

```

{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 1,
      "id": "c8dfebb3",
      "metadata": {
        "execution": {
          "iopub.execute_input": "2022-10-25T06:56:13.470484Z",
          "iopub.status.busy": "2022-10-25T06:56:13.469833Z",
          "iopub.status.idle": "2022-10-25T06:57:22.615465Z",
          "shell.execute_reply": "2022-10-25T06:57:22.611998Z"
        },
        "papermill": {
          "duration": 69.16216,
          "end_time": "2022-10-25T06:57:22.618152",
          "exception": false,
          "start_time": "2022-10-25T06:56:13.455992",
          "status": "completed"
        },
        "tags": []
      },
      "outputs": [
        {
          "name": "stderr",
          "output_type": "stream",
          "text": [
            "[nlTK_data] Error loading stopwords: <urlopen error [Errno -3]\n",
            "[nlTK_data] Temporary failure in name resolution>\n",
            "[nlTK_data] Error loading wordnet: <urlopen error [Errno -3] Temporary\n",
            "[nlTK_data] failure in name resolution>\n"
          ]
        },
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
            "\u001b[2m\u001b[1m\u001b[32m\n",
            "NECESSARY LIBRARIES WERE SUCCESFULLY IMPORTED...\u001b[0m\n"
          ]
        },
        {
          "name": "stderr",
          "output_type": "stream",
          "text": [
            "[nlTK_data] Error loading omw-1.4: <urlopen error [Errno -3] Temporary\n",
            "[nlTK_data] failure in name resolution>\n"
          ]
        }
      ],
      "source": [
        "import numpy as np\n",
        "import pandas as pd\n",
        "import seaborn as sns\n",
        "import matplotlib.pyplot as plt\n",
        "import plotly.express as px\n",
        "import re, string, unicodedata\n",

```



```

"from string import punctuation\n",
"from termcolor import colored\n",
"from collections import Counter\n",
"\n",
"from sklearn.preprocessing import LabelBinarizer\n",
"from sklearn.metrics import classification_report, confusion_matrix, accuracy_score\n",
"from sklearn.model_selection import train_test_split\n",
"from sklearn.preprocessing import LabelEncoder\n",
"\n",
"import keras\n",
"import tensorflow as tf\n",
"from keras.preprocessing import text, sequence\n",
"from keras.models import Sequential\n",
"from keras.layers import Dense, Embedding, LSTM, Dropout\n",
"from keras.callbacks import ReduceLROnPlateau\n",
"from tensorflow.keras.preprocessing.text import Tokenizer\n",
"\n",
"import nltk\n",
"from nltk.corpus import stopwords\n",
"from textblob import Word\n",
"nltk.download('stopwords')\n",
"nltk.download('wordnet')\n",
"nltk.download('omw-1.4')\n",
"from nltk.stem.porter import PorterStemmer\n",
"from wordcloud import WordCloud, STOPWORDS\n",
"from nltk.stem import WordNetLemmatizer\n",
"from nltk.tokenize import word_tokenize, sent_tokenize\n",
"from nltk import pos_tag\n",
"from nltk.corpus import wordnet\n",
"\n",
"from warnings import filterwarnings\n",
"filterwarnings('ignore')\n",
"\n",
"from sklearn import set_config\n",
"set_config(print_changed_only = False)\n",
"\n",
"#to see full text:\n",
"pd.set_option('display.max_colwidth', -1)\n",
"\n",
"print(colored("\n\nNECESSARY LIBRARIES WERE SUCCESFULLY IMPORTED...\n", color =
"green", attrs = ["bold", "dark"]))
]
},
{
"cell_type": "code",
"execution_count": 2,
"id": "99e14759",
"metadata": {
"execution": {
"iopub.execute_input": "2022-10-25T06:57:22.640874Z",
"iopub.status.busy": "2022-10-25T06:57:22.639655Z",
"iopub.status.idle": "2022-10-25T06:57:22.689417Z",
"shell.execute_reply": "2022-10-25T06:57:22.687658Z"
}
},
"papermill": {
"duration": 0.064341,
"end_time": "2022-10-25T06:57:22.692692",

```



```

        "exception": false,
        "start_time": "2022-10-25T06:57:22.628351",
        "status": "completed"
    },
    "tags": []
},
"outputs": [
    {
        "name": "stdout",
        "output_type": "stream",
        "text": [
            "\u001b[2m\u001b[1m\u001b[32m\n",
            "DATASETS WERE SUCCESFULLY LOADED...\u001b[0m\n"
        ]
    }
],
"source": [
    "data = pd.read_csv(\"../input/sms-spam-collection-dataset/spam.csv\", encoding = \"ISO-8859-1\", engine = \"python\")\n",
    "\n",
    "print(colored(\"\\nDATASETS WERE SUCCESFULLY LOADED...\", color = \"green\", attr = [\"bold\", \"dark\"]))"
],
{
    "cell_type": "markdown",
    "id": "feacf8dc",
    "metadata": {
        "papermill": {
            "duration": 0.009765,
            "end_time": "2022-10-25T06:57:22.713703",
            "exception": false,
            "start_time": "2022-10-25T06:57:22.703938",
            "status": "completed"
        }
    },
    "tags": []
},
"source": [
    "### <span style = \"background:#410F01; border:black solid; font-size:100%; color:#fff; border-radius:10px;\">Look at the dataset</span>"
],
{
    "cell_type": "code",
    "execution_count": 3,
    "id": "1b89ccc7",
    "metadata": {
        "execution": {
            "iopub.execute_input": "2022-10-25T06:57:22.737093Z",
            "iopub.status.busy": "2022-10-25T06:57:22.736305Z",
            "iopub.status.idle": "2022-10-25T06:57:22.757665Z",
            "shell.execute_reply": "2022-10-25T06:57:22.756478Z"
        }
    },
    "papermill": {
        "duration": 0.036411,
        "end_time": "2022-10-25T06:57:22.760094",
        "exception": false,

```



```

"start_time": "2022-10-25T06:57:22.723683",
"status": "completed"
},
"tags": []
},
"outputs": [
{
  "data": {
    "text/html": [
      "<div>\n",
      "<style scoped>\n",
      "  .dataframe tbody tr th:only-of-type {\n",
      "    vertical-align: middle;\n",
      "  }\n",
      "\n",
      "  .dataframe tbody tr th {\n",
      "    vertical-align: top;\n",
      "  }\n",
      "\n",
      "  .dataframe thead th {\n",
      "    text-align: right;\n",
      "  }\n",
      "</style>\n",
      "<table border=\\"1\" class=\\"dataframe\">\n",
      "  <thead>\n",
      "    <tr style=\\"text-align: right;\">\n",
      "      <th></th>\n",
      "      <th>v1</th>\n",
      "      <th>v2</th>\n",
      "      <th>Unnamed: 2</th>\n",
      "      <th>Unnamed: 3</th>\n",
      "      <th>Unnamed: 4</th>\n",
      "    </tr>\n",
      "  </thead>\n",
      "  <tbody>\n",
      "    <tr>\n",
      "      <th>0</th>\n",
      "      <td>ham</td>\n",
      "      <td>Go until jurong point, crazy.. Available only in bugis n great world la e
buffet... Cine there got amore wat...</td>\n",
      "      <td>NaN</td>\n",
      "      <td>NaN</td>\n",
      "      <td>NaN</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>1</th>\n",
      "      <td>ham</td>\n",
      "      <td>Ok lar... Joking wif u oni...</td>\n",
      "      <td>NaN</td>\n",
      "      <td>NaN</td>\n",
      "      <td>NaN</td>\n",
      "    </tr>\n",
      "    <tr>\n",
      "      <th>2</th>\n",
      "      <td>spam</td>\n",
      "      <td>Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text
FA to 87121 to receive entry question(std txt rate)T&amp;C's apply

```



```

08452810075over18's</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "  </tr>\n",
    "  <tr>\n",
    "    <th>3</th>\n",
    "    <td>ham</td>\n",
    "    <td>U dun say so early hor... U c already then say...</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "  </tr>\n",
    "  <tr>\n",
    "    <th>4</th>\n",
    "    <td>ham</td>\n",
    "    <td>Nah I don't think he goes to usf, he lives around here though</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "    <td>NaN</td>\n",
    "  </tr>\n",
    "</tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
  "    v1  \\n",
  "0  ham  \n",
  "1  ham  \n",
  "2  spam \n",
  "3  ham  \n",
  "4  ham  \n",
  "\n",
  "
v2  \\n",
  "0  Go until jurong point, crazy.. Available only in bugis n great world la e buffet...
Cine there got amore wat... \n",
  "1  Ok      lar...      Joking      wif      u      oni...
\n",
  "2  Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to
87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's \n",
  "3  U dun say so early hor... U c already then say...
\n",
  "4  Nah I don't think he goes to usf, he lives around here though
\n",
  "\n",
  "  Unnamed: 2 Unnamed: 3 Unnamed: 4 \n",
  "0  NaN      NaN      NaN      \n",
  "1  NaN      NaN      NaN      \n",
  "2  NaN      NaN      NaN      \n",
  "3  NaN      NaN      NaN      \n",
  "4  NaN      NaN      NaN      "
]
},
"execution_count": 3,
"metadata": {},
"output_type": "execute_result"

```




```

},
"papermill": {
  "duration": 0.02496,
  "end_time": "2022-10-25T06:57:22.835489",
  "exception": false,
  "start_time": "2022-10-25T06:57:22.810529",
  "status": "completed"
},
"tags": [],
},
"outputs": [
  {
    "data": {
      "text/html": [
        "<div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border='1' class='dataframe'>\n",
        "  <thead>\n",
        "    <tr style='text-align: right;'>\n",
        "      <th></th>\n",
        "      <th>target</th>\n",
        "      <th>text</th>\n",
        "    </tr>\n",
        "  </thead>\n",
        "  <tbody>\n",
        "    <tr>\n",
        "      <th>0</th>\n",
        "      <td>ham</td>\n",
        "      <td>Go until jurong point, crazy.. Available only in bugis n great world la e
buffet... Cine there got amore wat...</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>1</th>\n",
        "      <td>ham</td>\n",
        "      <td>Ok lar... Joking wif u oni...</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>2</th>\n",
        "      <td>spam</td>\n",
        "      <td>Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text
FA to 87121 to receive entry question(std txt rate)T&amp;C's apply
08452810075over18's</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>3</th>\n",

```



```

"      <td>ham</td>\n",
"      <td>U dun say so early hor... U c already then say...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>ham</td>\n",
"      <td>Nah I don't think he goes to usf, he lives around here though</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
  "  target  \\n",
  "0  ham    \n",
  "1  ham    \n",
  "2  spam   \n",
  "3  ham    \n",
  "4  ham    \n",
  "\n",
  "
text  \n",
  "0  Go until jurong point, crazy.. Available only in bugis n great world la e buffet...
Cine there got amore wat...
  "1  Ok      lar...      Joking      wif      u      oni...
\n",
  "2  Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to
87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's  \n",
  "3  U  dun  say  so  early  hor...  U  c  already  then  say...
\n",
  "4  Nah I don't think he goes to usf, he lives around here though
"
]
},
"execution_count": 5,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
  "data.head()"
]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "id": "b159c84d",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:22.858628Z",
      "iopub.status.busy": "2022-10-25T06:57:22.858233Z",
      "iopub.status.idle": "2022-10-25T06:57:22.877346Z",
      "shell.execute_reply": "2022-10-25T06:57:22.875820Z"
    }
  },
  "papermill": {
    "duration": 0.034605,
    "end_time": "2022-10-25T06:57:22.880639",

```




```

    "exception": false,
    "start_time": "2022-10-25T06:57:22.846034",
    "status": "completed"
  },
  "tags": []
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 5572 entries, 0 to 5571\n",
      "Data columns (total 2 columns):\n",
      "#   Column   Non-Null Count  Dtype \n",
      "---  ---      -\n",
      "0    target   5572 non-null   object\n",
      "1    text     5572 non-null   object\n",
      "dtypes: object(2)\n",
      "memory usage: 87.2+ KB\n"
    ]
  }
],
"source": [
  "data.info(memory_usage = True, verbose = True)"
],
},
{
  "cell_type": "code",
  "execution_count": 7,
  "id": "22fd4667",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:22.904491Z",
      "iopub.status.busy": "2022-10-25T06:57:22.904118Z",
      "iopub.status.idle": "2022-10-25T06:57:22.914979Z",
      "shell.execute_reply": "2022-10-25T06:57:22.913596Z"
    },
    "papermill": {
      "duration": 0.025212,
      "end_time": "2022-10-25T06:57:22.917135",
      "exception": false,
      "start_time": "2022-10-25T06:57:22.891923",
      "status": "completed"
    }
  },
  "tags": []
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "\u001b[2m\u001b[1m\u001b[32mThere are 403 duplicated values in the dataset\u001b[0m\n"
    ]
  }
],
},
],

```



```

"source": [
  "print(colored(\"There are {} duplicated values in the
dataset\".format(data.duplicated().sum()), color = \"green\", attrs = [\"bold\", \"dark\"]))"
],
{
  "cell_type": "code",
  "execution_count": 8,
  "id": "8366d31d",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:22.940416Z",
      "iopub.status.busy": "2022-10-25T06:57:22.939912Z",
      "iopub.status.idle": "2022-10-25T06:57:22.951016Z",
      "shell.execute_reply": "2022-10-25T06:57:22.949690Z"
    },
    "papermill": {
      "duration": 0.025662,
      "end_time": "2022-10-25T06:57:22.953420",
      "exception": false,
      "start_time": "2022-10-25T06:57:22.927758",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "\u001b[2m\u001b[1m\u001b[32m\n",
        "DUPLICATED VALUES WERE SUCCESFULLY DROPPED...\u001b[0m\n"
      ]
    }
  ],
  "source": [
    "data.drop_duplicates(inplace = True)\n",
    "\n",
    "print(colored(\"\\nDUPLICATED VALUES WERE SUCCESFULLY DROPPED...\", color =
\"green\", attrs = [\"bold\", \"dark\"]))"
  ],
  {
    "cell_type": "code",
    "execution_count": 9,
    "id": "e6165596",
    "metadata": {
      "execution": {
        "iopub.execute_input": "2022-10-25T06:57:22.976487Z",
        "iopub.status.busy": "2022-10-25T06:57:22.976027Z",
        "iopub.status.idle": "2022-10-25T06:57:22.987356Z",
        "shell.execute_reply": "2022-10-25T06:57:22.986267Z"
      },
      "papermill": {
        "duration": 0.025566,
        "end_time": "2022-10-25T06:57:22.989684",
        "exception": false,

```



```

    "start_time": "2022-10-25T06:57:22.964118",
    "status": "completed"
  },
  "tags": []
},
"outputs": [
  {
    "data": {
      "text/plain": [
        "target\n",
        "ham      4516\n",
        "spam      653 \n",
        "Name: text, dtype: int64"
      ]
    },
    "execution_count": 9,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "grouped_target = data.groupby(\"target\").count()\n",
  "grouped_target[\"text\"]"
]
},
{
  "cell_type": "code",
  "execution_count": 10,
  "id": "f090a982",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:23.013136Z",
      "iopub.status.busy": "2022-10-25T06:57:23.012702Z",
      "iopub.status.idle": "2022-10-25T06:57:23.022896Z",
      "shell.execute_reply": "2022-10-25T06:57:23.021836Z"
    },
    "papermill": {
      "duration": 0.024597,
      "end_time": "2022-10-25T06:57:23.025255",
      "exception": false,
      "start_time": "2022-10-25T06:57:23.000658",
      "status": "completed"
    }
  },
  "tags": []
},
"outputs": [
  {
    "data": {
      "text/plain": [
        "target      0\n",
        "text        0\n",
        "dtype: int64"
      ]
    },
    "execution_count": 10,
    "metadata": {},
    "output_type": "execute_result"
  }
]

```



```

    }
  ],
  "source": [
    "data.isnull().sum()"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "id": "57c3a760",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:23.049228Z",
      "iopub.status.busy": "2022-10-25T06:57:23.048169Z",
      "iopub.status.idle": "2022-10-25T06:57:23.270177Z",
      "shell.execute_reply": "2022-10-25T06:57:23.269285Z"
    },
    "papermill": {
      "duration": 0.236181,
      "end_time": "2022-10-25T06:57:23.272320",
      "exception": false,
      "start_time": "2022-10-25T06:57:23.036139",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "data": {
        "image/png":

```

*iVBORw0KGgoAAAANSUhEUgAAAfoAAAHiCAYAAAAAAKAA6/AAAAOXRFWHRTb2Z0d2FyZQBB
 NYXRwbG90bGliIHZlcnNpb24zLjUuMiwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8qNh9FAAAA
 CXBIWXMAAAAsTAAALEwEAmpwYAAASvklEQVR4nO3deVxV9b7/8fcWPKQxQCyBBwfMHFBL
 DG2Q1HDKsFseHAtvHm28mU36K03NHmqW1elmGR1vUqFk1pE0p6ys7DocstTwWmigQh4cY
 lJUpu/vDx+th6iEJ90g317Pf2yvvfZan7XZe7/2BLmMMUYAAMBKDep6AAAA4D6EHgAAixF6A
 AAsRugBALAYoQcAwGKEHgAAixH6P5hp06Zp1KhRtbrP4cOHa9myZZKkhQsX6oYbbqjV/dtk+
 /btuu66635znby8PPXs2VM+Pj569NFHa2ky97r33ns1Y8aMWt3n5MmTFRQUObNm9fqfuub
 mTNn6q9//et5RtT69GhNnny52vNdLpd27959Xtvq3LmzQkJC9NJLL53X+n9khN5CixYtUnR0tLy
 9vRUSEqIBAwZow4YNdTLl9u3btW3bNg0ePLhO9m+bTp06yd/fX8uXL692naSkJAUFBamoqE
 hz586txencZ/78+ZoyZUqt7W/fvn2aO3eudu7cqX/9619nnb9+/XrddNNNzul/J1DucOb+z5zPnZ
 588kn9/e9/r5V9nW7btm168803NX369Frfd31D6C3z4osv6uGHH9aTTz6pvlw87du3T/fff7/S0
 tLqZJ433nhDI0eOImvlqpP9/17l5eV1PUK1Ro4cqTfeeKPa8/fu3av27dtXe51fysd2qdi3b5+aNG
 mipk2bun1fxhhVvIa6fT/uUNe3pY4dO6qwsFAVFRV1OseljtBbpLCwUE8//bTmzZun22+/XZdffr
 kaNmyoW2+9Vc8///w5L/OXv/xFzZs3l5+fn3r27KmMjAznvJUrv6p9+/by8fFRaGioXnjhBUnS4c
 OHNWjQIPn7+yswMFA33nhjtQ9Uq1atUmxs7FnLH3vsMQUEBKhy5ZatWqVs/ytt95Su3bt5OPj
 o1atWlUJ2vr16xUWFqY5c+aoadOmCgkJ0bJly7Ry5UpdeeWVCgwM1MyZM3/z+rnrrrsUHBysil
 gIPfvss87cCxcu1PXXX68JEyaoSZMmmjZt2lmX37Jli6Kjo+Xr66tmzZrpkUcekSRIZ2fL5XlpKSI
 JV1xxhUJCQpZr6ftL9ejRQ/7+/goJCdGDDz6o0tJS53yXy6XXXntNbdq0kY+Pj6ZMmal9e/bouu
 uk6+vrxiSEqqsf9NNN+nTTz/VyZMnz5px90jRsk501pw5c+Tt7a1169Zp2rRpGjJkiEaNGiVfX1
 8tXLhQhYWFgJnmjEJCQhQaGqrJkyc7D5YVFRV67LHHFBQUFatWmnevHlyuVzOg3qLfi20bt
 06Z59nfhy0adMmXXfddFL391fnzp21fv36KrNPMtJF119/vXx8fNS3b18dPnzYOX/Dhg30ZcP
 Dw7Vw4ULnuE5/y3fFihXq0qWL/P39dd1112n79u30ec8995xCQ0PI4+Ojtm3b6tNPPz3HraH62
 8O6desUFxenn3/+Wd7e3ho9evQ5L/+rnj17Sjr1VrK3t7fee+895efna9CgQqQoODIZAqIAGDRqkn
 JycKtfDU089peuvv16NGzfWTz/9pLVr16pt27by8/PT/ffr9jY2CqvlP/nf/5H7dq1U0BAGPr166e
 9e/dWu//fMmDAAL366qtVlnXu3FkffvihJGn8+PEKDw+Xr6+vunbtqq+++spZ71y3pTN//r/1mC
 Kdevyli4uTj4+PYmNjneM408mTJ/XYy4/pz3/+s5o1a6Z7771Xx48fr7JOgwanElbXTzgueQbW
 WLvqlfHw8DBIZWXVrjN16lQzcuRI5/SCBQtMUVGROXHihBk/frzp3Lmzc17z5s3NI19+aYwx5p*



dffjHffPONMcaYSZMmmXvuuceUlpaa0tJS8+WXX5rKysqz9nX06FEjyRw8eNBZ9tZbbxIPT0+T
lJRkysvLzWuvvWZCQkKcy69YscLs3r3bVFZWmvXr15tGjRo5+/3888+Nh4eHmT59uikTLTVJSU
kmKCjIDB8+3BQVFZnv//eXHbZZeann34657HfeedJj4+3hQVFZmsrCzTpk0b8/e//92Zy8PD
w7zyiumrKzMlJSUnHX57t27m7ffftsYY0xxcbHZuHGjMcaYrKwsl8kMGzbMHD161Gzfv0EBQ
WZTz75xBhjTHp6utm4caMpKyszWVIZ5qqrjlvfSSs11Jj4+3hQWFprvv//e/OIPfzK9e/c2e/bs
MQUFBaZdu3Zm4cKFVWbx8fEx27ZtO+dxJiYmmqeeeso5PXXqVOPp6Wn+8Y9/mlqKCINSU
mJuu+02M27cOHP06FGTI5dnunXrZubPn2+MMeb11183bdu2Nfv27TNHjhwXN910k5Hk3K4ili
KcY/t1+7/epnJyckxgYKD5+OOPTUVFhVm7dq0JDAx0bgOxsbGmVatW5ocffjAlJSUmNjbWTJ
w40RhjTHZ2tvH29jaLFi0ypaWl5vDhw+bbb78965i2bt1qgoODzaZNm0x5eblZuHChiYilMCdOn
DC7du0yYWFhJjc31/nZ7N69+9++PXz++ecmNDT0nJc7F0kmMzPTOX348GGzdOISc+zYMVN
UVGSGDBliBg8e7JwfGxtrwsPDzfff2/KysrMwYMHjY+Pj/nggw9MWVmZefnll42np6d58803jT
HGLFu2zLRu3drs3LnTIJWVmRkzZpgePXpUu//fkpycbK677jrndEZGhvHz8zMNtpwwxhjzzjvmm
MOHD5uysjLzwsGmGbnmpnjx48bY859W/p3HIMSExONT7e3+eKLL8yJEyFMQw89ZK6//vpzH
sfDDz9sbr31VnPKyBFTVFRkBg0aZCZNmlTIWEpKSSxll11mli9ffl7H/kdF6C3y7rvvmmbNmv3m
OmfeKU+Xn59vJmCggJjJDHh4eFm/vz5prCwsMp6U6ZMMfHx8TU+sOTk5BhJzoOEmaeC2rp
1a+f0sWPHjCRz4MCBc25j8ODB5uWXXzbGnHrwey0x5ebkxxpiioiljyWzatMIZ/5prjH/+Mc/
ztpOeXm5adiwocnlyHCWzZ8/38TGxjpzhYeH/+bx3Hjjjebpp582hw4dqrL819D/3//9n7Ps8ccfN
3ffffc5t/PSSy+Z2267zTkyWzYsKHKMcyePds5/cgjj5jx48dX2cYVv1xhvvjii3Nu/1yhv/HGG53T
//rXv8yf/vSnKk9mFi1aZG666SZjDG9evUyr7/+unPemjVrzjv0s2fPNqNGjaoyT9++fZ0nKrGxsW
bGjBnOefPmzTP9+vUzxhgzc+bMKtdLdcd07733msmTJ1c5/8orrrTr1683mZmZJjg42HzzySe
mtLT0nNsypubbw4WG/kzffvut8ff3d07HxsaaKVOMOkEtk5NN9+7dndOVIZUmLCzMCX3//v2
dJyHGGFNRUWEaNWpksrOzz2v/pysqKjKNGzd2Lvkk0+a//zP/6x2fX9/f/Pdd98ZY86+Lf267H
wfUxITE83QoUOd84uLi02DBG3Mvn37qhxHZWWlady4cZUnaf/7v/9rWrRocdy+XnnlFeNyua08
oUBVvHVvkSZNmujw4cPn/TZWRUWFJk2apNatW8vX11ctWrSQJOet1A8++EarV65URESEYm
NjTXHjRknS448/rsjSPXt21etWrXS7Nmzz7l9f39/SVJxcXGV5ad/i7lx48aSpKNHj0o69VZ/9+7d
FRgYKH9/f61cubLKW7tNmjSRh4eHJKIRo0aSpGbNmjnnN2rUyNnW6Q4fPqyysjJFREQ4yyilpS
bm+ucDg8Pr+6qkiQtWLBAP/74o6666ip169ZNK1asqHL+6ZePiljQzz//LEn68ccfNWjQIDVv3ly
+vr568sknqxZTuY6hpmMqLi52rt/zcfpse/fuVVIzmUJCQuTv7y9/f3/dc889OnjwoCTp559/PutY
ztfvXv1/vvv09v19/fXhg0bdODAAWedM3/+vx7b/v371bp16/Pax9y5c6vsY//+/fr5558VGRmpl
19+WdOmTVPTpk01bNgw5+dwuv05PVyikpIS3XPPPYqliJCvr6969uyppoKCKp8ln34dn3mdu1
wuhYWFVTnm8ePHO8cbGBgoY8zvmfHx0e33HKLUIINTJUmlFy/WyJEjnfNfeOEftWvXTn5+fv
L391dhYWGv2+tv3U9qekw58/Le3t4KDAw862d06NAhlZSUqGvXrs4x9+/fX4cOHagYXnl5uaZO
nap33nlH33777b99XfxREHqL90jRQ15eXs6vstVk0aJFSktL07p161RYWKjs7GxJp74cJEndunV
TWlqaDh48qNtuu00JCQmSTj1QzJ07Vz/99JM++ugjvfjii+f8HPTyyy9X69at9eOPP57XPCdPnt
Qdd9yhxx57THl5eSookNDAGQOdeS5EUFCQGjZsWOXzwh379ik0NNQ5XdMXBtu0aaPFixfr4
MGDMjhxooYMGaJx4455+/fv7/Ktq+44gpJ0n333aerrrKmZmZKioq0syZMy/omHJzc1VaWq
q2bdue92VOP7bw8HB5eXnp8OHDKigoUEFBgYqKipzPUkNCQs46ltNdfvnlKikpcU6f/q308PBw
3Xnnnc52CwoKdOzYMU2aNKnGGcPDw7Vnz57zWu+pp56qso+SkhlnHz5ckjRixAht2LBBe/ful
cvl0sSJE8/axvncHi7E3Llz9cMPP2jz5s0qKirSl19+KUIVfu6n/0xCQKqKfIzvjKlyOjw8XG+88UaV
Yz5+/HiNv2pZneHDh2vx4sXauHGjTPw4oV69ekmSvvrqK82ZM0dLlixRfn6+CgoK50fnV+3cZ6r
pMUWqej85evSofvnlF+e+8qugoCA1atRIGRkZzvEWFhae9YQ3Ly9P+fn5uu222+rdF35E6G3iJ+
fn5555hk98MADWrZsmUpKSIRWVqZVq1bpiSeeOGv94uJieXl5qUmTJiopKdGTTz7pnFdaWqq
UIBQVFhaqYcOG8vX1db74smLFCu3evVvGGPn5+cnDw8M570wDBw7UF198cV7zl5aW6uTJk
woODpanp6dWrVqltWvX/o5r4mweHh5KSEjQU089peLiYu3du1cvvvjiv/U3Bd59910dOnRIDRo0
cF5Nn37cM2bMUElJiTiYmVtWW29p6NChhk5dz76+vvL29tauXbv0+uuvX9CxfPHFF+rdu7e8vL
x+1+VDQkLUt29fPfrooyoqKlJlZaX27Nnj/JwSEhL0yiuVkcCnR/n5+We9Y90ISxelpqaqrKxM6en
pWrp0qXPeqFGjtHz5cq1Zs0YVFRU6ceKE1q9fXyVa1Rk5cqTWrvunJUuWqLy8XEeOHNF3331
31npjx47V/PnztXnzZhldOzYMX388ccqLi7WDz/8oM8++0wnT57UZZddpkANGp3ztnkxbg+na9
asmX766SfndHFxsRo1aiR/f3/98ssvNf4K2C233KId03Z02bJIKi8v17x586o8gbr33ns1a9Ys58l
YYWGH3n//Wr3X50BAwdq7969evrppzV06FDnOiouLpanp6eCg4NVXl6uZ555RkVFRee93d96
TPnVypUrtWHDBpWWImrKlCnq3r37We8SNGjQQGPHjtWECROcd5pyc301Zs2aKuuVlZVJ0u+
+L/xREHrLPProo3rxxRf17LPPKjg4WOHh4Xr11Vd12223nbXuXXfdpYilCIWGHqp9+/bq3r17lFpf
eecdTjWRQR6+vpo/f75SUllkSZmZmbr55pvl7e2tHj166P7773deEZxp3LhxSkIJ0a9XsD4+PnrlV
eUkJCgglAALVq0SPHx8f/+IVCN//7v/9bll1+uVq1a6YYbbtCIESN09913n/flV69erQ4d0sJb21vjx
49Xamqq8/GBJMXGxiyMIJ9+vTRY489pr59+0o69VbookWL50Pjo7FjxZpPAH6vIJQU3XvvvRe
0jbffflulpaVq3769AgICNGTIEOf9bFjx6pfv37q3LmzrrnmGt1+++1Vltjxgzt2bNHAQEbmjp1qk
aMGOgcFx4errSONM2cOd05/T3//PPn9etjf/7zn7Vy5UrNnTtXgYGB6tKli7Zt23bWetHR0XrzzT



f14IMPKiAgQJGRkc6380+ePKIJkyY5f+jm4MGDMjVr1jn3d6G3h9NNmzZNiYmJ8vf315IIS/Tw
ww/r+PHjCgoKUvfu3dW/f//fvHxQUJDef/99PfHEE2rSpll27typ6OhoJ2D/8R//oYkTJ2rYsGHY9f
VVx44dq/y2ypn7r4mXl5duv/12rVu3rsrPr1+/furfv7+uvPJKRURE6LLLLqvx163T1fSYlp16x2X69
OkKDAzUN998o3ffffec23ruuecUGRmp7t27y9fXVzffflN++OGHKuv8+IFldS80clrLXlZ3RYHfMG
LECCUkJJzzyYYNsrOz1bJIS5WVlcnT09Ot+9q+fbvuuece5/sStaE2jw+nVFZWKiwsTCkpKdU+i
Ya0ZMkSPfjgg86rfpwb91q43aJFi+p6BGt06tSpViOP2rNmzRrFxmSoUaNGev7552WMOecrYpz
StWtXFRYW6rnnnqvrUS55hB4ALgEbN27UiBEjnl9Uli1bVuWjIVT1zTff1PUI9QZv3QMAYDG+wQ
AAgMUIPQAAFrPyM/qgoCDnLzIBAPBHkJ2dfdZf3ZQsDX2Lfi2Unp5e12MAAFBroqOjz7mct+4
BALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGK
EHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAi3nW9QD1
zeYoj7oeAbhgMTsq6noEALWEV/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDFC
DwCaxQg9AAAWI/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDFCDwCaxQg9AA
AWI/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0
AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDFCDwC
AxQg9AAAWI/QAAFiM0AMAYDFCDwCaxQg9AAAWI/QAAFiM0AMAYDG3h76ioKJXX321Bg0a
JEnKyspSTEyMliMjNXTToUJWWlkqSTp48qaFDhyoyMliXMT HKzs52tjFr1ixFRkaqbdu2WrNmjbt
HBgDAGm4P/d/+9je1a9fOOT1x4kRNmDBBu3fvVkBAGBYsWCBJWrBggQICAr7925NmDBBE
ydOICTt3LITqampysjl0OrVq3X//feroqLC3WMDAGAFt4Y+JydHH3/8sf76179Kkoxw+uyzzzRky
BBJUmJiopYtWyZJSktLU2JioiRpyJAh+vTTT2WMUVpamoYNGyYvLy+1bNISKZGR2rJlizvHBgD
AGm4N/cMPP6w5c+aoQYNTuzly5lj8/f3l6ekpSQoLC1Nubq4kKTc3V+Hh4ZlKt09P+fn56ciRl1
WWn3kZAADw29wW+hUrVqhp06bq2rWru3ZRRVJSkqKjoxUdHa1Dhw7Vyj4BALjUebprw19//b
U+++ugjrVy5UidOnFBRUZHgjx+vgoIClZeXy9PTUzk50QoNDZUkhYaGav+/QoLC1N5ebkKCwvV
pEkTZ/mvTr/M6caNG6dx48ZJkqKjo911WAAA1Ctue0U/a9Ys5eTkKDs7W6mpqerdu7dSUILU
q1cvLV26VJKUnJyswYMH5Li4+OVnJwsSVq6dKI69+4tl8ul+Ph4paam6uTJk8rKylJmZqauvf
Zad40NAIBV3PaKvjRPPfechg0bpsmTJ+vqq6/WmDFJEljxozRnXfeqcjlSAUGBio1NVWS1KFDB
yUkJKh9+/by9PTUvHnz5OHhUdtjAwBQL7mMMaauh7jYoqOjlZ6e7pZtb47ISQbqv5gd/loqYJvq
2sdfxgMAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEH
gAAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRug
BALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGK
EHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAs
RugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAw
GKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AA
AsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQc
AwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6
AAAsRugBALAYoQcAwGKEHGAiXF6AAAs5rbQnzxhQtdee606d+6sDh06aOrUqZKkrKwsxcTE
KDlyUkOHDIVpaakk6eTJkxo6dKgilyMVExOj7OxsZ1uzZs1SZGSk2rZtqzVr1rhrZAAArOO20Ht5
eemzzz7Ttm3b9N1332n16tXatGmTJk6cqAkTJmj37t0KCAjQggULJEKLFixQQECadu/erQkTJ
mjixlmSpJ07dyo1NVUZGRlavXq17r//flVUvLhrbAAArOK20LtcLnl7e0uSyrKVFZWJpfLpc8++0
xDhgyRJCUmJmrZsmWSpLS0NCUmJkqShgwZok8//VTGGKWlpWnYsGHY8vJSy5YtFRkZqS1
btrhrbAAArOLWz+grKirUpUsXNW3aVHFxcWrdurX8/f3l6ekpSQoLC1Nubq4kKTc3V+Hh4ZlKt0
9P+fn56ciRl1WWn3kZAADw29waeg8PD3333XfKycnRli1btGvXLrftKykpSdHR0YqOjtahQ4fcth
8AAOqTWvnWvb+/v3r16qWNGzeqoKBA5eXlkqScnByFhoZKkkJDQ7V//35JUnl5uQoLC9WkSZ
Mqy8+8zOnGjRun9PR0paenKzg4uBaOCgCAS5/bQn/o0CEVFBRIko4fP65PPvIE7dq1U69evbR
06VJJUnJysgYPHixJio+PV3JysiRp6dKI6t27t1wul+Lj45WamqqTJ08qKytLmZmZuvbaa901Ng
AAVvF014YPHDigxMREVVruqLKyuGkJCRo0aJdat2+vYcOGafLkybr66qs1ZswYSdKYMWN05
513KjlyUoGBgUpNTZUKdejQQKJCWrfvr08PT01b948eXh4uGtsAACs4jLGmLoe4mKLjo5Wen
q6W7a90YonGaj/YnbwK6qAbaprH38ZDwAAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAs
RugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAw
GKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AA
AsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQcAwGKEHGAiXF6AAAsRugBALAYoQc
AwGKEHGAiXF6AAAsRugBALBYjaHPyso6r2UAAODSU2Po77jjrOWDRkyxC3DAACAi8uzujN2
7dqJlwMFRYW6sMPP3SWFuxUV6cSJE7UyHAAAUDDVhv6HH37QihUrVFBQoOXLlZvLfxh890a
bb9bKcAAA4MJUG/rBgwdr8ODB2rhxo3r06FGbMwEAglukxs/omzRpoj59+qhjx46Sp03bt+vZZ
591+2AAAODC1Rj6sWPHatasWWrYsKEkqVOnTkPNTXX7YAAA4MLVGPqSkhJde+21VZZ5elb
7jj8AALIE1Bj6oKAg7dmzRy6XS5K0dOIShYSEuH0wAABw4Wp8aT5v3jyNGzdOu3btUmhoqFq2
bKI33323NmYDAAAXqMbQt2rVSuvWrdOxY8dUWVkpHx+f2pgLAABcBDWG/sUXXzxrmZ+fn7
p27aouXbq4YyYAAHCR1PgZfXp6uubPn6/c3FzI5ubqjTfe0OrVqzV27FjNmTOnNmYEAAC/U42





