+hO0H/wAFkP8A8TR/wrbwP/0J2g/+CyH/AOJoA2P7b0r/AKCdn/4EL/jR/belf9BOz/8AAhf8ax/+FbeB/wDoTtB/8FkP/wATR/wrbwP/ANCdoP8A4LIf/iaALms61pbaFfhdSsyTbSAATrz8p96x/hxq+mxfC7wvHLqFqjrpFqGVp1BB8peCM1c/4Vt4H/6E7Qf/AAWQ/wDxNH/CtfA3/QnaD/4LYf8A4mgDY/tvSv8AoJ2f/gQv+NH9t6V/0E7P/wACF/xrH/4Vt4H/AOhO0H/wWQ//ABNH/CtvA/8A0J2g/wDgsh/+JoA2P7b0r/oJ2f8A4EL/AI0f23pX/QTs/wDwIX/Gsf8A4Vt4H/6E7Qf/AAWQ/wDxNH/CtvA//QnaD/4LIf8A4mgDY/tvSv8AoJ2f/gQv+NH9t6V/0E7P/wACF/xrH/4Vt4H/AOhO0H/wWQ//ABNH/CtvA/8A0J2g/wDgsh/+JoA2P7b0r/oJ2f8A4EL/AI0f23pX/QTs/wDwIX/Gsf8A4Vt4H/6E7Qf/AAWQ/wDxNH/CtvA//QnaD/4LIf8A4mgDM066t7v45anJaTxzoPDtqC0bhgD9puOMiu5rJ0jwp4e8PzyTaFoWm6bLKux3s7RImZc5wSoGRmtagAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA//9k=) 　　　　![テキスト が含まれている画像