6NGj6/ZAgEvcsWPHdMstt6hz587q2LGj3nvvPbVo0UJPPPGEqKid02112r37t2SpOXLlysmJk
 ZXX321br75ZuXI5UmSpk2bpsTERN14442KiljQhx9+6Fy+f//+Kisrq8tDxHki9KhVmZmZeuCBB
 5SRkSF/f3998MEHuv322/XPf/5T27ZtU7t27bRgwQJn/fz8fG3cuFEvvfSS4uPjNWHCBGVkZGj
 Hjh367rvv6u5AgEvc6tWrddUUV2jbtm36/vvv1b9/f0mSn5+fduzYoQcffFAPP/ywJOMGG27Qpk
 2b9O2332rYsGGaM2eOs509e/bos88+00cffaRRo0apV69e2rFjhxo1aqSPP/64Lg4N/yZCj1rVs
 mVLdenSRZLUtWtXZWdn6/vvv9eNN96oqKgopaSkKCMjw1n/1ltvlcvlUIRUIJo1a6aoqCg1aNBA
 HTp0qPbPPQKQoqKi9Mknn2jixln66quv50fnJ0kaPny48+/GjRslnfpro/369VNUVJSef/75KvfBA
 QMGqGHDhoqKilJFRYXzhCEqKor7YD1B6FGrvLy8nP/28PBQeXm5Ro8erVdffVU7duzQ1KITde
 LEibPWb9CgQZXLNmjQQOXI5bU3OFDPXhnlldq6dauioql0efJkPfPMM5IkI8vlrPPrf//Xf/2XHnz
 wQe3YsUNvvPFGtffBhg0bOpfhPlh/EHrUueLiYoWEhKisrEwpKSI1PQ5ghZ9//lmNGzfWqFGj9Pj
 jj2vr1q2SpPfee8/5t0ePHpKkwsJC5/8fkpycXDcDw23qzV/Gg71mzJihmJgYBQcHKyYmRsXFx
 XU9EIDv7dixQ48//rjzSvz111/XkCFDIJ+fr06d0snLy0uLFy+WdOpLd3/5y18UEBCg3r17Kysrq46
 nx8XEX8YDgD+IFI1aKD09XUFBQXU9CmoRb90DAGAxXtEDAGAxXtEDAGAxQg8AgMUIPQAAF
 iPOAABYJNADAGAxQg8AgMX+P5bV6lZYD6aFAAAAAEIFTkSuQmCC\n",

```

    "text/plain": [
      "<Figure size 576x576 with 1 Axes>"
    ],
    "metadata": {},
    "output_type": "display_data"
  }
],
"source": [
  "plt.figure(figsize = [8, 8], clear = True, facecolor = \"white\")\n",
  "\n",
  "sns.barplot(x = data[\"target\"].value_counts().index,\n",
  "            y = data[\"target\"].value_counts(), palette = [\"#C82C02\", \"#410F01\"],\n",
  "            saturation = 1).set(title = \"Class (ham or spam) frequencies of 'target'\n",
  "variable}\");",
  ],
},
{
  "cell_type": "code",
  "execution_count": 12,
  "id": "fa3032b1",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:23.297362Z",
      "iopub.status.busy": "2022-10-25T06:57:23.296147Z",
      "iopub.status.idle": "2022-10-25T06:57:24.348531Z",
      "shell.execute_reply": "2022-10-25T06:57:24.347280Z"
    },
    "papermill": {
      "duration": 1.06822,
      "end_time": "2022-10-25T06:57:24.351746",
      "exception": false,
      "start_time": "2022-10-25T06:57:23.283526",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<script type=\"text/javascript\">\n",
          "window.PlotlyConfig = {MathJaxConfig: 'local'};\n",

```




```

"
                                if (window.MathJax && window.MathJax.Hub &&
window.MathJax.Hub.Config) {window.MathJax.Hub.Config({SVG: {font: \"STIX-Web\"}});}\n",
"
                                if (typeof require !== 'undefined') {\n",
"
                                require.undef(\"plotly\");\n",
"
                                requirejs.config({\n",
"
                                    paths: {\n",
"
                                        'plotly': ['https://cdn.plot.ly/plotly-2.12.1.min']\n",
"
                                    },\n",
"
                                });\n",
"
                                require(['plotly'], function(Plotly) {\n",
"
                                    window._Plotly = Plotly;\n",
"
                                });\n",
"
                                }\n",
"
                                </script>\n",
"
                                "
                                ]
                                },
                                "metadata": {},
                                "output_type": "display_data"
                                },
                                {
                                "data": {
                                "text/html": [
                                "<div>
                                <div id=\"9ac0bb0c-3376-4e43-90c9-
d3f735ee9ab4\" class=\"plotly-graph-div\" style=\"height:500px; width:1000px;\"></div>
<script type=\"text/javascript\">
                                require(['plotly'], function(Plotly)
                                {
                                window.PLOTLYENV=window.PLOTLYENV || {};
                                if
                                (document.getElementById(\"9ac0bb0c-3376-4e43-90c9-d3f735ee9ab4\"))
                                {
                                Plotly.newPlot(
                                \"9ac0bb0c-3376-4e43-
90c9-d3f735ee9ab4\",
                                [{\"domain\":{\"x\":[0.0,1.0],\"y\":[0.0,1.0]},\"hole\":0.4,\"hovertemplate\": \"target=%{label}<extr
a></extra>\", \"labels\":[\"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"
spam\", \"spam\", \"ham\", \"spam\", \"spam\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\",
\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"
ham\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"h
am\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha
m\", \"ham\", \"spam\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha
m\", \"ham\", \"spam\", \"ham\", \"spam\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha
m\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\",
\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"spam\", \"ham\",
\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"h
am\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"s
pam\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"h
am\", \"ham\", \"spam\", \"spam\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"h
am\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha
m\", \"ham\", \"ham\", \"ham\", \"spam\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha
m\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"spam
\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\",
\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"h
am\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham
\", \"spam\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"spam
\", \"ham\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\",
\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"h
am\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"spam\", \"
ham\", \"spam\", \"ham\", \"ham\", \"spam\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"
ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ham\", \"ha

```



[illegible]

[illegible]

[[0.0,\"#0d0887\"],[0.1111111111111111,\"#46039f\"],[0.222222222222222,\"#7201a8\"],[0.333333333333333,\"#9c179e\"],[0.444444444444444,\"#bd3786\"],[0.555555555555556,\"#d8576b\"],[0.666666666666666,\"#ed7953\"],[0.777777777777778,\"#fb9f3a\"],[0.888888888888888,\"#fdca26\"],[1.0,\"#f0f921\"]]],\"contourcarpet\":{\"type\":\"contourcarpet\", \"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"},\"contour\":{\"type\":\"contour\", \"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"},\"colorscale\":[[0.0,\"#0d0887\"],[0.111111111111111,\"#46039f\"],[0.222222222222222,\"#7201a8\"],[0.333333333333333,\"#9c179e\"],[0.444444444444444,\"#bd3786\"],[0.555555555555556,\"#d8576b\"],[0.666666666666666,\"#ed7953\"],[0.777777777777778,\"#fb9f3a\"],[0.888888888888888,\"#fdca26\"],[1.0,\"#f0f921\"]]],\"surface\":{\"type\":\"surface\", \"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"},\"colorscale\":[[0.0,\"#0d0887\"],[0.111111111111111,\"#46039f\"],[0.222222222222222,\"#7201a8\"],[0.333333333333333,\"#9c179e\"],[0.444444444444444,\"#bd3786\"],[0.555555555555556,\"#d8576b\"],[0.666666666666666,\"#ed7953\"],[0.777777777777778,\"#fb9f3a\"],[0.888888888888888,\"#fdca26\"],[1.0,\"#f0f921\"]]],\"mesh3d\":{\"type\":\"mesh3d\", \"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"},\"scatter\":{\"fillpattern\":{\"fillmode\":\"overlay\", \"size\":10,\"solidity\":0.2},\"type\":\"scatter\"},\"parcoords\":{\"type\":\"parcoords\", \"line\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scatterpolargl\":{\"type\":\"scatterpolargl\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"bar\":{\"error_x\":{\"color\":\"#2a3f5f\"},\"error_y\":{\"color\":\"#2a3f5f\"},\"marker\":{\"line\":{\"color\":\"#E5ECF6\"},\"width\":0.5},\"pattern\":{\"fillmode\":\"overlay\", \"size\":10,\"solidity\":0.2}},\"type\":\"bar\"}},\"scattergeo\":{\"type\":\"scattergeo\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scatterpolar\":{\"type\":\"scatterpolar\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"histogram\":{\"marker\":{\"pattern\":{\"fillmode\":\"overlay\", \"size\":10,\"solidity\":0.2},\"type\":\"histogram\"}},\"scattergl\":{\"type\":\"scattergl\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scatter3d\":{\"type\":\"scatter3d\", \"line\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scattermapbox\":{\"type\":\"scattermapbox\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scatterternary\":{\"type\":\"scatterternary\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"scattercarpet\":{\"type\":\"scattercarpet\", \"marker\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}},\"carpet\":{\"aaxis\":{\"endlinecolor\":\"#2a3f5f\", \"gridcolor\":\"white\", \"linecolor\":\"white\", \"minorgridcolor\":\"white\", \"startlinecolor\":\"#2a3f5f\"}, \"baxis\":{\"endlinecolor\":\"#2a3f5f\", \"gridcolor\":\"white\", \"linecolor\":\"white\", \"minorgridcolor\":\"white\", \"startlinecolor\":\"#2a3f5f\"}, \"type\":\"carpet\"}},\"table\":{\"cells\":{\"fill\":{\"color\":\"#EBF0F8\"}, \"line\":{\"color\":\"white\"}},\"header\":{\"fill\":{\"color\":\"#C8D4E3\"}, \"line\":{\"color\":\"white\"}},\"type\":\"table\"}},\"barpolar\":{\"marker\":{\"line\":{\"color\":\"#E5ECF6\"}, \"width\":0.5}, \"pattern\":{\"fillmode\":\"overlay\", \"size\":10,\"solidity\":0.2}, \"type\":\"barpolar\"}},\"pie\":{\"automargin\":true,\"type\":\"pie\"}},\"layout\":{\"autotypenumbers\":\"strict\", \"colorway\":[\"#636efa\", \"#EF553B\", \"#00cc96\", \"#ab63fa\", \"#FFA15A\", \"#19d3f3\", \"#FF6692\", \"#B6E880\", \"#FF97FF\", \"#FECB52\"], \"font\":{\"color\":\"#2a3f5f\"}, \"hovermode\":\"closest\", \"hoverlabel\":{\"align\":\"left\"}, \"paper_bgcolor\":\"white\", \"plot_bgcolor\":\"#E5ECF6\", \"polar\":{\"bgcolor\":\"#E5ECF6\", \"angularaxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\"}, \"radialaxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\"}}, \"ternary\":{\"bgcolor\":\"#E5ECF6\", \"aaxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\"}, \"baxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\"}, \"caxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\"}}, \"coloraxis\":{\"colorbar\":{\"outlinewidth\":0,\"ticks\":\"\"}, \"colorscale\":{\"sequential\":[[0.0,\"#0d0887\"],[0.111111111111111,\"#46039f\"],[0.222222222222222,\"#7201a8\"],[0.333333333333333,\"#9c179e\"],[0.444444444444444,\"#bd3786\"],[0.555555555555556,\"#d8576b\"],[0.666666666666666,\"#ed7953\"],[0.777777777777778,\"#fb9f3a\"],[0.888888888888888,\"#fdca26\"],[1.0,\"#f0f921\"]]], \"diverging\":[[0,\"#8e0152\"],[0.1,\"#c51b7d\"],[0.2,\"#de77ae\"],[0.3,\"#fb6da\"],[0.4,\"#fde0ef\"],[0.5,\"#f7f7f7\"],[0.6,\"#e6f5d0\"],[0.7,\"#b8e186\"],[0.8,\"#7fbc41\"],[0.9,\"#4d9221\"],[1,\"#276419\"]]], \"xaxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\", \"title\":{\"standoff\":15}, \"zerolinecolor\":\"white\", \"automargin\":true, \"zerolinewidth\":2}, \"yaxis\":{\"gridcolor\":\"white\", \"linecolor\":\"white\", \"ticks\":\"\", \"title\":{\"standoff\":15}, \"zerolinecolor\":\"wh



```

ite\","automargin\":true,\"zerolinewidth\":2,\"scene\":{\\"axis\":{\\"backgroundcolor\":\"#E5ECF6\",\\"gridcolor\":\"white\",\\"linecolor\":\"white\",\\"showbackground\":true,\"ticks\":\"\",\\"zerolinecolor\":\"white\",\\"gridwidth\":2,\"yaxis\":{\\"backgroundcolor\":\"#E5ECF6\",\\"gridcolor\":\"white\",\\"linecolor\":\"white\",\\"showbackground\":true,\"ticks\":\"\",\\"zerolinecolor\":\"white\",\\"gridwidth\":2,\"zaxis\":{\\"backgroundcolor\":\"#E5ECF6\",\\"gridcolor\":\"white\",\\"linecolor\":\"white\",\\"showbackground\":true,\"ticks\":\"\",\\"zerolinecolor\":\"white\",\\"gridwidth\":2}},\"shapedefaults\":{\\"line\":{\\"color\":\"#2a3f5f\"}},\"annotationdefaults\":{\\"arrowcolor\":\"#2a3f5f\",\\"arrowhead\":0,\"arrowwidth\":1},\"geo\":{\\"bgcolor\":\"white\",\\"landcolor\":\"#E5ECF6\",\\"subunitcolor\":\"white\",\\"showland\":true,\"showlakes\":true,\"lakecolor\":\"white\"},\"title\":{\\"x\":0.05,\"mapbox\":{\\"style\":\"light\"}},\"legend\":{\\"tracegroupgap\":0},\"title\":{\\"text\":\"counts in 'target' variable\",\\"font\":{\\"size\":30,\"x\":0.5},\"piecolorway\":{\\"rgb(92, 83, 165)\",\\"rgb(160, 89, 160)\",\\"rgb(206, 102, 147)\",\\"rgb(235, 127, 134)\",\\"rgb(248, 160, 126)\",\\"rgb(250, 196, 132)\",\\"rgb(243, 155)\",\"height\":500,\"width\":1000,\"uniformtext\":{\\"minsize\":25}},\"responsive\": true}
).then(function(){\n",
    "\n",
    "var gd = document.getElementById('9ac0bb0c-3376-4e43-90c9-d3f735ee9ab4');\n",
    "var x = new MutationObserver(function (mutations, observer) {\n",
    "    var display = window.getComputedStyle(gd).display;\n",
    "    if (!display || display === 'none') {\n",
    "        console.log([gd, 'removed!']);\n",
    "        Plotly.purge(gd);\n",
    "        observer.disconnect();\n",
    "    }\n",
    "});\n",
    "\n",
    "// Listen for the removal of the full notebook cells\n",
    "var notebookContainer = gd.closest('#notebook-container');\n",
    "if (notebookContainer) {\n",
    "    x.observe(notebookContainer, {childList: true});\n",
    "}\n",
    "\n",
    "// Listen for the clearing of the current output cell\n",
    "var outputEl = gd.closest('.output');\n",
    "if (outputEl) {\n",
    "    x.observe(outputEl, {childList: true});\n",
    "}\n",
    "\n",
    "});
}
},
"metadata": {},
"output_type": "display_data"
},
"source": [
    "fig = px.pie(data_frame = data, names = \"target\", hole = 0.4, title = \"counts in 'target' variable\", \n",
    "              width = 1000, height = 500, color_discrete_sequence = \n",
    "px.colors.sequential.Sunset_r)\n",
    "\n",
    "fig.update_traces(textposition = \"inside\", textinfo = \"percent+label\", \n",
    "                  marker = dict(line = dict(width = 1.5, color = \"#41F01\")))\n",
    "\n",
    "fig.update_layout(title_x = 0.5, title_font = dict(size = 30), uniformtext_minsize = 25)\n",

```



```

    "fig.show()"
  ]
},
{
  "cell_type": "code",
  "execution_count": 13,
  "id": "23a2778d",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:24.378990Z",
      "iopub.status.busy": "2022-10-25T06:57:24.377962Z",
      "iopub.status.idle": "2022-10-25T06:57:26.479299Z",
      "shell.execute_reply": "2022-10-25T06:57:26.478213Z"
    },
    "papermill": {
      "duration": 2.117594,
      "end_time": "2022-10-25T06:57:26.481852",
      "exception": false,
      "start_time": "2022-10-25T06:57:24.364258",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<style type=\"text/css\">\n",
          "#T_88c97_row0_col0,          #T_88c97_row1_col0,          #T_88c97_row1_col3,\n",
          "#T_88c97_row2_col0, #T_88c97_row2_col1, #T_88c97_row2_col2, #T_88c97_row2_col3,\n",
          "#T_88c97_row2_col4, #T_88c97_row2_col5, #T_88c97_row2_col6, #T_88c97_row2_col7 {\n",
          "  background-color: #3f007d;\n",
          "  color: #f1f1f1;\n",
          "}\n",
          "#T_88c97_row0_col1,          #T_88c97_row0_col2,          #T_88c97_row0_col3,\n",
          "#T_88c97_row0_col4, #T_88c97_row0_col5, #T_88c97_row0_col6, #T_88c97_row0_col7 {\n",
          "  background-color: #fcfbfd;\n",
          "  color: #000000;\n",
          "}\n",
          "#T_88c97_row1_col1 {\n",
          "  background-color: #63449d;\n",
          "  color: #f1f1f1;\n",
          "}\n",
          "#T_88c97_row1_col2, #T_88c97_row1_col6 {\n",
          "  background-color: #63439c;\n",
          "  color: #f1f1f1;\n",
          "}\n",
          "#T_88c97_row1_col4 {\n",
          "  background-color: #66499f;\n",
          "  color: #f1f1f1;\n",
          "}\n",
          "#T_88c97_row1_col5 {\n",
          "  background-color: #674ca1;\n",
          "  color: #f1f1f1;\n",
          "}\n",
          "#T_88c97_row1_col7 {\n",
          "  background-color: #64459e;\n",

```



```

" color: #f1f1f1;\n",
"}\n",
"</style>\n",
"<table id=\"T_88c97_\">\n",
" <thead>\n",
" <tr>\n",
" <th class=\"blank level0\" >&nbsp;</th>\n",
" <th class=\"col_heading level0 col0\" >count</th>\n",
" <th class=\"col_heading level0 col1\" >mean</th>\n",
" <th class=\"col_heading level0 col2\" >std</th>\n",
" <th class=\"col_heading level0 col3\" >min</th>\n",
" <th class=\"col_heading level0 col4\" >25%</th>\n",
" <th class=\"col_heading level0 col5\" >50%</th>\n",
" <th class=\"col_heading level0 col6\" >75%</th>\n",
" <th class=\"col_heading level0 col7\" >max</th>\n",
" </tr>\n",
" </thead>\n",
" <tbody>\n",
" <tr>\n",
" <th id=\"T_88c97_level0_row0\" class=\"row_heading level0 row0\"
>ch_no</th>\n",
" <td id=\"T_88c97_row0_col0\" class=\"data row0 col0\" >5169.000000</td>\n",
" <td id=\"T_88c97_row0_col1\" class=\"data row0 col1\" >78.977945</td>\n",
" <td id=\"T_88c97_row0_col2\" class=\"data row0 col2\" >58.236293</td>\n",
" <td id=\"T_88c97_row0_col3\" class=\"data row0 col3\" >2.000000</td>\n",
" <td id=\"T_88c97_row0_col4\" class=\"data row0 col4\" >36.000000</td>\n",
" <td id=\"T_88c97_row0_col5\" class=\"data row0 col5\" >60.000000</td>\n",
" <td id=\"T_88c97_row0_col6\" class=\"data row0 col6\" >117.000000</td>\n",
" <td id=\"T_88c97_row0_col7\" class=\"data row0 col7\" >910.000000</td>\n",
" </tr>\n",
" <tr>\n",
" <th id=\"T_88c97_level0_row1\" class=\"row_heading level0 row1\"
>wd_no</th>\n",
" <td id=\"T_88c97_row1_col0\" class=\"data row1 col0\" >5169.000000</td>\n",
" <td id=\"T_88c97_row1_col1\" class=\"data row1 col1\" >18.453279</td>\n",
" <td id=\"T_88c97_row1_col2\" class=\"data row1 col2\" >13.324793</td>\n",
" <td id=\"T_88c97_row1_col3\" class=\"data row1 col3\" >1.000000</td>\n",
" <td id=\"T_88c97_row1_col4\" class=\"data row1 col4\" >9.000000</td>\n",
" <td id=\"T_88c97_row1_col5\" class=\"data row1 col5\" >15.000000</td>\n",
" <td id=\"T_88c97_row1_col6\" class=\"data row1 col6\" >26.000000</td>\n",
" <td id=\"T_88c97_row1_col7\" class=\"data row1 col7\" >220.000000</td>\n",
" </tr>\n",
" <tr>\n",
" <th id=\"T_88c97_level0_row2\" class=\"row_heading level0 row2\"
>sn_no</th>\n",
" <td id=\"T_88c97_row2_col0\" class=\"data row2 col0\" >5169.000000</td>\n",
" <td id=\"T_88c97_row2_col1\" class=\"data row2 col1\" >1.947185</td>\n",
" <td id=\"T_88c97_row2_col2\" class=\"data row2 col2\" >1.362406</td>\n",
" <td id=\"T_88c97_row2_col3\" class=\"data row2 col3\" >1.000000</td>\n",
" <td id=\"T_88c97_row2_col4\" class=\"data row2 col4\" >1.000000</td>\n",
" <td id=\"T_88c97_row2_col5\" class=\"data row2 col5\" >1.000000</td>\n",
" <td id=\"T_88c97_row2_col6\" class=\"data row2 col6\" >2.000000</td>\n",
" <td id=\"T_88c97_row2_col7\" class=\"data row2 col7\" >28.000000</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
],

```



```

"text/plain": [
  "<pandas.io.formats.style.Styler at 0x7f74c6eb7810>"
],
"execution_count": 13,
"metadata": {},
"output_type": "execute_result"
},
{
  "source": [
    "data[\"ch_no\"] = data[\"text\"].apply(len)\n",
    "data[\"wd_no\"] = data.apply(lambda row: nltk.word_tokenize(row[\"text\"]),\n",
    "axis=1).apply(len)\n",
    "data[\"sn_no\"] = data.apply(lambda row: nltk.sent_tokenize(row[\"text\"]),\n",
    "axis=1).apply(len)\n",
    "\n",
    "data.describe().T.style.background_gradient(cmap = \"Purples_r\")"
  ],
  "cell_type": "code",
  "execution_count": 14,
  "id": "905ab6ed",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:26.507887Z",
      "iopub.status.busy": "2022-10-25T06:57:26.507513Z",
      "iopub.status.idle": "2022-10-25T06:57:30.761418Z",
      "shell.execute_reply": "2022-10-25T06:57:30.760495Z"
    },
    "papermill": {
      "duration": 4.270736,
      "end_time": "2022-10-25T06:57:30.764717",
      "exception": false,
      "start_time": "2022-10-25T06:57:26.493981",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "data": {
        "image/png":
        "iVBORw0KGgoAAAANSUhEUgAAA9AAAAKfCAYAAACG1yDIAAAAOXRFWHRTb2Z0d2FyZQB  

        NYXRwbG90bGliIHZlcnNpb24zLjUuMiwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8qNh9FAAAA  

        CXBIWXMAAAAsTAAALEwEAmpwYAAEAAEIEQVR4nOzddXxU17bA8d8Zn4zE3YO7150qFerue  

        uvu7u7u7nLb3ld3o0YpUCwEAiHuybif98eESUICJDQGrO/n08/N2bPPOevMexlmZe+9tqKqKkll  

        YQQQgghhNgwzWAHIIQQQgghhBBCbA4kgRZCCCGEEIIIXpAEmghhBBCCCGEEKIHJIEWQgg  

        hhBBCCCF6QBBoIYQQQgghhBCiBySBFkIIYQQQgghhA32AH8W/vuu6/6+eefD3YYQojNn7Kx  

        DvJ5I4ToI/J5I4QYKBv9vBG9s9mPQDc0NAx2CEKlrYR83gghBop83gghxNC02SfQQgghhBBCC  

        CHEQJAEWgghhBBCCCGE6AFJoIUQQgghhBBCiB6QBFoIIYQQQgghhOiBzb4KtxBCCCGGnpDb  

        jXvxfLwrl6NPTcc6YQrG9MzBDmurpaq7iULcS/5B43JhHX8ZMyFwwc7LCGE20xIAi2EEEEKIPq  

        WqKvUfvU3pdRfH2hJ23YsR9z2JIT1ECPbejnn/saiEw5CDQQAMKRnMu7Vj4gbPmqQlXnCiM2  

        LTOEWQgghRJ/yIzDRduf1ndpafvgKz7LFgxTR1i3s91H+5P2x5BkgUFTNy5wfBjEqIYYPEKcVQ  

        mcC+cNdghCCCHEKBXxeQm7XV3aw27nIEQJln4//oo1XdoDtdWDEI0QQmzeJIHupZCjhYUH74  

        67eMIghyKEEEIMSabsHBJ2ntmpTWMyYy4aOUgRbd309ngyjjutS3vCDrsOQJRCCCLF5kwS6l/zV  

        VQBUvfTkiEichBBCDE1ai43CG+8m9dBj0MRZsE6ezrhX/ivrbQdR8n6zybvsBnQJiRiz8xj58AtYp
      
```



8wY7LCEEGKzo6iqOtgx/CvTp09X586dO2D3a/7xa0pvupxgQz3TflylPiFpwO4thOhXysY6DPTnj
RCbu0ggQLChDq3Njs5mH+xwhpJB+7zx11SjMejRJ6X0+bWFEEPSRj9vRO/ICHQvBWqqMRcM
w1w0ApushRZCCCHWS2MwYmZKkeR5CDFmZERYLIQQ/4JsY9VLgdpqtPGJqBGVYEP9YIcjhB
BCCCGEEGKASALdS/6qCvSJSRAOE6ivHexwhBBCCCGEEIMEJnC3Uv+mkr0icno4uMJ1NYM
djhCCCCGEEIIIIQaIJNC9FKipRpeUjC4+UfZPFEEIIYQQQoitiEzh7qVAXQ36pGQiXg+BehmBFkIIIIY
QQQoithSTQvRAJBAG7W9HFJxB2u6SlmBBCCCGEEJsrWQKdy8E6mvQJSShaLT04hMJNko
CLYQQQgghhBBbC0mgeyFQUx3b01FrtRHxeYn4fYMcIRBCCCGEEKlgSAJdC8EmxvR2u0AKI
qCLiFJtrISQgghhBBCiK2EJNC9EHG70ZjMsWN9YhLB+rpBjEgIIYQQQgghxECRBL0Xwl43GqMx
dqyNT5QRaCGEEIIIIYTSkgC3QthtxuNwRQ71tnjJYEWQgghhBBCiK2EJNC9EPF0HoHW2eNI
CrcQQgghhBBCbCUkge6FkNuJxtQ+AQ21WAm1NA1iREIIIIYQQQgghBkq/J9CKolysKMpiRVEW
KYrpyqloJkVRChVF+V1RIBWKorytKlqhra+x7XhF2+sF/R1fb4RdLjTGDgm01UawWRJoIYQQQg
ghhNga9GsCrShKNnABMF1V1fGAfJgauBt4UFXV4UAzcFrbKacBzW3tD7b1GzliblenKtxai5VQa
/MgRiSEEEIIIIYQQYqAMxBRuHwBWFUUhAHVwB7Ae22vwwc3Pbz7LZj2l6fqSiKMGaX9kjY4
+5mCrck0EIIlRACvt9uJctxvnP3wQdrYMDjtCBZsacC6Yi6dkGZFgCLDDEUIMsn5NoFVVrQTuA
9YQTZxbgb+AFIVVQ23dKoDstp+zgFK2c0Nt/ZP7M8beCLu7TuEOyT/cQgghxIAJ1Ney+s7rmb/
/jyicvRvLzjoW76qVgx2W2EK5ly3mn2P2Z+EHM5l/wE5UPfcoladsMMSQgyi/p7CnUh0VLkQyAl
swL59cN0zFUWZqyJk3Pr6+n97uR7rdgS6tWXA7i+EGHiD9XkjhoieY+6v1LzyDKhq9Pj3n6l95xX
UtuPNmXzeDC1hr4ey+2/BW7IMADUYpOzem3Etmj+4gQkhBIV/T+HeE1ilmqg9qqpB4ANgRyC
hbUo3QA5Q2fZzJZALOPZ6PNC47kVVVX1GVdXpqqpOT01N7edHaBfxuLusgQ47Wrelf7SFEN
0brM8bIUT3nH/90aWt6cv/2yJGBexZzmjJNTfR8tO3Xdp95WWDEI0Yqjo7wR6DbCdoiXbW
uZZwJLgO+Aw9v6nAR81Pbzx23HtL3+rTqEstOwx91pCrFGYACdjoHPYhRCSGEEFsPy9gJXdrs
2+6ELs4CNGILZnWHO91wtQu7cb0zEGIRggxVPT3GujfiRYDmwf803a/Z4ArgUsURVIBdl3z82
2nPA8kt7VfAlzVn/H1Vtjj6TQCDaCz2aWQmBBCCDFA7NvuRMLuM2PHxpx8Mk86E0Wn28BZQ
vSezmqj8Nrb0SUKxtrSjjoR6/jJgxeUEGLQ9fu/Nqqq3gjcUE5zKbBNN319wBH9HdOming7r4GGt
YXEWjBm5w5SVEIIlCTWw5Sdy8gHn8O7ophlwi952AiMGdkbP1GITWCbPJ2JH36Pb/VKdDY75
uGj0Nnsgx2WEGIQyZ9re0iNRIj4/WiMxk7tWquVolxACYGEEANGn5iEfsb2gx2G2EqY8wow5xU
MdhhCiCiFiPaB3iJEf4UvQFFo+3UrrXYCESlbiGEEIIIIYTY4kkC3UNhtxut2dyIXRtnldgqI9BCCC
GEEIIIsaWTBLqHwh5XlwJiAJo4ixQRE0IIIIYQQQoitgCTQPbTuHtBraS2SQAshhBBCCCHE1kAS
6B4Ku9eXQNsIjNcNqKRCCCCGEEIIIIQaSJNA9F05mCysAndVKSNAZACyGEEIIIlcQWTxLoHoq4
3WiMXRNjocVKqEVGoIUQQgghhBBiSycJdA+FPd0nOfqLIVBr6yBEJlQQQgghhBBiEkC3UNhtw
uN0dilXWuxEna0DHxAQgghhBBCCCEGICTQPbTBWinYxAiEkIIIIYQQQggxkCSB7qh1j0BbCLtd
qJHIIeQlhbBBCCCGEEGKgSALdQ+sbgVY0WjTmOMJOWQcthBBCCCGEEFsYsaB7KOL1ohgM3
b6ms9oltbYMBECCCGEEIIIIQaUJNA9FPH7UPTdJ9DRStwtAxxUEEIIIIYQQQogBJQI0D0W8Xj
TrGYHWWq2EWpoHOClihbBBCCCGEEANJEugeivjWn0Br4qyEZCsrlYQQQgghhNiiSQLdQxuew
m2RKdxCCCCGEEIIIsYWTBLqHln7feoulac2SQAshhBBCCCHElk4S6B6KTuHuug80REegg62yBl
oIIYQQQgghmSSQPdQxO9H0eu7fU1rsRJqlgRaCCGEEIIIlbZkkkD3UMTn28AltJWQJEAIIYQQ
QgghxBZNEuge2tAaal0k0EIIIIYQQQgixxZMEuofUgB/NeqtWwWm1tg5wREIIIIYQQQgghBplk0D0
U8fvXX4XbYiUs+0ALIYQQQgghxBZNN9gBbA5UVSX9214BNrpGOCohBBCiP4XbGrEX10Jzmb
HIFsw2OEIMST5qyoInjdHSM/AkJI22OEIIIfqRJED3gBoMgkaLotV2+7rWYiHsdqGGwwMcmRBC
CNF/XEv+4Z/jDmDBATsz/4BdqP/oHSKBWGCHJcSQoUYiNH3zGfMP2oUFB+7MwsP3wvH3H
4MdlhCiH0kC3QMRnxenSfsK3ACKRos2zkLlIeughRBCbBmCrS2svOYCvMVLAAG7W1l+yZm4i
xcPcmRCDB2elctZdu6JhJoaAfCvWU3xeScTqKsZ5MiEEP1FEugeiPjXv4XVWlp7PKHmpgGKS
AghhOhfwfpaXAvndW5UVXxlqwYnICGGIH/FGtR1ZmUEqivx11QNUkRCiP4mCXQPbGgLq7V0
VjvBFkmghRBCbBI09ngMGVld2re29Z0hRwthn3ewwxBDID4ItUub1mpDF584CNGILYEaDhNsa
UYNHQY7FLEekkD3QMTXgxFom11GoIUQQmwxDGkZDLvzkU5/QM488UzixowfxKgGjr+2msrn
H2fBIXuw7Mxjaf1zDqqqDnZYYoiJGz6K3Euua2/QaBh2+0OY8wsHLyix2fKsKKb0litZMHs3Vt5
wKe7ISwc7JNGNfq/CrShKAvaAcMB5QgVOBYuBtoABYDRypmqzoigK8DAwC/AAJ6uqOq/rVQ
dWxOdFs5ERaK3VSRc5cYAiEkIIIfpf4s4zmfxj3JLStEnJRM3ciw6m32ww+p3qqpS+86rID94Ow
C+VStp/eNnJr7/NdZxkwY50jGUaM1xZJ1yDok770GgrgZTTj7m4aMG0yyxGQo0N7L8kjNwL1
oAQ01bL9H6+89MeOtTDKnpgxyd6GggrF6GPhcVdXDFUUXAHHANcA3qqrepSjKVcBVwJXAf
sCltv+2BZ5s+99BtaE9oNfSWmyEZAq3EEKILYiI0RA3cgxxl8cMdigDKlBbTdXzj3dqUwMB3EsX
SQItutBZLNgmTRvsMMRmzrdqZSx5bm9bgXfVckmgh5h+ncKtKEo8sAvwPICqqgFVVVuA2cD
Lbd1eBg5u+3k28loa9RuQoChKZn/G2BM9GoG2WAjKFG4hhBBis6fR69FzBv3bTeZBiEYIsTVY
33LRjS0jFQOvv9dAFwL1wluKovytkMpzqjYgHRVvavb+tQAA/+skg2Udzi/oq1tUEV8PhT9RtZ
AW+0EGxsGKCIhBBCib4T9PtZLFuOcP5dgk/w7BqBPTiX/qIs6tRkys7G0I9Fnsfnz11bjmPcHnh
XFUqhQCDEXDSftyBM7tSXPOgRz0YhBikisT39P4dYBU4HzVVX9XVGU4hIO145RVVVVFKVXV
TkURtKtOBMgLy+vr2Jdr4h/4yPQOpsdf/nqfo9FCDGwBvrzRoiBFGxuouKZh6h69lGIRDPCGse



oh5/HspVN2e5O4h77MO61j2n97ScM6ZnEb7cz5oJh/XpP+bwR/c254C+WnXMCgepKFIOBgq
tuJe3IE9DFWQY7tK2eNs5C/iXXkrjrTFz//l1I7CTs07ZDZ48f7NDEOvo7ga4AKIRV/b3t+D2iCXSt
oiizZqqpWt03Rrmt7vRLI7XB+TltbJ6qqPgM8AzB9+vR+L4kZ8ftR9PoN9tHabDKFW4gt0EB/3gg
xkFwL5L19MOxY2/xYiqeflARdz+21U8b1MVZSNhhVxJ22HXA7imfN6I/BVubWXnNBQSQo1+t
1UCAVbdciXXiVOxTtxnk6AREdz9I2e9gUvY7eLBDERvQr1O4VVWtAcoVRVlbnAmsAT4GDipre
0k4KO2nz8GTISitgNa00z1HjQRnxdFv7Eq3FJETAgHxObFs7KkS1vLD18RamkehGiEEP0p2FCP
e+miLu3+yvJuegsh1mcgqnCfD7zeVoG7FDiFaOL+jqlowFlwJfTfT8luoXVCqLbWJ0yAPFtVMT
nQ7OREWid1U6oVb5wCCGE2HyYutmr1j59B7QyZVCILY4+IQITQRG+1aWd2g3pg16vV4jNSr8n
0Kqqzgemd/PSzG76qsC5/R1Tb0X8vh5N4Q61tqCqKtHtrIUQqoihzTZpGqmHHEP9f98EotMHc
y+6Cq1UmxZii6NPTmH4XY+z9lyjCDsdoCjknHsZljHjBzs0ITYrAzECvdmLVuHe8BRujcGlotUSdr
u63fpCCCCGEGGoMqekU3XQPmSecTtjxlwwDGNWzmCHJYToJ/Hb7MCKj3/EX74aXUli5mEj0
ZrjBjsslITYrkD3QNjn2WgVbgCtLZ5QS5Mk0EIIITYbOpsd2+TuJooJlbZE5vxZn0s3xBC9Ex/7
wO9RVB7MAIN0S8hlanELYQQGgghBBbJEmgeyDs3fg+0ABaezzB5sYBiEgIIYQQGgghxECTB
LoHerKNFYAuPoFgQ/0ARCSEEEIIYQQYqBJAt0DEb+vZyPQVjuB+toBiEgIIYQQGgghxECTBLo
HejoCrY9PkaRaCCGEEIIbZQkkD3QMTnRWMwbrSfNj6BQG3NAEQkhBBCCCGEEGKsQLdA
xG/H0Wv32g/fUIiwQYZgRZCCCGEEKILZEK0D0Q8f16NAKti08kWF83ABEJIYQQGgghhBhokk
D3QCTgQ+IBETFdfALBpoYBiEgIIYQQGgghxECTBLoHlj4/mh4UEdPabIScTiLB4ABEJIYQQGggh
hBBiOl62IFRFD1wNrBLW9MPwFOqqm7x2ala8KMYNr4GWtFo0cXHE2ysx5iRNQCRCSGEEII
YQYKL0ZgX4SmAY80fbf1La2LV7E70Qj3/gaaAB9QhLBBkHLHYQQGgghBBbmh6PQAMzVFW
d1OH4W0VRFvR1QENRplcj0AC6heQCukhMCCGEEIIbY4vRmBDiuKMmztgalorUC470MaWI
RVRQ0GUXqwBhpAZ0+QEWghhBBCCCGE2AL1ZgT6cuA7RVFKAQXIB07pl6iGkljfh6LToyhKj/p
r7fEE62UvaCGE2Fp5Vi6n6dvPcS9ZSPKes7BvtzOG5NTBDkulzYpr0QlaP/+IQH0tKfsfim3adug
slsEOSwghp5Aq6r6jalol4BRbU3Fqqr6+yesoUP1+9AYe7b+GUCfkliurIflixJCCDFU+SrLWLXlQ4
fjLywBo+Ohdc64irzr0DRagc50iE2D66l/7DomFmE3S4A6t59jVGPv0LKfrMHOTIhhOj9NlbTgP
HAZOArVFO7POlhpwr2d7QK+IT04lUFnejxEJIYQYqjzLFsWS57Uqn3oQX8WaQYpliM2Pc+5v
seR5rfJH7iLoaB2kilQqol1vtrF6FRgGzKd97bMKvNL3YQ0dqt/Xoz2g19KnpOKvqujHilQQQgxVa
iTStTESBrWbdiFEtyKBQDdtfhRVHYRohBCis96sgZ40jFXVrevTK+Lr5RTu5FT8NTKFWwghNleB
hjrcSxcRamnCXDQCy6hxDodYY8bjcmMoln/5K24kWPQJ6cQbGyltWUcdxgR7LyBCF2IlcFTsh
RPyTi0JjOWsRMwZmT36nz7jO1RDAbUDol0ztmXootP6ONihRCi93qTQC8CMoDqfopISlr4ft2
uwA2gtdmJ+P2E3S60Fms/RiaEEKKvBeprWXHnHTR/81m0Qatl90Ov4C5ZSuP/fyB9u53JPOY
U4ka06fZ8c34RY1/5kLp3XsW54C/SDjmGpD1nodH3bCtEITZ3jnm/s/iEg4l4PQDEjZ3AmCdfw
5Rb00NrWMZOZMwzb1Hz5ouEmhplPfQY4nfYtZ8iFkKI3ulNAp0CLFEU5Q8gVjxMVDWD+jyql
STi96Mx9HwEWIEUDKlp+KsriRs+auMnCCGEGDLcS/5pT54BwmFKb7mSxJ1n4ilegqd4Cc3ffc
GEtz7HmJH7ZTWsYyZgueFu1GAQTYcaGv6aKvyV5ejiEzAXDEPR9eafYCGGvrDXQ/nDd8eSZw
DPkn9w/PV7rxJo9+L5LD39SOJGj0dns7PqpisovPEeMo4+qVO/YGM93jWr0ZrjMBUOQ2s09d
WjCCHeevXmX++b+iuoloSzi86L0cuRAn5yKv6pcEmghhNjMhFqaurQFqivRp7RvQ+VfsxpV6fL1J
tCB5kZUrxD9anqszTI/LsvOPp5AbTWKwUDB1beRdsTx6OJkWx6x5Yh4PHhWLOvS7q/sXW0Y
x9zfUEMh3lvmx9oqn36I5H0PQp+QCIb7+VKKLZgF7/KloNGQfeaFZJ9xPvrE5H/1DImPEgwSrK
tBY45Dn7T5PocQW7oeV+FWVfWH7v5b+7qiKL/2T4iDK+L396oKN6xNoKWQmBBCbG5MhcN
BUTq1xe+4G855f3Rq6270WA2FaPr+S/45dCZ/7T6ZITdcgrdsFcHmJkquPp9AbXQFIBolsOrm
K/AuX9J/DyLEINAIJZN2yNFd2m2Tp/XqOhqjuWtbnAVFG/29C/v9ID96dzR5BohEqHzqQVz//N
37olcl7+psVI57EX/tMYWFR+xF849fo4ZCgx2WEKlbvd3GakO2yHkzEb+3V1W4AfSjyfhKysHH
NjsWEaPZ9Tjr8RGjxN224v0w4+n9dcfY33s2+9C3PDRXc51L/mHpacpha9sFWowSN3br1DxxH
0EW5rxFndNlmVrK7GIURSF9KNOJO2I40GjQWuLZ9htD2HtZQJtn7E9Wlt8p7a8i65GZ7MB0Zki
LT991+U878qSTQ9+EEUCfsofvYu6915DDQbXrVrJktOOxLVs0WCHJoToRi8uwNoiq3NH/P5N
mMKdgl++GakhxGZHYzCQsu9B2KbMIOJxY0jPJORyMvLhF2j9/WesE6eQsMNu6JOS8VWWR
ysNGwzEjRiNZ0UxrlONVd0Hb5J9xgWY8gvxla3q9JohI2sgH02IAWHKLWDYrQ+Qc/YIKHoDpu
zcLn0iAT+ekmX4qyowZuUQN2J0p3ozlIfjmfDWJzR9/zWhxnoS99gH25RtYq/r7AnYpm1Ly3df
dLquMa+g356rPwVqq6n/6N30jeEw3hXF2MZPHpSYhBDrJxVMNiK6BrqXI9ApqTjWme4nhBBi
82FMb1/frl2zkHrgYaQeeFiszb1sEuT00Tw2Lds6eTrZp5/f5TqG9Ex0iUkMv+txlp55NGGnAxSF
3AuuxDJ6XP8/iBCDQGMwYi4Y1u1rkWCQuvffYOV1F4OqqgJQdOsDZBx5YqelEZYxE7CMmdD
tNbRmM/mXXIdn8QICdTUApB1xArZJvRvpHio0pjgMqemxz501dPb49ZwhhBhMfZIAKxvsvm
J+P293n7EkJqBv6KsnYlSQgxmNRwmOpXn+v0Zdc1fy6K0YhtyY4/277A6qiUHTjPRIuSEkPz
Lp4x/xl69GI5iEedhlKau6zyF2NJ5V62g9KbLo8kzGKqy6uYrsM/YHsul7reH64513EQmfVAn3tU
r0VqsmleNQGe191PU/cuQmkbhTfdSfM4JsffFNmN7LGMnDnJkQoju9GUCfUifXmvl04D3csp
3CmpBBvrfih9aGRLBSGE2KJEf6cf/3Wpb3lh68Z+diLuBctINTaQtzIMehT03EvW4Q+JQ1zfiH
m/MJBifiloSPU1IAaDHZqU4NBgo0NMKJ31zJm5WDMMyunD6Dadv6aKUEsThrQM9Ekpvt4/af



d9mPjeV3hWLkcXn4h1/CSMssxDiCGpxwm0oiiHancDaURHmxVAVVXVTvSHLbLSQcTnQ9H1
LoFWtFoMaRn4ystkKyshhNjCaC1Wkvc/BM/aCsBt4rffGVNmDqbM6Bf6ljK/RLeuqq7EmFfAyP
uexj59u8EIWYghw5CZg9YWjxoOYczOxV9ZjQLVDZIEuLfUSITmn75hxRXNeqyvXtsJCPue6rX0
8k1BgO2KTOwTZnRT5EKIfpKb6pw3wMcpKpqvKqgdLVVbWuT5y1ZxOdF08ttrCBaHMZXVtoP
EQkhxNYt5Gil6ZvPKbnqPMqffBB38dKNn9THUmcfrdl+B0UPtFqyzrgA+4wdYq97V69k2X+OI1
BdCUT3j152zvGyxaHY6pnzCxnz7JtknXou5sLhZJ1yDmOefRNz3uY508NbpuxlZx1HsL42erxyO
cXnnkSg7VglseXpzRTuWIVVN+IbiqloWmAUKmq6gGKohQCbwHJwF/ACaqqBhRFMQKvANO
ARuAoVVVXb8o9+8qmFBEDMKRI4F0tCbQQQvS1hk8+YOW1F8WOq198nAlfv465cPiAxWDOK
2DkfU/hu/AqFJ00U35Rp3oZ/spywi5np3OCDFX4q8o325E2IfpC0NFK5XOP0/z1JwA0ffl/uluXY
Bk9AZ198xuX8ZWvQQ3407X5q8rxV1diaNsOTWixZdnoCLSiKle2Td+eqyjK24qiHLO2ra29Jy4E
OibfdwMPqqo6HGgGTmtrPw1obmt/sK3folr4fJueQK9a0Q8RCSHE1stfW03Z/bd2ags21ONes
nDAY9FaLFhGjyNu+KguxSb1icmg6fxPrGlwoktMHsgQhRhyfKULseR5reavP8G7avPcw1mf3HW
9s9ZIRRefMPDBCCEGRE+mcB/Y9p8N8AB7d2g7YGMnK4qSA+wPPNd2rAB7AO+1dXkZOLjt5
9ltx7S9PrOt/6DZ5Cnc6Zn4JIEWQoi+FymgBgJdm00hQQhm/UzDRIBwxU2d2opuvHu9W/slsb
VQw+Hu24f73BPmYeNlvf8K9sbFIWiWx7AtJIOSRdCbNxGp3CrqnoKgKloLwMXqqa0nacCN
zfg3s8BFxBNAGH6LTtFIVV135SVgDZbT9nA+Vt9w0pitLa1r+hB/fpF+FNqMINbQn0mIX9EJEQ
Qmy9DBIZZJ91EWs6jEjr4ixYxoz/19eOBIOoagStwfivr6U1mkg//nRs2+xloKYKY1YucSNHo2i1/
/raQmzOTIXDsE6ahmvBX7E266RpA7oEoy/pLBayzjyfhF1nEqirxZSbT9yIMQzy+I8Qoh/1Zg30x
LXJM4CqqS2KokzZ0AmKohwA1Kmq+peIKLttUoTdX/dM4EyAvLy8vrpst1SfD80mfJkypKYTqK
8IEghs0gi2EGJoGMjPG7FxiqKQfuQJGfJSqXnzJczDR5F5whlYRo7d4HmRgB/3siX4yldjSMvA
mJVDoK4GrTkOY04+rgVzqXzuMcJuF1mnnk3CTrv/6z1ldXEW7J0n/6triK3L1vB5Y0hKYdgdD
+MpXkKgrgZDWgZxo8aiT+r98gZPyTl8K4rWqxYRo/HkDY4a451Fhv2qdsOyr2FEAOvNwm0RI
GURFVmwEURUnqwfK7AgcpilLMAF24GEgQVEUXdsodA5Q2da/EsgFKhRF0QHxRluJdaKq6j
PAMwDTp09Xe/EMvRb2eTdpBFR6TCKpOFbs0q2shJiMzaQnzeiZwyp6aQfdRlphxyNRqdH0W
x4NZKqqjT8731KLj8bgIKrbmXl1efjLS0BjYbMU8/BV1pCyw9fAVA891dGPfEqKfse1O/PIKRHW
8Pnjb+pEccfc1hz/62EXU60Vht5l1yHPj0bY1JSj6/jmPsb088mljPC4B9mx0Z8eAzsW3keiNQX
0skGMSYkbXRzxMhhOjNp8T9wK+KotyqKMqtWByiW1utl6qqV6uqmqQqagFwNpCtqqrHAD8B
h7d1Own4q03nj9uOaXv9W1VVB/UfENXv36QRAABDVk70C5oQQog+pzUYe/Rl17dmFStvvAw
A+4ztaZnzftncyRC9XOPYZuyTadzqp5/nLDf1+cx7G185Uup+zem2NV6sMuJ2X33YKvdHmp
rxF2u1h9z03ROjUmM4pOh+OPX3AtmNerWEJuF3Xvv8H8/Xfi771msOahO/DXVPXqGkKlR+P
R6BVVX1FUZS5RAuAARyqqquqSTbzvlcBbiqLcBvwNPN/W/jzwqqloK4Amokn3olps4hpoAEN6li
TQQgjRj1RVxb1klc75c1E0WmxTZ2AZNa5Tn2BjPRGPGwDL2lnUffBml+uEnQ4UvR41EkHRatH
ExaEoMhllRF8LtbBefh/XinichFqaen4NlxM1HCL/ypsJntSjMZshEun13suuv/+MzUwBqHjsXvQ
JSWSdek6vriOE2Lr0Zgo3bQnzJiXNqqp+D3zf9nMpsE03fXzAEZty/f4S8fs2eQ2zMTMLT8myP
o5ICCHEWq75c/nnmP1j+7BqrTbGv/EJ1vGTYN00RhOm/CJ8ZaV4VhRjHTeJ1t9+6nQdbXw82
WdciKLVEAkeiN9uZ5nKKUQ/0CckoTGaiHSY4aExmtAndd4OyrtqBc75cwm7XVjHT8Y6fjKKLvq
11ZCcSvqRj7LymgugbaKilSuHkfc/06tYwn79oUtbzZsvknr4cejt8b19NCHEVkk+HWxEdAR6Ex
PorBy8K3s+JUklIUTPqapK9evPxZJniE4Hbfjsw079tHFWmK86i7hRY2md8wPJsW5Gn9pebCjt
qBOJ33Zn6t57jfJH76Hy6YdYcvqRtP7+y0A9ihBbEZWccy6Nze5T9Hpyzrk0lggDeFaWsOi4Aym
59CxKb7iUhUfsTevvP8deD7tdVL34RKdzAlUV+KvLexWJsZv10qbCYWiMpt4+IBBiK9KrEitUcT
vR9nkEegcvKtWoqqqbGcghBB9LRLBX1XRpTIQXdnp2JRXgKpGiBs5hqQ9Z+FZUUzuBVehBvy
YR4zGNnkaDZ/8l0BdTftJ4TBrHr4D25TpaOMs/f0kQmw19KkZNHxzP3L+c3F0yYRGQ8MX/yP
14KNifZxzfYXQcS1yOMyaB29v+320Egn4CTXWd7l2203u0rYh8dvviE7l0BINPHWGE3knHUxW
uO/38pOCLHlkgR6l9SAH80mjkBr7fGgRgg1NaJPTtn4CUllXpM0WrJOPZUHL/93Kk95cDDu/R
LP/IEzAXDcf0zD0NyKs75f5K0+z7Eb7MjGoOBYFOXDR8l1NYQ8fslgRaiD5nzCii67k5KLvsP/qp
yjFm5jLjvKUy5+bE+weYN/T5a0aekkXHCmZQ/dEd7B40Gy7hJXc7bkLhhlxj/2se4Fy8g4vdjGT2
+T/aUF0Js2SSB3oh/MwKtKArGnDw8K5cTLwm0EEL0uYSd92DYnY9S8cT9KDoteRdeg33G9l3
66Sw2knbfG9vUGQSQK9Ha4jFI58Zet0/frss5mSeegT6x59vqCCF6Jn67nzj4328lIntajS0rFuM7
+zfZp6/t9jO4VvXY/eEWrpea15zCkZZB32fVYe5IAA5jzizDnF23agwghtkqSQG+AqqpE/sUINIAX
MxvqvqLit9mhDyMTQggBoI9Px5je4bd9iBoNjiyc9FZbRvsr49P7NJunTiV0c+8Sdm9NxNqbiL
rIH06jGQLITZOVVXcSxfhWb4ErTkOy7hJmHLyuvZqPKZjSE3v5gpgnTSVMc++xep7bybU1EjW
KWeTclDnGrPGjCxyz72M9CNPrgMyo7Ot//deCCH6kiTQG6AGg6BoULTaTb6GIT0T74riPoxKC
CHEWq5F81l03IEk7rlN5uEjcc2fS+LM/bCOHrfxkzvQGk0k7zmL+Bk7EAKEMKSm9VPEQmzZH
H/OYfGJB6MGAgyYh41kzLNvYy7o+SivxmAkaeZ+2LfZkUgohGEDM0E29LuqhkJ4VpUQRKvDk
JGJuXC4VNcXQvxrkBvWl/ZwmotY2YOjr9+660IhBBiy+ZaspDm778i1NJM/l67EmxuxvX3Hy
TsuBv2GT0mVJd89bLpB95As6//6Thkw8AaPziY0Y98iIRr5dQawvG3DxM2e0jYL6qcvxrytDa7
ZiLRql1tVfc1cUnDMhzCrElCnvcVDx6L2mHHYchJRW0Whx//orjr996lUADtP7xCy0/fkOwuYnE



XffEvs2O6BO6zh5ZHzUUov7jd1lx9fmowSAao4mRDz1H8j4H9vaxhBCiE0mgNyDi86lxf/VNY
xZOxjW9FHEQkhxJYj7HGhGExo2vZ2dS9dxKKjZxF2OQGoeu5RCq6+jdq3X6bm1WfJu/R6cs6
+JDaCplbD+CrKslwciz4ljdwLrwJVRQ1HqH3nNSqfeQgiEXSJSYx59i3sU7fFueAvlp5xNMGGOI
AUcs65jOwzzkcne74KAbQtX/O40Zjje1aG/Z5SZy5L9UvP42vrBSApH0Oj03f3F0tf8xh6RnHEH
a2AID75ouMuP8Z0g45aiNntvOWlrDiquNQyEgOihScvnZxI0e1+drntVwGO+qFQQb6zFkZmPO
K+zT6wshhhaZx7IBea8Hzb/cysCQnkmgtpl37/xzkllsRXwV1dR+cITLDx8b0ou+w/OhfMAaP3
tx1jyvFbDJx+QsNPuAFQ8di/+irLYa4pWS+Zxp2PlyCTsaKX84bsof+RuWn/9EVAhEgEg1NzEim
suxFddSenNV0STZwBVpeLxe3EvXtj/Dy3EZsC7aiVI993CwsP2ZNVt1+ApWdar83VWO855v8e
SZ4CmL/4H6gZO6oZz3u+x5Hmtymcfxl9bs54zugrUVseS57XCLmf7738fiQQC1P33LeYfsBOLj
tmfBQfuQvNP3/TpPYQQ4sk0BsQ9njQGE0b77gBik6HIS2z0z8mQgixtVJDIapefILVt12NZ9liGj
5+l8XHH4SntISwx9ulf9jrQWOMzgSKBANE1vICbJu2Dao/QOtvP8XanPN+J+x2ou9QoMi7fCnB
hjpc8+d2uYe/uute0kJsBUktLay48lwqn3wAz/KIVL/0JEvPPr7z/ugbu4ajBcefv3Zp95Wv6lUsY
W/Xz4KI10vY5+nxNQwZWV12UdHa4tGnZvQqlo3xriyOjnS3rfk0Ox0sv/gMfG17SwshtjySQG9A
xOf91yPQAMasbDwrl/dBREllsXnzV1dQ/crTndrCLifupYul335nWGfKaMq+B9Hy87cApB1+fJdq
voakFJwLuibF7sULIRs+KnZsHjUWfWo61inTu/Q1ZuVs8vMIsaXwrl6JY27n5NdXWoKntOfL0H
TxISTsvHuXdsuorkX9Qi4nvoo1hNaZdQJgmzK9y7Tv9GNOxpSV26Xv+piLRjDyvqjS/G0NjsjH3
oWc15Bj6/RE/6a6thsl7VCTY0EG+v79D5CiKFD1kBVQNjRQTH8+wTakJ6Ft1QSaCGEQKNFYzQ
RbhutiTUBDNgmTmX8ax9T8fRDBJsayDzudALNjZiHjSR53wNJ2f8wNB0+k4PNTXhKI5N60BH
4K9bgXtl+Fdu+zY7Uf/QOAPqUVEbc8QimjCyKbrqPZWceQ6C2GjQacs+/EsvY3u8dK8SWRtHru
23X6Hv+VVGj15N1+gU45/8V24Ek/ZhTsE2Z0amfa9F8Vt1xHe4IC7GMnUjhNbdhHT859rpl3CR
GPf4K1a8+S6i5ibTDjsU+Y0c064mx2+fRakne/xAsYycSaKzDkJ7ZL2uTjRmZ0T/8dUiidYIJ6JN
T+/xeQoihQRL0DyH4vf96CjdE94L2IMhWVklIYcrOJe/ia1l1y5WxNmNuPpYx41F00sxFl8m94G
rQKGj0BiJ//0nCTrtjzMXf7FAF1bXkH1ZcfR7uf+aDVkv6MadgyMii+dvPsU6ZQdrhx5E6+whCLS
0Yc/MxtY0y2yZMYelH3+Arl0Nnj8dcNLxTUI7E1spcMly0w4+j7r3XY20Ju+6NedioDZZVlWXkG
EY9/gq+VSvozHHEjRjdab9nf00VNW+8QML2u2AZNRZ9Ugo1b7xA7gVXYczlAkDRaGn941fM
hcPRTUmk+YdvojNUekIRFMxwFwzEXDe/1uT1IHjaK4bc/zMobLkENBtFarlx84FIM2T0fLRdCbF4
kgd6AiM/zr7exguj0wJZfvuuDilQQYvOXesjRmHILaP7pG8yFw0nYaQ9MOfn4Ktaw8vpLSNhxV
7T2eGrffAnXgr9i5+VfdQsZJ56Fv6KM8sfvjSbPAOEwta89x7C7H8M6fhJaazxxRSPWe39jZjbG
zOx+fkohNi/aOAt5l15Pws574PjzV6yTphG/3U5dto5yLviL2ndfw1++mvSjTiJ+x13Rx7f3cf7zN0
tOPISw24kaDmPfdidG3PdkbPq1v7qSsNPJmrduj52Tsv+h+KurYgl0628/Uf38o5iLRqC12nAvW
UjVi+kMv+2hTIO7vatX4i1dgTbOQtzIMeiTkvvzLeqWxmAg7bBjsU6dQaihIVqFu5dbdkgkNi+SQ
G9AtlhYH0zhzsrBu2olqqqjKEofRCaEEjsvfXwCSTP3JWnmvp3aXUsXk3HMyajhMMGG+k7JM
ODFo/dgGT0eb1kpzj9+6XJdz7LF1L79CtlnnN+v8QuxpTKmZ5J64OGkHnh4t6+7ly5i0bEHEPFG
i3m1/PQtw25/mlxjTgYg7PdR8+ZLFNxf4pGg6LV4a+twjl/biyBjvi8sT3b12r45APSjz0lduxdvZ
KCa27DtWg+oZZmci+6BteiBfhqqjC31UFwzp/L4pMOjVXrTpy5H8NuexBjemafvic9oeh0WEaM
gfX/3U4IsQWRlmlbEPF6UPT/PoHWWW1oDMZeVblUQojNjb+ulpY5P9Dyy/fRNcbdiASDBOPrC
fs6V9kN1NXS8t3neJYvZcUV5xBsq01ybtjxrnGLzRmM5Zxk7u8rk9IQmM2k7zPQX3yPEJsaXy
V5TT/9A2tv/9CoKmh1+c7//k7ljyvVf74vbGCWRGXk6SZ+1H31kuUXHwGJZeeib98DTqLvf2E9
e0t3aHdMml0a+6/jYaP36Plx29Yc98t2KbMQNFoAQi73ZTdd0unra6av/msyx/d+oqqqgTq67ot
eCaE2PplAr0BEZ+3yxYIm8qYnYtXKnELibZQ3lUrWHLKoSw+/iAWnzCbRSccejKe0pFMfz4piVl5
7EfNn7UDx+afgWrQg9lrrbz9R++6rRPw+wm4XKAqaOEun8xN2mYnj1x8hGCJx5j7oU9qL9CTtf
QBxYycw+rFX0CeldBujGgoRqK8j7Pf14ZMLsXlW13EP0fuw5KTDmXRMbMovuA0fBVr/vV1FU
UDa2fXGYw0fv5RbCsrNRik5pWnO1WkjiscjmmmdKc6mgqJOyy68pSVE1vk9Rv7ZVCjG0qHXA5
cixZgSM8k7bBjSZq5H4pej38Tt44KOR0Emxq7fc1fWc6ah+5k/gE7sfjEg2mZ8yNqOLxJ9xFCbB
lkCvcGhL1eNMa+SaANGVI4Vy4nYYdd++R6QggxGEJuJ+5FC/GtWYU+LR3r+MkYkINp/OoTPE
sXxpf5S5bR8MkH5J0fLRYWbGmm5PKzYyNEzd98hmv+n0z88HtM2bk0//g1EC0olnvR1aAojH
zwWapfegpPyVKS9z4QXUli9R+9i2vRfOo/fJuMY09Ba7VhHjEanT2B0hsvxbdqJeYRoxn9xKvED
RsZi8dTWkl1S0/R+MX/sE2eQe75V2AdL9W3xdYhEgxS9eKTBKorY220Od/j+HN0l63hNsSUX
0TWGRegtVhQqyE0RhO65FR0idG1x6GWZlp/+b7LeZ6V7YVUDWkZjH7ydSqeul/WOT8Qv8Ou5
JxzWadCY9DNcJeNjvadTJeYTn4l16K12ghUV6KJiyPlwMPQ2BN6/CwQnXLe+tN3ID14O2Gng
6wzzidl1sEY2ipoq+EwVa8+Q9UzjwAqrK9lycmHMPH9r7F0mNKrewkhthySQG9AxOPus+qsxo
wsPCukErcQYvOIRiLUvft6pwwraqYccTdHN9+H4veua5NY5P0JbAu0rXx1Lns0jRpM6+0giPi+Ou
b+h6PVYJ05FozfQ+NnHtPzwFRctH5F3wdUoBgOe0hKcC+eRdeo5lN13CxGvh6rnHweixcWW3
3Mq+ZfdQNK9N+EtWUB9x++Sf/G1AAQdray85klcbeumm776Pxx//cqk/36HKTe//94wlyaliNe
N469fu7S7ly6CQ3p+nbDHRfN3X+JdsSzaNOuy/PaHCdbXYkjLwJCWHt02qqaq03nm/M4jzpZ
RYxlx75MEW1vQxyd0+a5lm7YtGqOp0yh01unnYUHJAYDY3lghNZ3SGy8j2FAHQNLm/cg4+aye
Pwzgmj+XpWceHTtedeNlaPR6Mo4+GYBAXTU1rz3f6Rw1FMK9fKkk0EJsxSSB3oBwXybQWT



m0/CyVuIUQmy9fWSlId9/Yqa3+v2+RcexpJ001i+bvvuj0WvJ+s2M/a4wm0GrRWmykHXI0Zffe
HJuOad9hNwqv50Woy0nt2y/HzglUVdAy53u0Njt177zC2Fc+ZOW1F5F5whkoBgManZ6mbz4j4
vNCJEKouQmNyUzE56X1l+9RL7gKRavFX746ljyvFWpqxFtaIgm02CpobfEk730glU8/1KndNnW
bXl0nWFvdnjwDhMNUV/w0CbvuGX29oY7EPfbFXDQcrTkORasj2NKI0k1BVo3BiLHTqHO7+Bn
bM/alD6j76G2C9bWkzj6KhJ1ntt/W7aLm1ediyTNA0zefkbjnrF49T3ffy6pfeqU/Q9DZ4vWr9En
JuP3uDv10a6zvEQIsXWRBHoDwh4P+uS+2RLBmJWLR9ZACyE2Y2GPu8u6RICQ20HCzjPJJu/R
61jx0B4TDpBx4GPHb7Yx39UoMGVmYC4aR85+LCbU0U/3SU7HkGaJTSYPNjQSbm7pc2/n3n
9imTEcXn4A5r5CCK25ixZXnEWptBkUh88SzUPTRaZ2K0YgaCgKQvM9BKNpowSGN0YSi06GG
Qp2urTHH9dl7l8RQpigK6UediGvRALp/+Q40GrJ00Qf79016dZ2I14s+NZ3ssy5EZ0+g4ZP/4lr4
F0rblGtFp0PRamj8/H/4K8oASNrnIGxTtb200X7bHYnfdsfun0ejwbVkyZf2jmute6K7egn61HQU
ffTrsT45lYJrb6f4nBNir5sKh2MdP7lX9xFCbFkkgd6AiNeDxpDVJ9fSp6YRdrQQcjrQ2ewbP0EIIY
YYY3YelgmT2/dfBliPPRY1GKTsgdsl1FYz5uk3MGTmEKytZtExswi1NJO872zyL7+BrNP0w108
hJo3Xuhy7VB9LbaJU1i3drd92ra4/vmbgqtvRzGWX3PTdHkGUBVqX75KfluuQ5DWgaoKmo
RNJeB5C8b3sblN+EdlnX0rFo3fH2hJ334e4EaP78u0RYkgzFwxj9BOv4FuzCkVvwFxQ1GWWn
b+2Gsefv+L863esEyZj326n2PZTANYZ21OYmk7tmy8SbKwnedYhZJ50JvrU6NRqdDpafvmeU
HMjiXvsS9jpoOnL/5G01/59+iyMrBzid9yVps8+6tQeN7x3v9PxO+yGLIGRUEvbZ4pWS+45l6l1m
WN9Enffm/Fv47rn3nok1KxTZkhM1eE2MpJar0BEa8nOu2wDygaDcacfLwrl20bPL1PrimEEA
NjN5DliHufouzem2n+/kuyz7oYfUli5Q/cjj45leSZ+7LyuosZduejLD21fR/Zxs8+RGu3M+yWB7B
OmkryfrNp/PRD9ClphJ00lqEg5lFj0Wi1ZBx3WjTBVlWsE6eRuNteJ021Pwm77kmguhLfOpW9
AXQJSYx95UPUSJjkfWdjLihEa7HFxtfo9WSd/B9sk6fjXvoP5oJh2KbMQJ+YNCdvmxBdhc5mx
zqu++J5lY+HNQ/eQd07r8TaEnffhxEPPI0+PhGAcEsZJZeeiRqMzvTwLF9KzvIXkNg2vVpRozM
+sv9zES1zfkSXkEDRjfd0mmrdFzQG13kXXlM3tArv8RLQask+4wLs2+zQq+tYRo1hwtuf4fjrd8J
uN7Zp22JbZ22z1mgifsb2xM/Yvi8fQqixGZMEegPCXg+abtbtbCpjVi6ekmWSQAshNluWkWMY
9ciLBBpqafz0l1bfcW3stdY/fib3gqtw/PELhswclh53bLS44eP3SJ41G63BTOap55K832ycf/+JIT
Ud29RtaP7qEyqeeADLpGmMu09Jwm4Pjrm/svKGS9EnJhO/7U7ok1lwjxob/cLcMaYx47GMH
LPBuPWJSSStvjdu+/d92+KEJuBSDCI65+/af31B7QWG/Hb7YxI9LjY6/7VKzslzwDN332Bd8Vy
9N02BcC9ZGEseV6r9q2XST3ka0IKhqGLT8Q+fQdCLU3EFY1AG2ch1NqCZFk0TueE3G5cC/7C
X7kGY3Ye1knT0F16t67YmMoc49/4P/xlq9CY4zAXDkezCVuPxo0YQ9yIDX9+CCFER5JAb0DE6
0HpoyJiAMasbDzLI2y8oxBCDGFasxnvyhKqX3yiU7saCBB2u9DGJ5B/9c0oqol2lYmw20Hzd19
S9/ZrOof+Suap51B21w2x80yFw0ncdU8iXg/O337C+dtP5F54FY2ff4wa8KMxGFHDlQyp6Yy48
1GWnnUswfpaFJ2O/CtvJm70+IF+C4TY7Dh+/5nFJx8KkQgQLSw24a1PsYyJ/v5EgoFuz+vYru
2mboDWYoklriGXk4jXTdk9N8bqH0gSkxh2200x/mG/n+oXHqfmzReJGz4az4pLZBxzCllnXoh2
nUELd8ky1GAAC9FItKauMwINickYEvumVo0QqvSUJNABEPF5+3YEOjuP1t9/7rPrCSHEYDPDXV
tP09SfdFuFS9HrMeYwsuulyQq0tpBx4Glm774MaCJC4xz4YMzKpeOKBTuf4Vq1AN/ulTm30v
//EMnocroXzyD7rwtgesbbJ05n04Xf4K9egtSdER530+v57WCG2AGGfl/LH74slzwBhZystc76P
JdCm/CJsU7fFOe/3WB9T4XDiiKbEjs0jRqNPTiHY2BBryzn7UhRdNIEOu5zUvfd6pyKBoeYmfGt
WxY7di+cTCQRlrmkrfrn/+JmnmfkQCAdyLF2Bvqwrur6+l6etPqXj4LkKtLaQefizpR52MrR/2bvcs
X4pj7m+EPS5s07bDNmEKik6+Hgsh1k8+ITYg7PX22TZWAMbsXLxSiVslsZITQyEaP/uYrFPPY
c39t8badYlJxM/YkWXnHE/Y6cCUX4QhLYowy4F92rY0f/cFSXvOlVLKM10v2uGLPUS/zHvVIQy
7/SGS1tmaxpiZjTEzu1+eTYgtkRoKEWRpWuU+5GiN/axPSKTwxnuo/+B1Wn7+Htu0bUk/+uRog
b423rJVZJ5wJiFnKyFHK+aiETR99wWJu+3VdsEgYZezy30igfZR7LDfj+PvP7COHkfCznsQ8Xp
w/P0H9h12ifVx/v0npddeFDuuff0FtHHWPk+g3cVLYsUOAdBqGffKhyRsv8uGTxRCbNUkgd6A6
Ah03xQRAzCkPRNqbiTkccQKz2jZ+ghBCDCJHcxOrly7G1dpCzrAR5l2MvrvJJSaTcuBhNH/7OfIX
3oxn+VL0KWkk7rYXrb/9RNjpwDpxKvE77kbV848T8fswFQ4j87jTMGRkk3Lg4dT/963YfTTmO
JQOn7X65BTSjzoRc34RWot1wJ9biC2Nzmoj65SzWXHV+dEt3SIRUFUSdtw91ifkcLDm/lvwla3
CMmY8zn/4F40n7EvfYAhJVplW/W6WfPQHWh8WitVurefQ19ShpqJDriE1Kif3YUzot0UCjwd
oh8VVQSNp9byqeuJ9Qcx06xCryzr4kthUWgHvBX12eoeGTD0g/6iTiiob32fvS0ueH9uQZlBym
4vH7sE2Z0akStxBcDcQJ9AZeFf6UPpzCrWi0mPKK8CxdhF2qOQohhrDm+jqeuPpivnnvTQBM
cXE8/tFXJHjcuJctlm7kGMzDRIL94hNYJ88gedbBNH31SWxf1aS99u80Ou1btZL6j98zRRH+tE
nobXZafnha4w5eaQefBSW8ZMx5xeBAPbR4zEX9t2XZCEEBItGUvTliwRXLEMxGNCNHlvbaCK
+7XVv2UpafvwGAF9Zaew878qSWAJtzMkDrZaws5WwMzp6nXrl0ahtU7aVUBBT4Qjyr7iJuvde
R5eUQsZxp8ZeB1A0CuUP3xUbqQ41N1H+yN2Mebb9j2r6vt1ZMzMRmfV221Ag02NXdsa6lCD
lei78RMhxBZGEugNiPj6dgo3gCm/ENfIBZJACyGGtJIF82LJM8DwMeNxffAGVW+3V+INPuBQ0
o45BSUUIlBXQ6i5Ea3FgqmgiljP2+WaroXzSNxjP9SCIII0OJSk3fZCMcVhLhyGMT0T66ixA/JsQ
mxt/M5W3KtW0njdhbEq2vqUNJJvFRamTQVAUTTdnqto2tt91VUUXn0rDZ/8l2BjA0l7zSLschL
xeYDo7BTn/D8xxCeSc86lQOEw7sULsE1v/84TcjQ6TPMOu5ydpNbJ0/HIFcYWzutGAzk/OfiW
CIP0WnprsULcC9dhM5mwzJhKua8gl69Lwk77U7F4/d2ass8+T/obDJLUaifpJAR4caCqGGQih