自動的に生成された説明](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDkRXhpZgAATU0AKgAAAAgABAE7AAIAAAAOAAAISodpAAQAAAABAAAIWJydAAEAAAAMAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAOW6g+eArCDkv4rlk4kAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzcxAACSkgACAAAAAzcxAADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIwOjA2OjA4IDE1OjE1OjI3ADIwMjA6MDY6MDggMTU6MTU6MjcAAACDXixwIADKT8lUAAD/4QsgaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyIvPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIj48eG1wOkNyZWF0ZURhdGU+MjAyMC0wNi0wOFQxNToxNToyNy43MTM8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIj48ZGM6Y3JlYXRvcj48cmRmOlNlcSB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6bGk+5bqD54CsIOS/iuWTiTwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCADAAbkDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD6RooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAorO8Qa1b+G/Dt/rN8ksltYQPPKsIBcqoyQASBn6kVxtt8XPtlrFc2vgDxpNBMgkjkTToirqRkEHzehBoA9DormPBvjqz8anUktdN1LTZ9MnEFzBqUKxyKxGeis3b1xTvFXjzSfB0sKavb6m3nlFie10+WZHdmKqm9V27yRwuc9PWgDpaK4Fvixby69Jomn+FfEl3qccH2g232SOEiMnAc+ZIuFzxmtrwX4tk8X2N1dPphsEgnMKg3kNwWI4YHy2IUg9Qfb8ADpKKKKACiiobyaS3sZ5oYGuJI42dIUIDSEDIUE8ZPSgCaivL0+NEEzO9tol1OraXbXlvbqSJ5ZppTEINuMAgg/NnGOa6e28ZSXPxOl8JJp3ywaUl/Pded/qnZ9oiKgYJxznP4UAdTRUN5dJY2M93KGMcEbSMFHJCjJx78V5rZfHrw1f2UV1BaTJHKu5Vn1TTIXH1R7sMp9iBQB6hRXndh8afD+oeINN0iO1uPtGpT+RC0V7Y3Kq2M5YQXDso464r0SgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOQ+JF7o0vg7VNB1PxBpWj3WqWUsMB1G7SEHcMbsE5IBPauJ0PxD4wsfDCQ6b4x+Gdxp2j28MEtz9omcRLgIhkYSYUsRgZxk9KX4lan4f0n42+GLrxbHHJpo0u5VlltGuRuLDHyKrH8cV5x4mv9F1PQfjJd+F0jTS5F0TyFjtmt1GHUNiNlUj5g3YZ696APVPhpqel6TrGuXGt+NfCd9qfiC/jmSHStQUrv27AiqzFiScYHNV/iu2m3EV1omq+MNUvZdQYeR4b020tZZiQcqFPlF0AIzvZhjB57VzPjLxL4A1i/wDBNt4Ngt49QTxRYPKYtJktT5YLA/O0ag/MV4z79q1/jhomhWOseFrxrPRbE32rs17dX1oGhk/dHmbBUsP+BD1oA5FPDvxafxxfTPqsra02gI0qxxwo0tuZsG1Wfy9glA+beFzn867/AOFF3pFqtnoml+MNStH0/dHJ4Z1SztbeZTg5BAiDtyd25WOT1PUV5dBb+F/+FqXSNqfw9Fj/AGUhWVtNY2Jk8zkKnn5EuOp3dO3evQfgbpGi3OveK7yG20a8az1RDZ3en2wWGMGLBMOSxUH03GgD22iiigArF8YWwvPCV/CdCi8QFkBXTJWVVuGDAqCW4GCAefT1rarF8Yw6dP4P1KPWRpxtPJJb+1CBbBgQUMmewcKaAPnrRPhXd33jzXpbODQ7m9022ie4sX05W05bhiSbQcbhtTZ+8XnOc5yRVG2sdOtdQ8X6j4/8L2tm95obahotlcIV+yJEzQRwgcYb/V8emDVHStJ0iGLRrLVdAsrvw/puprNqfi3T7OWSO5U7iISDGDs3kKSuVwF/HsPGtn/wsiTT7r4e+Gob3w94PHmI7q0SalhlLW0C4+ZAE645PA9wDufD/hm78GfAy1i0eOwspvsBvNWW6tXm84mHMgASSMh8ADOe3415wZtc8OfD+DxBHp+pWHhtIIpYZILq6VEjkKhMRrrG4Al1428Z5r1y68f+GvEvwxutQg1GztYb6KSy8nUrxbEpKVIaJncHawGf4W6ZwRXz3400vSNP+Ht1Fp914ZlaBIUj+yavp11csBIg48uxjmc46nzAcZJyMggHc6jB4k0HUdF1XxDpVxGV1CKGxudQee7SGeT5VOw6s+O+TtPFe/6et6mnwrqslvLeBf3r20bRxsfVVZmIH1Jr5rTSdAe6sZ5b/wAIL9juYrlDb+I9Lt2LIcjLRaejEeo3DNfS1le2upWUV5p1zDd20y7o54JA6OPUMOCKAJ6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiuf8ReONC8LX1tZ6zcXCXF1G0kMcFlNcFlUgMf3aNjBYdfWsv/hbPhT/AJ66r/4JL3/41QB1cmm2MupRahLZ273sKGOK5aJTJGp6qrYyAe4FQXmgaPqCXaX+k2N0t7s+1LNbI4uNmNm/I+bbgYznGOK5v/hbPhT/AJ66r/4JL3/41R/wtnwp/wA9dV/8El7/APGqAOtubG0vViF5aw3AgkWWISxhvLcdGXPQjPBHNE9la3UsMtzbQzSW774XkjDGNvVSeh9xXJf8LZ8Kf89dV/8ABJe//GqP+Fs+FP8Anrqv/gkvf/jVAHaVBBZWtrNNLbW0MMlw2+Z44wpkb1Yjqfc1yX/C2fCn/PXVf/BJe/8Axqj/AIWz4U/566r/AOCS9/8AjVAHaUVxf/C2fCn/AD11X/wSXv8A8aqOb4weDrcIbi71GIO4jUvo14u5icBRmLkk9qAO4pksMdxC0U8aSxuMMjqCGHuDXHf8LZ8Kf89dV/8ABJe//GqP+Fs+FP8Anrqv/gkvf/jVAHZhFVAiqAoGAoHAHpTYIIraBIbaJIYoxtSONQqqPQAdK47/AIWz4U/566r/AOCS9/8AjVH/AAtnwp/z11X/AMEl7/8AGqAOuNnbMpVreIqWLkFBgsep+vvTf7Osv+fO3/79L/hXJ/8AC2fCn/PXVf8AwSXv/wAao/4Wz4U/566r/wCCS9/+NUAdZ/Z1l/z52/8A36X/AAqZI0iQJEioo6KowBXG/wDC2fCn/PXVf/BJe/8Axqj/AIWz4U/566r/AOCS9/8AjVAHaUVxf/C2fCn/AD11X/wSXv8A8aroPD3iPTPFOlf2jok0k1t5jxbpIHiIdDhhtcA8EY6UAalFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFDXNasvDuh3er6rI0VnZxmSZ1QsVX1wOTVq1uYr2zhurc7op41kQkYypGRx9DXnvjfQtH8SeKR4a1nxd4g099ZtS0enW7xpbTKmAyqzRHLcbiu7OOcAVzPg1vGTwG88H61Hd2cV7d6Vb23ifUN0c8cTKI5YhDAp3YSQEEnAHfsAes2niPTb3xLqGg28zNqGnRxy3MZQgKsgyp3dDn2pPEviXS/COgXGs67ci3s7cDc2MlieAqjuSe1eR+Em8e/wDC+vEn2iHw59oMNj/aYSWfYIcceSSuS+3P3sDOKZ4+tvEZ+MVhc69qapotlbPd6d5WhXN/bwy7toEkcbZMm3c24nAxwOaAPU9I8f8AhPXHtotM8SaVPc3SgxWqX0TTEkZxsDE5x1Hauhr5h0vxYniL4syar/wlNox0a3EWkzW3hO6mjuN6/v3ECPuRkJCEsTkEYxXsvwy8T634jh1pfEAUmzvfKtJvsElmZ4No2yeVISwyd35EdqAN/T/GPhzVdYl0nTdbsbrUISwktYp1aRdpw2V68HrW1XB/CL/kV9U/7D2of+lDVytz4dt9C/aY8MSwXd/dPe2N5LI17dPOVO04C7j8q89BxQB6ZrXjHw54cu4rbXtbsdOnlTfHHczqhZc4yAe2RW1XB/G3/kjHiH/rlH/6NSu8oA4vU/8Aktnh7/sC3/8A6Nt67SuL1P8A5LZ4e/7At/8A+jbeu0oAKKRgWUgMVJGAw6j868n8W/Czw5pXgHWtWvbzUJ9atbSa7XXbi8f7SsqqWXBBAAyANigD8cGgD1mivPtG0Wf4ifDPwpceKL/UIg9rHcXtvbzeSL7KceYVw2Dw2AR1+mMG30i08DfHzQNH8ICS107VtOuJNR05JWaGIICY5QpJ2ksNvHv6mgD1+iuHv/hR4d1m+1G/8TSXur3F07NHJdXTAWSHokIUgIB64z6k1S+BuqX+q/DSNtQu5b5Le8ntrW8mbc08CPhGJ79x+FAHotcX8TP+Qf4d/wCxk03/ANHrXaVxfxM/5B/h3/sZNN/9HrQB2lFFFABRRRQAUUUUAFFeZ/FO5udW8VeD/A9vdz2ttrt1NJftbyFHe3gTeY9w5AbnOOePwNKOxh+HHxq0DS9BMltofie1uI5LEys8UVxCocSICTtJBCnsc/kAes1xfwu/5FzVP+w9qX/pVJXaVxfwu/5FzVP+w9qX/pVJQB2lFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHmnxwl0ybwhBpjRPceIru5QaFFbvtuEuQwxKhHKherHpjjIyK5zwv4ZuPEnwaj8N2VuNI8VeEtR2xTS5Kx3SSeZ5gbHKyK5z/vd8DPqcPhDRYvF1x4n+yGTVp4li+0SyM/lIBjCAnCA99uM1pX9il/p11ZmWe2FzG0bTW0hjlTIxuVhyGHY+woA898Kb/wDhoDxv5u3f/Z+n7tvTOw5xWH4+sZbnXNSk1O11KXTI3+a48Q6qLXSIAVGNkUR3z/7rd8jIr0Hwp4D0zwleX19aXWpX99fqi3F5qV21xK6pnaNx7DP8qjufhx4dvdan1e/guL2/kJMUt3cvOLUnvCjkqhB5GBxQB886ZL4k0y5trXwm1zcy6Rbzaj4dubqzMRuoCoS6iSNmLNGch493OEx3GPc/hZp3hufR28T6JqU2t6jqg/03VLxs3DMOsTKOIwvA2DgADqMGtXRfAdnpPiRtfu9U1PWNU+zfZY7jUJIz5MROSqLGiKMkdcZ96tWXgrQNN8VXHiLTrBbTUrqMpcPA7Ik3OdzIDtLf7RGeTQBg/CMhfCuqljgDXdRJJ7f6Q1c/4i1jTH/aK8HXCajaNDHpt4HkE6lVJU4BOcCul8DeHvEnhuTUNO1SDSZtKub66u0mhupGm/eyFwrRtEF6HB+b860/+FceCP8AoTfD/wD4K4P/AImgDF+NMiTfBPXpInV43giZXU5DAypgg9xXfVwnxF8L694g8HzeFvCtpo1tps9vHDvuLiSE24RwQqRpEy7QFAHIx6cV3dAHF6n/AMls8Pf9gW//APRtvXaVxep/8ls8Pf8AYFv/AP0bb12lACMwVSzEAAZJPavEfEvjDSfit4m/4Re21uzsvCFjKsmrXst0sR1B1OVgiyQSmQCWHpwem722SNJomjlRXjdSrIwyGB6gjuK53/hXHgj/AKE3w/8A+CuD/wCJoAq+J/iJ4X8F+D11ia9t57TBis4bN1c3DqMbExxxjBPQd65j4ZfYpdYu/FfinXdJm8Va7tjSzgvY3+xQZ+S3QBuT0Jx1I9ck9zN4I8KXFjb2U/hjRpbW1Lm3gfT4ikJcgttUrhckAnHXFNtPAnhCwvIrux8K6JbXMLB4podOhR0YdCGC5B9xQB598R/iBDrmrP8AD7wvrNpY3E+Y9Y1WeZUjsYeA6KSQGlIOMDpz0OSvonhKLQLHw7baV4UurW4sNPjWFfs86y7f94qfvHkn1JNQz/D/AMG3NxJPc+EtCmmlYvJJJpsLM7E5JJK5JJ71o6ToGj6BDJFoWk2OmRytukSztkhDnpkhQMmgDQri/iZ/yD/Dv/Yyab/6PWu0ri/iZ/yD/Dv/AGMmm/8Ao9aAO0ooooAKKKKACiiigDy74pMugfEDwH4yuyE03TbuezvZmzthW4j2K7eig5yfcVDqt5a+Mv2gfCsWi3MV7beGrS6vL2aBw6I0yhETcON2QDj0zXqdxbw3du8F1DHNDINrxyKGVh6EHg1X03SNN0a3MGkafa2ELHcY7WBYlJ9cKBQBcri/hd/yLmqf9h7Uv/SqSu0ri/hd/wAi5qn/AGHtS/8ASqSgDtKKKKACiiigAooooAKKKasiNIyK6l0xuUHlc9M0AOopskiRLuldUXpljgU6gAoqNbiF7h4EmjaZAC8YYFlB6EjqKkoAKKCQqkscAckntSI6yIrowZWGQwOQR60ALRRTXdY0LyMqKOrMcAUAOopCyhCxYBQMkk8YpaACiiigAooooAKKKKACiiigAormvG/jO08IeFtT1ISW093Yweatm9wEaQ8YHcjr6Vo6Hr9jrmn2s9tdWzzTW6TPDFMrlMgE9OeCcUAalFcxpXi59R+I2veGGtFjTSbe3mW4EmTL5qk4244xj1qPx142XwnZ29tp1p/amv6g/ladpiNhp27sf7qKOSx47UAdXRXC6f8AEzf4m0jw3rXhnWdO1fU42dRIkDRLsXLsSkzMFGMAkc13VABRXK+G/G7eJLu88nQr22020nnt31KeaARF4nKsNofeBkHkrioIvi14Gm1BLOPxFbl5JfJSUo4hd/7omK+WT9GoA7GiuW8a+NJfBVhJqNxoF9f6bDGHuLu2lhAhy20Aq7hieR0B611NAHC69e2tj8Z/D0t7cw20f9j3y75pAgz5lvxk11H/AAkmh/8AQZ0//wACk/xrkfE2kabrPxj8PW+sadaX8A0e+fyrqBZV3CS3wcMCM8n8zW//AMK+8Gf9CjoX/gth/wDiaAL/APwkmh/9BnT/APwKT/Gj/hJND/6DOn/+BSf41Q/4V94M/wChR0L/AMFsP/xNH/CvvBn/AEKOhf8Agth/+JoAv/8ACSaH/wBBnT//AAKT/Gj/AISTQ/8AoM6f/wCBSf41Q/4V94M/6FHQv/BbD/8AE0f8K+8Gf9CjoX/gth/+JoAv/wDCSaH/ANBnT/8AwKT/ABo/4STQ/wDoM6f/AOBSf41Q/wCFfeDP+hR0L/wWw/8AxNH/AAr7wZ/0KOhf+C2H/wCJoAv/APCSaH/0GdP/APApP8a474j67pE9h4fEOq2UhTxFpzsEuEOFE6kk89B610X/AAr7wZ/0KOhf+C2H/wCJrkfiH4K8K2djoLWnhrR4DL4g0+KQxWES70adQynC8gjgjoaAO9/4STQ/+gzp/wD4FJ/jR/wkmh/9BnT/APwKT/GqH/CvvBn/AEKOhf8Agth/+Jo/4V94M/6FHQv/AAWw/wDxNAF//hJND/6DOn/+BSf40f8ACSaH/wBBnT//AAKT/GqH/CvvBn/Qo6F/4LYf/iaP+FfeDP8AoUdC/wDBbD/8TQBf/wCEk0P/AKDOn/8AgUn+NH/CSaH/ANBnT/8AwKT/ABqh/wAK+8Gf9CjoX/gth/8AiaP+FfeDP+hR0L/wWw//ABNAF/8A4STQ/wDoM6f/AOBSf40f8JJof/QZ0/8A8Ck/xqh/wr7wZ/0KOhf+C2H/AOJo/wCFfeDP+hR0L/wWw/8AxNAF/wD4STQ/+gzp/wD4FJ/jXN/CqRJfDGoyROro+u6iyspyGBupMEGtX/hX3gz/AKFHQv8AwWw//E1kfCa3htPCuoQWsMcEMeuaiiRxqFVVFy4AAHAAAA/CgDuKKKKACiiigAooooAz9dl1SDR5pdBSxe8TBUahK0cIXPzFmVWI4z2ryFz448E/EHUtbu4/DMt74vaGOCzjubp/mtoSMLtg7rzz3r2XUdOtNW0y40/UYEuLS6jaKaJxkOpGCK8E8Hzi0+Ptl4U1XXBqFh4ZiuoNDnkU7jI6JugZ+jPHGSv4dugALHxd+IZ1z4XTWU3hXxNYTtNbO8t3pMkUCssqkje3qRgZ68V3vif4kXmm/De68S6R4X1dp0Eii31C2+zmDapJllVjkJgduvTjOa4Tx/P4gh+E3ijRfExuLptM1q2W01KZMfa7d5kdOQACyhtpx7CvT/iPqR07wnJul0GOC5fyJ/7enMduY2ByMDlzx9zjPPNAHit7bap4e8D3HiTUtE8Tx+I9bVN+rQ64kEM1xL/qsxwz52LuAUFRgADivRPDvi/VvBmkeGNA8Y6Frhu76eOw/tK8vbe4824fJJJWVm29cZHQCvFde0LQ7C+tPEeq6db33htw1tPNBpw0yINIcCS1j3CScoDu+YEEDg+nW+BPDr+JnttF0fRtD8P6n4deIalrqW8Ul1KynMcluhGAHUK3mMMfMcA9wD1H4jWuu6ojWB1C30LwmLYy6vqnm/v3TJ3QoP4AVHLc9cD0OTpk/g/xB4Q0Zhq+o+DrOCArZWSaybF5IM4SRlVwWDBcgtzyfXNZv7Qmga5qXgnUL062INC0+1SU2MUeJLqfzAuZG6bACCAB17cA16EPCHhrW7GxudZ8PaVqFwtrHGJbuyjlcKF4XLKTjk8UAcvbaV4MtrqKdfiRqkhicOEl8WO6Ng5wyl8EeoPWuL+P2o6rpE17cslvHa6jpsmmrbrqTNJdLlX87yPKwvlkNlt3II5HQ7ek+CvCsnx98QafJ4Z0d7KHR7aSO2awiMaOWOWC7cAn1rL+JtrJdfEI+GdJ0LS01TxMirLfzaqfMezjALRbGjIg3kFQULEkE4z0AOY1nxhdx/C/xRqF01o02vmLTwBqju9mcKoga3aJPLxGspLE8nHUHNen+HfGXiLTPFOjeAbjwdaW7x6dHMZotZEoitkYRb9vkjJyOmRn1rlovDNp8YIdQ1m/8N2elR2Vpd6Zb38ep+ck0yAIkxUwpuVfmw5bt06EX/gLdXvjC81Txpq5iM8dtb6NCqSBiBCgaRjjpvZg359epAPaqKKKACiiigAooooAKKKKAPOfHknhjR/FmnXfjDwdo93pOogW82u3NpHK1tP0jWXchIjI4D5wD1wBmuC8M6RoF1oen+JLu/vvCv8AbWtXdrptz4ftYLMeRJIBFFK6xE7cw/KSerc56jtfiZrT+KJZfht4ZiW71LUFVdSuGTdFptuSCXc9N5H3V69+OKZo3gmWP4U6t4M8fzRw6Pp0zR2epiVIy1qhWSOUnojKeCT/AHfxIBz/AIS8FSQ/HrxJD/wlfiOQ6bDYzmV7xC93kbvLmOzDJxjAA4J5qn8RdF0XQfikfEk95b6i09m0T2M3iG4trpZi+R5QgVpNgXjYBj5q6jwFqFjqvxx8a3ekX8GpWn2GwRbq3mWVGIRgfmXgng1keKtRtIfGupxabq1qbmaT97ZeFdOU6lN8oBFxdMSsIHIJ+Ugc8HNAHB+EVk17xymqt4Ykki8QwtFogn8XXSNAtuoM8fnIGdtxO/Y+MYPFerfB68eDVfFGi3eofariC+85IYr2e+is4yABD9okUBmUjkZzz65x5cngDUNS8WXfhm2ktdLuNS09tVt7Oxu3n/syeIiONpJiclpFZlcjGSwPOBXsXwz8UaQbRPCbaMvhjXbCMm40byiikDrLE3SRCec5J55z1IAfC21gvvBetWl5Ck9vPrepRyxSLuV1M7ggjuCKp/GtdK0v4NXWixWUO67MVjpdlEgGZi42BF7EAE8envVj4a339meAtevjb3F19n1nUpPItYzJLJidztRR1Y9AK4XRvEuo6v49Pi/x74L8XGayzFo+mW2izSQ2akDdKWIG6Q+uOPy2gHbfFS2uLL9nvUbW9k825gsLeOWT++4eME/iQa9Jrzf4sagNW+Aes3y2t1aCe3jcQXkJilj/AHycMh5B9q9IoA4vU/8Aktnh7/sC3/8A6Nt67SuL1P8A5LZ4e/7At/8A+jbeu0oARtxU7CA2OCRkA/SvLvEXgzxDp3g3VtduvH+tf27Z20t4ssMohs02KX2fZ/u7MDGTk989q9Sryjxtq8/xJ1t/h54TmJsVYHxDqsRytvGDzbqehkYjB9OQf4sAG1px1/4hfD3wzqEGty6D9sgSfUms4gJpQU6Ruc+WC3OcE4I/HItX1TwT8Z9D8OQ67qWr6RrtncO0GpTm4ktZIl3bxIfm2nGMHufpjudW1fQ/AfhP7VqM0Wn6Xp8KxRr7KuFRB3OBgAVyHw80bVde8S3nxF8V2zWd3fQC20rT5PvWVnnI3f7bnk+mT64ABqat4Gv9Uv8AUb/VvHGuWkBYvaRafcC0js4wP4sD94QcnLcY7VD8G/EWreJvh5Fea7OLyaK6mt4r3YF+1xo2FkwOPUf8Bq34i8KeEPijYf6fP9vitJJIN9nfMojcHDKwVtpII/iBx+NZPwZ1q8v9H1vSbi6S/tdB1SXT7K+SNUE0KY2jCgLkDuB3H1oA9Hri/iZ/yD/Dv/Yyab/6PWu0ri/iZ/yD/Dv/AGMmm/8Ao9aAO0ooooAKKKKACiiigDz34m65qq6z4Y8IeHr59OvPEV1Is15EoMkFvEoeQpnoxB4PbB+opWdzqfgb4u6V4buNZv8AVdF8RWszWv8AaU3my21xCNzASEZKlccHuR+LfiNjSvi98OPEF0QtlHc3VhLIxwqSTRbY8n3O78qPF2NY/aF8C2FqQ76PbXt9d7TkxJIiomfTLKB+NAHqNcX8Lv8AkXNU/wCw9qX/AKVSV2lcX8Lv+Rc1T/sPal/6VSUAdpRRRQAUUUUAFFFFAGR4o0/V9V8Pz2Xh/VV0i8mwv2sw+YY0z820ZGGxnB5x+oytI+GnhvSdD0nTUszP/ZV0L2G4lY+a9zzmVmHJJz9OnoK6yigDz74ieEPFfjcNosN/o9r4dlkhlkZ4JWuwUYMQDu2YJHp3rpfFWmatq2mxW+h3WnWsvmhmmvrL7T5YAPzIu4DeD0zkda3KKAPI9c+FOtXK3yW89jrV9qNm1rNruuXMjT2wcMriCBI/LQbWONrLyeemTuX3wriFro934c1e40bxDo9jHZQ6nGgk8+JFC7JozxIvGeeQcc8V6BRQByvjnwvqHi34Z3/h4Xdut/eQJG1wyFIywZWJwMkA4PHNbM8GpQeHDb6Q9quox24SBrpWaHeBgbgpDY+nNaNFAHC+EPCPiKy8bap4p8W3umS3t9aRWgg0yKRY0VCTnLknJzVbxt8M7jx7rDpq95p1voxCArb6apvZAvO03Dk7RnONoBwceteh0UAeVD4PalHYp4Zg8X3aeC97M+miFVuChOfI88cmMnOe+DjnrW1bfDRPD/ja21zwVepo1pJGkGpaWsO6C5RFIRlGRtccDP8A9fPd0UAFFFFABRRRQAUUUUAFFFFADRGiyM6ood8bmA5bHTJpJoY7iCSG4jSWKRSjxuoZWUjBBB6gin0UAVbDS9P0qIxaZY21lGxyUt4VjBP0AFSG0tisym3iK3GfOGwYkyMHd68cc1NRQBm6T4d0TQBINC0fT9MEn3xZ2qQ7/rtAzWjsXeH2jcBgNjnHpS0UAcx4b8EReF9Qup7DW9Vltbmea4awnaEwLJK25iMRh+p4+b866eiigDmPGXgeLxtZvY3+t6tZWEsYSa0smhVJcNuBYvGzZyB0I6V09FFAHF6n/wAls8Pf9gW//wDRtvXaVxep/wDJbPD3/YFv/wD0bb12lADJokngeKTJSRSrYYg4Ix1HI/CuAi+BHw6t1Ig0GaIMckJqd0Mn8Ja9CooA4u/+EfgnU9EsNIv9IknsdPeR7aJr+4+RnILEnzMt07k47YzSaZ8IvBWj6pbajp+l3Ed1ayCWJ21K5cKw6Ha0hB+hBrtaKAOR1P4WeDNX1Ce9vNDjFxcnNw1vNJAJ/wDfEbKH685BzXRaTo+n6DpcOnaNZw2VnCMRwwrtVe5/EnnNXKKACuL+Jn/IP8O/9jJpv/o9a7SuL+Jn/IP8O/8AYyab/wCj1oA7SiiigAooooAKKKKAKGtaJpniLSpdN1yyhvrObG+GZcg46H2I9RzVLw74M8P+FGuH0HTY7WW5x50xdpJJMdAXcliB2GcVuUUAFcX8Lv8AkXNU/wCw9qX/AKVSV2lcX8Lv+Rc1T/sPal/6VSUAdpRRRQAV5f8AE+K/1b4jeA/Ddtr2raNZ6odQN1JpN2beV/KhV0+YA9DnqD1NeoV5t42/5Lt8MP8AuLf+ky0AH/Cm/wDqo/xA/wDB5/8AYUf8Kb/6qP8AED/wef8A2Fek0UAebf8ACm/+qj/ED/wef/YUf8Kb/wCqj/ED/wAHn/2Fek0UAebf8Kb/AOqj/ED/AMHn/wBhR/wpv/qo/wAQP/B5/wDYV6TRQB5t/wAKb/6qP8QP/B5/9hR/wpv/AKqP8QP/AAef/YV6TRQB5t/wpv8A6qP8QP8Awef/AGFH/Cm/+qj/ABA/8Hn/ANhXpNFAHm3/AApv/qo/xA/8Hn/2FH/Cm/8Aqo/xA/8AB5/9hXpNFAHm3/Cm/wDqo/xA/wDB5/8AYUf8Kb/6qP8AED/wef8A2Fek0UAebf8ACm/+qj/ED/wef/YUf8Kb/wCqj/ED/wAHn/2FX73UfibcTX0+k6NoNna20jrb299PJLNeKvRg0eFj3dQDk+uOta3w/wDGEPjzwPYeIYLdrX7UrCSBm3eW6sVYA9xkcH0IoA838VeFb/wDq3g+/wBN8ceLr/7d4ls7Ce31LVTNC8UhYsCoUZztxz6mvbK82+Mn/Mhf9jnp3/tSvSaACiiigAooooAoa5rVl4d0O71fVZGis7OMyTOqFiq+uByatWtzFe2cN1bndFPGsiEjGVIyOPoa898b6Fo/iTxSPDWs+LvEGnvrNqWj063eNLaZUwGVWaI5bjcV3ZxzgCuZ8Gt4yeA3ng/Wo7uzivbvSre28T6hujnjiZRHLEIYFO7CSAgk4A79gD1m08R6be+JdQ0G3mZtQ06OOW5jKEBVkGVO7oc+1J4l8S6X4R0C41nXbkW9nbgbmxksTwFUdyT2ryPwk3j3/hfXiT7RD4c+0GGx/tMJLPsEOOPJJXJfbn72BnFM8fW3iM/GKwude1NU0Wytnu9O8rQrm/t4Zd20CSONsmTbubcTgY4HNAHqekeP/CeuPbRaZ4k0qe5ulBitUvommJIzjYGJzjqO1dDXzDpfixPEXxZk1X/hKbRjo1uItJmtvCd1NHcb1/fuIEfcjISEJYnIIxivZfhl4n1vxHDrS+IApNne+VaTfYJLMzwbRtk8qQlhk7vyI7UAb+n+MfDmq6xLpOm63Y3WoQlhJaxTq0i7Thsr14PWtqvP/hXD9o8G61B5kkXma3qSeZE2HTM7jKnsR2rifiN4T8HeEvCMp0fUrpvG0ZQWNymoSS6jczlh1UNkg5ORjAzxzQB65rXjHw54cu4rbXtbsdOnlTfHHczqhZc4yAe2RW1XnHxYN437P+qHVAovjZW5uQo48zfHux+Oa9HoA858YaHFr3xe8O2s17qFkBpF6/mafePbycSQcbkIOOenfA9K0v8AhWVn/wBDN4t/8KC5/wDiqNT/AOS2eHv+wLf/APo23rtKAOL/AOFZWf8A0M3i3/woLn/4qj/hWVn/ANDN4t/8KC5/+Krs2JVSQpYgZCjqfzrzXUvE3xNs/D954j/4R/RLaztEac6TPPI120K8kmRTsDbQTtwfrnggGv8A8Kys/wDoZvFv/hQXP/xVH/CsrP8A6Gbxb/4UFz/8VT5vFus614V0PU/A+jR3r6yiyeZeziOGyUruJkxlm54wo6jqOM0NK8ZeI9N+Iln4R8bWmmmTU7aSfT7/AEwuscrRjLxsjkkEDJznH58AFz/hWVn/ANDN4t/8KC5/+Ko/4VlZ/wDQzeLf/Cguf/iqm+JPjf8A4Qfwsbq1hW71W6cW+nWjAnzZT3IGDtUZJ6cDGRmp/hx4lvPGHw70nXtTjgiur2NnkS3UqgIdl4BJPQepoApf8Kys/wDoZvFv/hQXP/xVcr4+8AWtjZaGya/4lm87XrCEi41qeQKGmA3LluGHUMOQeRXr1cX8TP8AkH+Hf+xk03/0etAB/wAKys/+hm8W/wDhQXP/AMVR/wAKys/+hm8W/wDhQXP/AMVXaUUAcX/wrKz/AOhm8W/+FBc//FUf8Kys/wDoZvFv/hQXP/xVdpRQBxf/AArKz/6Gbxb/AOFBc/8AxVH/AArKz/6Gbxb/AOFBc/8AxVdpRQBxf/CsrP8A6Gbxb/4UFz/8VR/wrKz/AOhm8W/+FBc//FVN4+8Z3HhaHS7LR7FNQ1vWroWthbyPsjBxlpHI52qOTjnn8ap6J4x1y18fL4R8a2lhHd3dqbrT73Ti4huAv34yr5KsvXqQR6UATf8ACsrP/oZvFv8A4UFz/wDFVD8IrVbLwhfW6STSiLW9QQPPIZHbFy4yzHkk45Pc5Peu6ri/hd/yLmqf9h7Uv/SqSgDtKKKKACvNvG3/ACXb4Yf9xb/0mWvSa828bf8AJdvhh/3Fv/SZaAPSaKKKACiiigAooooAKKKKACiiigAooooAKKKCMjFAHHeP/D3i/X7FofCXieHR42iKSwtagvKTnOJskx5GBwpI6+1Vfg7qenX3w8hstL0v+yf7IuJdPuLLzfN8qZDl/n/iyW3Z9WNNXwL4qsI5rTQviFe29hK7MiXtjHdzQ7iSQkzEHHPG4NgVv+DfCGn+CPDqaTpjSygyNNPcTtukuJW+9Ix9TgfgBQBynxk/5kL/ALHPTv8A2pXpNebfGT/mQv8Asc9O/wDalek0AFFFFABRRRQB5p8cJdMm8IQaY0T3HiK7uUGhRW77bhLkMMSoRyoXqx6Y4yMiuc8L+GbjxJ8Go/DdlbjSPFXhLUdsU0uSsd0knmeYGxysiuc/73fAz6nD4Q0WLxdceJ/shk1aeJYvtEsjP5SAYwgJwgPfbjNaV/Ypf6ddWZlnthcxtG01tIY5UyMblYchh2PsKAPPfCm//hoDxv5u3f8A2fp+7b0zsOcVh+PrGW51zUpNTtdSl0yN/muPEOqi10iAFRjZFEd8/wDut3yMivQfCngPTPCV5fX1pdalf31+qLcXmpXbXErqmdo3HsM/yqO5+HHh291qfV7+C4vb+QkxS3dy84tSe8KOSqEHkYHFAHzzpkviTTLm2tfCbXNzLpFvNqPh25urMxG6gKhLqJI2Ys0ZyHj3c4THcY9z+FmneG59HbxPompTa3qOqD/TdUvGzcMw6xMo4jC8DYOAAOowa1dF8B2ek+JG1+71TU9Y1T7N9ljuNQkjPkxE5KosaIoyR1xn3q1ZeCtA03xVceItOsFtNSuoylw8DsiTc53MgO0t/tEZ5NAHLfDjVLbRPh/r+p37OttaazqU0pRC7BVncnCjk1j/ABBufhlqngrUfFNrf6VHqr27SWWp6fMsd404T5AGUhy2QAVPbg4rqfAegeJfDE1/YajFpUmlz391eR3EF1IZx5shdVMZjC9+Tu/Ot9PCXhyLUPt8fh/S0vM5+0rZRiTOc53YzQB5340udVu/2W5bjxCjpqcmm2zXIkXa2/fHksOxPUj1zXrdcR8TtA8S+LPDN34f0GLSVtb2JVluL26kjeNg4b5UWJgRhR1I6129AHF6n/yWzw9/2Bb/AP8ARtvXaVxep/8AJbPD3/YFv/8A0bb12lABXnXxZ8STrpa+CvDg+0eJPEcbW0MSn/j3gbIkmc/wqF3YPryM4NehTLI0DrC4jkKkI5XcFOODjv8ASvJdG+EPi7Qtc1DWbP4ixSanqJ/0i7udBSWQj+6C0vyr/sjA4HoKAPSPD+j2vhXwpp+kwSAW2m2qQ+a5xkKvLn0zgk/WvPvC0rfEj4sP41hQjw9oMElhpErDH2uZuJZl/wBnGV9+O4IGh4u+HXinxh4LtdBvvHfktvkN/cQ6WE+2KT8iFVkG1QMgjJ3cZxirXh7wZ4y0KSwgbxxaS6VZlVNhDoEUAaMfwBhIdvHfFAHnkvjDSdb8ZeIfEfir+0Lb7BbT6doNk+l3LhEKkPcFljKhnPHJGB17Gux+AGt2OofCfStPtXla50+HbcK8EiKpeSQrtZlCv0P3Scd8V6NqFp9v0y6s9/l/aIXi34zt3KRnHfrWR4G8L/8ACF+CdN8PfbPtv2GNk+0eV5e/LFs7cnHX1NAG/XF/Ez/kH+Hf+xk03/0etdpXF/Ez/kH+Hf8AsZNN/wDR60AdpRRRQAUUUUAFFFFAHl/jv938evhnLcf8e5/tFF3cgSGAY/Ekr+XtSeO/3vx6+Gcdv/r4/wC0XkI6iPyV6n04YV13jTwbb+MtMtoXu59PvbG5S7sb63xvt5V6HB4I7EHrVLw94En0/wAWTeJ/EWtya5rDWwtIZDbrBFbRZyQkYJwSeSSaAOwri/hd/wAi5qn/AGHtS/8ASqSu0ri/hd/yLmqf9h7Uv/SqSgDtKKKKACvLfiNqVjpHxn+Gl9qt5b2NpF/avmXFzKscaZt0AyzEAZJA+pr1Ks/VvD+ja/HEmu6RY6mkJJjW8tkmCE9SAwOKAMv/AIWP4I/6HLw//wCDSD/4qj/hY/gj/ocvD/8A4NIP/iqP+FceCP8AoTfD/wD4K4P/AImj/hXHgj/oTfD/AP4K4P8A4mgA/wCFj+CP+hy8P/8Ag0g/+Ko/4WP4I/6HLw//AODSD/4qj/hXHgj/AKE3w/8A+CuD/wCJo/4Vx4I/6E3w/wD+CuD/AOJoAP8AhY/gj/ocvD//AINIP/iqP+Fj+CP+hy8P/wDg0g/+Ko/4Vx4I/wChN8P/APgrg/8AiaP+FceCP+hN8P8A/grg/wDiaAD/AIWP4I/6HLw//wCDSD/4qj/hY/gj/ocvD/8A4NIP/iqP+FceCP8AoTfD/wD4K4P/AImj/hXHgj/oTfD/AP4K4P8A4mgA/wCFj+CP+hy8P/8Ag0g/+Ko/4WP4I/6HLw//AODSD/4qj/hXHgj/AKE3w/8A+CuD/wCJo/4Vx4I/6E3w/wD+CuD/AOJoAP8AhY/gj/ocvD//AINIP/iqP+Fj+CP+hy8P/wDg0g/+Ko/4Vx4I/wChN8P/APgrg/8AiaP+FceCP+hN8P8A/grg/wDiaAD/AIWP4I/6HLw//wCDSD/4qj/hY/gj/ocvD/8A4NIP/iqP+FceCP8AoTfD/wD4K4P/AImj/hXHgj/oTfD/AP4K4P8A4mgA/wCFj+CP+hy8P/8Ag0g/+Ko/4WP4I/6HLw//AODSD/4qj/hXHgj/AKE3w/8A+CuD/wCJo/4Vx4I/6E3w/wD+CuD/AOJoA4T4n+LPDmvXXgW10PX9L1K4Txhp8jRWd7HM6qC4LEKScZIGfcV6/WDaeBPCOn3kV3YeFtFtbmFg8U0OnRI6MO4YLkH6VvUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHF6n/yWzw9/2Bb/AP8ARtvXaVyfibwnq+q+J9O1zQdfh0m5sbWa2Im0/wC0iRZGRj/y0XGPLHrVf/hH/iB/0Pen/wDhPj/4/QB2lFcX/wAI/wDED/oe9P8A/CfH/wAfo/4R/wCIH/Q96f8A+E+P/j9AHaUVxf8Awj/xA/6HvT//AAnx/wDH6P8AhH/iB/0Pen/+E+P/AI/QB2lFcX/wj/xA/wCh70//AMJ8f/H6P+Ef+IH/AEPen/8AhPj/AOP0AdpXF/Ez/kH+Hf8AsZNN/wDR60f8I/8AED/oe9P/APCfH/x+s/WPAnjXXIrSO+8dWJW0vIbyLZoIH7yJgy5/f9MjpQB6JRXF/wDCP/ED/oe9P/8ACfH/AMfo/wCEf+IH/Q96f/4T4/8Aj9AHaUVxf/CP/ED/AKHvT/8Awnx/8fo/4R/4gf8AQ96f/wCE+P8A4/QB2lFcX/wj/wAQP+h70/8A8J8f/H6P+Ef+IH/Q96f/AOE+P/j9AHaUVxf/AAj/AMQP+h70/wD8J8f/AB+j/hH/AIgf9D3p/wD4T4/+P0AdpXF/C7/kXNU/7D2pf+lUlH/CP/ED/oe9P/8ACfH/AMfrU8F+GrjwroMlje6iuozzXk93JcJb+SGaWQuRt3Njlj3oA6CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA//Z)

　[プログラム]

整数型: K, T

K ← 1

T ← 1

while (k が購入行数以下でかつ T が対象行数以下)

if (購入[K].品番と対象[T].品番が等しい)

対象[T].数量 ← 購入[K].数量

( a )

( b )

elseif (購入[K].品番が対象[T].品番より小さい)

( a )

else

a と b解答群

ア　K ← K + 1　a

イ　K ← 購入[K].品番 + 1

ウ　T ← T + 1 b

エ　T ← 対象[T].品番 + 1

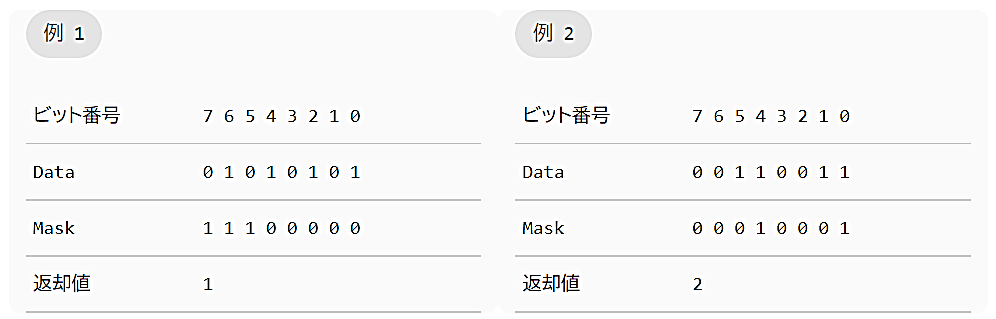
( b )

endif

endwhile

　問８：次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。

　　　関数 BitTest は、 8 ビットのデータ中の指定したビット位置の値を検査し、その結果を返す。 検査される 8 ビットのデータは引数 Data に、検査するビット位置は引数 Mask に、それぞれ格納される。 Mask 中のビットの値が 1 であるビット位置に対応した Data 中のビットを検査し、返却値として、検査した全てのビットが 0 なら 0 を、検査したビット中に 0 と 1 が混在しているなら 1 を、検査した全てのビットが 1 なら 2 を、それぞれ返す。



　　　プログラム中のabに入る正しい答えを、解答群の中から選べ。 なお、本問において、演算子 "&" 、 "|" は、AND および OR のビット演算を求めるもとのし、 "～"B という表記は、 8 ビット論理型定数（ 8 ビットの 2 進数）を表す。 Mask 中には、 1 のビットが 1 個以上あるものとする。

［プログラム］

◯整数型: BitTest( 8 ビット論理型: Data, 8 ビット論理型: Mask)

整数型: RC /\* 返却値 \*/

if ( a )

RC ← 2 /\* 返却値は 2 \*/

elseif ( b )

RC ← 0 /\* 返却値は 0 \*/

else

RC ← 1 /\* 返却値は 1 \*/

endif

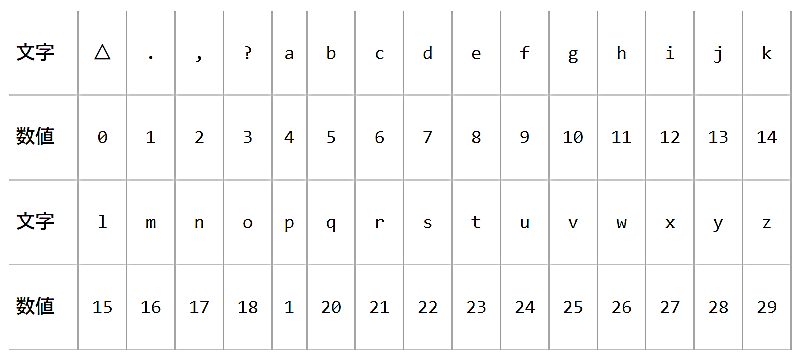
return RC /\* RC を返却値として返す \*/

解答群

|  |  |  |
| --- | --- | --- |
|  | a | b |
| ア | (Data & Mask) = Mask | (Data & Mask) = “00000000”B |
| イ | (Data & Mask) = Mask | (Data | Mask) = “00000000”B |
| ウ | (Data | Mask) = Mask | (Data & Mask) = “00000000”B |
| エ | (Data | Mask) = Mask | (Data | Mask) = “00000000”B |

　問９：次の記述中( a 1 )・( a 2 )・( b )に入れる正しい答えを解答群の中から選べ。

　　関数 calcCheckCharacter は、引数 input[] で指定された文字列の誤りを検出するための検査文字を返す。文字列は、以下に示した N 種類 (ここでは N = 30 ）の文字から構成され、それぞれに数値が割り当てられている。空白文字は “△” と表記する。文字列の長さは、引数 len( 1 以上とする）で指定される。



**19**

　　［検査文字の生成の手順］

1. 文字列の末尾の文字を 1 番目の文字とし、文字列の先頭に向かって奇数番目の文字に割り当てた数値を 2 倍して N で割り、商と余りの和を求め、全て足し合わせる。
2. 偶数番目の文字に割り当てた数値は、そのまま全て足し合わせる。
3. ① と ②の結果を足し合わせる。
4. N から、 ③ で求めた総和を N で割った余りを引く。さらにその結果を、 N で割り、余りを求める。求めた数値に対応する文字を検査文字とする。

［検査文字の生成例］

文字列 “ipa△△” に対し、生成される検査文字は “f” である。

　プログラム中の　　　　　　 に入れる正しい答えを、解答群の中から選べ。ここで、 a1 と a2 に入れる答えは、 a に関する解答群の中から組合せとして正しいものを選ぶものとする。関数 caclCheckCharacter の中で使われている関数 getValue は、引数に与えられた文字に割り当てられた数値を返し、関数 getChar は、引数に与えられた数値に対応する文字を返す。配列の添字は、 1 から始まるとする。

［プログラム］

◯文字型: calcCheckCharacter(文字型の配列: input, 整数型: len)

整数型: N, sum, i, value, check\_value

論理型: is\_even

N ← 30

sum ← 0

is\_even ← a1

for (i を len から 0 より大きい間 -1 ずつ減らす )

value ← getValue(input[i])

if ( is\_even が a2 と一致)

sum ← sum + value

else

sum ← sum + (value × 2) ÷ N + (value × 2) mod N

endif

is\_even ← not is\_even

endfor

check\_value ← b

return getChar(check\_value)

( b )解答群

|  |  |
| --- | --- |
| ア | N - sum mod N |
| イ | sum mod N |
| ウ | ( N - sum mod N ) mod N |
| エ | ( sum - N ) mod Ne |

( a1 )( a2 )解答群

|  |  |  |
| --- | --- | --- |
|  | a1 | a2 |
| ア | false | false |
| イ | false | true |
| ウ | true | false |
| エ | true | true |

問10：次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。

　　手続 makeHeap は、配列 data に格納されている hnum 個（ hnum > 0 ）のデータを、次の (1) ～ (3) の規則で配列 heap に格納して、ヒープを配列で実現する。 配列の添字は、 0 から始まる。

(1)：heap[i] ( i = 0, 1, 2, ・・・) は、ヒープの節に対応する。

(2)：heap[0] は、根に対応する。

(3)：heap[i] ( i = 0, 1, 2, ・・・) に対応する節の左側の子は heap[2 × i + 1] に対応し、右側の子は

heap[2 × i + 2] に対応する。 子が一つの場合、左側の子として扱う。

ヒープの例を図 1 に、図 1 に対応する配列 heap の例を図 2 に示す。

図 1　ヒープの例

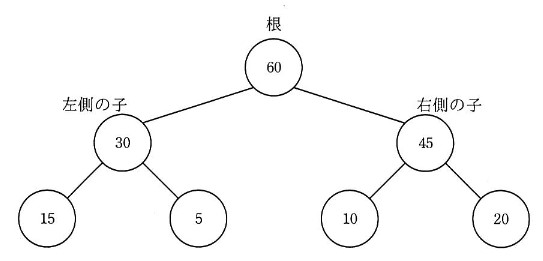
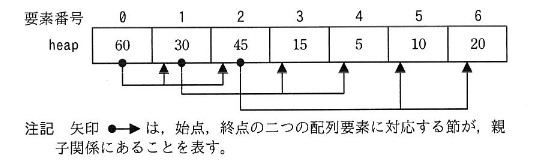