9XJzGIF+Iq5upSUIIMZQ01dZ0Op617/44nrq/U1vj/31A4q57U3bPjehT08k9+xKKLzqNrNPOJa6
bbaXMhcMJe92Y8osIO1rRZuVgysrt1+cQqODf5cL73zc7bUEVbKgtPQf2OcAAMyFRdE92j/7K
NbHNn17TMPai4jplFZWXXHkuCTvtjimvkPqP3sE+Y4dYkh12ODdb42mZ8z0tP36DxmQm49hT
UX2+2DWMGVkoBgNqh3XRisHQqa6BfcoMRj7yAu6li4j4fcQNG4ltu507PVPLL9+z5LQjYvUTTI
XDGPvi+5jzCnv8vlgnT2PMs2+z5qHbCTkcZJ9xPkI77d/j84UQW6d+TaAVRckFXgHSARV4RIXV
hxVFSQLeBgqA1cCRqqo2K4qiAA8DswAPcLKqqvP6M8b1Cbetf46G1HfMhcNp/u7LP2mEEL0
Jc+qlyZx7nvuptp1Oh46qlHSlpPIKiqxO+0Gyn7HUKgvhZFUVCMBnTxCQTra6l87IGG3/kolc88
TLChnvRjTqb2zZcA0MRZyDztXOJGjGHP2cfjWTQ/ur3N7Q+TNHO/LpW0w24XnhXFhBytmPIL
e/WIWAjRmVGnI9JQ16Vd43TEftZabBRcezuJu++De+kizMNHkrDjbhjalUA6JNTyTrlHGpee46
Iz4t9xvYk7rFvp1oFla+X9GNOIXHxvVAMRgyZ2YRa2wuYWUaPp/DaOyi9+Ypo8qvVUnjtHVjW2
Y70NnEqtoITu32eYGsrg++5sVPxQd+qlbgW/NWrwzqt0UTSzH2xb7MjaiilPjGpx+cKlbZe/T0CH
QluVVV1nqloNuAvRVG+Ak4GvIFV9S5FUa4CrgKuBPYDRrT9ty3wZNV/Dril14umHwplGHPy8V
Wslez1dLufohBCDCbXkoUsO/t4dMNHozWbSVy0gEcff57QmlUol9+lotXS/P2XmleNJOLZ4i5e
QtEt9xNqaiTQUEck4Cf/yItwzp+Lp3gx+ZffhNZmwzxsFKbClpadeRyeRfOB6NrH4vNPZtLHP2Id
0/7lOdjSTPnDd1H98lMAa00JhH3hHexTB+WfAyE2e1pTHOIHnsDqO67r1J647U6djhVFgxqJEG
xqwBwe1mVatxrwo7Xhk3nqOWi0WsJuF2GXX7aNIc6egHXUGJZfcEpstNuYV8Dwux6PXcNbv
praD94i94lrUYNBFL2e2g/elN7nmcQVFPXoeVS/j0BtdZf2TgXBekGmbAsheqNfE2hVVauB6rafn
YqiLAWygdNAbm3dXga+J5pAzWZeUaPVJn5TFCVBUZTMtusMqljPg8bU9xUkNH09ptx8PMVL
sE2e3ufXF0KI3gr7fQSQKIAMBhzFS3EedTLvvvMmbreLQ088DeN3X5Cx6540fPg2TV/8L3rSV5
9gm74dhtQMgnW1rLzuoti6Rn16JvX30CvtITKZx4m1NqMefgoRtz/NO5Ff69z8C+stJOcbR
78YJY8gwQdrRQetMVjH/tl3T2hP5+O4TY4qjhMCgaMk85m4b/vYfWaiPt8ONja5schOutj1e3X0
PjphwA0fPQO9m13YvQT6BPTAYgUFNFsKGWxD32A1RCzU2svudGUvY7CljHX1dD1UtPd5oq
7l+zGk/JMhK2iybr3tIS3AvmonrdxI0ai6d4CZ7IS/GuL05xAq1PTSPjmFOoePy+9kZFwTJ24r96
n4QqoicGbA20oigFWBTgdyC9Q1JcQ3SKN0ST6/I0p1W0tQ14Ah32ePq8gNha5qlROOb+Kgm
0EGLQeMtK8VdVoDGAfj0v1S/8ARaqw3dtdfx6zWXxf09fP0VPPnGh/hKlrUnz22cc38j98KrCD
TUxpJnU34RmSecTu07rxJqaST92FPwrS6l5cev0ZjM6JKSCa1T+VafnNrp2F9b1SVe96L5hFpb
JIEWYhMEmxuoeslknbbm7yLrwVFQ/MPXxGoqSLtoCMACc9bHEue13L8/jOuruQtI3HkPAOI
TEVrsdLy3ReEWpqxjJvlqldeQJ8a/RoXCQQJVJazL9DbexnjdFM3mU34CleJGvBPkyTppJy0BF
ojJ1n/YXdTjwrlhPx+TAXDseQlh57TVEU0o85GSIRql9/AUNKKgVX34p1wpS+eLuEEGKDBiSBV
hTFCrwPXKSqqqPjumJVVVVFUdT1ntz99c4EzgTly8vry1Bj+noLq44sYyfQ8u03ZJ9+fq/PXVNS
TN6lUf0QIRCIOWPxeTPQWub8wLKzjyfsdKDodGSdei6WsRMJ1NXw14/fdenvKCvFnpJCzrmXg
QqqGqHuvdcJ1teCRhOtcNEM/eiTWHXHdbG1iZUrisn+z8Wkn3A6xsxsht/xCMvOPRHcyQAYT
z4by6hxn5nys7vEoN9253QtY2CCbGl6q/PG21iMrkXXEXFw3dS9fxjACTuthcJu+4Z6xP2uLs9t
2N7uLWZ+v97H9vk6WhtdnxIq0Cnjya4Kan41QjGvWbhf/lp9KnpRHw+ws5W11JhsWtojEbgP34
X7/KIAPjWrMK8bAn2bXaI9QnU1VJ2703Uvf8GAkACIkY/+TqWDPX6TVm55F16PRknnIHGZEK
fIOuXhRADo98TaEVR9EST59dVVF2grbl27dRsRVeygbWVLSqBjiVZc9raOIFV9RngGYDp06f3Kv
nuqYjXg6bfEuiJVD7zCJFAAI3B0OPzfvjoPW4+6UhuevlddpI9WL/EJoTobCA+bwaSv7aakkvPlt
xWPEgNhah85mEKrruD8kfuwRjNaCYU85kfF4epsIRNLgcJHq81D/3GBG/D02chbwLrqLpu8+
xjJmAd+VyALRWG4G6mk6FfQAAp/2Q7HMuZdWd15E2+yjGPPMWweZGjFm5WMd07LK3q2
XcRAquvpWy+25BDQYxFRReP1d6KyyTIFs2ftr88YfoHXO93hLS2JNzd9/ReKes2LHhVRMbNO
3J1BVgX700Ilrl6PR6zFm5cT6hByt6OMTKH/oTiBa1yV75FgC9bUYUIJRdDr02+2KbtR45v02B5
vdzpQddsHdoR5r2OWMJc9reUuWEna2b23l/PsPTIXDGXbHw6iBABqLheo3XqDo2js6fW9SNB
qMGVI99jYJIURP9HcVbgV4HliquoDHV76GDgJuKvtfz/qOH6eoihVES0e1joY658Bwt5oFe7+o
LPaMGbl4Jw/l/gOf3HdmE9efo7RU7fhy7dfIQRaCLFJgo0N3RffscVjmDCF0enpnHHhZWxnt2G
cti1VixcyOieP8svOiq1rjHjclD9+L6OffJ1wwlvWaifrtPOo++9baExdiyNqLVZc8+diSEph0dH7xdq
zz7wA64TJXfrrrDayTjmHxN33lex0YszN67T/qxCid9RwEMfvP3dpdy9Z2H6g1WK54Cr++Oxjfv
vhW6Zssz17HXwEdCgkpv91H/0buzYX1FG0zefYt9hFWBMegONJlaaV69iSnY2Ea2GuT9+y4y9
2n/v0XX/1bPjtqGK0UjD/32AZ9kiADTmOEY88CxxZyuadZZ8CCHEQOvvEegdgROAfxRFmd/W
dg3RxPkdRVFOA8qAI9te+5ToFIYriG5jdUo/x7deYbez39ZAA1jGTKDI5297nECHw2EW/z6Hs26
5h1fvvXf4hJCbHn4TCVK0twNjeRIZKCIOTobQHWHyT2KwqrqajJPPx/9vN/YZ9ZBRFqaqXnt
ObRrVmG+6Bqyz7yQkKOVQF0N5qlReFcU0/rbj1gnTiXs9ZB0wGFYJ0/DkJJG7ZsvdKqGmzr7S
EJuJ1XPPNlprspnHiF5v40xTZrWJWZFPyNuuCxXEalvKHOD8dvtjG91aad267hJsZ9dDgeP3HY
9i36fgzU+gf8uWcSfP//A7U+8wNq5H8HGBizJ500xz6okQiBmiqavv0CnSXaw9HajDXGr/v6M
4TcLgDGjB5HePv2at9ai4WkmfvR9M1nsbakmfuhjWvfCstXXoZn2SLixk9Ga7Xh/vtPqI95Cvs6e
0EHmxrwa1CEXeHuXB4v35vE0Kltf7CvfpwPo2Up7ZTX8VOLc/Y+qpsMuJJq7/tpmyz9ieymc
eJu+ia1A0mo32X108FHTSMrkjRuFxOWItaiQ+SdYDCiE2zO/18uXbr/LYIRcS9PvJHT6Su+56gD
WXn0OotRlRf8f2n0v4tbYGGQ3Mj9unbYzDGEbGEyTnlHNxLFuL4/WcMaRm4liwk9cDDWXH1+V



hGjSPtsGOB6B8cXfN+x1QwDEN2LoU33ot70XxQogXC6j58i6SZ+xHx+7Et6nbzgghek7RaEg8
+Gicf0e3lwNI2v9Q9CNGx/rUNTeRk5nFmfc+jL6hjlBSCT8vmE91XS3ZbX3iRo3Bunoi5Y/dC5EI5
uGjKLjQfJC3FQBTQf30Q0y7742/aCTacATlx68wd9iD0IBXiy4xifyrb8VXUYYxOw9fyTL8de0zY0
JOB9orb+X/fv4Bh9PJ7qddgLm8IlijGeLjAXAvX0rxhafiLV4CGg3ZZ11E9unnd7uXsxoOo2i1ffyu
CiG2VgNWhXtze3Y6up2K2FfMw0ai6PS0/vYTCTvsutH+JQvmkTNsBlqikJaTR8WK5cRvs32/x
SeEGLocTY0s/esPSHf/Q3bRcMb02I6UzPZ1gH6vl/KSYhwtTej0BuZ88hHn3f0w5jgrPo+LSp2B
zOfelVhfQ6XbQ0ljPRPHTqJ08UIMIVXkDB+Bu7kRcdDRkIS3g9ex7N0MXmXXEfN6y8w7pUPK
b3hUIZedXGpBx1B1QtPAGDIzGb43Y9TcvHpaK02kvY+AN+qIXiWLCy2aTrG3AL85atjcWriLJjyC
gb43RNiK2Qy0/jbT5h0/g+OQBC93gCNdbjLy0ieHv0uYY2zMDsnm/o7ro2dttu+B2GMT4gdh30
+6v/3Pqmzj0SXkEjL91/R900XmCdNxxifgFmnRTdrNis8XqrLVmEwmSg/+hQSdO3JqzYpBd3h
J9LobKXRbCM5Mwv75G3Qdqgn6xg5jnBTA7vuuz8Bn5/E9Axas3PQp2VG4/D7WfPo3Rh33A3
tWZegDQVpfv057NvsQNKue8Wu07JklY2f/Bf37z9j3XkmyfvOJn7UmP56l4UQWwlJoNcj5GhF
azZvOMmUhsFxF33pOb153uUQK9cvJD03Ghl2tTMBcPxljBOEmghtjrBYJD3n3qEV+9pX8qx0
/4Hc9ljz2FPTKJs+VK8LhcVK5bz+1ef4WxuYp9jTmL5gnm8/ci9AGg0Gi599Fm0Gi2PXXUhVz3
5EtcfdwjhUAgAe2IS1z37Gg3z/uB/c35in9MvJOPvP3AvX0LGcacRaGgg89RzqHjsPuJGjgWtFs
JhAtWVeEuWgUZD20Wk/oM3Sd73IPKvuAl9ehapBx/Jqtuvxb1oPsa8Akbc9RjmwuGD8j4KsTx
x1tXgycrl6y8+ZfylkTgjKj+sXM7szPYCYfEKLH3jhU7nNX/+MaOOPjl27G9qJPf8K6l96yWCjUkz
zoYnT0eRW0rHKgzUKloeOzKC4m0FRNMSs/gqidefJ0jkJbBXx+/zxM3XBlrO+eWu9npPbaLh
q9no9efZGs7Bzsdjtvv/gU51x3067mJulzMgk1NxKedRiVLc3ULFuC0RxH3ukX0ur2sHb82VVZ
zqrLz8azNLQ02jnvD9xzf2X4w89jTkrpi7dVCLGV2vjc4a1UyNGKxtx/I9AACTvtTuuvP+FaNH+jf
UsX/0NG20hNQmoateVI/RqbEGJoqlxZwhsP3Nmp7edPPqS8pJh3HnuAS/bfncW//8KjV16Aq
qqk5eShM+gxmc3857b72P/kM7Anp+BubeW9Jx/i7ufe4P2nH40IzwCO5iYW/zGHpP++wUxX
Mz6Ph/iddidYX0fp9RdT/95r6OyJpM4+kmBrCxnHnho7NxlMdJoq2fj5xzm/kbiTrsTv+10jH/1l
6Z89ScT3/ua+HXWMwoh+ok5Dq/Xy55eBxk/fU3Wd59yQHISHm3718CAxw1q18LFAxd7dWx
DUjJl99ylr6yUsMtJ3TuvEvH58Wuj4zEOr4ePnnsyljwDNNXWsKa4vep2U2UFT9/SPsoN8PQt19
JcVRE7bqmr5dj9ZrFj8ULGff0/ztttD6qWLSYUDERf9wep93p47KoLefmum3nmxit5/+lHaW7bR
QDAU7w4ljyv5fjle1zLFvfrRNCiHvJARoEYUcr2n5OoLUWK2IHHs+Kay9CbdsTdX3KS4pJz4uO
QCekpFG9ZnW/xiaEGJoCPm+nZBcgzmZjzfJlIBUvYcZe+xEKhn6wispmf8Xv3z6ESv+WUB9T
RVPXXcZSSlPHH3B5fz5zRdM331vtEYjrQ31Xe7T0tCAMSsH7x+/kKXXUXrdxTh+/5mw20Xzd1
+w5t6bsE6cStNX/4d+bT0GrRbLmPHed5hVY8jMjMx/Q9G1rVvUxScQN2wkhhsppCvEQPH5v
CS5HDRvuwsfWhL4Nnc47oJhplTbP0u09njiOuyZDNHfX0Ni+2itr7ysS5Ld8OkH0FbflBJRaWns
+nniaW2J/dzU2NDIMywcCtHY0BA7TtUouO+7hUBZKaGWZlxvVEco4lQ233cbgcfP8k7tbW2
DI///gtdRXIsWOldeAdWnW0y6EED0lCfR6hBytaDr8JbO/JO6yJ4qiUHLrLajd/OUXousZm+trSU
qPrv1JSE2ldo2MQAuxNcoqGNZI+caJV9zAmuXL+PPrzykvXkpKRibP3nQV5SuW01xXy6v33lrV
Fs9uhxxB8fy/eOr6aAL91sP38OSdN7HfcSd3uc+k7XfCtXAEAGpDHcEORYAAvKUIRLxunHN/Q
2MwYp0yg8Lr76LiqQdJO/RyCm+4m9zzryD1kKOJGz1OquMKMYiMGi3IKNx6/RX8+Pn/8el7b
3LjDvFR1GHL9ojTQcqbH50838EY0jNJ3GNfSk4+m2CH5Fex2btcW5+WCcbo77dZp+GAQ47s9
LqiKlZusO44PS0Na4d11QDW+ATS09Njx9o1q7rcx//lJxjbq5qFQ2lixZ26eOor439rLPbsU6c2un
1xN32QjsA3+2EEFs2SaDXI+xyou3HKtxrKR0NuedfQcsPX1N23y3dJtEVK0tlzcpB2zYtMj45lcaa
qn6PTQgx9FgTErj8seeZdeJpJKamsfcxJ6CqEb5882VmHnEsux9+Dlqi5bhLr2HEpCmx83759C
Om7rYnf3z9eafRlfj1ZwoKijpyhtlZ82jYPRYLnnwSZKWL0S2mTG67tYLajR4V5diGjEG/ZgJ5F10
NatvuxrHrz+x/MJTMWRmYZ+xAxnHnop1zPh+fU+EEBsWUCO8/84bndqCgQCLlrPZw6Z46h
+8Ql8ZaUk7LwHwaYG1jx0B6EOCafDnoA+r7D9llo2s00lXCMjihrjEYm5eZx6sVXkZqdw/AJk7j
+nodJ61BTxurzcW1N50aHV1/nZqdw5XX3ozV6471MSQkdHkGfUZWBfuXBI0e7Xbdo0ufnO
z2Nd1hjZbkfQ8i67TzSNx9b7L/czH27XYm0g/VuAON9TT/+DW1775K628/E3I5N36SEGKzJUX
E1iPkdAZYXym1FisFV99K2f234q9Yw/A7H+I07/KSZaRm58a0450SaaqtGZDYhBBBDT97I0Vx0
/xOcet1tqKrKZ6+9yJk3342juYnVSxfx1HWXoSgKexx+DIVjJ/Dlm6+QkJpGKBDO9nq+ijVMr69iu
0eeRaOqHl/4mMZ3XwXAkJ6JKSePIIOOoOHjd2PnZBx3Glp7PJIX30KweAkkp6J2mJbZ8v3XD
L/zkS73EkIMPJ1G093yZtC0J5PNPh+aMy8m+PKT1L33OvqUNEwXXEWD18vabyBldXU4dtmL
EYlJaMNBWgwm3vrs/7hop90B8DmduNMqPhtDtN22xO/z0t1UxOm1HTy2q4RQEH31P1cNf
sQAGlJGFqa8D91P4EHn4vFYsrJx5CVQ6BtXbSi05F22DEoxmgirtHpOHCvfamrqWbhH79iMjk
49rSzyUtlbL9GbgHeBX+hMRkxZuWh6PWoGi1xvSxcGHK04vhzDg1f/A9zXiFJe87CMnppc9dX
33Uj9e+/HmsruPo2sk49R7bOEmLJQn0eoRdJn4vItaRLj6Bwmtup/qVZ/h71g4UXn0ibbvtjapGq
Pn1J4aZjKiqiqloW0zx+Dxu/F4vxN6sFC6EGLp0ej1JaelUl60iMSWFR644n0POPJ9PX3k+1uert
17luEuvYfI0O3H2eZfgrCjnpvsf54Hbb8DR1AjAsPGTqPb7caRnU+howfXCE2QecxJx2TkQIRB2
u1h6+IGkzj6CUY+/jHd1KYa8QgzZefhrqvDO+YnWOD+Tc86lnelZGZYjhBga9KrK4cccZ6qp0xmR
mUIEUfh1wd+MHZuQ1ictNZXHn3mc/B1mUIQ0jNWVFcZ730dcduf9sT72pCRee/B9aspWoyjR



8eCzb7+fUNvUanN8InO+/4Yv320f7f4OuP759mOHTo/p6FNwPf0gqCohRcF61sU4dIZYn7DfT
+75V6L6fUR8XvRpGaiKhqDTAYCiN5BmjuOUMaPx77UvukiluLoa4gzt17Bm5RDaZU9afvoG1e
GAhGTid94DUzf7RG9lw6cfsvKaC2LHVS89yYR3viCuaAQAnpJlnZJngLL7byFxf32JGzaiV/cSQ
mweJlFej+gU7oFdJ6Mxmcg+8wKcf//Jmgdvp/jC0wCVZI2OdK0G3703Yjr/KhRzHPHJKTtX1Z
KRXzCgMQohhg5/SzPhqgo81ZWMnbEdS+f+1qVP0OXggn33p/KCUyASwWCz89CDT/DUqy8y
ZtwEtp22Da2t2dx4+w2ccOGVpMw6hEyNhvKH0lf69q1ehS4+AW8gyC133MSlt91L6NVnMI8c
Q/J+s3GvWBbrq7XfK7Tnfv3+/EKIHlJgQk4uCe+8hP/DIQAcvMe+pFutsS761IZOPeRwQq3NeP
75m+zR49hp2gxMzU2xPmGXi900PhKDyUTA58OWkEjxvD/ZfY/o3su01mbmfPZxl9uvWbYk9r
MtPoGGYSPIf+BZ1IZalJR0aklhUuzxsT6qTkfLT1/T+OIHoNVizMgm99Lr2wsWOH2seegOUmcf
gUanQdHG4fb5cC9ZSNJu7ftAJ4weR0KH0eLeCtTVsuaBWzu1hZoacS9eGEugw21JfUdqlEDY
49rk+wohhjZJoLuhqiph3NAR6A7sk2ZgW3KDCIBP4pGy/0Xn832e8/C9NdvBL/7AsOsQ7AnJt
NUVYmJtBBbKXfxYlbccCmuP3+lIDObiZdcz3d//8XfP37Xqd9u02ZQedV5seOw00HD/bdyzPHH
U/PqczS++hRxY8bz1BPPo0lIxBclodOAbdq2OP/6HQctzU7qlUfSVL6Gr4qXUTh6HKEP3kRVVU
w5eZTdezPjXv0IQ3lqik6PbfJ0LOtU8xVCDB6tTofjp2/xr1oZa2v59nOSZ3b4Q5ei4vn6EzzLFhM
3ZjzO999An5SM7bz2/ZpzExJwJSSQZtBjSE6m3uNh9D77Y2ybmwxmRk2cgwNVZWd7t+xQF
h80l/70/9S9t2XsbaE3fcmvuPUao8nmjwDhMP4K9fQ/O1nJO4UrfCvs9mxjBIP5ZMPdLpP4sx9
Nu0NWg81EiYS6Fq1W+1QvdxUMAytLZ6ws70ieNzo8Zhy8vs0FiHE0CFFxLoR8ftAUdB0mAo0
GDQGI2i11FdXkpCWgW77XQh9+TFqOlwtMUnWQQuXlQo6WllxzYW4/vwV46Tpel8+lW9/+p4J
2+5AWk5erF96bh7WDvuxruWvKM0UW0CwrWktZ+kiTIEAlnAYc2kxqCqWMRPlvehqcs+/gswT
zyRosfLMe2+RklfAfvI5tH7yAY7ffilSCDD2pQ+wT9+ezONOI+OoEyV5FmKoCYVx/PFzl2ZP8eJ
OfSyjxmLfbi8pSVYx08mYZc9UQP+WJfE5GSmJCaQ0tJEXGkxo80m8tMzCLSV9zKqKocdeiQ
We3u17jGTpjlsPSN2HG6sp6VD8gzQ8t2XhBvbK/37VhZ3idX556+o4ejnmTEji7SDjyJuZf1b62
W9KNPxtSxwFkfMGZkkXP2JZ3aNHEWLGmMxI7NBUWMfek9rFOmo+h0JM7cl5EPpyu+l1PF
hRCbDxmB7kbY6URrsW684wBoaajHYDBG1zqbs8GeQHjeb1gTEmisrR7s8IQQgyBQU4Xr7z/R
2uzU7TyT+66/AoD/vv4SR194BTnDRqCiklc0HM3yJV3ONxUU4S8vi3GPFwJ+9gsAgs0NRHxelN
4flbOV0Ha7oK5eiSbgxpPicQvN7r3NyXi46dyu1r0fXWSsGI4m77S1VtoUY4tRwGNvU7fCtLu3Ub
swviv2sMZlp/uHr2PZ13pJlmaQKGHZHezHAI MdD/eqVNFjr8Wn1pHvcZJWvxry2+rVWQ7rbyc
3nXUy9CgadloS6GhI07eM1irb7r54d2+NGdv0jXPz2u6Bv2xFA0WiwTppKxolnEqyrQWM0ETd6
PNYJU7qc92+IHXYM+uRkat58CXPhcDJPOKPLHwntU7Zh3Ev/JeRoRZ+UjHaQZjAKIQaGJNDd
CDsdaO0GRgJdW76GxLQOeyOOHEto/lys9nia62o3cKYQYksVUFV0CYkY99iPV196ttNrjeVlZn
pInxpfeJyg14v+IHPlv+lm1jx4O2owiD4llcwTzqTsnvplPutCmr/9An1KGuGWFrQWK+aiEay5+TL
SZh9B9WvPR9fyOVqAaCxc7LbRGE2chRH3PimjzUJsBsJ6lwl77oenZAnuf+YDkDL7SLR5BBE+
EZ83ljyv5VtdSrjDlkwtXi/LdSayFIU4rZa6sAo6PcOC0WnOSjicv7qClJRutN9/hS4+gcRdZhJyNM
euoUtKIW7sBDxL/om1xY2b2Gm7vlgCKQcdTsP/3gdVxTxINNYpMwg4HJza1kGbcvLJOPJE/L
XValxGDCIpfv2dWJISSP98ONJPehIFJ0ORdP95E2dzY6um32yhRBbHkmguxFytqlZ4AJi6107
ZjUJqe3/KGjyCgl+9BbWPQ6gSUaghdieFwuasvLCNZWkX/VLdT8Mx+vK1qoZtvd9uCcM89Hq
SwjvGIz6cecQtl9N7P8nOMZ9firZJ91EagqYbeLNQ/eTiQYQGu1k3bE8ZiHjyLU1linpBhz4XD8q1
ag0RsiNdR1un/KgYeRdthxJO68B/rkNEwFRbFKvEKIoSSUDhI0hcg6/XyCDfVo9Hp0tniCHUZ9F
Z0ey5gJJO83m4jfh8Zoovn7L1F07X2CGi2Ff83Bv2AuYsDBHlfxujsJtk3hVrUabBMmU3rTFdinb
0/YOUr5l/cw8uH23QFCTfWkHXos/u3KaZ3zA/Hb74lx049QU32sj2PFcvxVFeRecCElvirKqh+
6SIMO+4WS6CjMeswddjmsz8N9rl+lcTQlQlON6lj0ENj+k1VWSIJHUagIZRUVK+Xbk2WZTICLc
RWw+fx8H8vPcMOefk0PXwnyg67kDf7KB7aY18CFWtlyc1nxeX/ldjYAlC5aAQFI9/lsvN0wrNyG
S0/fYtrwV+x62UcdxrwGTtgyi0gEgXgHjYsfWIKxeccD0D9/94j/4qbqHrhcyKNDSTvcxC5512JO
a8Ac4dRKYHE0GfS6fBUiVN61w3tjVotl+5/uv3QnkDKQYdTdveNsbs/1yEPjk1dmxorEODiv2OR
wlptRiW/oP/vdfRzNgh2iEYou7/PiD3gqvvrVqB1jqKxN33wTI/Lom7zARAYzbT8NmHJO62F1k
nn42/voaGzz4k75Lr2u8zZj1T96Ps8POAonnX4nBliO8QojBJwl0N0JOx5AZga5ZvYppu+0Z01
YUDZq8QHjbm2UKtxBbgXA4TGXpChqqKklPTCJcspSc/1yEr3IN/rpqqi45k5RDjqL+r99iyTOAt7
QE9/IIF53J8GGORJ0Ow9/TSXuJQtJ2GE3FJOJpo/fJVBXQ9oRx+NZUUY42Hkly8Qbli0BFnG
TCD1kKOJeL0YmJLRmmTfeSE2S6EQDR+/17ktHMa9eCEcdET0sLWJNQ/eOall5dMPY99259i
xNjEZ7QlNESxdjsbv5dbQPwOu2II+ABQtBpUr4eyu2/AkJ5J2OMh7Gyl4Nr262r0BjKPPZVvt11
NqLkJXWISHdfdiUbfYYQ3zoL9urslffoBOFP9t9gX/dgJmCxDY3BDCLF1kwS6G2GnY0h8UYxEV
GrL15CUkdmpXZOTj6Wmgub6uvWcKYTYEridDj556Vmev/VagoEAwydM5qlrr0NdthjLDrtQds2
FqAE/ensirQv/7nK+b9VK/JXlOP6cA4pCwdW3knLA4TR+8X9otBqS9z0lRS2emjdfIHGn3Sm9/h
JCrc2Mf/sz4teOKAkhNn9aDYrB2LW5Q8HUsMfdqel2EF324Wrf51i1WPHfcQ3Bqopog6Jgu+U
BlmMnRV/X6ck8/gxC/8m0LbMzFQ4rL1adts5pTdeFltbHWpuovTGyxjz4vuxLpqFfxFqaiJy8LG
gUVCaGnA9eg/q903AavtXb4UQQvxbso1VN6J7QA9+At1YW43RbMa0TjVHTVYU+qoKWHvq1
3OmEGJLULJgHk9dfznBQACAFf/M5/sffgBVJdDUIh/NKgDqP3qbpD3363K+bfp20P/+l3qqqt



S+/QqauDhskybT8MXHOP6YQ9jZir+slOUXnU6oNVrop+NUbyHEFiCiknXqOZ2aNCYz9unbxY6
1Fiv6dQpxaS1WdPaE2HF4VUI78gygqjhffhp/a2P02OMm0NpM0U33knv+FeRefC15F12Dd/Wq
2CmBmupOhckg+r0UFMVO44bOQbPy0/hu/Z8fFefh/fem7BP2w5dYjJCCDHYZAS6GyGnY0gk
0FWIK0jJyu7SrQSmQ2MdBAP4vd7oFIdCiC1OZemKLM07brstra89i9Vql27bnTAnpWAEngLziD
GkzD6Shv+9h6LTkXnyf3AtmlcaCsXODTbW0/r9V6iRCOMHHkugqYHWeX/g+PPXTvfQJ6Wue
1shxGZM0WlpmfMdbVfdimvJP2gtFkx5hXiWLYNhh12jfYxGcs+/kqoXHsdXVooHm5vs089H02
FGnsbv5RXSPJ+s9EYDLgWL8C1cB6atiJiGnMcWpMRnTWesNuFxmHEE2fB0GH025CWgWlW
oLb9YRBAMRgwpLfPtov4/WSddi41r79AxOclYZeZGDOzCLscnRJ6IYQYDJJAdyPkaBkSe/hVrF
x08jrTtyFadVJJSaNI1dFcV0tGfsHAByeE6FfBQACzuXMthv9cfAWJ9TVo9j8EY3Yu5lkHowYCO
Of9QfXLT1N0200kHnosoeYGdLZ4lp5xVKfzUw85BsViJX7KDPxVFUQ8bqwTp1L1xH2xRnTYUI
Rt6jYD9pxCiP4X8fmJqCpxo8ehMZtRDEaMWTm0/jkn1kfVGVDy8knccxZao4II0I+Sk0vEalr1sU
+citLUQPXLTPxebHP2IHCm+9Dv/azSqOgNVtZfzGp7Z8p2XkMv+fx2DWMWTnknHs55Y/cBe
EwaLXknHs5xg7VtEPNJTR9/RmZJ52Fotfj/PtP6j54k4xjT+3nd0oITZOEUhuBOvrMObkDXYIIC
8vZtj4Sd2+pknPJM/IobleEmghkRLfVFXz77mN0PPpzzPnyPQ449iZHF7DucgGuf+ax/LyTY3
2T9j4A29RtKLnwVPluvR7n33+StPeBjLj/aSqffphgQx3J+x6E6vcRWLOKYF4hq2+7mrDLycT/fs
vE97/GvWwx2jgLIgITpMq2EFsYjdVK2r4HsfTMY2LrnA0ZWYx48LIYH3dEpbmmBqvRiNZmA4+
G5sYmLEnPJLb1UVubqXz2kdg5jj/nYMwrlGf0xGhDIEDVC493mvnr1yDe/FCERbfJXpcVU7j5x8
z/M5Hifj9alxGql58kvjtdsKUGZ11Z5++PWV330TI0w/FrjPyoefR2eN79dzB5ibcixfgr6nCIJOPZd
xE2atZCPGvSQLdJUBdNdYJkwc1BIVVqSgtYbt99+/2dSutk8zSITTV1gxwZEKI/hbweNA6W5g+
cTLjpkzn4AM0JSc3F9XrJtjSjKLTK3PuZWIMJhx/zKHpy/8j76JraPnhayl+L83ffk7CLjNRlyHMB
UXYJk+n+bsv8VeuQdHp0KdlxNYgqqqKbclURBOMDPJTCyH6TTBI5QtPdCoSFqipwjXvDxK23R
EA1eMiORREP34SgZpqjNI5mPw+gh5X7BxfWWmXSzd//RmZp0TXV0fCIYld7BAScrTEflb0Btl
PP46Kx+7Ft2YVprzC6Ehzh2ne1nGTGP/mJ9R/+Db+mirSDz+O+G136tUjh9wuyh+9h+qXnoy15
V16HdlnXoRgr+/VtYQQoiNJoLsRqK9DI5C48Y79qKm2BkVRsNi6/2urJiOL5GCAXrYql0KlZysai
RB2OdHGWWB07R/FgYY6Kp5/DOerzL11oel1NeSk5GJb8Uy0Gjwl5dR+dSDsf7pR52EdeJUliq
EgAJq2L6Ge4iXYt92Jxs8/7nRfXXxCLHmO32FXziXD+/tRhRCDLOLzEqgq79Lur24vCGY1x9Fa
V03p9RfH2jK0043kg4+OHesSkrcpwxz8JBpj9HNHa40n9bBjqXj07k59rJOmtx8oUP7o3YRaok
ULfWtWUf7o3Yx56YNO59gmTcm2aVovnrIz78rInZJngDUP3UnSXvtjGTI2k68rhBBShbsbwYa6
bv+RGEhly5eSkVuAoijdvq6kZmDxe2murOj2dSHE00UtLWH1ndeZ4JdDKb3ISjLl8Zea/7+a1q+
/YLCm+5DURTMmdm0/PQtpTdcSrCuhsnHu50rdp3XiFpr/1Rw2GSZu6He9kilPrI05iTh3n4qE
79c867An9IOflX3cyw2x9CH5/Q788rhBhcSpyF5FmHdGm3T9++vY/fs+WTD3R6veb151E7jEBR
rTbsM9rP0VqspB58FJG2gmD6hATid96drDMvQGuzY8zJZ8R9T6HpMPU6UFsdS57XCrU0E6ip
/HcPuY6w09FNY5iwy9W1XQghekFGONcR9nmJ+LxoB3mfwVVLFPoak7ve1xW9nqDFhrd4yQ
BGJYT4t4LNTSy/7D+45s8FoGbVSlp++Z4Jb31K208nEgxQdOejEAgQam0m5HJQ1bbmUA2HI
RLpfEFVxVw4nEBJA97IS2n84n+kH3MKhpxcFp8wm8xjTyXz5P8QdjmxTpiCddl0Mk84A0UjFz8
VYmuhhELyt9mBjJPOov6919FarGSdcQHaxPbBgrDb3Wntcqy9QyKqMZsx5uaTu+NuEA6jRiKE
HK1o7dF1xaGmRqpfeIKMk84iceYsFI1CyOHAMEcHEtsSb53FBlpttIDYWlotuvXMuNtUpvwidllJ
hJqbYm3GvAKMufl9eh8hxNZHEuh1B0tr0SUmR3fkd6CULI7AtnvN2mCfUFIKIK/aYB8hxNDiKy
uNjC9r2bfdGXfJUpq+/IRAQx3+yjUk7LwHodZm1GCInHMvo/rIpkw7HRjSMwI0WLqhT0kj5HIS
N2IU+uRkUg8/Fn1CEIUvPYlt8gzs2+1E/HY7o7NYB/pRhRBDhalQam7ENnEa1jETon+Eb22BtqU
fABqLBWNOPv6KslibLiERbYeiWxq9EfV0HWj+/kuCjfUk7LQ7lRETcbe2Qna0pkLyPgey5Jj2+126
+ERGPvZi7NhUNlzs086l8pn2YmTZp52LqaCoTx/ZIjPH20ffpfS2q3D9PZf47Xeh4JrbMaam9+I
9hBBbH0mg1xGoq0U/yNO3vW4PDVWVpG+kErgmI4u4hfMGKCohRF9Q9IZOx8Pufhyt1YZr3p
80/N9/yTzxDCxjJ7LyuovxVwOgNaeQP7IN7H6ruvJv+wG6j96B9fCeVgmTiX7tHPRJCSx5MSD
GfnwCwQaGyg+50Rsk6ejxKwPkrK8izEVk5VVSleD6tuv5awsxUA9Elci+60tZH0ejJv/wGql95
Fudfv2EZN5GsMy4AQ/tnViToRxMXR+le+6KggFZLSLkBY050VfdjNFHVITEGCLU2417yD4k77
g6AKTOX5AMow5CVS7C+Fn1qOrap22DKzOnz57ZNns64Fz8g1NqCPjERrWVwZxcKlBYMkkC
vl1BXM+gFxFYtWUh6bj5a3Yb/z2PIKyDx958HKCohRF8wFw0n7cgTqXvFeJ32RNFp0VjsaC1
2sg+8wLQal4faTufwiOP3+I9dcfCTtaaP3jF8zDR1Lx7COMfOBZWud8j6dkGSWXnkX22ZeQe/G
1VL34RLRSrari/PtPAOoKhMhRMBHCLH1UVUav/y/WPIM0VoMvoo17X004Jg/I5QDDiX96BO
J+Py0/PQdaYcdG+sScbsouej0Tpd00/w4L0MmR2/j9xP2uLvcPuzuv07YNn4y5vwiQs1N6BKT
+nVrKZ3NlItXCSH6ICTQ6wjW16Ib5KI6xfPmklU4bKP9THIFqGoEf3MTxsTBHTUXQvSM1hxH3
iXXkrz3LHQpabR89yWKcqtuuZLh9zx0/cfv0/rztwAk7TmL9KNOovbtI/GWLCXvshslu12U3ngZ
3pK2wmOKgn3adqy69Sq8K5eTuMuene4XaKwf6EcUQgw14TC+VSu6Nac7LAdRQ0FMWTmU
P3wnoZZmtDY70edcihoMxPr4K7tW8m76+tpYNIbG7BzSjzyBsntvbu+g1WKbMqPLeZLYCiE2
V1JFZh3+2ppBT6CXzvuD3BGjNtpPp9fTrGio/O7LAYhKCNFXjGkZBJ1OQg0N+KurqHj6IYbf+w
S+stWx5BmiX0z1yalozHEK7DKTsMeJPjW9PXkGsk47j+bvv8JbWKLWmRfSus6sIPQjThiw5xJC