図 2　図 1 のヒープの例に対応した配列 heap の内容



　［プログラム］

◯makeHeap(整数型の配列: data, 整数型の配列: heap, 整数型: hnum)

整数型: i, k

for (i が 0 から hnum より小さい間 1 ずつ増やす)

heap[i] ← data[i] /\* heap にデータを追加 \*/

k ← i

while ( k が 0 より大きい)

if ( a )

swap(heap, k, b )

k ← parent(k)

elseif

break /\* 内側の繰り返し処理から抜ける \*/

endif

endwhile

endfor

　[関数の仕様]

* プログラムおよび解答群の中で使われている関数 lchild( 整数型: k ) は、添字 k の配列要素に対応する節の左側の子の添字 2 × k + 1 を戻り値として返す。
* 関数 rchild( 整数型: k ) は、添字 k の配列要素に対応する節の右側の子の添字 2 × k + 2 を戻り値として返す。
* 関数 parent( 整数型: k ) 関数は、添字 k の配列要素に対応する親の添字 ( k – 1 ) ÷ 2 （小数点以下切捨て）を戻り値として返す。
* 関数 swap( 整数型: heap[], 整数型: i, 整数型: j ) は、 heap[i] と heap[j] の値を交換する。

a に関する解答群

ア　heap[k] が heap[lchild(k)] より大きい

イ　heap[k] が heap[parent(k)] より大きい

ウ　heap[k] が heap[rchild(k)] より大きい

エ　heap[k] が heap[lchild(k)] より小さい

オ　heap[k] が heap[parent(k)] より小さい

カ　heap[k] が heap[rchild(k)] より小さい

b に関する解答群

ア　heap[hnum - 1]

イ　heap[k]

ウ　parent(hnum - 1)

エ　parent(k)

問11　次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。

　　関数 MUL は、二つの整数 M 、 N を受け取り、その積 M × N を返す。 積は、加算とシフト演算を使って求める。 M 、 N 及び求めた積は、いずれも符号付き 2 進数の整数で、負数は 2 の補数で表現する。

　　関数 MUL が受け取る M 、 N は、いずれも 4 ビットで、各値の範囲は -8 ～ 7 である。 求めた積 M × N は、 8 ビットで返す。 ただし、プログラム中では、 M を 5 ビットに、 N を 8 ビットに拡張して処理を行う。

　R は、 13 ビットの作業用変数であり、最下位から順にビット番号を 0, 1, ・・・とし、最上位（符号ビット）のビット番号を 12 とする。 R は、指定した一部の範囲（例えば、ビット番号 12 ～ 8 の上位 5 ビット）だけを符号付き 2 進数とみなして部分的な算術演算ができる。 また、値の検査のために指定したビット番号の内容を取り出すこともできる。

　M = -5 、N = 3 の場合の処理過程を、図に示す。 図中の記号 ① ～ ⑧ は、プログラムの ① ～ ⑧ の行の処理と対応している。 なお、 ⑥ の行の加算では、けたあふれが起きても無視する。

［プログラム］

◯整数型: MUL(M, N)

　　符号付き 2 進整数型: M, N, R

　　整数型: L

1. M を 5 ビットの符号付き 2 進数に拡張
2. N を 8 ビットの符号付き 2 進数に拡張
3. R のビット番号 7 ～ 0 に N を複写
4. R のビット番号 12 ～ 8 を 0 で初期化

　 for (L を 1 から 8 以下まで 1 つずつ増やす)

1. if (R のビット番号 0 のビットが 1)
2. R のビット番号 12 ～ 8 の内容に M の値を加算

endif

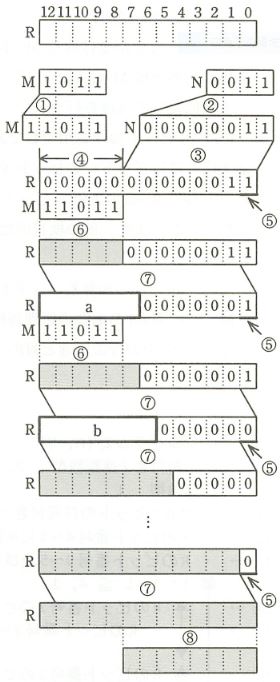
1. R の全 13 ビットを右に 1 ビット算術シフト /\* 空いたビット位置には符号と同じものが入る \*/

endfor

1. return (R のビット番号 7 ～ 0 の内容) /\* 返却値（括弧内）を返す \*/

図　プログラムの実行例（ M = -5, N = 3 ）

注　網掛けの部分は、表示していない。



**ここ**

**ここ**

a に関する解答群  
ア　　イ　  
ウ　　エ　

b に関する解答群  
ア　　イ　  
ウ　　エ　  
オ　　カ　

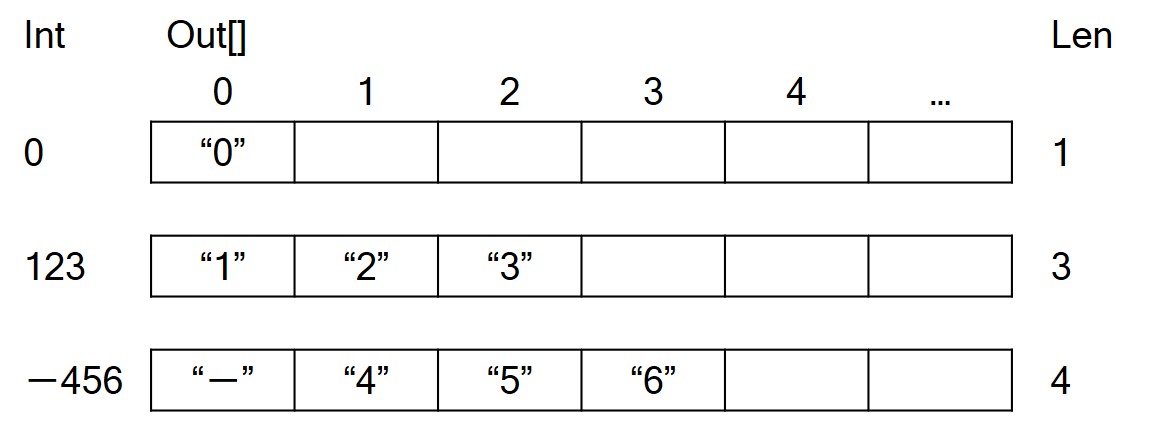
問12　次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。　※トレース基礎　スタック

　　スタックを使って整数値を 10 進数字列（文字列）に変換する IntFormat 関数である。 IntFormat 関数は、引数 Int で指定された整数値を 10 進数字列に変換し、その先頭の数字（文字）から 1 文字ずつ順に引数 Out[] に格納し、格納した文字数を引数 Len に格納する。整数値を 10 進数字列に変換する手順は、次のとおりである。

1. Int の値がゼロの場合は、 Out[0] に “0” を、 Len に 1 を格納して、関数を終了する。
2. Int の値が負の場合は、負符号を表す “-” を Out[] に格納し、 Int の値を正数に変換する。
3. Int の 1 の位から上位に向かって、 1 けたずつ 10 進数字に変換し、 Push 関数でスタックに積む。
4. スタックに積み終わったら、スタックに積んだ文字を順番に Pop 関数で取り出して Out[] に格納することによって、変換後の 10 進数字を正しい順序に並び替える。
5. Len に Out[] に格納した文字数を設定して、関数を終了する。

　引数の値をスタックに積む Push 関数と、スタックから取り出した値を戻り値として返す Pop 関数は、あらかじめ用意されているとする。配列の添字は 0 から始まり、 Out[] の要素数は十分に大きいものとする。プログラム中の各演算で、あふれは生じないものとする。 IntFormat 関数の変換例を下図に示す。

図　IntFormat 関数の変換例



　　[プログラム]

◯IntFormat(整数型: Int, 文字型の配列: Out, 整数型: Len)

整数型: L, I, Idx

文字型の配列: Chr ← {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"}

文字型: T

L ← 0

/\* 引数 Int がゼロの場合の処理 \*/

if (Int が 0 )

Out[L] ← Chr[Int]

L ← L + 1

Len ← L

return

endif

/\* 符号の処理 \*/

if (Int が 0 より小さい)

I ← -Int

Out[L] ← "-"

L ← L + 1

else

I ← Int

endif

/\* 数値を 1 けたずつ数字に変換してスタックに格納する処理 \*/

Push("#") /\* スタックに番兵として "#" を積む \*/

while (I が 0 より大きい)

Idx ← I - (I ÷ 10) × 10

Push(Chr[Idx])

a に関する解答群

ア　I – 10

イ　I – I ÷ 10

ウ　I – ( I ÷ 10 ) × 10

エ　I ÷ 10

I ← a

endwhile

/\* スタックから取り出す処理 \*/

T ← Pop()

ｂ に関する解答群

ア　L ← L + 1

イ　L ← L – 1

ウ　I ← I + 1

エ　I ← I – 1

while (T が "#" でない)

Out[L] ← T

b

T ← Pop()

endwhile

Len ← L

return

問13　次の記述中( a )・( b )に入れる正しい答えを解答群の中から選べ。　※トレース基礎　挿入ソート

　　手続 InsertSort は、引数で与えられた配列 A[] を、挿入ソートで昇順に整列する。配列 A[] の添字は 0 から始まり、要素数は引数 N で与えられる。挿入ソートの手順は、次のとおりである。

1. まず、 A[0] と A[1] を整列し、次に A[0] から A[2] までを整列し、その次に A[0] から A[3] までというように、整列する区間の要素を一つずつ増やしていき、最終的A[0] から A[N – 1] までを整列する。
2. 整列する区間が A[0] から A[M]（ 1 ≦ M < N ）までのとき、 A[M] を既に整列された列 A[0] , more\_horiz, A[M – 1] 中の適切な位置に挿入する。その手順は次のとおりである。
3. A[M] の値を、作業領域 Tmp に格納する。
4. A[M – 1] から A[0] に向かって Tmp と比較し、 Tmp よりも大きな値を順次隣（要素番号の大きい方）に移動する。
5. で最後に移動した値の入っていた配列要素に Tmp の値を格納する。 b. で移動がなかった場合には A[M] に格納する。

　　[プログラム]

　　　◯InsertSort(整数型の配列: A, 整数型: N)

整数型: Idx1, Idx2, Tmp

論理型: Loop

for (Idx1 を 1 から N より小さい間 1 つずつ増やす)

Tmp ← A[Idx1]

Idx2 ← a

Loop ← true

while (Idx2 が 0 以上でかつ Loop が true)

if (A[Idx2] が Tmp より大きい)

b

Idx2 ← Idx2 - 1

a に関する解答群

ア　Idx1

イ　Idx1 + 1

ウ　Idx1 – 1

エ　1 – Idx1

else

Loop ← false

endif

endwhile

A[Idx2 + 1] ← Tmp

a に関する解答群

ア　I – 10

イ　I – I ÷ 10

ウ　I – ( I ÷ 10 ) × 10

エ　I ÷ 10

ｂ に関する解答群

ア　A[Idx2] ← A[Idx2 + 1]

イ　A[Idx2] ← A[Idx2 – 1]

ウ　A[Idx2 + 1] ← A[Idx2]

エ　A[Idx2 – 1] ← A[Idx2]

endfor

問14　次の記述中( a )・( b )・( c )に入れる正しい答えを解答群の中から選べ。　※プログラムの分割

　手続 TabSpc は、タブ文字を展開するプログラムである。

1. TabSpc は、引数で指定された文字型配列 Src[] を先頭から調べ、 Src[] 中のすべてのタブ文字をそれぞれ一つ以上の間隔文字（スペース）に変換して、引数で指定された文字型配列 Dst[] に格納する。
2. 文字型配列の各要素には、文字を 1 文字ずつ順に格納し、最後の文字の次の要素にはシステム定数である EOS を格納する。 なお、配列の添字は 1 から始まり、添字の値を文字位置と呼ぶ。
3. Src[] 中にタブ文字が出現した場合、次の文字が最も近い右のタブ位置に格納されるように、タブ文字を一つ以上の間隔文字に置換して、 Dst[] （要素数は十分に大きいとする）に格納する。 ここで、タブ位置とは、整数型の引数 TabGap で渡されるタブ間隔（≧ 2）を用いて、次の式で計算される文字位置である。

タブ位置 = タブ間隔 × n + 1 （ n は 1 以上の整数）

1. タブ間隔が 4 のときの実行例を図に示す。 “j” を Dst[] のタブ位置である文字位置 13 （ = 4 × 3 + 1 ）に格納したのでは、タブ文字が間隔文字に置き換わらないので、最も近い右のタブ位置である文字位置 17 （ = 4 × 4 + 1 ）に格納する。

図　タブ間隔が 4 のときの TabSpc の実行例

グラフィカル ユーザー インターフェイス, グラフ, アプリケーション

自動的に生成された説明

　[プログラム]

　　　◯TabSpc(文字型の配列: Src, 文字型の配列: Dst, 整数型: TabGap)

整数型: Sidx, Didx, N, TabPos

Sidx ← 1

Didx ← 1

while (Src[Sidx] が EOS とは等しくない) /\* EOS: 文字列の終わりを表すシステム定数 \*/

if (Src[Sidx] が TAB と等しい) /\* TAB: タブ文字を表すシステム定数 \*/

N ← ( a ) ÷ TabGap

TabPos ← TabGap × N + 1

while (Didx が TabPos より小さい)

Dst[Didx] ← SPC /\* SPC: 間隔文字を表すシステム定数 \*/

b

endwhile

else

c

Didx ← Didx + 1

endif

Sidx ← Sidx + 1

endwhile

Dst[Didx] ← EOS

a に関する解答群

ア　Didx + 1

イ　Didx – 1

ウ　Didx + TabGap + 1

エ　Didx + TabGap – 1

b 、 c に関する解答群

ア　Didx ← Didx + 1 　ｂ

イ　Dst[Didx] ← Src[Sidx]　ｃ

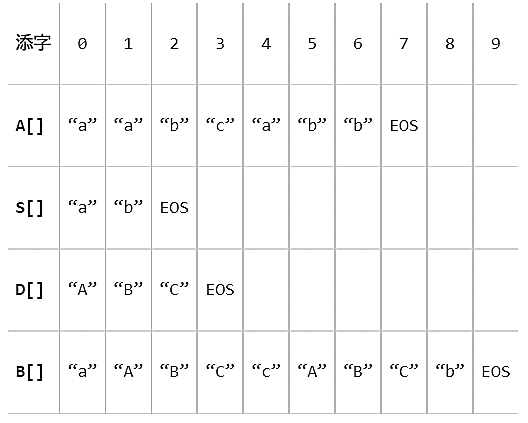
ウ　Dst[Didx + 1] ← Src[Sidx]

エ　Dst[Didx] ← Src[Sidx + 1]

問15　次の記述中( a )・( b )・( c ) ・( d )に入れる正しい答えを解答群の中から選べ。　※文字列置換

　　手続 Replace は、文字列を置換するプログラムである。 Replace には、文字型の配列の引数 A[], S[], D[], B[] があり、 A[] を先頭からコピーして B[] に格納する（ B[] は十分に大きいとする）。 このとき、 A[] 中に S[] と一致する文字列を見つけるたびに、それを D[] の内容に置き換える。 このプログラムで扱う文字型の配列の添字は 0 から始まり、配列の末尾には定数 EOS が格納されている。

図　Replace の実行例



［プログラム］

◯Replace(文字型の配列: A, 文字型の配列: S, 文字型の配列: D, 文字型の配列: B)

整数型: Aidx, Sidx, Didx, Bidx, Idx

Aidx ← 0

Bidx ← 0

while (A[Aidx] が EOS と等しくない)

if (A[Aidx] と S[0] が等しい)

Idx ← Aidx

a

do

Sidx ←　Sidx + 1

Aidx ←　Aidx + 1

while (bと A[Aidx] がEOS と等しくない)

if (c)

Didx ← 0

a, d に関する解答群

ア B[Bidx] ← A[Aidx]

イ B[Bidx] ← S[0]　ｄ

ウ Idx ← 0

エ Idx ← 1

オ Sidx ← 0　a

カ Sidx ← 1

while (D[Didx] が EOS と等しくない)

B[Bidx] ← D[Didx]

Didx ← Didx + 1

Bidx ← Bidx + 1

endwhile

else

d

Aidx ← Idx + 1

Bidx ← Bidx + 1

b, c に関する解答群

ア A[Aidx] と EOS が等しくない

イ A[Aidx] と S[Sidx] が等しい　ｂ

ウ A[Aidx] と S[Sidx] が等しくない

エ A[Aidx] と S[Sidx] が等しくなく、

かつ S[Sidx] と EOS が等しくない

オ S[Sidx] と EOS が等しい　ｃ

カ S[Sidx] と EOS が等しくない

endif

else

B[Bidx] ← A[Aidx]

Aidx ← Aidx + 1

Bidx ← Bidx + 1

endif

endwhile

B[Bidx] ← EOS

問16　次の記述中( a )・( b )・( c ) ・( d )に入れる正しい答えを解答群の中から選べ。　※基数変換

　　RadixConv 関数は、 M 進数字列（ 2 ≦ M ≦ 16 ）を、 N 進数字列（ 2 ≦ N ≦ 16 ）に基数変換するプログラムである。

1. M 進数字列および N 進数字列は、数字を表す文字だけで構成され、空白文字は含まない。 11 ～ 16 進数の 10 以上の数字には、 “A” ～ “F” のアルファベットを用いる。
2. RadixConv 関数は、 M 進数字列を整数値に変換した後、その整数値を N 進数字列に変換する。その際に、 M 進数字列を整数値に変換する MToInt 関数と、整数値を N 進数字列に変換する IntToN 関数を使う。
3. MToInt 関数と IntToN 関数は、数字を整数に変換して返す ToInt 関数、整数値を数字に変換する ToStr 関数、文字列の長さを返す Length 関数、および文字列の一部を切り出して返す SubStr 関数を使う。
4. このプログラムで使われる関数の引数と戻り値の仕様を表 1 ～ 7 に示す。

表 1　RadixConv 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Frdx | 整数型 | 変換元の数字列の基数（ 2 ≦ Frdx ≦ 16 ） |
| Fnum | 文字列 | 変換元の数字列 |
| Trdx | 整数型 | 変換後の数字列の基数（ 2 ≦ Trdx ≦ 16 ） |
| 戻り値 | 文字列型 | 変換後の数字列 |

表 2　MToInt 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Rdx | 整数型 | 変換元の数字列の基数（ 2 ≦ Rdx ≦ 16 ） |
| Num | 文字列型 | 変換元の数字列 |
| 戻り値 | 整数型 | 変換後の整数 |

表 3　IntToN 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Val | 整数型 | 変換元の整数 |
| Rdx | 整数型 | 変換後の数字列の基数（ 2 ≦ Rdx ≦ 16 ） |
| 戻り値 | 文字列型 | 変換後の数字列 |

表 4　ToInt 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| P | 文字型 | 変換元の数字（ “0”, “1”, …, “F” ） |
| 戻り値 | 整数型 | 変換後の整数 |

表 5　ToStr 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| Q | 整数型 | 変換元の整数（ 0 ≦ Q ≦ 15 ） |
| 戻り値 | 文字型 | 変換後の数字 |

表 6　Length 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| S | 文字列型 | 長さを求める文字列 |
| 戻り値 | 整数型 | 文字列の長さ |