DE2KyUzSPgd2abd03P9dhYrH7otVyA47Hax58I50/fUpaV2uYczJR2mb5q0oCkI77U/e5TcSN2
os9m12ZNTDL2CfSk0fPo0QQgwuGYFeR6C2Cn1S8qDdv76qEq/TSWr2+itwd+QwW2j47guKDj
16452FEEOct2wVq264jLfvfEzyAYdiGTMBfWlyNa8+16WvZ/kSMo4/HeukqUQCAWrfepm8S6
9HYzRizMnHkJFF2Okq7bBjMWTn4Pj1R0ItTaAo5Jx9KfZtdhiEJxRCDCWRYADbj01Jqa6k4X/v
oTGZyTzlblwZ2e19/D5CrZ23I1ID/k7Tr7UWK+ZR4/AWLwai04JkHHsqari9enfc8FGY8gpl2Xc2
GAyYe/h9RgghNheSQK8jUFuNuWjEoN1/wc/fUzh2PJoeVgH3JCbj++v3fo5KCNGXnNUVjHn6D
WpeflakvWdRdu9N6BOTSdhxN1zrFAa0TpmBMSMbrdkSbYhE0CUkEmxqJFhfhym/CPvk9jXO
KfsdHJvGrZXiYUIIQFEjBKorCTY3kXP2JajBIJ7iJZ2+72itVjRxFild1zBrteg6LhHTaljfZgdS95uNG
g6h6A14SpcTv/3One6nMRgx92ApmhBCbl6G3BRuRVH2VRSIWFGUFYqiXDXQ9w/U1QxaFW5
VVfnj688ZPnHKxjuvPSctC6WhjmBTYz9GJoToS3HJKZTddzOq30v5g7cT8bjxV67BkJWNuXB4
rJ918nSsk6bjWbGUKsvOYuW1F5FywKEEmxrRvPb0NIK3PCuyz20Fqskz0KIGEVvoOXn72n98
WsqrHr+PymcepvnrTwl1qJGgsSdSeP1dKHp9tEGrpeDqW9F02J85EgxiSM+k4pmHKX/0Hlr//B
VzXpGsZRZCbFWG1Ai0oiha4HFg6AC+FNRI9VVV0yEPdXQyH8a1ZjyMwaiNt1sXzBPCKRCJ
n5hT0+x56SgtNio/XXH0jZ/9B+jE4I0RdcK4vxLF6lc94fxJ+zCw2f/jf2Wvmj95B+5llknXURgZq
7NvvjEZvwFw4nLxLrsOUX0jY40YXn0jqocdgmzQNzdovu0llsR6RgB9P8alu7f7KitjPqgbiRo5l1B
OvEqyrRZ+ShjY1BYymWB/L8JFUPHo3mcefmjmlw4Fo4D0WvR5+UMiDPiYQQQ8FQG4HeBlihq
mqpqaoB4C1g9kDd3LtqBfrkFLTmulG6ZUwkovLpy88xZefUXo4fRsgPjmFyohK3Qdv9WN0Qo
i+Eqwsx7uiGMVgxLnwL+K36zD1MRym9s0XURQF+/Tt0CWlohNWMZOInGv/bGMHk/yXgeQ
delZJ060u4z6CCF6RGM0krTHvl3aLeMmxX42p2USdDkwZOVgHjEaQ1Y2EY8bQ4fZLJbR4xl5/
zOgKHhLS8g4/gxS9jt4IB5BCCGGjCE1Ag1kAx33SKgAth2om7uXLsKUXzRQt+vkm3dfJxwKM
WLytF6dl5iWzlcztUyY+xu+yjWYsvP6KUIhRF9Qg0EaP/+YjONOpqlpyi48haCzU14li1CYzKTE/E
16D0yWXz8QYx68jVSuqmcK4QQvREJhkYbW/cxUto/fk7FJ2OtKN0xJjVXkRmb7NjKRYOe9ki
wm4XGIMc5pGjMWVkdqWdfwkrOMnrXsLIYTYagy1BLpHFEU5EzgTIC+v7xJG15IFGHMGNGG
NRFS+fuc15nz6MQefcW6Pi4etZbZYiSgazNO2peb1Fyi44qb+CVSlrVRff94Yc/JRw2E8y5eSd+n
1BFuaybv00hRFQ8TvQw1HWHrqYWSdcQH26dv96/sJITYf/fX9Ji47I9bvvyTrtHNJP/okFJ0Owi
qhDhW2AUw5eZgG+HuQEEJsbobaFO5KoON+BzltbZ2oqvqMqqrTVVWdnpqa2mc3dy9agCmv
5+uP/y1HczNPX385C37+gYPPPBfrJu4/nZKVg7toBLVvvoS7eECWiwux1ejrxvLqLEMv/sxFL2
eikfvwbXob0LNjTjmszGYk4d52HCmfDqH/Euvx5Dcd59vQoihr7++3ygaDdbJM3D+9Turbryc1b
ddS9jZinnEmD67hxBcBc2G2gj0n8AIRVEKIsBORwPHDtTNPcVLSD/6pAG5V/mK5Tx/y7WMm
DSVPY44Fq1m0/+WkZqVQ111NdsfdSLF557I+Df+D0NaRh9GK4ToS/Hb7Ilg+PZOwx41Gq0OXI
kF6QuJghyWE2ILZJ0zGmF9E8v6Homi1mPMK0BiMgx2WEEJsdoZUAq2qakhRIPOALwAt8IKq
qosH4t6+yjVEggH0AzDis+CXn3jnsfvY5aDDGDZu4r++XmZBEYt//4X9jj+FYEsT8w/chalb7yZ53
9ko/yLxF0L0n7hBqrcghNh6Ge12jPaxgx2GEEJs1oZUAq2gquqnwKcDfd+6998kfrude1UBe0Pq
q6tYuXA+kUiYxLR0EpJTcTta+e2L/2PFPwuYdcJppPfROqOsgik+fOsVPC4XaQcfRdyIMax56E7
WPHwXOWdfQsr+h8pWN0IIYQQQgjlw25BHowqJEIde++Rs7ZF//razXX1/G/F56meP5c8ke
OQaPV4mxpxuN0YDSZyB0xmiPPvxRDh30V/y29wUDBqLH89d1X7HLQoVjHTcRy8324Fs6j6o
UnKLv3FrJPP4+0w45BZ0/os/sKIYQQQgghxNZEEmig+fuvUPR6TIXDe9Tf63az6NefWTrvT5rra
tAZDMQnJuN2OigrXsK4bXbguEuuWAcuLVFE3fcmS/feJXpe+xFnNWGoijYJk3DNmkanhXF
NH7+MWUP3IZlZATMhcNRtFpCrc0EmxpRA340JjOmveJsk6dJn7EDpsJhnUbjVVU1Nslv6qck
NOBotOjT0rGmJ07KptmCyGEEIIlCRau1RVHewY/pXp06erc+f03eTzPSuK+eeo/cj5z8VYJ0zu
to+qqjTX17Ny0Xz+mfMzyxfMl6doOHmjxpCQnElwGMTrcql3GMkuGo4pbnASyjmF/Y+G6kqOu
+xa0nNyu7we9rjXlCwjWF+LqppoLVa0VhsavYGI30egthrvqHW4ly4i4vdhLhyOxhxH2NGKr3w1q
CqGtAw0cRalhAk5WgnU1WLMYMI6aSq2SdMxjxiFISUNNroljyBAIpOhy4hEWNmDpqN/FFBV
VXUYBBF++z6fRC9NBG/x/u337eCCFEG/m8EUIMFPI3ce2uhHoYFMjpTddjrt4Md6SzdHGvC
K+ePZxImqEcChE00/H53ET8Pu6nJ+Yms7M8RPQG/RQW0mkthltYG17vaFsxYA9y7pyVaCliff
OPzXWZoyLw2A0odXqUBSIhMOEQyFUQFE0aHU6dHo9Wq02us1FfAJTDzgMjd+Hv7qSSMCP
MTsX25QZaG12WDepDYfwV1fiWVFM01efEvF5ex6wVgvhcl+7K3oDik6HGolaAJayq2haPEv1f
NYIaDPbgOno05jg0BiOK3oBGr4v+r8GIYjSi0RtAq40m8Gv/gx4I9KqqRuNSI6jhCGooSCTgRw0E
iQQD0bg7nxB9nu7i1GhAq00jN6AY9NH4dHoUrQYUDShKv/6RlYsHeJUgwEigQBqMlgaDkEka
hoNik6PxmCl/uFDp4/Gvja2TYyz+/ey7d6hYPS1TveO/v/H2nunHnQE6Uee0IfviBCCCCGE2Npt9
iPiqLUA2U97T/KoDE9nWUZq2v7Rr/AF3IEVFVv0GDu2ltFVZVIBMIqbFZvIBZ0irlp+3yruMKNq
6rS88y2pzEpKgk6jTFFqzFYNeg0ikJEVXFCLVEIsHWSBp0RdRwREVFLBoFF2iRtHHaxW9Va
PotD1MwsKqSkAlEISJhFDVilqqVVD0iqLxKWh0MrS9VfjJHwy6us67qhenNKiquu+GOvT282YjU
oCGPrWULIPteW+Ezw/+zddXRbRxbA4d8Tg2WZGRM7zNQ2KTPjlpMzMXHLtOUm5W7bLT
MzN02ahskxxXbMbult39Ike3aASeWlbnfOTn2m0f3abeyrmbmzrb5XEPlmSL5fjNUXoP1hK8E
mtkDKVYYWjFmwCs3NT7jeidlZ9A9zVFUearqjptoOPoK9vS82xLzwl1vNsS8/SX7bV12xbfK5t
8Zlg23yubfGZemuovQZDKV6JNTKGUqwwtOldSrEOJbJlsBBCCCGEEIIIsRkkgRZCCCGEEIIIT



aDJNDdzR7oAPrYtvQ829KzwLb1PNvSs/SXbfU12xafa1t8Jtg2n2tbfKbeGmqvwVCKV2KNjKEU
KwyteldSrEOGziEWQgghhBBCCCE2g/RACyGEEIIIIYQQm0ESaCGEEIIIIYQQYjPoBjqArbX//vur
X3zxxUCHIYQY+ja5Pri83wgh+oi83wgh+ssm329E7wz5Hui6uqGyrkQYqiT9xshRH+R9xshhBic
hnwCLYQQQgghhBBC9AdJoIUQQgghhBBCiM0gCbQQQgghhBBCCLEZJIEWQgghhBBCCCE2w
5Cvwi3EUKWqKs6CITiLC9Da7FhGjcUQlZDQYQkhBBCiAHgdzLxrFqOu6IMY2o65hFj0FmtAx2
W+AdJoIUyIC1zf2HZaUehetwAx051AHn//g+GpOQBjkWIIYQQQvQn1eej5u3/UnTrVeG27GvvIO
3089AYjAMYmfgnGcltxADwNjVSePu14eQZOPHbz2lbtndgghJCCCGEEAPCWbyG4ruu79JW+s
BtOAsLBgisSGSQAsxAPztbbgKV3Vr99bVDEAOYrBSVRVPbfVAhyGEEIKICPM2NaB6vV0bAw
G8DfUDE5DYIEmghRgA+oQk4vY7pFu7OTd/AKIRg5Vj5VL+3ncHAh7PQIcihBBCiAgypmWij+9a
C0drs2PKzBqgiMSGSAItxADQGo1kXXYD9Im7B7dtdvLufxLr2AkDG5gYVJxFBfiaG2n6+buBDk
UIIYQQEWRKz2TUM69jyhkGgDEjm9HPvo4pK3eAlxP/JEXEhBggluEjGP3Mf3Gvq0BjtmDKkG8
YRVeu0mK00XZqP3yTuL32H+hwhBBCCBF0VN3YPxbX+Ktr0Mfl4AhMWmgQxI9kARaiAGktd
qw5l8a6DDEIOUoWkP8/odS/9kH+F1OtCbzQlckhBBCiAgyJCRhSJDEeTCTIdxCCDFluUoLseSP
RhcTh7usdKDDEUIIIYTY7kkCLYQQg5RrbTGG5BT0cfF4qtYNDdHCCCGEENs9SaCFEgiQ8jsd+
Jqb0ccololuNw11dOdAhCSGEEJJs9ySBfKIKiQci1tgRjSiqKR0MuJhZPtfRACyGEEIIMNEmghRBI
EHKVFqFPTgNAHxOHu1ISaCGEEKlgSYJtBBCDELudeXo4+IB0MXG46msGOCiHBBCCCGEJN
BCCDEI+dtA0ZotAMEiYjIHWgghhBbiWekCLYQQg5CvtRmNObjusy42Dk+NJNBCCCGEEMtog
m0oiiZiqJ8ryjKckVRlimKcmmoPU5RIK8VRSki/YwNtSuKojymKMoaRVEWK4oyJZLxCSHEYOV
vaenogbbH4m1sRPX5BjgqIYQQQojtW6R7oH3AlaqqjgF2BC5UFGUMcB3wraqq+cC3oW2AA4
D80L9zgKcjHJ8QgqKvrYWNKEEWtHp0NnteOpqBjgqIYQQQojtW0QTaFVVK1VVXRD6vRVYA
aQDhwEvhW57GTg89PthwCtq0B9AjKloqZGUMUQghBiN/awzakzm8Y9LkHnQqgghhBADrN/m
QCuKkgNMBuYCYaqqr8kWAUkh35PB8o6nVYeahNCiO2Kv60VrcUS3tbFxuOpkqWshBBCCCE
Gkq4/bqloShTwLnCZqqotiqKE96mqqiqKovbyeucQHOJNVIZWX4YqhNgM3qZGXCWFKHo95p
zhaK1RAx1SxAzU+42/rTU8hBtAF23HW1/bb/cXQvQ/+XwzOHLqa3CtLUZrjcl8LB+NwTDQIKh
BIDEE6AVRdETTJ5fU1X1vVBz9fqh2aGf6yf2VQCZnU7PCLV1oarqbFVVP6mq0i0xMTFywQshu
nEWr2HFwcey+Mi9WHTIrhTeehXubXho8UC93/wzgdZG2fA21vfb/YUQ/U8+3ww+bSuWsvjof
Vhy9L4SPHhny58EG9z00CHJYQYQJGuwq0AzWMrVfV9uNOuj4BTQ7+fCnzYqf2UUDXuHYH
mTkO9hRADTFVvqt95jdYfc8Ntte+9QcvcXwYwqm2Tv70tXIUbQBdlw1tfN4ARCSHE9sXvaKf0
/ltxy0JNgQCID9+H+3LFg1oXEKlgRXpHuhZwMnAnoqiLAz9OxC4F9hHUZQCYO/QNsBnQBgw
BpgDXBDh+IQQvRBwtNP47efd2lvm/zEA0Wy7VFXF397WtQfFi0JtBBC9CNfUyPNv/3Yrd1VVj
oA0QghBoulzoFWVfUXQNNa7r16OF4FLoxkTEKILacxW4jZeU8cq1d0aY+aOHWAIt02qR43KAo
avT7cprNF09YgCbQQQvQXbbQd25Qduo2yMqZIDFBEQojBoN+qcAshhj5FoyH5uFMw540Mt8
Xuvi/2HXcZwKi2Pb7WVrQWa5c2rS0aX1PDAEUkhBDh12UjZwb7kKf0DefPeWUc4ka03EAox
JCDLR+qclthNh2WPJGMfa1j3EVFQSrcA8fgd4eO9BhbVP87a1oekqgGyWBFkKI/mQbP5kJ73+
Pq7QIbZQN8/AR6LbhlSeEEJsmCbQQoteMickYE5M3faDYIv621i4FxCA4hNvb1DhAEQkhxPbLI
J6JKT1z0wcKibYLMoRbCCEGGX9bKxpl1wRaY7agetwE304BikolIYQQQkgCLYQQg0xPPdCK
oqCNtuOVedBCCCGEEANGEmghhBhkfk2taEyWbu26aDu+hvoBiEgIYQQQoAk0EiIMej421rQm
Mzd2rU2095GSaCFEEIIQaKJNBCCDHI+Nta0ZiM3dp1UTZ8kKALIYQQQgwYSaCFEGKQCQ7h7
qEHOioarwzhFkIIYQYMJJACyHEIONvbe5WRAXAgxWfV9aCFkIIYQYMJJACyHEIONvBUHTY
wJtw1tfOwARCSGEEIIkARaCCEGHV9bKxqTqVu7zhaNt6FuACISQgghhBAgCbQQQgw6AacDj
bF7Aq21RskyVklIIYQA0g30AEIICtmUv1+2pYspH35YjQWC7aJUzHn5g10WH0u4HahGAzd2r
W2aLxNjQMQRbiqFMDAdqXLaJt6Si0JhNRE6ZiGZ4/0GEJlCSQlwm0EGLIaP7zV5adcjj4/QAY
UtiY++pH29yHwIDTicbQfRkrbZQNnyTQQogt0DL/d5adfbiq1wuAPiGRca99giV/1ABHJoQQQ4s
M4RZCDAl+Rztr/3NPOHkG8FSto3X+bwMYVWQEXD0n0LqoaPwtTf0fkBBiSPO7XZQ/9XA4eQ
bw1tXS9PuPaxiVEEIMTZJACyGGHlDbjadyXbd2T922V1Qr4HKiGLsn0BqLBb/LSaDTh2AhhNg
U1ePBva6sW7ununoAohFCiKFNEmghxJCgJ40j9eSzu7VHz9hpAKKJrIDbjaaH0dCKRoNOhnEL
IXpJZ4sm9ZRzurXH7Lx7/wcjhBBdnMyBFKIMGQmHHEXA7aLypafR2WPJvuY2oiZMGeiw+tyG
hnADaG12fE0NGBKT+jkqlcRQFr/flfgdDtY99zham43sq27FNmnaQlclhBBdjTQQoghw5icSuaF
V5F89EkoBiP6mNiBDikiAm5Xj004IVhlzNvY0M8RCSGGOkNiMhnnXELSEcei6PToY+MGOiQhh
BiSJIEWQgw5hqSugQ4hYISfDzUQQNH2/PYcrMQTcbQQYssYEpMH0gQhhBjSZA60EEIMln6X
E43RhKloPe7XRdnwSQ+0EEIILcSakARaCCEGkYDTiWYDw7cBNNYovM1SREwIIYQQYiBIAi2EEI
NIwB3sgd4QrTUKX0N9P0YkhBBCCCHWkwRaiEFKDQGGOGQxAAJOJ8oGKnAD6Gw2vJJACy
HEoCN/t4XYPkgRMSEGMdaF86l640XcFWWknHAG9pm7o4+JGeiwRD8JuDY+hFsbZcNZUtSP
EQkhNgYR+Fqj6j56h6bffyT+gMOI3+dgTBIZAx2WECJCJIEWYhBpW7alpccfRMDtAqD5tx/Ju/
9Jkv910gBHJvqL3+IEYzBscL82KlqqcAshxCDhqa5i5fkn4VyzCoDW+X/Q+tdc8u97Cq3VOsDR
CSEiQYzWcZGltP49L5w8r1f+xiOy7u92JOBybHQltzYqSqpwCyHEIOFYszKcPK9X/9kHONcWD
1BEQohikwRaiEFE0fTwn6RWAxtY0khsewlu10Z7oHW2aLxNUoVbCCEGHZ7+bivKBpciFEIMfZ



JACzGI2CbPQGuN6tKWecn16GNiBygi0d82VURMGxWNR7kJVVX7MSohhBA9MeeNxDP2Qpe2
pK00x5Sd00ARCSEiTeZAC9FPvM2NaPQGtJYNz4myjh7HuDc+pfaTd/GsW0fi4cdgnzGzH6M
UAY3gcqLRb7gHWMmwoDEY8Le2olu292NkQggh/smYmMzlx16i4bvPaZn3O3F77k/MLnuhN
VsGOrQNUIUvX1MjGosF7UaWTRRC9EwSaCEizFNbTd0n71H5ymz0SclXXId9h12RtH1/J9f1L
hJRI2b1L9BikHD73KibGQIN4DOHO03sV4SaCGEGATMucNJP/Mi0s+8aKBD2SRXWSnVb79K7
ftvYh45hswLryZ68rSBDkullUWGcAsRYbUfvk3xndfhKi2idd7vLDvtSFqXLhzosMQgtakeaABdtB
2frAUthBCiFwleN2VP3E/5Ew/grlhL03dfsOzkQ3H8owiaEGLjppAK4rygqloNYqiL03UdpuiKBW
KoiwM/Tuw077rFUVZoyjKKkVR9otkbEL0B299LeteelJro99P28L5AxOQGPSCc6A3nkBrbdF4J
YEWQgjRC+515dS8+3qXtoCjHUfBigGKSlihKdI90C8B+/fQ/oiqqpNC/z4DUBRIDHAcMDZ0zlOK
omgjHJ8QkaU3oLN3LwD2z0JhQqzndzk2WoUjQBtlw9soCbQQQojNp+j1PdZh0ZjMAXCNEEN
XRBN0VVV/AjZ3wdLDgP+pqupWVbUYWAPMiFhwQvQDfbSd7Ktu6dqWklRtivxfW/Qs4HCg2U
gVbgCt1YZPEmghhBC9YErPluvKm7u0WUaPwzJq3ABFJMTQNFBFxC5SFOUUYD5wpaqqjUA
68EenY8pDbUIMaTG77MG4Nz+n5c/f0MXGYZ8xC8vwEQMdlhikAk4nutiNL1ums9nw1Nf2U0R
CCCG2FUIHhO95WB6tf83FIJWLbdpOmFLI47YQvTEQCfTTwJ2AGvr5EHBGby6gKMo5wDkAW
VIZfR2fEH1KYzBinz4T+3RZjmoo6u/3G7/LgcGQstFjtLZovHV1EY9FCNG/5PONiDSdLzYXfYid
pe9BjoUIYasfq/CrapqtaqqfIVVA8AcOoZpVwCZnQ7NCLX1dl3ZqqpOU1V1WmJiYmQDFkJs1/r
7/SZYRGzjQ7h1tmh8DZJAC7Gtck83Qggh+PV7Aq0oSmqnzSOA9RW6PwKOUxTFqChKLpAP/
Nnf8QkhxEAKOB1ojJuYA22LiJiQgghhBADIKJDuBVFeQPYHUHQFKUcuBXYXVGUSQSHcJcA5
wKoqrpmUZS3gOWAD7hQVVV/JOMTQojBJuB2bbIKt85mx9e4ufUzhRBCCCFEX4loAq2q6vE9
ND+/keP/Dfw7chEJlcTg5t+MldzBHmhJolUQQggh+lu/D+EWQgixYapr0z3Q2qgo/I42VJ+vn6IS
QgghhBAGCbQQQgwwqfpcTjcG00WMUjRat1Ya3qbGfohJCCCGEECAJtBBCCDCoBwtlEz3QADq7
Ha+sBS2EEEI0a8kgRZCiEFE3YwiYgC6mDi8dTX9EJEQqgghhFhPEmghhBhEgj3QGY8iBqCzx+
Ctkx5oIYQQQoj+JAm0EEIMEqrfr3o+g2vUCCzhaNR3qghRBCCCH6ISTQQgghxSKzvfVYUZZP
H6qLteGqr+iEqIYQQQgixniTQQgghxSARcljSbMXwbQk04a6ojHJEQqgghhOhMEemghhBgkAi4n
GuPmJtCxeGolgrZCCCGE6E+SQAshxCARclS3awkrkCJiQgghhBADQRJoIYQYJAluZy+Gcmfib
ZAEWgghhBCiP0kCLYQQg0TA40bR6zfrWK3djrehATUQiHBUQgghhBBiPumghRBikOhND7RGp
0drseBraoxwVEIiIYQQYj1JolUQYpAluFyBPQcaQB8bh1fWghZCCCGE6DeSQAshxCARclVQbOY
QbgCtPRaPJNBCCCGEEP1G15uDFUVJBqaHNv9UVVU+uQkhRB8J9kBV3hBuCFXirpW3YSGEE
EKI/rLZPdCKohwD/AkcDRwDzFUU5V+RckwllbY3AZcTjX7zh3DromPw1FZFMClhhBBCCNFZb
3qgbwSmr+91VhQIEfgGeCcSgQkhxPYm4Hah6DZ/CLc+NhZ3ZUUElXJCCCGEEJ31Zg605h9Dt
ut7eb4QQoiNCLhdm72MFYAuNh73uvlIRiSEEEIITrrTQ/0F4qifAm8Edo+Fvis70MSYus5i9fQMu
933NWVRE/bkaiJU9FZrAmdlhAb1dsEWWh8Xj6dqXQqJEkKlwcncvdtG2aAet835DH5dA9lyZWlaP
GOiwhBDbgc1OoFVVvVpRIKOAwaGm2aaqqvh+ZsITyCs61JSw7/Sjca0vCbSP+8zyJh8iUfTG4B
Zy9q8Ktj0vAUyNzolUQ25+mn75l5bknhLcNyamMe+0TzMPyBjAqlcT2oFdVufVVfRd4N0KxCL
HZXOVrCbhdGFPT0f6jZ7I92cluyTNAyT03Y99pVwwJSf0YpRC943c50PSmCndsPN66GtRAAE
UjM2qEENsHb3Mjppfd2qXNU11J66K/BiyB9tRW42tqRJ+YjD4mdkBiEEL0j95U4T5SUZQCRVG
aFUVpURSIvVGUikgGJ8Q/+RztVL/1KgsP2pm/95nOqkvPxFm8pssxAaez+3ktzQQ8nv4KU4gtE
nD1bgi3Rq9Ha7Hira+LYFRCCDG4qB4Pvpambu1+R1v/x6KqNP3yPYsO340/99uBZSceTOuSv/
s9DiFE/+INl8X9wKGqqtPVVY1WvdWmqmp0pAlToiftSxey5rqL8Lc2A9D47eeUP/0IAa83flx1Fi
0tmhQIHbb2qnnYkxJC2/7HO20LpxP3Wcf0LroL/xOR/89hBABEHA5e7UONIAuPhFPtcyDFkJs
W9w1VTT+/B31X32Co6igy5DYjJpp1/Q9QStlghxk/ovwBBnUQErzjkeT2hFhPYVS1I18WI46mo
2caYQYqjqzRDualVVV0QsEiE2g7Nwdbe2us/eJ+uy6zGmZeBztONraSHjgqtAUfC3NqMxW0g6
4rjwENeAx0P1Gy9S8u8bw9flufleUk86q1fzT4XoawGXC41h89eBhIAhsepKGIAPjklIEQmuirWsv
vQsWhfMBUAbZWpsqx9gmzgtfEziUSeGAGxUvfoc+pRUsi65bkASaNaEgKuriPf3GtLcK8r79W
0Mb/LSdviBbSvWlohPpGoiVMxZWb3dbhCiD7Qmwr6vqlobwlfAO71jaqqvtfXQQmxlfoe/hhZ8
kaiibIBUP/ZB6y5puNb6ahJUxn11H+79D47i9dQcu8tXa5Res9NxMzcDevIMRGKXlHNC7icKPpe
JtAxcbiWtBBim9l6f244eQbwt7WY9rH7GfXky2hNZgCMSSmkn3kRSUedgMZgQmuxDEis+rj4b
m0aswVdtL1X12n4+INWX3pmeNs6dgKjn30DY1rGVscoxGCgKEoMcIKqqk9F+D6HA6tVVV0e
qXv0Zgh3NOAA9gUOCf07OJBCEhEhUeMnY5+1e3hbYzSRc8O/0UfbcVdWUHL3jV2Ob1v4F+0
rlnZp8zXWg9/fpU31+fA1NUQsbie2R8DI7HUPtC4mFk9VZYQIEkKI/ucqK+nW5lixhEB79zn0+pi
4AUueAcx5l0k/77lubcNufwBT9rDNvoanpori06/v0ta+bDFtyxf3RYhCDBYxwAWbOmg9JWhLK
qQeDkS0R6w3y1idvrH9iqJcr6rqPVsfkhAbZkxJY8TDs2lfuQx/exuWYfiYRowGgmvo+lqau53zz
6lixvQsdPZYfM2N4TZdbBzGjKzIBi/EJgTc7t73QMfF415XFqGhBCi/0VNmNytLeHgo9DFdu/tH
Wg6axQZ519J7B774ampxpSZjWXkGJR0dVg2JeB29fglfqC9vS9DFWKg3QsMVxRlIfA9MAGIB
fTATaqqfqqoSg7wJTAXmAocqCjKKCBJQC1QBvylquqDiqlMB54EEgl28p4NxAghArsPiniTcJS
qqoV9/SC9WsZqE44GJIEWEWdITMaQmBze9re10r56BX63i4SDj6Luo7fD+xSDEUveqC7nmzK
zGfXs6xRcfT7ushJM2bnkP/A0pnRJoMXACrh73wOtj0+k9e/5EYplCCH6XsDjxlGwEndlBcbUdC



z5o7os4WebNIOcm+5m7UN3EXA5id//MFJOPGPQLtens0Vjnz5zi883pKST9K+TqP7fS+E2Ra/
HnD9qwywJMfRcB4xTVXWSoig6wKKqaouiKAnAH4qifBQ6Lh84VVXVPxRFmQ4cBUwkmGgvA
P4KHTcbOE9V1QJFUXYAnlJVdc/QdT5RVfWdSD1IXybQm/9VmxB9xNfSQtkT97PuuccBSL/gS
IJOPJO6z97HPCyf7KtvC/dQd2afMZMJ73yNt6EOfXyCrA8tBoW4y4XS2wQ6IUl6oIUQQ0bA66
Xm3dcpvOlyUFVQFibf9QjJR5+Mogt+LNVFR5N2+gXE7X0ggseDMT0Lrdk8wJFHjkavJ+P8K9B
G2ah59zVM2cPlueY2rKPHDXRoQkSKAtytkMquQABIB9b3jpWqqvpH6PdZwleqqroAl6loHwM
oihlFzATe7jTao3fLmGyFvkyg1T68lhCbpX3VsnDyDFDx1EPYzSxk/DtfY0hMRhcqLtYTQ2IShkR
JnMXgEXC70fR2CHd8lp6aKtRAYND2zggghxHrO4jUU3XZ1MHkGUFWKbrsa27QdseZ3fOGtKAr
mrNwBirL/mTKzybnuDtLPugiNxbRzy9CbANOJDj0eqqqql5FUUoAU2jf5sxd0ABNqqpOikx4m7
55X5EeaNHvLXV3dpa//wN1euVPz5iyAm4e98DrTEY0EZf46mpilBUQgjRd3wNdaheb5c21ev
FW183QBENHopGgyEpRT6/iG1VK7D+/9x2oCaUPO8BbGjNtl+BQxRFMYV6nQ8GUFW1BShWF
OVOCBccm9jDfSKiLxPotzd9iBB9y5SVA/8o1GEEdMx5DUgoAqqriLC2ibekiPA31AxChEJtP9fS+B
xrAkJSMu0KGcQshBj9DagZaW9clnrQ2e7flmgJeL47C1bStWIKvrbU/QxRCRICqqvXAr4qiLAUm
AdMURVkcNAks3MA584CPgMXA58ASYH3F4BOBMxVFWQqSAw4Ltf8PuFpRIL9Dhcb63CaH
cCuK8jgbGZ6tquoloZ9393DuCwS/KahRVXVcqC0OeBPIAUqAY1RVbVSCA9j/AxxIsJLaaqqLu
jl84jtjGXEGEY8PJvCW67E39qCaVg+efc+gT4mFr/LSd1H71B0x7UEH02YR4xmxMNziBozfqDD
FqJHW9IDDaF50BVMHWHCEQlhbB9x5ydy6inXmH1Fefgra1Gn5jMildndxmu7W1qpPLIZyh/6i
FUrxf7Trsy7M5HsAzLG8DIhRBbS1XVEzbjsH90/n9QVdXbFEWxAD8RKIKmqmoxsH8P9/iVCC
9jtKt90PMJBmoCpgAFoX+TgE190nuJ7g92HfCtqqr5wLehbYADCFZdywfOAZ7ejNjEdk5jMJB
42DFM/PhnJn70ExPe+oKocZMAcKxcprrrLiLgCE6lcK5eQdGtV+FrbRnAilXomaqqwTnQht7Xw
NDHxeOuWBuBqIQQou/FzNqdiR/+wMSPfmTihz8QM2v3LvtbF86j7D/3hod6N//+E5Wvzkb1+w
cgWiHEAJsdWvpqAfDuYOhg3WQPtkQqLwMoinl+sL0qqr7Q9jPAz5s496fQel6dHQbsHvr9ZeA
H4NpQ+yuqqqoES5nHKlqSqppq5WY/jdhumbNuyrW51haHf4+aMAXFYKbt4Xy8tXobNH9GJ
0Qm6b6fKAoKFptr8/VxyfiWlvS90EJlcQUUv1+0Gg2uB6yMSUNY0paj/valy7q1tbwxUdkXXQ1+
vjEPo1TCdG4bWavdb/qzRzoWKBz1hEVauut5E5JcRUdJcvTCS6OvV55qE2ILWJITMaUIUvuTX
ejT0hEYzCSfe3taCzb7IlyYugKuJxb1PsMwSHcrvLSPo5ICCF6z9vcRN1nH7D0IMMpuPjCWhb8
iar2bqEWc073aYtRk6ahiZlv4UQA683y1jdAyxFQOUHghW3dwVu35qbq6qqKorS6+WvFEU5h+
AwB7KysrYmBDHA/O3tuNeVoRhNmDKzN/hN9ZawjBIP5qXXU3D1eRAIAND824+YsnlXpsh3M
2Lz9Nf7TcdRmPcsgTakJCEe115H0ckhOhv28Lnm8ZvP6fgqvPC23Wfvc/4t7/CNn7yZl8jaso
MEo86AUNCEopeT/vqFWReeBXaLXyPFEKlvtSbHui9gRclVkl7D9hJVdWxtuCe1YqipAKEftaE2i
uAzE7HZYTaulFVdbaqqNUVZ2WmChDeYYqZ3EhKy85nb/324GFB86k8pXZ+Npa8TkctC1dSP
PcX3BX9fh/gc2it8fSvnJpOHleb91zj+N3u7Y2fLgD6K/3m4DbibIFFbgB9lJeNaV97qXRwgxuAz
1zze+5ibKn3ywS5vq8dA6//eubYEAjsLVNP32E47C1aj/+Duti7JhyR9F5cVPUv7EAYiAVqZeCSE
Gid4k0M+Hfu4FPAI8qSjKpVtwz4+AU00/nwp82Kn9INA6XjsCzTL/edsV8HqpeO5xmr7/MrjtaK
f49mto/Xseax+6k0WH7sbS4w9i8ZF709bDXKjN1dOQWEVvQFH6cgU3lbZewOXA4h5ordmCx
mzGW1ez6YOFECJCVBTQ67u1K9qOAY9qIED95x+y6OBdWHbSiSw6eBfqP/+wSxLdunA+pffe
QsDIBKDHq0+oeu2Fbom2EEIMhM3OIlRV/R74N3AzMAeYBpy/sXMURXkD+B0YqShKualoZwL
3AvsoilJASff73tDhnwFFwJrQ9S/o3aOlocTbUEf9Zx90a29ftpiq154Pb3uq1IFy363429u36D5xe
+3fbVmg9LMvQbMFSwUJEUKB15b3QAMYkIoIkJgQYkDp7XayLrm2S5vGYsU2bcfwtr04kIKrzi
MQGgkWcLsouPp8nMWF4WPafncvslv3ybv4GusjFLkQYrBTFCUntlb0gNvsOdCKonwLWakmx
D8D01VV3Wh3h6qqx29g1149HKsCF25uPGJo00VFYRk1lpa5v3Rp18fFYR09DkveSNqWLcKxc
hmt83/H29y11mrt9X2iJk5l3BfUffp+/jbWkg87BhsU2b01WMI0We2pgcawJCUgmttMdGyFrQ
QYgDF7LYPY158h9qP38GQIErCAYcTNWZCeL+3toqA24V17ESso8bSvnIZ7csW4amtWjl8HwB
TZk6361rGTEBjjeqvxxBCiA3qTRGxxcBUgotbNwNNiql8rqqqMyKRiSHP29Slt6EOfUwc+rr4Lvu
0VhvZV9/KslMOD6/TbN9tH0xZwzAkJNH0249ET9mBhAMOp+XveehjtqTgOyiKQvTk6URPnr7
VzyNEJG1NFW5Yv5RV8aYPFEKICNJZ04jdbR9id9unx/2GpBSyr7uTtoXzaPr1B6lmTiXhkH9hT
EoNH20btgO2qTvS+tcfQHD+c9bF16A1ySoaQgwFvw6znwDcDWQB4EbZhu1v94HI9YqijlHm
EmwVtZhwEkEiy8aCi5kPIIVVYeiKC8BTmAyKAScAZwC7ATMVVX1tC0NYrMTaFVVLwdQFMU
GnAa8CKQAUhJRdNO66C8Kb7iE9hVLMeeNJO/ux4juNIQLiHrKDCZ++APOwtVoTGZ0sfGsPP
9EPKFqwnWfvoe1eAL5Dz6D1tL73mchhpKAY7mVPdDJuEqK+jAiIYToexpbNNVvYKrqACAhq
p10ApWknjEseFjdPZY0s+9FMfKvVG9HoyZ2bL+sxBDRCh5ngNYQk3ZwJxfh9npgyQ6HzheVd
WzFUV5CzgKeE9V1TkAiqLcBZwJPB46PpZgwnwowXpbs4CzgHmKokxSVXXhlgTRmyHcFwG
7EOyFLiFYkfvnLbmgPFR8ba20r1qOt74WU2YOlvrKLoN/1/HXbW0leeegKemCgDnmIWsoPs4
Jn70l6bM7C7HWoaPQBdtp/zph7GMGBN0ntdrX744PE9KiG2Z3+nY6jnQLfN+3/SBQggxgFwlr
eHkOdxWVICrpAhjYjIQ/BJ+5TldZwGmn38l2Vfd3GW5S2fxGhyFBWitViwJrmPYgiTbUVKiy/kS
Ah43lhGjuww3F0JskbvpSJ7Xs4TatzaBLu6U9P4F5ADjQolZDBAFfNnp+I9DyyYvAapVVV0CoCj
KstC5C9kCvRnCbQleBv5SVdW3JTctQ4+/rZWYJx+k4tIHgw1aLSOfeJmE/Q7Z4Dnu8tJw8ryer



7kRV1lJtwQaoOWvuRhTM/A1N3a/mKIE/wmxjQs4t3lId1IKrrKSvgtICCEiQd1AJe1O7W0L53fb
XfvB/0g/8wL0cQkAtP49j2WnHoG/rRWAmN32Je+e/2BMSdvsUNpWLGHNRRfSviiy42ocuLp5R
T/8X+/SZm30NIUQ3G1rEvi8Wt3d3+t0PmIGXgMNVV2kKMppwO49HB/4x7kBepcHd9GbKtw
Pqqo6V5Ln7Uv76hUdyTOA30/hDZfgqijb4Dna6JjuPdSKgs4e0+Px7soKWhfOA42GmN327rlv8
bBjCHg9Wxi9EENHwOXsVjG+N/SxcfiAm/E7HX0YIRBC9C1Fpyd+/007tMXvfyoOpa/MqZldjvP
kjcKjTk4ncvvaKP0gdvDyTNA049f9Vi9e2Na5v4aTp4BfA31VL70DD6nlPcRyius7WX71rIBIYqi6l
ETI3SPLrY48xbbh57WlfU1NuBrboL07n/gAMy5eWRfezvNv/+MddQ4HIWrsI4ej3n4iB6Pt+SPp
PqNFzEPy8eQIELmpdfhra3BkJRC+6pluMvXwtQdezXiG3F1i5jpWi0GJJTcJeVYhXkug8jE0KI3l
F9PtZVIWj0BgxJyV326ex2NBYrWZfdgKemCn1SMq61Jeg7fckePW0HrGMn0L5sMQAas4XMS
69Daw4WEf01ttK2fEm3+7o38uV+T5yFq7u1ta9cir+IEZ1ZCpYJsYVuoOscABHqD0SbgbmAr
Whn7YI3SdMEMixUabMHNbQwe8Pt5nzRmJM3fAQKY1eT/T0mdR/+QnlTz1I1MSppJ99yQarZ
1pGjCF2l71oX74YX0szNe+9gS7aHkzSAwHSz76kj59KiMHH73Ru9frkhpQ0nCWfKkALiQaMu6
KMihefpuq/c9BF28m54d/E7XswulAxUHN0HrG77kXBdRehNVvwOx3k3/sE5ty88DVMmTmM
nv0/2lcsleByYskb1eV9TR+XQPX+h1Dz9qtd7m0ZOaZXsdqzmzKDqtee7tMXtuT/6xJTEPrYQIm
RWUfPrvw6zQx9X4VZVtYTgalDrx/stPvpHo4/bSPnnvbP43tDEmixUZb8UYx8/CUKb7gEX1Mj5
uEjyH/oWfSx8Rs8x1Ndxc0LTg4XBGtb9Bcrzz2eCR/8gKmHXmtjUgrJJ5x06cN3Eb/TrgQcDtpX
LEFri2bYLFdhGTU2Ys8nxGARcDq2PoFOTMFZUthHEQkhR09Vv/s6IS88CYC3rpaCK85h3GufY
N9pFwAUrZaEg47EMno83upK9MmpWlbo2i6zio0pqZjTE3v8R4avZ70cy/FXbGW5t9+RGMyk3
XITUSNn9yrWkOn7UTamRdR+cqzqF4vsXvsS+IRx6HRbPYMRyFED0LJcl8sWzUoSQtINkrR6Uj
Y/1Cixk/G19yEISUVQ6iAx4a4ykrwtzRjSE3HU1kBgLe+DldpcY8JNASrcY948BlcZaXE7rEfAbcb
nc0W7AEXYjsQcDpQtqKIGIAhJRvNycGmDxRCiAjwNNRR/dYr3dphFs4LJ9AQTKKt+aMgf9QW
38syLJ9Rz7yGu3wtGpMZU1Z0tyR8U0yZ2WRfdQsJhxyF6vVgGjYCQ2zcFscKhNg+SAltNing8e
CuLKfpx2/R2mzE7LxHt2UeVJ8PVVXR6PWg1ZJy/Ol4G+sxDx9B43dfBJfX0WY4mrbq86Ho9F
hl6KnYtvdjg0W2ttcxpQ06r/+rG8CEkKIXtKazJizcrstSWll6nlltN/tQms0bdG9/E4HbUsX4Sxajd
YShp1Y8nv+hnCVVkeLBS2dBG2ydOxTd8J4z9i0RiN2CZM2aYwvvpWBu8z/Il2KbuQPT0nTA
kJG3VNYUqg5cK0GKTWv78hWWnHgmqCkB5lI3x//sc65jxqD4fLQvmsu7Fp/E1NZBx4dWsue
YCPFXrwudnXnY9lIHjKJ/9H3ytLUSNmRDuiVZ9Plr++oN1LzyFr6WJ1FPPI2bn3dFFRQ/lswoxU
AluJ5p/FNvpLUNKGq7Soj6KSAghekdrSjZxybW0LJiL6gmuoGHKzSN66g5dJnMWF1L74Vs0fv
9lcNj0Ycdizh3eq3s1//kbbX//SdPP36GPT8TX0gQ6PZbQXGpvcyNFt1x47efh89JovZUcm++F
53ln0vUbjlPFR0FV19Ayx8/BxteeJLU0y8g59rbtmppQiHE4CUJtNgov8vJulfmkHraeWgtVtyVfd
R98i5Nv/2Adcx42lYspemX77GOG3zn7/R+tcfXZJngOrXXyTtzAupenUotrETafz+S4bd+gBao5
HWRX+x9KRDw0XKWub+ysgnXybhgMMH4GmFGDj+PhjCrY9PwNfchN/pQGvuuw+IQgixuTy1
1eTech+q242i1+NtasTX2hLe722oZ/WV54TXem5b8jeNP3/HmDlvoo/rqK+i+v24yooJeDyY0rP
RWQ0d12hupHX+75Q/2VFDqOnn7zBlZIUTaOea1V2SZ4CaN18m7eSz0Y0Z32fP61yzsiN5Dql8
+RmSjz0Z64jeFTUTQgwNkkCLjQp4PcTsuAtITz6Ar7EB87B8cq69HZ/TgaOwgNoP36Lq1dmo
Ph9xe+6PeVh+t2v4He14qiuBYI9z7QdvkX7GhVjyRtL449ddKnwDVDz7H2J32xftP74hDni90At
X462rwZiWiSl3OIqy4WHhQgwlfVFETNFoMaak4Sopwjp63KZPEEKIPuRtbsRdUUbzL98Hi3sZT
SQffxquogJsoQJfzul1eOvryLz4GtRAAEWjoeaDt3AWrwkn0N6mJqpef57yx+8n4HYRu+d+5N5
0D+acYC+1r7mJmnde63LvgNuFs2gN7Bna9np6jDHgcfpMwc8PdownEED1evv0PkKlWUMSaN
Ej59oSHCuX4nc58TTUYUHJI+30Cwi4XQS8Xuw77kb9Fx+GK20CNHz70eYRozFmZAXXbg5JP
PxY6r/+DEWvRxdTB78PQoU+NMbuPW4asz8fz2/20XN269RdPvV4PejMVSY+cTLX02xb4Re
ASH6V8Dt6vG/h95av5SVJNBCiP6mouCtriR62o7BYdtaLd66WIR/IHyMYjSSfPRJID1xP6rHg2l
wkHnRNSid3v9a5v/O2gfvCG83fvcxowcht1yL4pGg84WjaLv/oWjptPIG0tuHqacYbhKOqa1RE2
ZgSlNwJ8+s3l4fpeiQD2nffEIjXbp/cRQgwekkCLbtpXr2D5aUeGh2Lr4xMYftejrLzg5PA86OQT
z8Tf1trt3MbvvY/odIUzHkMV2kRiYcdg7+1BVNGFmmnnUvF80+Sdsaf4eraMbvtFkTDxJwu8
LXyDjvcrSmrkVFnGtWUXTrleH7B5wOCq46l4kf/bTByt5CDCUBp30rh3ADGJJTcBZJJW4hRP9
Tff7Mw0dQcs/NBJwOACwjRhM9faeOY3Q6yp98MDxHWvV4KH/yQWL3PjB8TOvCed2u3fDIR
2RccAXGpBT0sfFKXHAFhTdcGt6vjY4havyk8LYhOZXrz7x05X+fo/m3H4nd+wCSjzkFfUzfVtk2
pWUy5vm3qXxlni1//krc/oeS/K8T0dmklosQ2ypJoEU39V9+3GUes7e+joZvP8eUIRsuUFT34Vt
knH85df84N2r8FKxjJzLqiZcluD3obDacxWuwjB5P00/fkHPdHdh33CVYrRuWjZ/MuDc/p/7Lj/A
1N5Nw0BHYPszoFpOnujKcPK/na2zAW18rCbTYJgSczq0ewg1gTM3AsXpFH0QkhBC9pDdS9/
mH4eQZwLF6Bb6W5vC2v6mxy5fmEBYB42tqCG8bErsXVDTnjYTQtC1fWyttSxaSfc1ttC1ZiD42
DkNaRni62HqWEaMZduv9+Nrb0EXZer3M1eayjhrL8DsewudoD/aOy/QysZ3bM0ZzAnA3kAWs
BW74rimwVetCK4piBd4CMgAtcCdwX6jtAMAJnKcQ6hpFUQ4BbglMQD1woqqq1Yqi3AbkAsN
CsV007Bg6vwl4RFXVTC6/kARadNO+fHG3NtfaEgzJqeEE2jZpOtaxk7CMHltj1TIA9AIJpJ52bri
65frqk+bcPMY5eSQd+q8e72ebMGWTS0gYU90Dw7oDhCpA9AIJp6RFWloCridaAxbtpxLZ8b0
TBp//q4PIhJCiF7yenCVFHzr9jXUh383JKeiMVu6JNkaixVjcmp425l/iqjU2lB9BcQ7F1OPubkc



O9xoL2d+s8+wDp6HDG774u/rZWq/z5H+rmXdbu3otWij7b31RNukKLT9ct9hBjsQsnzHGD9nlp
sYM6eMRq2MoneH1inqubBAIqi2Akm0M2qqo5XFOUU4FHgYOAXYEdVVVVFUc4CrgGuDF1n
OLAHMAB4HTHkVdVrFEV5Hzgl+GBTgUgCLbpJOPAIGr78uEubfdburJv9HwCyLruB5nm/UXLv
zaSfeymKVoc2yoYpKwfl8BHhc3ytLTiLClADAcy5eehjYrc4JvPwkeTf/xSFN15GwO1CFxPLiEef
CybWQmwD/C5XlzmAW8qYnomrZE24O18QQvQXXVw8iYcdQ8Uzj3Rpj5o4Nfy7KWc4I596lda/
5olaAl0G25QdMOV0LGNIGTGatLMvAb8vOE/aYsU6YnR49Jo+IZHcW++j7qO3Kb3vFrRROaSec
jZRoUJlQogBdTcdyfN6lID71iTQS4CHFEW5D/hEVdWfQ6M93gjtfnY/+aTAbypKEoqV7o4k7
X+VxVvVa+iKEsI9mR/0en6OZsTiCTQ2zBH4WqcRavRWqKwjByDISFps86zz9yVzMtuoOLph1H
VADk33oMpK5uMC69Ca7HSvmo5AY8H+067UnD1+eD3o09MJU+exzFI5aLR63FvRKX4jvtp+P
oTAKJnzCTv3icxb2HxDo3BQOLhxxl1aRrehjqMqemY0rO26FpCDEYBI7NP1gzVWqxoLFY8IRU
YZXqDEKITZ0KRjjUr0RiMWEaO6dLr2ydUFcvlMcTvfj1X32C1moj9eSzoJjMXQ9zOVn3wpMEH
O1oLFZGjJnQZb+vvZWmH74KV9o2540k774nO11ApW3ZlqLGTyZq/GQURQ5nSSEBp7Nvn0cl
ssU29AF9qz64q6q6WIGUKcCBwF2Kony7flfnw0l/HwceVIX1I0VRdgd63SMO3S9gKloXIUNzx
ENsJm5sSTQ26iWBX+y/NQj8Le3AcEe5Pz7n9pkj23b8iW0LV2IMS2DcW98hjbATvVbr1Lzv5dw
V5aDRsPwOx8levpMCq44O3yet7aa0vtvxZiRiXEXGJp+/p6Grz/BID0MRaul5c/fqPvOPTlvGqLn
0nRaLAMY4celsoSYqgLuN19MgcawJSehaNwISTQoiwtqWLWHbKYfiaGgGwjvpyEcdepyVE
z7G29RI2+IFOEsKMWWkETVhSrcv352IRbQtXoCvpQXrmPFEjZsU7hn2NdZTev9tGJJTyTjvCgJ
uJzXvYHGbME+YyYArpJCVI9xTngld8DRzuorzmHSJz9jDq3h3PbXn9S88xqGtAx0UTYcBSupf
OkZLKPGoTOB8dTXysrMpfTB2zGlZ+JrbsKYnomvrYW+FvB4aFvyN+0rlqCLicM2cSqmzOw+v4
8Q25C1Bldt99S+xRRFSQMaVFX9r6loTcBZoV3HAvGfv4earMTnNMMcOrW3LcnkKBvg/yONk
ofuD2cPAM0//oDrYv+2mgC3bpwPktPOJiAK/gNrtZmZ/Ts19HodRhS07HvtAvR02ay+urzSDvt
vG7nO1avwFsfnoFuumwROTFcRfvyJag+H8nHnkrLovkEvN7wH1ohRFAGtF6oouubt2RDajrONa
uJ3XXvPrmeEGJoC3i9VDz3eDh5BmhfupCWub+EE2i/20XFnMeoePrh8DFJR59M7k13hytKO0
uLWHbakhLQ6MhNRpGP/8WcbvtE9y0RGHJH03rwwkYUzPwt7bgqa3G2GnEmKe6ssv8Zwiur
OGurgwn0K6KMrKvvQNncQH+lmYSjziOlnm/462uRJczDEWnx7F6OZnnX0Hror/QxyVgys6NSA
90449fs/K8E8OFTM0jRjPmubcwZcgoOCE24Aa6zoEGclTat8Z44AFFUQKAFzgeAeIVRRIMcG
e5eNDx94GvK0oSiPwHcHCYX1GEuhtkK+1NVzYq7P0axT+U8Dnw7FmJcnHn0b9p+/jqanCkje
Cmvf+R83brwLQCDT+8j2Jhx6Nxti92JE5fxT62GCBj7id92DVxaeH+nwA1H36HiMemSPJsxA9
CDgdaExbX0BsPWNABu2rlvfZ9YQQQ1vA2U77kr+7tTvWrAz/7ioupOLZR7vsr3n7VVKOPw3bp
GkAtC1a0JE8AwQCIN57K9GTp6OLjkFrNpN29sU4Vq9A9bhRVZWew/7VZe1IfWlyGqOpSyVujc
mMISklvG0dPZ7Vl51JwNEOQP0XH5F7y30o5uBQcNXvw5icQukDt4fP0drs5D/w1Ba8Ohvmra
+j+K4buqWC4ly9gralCyWBFmIDvmsKvL5njAb6uAq3qqpfAI92bgvNgX5AVdVr/3Hsh8CHPVzjt
n9sR21o38ZlhZltkD4ugfgDDu/Wbhk5psfj3ZUVID5wO4U3XU7Nu2+Q9K8Tsc/cjZid96Dm3de6
HOtcvQJDUjJNV/5AyklNhZeU0MXEkN3lLZhC3x43/f5TOHkGQFWp++RdVL+/bx5SiG1lwO3qk/
nP65kysnv8Ek0IsX3SRceQ0MNKGPYZs8K/+52OLitdhNtDSSyAr6Wp235vXTUBVzAZdtdW462
vpX3xX6x96E7WzXkMd/lafI0dS1SZc/Ple+Dp8Lr3isFI3v1PYe5URMyxZmU4eV6v6o2XwvEpGg
3Vb3f9fOJvbcZTU7Whl2CL+N0uvHU13dvbWvv0PkJsa75rCrz+XVMg57umgCb0c6uS58FGeq
C3QRq9nrSzLsJdWUHTD1+hsVjJvuqWDS4Vvffpe6yb8xgAfo+H8qceluuqW1Ddrh7/mKKqtMz
9hYCjncxLrsWUIYvf6aR53q/E7blf8JDOyXNlWLPJZdWE2C4FnA40fVCBez1Tdi6OgpWofj+KVtt
n1xVCDF1JRxyPs6iAuk/eQ9EbyLzoamxTdwjvN2fIYM4biXPNqnCbPjG5S3Vs65gJwS/OO/XIJ
p9wBvrQkplBt4fW+XOp/fBtAHyNDZT9517yHnwmfLyiOZBw40FYR4/FU12NITkZc25+I1UDelp
FWfV50FmsoQMUVF8Pnyn6+P30mJxK8glnUPn8E13uYrNrC4eEEKJ/qaqaMxD3IR7obZQIN4
9Rj7/EpC/+YNJnv5J66rloo2zdjvO1tlL91qtoLFZMWbnhOZjOkkJs03Yi/uCjuhvxzMjGNCwfQ2o
63vq64NrMOj3u0iJiZu4WPj/xkKPCvdPrpZ52nnyYF6IHfgezX2kRW0prsaKzx+DqPNRSCLFdM2
XlkHffkOz+ci6Tv/ldjPOvCK+rDKCPT2TkeY8Rf+DhaG12YvbYjzHPv40pLSN8TNS4SYx+7i3MI8e
gi4sn4+JrSD7u1PXDkFE0Cg3ffQ6AMTOLXWhal7OwoEssikaDZfhlYmbuimX4yG5L7kXvuAvK
P4oqpp9zGTp7DACG+EQyLriyy36NyYt03JZfUHRakk77VzSz7scXUwsltHjGPPC00SNnbDpk4
UQ2yzpgd6Gaa3BdRP9blewJ6qHAKV+I4OUk8/Gs64c97oyEo88DseqFUSNm4R19DiSjj4J64gx
NHZ30bZJ0zCkplPy7xuJ22M/VDWAMS2L1vl/ELvHvth3mIWvpZmW+b/TunQhl/7zPHUfvU3A
4yX1tPOw7zCrhyiFEAGXs097oAFM2cNoW74I87C8Pr2uEGLo0prMWPJGbnC/zh5LOlEnELPb3
uhj49HFxbZrzEYiNtjX6KnzCDgdqFPTA4nzd6mFhidT4Tc24ejoIV6KJs6JNSMCSndbuXqqoE
nE40ZnOXawDYJk5I3OufUvnKbDw1laSeeBb2nffockzCwUeii46h6vUXMGZmk3rimUSN6fvE1p
SeRfZVt5B66rloTBb0dnuf30MIMbRIAr0N8LdeOvr0EbZ0Ed3vLF7mxpo/OErKI+ejT4phfSzlJi6
6o7hb3oDXi9NP31DzZuv0L58cfi85ONOI2bnPSI77D6q/vscurh4YnbdG31CEtE77IzWaMJTW4
195m5ETZhC0uHHhHuWK9+ZQ9GtoaWqFIWYvQ9k+N2PYUrb+PJZQmzPAi5nt96WrWXKzKZ
92WIS/zGKRAGheUJ3tLP2oTvDay8DxOy2NyMeeQ59TNdEWvX7UP2+4DSvTIPLtBYr9p12oeDy
jmUutTY7o2a/0eV8R1EBNe+8RuMPXx072z4kHX1icJnKEEVRIj4yA9vk6RAI9Dh6TR8TR+Kh/



yLhoCNAo+mWhPclRaPp+zWzhRBDlgzhHulca1ZRcNV5LNhzCstPPYKW+X+E99V/8TEFV5xL
26K/aPz6U5addChtSxeG97tKi3BXIHdJngFq3vkvvuZGql57HgBfQz11H7zJ2gfvoOnHb6j95J1gj
/OOu6CzRoX/sLkrKyh98M6OC6kqTV9/SvuSBZF7AYTYBgSczj4tlgahHuhO/70LlcTGOEsKuyT
PAE0/foOzsGNODMDjpv7rT1l0xF4s2GcGJffhquiY2lXZ2UFFc/+p8s1/K3NtC2cF972NNSx+vK
zqHjmERwrl1Lx7COsvvwsPA313WJSFGWTU78UrTaiybMQQvyTJNBDMlElmTU3XEr9p++jety
0LfqlZacdiaOwAG9TAxXPPNlleNXrpfWvjgRb9fnA373Yl+rzBYtZdCoSEubz0jp/LktPOARHwYq
u5/n9BNzd12AMeD1b+IRCbB/8Tke4lm1fMeUMC67D3tN/x0lI8Q+qt+dCnwFvx+eEtiV/s/LcE3
CXIRBwtLNuzmNUv/4iaqjgqOr1EHA5ul/D1fHZoH3ZYtqXLOyyv33JQtqXLeqDpxBCiMiTBHol8
1SU0Tr/9y5tAUd78NtjRZNaL3EzhSDCXDNNY2/f19jzUqid9wlXD1zvb9D8U8fCT2nffs0m5Mz8
LX3ASA6nHjLOpaFMSYmk7a6ed3adNao7COHLuljyEdiG4jFXfDuHWxyWgaDS4y0r69LpCiG2T
OWc4tmk7dWkzDcvH0qmOQvvvJaScCaAZl11PxxVXk3X5jdR98xne2moALFm5pBx/RpdrKDo
dtknTw9vh6tlaLfEqPAQ8J5W7xBCiMFI5kAPYRqLBY3ZQsDZ9dvegNuNzhZN1mU3sPL8k8Lt
Wls0tqkzaPz2cxrVqK1RmEZpWXY7Q/S+N0XtC35m+gZM7GOn4I2ysbwux6l9t3XqP/y16yJx2
POH0XZf+7puF5UdJf7KlotqaeeyEljeo3X8UyYhRpZ1yJX9UZf8IIYa4QAR6oBVFwTJqLC3zfse
Uldun1xZCbHs0FivpZ19E4/ARtP79J9YxE4jd+wA0nVbwMGUPo+qNl3CsXAqAYjAw7Nb7UUY
W8DG2HwARe8t91Lz3BrqYWFJPPhvT8l7CZbqYODKvvAn8fjzrKjCkpoFOh+4f86wdq1fQPPcX
vM1N2HfcGduEqX3+RSNA+6pINP/xC/62NuW77kzUhClo9Po+v48QYtshCfQQ5He78LU0o5gs
ZF1+lyV33xjel3jUiXgbG3CuLUaXmMzwux+j5a8/MKamE7fPQXiQ1lF0+zWoHjcAisHlyMdewDx
qHLF77l+3oRbHsOUse/Nl4vY5kMSjif9nEto+PlbVI9wSvG+MbvujXVU955lY0oqaeeer/LRJ6EY
TGh6qPwthOgq4HRG5A0bJW8kzXN/IemoE/r82kKlbyurpJBVF52GITkN6+hxtC36i9oP3mTihz
8QNX4yAN7GepxFBcQfcBg6eyxNP39H9Vv/JSFUrNDvdLD2gTsluJ1kX3sb/tZWVI9xDsPveAhz
xnFAcKkr5+qVuCvLsY4cS9OvP2BITSemU5Xt9oKVLd/9KKKn7IA2ysbqS89k+B0PE7fX/n36zO
0rl7HkuAPxtzQFGzQaxr78PjGzdu/T+wghti2S3Qwx7SuX0vz7zziLCqj/8iOyrryFrktuleB0YBk9ju
bfffLssXuo+/BNEg4+iorZ/0FjMtPU3op95m7Uf/5hOHmG4FDsus/ex5CagTEhCUNaBkU3XwF
A619zafr5B0Y+8TJxu+3D+Le+wFFUgD4ukajxkzAkJG0wTq0lKtlvhRDdDL+jvc+XsQKwjBpLxT
OP9vl1hRdbHl9rC6rXi7u8FHD5aUd7S3P494DLSfbVt1Dz7ut462qJP+BQFK00v6MdnS2agMO
BJW8kxows2hb+BRoN0dffhb+9vdp1WogaNwl/azN1n75H1MSp2MZNxt/agJ6mdcnfpJ12PtX
v/BdfUyMJBx1By4K5R084C521o0d8azX9+kNH8gwQCFD25EPYpu6l1mTqs/sllbYtkkAPIZ7qS
grvuA5L9jCq33w52FZZTsXsx7CMGI2zaA31n38AQGtDPW3LFpF54dWsfteTfANR98h7eHqpc+l
uaaSkpwlmwkoTDj0Fri8bf2gJA86/f4ypZg23iVCxjJqAxmYMFj6TipRB9xtfWgsbUvWbB1jJlZuO
tq8FTV4shlBHPry+EGFqcZSU4ClaiMRixjByLmbHji3B9XAKGtAw868rDbdroGPRxCR3HxCdSeO
NlweWrgKpXnyPtzIsGNIJGFxePbeoMat97g8Yfv0FjMpNy4pnE7L5P+BqKTklff58L12do+vEbnE
VryLv38Y776nQU3nNTeLvypWdIO+ti/G1tXRJoT3UVztlitBYLpmH56CzWXr0evs7J8/q2xrrQPG
1JolUQPRuwlMKKopQoirJEUZFfiQLMD7XFKYrytaloBaGfsZu6zrbMXVlBw/dfUf/1pzhLinCWIR
l1cgx1n38YPqbq9RfJvvoW4vY/IPqvPu5yvurxol22k3j4sQA4S4tIPPRf3e4TPWMWbcsW0fj9l3ir
KrHvuEvXD/OKgqe+ltK7b2LRobux9NgDWHzMfrSvXBazBxdio+NvbY1IAq1otFjHTaTpp2/6/Np
CiKGIzeF8Vpx9HCvPOpblpxx00W1X4ljTsUSVp6aKzluvxRoarm3OH0XuDXfiCRUlg+BSV+vXf17
/nlX7wf8lHl509zY20vTL9zT+8DWoKgGng3XPY63al34GgGXq1txQ3dZCQGXq+M+xYXd4q//
/EMldKwq0L5iKYuP3pelxx3AokN3o/T+2/DU1/bqNYnZeQ/4R4dA2hkXolvqu15ulcS2Z6B7oPd
QVbWu0/Z1wLeqq6rKmp1oe1rBya0geUsLmTFucfjDP1x08XGMerp/+JrbklfF4chKrkUBWfBS
kofulMxr3yAzh6D7x89zL7mRlZlpSQdczJRYyfia24i++pbqf34XUAl4aAjafz+K/D70cbGYxkzHq3
Vgil7GPq4eFwV5Zzh82iZ9xtVr78Qvq57bQnlTz5l3oPPol3A0FMhtif+tlZ09pilXNs2eQZ1n39A0
pHHR+T6QojBL+DxUPPWqzhXdyw/2fD5R8Tuug+WvFCBL60WV3kppowsYnfdC3d5Ke2rlmH
MzAmfo4+JJ/38K9DodARcLnSxcTT/NRd9TBwAqttJ08/fGU5Hwv6H4a6povXPX7us2qFsoN5
D53Z9fEK3/frEZHTRweKlfpeTtY/e02WoedUrs4ndfR/ldt93s18X28RpiHnhHcoevw9fUyNpZ19
C3F4Hbvb5Qojt00An0P90GLB76PeXgR/YThPoxp+/CyfPAL7GBmrefY0oqTsQu9f+1H/+Eagqy
f86icZfvkOfmEz21bdReP3F4XMSl8fgrauldf4JB9zCq6yUqyx1H6wO1kXXET7ooyyp96iIDTgX
XMBNlVJkQv+egMRqJGjeJdS8+Te5N96CzReMsXN0txqbffsDf3lg2KaVfXhMhtlX+tlaoPSw7
1xdsU2ZQ+cps/E4HWRnI0yclbY5nroamv/4qVt757WXNVot6+Y8hurxdBygKMTtc0h405w3grJ
H78a9riy8f8Sjz4W/ANTFxpFy2nkYYuNo+PpzLMPySDvtvPA60QDaKBsJhxxF3cfvhtsSdjKna1
jZQ/7TritSEnDs77nWqsl85Jr0VqD9VV8TU00//Fzt+dxFRd2flrcDBqDgdj9iZ6+o6oPj+6aPvmn
yyE2G4NZAKtAl8piqICz6qqOhtlVIW1MrS/Ckju6URFUc4BzgHlysrqj1j7ncZoIOPCq3CWFGJMS
UMXG4/WZsecmcPyM/4Vnn9U/8WHjH7+bdxrS9AnpzFq9hu0zPsdrdWKR7mJypeeAYIJePmT
D2AZOZaM86+g+PZrSD39AnKuvwtvcyPm7FxWX3Rq+P71n39l9lW3Uvrg7dh32Bnz8BHdYoyZ
uTta+3Y9yl5sB/rj/cbf3orGFJnkVhdlw5l3kqYfvYf+/0Mjcg8hRN+I1PuNxmLBvtOuuEqKurRbx0
wl/676fF2TZWbV7VJ4tH35ko7kObS/7MkHiZ4+E2NKGvj96KPtrLnmwvAhtR++zejZb4S3/W2t
qd4fWVfjK+pAZ09lvYVS/CFhoEDBDxuko85BbQaVK8XrcXasX40oluNxT5zV7w1Vdh32g1/ey



t1n76HKWf4Fr0+UvhUCNEbA5IA76yqaoWiKEA14qirOy8U1VVNZRcdxNKtmcDTJs2rcdjhqqA
x0PD919S9sjdmEeMJnryDJoXzCVuj/1Qk1NxlPwQccGV1H/2QXBllKpS/c5raHQ6oqftiCbKhrtq
HfWfvod9h53JvORaUFUsYyYQNWEKbQvnox5yNL3PIRGp0MXE48hPYN1sx/rEofq9elsXI2i1e
NraUlf0jy8adT/caLABizcsi48CoZvi22ef3xfuNvb0MTOR5oAPuOu1D1xouSQAsxyExq/UZrMB
K7y960/DU3Plw7dp+DMKSkho8xZmRhSErBU1PvcV6Urcs68gGno9u1ffW1BFxOAdy1Nax74a
ku+wNOB60L54eXhtLaomn+/SfqP/8QjclMwOVEFxtH2jmXhs9p+fM3yh67F7RaFK001ePGnD
8K+4xZ6KLtal0mMs6/nPKnH6H8mYfR2WPluPBqrGPGd7m3qqq41hbja/FkJSCKSN7C19BIY
ToMGAJtKqqFaGfNYqivA/MAKoVRUIVVbVSUZRUoGag4utvfqcDR8FKnMVrcJeVok9Jl+Gglym
8/mJy7voP+vQMSm+4BE9IBZbR48i85Fpc5WshEMDb1IB9p10JtLWhjbaTdNjRxO97MN7aanx
NDfhaWyh/4gEyL7mOtoXz0NujcRSsovLIZ8i753FqP3obAv5uMakBP4mHH0PxPTfT+PWnRE+
fSd6Dz2BMTsWSPwqDDN0Wok/429sjUkRsPfvM3ah+61UcRQVYhuVH7D5CiMHJ73JQWVNN
yo33oKmpRDEYcNhiqK9YS1zoGGNKGiP/+yH+qkrc5WsxPKRhSEnDnDMsfB1zbh5oNOFRcAC
JRxyHaghVrNbpwN/980Tn4l+q203mzfdSUVtDfX0d8fEJpCcmobo79X6v/0zi96Ouv57PR3Dwl
qh+P/Wff4RI+Egs+aPQ6PS0/T2PqLETMSYHvxRQAwHqv/qENVefj7+9DZ09lhH/eZ7YXffaqtD
SCCEGJIFWFMUKaFRVbQ39vi9wB/ARcCpwb+jnhxu+yrbD73JS+cpsSu+7Ndig0TD8wWfwVq0
jauqOmEePxbWpiknnok+PhF9XAKemipa/vyVpp++JeuqW/FUrcNTtQ5DUgqWkWMoufeW8Bx
qY1omycedSuN3X9Dw7efBJHzEaNLpuRS/00nLvN/luuJmWub93hGUVkvMrnvhb2uj8etPAWi
Z9xst835j1LOvS/IsRB+KdA+0xmAgdo99Wff8k+T9+9GI3UcIMTgpRjOJ2dmUXnk23rpgpWrr+E
kkX31b+Bi3w0Hbrz9QfMd1oAYT1cxLr0drj8Gcmh48pqaK0XPexFVaDAroY+NwV1cRaGsG0kG
rkHLKORTfdnXHvQ1GoizMDm972tsobmlF43CQotXgdDgoaW0lr71jHejo6TMZ+/qnuloL8Dud
wWlkBiO66BgAvHW1GFMzKH3oTvvtwXWq7Tvgrexo5Cqs7iA1ZedhaLTYR4+Anf5WlZdcaT
Pv4JU6b0RAshxtxA9UAnA++H1hLWAa+rqvqFoiJzgLCURTKAWOGaD4+pVzzSpK778tvG3+1
0n8VVXN6uXL2Pf8K3Et+ZuSO68L77ftCvxBx5O2hkXYZ00nbWP3EXmRVdT/uSDwf2zdsc6Z
nw4gXavK8NVVoK3rhbrqLF4KitwrF5BzK57o7GysY6ZQP0XH5J97R00//YjWpuNlONPxxzqHE
sO271bvG1LFxG/z0ERfU2E2J4EHO1oi9gDDRC/78EUXHsR6WddjDI3y+YJCiGGJldrC41vvhpO
ngHalyzEv2oZ7LxH8Jilf1N6763h5Bmg7PH7iJ4yl5xAm3PycKxYRMDnRfX5UX2+YHKr0QKgoI
x462vJu/9pnMUFKEYj1hFjaC9cTexuewfvG2XDuuQv2t97g/bQfaxHHo9zdMfwa7/LxZqrz8dds
RYIVuge+cQRHQ+k11H7ybvH5Bmg+Y+fSTzyhPC2p3ldKcefhSzwVvArPx+h+Kpq8ZTUykJtB
BiqwxIAq2qahEwsYf2emC7G1vjdzlJO/MiGr/9HFiz+HRdJcaqavabOJEUahQ8dFeX45t//4nKE8
+gbckCPOvKyb3hLvdy1k9s/vUHsi67ocs5zjWriN3nlGr+93K4zb2unNjd9ib31vtXlrbhLF5DxsXX
YBkxGn20nYDXS/QOs6h9/39drUdOSYCr4IQ2ydVVF7HREdwg2gs8eQcNARFN91PWOefyui
9xJCDC46n4/2VUu7tXcuKuZtaCDg9RCz295EjZ2lo2AlDd99gaehY7XRgNcNW3N33+Nt76W2
D32xZ+QxPqUWyWAKpaJq62Fph+/QRufiCF/NibkjrnWutYW2v/xuaL9/f+RcNCRHdtL/w4nzxC
sy1L+7CNYx03ClJqG6vXiWLOCy6ixxO62D76mRuo++wBfc2PHfRKTaFu2iNb5f4Tb4vY+EH1c9
yWyhBCiNwbbMlbbFV97Gw1ffETJfbfib28j8bJcM/anZEL5rP7jrPQ6bQoBmOXb1jXc5etpWL2
YyQfcxLNf/1Jwv6HdNkf6FStEiB6xiw0Gg2utCxnSOMWWjHTMQWG0v0pGnd7qHR60k/+xKa
5/6KJ1R1M37/Q7FNmdEXjy+ElFhgR2MwoGi1Eb9X/H6HUHjDpdR/+THx+x2y6ROEENsEn8GI
bY/9MOcWETV2AqqQ0vjD15gmdvztN6RIMPYuR6j96B3Kn3mEqPFTGH7nl12LiDnaqXrjZeL3O
gBFq8FRWIAaCiSreWss0dDWSvm/O77Eb/39R/LmvBne1vl9XXq5AVDVYHult6mRf/JUrcMXG
uatj0sg59rbaZn3G+tefBp9fAlZ51/eZai4v6WIS/IM0PDNZ2Reeh1CCLE1NAMdwPasbeE8Cq4+
H29dDQNgng/qvPiZeUZjqddH48Tus0OtY1s15DPvM3bqcpzGa0OSNJOHYk1n3/JNY80YEv0U
ODolHYzR1FPpQFBI0OYqYXfeiZcnC4AW0WqK00J6Pfv6Rhb/8sNEYraPGMuGdrxj7348Z/87
XDL/n8eBSFUKIPhGc/2ztI3tpDAbSzrqlwpuvwNvY0C/3FEIMPLfLhXb6zkTvuAsaik5P0tEn0dR
5bXgNID/zCC1//AyBAG2L5IP6wG2ooc8WAH5HOzE7zKJizn8oe/x+3GUI6OMTUNVgoS+Pz03
ta891ubfq8+FcvCC8bU1JxxAaEr6eITuda0pHmzk3r9szxO93CNpQrQhFq8VZVEDtB2+ietx4Kis
ovf82VH9HcbOAs73bNYLP0L2SuBBC9Ib0QA+gpt9+6rKdetr5IN52NSnHnkLVf4N/gJp++pZR
99AG2Wj6cdvMA/LJ/nks/n9z9+xJqUTnz2MlVl/kHzMyWgMRkx5I4g993Ka//yVzluzQZeaji41g
9qP3kZ7xHE0Z+TgD6i89sUnrFy0gNaAn100OWKjcRpT0iRpFiJC/G1taM2RWQO6J9ZRY4meP
pOiW69k5GMv9tt9hRADr6fTEO11U/LS03jWlQMqPX0n0s67PHyMr74e99qSLuf5GhvwVq8Lb
yuKhspXZoe325b8jSEtj5jd9w3ud7t7XNNe02mEjd/jJvva24PXrq9FH5+ILjYOV7tjKprbYiXhrt
pflZfM2NWA4+Cu2MnQm4g2tSe+ttqHnv9W73caxYQsy00wOgtUZhHpYfXPlzXDZIRkSmy3gb
G2hfvhhPdSWmrBwsoyegs/bPF6NCiP4nCfQAUUVVnT2mS5uSmlwhIRHHquXhNvPwEdR/+Q
me6kpSTjqLgM9LW0khOzueHW67ntqusxFK5CDaiknXkhFSkZBAIqP1RWkxJQ+PHtN5m19
wHsYo3C9c2nvPHFF1QUrQlFpyt/VH89shCiB/721ohW405J8jEnsebGy2QotxDbCWNAPERLj8L
JM0DLvN+J2/tA2GM/IDhtC6222zJUmvVLVAHu6spu127+/WcC3tC0Ma2O5ON0oeTum7FO
nk6grRV3WSnmEaPDx/sd7bQtXkd9Zx9gHD4Cd+Fq4g88HHtMXPiY1TU1PHr7jex36FFER9v56r
OPySst44q77g/dR4shNQPF0fSRNnCVyp6A4mHHo2rvJT25UuwTZqGNIyWjdHQuxdvE3ytLax