表 7　SubStr 関数

|  |  |  |
| --- | --- | --- |
| **引数／戻り値** | **データ型** | **意味** |
| S | 文字列型 | 切り出し元の文字列 |
| Idx | 整数型 | 切り出しを始める位置（先頭を 1 とする） |
| N | 整数型 | 切り出す文字数 |
| 戻り値 | 文字列型 | 文字列型S の先頭から Idx の位置から N 文字を返す |

　[プログラム]

　　◯文字列型: RadixConv(整数型: Frdx, 文字列型: Fnum, 整数型: Trdx)

return IntToN(MToInt(Frdx, Fnum), Trdx)

◯整数型: MToInt(整数型: Rdx, 文字列型: Num)

整数型: Idx, Val

Val ← 0

for (Idx を 1 から Length(Num) 以下まで 1 つずつ増加)

Val ← a + ToInt(Substr(Num, Idx, 1))

endfor

a に関する解答群

ア　Rdx

イ　Val

ウ　Val × Rdx

エ　Val ÷ Rdx

return Val

◯文字列型: IntToN(整数型: Val, 整数型: Rdx)

整数型: Quo /\* 商 \*/

整数型: Rem /\* 剰余 \*/

文字列型: Num

ｂ～ｄ に関する解答群

ア　Quo ← Quo ÷ Rdx　ｃ

イ　Quo ← Quo ÷ Rem

ウ　Quo ← Rdx

エ　Quo ← Rem ÷ Rdx

オ　Quo ← Val　ｂ

カ　Rem ← Rdx

キ　Rem ← Val

ク　Num ← ToStr(Quo) + Num　ｄ

ケ　Num ← ToStr(Rem) + Num

b

Num ← ""

while (Quo が Rdx 以上)

Rem ← Quo mod Rdx

　　/\* + は文字列を連結する演算子 \*/

Num ← ToStr(Rem) + Num

c

endwhile

d

return Num

問17　次のプログラム中の　　　　　　に入れる正しい答えを、解答群の中から選べ　T

　　関数ToInt(文字型：P)は、引数Pで指定された1文字の数字(“０”, “１”, ・・・“F”のいずれか)を数値に変換する。”０”～”９”の数字は、０～９の数値に変換し、”A”～”F”の数字は、１０～１５の数値に変換する。

　　配列の添え字は０から始まる。

　[プログラム]

　　〇整数型：ToInt（文字型：P）

　　　整数型：Idx

　　　文字型の配列：Code　←　｛“０”, “１”, “２”, “３”, “４”, “５”, “６”, “７”, “８”, “９”,

“A”, “B”, “C”, “D”, “E”, “F”｝

　　　Idx　←　０

　　　while( ) /\* 文字の比較 \*/

Idx　←　Idx　＋　１

　　　endwhile

　　　return　Idx /\* Idxを関数の返却値とする \*/

解答群

ア　P　＞　Code[ Idx ]

イ　P　＜　Code[ Idx ]

ウ　P　≧　Code[ Idx ]

エ　P　≦　Code[ Idx ]

問18　次のプログラム中の　　　　　　に入れる正しい答えを、解答群の中から選べ　T

　　関数MatchCounter(文字型の配列：SorceText, 整数型：Textlen, 文字型の配列：Pattern, 整数型：Patlen)は、長さがTextlenの文字列SorceTextの中で、長さがPatlenの文字列Patternと一致する部分の文字列(以下、部分文字列という)の出現回数を数える。ここで、０＜Patlen≦Textlenである。

1. MatchCounterの処理手順は、次のとおりである。
   1. 一致する部分の出現回数を数える変数Counterの値を０に初期化する
   2. SourceTextの比較開始位置を先頭から順に1文字ずつ後ろにずらしながら、その比較開始位置から始まる長さPatlenの文字列とPatternが一致するかどうかを調べ、一致したら出現回数Counterの値に1を加算する。
   3. Counterの値を返す
2. 文字列は、文字型の配列の各要素に1文字ずつ格納されている。また、各配列の添え字は０から始まる。

　[プログラム]

　　〇整数型：MatchCounter(文字型の配列：SorceText, 整数型：Textlen,

文字型の配列：Pattern, 整数型：Patlen)

　　　整数型：Counter,i ,j ,k

　　　論理型：Matchflg

　　　Counter ← 0

　　　i ← 0

　　　while( i + Patlen ≦ Textlen )

　　　　　j ← i

　　　　　k ← 0

　　　　　Matchflg ← true

　　　　　/\* Patternと一致する部分があるかチェック \*/

　　　　　while( ( k ＜ Patlen ) and ( Matchflg ＝ true ) )

　　　　　　　if( 　　　　　 )

　　　　　　　　　j ← j ＋ １

　　　　　　　　　k ← k ＋ １

　　　　　　　else

　　　　　　　　　Matcflg ← false

　　　　　　　endif

　　　　　endwhile

　　　　　/\* Patternと一致する部分があれば出現回数に1を加える \*/

　　　　　if( k ＝ Patlen )

解答群

ア　SourceText[i] ＝ Pattern[k]

イ　SourceText[k] ＝ Pattern[i]

ウ　SourceText[j] ＝ Pattern[k]

エ　SourceText[k] ＝ Pattern[j]

　　　　　　　Counter ← Counter ＋ 1

　　　　　endif

　　　　　i ← i ＋ １

　　　endwhile

　　　/\* 出現回数を返す \*/

　　　return Counter

問19　次のプログラム中の　　　　　　に入れる正しい答えを、解答群の中から選べ　T

　　手続きMakeList(文字列型：Mvalue)は、リストを作成する。リストは、構造体の配列で表現する。構造体の定義は、次のとおりである。ここで、Listは大域変数であり、配列の添え字は０から始まる。添え字がnの要素List[n]のNextとValueは、それぞれList[n].Next 、List[n].Valueで表す。

　構造体：List[100]｛

　　　整数型：Next ＝ －1 /\* 次の要素の添え字、初期値は－1 \*/

　　　文字列型：Value　　 /\* 要素の値 \*/

　｝

　　手続きMakeList(文字列型：Mvalue)は、1回の呼出しで、文字列型の引数Mvalueを値とする要素を作成し、リストの最後に追加する。整数型の大域変数Listsizeには、リストの要素数を格納する。リストの要素は、

　List[0]から、List[Listsize – 1]に格納する。リストに登録する要素の個数は、配列Listの大きさ以下とする。また、リストの先頭の要素が格納されている配列Listの添え字は、整数型の大域変数Firstに格納する。

　　Firstの初期値は－1とする。

　[プログラム]

　　〇大域：整数型：Listsize ＝ ０ /\* リストの要素数 \*/

　　〇大域：整数型：First = －1　　 /\* リストの先頭要素の添え字 \*/

　　〇大域：構造体型：List[100]｛

整数型：Next ＝ －1 /\* 次の要素の添え字、初期値は－1 \*/

　　　 文字列型：Value　　 /\* 要素の値、可変長文字列 \*/

　 　　｝

　　/\* 手続MakeList \*/

　　〇MakeList(文字列型：Mvalue)

　　　if( Listsize ≠ ０ )

　　　　　List[Listsize － １].Next ←

　　　else

　　　　　First ← ０

　　　endif

解答群

ア　First

イ　First ＋ １

ウ　First － １

エ　Listsize

オ　Listsize ＋ １

カ　ListSize － 1

　　　List[Listsize].Value ← Mvalue

　　　Listsize ← Listsize ＋ 1

　　　/\* 手続MakeList　の終わり \*/

問20　次のプログラム中の　　　　　　に入れる正しい答えを、解答群の中から選べ　T

　　 ここで、配列の添え字は0から始まる。

　　手続きconvert(整数型：num, 文字型の配列：str)は、金額を表す時のように、整数を３桁区切り形式の文字

列に変換する。

1. 次のルールに基づいて変換を行う。
   1. 整数値が負の場合、先頭にマイナス符号をつける。
   2. 整数値の下から3桁ごとにコンマを挿入する。

変換例を表に示す。

　●表　変換例

|  |  |
| --- | --- |
| 整数 | 3桁区切り形式の文字列 |
| 1234567 | 1,234,567 |
| －57482 | －57,482 |
| 63 | 63 |
| －999999 | －999,999 |

1. 手続きconvert(整数型：num, 文字型の配列：str)の仕様は、次のとおりである。

引数：整数num、変換後の文字列str

返却値：なし

機能：整数numを3桁区切り形式の文字列に変換してstrに先頭から格納する。

　　　strの末尾に、システム定数EOSを格納する。

　　　strには変換後の文字列を格納するのに十分な領域が確保されている。

　　　整数numは9桁以下とする。

　[プログラム]

〇convert(整数型：num, 文字型の配列：str)

　　論理型：minus

　　整数型：i , j

　　文字型の配列：table

　　文字型：tmp

　　minus ← false

　　i ← 0

　　j ← 0

　　table ← “01234567”

解答群

ア　i mod 3

イ　( i + 1 ) mod 3

ウ　( i + 2 ) mod 3

エ　( i － 1 ) mod 3

　　/\* 整数値が負かどうかをチェックする \*/

　　if( num < 0 )

　　　　minus ← true

num ← －num

endif

/\* 数値の下位から順に文字に変換する \*/

　　do

/\* 1桁の文字を格納する \*/

　　　　str[j] ← table[num mod 10]

j ← j + 1

/\* 次のけたを取り出す準備をする \*/

　　　　num ← num ÷ 10

/\* 格納した文字の数をカウントアップする \*/

　　　　i ← i + 1

/\* 3桁ごとに区切りのカンマを挿入する \*/

　　　　if( 　　　　　 = 0 and num ≠ 0 )

str[j] ← “ , ”

j ← j + 1

endif

while( num ≠ 0 )

　　/\* 整数値が負ならマイナス符号を付ける \*/

　　if( minus = true )

str[j] ← “－”

j ← j + 1

endif

str[j] ← EOS

/\* strの要素の順序を逆にする \*/

　　i ← 0

j ← j － 1

　　while( i < j )

tmp ← str[i]

str[i] ← str[j]

str[j] ← tmp

i ← i + 1

j ← j － 1

endwhile