9+C6qXns+3JZ76/2knnw2ikYGegqxLZL/svuJt7EBT3UVamjeju0iIDDgSetl3yMao/BU1uNKbv
rmovtS/+mbcnftOaPYI50HP/59hs++u0XLn/kGRwqJBxxHNVvv4ovLgG7PQZH+Vo+fft1nn/iYd
YsXcziv+bStugvPlv/ZtSEjtp9vgEdj54473PQojlivQSVj3RGE2kn3kRhbdeha+tdMnCCGGtIDXS
/vyxd3aXWWI4d9VIOWEM7rsTzss6C5DuANx8d2uYR4zDq9OD4Bfq0Ud04n2G+/hxeY2PolXn
D/0+g6DdnWarU0GUwsnrUXz6yrZvGsvWgymNCGrgFQXIJMe0sL7/33RV566lHWfRey/K+5OH
zB5L61uZm4cy7FIJuH9YQzsR58JKYx4/GkdnymlIWN5rUltPz5G4bkFBp++AqXx4028zF9wFG
wskvyDFB67y24Sgr79D5CiMFDeqAjzO920fTTt5Tcewu+5kZSTz2P5KNPwt/eRvnTD5F2+gVYx
k3CmJgcrMZ96vn421uxjBqHY+VS2pcvIXbP/TDsvh/fzf+T92c/CcDyeX/w66cfctdr79GsN5B2x
gUEXE7UFcv48NuvusQwfco0HB++Seqp53Lqlf9ix4OPRKPRkDdhMpl5lwbIZRFChPjb2iJegbsn
1tHjiBo7kbUP/5tht9zb7/cXQvQjs5nY3falsvjpLs1RE6eGfw/o9dim7oBlxGg81ZXo4xLRJyaB0R
g+pk5nwDRzd1y//QCAzh6LZ59DaHe7iAMMislP8+by+HWXhc/55fOPufu195k6NvgFvkNvef7T
j1mxKDgveuVff7Jg4hRunLUBMaFzcnl6CpetN2OX3THpg0l2u8vJLwUFt06401+89jKxSckcd/G
V1K4tJSd0/NqSYj5cV8kehX+H4nHhmTidV7/7mtNLisgbP2kLX8ju/C3dC70G3K5wwTMhxLZH
Eugla1u8gJXndqxLWPblv9GYzcTtdSCxex2lbcasYO9wZTn62HiiJk5G0emx7rwHvpYW/GXFGb
KTaTVZ+eyM47peu7mJtWskyBo2nFY0JKVMEkiolKrT0PBdDzqcfDWAcelUko44DnPOMDIka
RZi0PC3D0wCDZB87CmsufZCUk86C/Ow7kV7hBDbBo1Oh33mbrSvWEzL3F9BoyH52FMxdKp
vorFEUfrYfV3mDOsTk8mf/UZ4W1UUVkbHTHdegy7gZ63DSf3yJRwbmgPd4nLywXNPdbm31+
NhxalFTN3nAACqHc5w8rzeikULqHY4WR9NqlbDv045k/dee4mA38/oSVPZZ9oOBDzB0dAavY
Gaijl+fuGZ4H0bG7j/4rO55cWOat9et5tvP/ulbz/7qMu9TgrNo+4rpuxctNYo/01t4TZz3kiM6VI9e
h8hx0AhCXSEtS6Y17VBUTamp1L+9M0knX4+zqICvPW1aAwG3GWIKBoNjul1NNVU8/DXX3H
mORes8PE7a147A62u+/9cOr2ev3/7mTcff5Cb732UjJgYztp3f0478VRqCIYTp0Dy3gdgzs1D12l
ukBBicPC3tw5YAq2LthN/0BEU//sGWRtaiG2Yz+NB0Wiw5l8iZuc9UQN+fE2NBFwdFam9tdVoo
2xkXXVLChK9k5nmX3/AVVkoOZ7qFJ+XSbvuzumP3k1LQz37HH0CB0yYgN7tBECj1aPV67vd
X9epzR0ldNsP4OnUrouyMa00gOnX34JPVTGXFMJwTOTH+Yvq/DTh+90OV9VWVRKO9aOTH
+eR1rOMNZ1Wut61NTppA/P39yXbbOYc/MY/cLbFN5yFc5Vy7DvtCs5N92DQdabFmKbJQl0h
Ok7zRcy5eaRft5INC/4k6QjjqXu03cxZw/Dkj+K4juvg9Afj/gDDyf1Xyfh+uhDspubqZ+yE7qSQo6
9+CpevPvW8PXiklOITurhl08/4Nobbkfz6F2sDRXUMKZnkXLkcQR8PmzjJyOEGJz8ba1ojKZNH
9iHmuvrKS9czcjJ04nf9xDWXHMBLfn+J3r6Tv0ahxCif2hVlerff6QuMY3CyiQMBgM5UTYstTXh
Y3wmM/addmXtg3eE21JOOAPV2vHleyA9kx9efZFdDjkSa7Sdhh/8gPfgwwisr7LtcXHGSWfw1l
1Xhs8xR0WR2qm2iz0mhqm7781fP3wTbpu6+97YY2l6jknPxHzMydR99DbepkZi9z2Y6B12xhu
aj60nQFxyCs31dV2eM7pTPYm25mYOPPVsChYtYM3ivxkzYycy80bQ2thldGwcfck+fSbj3/gUX
0sThvgEtFbpsBBiWyYJdB8LeL20/v0n1W+9iur1knj4scTudwgJ+x6Mp7oSb0Md8QccRtvC+TiL
1pBw8FHUffle5uxhOluD1bHrP/sA65jXPPgE9x10Zm0hJLimQceyJVPs8fX3xGZv4lskeNoaG
6knVfa7DHx9DeqRqlu2ltfk7iUed0GOCQojBwd/WhqbTHMNIa6yt4eHLz8MaFc2Pce9w7l0Pk
Hjk8RTfezMT3vkapVPBICHEtiHgdIGTns0tV15EIFRIOzo2jrseflyU0DFaRc06F5/COMYC1rETcK
5ZRdUbLzJ8/OPD11ITXMjfv/2E2WJG43RQW76WfX99gJuefpFowGS2MHXYcG6+837+mP8n
sbGxTBSzjqT4js6EWFxltMOPYocp01i2chlJR45hdHYOMQE1flxryd+U3Hkd9p12xZl3kspXn6NI
wVxS7/oPABajiVPPvpA7rryQQKjzIT07l2FZ2eFr1Fet47nbr2fy2PHkT5zCyvlz+fp/rzJp591JHza
8z19jfUws+pjYPr+uEGLwkQS6j7UunMfSEw409ybXffwOY1//FMfyxbTM+w1fezuG+CSsYydiz
Mii7PH7cRasJHb3fUm/8CpcxWtwlRShqrBmycJw8gzW22cf4Xa0c+5VN/HUPbdRUbyGfY8/lb
wx41E6VdJcz9vYQNSocf327EKI3v01taDtXyrcn736PKOmTGf6nvvx0QvP8OdXn7HjvgdQ//mH
NHZ70fF7H9hvsQgh+kfAYObdd/7HyAmT2Xvv/XB73Hz4/tssLy5m/acEX2sTmZdcR/uyRTT99
C3WsRPlue50vE2N4es4nS7u/feD80m7BMqK2PW4E1jQ5sDv9gCgBvy0PPslhiV/s9/wEfjXLM
X90f/Q3nl/7LIHAHqfl6Y7riEjysbwjCzc775KU1srKa98GL6PZ10F+P00//J9uM1VXIjWE7yPJSW
VkSklPuuByhrqMdiNfPjyb00rH2ckJqOnqDgaJISyhatgQAq91OXEpqRF5jlcT2QxLoPta2dBG
ZF18DgKe2hphd9sDf1oK3sR5DUipRyamogOrzUXjdxfbWtHfXWPJH0X5Y/fhWluMZdRYEg47
mvbvul2/frqKkqrKpi025401dbicrSTO2ESWo8TFnctyhG39wH98chCiK3gb2tFFx3TL/dytLWY9l
9fOfGK69FoNOywwzWf889Zr7LDvQSdqfRKl991K3B77oawfjimE2CZ4CLDLrnuQU1ZE+0tPojW
Zue7YU6nqtPayJSmFkofuxLFqefCc6kraVyx+GMvho8Zn5WJ69vPiD37YIS3G7/bzU4V5ehCiav
W68XbUA+qinPNqvB5voba8O9utwslvvc5Vi7r1O4M/24d17Hk5npx+x6EJZT8ao0moaaPJL2s
hFSbFUXRoLVFeZ26o9MgM38k1z3zMvdfEaZupxNrdDQ3znmNtJxh3a4thBC9IQl0H3KVldL47
Rc0h5Z3sO+8O+2rU9EAik5H3L4HobVFU/H0w0SNm4Q/tP5qynGnsfbhu8lVHB0rl1F8+7Xsec/
jvPbUo13uMevAQ2mqq+Wl0FzoUVOMmee265n98beYHA5qXn8eRacn44lrsC+Y1V+PLoTYQv
62VgzJ/dMjsmzu76Tn5mEKfdhNycpBq9ezZsLC8idPp/6z96l57w2Sjz6pX+IRQvQPvUbHGFRq
P30fCFb/b3vhSUY/PKfjoJbmcPK8nqeyArWpYyScWaNFN2Yirq1qH4fGksUKTvm7ChGZjCQc
PCRVDzzSMdFFAXLiDhHtbfiiErF8/a4nCbISsXtyUqvN0Wm0DGVbdQ+eyj+NvbiNvEay77Yve
Fh0+xjpiFPryOFxri9BarJhy89B2Ksio1WrZ7fCjyRs/icbaGuJT0rZ46LZjzSqcxQVobXasI8Z0qW8



jhNj+SALdhxq+/TycPA00L1tMwoFHUnjz5RCac5R83KkEfD401o4/FChKI+UPIDiH2eLzcPkjz/D
R80/R3trCQaeaja01BdUfCJ2moDMEK1KuWFPATf+m/TTzgOtBIN6FopGE9kHfKJsNX9b/1X
hXjb3N7JHjg5vK4pC3vhJ/P3jt4yYOJnkY06h9KE7STjkqC4fRIUQQ5vqctD25Ufd2h0L5sLhwx
g6PWgKKCqXY5R1le+JpiUVn36Hi3zfg9uW6Plve0BDBnBuceqzoh17ATSzryl+s8/RBcTS+IRx6
Lv1NPd0tpG61EnEbt4Pt7FC9BPMELjhGkY2zrWTF75m8/58+cf00fm+zAZDHZ53df8cuGZPPP
Fz8RI5YSPMyQmYUhm2uBzazQaMvNHkpk/cvNfrH9onvcby087ioAz+CVB3L4HMfyORzAkJW/
xNYUQQ5tkWFvA72jH72jv0uaurqThm8+6tCUedgylD94eTp4Bqv/3MklHHo+i1WINVcfWdPrjtJ
5iMKKx2vjlk/dIG5bH+J124Z0nH0ENqOF1ng885SxaQ3OTYhOT0ej1mHOGYc7MkeRZiCHC21i
PrLOvSqQEAioFS/4ml6/rB8lhY8ez7M/fUVUVy4jRWIbIs+75JyMejxCi/2iNjvTDui/ftD7xhWCinHj
YMV32x+6+D1qzJbztqawlJ88Q7MmufuNfDn7Q2sptzdR9+j7GnGGkXH0LSWddCBoNzQvmh
s+Jt8fw5/w/eb6wmPk77MZza4r4868/ie9UhTsxLYMV8+dy+Vknfc4px/DBS3PIHp6PvnPnQz/
wNjdTfOd14eQZoOGTr2lbrBf4xBDCD6SZfWCr72d+q8+YelJh7L0hIOP/fQ93LVVeBvqKbz5Ci
wjx3Q5XmuNwtdQH9425eYx7K5H0Ccm41i9guipO5J52Q1oY+NIPe28Lufaz76EI195nhl7H8D
SP37l6/+9SnN9HR/MeYLo+AQuuu9RZuyzP+8/+zgTd96NkZOn9strlIToW77GBrT9kEBXI5ViM
JqwdfqQChATn4jeYGBdUSEASUefxLrnn8DX3NjDvYQQQ5EaCJB2whloozqWVzLIDidmwpTwtr
e2moDHTd6Dz5L/8BzyH3oWXWly7oqy8DGeTsO513MWFuAPVdDWmswoo8fzR2kpt953Fw/
NeYZijw9jp2WsWn1eDj/gQA496DCMRiOHXW4R+x/EK3eJs6GkeMmkDd2fEesFgsnXXI1PrXn
NaQjxd/WgmP1im7tnk7Lfwkhtj8yhLsXWub9ysrzTgxvr774dHJvvhjdjZjZNP33DyCdfxZiRjb+pA
VdZKaqiYB4xGufqFST96yT0CYmU/PtGFJ20IJPOwjpqLIU3Xkb2zfcRGJZP3tOv0bS2mMr2dp7
98D2W/j0fg8nESVfeyAv/vhKAl8NB4dJFvHT3rVzx6LOcet2tTJi5K/FSVVKllcnb2IAu2h7x+xQvX
0Jqdm6P+zLyRrBq4XzSh+dhTE3HNNvHyp/9DznX3BbxuIQkadVAzStXEr+Q8/ggapC0evRxyf
gXrMKdtwZAENaJvYddqbq1dm0LfoLy6hXJB93apfkV5OZ0+3a9j33w2cMTvklKAordUaeuP+6
8P5Ff/zKw29/GI4uy2y28NXChey+w0yGJScTsNr4Ye5v7DdybEe8S/7mqosup8rnX+Nxk5aYTO
B/LxA1o3/XqtcnJBG3z0HUH+aOr2f06ftlsIQQQ4ck0L1Q/fZr3doaf/qG2L00IO/eJ2iZ9yuVLz6N
6vViGTmW2H00InrydNY+dh/mvBGU3ntL+LyKpx8m+5pbSTzqeDxOB3VJqSwoWMXL996Os6
1jPrTH5cLn84a3x+04i5V//YnOYMDIaGfK7ntJ8izEEBVwu1E9bjSdhkhGStHypSRIZPW4Ly13OK
sWzGfPo44DIPHWYyi88TSLsz7wlfXxCxGMTQkSWotViSs+iYvZjtP41FzQaEg89mrgDDus4SKN
QMedx30XBZTEdK5dS9ujdjHr+7fAh7fZY4i64kqYXnybgdGDbafV8u++L12DPsMvn5Z3ZT3S5d
8DvZ/mCeUzYc18AottaGTdyNLfcdU15WUkZWRYyZU3EN3aEj5Hp9VQceOlaAEz0Eiwx7y/V6
nXGo1kXXo93rpaWub+gtYarfb1d2ld271KuBBi+yEJdC/oY2lxZeWSdMzJBfXONDo9xoxsLGP
G0/LHT6yb/RgpJ5yBISmFgM+Lv7kRXWw+qRddQ/UzD3e7XuuCecTstheLqpp56f67uOzhpzFb
o7ok0AAmswWLzcaMvQ4gb+lkfv7oPW6a8zr8fEb7FESQgx+3sZ6tPYFYFCXyHwvXrlrOnkcd3
+O+9Nw8vn/3Tfw+H1qdDkn8ltHTZ7Lu5WfJvuLGiMcmhlgs1R+g5bcfg8kzQCB7QdvEjVpWv
gY77qycPK8nq+pEU9ZCUyeDoC9rQXvrN2Jzs0DlXMIlwtNqWmmnw8ALQrWqO7zIl26jo+bWp
MRzcN3cN3hx+BNTkNfvQ7Xw3eg6ZSo26fPRNHrUb0dHQgZ51+JLIQDpj9Z8kYyevYbuNeVoT
FbMWVm98t7thBi8JI50L2QfPRJpJxyNmsf+TfItZ6l3+XC19RA1StzQFFIOeEMPNNWvtC6cR81
7b1B40+X4GmpxtTVjysnrjd1DShr+1AyeelRasrLmP/tl5x63iVdjklIS2fcTjt3h0PKDZsGGaLaS
MTBSNQlb+qP56dCFEBHgb6tDZlj9829nWRnNDPBhJKT3uN1ksRMfHU15YEG5LOPBwqv47p
9sKAUKIoUchQP08X9EYTDh33gPblBmgKDg6rdWsi4ID0XXvV9F1HoVii6bmtmsouf8Sm6+gp
Kzj8NiNqPzh+ZAqyrHn3ZOI/NtMbGM6IT93zpmApmX30BsXDzRhSuJjYsn8/lbsY7pmPMcNW
4S417/hlSD/4Vt+kxGPv4ScfsctEXPrqoqfpdri85dT2eLxjpyLOasHEmeHRDSA90bppzhrLn5Sjlu
vgb7jrtQ9+Fb1H74JsPufAj8Kpb80TR8/Skt8/8gdrd90MfGU3r/bcTuvjFRU2fQ8NUn+FubAdAnJ
GKZti0333o9LY3BohwLfvqOw+5+kFufe425P3yDPTae+NQ0KorW8PDI57H3MSfy88fvkzF8BD
UVZSRWrCU2acPLNwghBjdfQ32/zH9eu2YViWkZaDdSnT8105fCpYvCy1wZU90xjBpHzXtvkHr
y2RGPUQGROW5/gMQjjkCBGn/8Gq3VRu6N/0Yb2yk51htlOeksHKuWYx0zAcealejE1GmpvAhg
XXluAo6imqpPh/VTz2l+dEXANDodYxKTeOOux9i6eqV2KJsJmN0JtHcsSye3+mk5c/faPjqk3Bb
3L4Hk3Dlv9CFipwpGg3RU3fENnkGqt+PRq/foud2rFIFzbuv0/TrD8TtfSCJhx6NOWfYpk8UQoiN
kAR6M3gb6nGWFYPeSM61t1Mx5zHWPf0IURMmk3//UzTP/YWqV59DY7GQcuypaMxmqt94k
agJU7BNmY7WaqPknvpJvvo21IAfrVEwZg/jkcceZPWSReH77DRrV6puvoLovQ9i5fw/aW9p5s
J7HuH5u24iJTuX3Y84msz8kZSuWsET116KNTqaJ77+vcu6rkKlocPbUN8vFbjLVq8kMS19o8e
kZuWyZvHC8DxogLi99qfy1TmknHSW9LoIMYQZLFYMSSkUXndRuK3ptx8Y3WnYtL+5Eevo8Ti
LCqh67XmiJkwmZtZuXVYT8dfXdru2u6IMQkt7+gMBFLudHK1C7pgxBAIB0Ch4DR3LdTpWLO
mSPAM0fPUJKSecgTFxry7tikazxctyumurWXXxaThWLQegfelCmv/4mVFPv4beHvkvLoUQ2y5
JoDehbekiCu+8joR9DyJq/GQKb7oU99oSAByrV9L8x8+UP/lg8ODmRtY+ejfZ19506/w/aFu8gK
RjtQf5yUL8rS0U3XIFAAlpeT/4rH5KUIYtGoyEQCDB91z2YGheHc105sSYTp1x3K3HJKej1ei554
HHiU9K4aJ+dcLS2hmNrb2mhaNliSaCFGKK8jfVdlpWJlJKVy0kf1n0aSWepucP56eN38QcC4Z
5q65gJBFxOWub9jn3GzljHKYSIEJ+Hqpdnd23z+2mZ+wtXuwaTVn1sAsW3X4OrpAilj9/w1IU



wKhnXg+fYszM5p/i9jkIncUavE1AZXXBkoZID8Pb2IDGYEQxWygsKSZ15m4ABNw9D6cOuJxb/
ZiduYoKwsnzei//lYrZA36ibL0pxBiy8kc6l3wNjVSc00FxEzfiYDHQ+vC+eHkWTEYSLvgChq+/b
zbeY6ClRjTMgEwpqYRtdf+RO2xH2g0GHOHM+w/L+BZs4Jj9j+lx15/n3tuvJ3j4uNwPvcYKAq2
nXZlZPQdmbDTzoyetgNTd98bi83WYw+QbguHNQkhBp63vjbiCbSqqqxdvZLkHj74dma12TBbb
VSVFIfbFEUhd0/9qPrvnIjGKISILMXvR+nUC9xpT/g3T01IOHlez1tX22UdaH1iEIlX3YIhORXFYCD
xsKOx77JXeO60LSaGBoeTI197hZVrS1m0ehV333cn9k6rhRjTMrGM6PrFv2XEalzPa8SsKV6
ms+9sXYhhNhckkBvhLNoNYaEZMwjx+BrakT1+Ug+/nSG3f4gw+96FH9zM4bE7kV59LFx+Fpb
iNvnINzVlay98GRMo8aQ9/L7pF5wFVqfj7L7byfgaMej0RAfn4BSX4N95m6MfeUDEnfalaT0jC7
XTERP5LTrb+/SlpiWzvBxspSCEE0Vt740XYSHcDfW1BAIBLDFxG7y2NSc4DzozmJ23oPGH77
G19wYqRCFEBGmoJB2+vld2jRmC9HTO9ZV1trs0MNwaV2n9w7HimXoYuJIOf0CsQ64GevkH
Wj68WsCoSWotCYz0yZMZMKsXXj3kw/45c/fOfPqm0jv9EWhMTOb7OvulPm40zDnjST5uNPlv
vZ2TD2sMb01zMNGED1z9y5tCYcdl2s4CyG2mnwNtxGKwYRI1Fhq3niJ2L33x5iZi3XcpOAYDx
oN6+b8h5zr7qB57i+oHjcA+oQkzPmjybz0QjQmE4pGS9z+h6FLSuW/LzzLrIMOJ6OyAuPwETT
bY0gZPoLEvfaH404BrRZtp2ld/7TvcacQkp3Lb59/RFb+KHbc90DScuUPgRBDlbe+FuvIMRG9R
8nKpaRuZuXYtJxhFCxawK6HHhlu09miiZ04lZoP3iLt1HMjGaoQlICKGisVobd8TCqzxtMIFW1
S2+sxmAk+dhTqX7jxXBb/IFHoDEaw9uKPYaiq84jetqO6GLiaP7jF8xjxoXXsvfW19HwxIMklxZy
7bGn4m9qpO6KM3FedyEpoHoo+1YRoxGDahEz9odrdmCZcQYdNF9+2WiPjaO/Hsfp+mX72n
56w9idtoN+067orV2X2ZLCCF6QxLojVADfqpENYN51FgseaNxV5TRtnQRDd98ijE9k9yb7qHq7
VfJvOhqAh43ilaLZdQ4Ai4XAWc79b98z8KKdH5ZvJii1/+Lz+ulbF0F5517Ma2HH09CZg4pWTn
Bm4XmD22MLTaWWQceyqwDD43sgwsh+oW3oQ5thKtwFy5dTHLWxodvr5c2bDi/fvYhgYCKR
tORCmfuuJdVr79A6innSDExIYYirYa2v+eher3UfvoeWouVpKNOWBWalgbga2nCNmUG1IHj8NR
WoY9LQB+fgDe0UghAW3QM0bP2oOWX74KXtUUTfeJZODxurlBiMOlqK8ZTXUn1Y/d2XLuxox
AZgCktE1NoqlskmTKySDnuVFK00zXi9xJCbD9kCpCgeBrqCLS3E/C4yb7yFtw1VaBRck9bi7e2
mraF8ym+63oS9j2EtQ/fRfmTD9L43Zf4mpvQxyVgTMukdsbOvPD0/az++y98Xi8ATfV1+FLTGB
P3AWTmjRjgpxRCDCRFQ33Eh3CvWbKQtM0cshgVHYPJEkVlcWGXdudvYCFjbWmn9e14kQhRC
RJiiqqDVUjHnMTzrynGuWUXpfbD26V3W2WOpvV3FUVoKp462upeOE9PGJHcc4HRhiY8m
87HoyL76G1JPPwbHkb1StFgBjcgqJhx7d7f62yTmi/5BCCNFPPAf6H1wVZThWLKvj0VrszHm
hbcpuFCmHMsWAXc7x74kH3MK1W+9gur1ghoAIHrGLFJOOb1tdAyVb7xIwr9OxOTxoCgKqqq
Gr/+vCy5n3A5SzVYIESuUqLVFrge6ub6e1sYGEjxhFVn6cOGs3rRAtKHd1TtVjQa4vbcj6pX5xA
9RT4ICzHUKIEA9V981K3dsXJp+HdfazMJex1I2WP34m9vQ2M0kXHhVV3qH2grK6j8+N0u19
DHJ5B91AnB+ygKKSecgbe+jpr330BrtZF9za1Ey+celcQ2RHqgO3GsWcXS4w9ixdnHobVYqHrn
Neo+eS+cPAM0fv8V+sQKFEPwW1t9UgpxJ/2QjAuvQp+UQuPXn2I88HDO0v041tbXccfLbzF62
gzSh+Vx2cNPMeugwwfo6YQqG0nA48Hf2hzRHugV8+eSmTcSTS/WUU0fls+qBd17mmN23Zu
Gbz/HW1/XlyEKlfpBAKXnoqcJSeHftSYzZU/cj7+9LXiO20XZ4/ehDc1vBjBEdZ8/rLtL6LJtysxm+
F2PMOWb+Uz+7FdSTzgDnTXyy/UJIUR/kR7oTuq++AjbjFnkPfAU9Z+8hyEujvrff+p2nGttMcaU
NNDrMaZnoWq0uloL0FpttC5diGPKOGYdeBgjd9qFsTN3ZfKe++H3erHFbroKrhBi++AqL0WfkB
TRJVUW//4TWSNG9eqc9OF5fPfe//B6POg7LXujs0Vj33EX1r38DNIX3NTXoQohlslolPnok2j58
xdUnw8AXVw85vy09wefow1/qJr2eqrXi7e5ObxtSs/CkJaJZ11oaStfle3MC7sMBQfQGAyYs4d
F6GGEeGJgSQldoqqxuRUosZOoODys/FUrcOUMwzblB1wFqzscmzUxGIEz9wNU0Y27csXY8r
Nx1NdScXs68m94yFiDjiMmSefGT5e3+Pai0KI7ZmruBBjuYPre6t9tZWipctZdaBh/XqPJPZQkJ
KKkXLlZyB0tQu++IPlOzi068j49zLpJKtEEIOl0t6Gz+kg64qb8DU2oOj1KAYj/vb28DE6eyzaKBv+tt
aO8/R6dHHxHdhtC2lnnl+/rZWAy4UuJhZjRg4aWVtZCLEdkSHclS1//UHLwnl4a2vwVK0DwFV
ShCkrB8ul0eHjYvc+EENKGqrXR9vf82j65Qd0NhumnGGMf+sLkg87BnOUDFUSQmycs6QQQ1L
3IZV9Zd63X5A9ajRGk7nX52bkjWTFvLnd2o2p6VhHj2fdK7P7IkQhRD/RWKzU/O8ISu+/jcpX51
Ax5zHKHr0b55qODgKNwUjuHQ+Hl6RSDEZyb30ArbnjPUR1OTcmZ2HKGoY5Jw/rqHG0LI6AIS
m1359JCCEGyqBLoBVF2V9RlFWKoqxRfOW6/rqvt7EBfWwCvk7fvAKUPXYfUZnNMOI/LzBq9
hvo7LFUPPUQrfN+pfqtV0k++iTKHr0Hx8rlmNly0UhvsvBiMziLCjAkRyaB9nl9/PjB04zfaectOj97
xGiW//I7j/uSjjqBdXMe61JYSagxuAUA1e8P/u5yBougAqidjgkE0FjM5D/0LMPveZwRj8xBF5+A
onR8VLRNmYGnupKimy9nzXUXsu7IZ0jc7xA0en0/Po0QQgysQZVAK4qiBZ4EDgDGAMcrijIm0
vf1ORw4Vy4j4GwPFvWJjevYGQigi44Go5HiO69H9XrlvuHfJB9/OnkPPUv78sW0zP2FdXMeo2
WhLPEihNg8zul1GFLSlnLtnz95n5iERJlZnm/9539KSM/A6Wijdl15t33GtAxsU3ek7PEHtjZMIUQ
/UX0+Uk8+u0ubotMRNXFKeNuUIELA6cJVUYanuhJXeSmq24XOHhM+Rme1kXry2Uz67Dcmf
zmXUY+/hDI385bJE0KlbcVgm7QyA1ijqmoRgKlo/wMOA5ZH8qZ+p4OmX75Hn5BE619zST/n
UhyrV+AuLyV2z/2xTd+JgMtF6mnnEXC78be3oei01H/5CbVvvRK+jmPiMuL30iCSOqohtGu0il
MKSf3+XVrK9fx9ZuvcsTZF23xNTSKQu7ocSz69Wf2Pvr4bvut/3UCa667mOTjTsWSN3JrwhV
C9ANFhfaCveTeeh8N336Bzh5L9lxZuCs7viQzJiVjzR9Fa3srqtuFxmzBnDMcc1Z012spCqbMLf
tyTgghtgWDqgcaSAfK0m2Xh9q6UBTIHEVR5iuKMr+2tnarb6rR6YjeYRYN335G4mFHs/bhu2hd



OA9Fq6Pxx29R9Aaa5/6Kp7Ge+AMPJ2aXPAl9/39dkmcA87D8rY5FCDG49PX7DUDA7cZbX4s
hMblPrree2+XipbtvYdru+xCbmLTpEzYid+wEFv70XY/7dPZYEG8/IsIbL0MNBbLqPkKIDpF4vwH
QR9mwjZvA2kfuleBy4ioppPL5JzClZXY5LmrMeFJPOIP0sy8h7aSzSE2YsoErCiHE9muw9UBvFI
VVZwOzAaZnm6Zu4vBN0ttjiNltH+q//oyK558k8+JrQaPBmJ6FMTuXxqWLMi8cjX3iNEzpwT8
2GeddQeu83/FUVwiQf+Dh2CbP2NpQhBCDTf+/3wC4ykowJCajaLV9cTkAnO3tvHDXtCtEJ27
x3Of00nOH811DHVVlpaT00NsUt/eBNP/+E5X/fY60U87Z6vsJISLzfgOgj43FmJNHxgVXEnC0
gUaD1mZHFxPX4/Eyp1klITZssCXQFUDnr0MzQm0RZ5+6AyMemY17bQmq14shNQ3z8JHo7
THYJ03rdnzUmPFMePcbnMvr0JjN4WOFEGJTWub91icjVnxeH+WFq1mz+G9++eQDckaNZea
Bh6loylZfW6PRMHLyDP74/BMOP+fCbvsVrZb0cy6l6M7rsO+wM9aRES9XIYTYCtETp6BoNLhK
C9EYjJiy/8/efYfHUV0NHP7NbK/q3ZJsy71jm95b6KHzAQmhQwIBekJP6lQeSiBA6CVAglSEEA
g11AAGG4yNe1PvXdt3Z+b7Y+W11isX2bLWss/7PDygmDMZsyv2as/OveeMwJV+crdEkKIYW
d7S6C/AcYqjKKeOJ8KnD6UF3cPWEK7glTNvt4W/ElbMUjtmFEQogdUfsHb+PeiqmRAV8P7//
1L8x5/208mdkUlo/k0FN/usVFwzZk0m578Lc/PcBhp5+Jw53a99lWPIKin5zLkVp+j+n//AhLTu
6gXl8IMbg8U2fgmT0j3WEllcSwtl0l0lZhxBRF+SXwLmACnjYMY1GawxJCiEGjh0N0zfkfBaeeNe
DHarrOtx9/wJtP/5ny8RM5+eLL8WRmDX6QvbyZ2YyaPJV3//o8x513Ub/HZO5zIOGmBhacfi
Tn/sH9tKR2yweIYQQOho0264SaADDMN4G3k53HEllsS10zfkce/kozB7vJo8NB4P88NX/qFy2
mLbGempWLicK4cfnfzCsuGpgru7gcfzqt/uo+Js3dn/lxZ/R5TcOLpmD1e5h+zH4Wnn0PeMS
fhHDDxUNd4CyGEEeJsD7a7BFoIlXZUhmFQ9/iDZOy290aPC4dCfPrG3/j4n69RUDqS4vJRll+fx
OwDDsWbnTNE0cY5PR40PeWnPH/XLRx/wS+ZdcAh/a6xzvnR0XhmzKbt3TdZcv7/EWltwVY8
Asfl0bgmT8czc3e8s3bD7M3c7GvrkQiGmGyOwbxGQkhhBBCbDIJoIUQYog0vvgUkeYmRvziN
/3ub29qYt7HH/DZm69TVD6K486/mKzcrWtHNRhKRldwzFkX8N7Lz/PZv15n3x+fwKRd98Dp9
iQdZ80vpOiM8+GM89HDISLJYQb6ghVraHzsw8JrFiGNb8Q5/hJOEaNwTaiDhtJGZbcPBLSLIV
hnO/6li+j68IN65s8l2taKoiqYXG7cU3bBu/veuKftgq2gGFQVzddDtL0Nracbw9AxudzYikqwj6zA
vF5sQgghhBCDQTGMQeuSkBazZ8825s6dm+4whBDD3yZLVw90vll2tNPw7GOE6qpp/+87aJ0
ddEydScBmJ+Tz4+vqoLujnZ7ODnRNA8CblcXoSdPw5gztnebnYejQULmKquVLiUbCALi8XrLy
CvBkZeNwubBYbagmEyPHTyK3uCT58ZpGuL6WcG0VkaYGom0tRftb0M0hxDG2EeU4x03EO
W5ivE+2ArgOdglrlxYFypRQ1SqMWCxxvGp3YHK5QVHQgwE0vy+xTzGbMWdmo5hMaAE/Wk9
3v8/L5M3E5Haj2uwoqgqimlyo1qtqHY7qt0R32exoJgt8anpa+/CGwYYBoauY8Si6OEwesCPF
gygh8OgaWY9Z7LgcnhRHU6MTndmJxOVLsDxWrtPa8KKOvu8A9CNfXkX4BB/G++gaHpGfO
MlxY03+mPRTF6/x9UTKbe56iCoWNoGoamxeMymVAt1njMFguq2QJqnzgnMGKx+GsRCWN
Elut+X30eq1qsYDIIP+f1n28i3t7/1rR4zJfllmZ0Pf77Mpvj8VhtSb+jzXote69hrP1dro0/GsGIRjGi
OfgxJhOK2YJq7X3+JnMi/qwDf4R70rSB/DYGFbwrQogNGOQ/JmLYJ9CKorQAVYN4ylygdRDP
l2470vPZkZ4L7FjPZ0d4Lq2GYRy+sQMGOt78yGXJvCHfUbH254aYRoz+x1zDiKdim3vu7YEC6
obyEk1XYjGD8NBGFKcqKNkmxZppUpNmWemGgU8nFjIMXQHsiqJ6TlrMxBjB7X1ftPXmLuBA
PosM+njT3Abjd4TvBLrtjGcYoXhFW8usHRT440YmGGfQA82RVHmGoaR2vh5mNqRns+O9Fx
gx3o+O9JzGS076mu2lZ6vHfE5wY75vHbE5zRQw+01GE7xSqzbxnCKFYZXvMmp1uFETXca/y
kGGgAAjFJREFUQgghhBBCCCHecCAJtBBCCCGEEElIsRkkgU71eLoDGGQ70vPZkZ4L7FjPZ0d
6LkNIR33NdsTntSM+J9gxn9eO+JwGari9BsMpXol12xhOscLwinc4xTpsyBpolYQQQgghhBBiM
8gdaCGEEElIYQQYjNIAi2EEElIYQQQmyGYd//8vDDDzfeeeddlchhBj+NtDReB0Zb4QqG0TGG
yHEUNnkeCMGZtjfgW5tHS59zIUQw52MN0KloSLjJRBCbJ+GfQlthBBCCCGEEElIMBUmghRBCC
CGEEElKzSAJtBBCCCGEEElIsRkkgRZCCCGEEElIITbDsK/CLcRwFqqrIVS9BrPbg71iPGanM90h
CSGE6BWuryVYtRqTy42jYhxmlzvdIQkhhEgzSaCFSJOehd+x5NyTiba2AFB0zkWUXnwlqzsNEc
mhBDC98N8Fp97CtGWJgAKz7iAssuuxpKdm+blhBBCpJNM4RYiDWK+Htbcldl0ieQZoePoR/lu
+T2NUQgghADS/j8q7bkwkzwCNLzy074f56QtKCCHEdKESaCHSINbZQc+3c1K2hxtq0xCNEEK
lvqJdHfTM/Sple7huxxijjVgs3SEllcSwJQm0EGlgyC4mY4/9UrbbSsrSEl0QQoi+LFk5ePfaP2W7v
bQ8DdEMrmh7G98etnu6wxBCiGFLEmgh0sDkdDPy2luxje9MKaqjLjkalyTp6c3MCGEEJgcTsq
vuAFb2cj4BIWI5BeX4506l51hDYpYV2fS1HQhhBADiOXEHegT96SpTPvbe4SqKzG5PthGj0G12
tldlhBCCMA9cQrTXnuPUNUaVJcbx+gxmGz2dle11fRQAD0UTHcYQggbxekCLUQaWfMLseYX
pjsMIYQQ/bDmFWDNK0h3GINKCwYxYjH0aBTvYkl3OEIlMezIFG4hhBBCiJ2EHgwk/VsllcTAS
AlthBBCLGTWdt9Ww+H0hyJEEIMT5JACyGEEElSjDS5Ay2EEFTfEmghhBBCiJ2EHogznlpQC
okJlcSWkARaCCGEEGInkZjCLZW4hRBii0gCLYQQQgixk0hM4ZYEWgghtogk0EIIYQQQOwm9d+
q2LIO4hrBii0gCLYQQQgixk9AC/vi/pYiYEEJsEUmghRBCCCF2EjKFWwgtho4k0EIIYQQQOwk96
AdVIQRaCCG2kCTQqgghBA7CT0YxOzJkDXQqgixhSSBFkIIYTYSWH+PyaPV9ZACyHEfKpLA
q0oSqmiKB8pirJYUZRfiqJc1rv9JkVR6hRFmd/7z5HpiE8IIYQQYkekhWky3B65Ay2EEFvlnKbrx
oDfGlbxraloHmCeoiyv9+673zCMe9MUlxBCCCHEDksLBjF7vGghuQMthBBbli0JtGEYDUBD73/3
KlqyBChJRyxCCCGEEDsLPRTEmleAHpAEWgghtkTa10ArijIS2AWY07vpl4qilFAU5WIFubLSF5
kQQgghx15F770DrcsaaCGE2CJpTaAVRXEDfwd+ZRhGN/AoUAHMIH6H+g8beNwFliQLMVRRIb



ktLy1CFK4TYCcl4I4QYKkMx3ujhYLyImLSxExKILZK2BFpRFAvx5PIFwzBeBzAMo8kwDM0wDB
14Atitv8cahvG4YRizDcOYnZeXN3RBCyF20jLeCCGGylCMN3oohEnuQAshxBZLVxVuBXgKWGI
Yxn19thf1Oex44Iehjk0IIYQQYkelBwMyhVsIlbZCuqpw7w2cASxUFGV+77brgNMURZkBGElcG
E6ghNCCCGE2NH0SiGrmNytCkzJZUQQmyRdFXh/hxQ+tn19IDHlOQQQgixM9BDQVSBhCvMq
w+H0h20EEIMS2mwwi2EEEEllba9tQm0arWhyx1oiYTYIpJACyGEEELsBPRgANVmQ7XZ0MOS
QAshxJaQBFoIIYQQYiegBYOodgeK1YYekincQgixJSSBFkIIITYTYCcSncNviU7hIdbQQQmwRSaC
FEEIIIXYCeJcAYrWhWCwY0SiGpqU7JCGEGHYkgRZCCCGE2AlogfgaaEVR4uugQ7IOWgghBko
SaCGEEELInYAeCqBYrACoNrv0ghZCiC0gCbQQQgghxE5Aj0RRLRYAFJsdPRRlC0RCCDH8SAIt
hBBCCLEz0GloJhMAqtmCEYmmOSAhhBh+JIEWQgghhNgJ6LEoqPEEWjGbmGKSQAshxEBJ
Ai2EEELsRMwotHEHWhM5nhCLYQQYkAkgrZCCCGE2AkYmpZloBWTCSMWS3NEQggx/EgCL
YQQQgixEzCifaZwSwlthBBbRBJoIYQQQoidgBGLopjiH/0Us1nWQAshxBaQBFoIIYQQYidgxGLJ
U7ijkkALlRASQlthBBCCLETMGKxxBRuVBOGJIO4hRBioCSBFkIIITYTYCeh9qnArJjNGVBJoIYQ
YKEmghRBCCCF2AkYsgmlyA2uLiMkUbiEGEGChJoIUQQgghdgJGNLauJgk0EILsUUkgRZCCCG
E2AnEq3DH70BjMmHEtPQGJlQw5Ak0EIIYQQOwEjGgWZwi2EEFtFEmghhBBCiJ2AEYuhmN
cVEdMlgRZCiAGTBFoIIYQQYidgxKloqvSBFkKlrSEJtBBCCCHETkCPrWtjhapiaLIGWgghBkoSaC
GEEELInUB8CvfaNdBmuQMthBBbC0JtKlopYqifKQoymJFURYpinJZ7/ZsRVHeVxRlRe+/s9IRn
xBCCCHEjsalRiExhVuVImJCCLEF0nUHOgb8xjCMScAewMWKokwCrgE+NAXjLPBh789CCCGE
EGlRGX2mcMercMfSHJEQQgw/aUmgDcNoMAzj297/7gGWACXAscBzvYc9BxyXjviEEIIIXY0Rj
SamMKNKgmoEEJsibSvgyVYUZSSwCzAHKDAMo6F3VyNQkK64hBBCCCF2JIYWS6rCrcsaaCG
EGLC0JtCKoriBvwo/Mgyju+8+wzAMwNjA4y5QFGWuoihzW1pahiBSlctOSsYblcRQ2dbjzfp9oI
1YZNCvIYQQO7q0JdCKolilJ88vGobxeu/mJkVRinr3FwHN/T3WMLzHDcOYbRjG7Ly8vKEJWAix
U5LxRggxVLb1eKNHoyimtVW4pQ+0EEJsIXRV4VaAp4AlhmHc12fXv4Aze//7TOCNoY5NCCG
EEGKHpGmwtoiy2SxrolUQYguY03TdvYEzglWKosv3XYdcCfwqqlo5wJVwCnpCU8IIYQQYsfSt
wo3qgkjFEpvQEIIIMQylJYE2DONzQNnA7oOHMHYhhBBCiJ2BEY2ta2NlliJiQgixJdJehVsIIYQQQ
mx7hhbrswbajBGTBFoIIQZKEmghhBBCiJ2AEYuuWwNtkj7QQgixJSSBFkIIITYTYCeixPIO4TSa5
Ay2EEFTAEmghhBBCiJ2BpkkbKyGE2EqSQAshhBBC7ASSqnDLGmghhNgikkALIYQQQuwEjJQp
3LIGWgghBkoSaCGEEELInUA8ge4zhVsSaCGEGDBJoIUQQgghdgKGFutThVumcAshxJaQBFoI
IYQQYgdnGEbKFG5diogJlcSASQlthBBCCCLGj0zRQVRQ1/tFPMZtlCrcQqmwBSaCFEEIIIXZwhrZ
u/TMAqik+pVsIlcSASAlthBBCCCLGD06N9WlGbilMkiAkhhJaQBFoIIYQQYgdnxKlo5nV3oBWTG
UPWQAshxIBJAi2EEELsYMzYtp6CbRM4RZCiC0hCbQQQgghxA70iEUTLawAkD7QQgixRSSBF
kIIITYYwRmxAFIRsXgfaEmghRBioCSBFkIIITYYwRnRWOOubkmghRBiwCSBFkIIITYYwRlaDEXt
U4XbZi5P6xZCCDEgkALIYQQQuzgjOj6VbhVDE1LY0RCCDE8mTd9iBBiawVWLSe4chmq04Vr
wmSseQXpDkklIQCINDfhX7YIPeDHMWYczorx6Q5JbAPxNdDr3YGWKTxCCDFgg5JAK4oyHdi3
98fPDMP4fjDOK8SOoOe7b1j0s+PQ/D4APLvuyfj7nsBWUprmylQQO7tQfS0rfnMh3XM+B0B1u
pj8/D/xztwtzZGJwWZoWlICjckEmoZhGciKkr7AhBBimNnqKdyKolwGvAjk9/7zF0VRLtna8wqxl
9ACfqr+cGsieQbo+eZLeubPTWNUQggR5/t+biJ5BtAdfqrqvUt4Nvlo8RwZETXq8KtKGAY0RI
HbQQQgzEYNyBPhfY3TAMP4CiKHcBXwIPDcK5hRjWYn4fgeWLU7aH62vSEI0QqIQLN9SibPM
vW4Tm82FyutMQkdHw9PX7QNNbiVuLadb0BCWEEMPQYBQRU4C+VSi03m1C7PSs2bnkHnV
iynbXpGlpiEYIIZK5JkxJ2ZZ7zEIYsnPTEI3YpmlJbawAFLPcgRZCiEajAT6GWC0oig3KYpyE/A
V8NQgnFeIYU8xmSg680Kyf3QMEF9f0OqGu3BPn5nmylQQAtzTzJL6pnsweJ3m7MPPYris36
ekmiJ4U+PxVDU51998VZWUkhMCCEGYqv/QhgGcZ+iKJ8Ae/duOtswj0829hhFUZ4GjgaaDcO
Y0rvtJuB8oKX3sOsMw3h7a+MTIt0cl0cz7v7HCdfVoFht2EvLpWCLEK7YHZ7KDzjLIOPAw9H
MJWUorJ4Ux3WGlBMGJRMPVzB1p6QQshxIAM1Ifm84GGtedTFKXMMIzqjRz/LPAw8Px62+8
3DOPeQYpJiO2GyeHEOUZawwghtj+KomAvLU93GGlBM2Kx5Crc9K6BljvQQgghxIFudQPdW3L4
RaGLd+mcD20AiT8MwPIUZeTWXlSIIYQQQmza+n2gQe5ACyHElhiMO9CXAeMnw2gbhHP9
UIGUnwFzgd8YhtExC0cUQgghhNipbegOtC53oIUQYkAGo4hYDdA1C0d5FKgAZhCfDv6HDR2o
KMoFiQLMVRRIbktLy4YOE0KlrSbjrBiqGzL8Wb9PtDQW0RMqnALlcSADMYd6NXAx4qivAWE1
240DOO+gZzEmlymtf+tKMoTwL83cuzjwOMAs2fPNgYasBBCbC4Zb4QQQ2VbjjeGFoP1qnBjN
vf2gRZCCLG5Bi0Bru79x9r7zxZRFKXIMlyG3h+PB34YhNiEEIIIXZ6RrSfKdyqSe5ACyHEAA1GG
6ubN7ZfUZSHDMO4ZL1tLwMHAlmKotQSL0J2gKlOM4gXIKsELtza2IQQQgghxlaKiEkVbiGEG
KjBam01MXuvv8EwjNP6Oe6pIYhFCCGEEGKnE5/CvX4RManCLYQQAzUYRcSEEEIIlC2LF5ET
PpACyHE1pIEWgghhBBiB6f308YKSaCFEGLAhiKBVobgGkIIYQQYgP6XQNTmskUbiGEGKChSK
AfHIJrCCGEEEKIDTCi0f7XQEsVbiGEGJctLiKmKMo44EqgvO/5DMM4qPffz27tNYQQQgghxJb
b4B1o6QMthBADMhhVuF8DHgOeALRBOJ8QW8XQNKJtLahOF2a3J93hCCF2MrGebvRgAEtu
PooqpUbE9sGlxVDMYr/7FJMJPSoJtBBCCDMRgJNAXwzAeHYTZCLHVgtVraHj2z7S88Qq00W
Mpv+omvLP3RFFkKb4QYtsydJ2uOZ9TddeNhGqrKDj5DAp/cg72EeXpDk0I9GgU1WJJ2qaYTBj
RSJoiEkKI4Wkwvhp/U1GUixRFKVIUJXvtP4NwXiEGRaUHQXnwDhgefZRYRzs98+aw6lZjCCxfk



u7QhBA7Af+ShSw+6wR8C74l1t5G3Z8foO6Jh9BljanYDhiRCMp6CTTSB1oIIQZsMBLoM4mvgf
4CmNvnHyGGVKShlpY3XkvaZkTCBFYuTVNEQoidSWD5kpSCTE1/fZZIU0OalHJiHSMaQTGTn4
Xb3H8RsXB9LcHVK4cqNCGEGFYGI4G+GphuGMYo4Bnge+CkQTivEAOi2uyYPRkp201OdxqiE
ULsbEyu1LHGkpWDarOnlRohkunRaD9roM3okdQp3M1/f4m6px8eqtCEEGJYGYWe+neGYXQrir
IPcBDwJCBrosWQsxWVMPK3tyVtc0+fhWviiDRFJITYmbgmT8c5fLLStpG/uwNrXn6alHJiHSMa
SU2gzeZ+10BHWpuJtrYMWWhCCDGsDEYRsbWVt48CnjAM4y1FUW7b2AOE2FZyjjwee+ll/lsX
Yi0oxD19FrbC4nSHJYTYCdhlSpn4+F/p+X4u0bZWxJOm4Z46l91hCQFsaAq3qd81+tHWZqJtr
UMVmhBCDCuDKUDXKYryZ+BQ4C5FUWwMzp1tlQbM7HSRsfs+ZOy+T7pDEULshOyl5dhLpeq
22P5scAp3OJxybLS1hVhH21CFJoQQw8pgJLqnAO8ChxmG0QlkEy8qJoQQQgggthtMbLCIW6S
eBbm8l2tE+VKEJlcSwstV3oA3DCACv9/m5AZCSo0IIYQQ2wmjvzvQZku/a6Cj7W3EujxdB1FI
UmFQgjRl4yKQgghhBA7uH6ncJstKVV4DV0n1t2FarUR6+4cwgIFEGJ4kARaCCGEEGHHZ0Sj/U
7hXj+BjnV2YHK5sWRIE22XddBCCLE+SaCFEEIIXZwm9vGKtrWgkjA5M3k5gk0EilKWIwqnALI
YQQQojtmBGLpSTQqtmMvl4RsXgCnYnqcBFtl1ZWQgixPkmgt00xgJ/gymVo3V3YykbhKBuZ7
pAGRPP7CKxchubrwV4+GvulskE9f8zXTXDvViv5R1ZgLykd1POnW6y7i8DK5eihAl5RFdiKRqQ7
JLETiba3EVy1DEPXcYwehzUvP90hbZZleyuhVcsxdAPH6LHbRdx6NEpw9QoizY3YCotxjB6LYj
KlO6wBC1aullXhTkrG8eY8ZjsjnSHJLaAvoEp3EaknzvQngxUh1MSaCGE6lck0NuZaFcntX+6l/
onHwLAnJHFxKdexTtztRHtnmi7W1UP3gHjS88AYAlJ5eJT72GZ9rMQTI/pK2F6vtuo+nlZ+Pn
z81j0lOv4Z66y6CcP90izY1U3nk9L98FQBRYTETn3wV96SpaY5M7AyC1ZWsvOoiur/+HwDOiV
MY/9CzOEePTXNkGxesXsPKKy+i+5svAHBNmsq4Pz6T1rj1aJSWN15l1XWxu/8Wa2M+8Pj5
Bx5HlqipC2uger66nOWXHAqmq8HFIXSS6+h+NyLMbs96Q5NDJAR20ARsZQp3G2YPF5Uu00
mcAshRD9kDfR2xr9oQSJ5Boh1dbD6xt8Q6+pMX1AD4Fs0P5E8A0TbWqm8/XfEfD2Dcn7/wv
mJ5Bkg2tpC5d03o/I9g3L+dOv5fl4ieQalNNZT+6d70MKhNEYldhYdH7+fSJ4BAkt+oOWN19IY
0ebp+Oi9RPIM4F+8kNY3/57GiCC4ZkUieQYwlhFWXH0xocrVaY1rlCJtLay45uJ48xgGNQ8eA
eBpYvSG5jYlpvbxirS1ozJ48Xs9hBpaxnKEIUQYliQBHo7E2mqT9nmX7SAaFdHGqlZuEhjavw93
32NNkhfAlTra1PP/+2cYfMFw6YE16xM2db9zRfEurvSEI3Y2fRNQtfq/OxD9N4kcHvVN+lfq/Oz
DzEOLQ3RxEVbmhPJ81p6wE+krTINEQ1crKODchVlyvZwP+O82P4ZsQ1N4Y4mbYu2NGP2Zm
DyZhBtlQRaCCHWJwn0dsbWz3pez6w9sGTlpCGagbMVP653zthrf8xZ2YnYfns/68Ez9z4Q8zB
5fTbFOXZCyras/Q/FkpGVhmjEziZznwNTtuX86GhU8/a92idzn4NStmX/6Ki0rje2Fhaj2uxJ20ye
DKz5RWmKaOAsOXk4xk1M2T7YdS3E0OiviJhiNqdO4W5vxexzYvZ4ZQ20EEL0QxLo7YXr0jTKr
74l8UfONqKc0TfdjdnjTXNkm8c9dQalv7oOej+42keOpvyqzmzE5XYNyftfUXRhyvVWgxv/XtY8eS
9nlv8Xk2DGK2nimz6L4vEsSz885YTIIF1yGarWmOTKxM8jc9yByjz4h8XPgVgeRc+TxaYxo82Tu
fzA5Rx3f5+dDyDn82DRGBI5RYxh7/xOYXG4ATN5Mxv/x6WVFVFNKSlc3YOx/GWhBP+hWrlE3
34tz/KQ0Rya2hN7vFO7UlmKarwfV4cTk8RLrbB/KEIUQYliQDMNlZ4UV5WngaKDZMLwpvdug
VeAkUAlclphGBuduzx79mxj7ty52zbYlaZH04QqVxHr6cZeWo41ryDdlQ2IH0SXLmCzefDXjYS
a+7gVsPVI5H4+QMBHGUjseTkDer5000LhwitWYUeCmlrG4U1e8e4uz4MbLKyo4443qxP8/sIV
q6KV+Eur8DsHR5f3ml+H8E1qzCM7SvuYNNVqoq0tWPMLSJeOTHc4WyTc2EC4vhpzRiaOkWO
GZSx7dCQjjeGYfBFRSaTX3wzqYhdqKaSusceZOaH666z4MRDyD36RExud8o+lcSwNhwqVw
4T6ZyX9yzwMPB8n23XAB8ahnGnoijX9P58dRpiS5twUwP+xQvQurtwjBmPJTss33SEBoEfC+Bf
/QHDNSiw5ubgmTd1gYqxaLLjGbb57FKrVimv85G12/nQz2ey4Jmzfzy+wYin+pYviv4tJ07CXlq
c7JNGPcENdfDzx+XCMnYBrwmQUdeMTj0wuN+7J04cowv6FeuPW18Y9ccomK1ebXG7cU9lb
d38c5aNXlI9OdxbxVZYhK1w+5p6Hqqpwr9kIXo4hGv8ZJz9TDUX66wtlLb++6i/KtxaMlhqs6N
a7Wih4FCGKYQQw0LaEmjDMD5VFGXkepuPBQ7o/e/ngl/ZiRLocEMdy359Hj1f9xbyMZmY9N
RrZO13cHoDA9re/TfLLZsn8XPOEcdScet9202CL4ZOz/fzWPTTHycqn9tGIDPpmb/hrBiX5shEX
6G6apZdfCa+Bd8CoFgsThR2dTL33C/NkW1cqK6apb/4Gf4fvgPicU9+7h9k7LFvmiMT24vg6pU
sOvckwlVrAFcdLqb85V94ZsxOc2TbLyMaSZm+Db1VuGPJRcT0YADFZkOx2dAlGRZCiBTb2xr
oAsMwGnr/uxEYXnOXt5Lvh+/XJc8Amsbqm68i2pHeNUjh+lpW3/ibpG1t/3kDv7Qy2enosRj1zz
yS1DysXFtF52f/TWNUoj++Bd8mkmeI34GqvON3RLfziU6++fMSyTP0xn3nDcR6utMYIdiedH75
SSJ5hnh189o/P4i+3lpesY4ejaCYLSnbFbMZI7peAh0KotpsqFZJoIUQoj/bWwKdYMQXZ/e7QFt
RIAsURZmrKMrclydp8VCrJ9WVeHaKvSAPw3RrKMF/MQ6U2Prb5vYsRmREIHIS1K2BytXpSG
aoTFcx5toe1vKtlBVZdrHk03pr+pvqGoNmN/7jlsMnVD1mpRtwZVL0cOhNEQzuLbVeGNEYxtM
oFOncAdQrTZUqxU9HCJdtXKEEGJ7tb0lO2KohQB9P6734aZhmE8bhjGbMMwZufI7TgFpJwV
42C99U15x56CJb8wTRHFWQuLyVhv2qdiseAYPTZNEYl0MTndFJx8Rsr2rP0OSUM0Q204jff9V
UrOP+G07b4oobOf9f95J5yGNW9wixGK4Stzr/1TtuWf/NNh061iY7bVeLPhKdxmjPUSaD0UQrX
b42umVTWISrcQQuzstrcE+I/Amb3/fSbwRhpjGXKuydMZ/6fnseYXgqqSd+wpjLJoN6iW1G+Nh
5LZ7WH0TfeQedDhANjKRjLxiVekaMt0KueIYym58DIUqw2TJ4PRN9+Ld/Ye6Q5LrMc9ZQbjHn
waS24emEzkn/xTis7++XZfQdk9ZRfGPfjUurhPOYOiMy/c7uMWQ8czc3dG33Y/Jm8mitVK8Xm
XkHfMSekOa7umb2wNdJ8p3lamxZNtS7x1omqzo4cCQxanEEIMB2krlqYoysvEC4blKopSC9wl



3Am8qijKuUAVcEq64ksH1WoI9/Af4521O1owgLWgGJPNlu6wAHCOncCEh54h0tyE2e3e4VpH
ic1nKyyM/IobKTz9XBSzGVtRSbpDEV0w2R3kHXMiGbvTjR4OYS0sHhb9xE0OB3nHnETGbvS
Mq7Jf0DF7vBSdfg7ZBx6GEYthKyrpNzkU6xixDUzhNpkwYjEMw0BRIN71z/ZEtW7FbkCLBjFnZA11
yEllsd1KZxXu0zawK/0lp9NsKKZYxgJ+fPPn0vnph1gKCSna50CcY/u/oxyqrqTzq08JLFTcXu57
4501+Hcb9UiEngXz6PzkA0wuD5n7HYR70RbV44YHlrJJk2rhglrQfqWgATXrKTzi08lrVIFxt774
521B2ZvXmY9drDjDlatpvrL+j59muc4ybh3WMf3B0nDOo1xNCSL+82nxHZwB1oVU0UElOsVr
RAANVuT+xBbXb0oBQSE0KlvuQr251U58fvs+yXZyZ+rsvJZcpf/5PSiijc3Miyy87B9/08ABqeeY
QRl1xF2SVXD+o3/t1ff8GiM4+D3mltX+6hymv/EeSaCGGqVBtNYvOPZlw5WoA6p/+E6NuvJvi
My8c8lhiAT/1TzxE40tPJ7a5Z8xm3ANP4SgbOeTxCDHUjFh0g3+z4904I2C1ogf9qHZHYI+8Er
dM4RZCiL62tzXQYghE09qpuufm5G1trUktb9YKLI+SSJ7XqnvskLVIYMWjxYOUfvYfYnkGUDz+
+j89MNBu4YQYmj5lyxMJM9rVd/3e0L1tUMeS2DpDzS+8lzSnt/8uQSkFZ/YSejRjSTQFkuiErcW
jE/hXku12dDkDrQQQiSRBHonZMSiSX1819JDqS1A+uuraUSj6LFoyvYtpmnE+uING+vpGbxrCC
GGVH9jhx4KgBYb+lIiUdC0fraHhzwWldLBiEZQTBU5A937ftWDAdQ+tVdUmw09KHeghRCiL0m
gd0LWvAJKLrg0aZtiseCeMiPIWMfY8VhycpO25R53KcufzU5XZSce3HyRkUh64AdtzWSEdS6
1/hJmFzupG2FPz0PW9GIY/FOWY8Geu1PriWFG2w7oMQOxpjY3eg+/SC1oIBID53oBWRDT0k
d6CFEKlvWQ09k8o77v8wOVw0/OUJrEUjKP3Fr3FNTI1v7CgdyATn/kHDS4/S8+035B5zEnnHn
4rJ4RzUeDIP+BFj73uc+icfxuTNYMQvLsczffagXkMIMXScY8Yz+S//ou6JhwgsX0z+ST8h95gT
01lt2ZqTR/k1t9Ly+st0fvoBrqkzKPrJebikFZ/YSWxqCrcRic8q0wMBVGuf09BWG5rcgRZCiCSS
QA8iPRbDiEUx9SnAMVgMXSfc043JbEbTNMwWK2bHll/HmptP4U/Olfe4k1Et1qQ/mOtzt5pG
xe//iB4KYvZ4B3wtLRRCMZtRN/LB2ZKRSf5x/0fOj44C1bRNXkMhdjRrK+Yq6pZPJgoFAlhsNky
9fZYj4TCKomDpbR1lxGLosWhvMaEQJqdzs6/tmT6Lcfc/gR4KYfZ4tjjGzaVHIhgYmPoZzzxTpu
OaNVJo26+xeDOTpqkKsaOLT+Huv5e6uralGPFIFkITuK02qclthBDrkQR6EBiGQc+3c6h/6hHCd
dUU/vQ8sg46DOsg9Upunz+P4A/zIRL/YNv6nzdQLBbyzjifvAM03apk0+zavA+1qsWCakntlbkx
0bZW2j96l8a/PlmteATF51yMZ9buif6S/TE53RvcJ4SIC1atoeVfr9H+zr/w7r4PhaeiiXOAd1Ob6
2r57M3XefelZxkzdQbHX3gJLXW1vPrwHzCbLZx++TVUeDw0vfQUnhm74vvhe/wLvyPniB9jHz
WWukf+gGfXPSk87Wxc4ydt8DpbMnYMIbYO0T3nc+qeeAgjGqH4vEvI2HN/zC5Xciyqim0I2gQ
Ksb0xYtGNrIHuM4V7vTvQiiTQqgiRQhLoQeBf9D0//OSYRBGOlVdfzKgb7qL4rJ9v9bkDNVXU3
X8btvxCnGMmUHnXDYI93V9+ivrkq+QfdNhWX2dbaPn331Iz81UA+BZ8S/t/32Haa+/hnrpLmiM
TYviK+XpYfcvVdH70LhCvdt3+4dtMfeU/2Ao3ry9uLBrItYf/wN8ffRCA1YsWUj5+En++4arEMYfu
dwDhpx+m8PRzqH34HiLNIjYnrZR1wKOasbBqff5z2D95m6qvvYC8uHeRnuvl65s1h8VknJn7u/
voLJj75Ktnb6dgoxFDTI5tqY9U7hTsYREmawm1FkzZWQgiRRlqIDQLfD/MTyfNadX9+gEhr81af
u3vJQrxTdyFYuZquLz5J2d/2xqtbYf1tldrWQt3jDyZtMyIRevppLSWE2Hyh6jWJ5HmtcE0VgZXL
N/scTbXVvPHkl4mfJ87enW8+TD5nHgZ6MIDJ5Uokz2t1fPw+3lm7AxCpryU4gGtvCy39jIMNLz
yOoetpiEal7c9Gq3Bbzlmq+VrQj9q7fANAsVrIDrQQQqxHEuhBoFqsqdsdhs1//VGAzq32Rz/Ztg
w+l2zpzd/Txq02Ay9x/vRtZaCyE2TTGZoJ9IEKp586dJm1QT5j4fkqPhMLb1aioYa6/R35ILkw
mjT9/2dBQG62v9at+JbRtZLiLEzsSIRWEDA6AVkxkjEm/pFr8DvW5siBcR8w9JjEiIMVxIAj0I3NN
nYc7MStpW9pVfYcnO2epze6fMoGfVcjZT5Kxxz5JfWAVi4WcH5+01dfYFiyZWZT95vqkbeaML
DwzZqUplIF2DPAFRfScfk7SNs+sPXCMBh/Z5ygoK+fMq29M/Lx8/jx2Pfgw1D4FwWpCYSz5h
UQaG3Cut8Y5/7j/o+Oj94D4+JfudlC5x5yE0nedtapSdMb5G623IMTOxIhuag10fAq35vel9oEOy
BRuIYToS9ZADwLnmPFMeektOj5+j3BDHdmHHlln5m6Dcm57fiFIV96Ef+IC1EiEMxf+Cd/Cb8F
qw3vAoeTsvs+gXGdbyDrgMCY9/0/aP3gba0Ex2QcemvYP2kiMdyabndJLriJjt73p+vIT3NNmkr
n3AQMqWqgoCkeccQ4jxozj6/f/Q/n4icw68FAe+M+nfPH2G5jMFiYcdjSIJ51G56cf4Jm107GuT
olrl5Gx576YvVkoFgsFp51N5j4HYM3L34bPeNM802cx9ZV3aP/gbXQtRs4hR+GZLI/WCbGWH
olsYg30uj7Q6np9oGNtrUMSoxBCDBeSQA8S14TJuCZMHpRzhZsbiba1gqJg5BaA1YJn0jT8Bv
giYRyz9sBsNuMydKldBzV1wVklBYm2tmBye7Csd7d8k48NBgg31GHEYigmE5bsXExuD5HmR
kwOB5bs3H4fZ3a5yNrnQLL2OXCT14g0N6LHYtgKizFiUSltzZicTixZW34XP9reihYMYs0v3OZV
glUYSrb8QvK0OZG8Y07c9MEb4M3KZo+DD2PG1OIEA366QkG8GZkcccY56NEoFhRMmVnkH
X8aRjSKJScX1WJB8/cQ7eigbJeb0X0+T0stlwk31gPGZhc060+kpQk9Gkv1ONADAUwuJ5rPjy
UnF5PDiRYOE21txuR0YcnKBkXBWIBE3gmY/J6sebk9Xv3WQv4iba3YfJ4sWrkbnF8WyoIbiG
GSLwK9wamcJvN69pYBfyY3eu6c6g2G7r0gRZCiCSSQG9Hom2ttH/yPsHISj2863OgwJg7H6b
1s49Yk5PPM48/TFd7GwedeBq7HXIYuV0dhJ56IFHX30n2gYclFf7oK7ByGVV/uJX299/CMWY
8FTfdg3f3fTZreqN/+Wlq77yBzk8/xDI2AsXnXlwlV4DWt/9J6z/+irWgmFE33k3Wfgdv8Pobo/l6
aH37n1TdcxNalEDRWT/HkptP5e2/xV5azuib7iVz7wM2+le/P3okQsenH7Lm5qulNNWTD8LpLF
50OfayUQOOT4gdVXDNSjq/+pzOclj3v57DW3//K96sbl674JfM/e977HHoEexZWkrz/bfH30fHn
0bBaWdTDfeNdM/5HPe0meQdewrN//grl6++GeekabS8/jl1D94JGly4+EoKTjwdS07/X7D1Jxb
w0/70v6i883q0nh7yjj8VVJVORxvOinH4fvie0kuvpuHxz2h963Xs5aMZc/uDdM/9irrH7kcxmSk
47SycFePJ3P9grLnr7oz7ly2m8q4b6fzKfVyTjL6xnvwzt5jG7yy/QusXkHN/bfT+s4b0EZWMPq



me8jYc7+t6uEtXOaKT+HecAKdmMld8KPa192BjveBlgRaCCH6kr/c25GOLz4hUltN3eMPonV3k
rnnftQ9eh9thcXcc/N1tdbUEw2HefelZ/nu049odLhQnG6WXXQGvh/m93vOmK+HVddftVu7b4
KuE1y+hEVnnUhw5dJNxhPt7GDFFT+n8+P3QdcJLFvMmluuJrh6JXpPN0YsRriumqUXnoZ/8YI
tes7d337Nymt+SbStFT0Yo07R+4g21mP2ZhCqXM3ic0/Gv/SHAZ3Tv3ghSy88jXBdNUYsRvOr
z1P7+B8THxCE2NlpwQD1LzxJYNliPluwgH++9BzRcJi2xgaeuuW37Hbw4RQqUHPIL9a9j157gf
qnH0bz9YBh4Pt+HrWP3EvGbnux+JyT8H8/j8rbrkXr6ULr6abqzuvp+PyjAcXlmz+XFVf8nGhrC3
o4RNNfn0W12fDNm0Ng+RjiHW2suu6yeG0wTSPSWE/n5x9Rfe8taL4eYl0d1D12P+GGWjo/+2
/ivNGOdpZffj6dH78HhoF/0QlWn3MSwcpVg/zK9k8L+Flz23W0vvU6aBrBVctZfM5JBjYvGZLrC
6FHNzKF22ROdbLRg4HkNIY2G5ok0EIlkUQS601I9zdf0DP/m8TPjjHjCaxYsmVjY8qxX/znX3S
2t+E+5iQWjA1+EAxV19I95/OkbUYkTHD1yk3GE66rwf/D90nbNL+PWEcbjF9ChYzBoGVyzZ5vv
70/ZC7Vscn7+Pdda/eC8Y/bA5EYOVS6FMhGKD5by8SbWnaohiF2NFEghswWSz4wmHef/uNI
P0dLU1kGHRk+6j93TfJ7LMkl9ragupwYkQi+JctTjIP89/+MqC4utYbqwa6PnoX7+570/7B22Tuf
QCBpT9gKx0JxNc+d331WcpjfAu+pevLTzE0DYBwXTWBjclfxGm+HoJrhiaBDjfwX5P3Poxod
MBjmxBbytoGmgzem8Vbi2QuqZaD0kbKyGE6EsS602IJTsXa35R4mcjEkGxWPC6U1u05BWP
wGq1EaurAsCcndvOU0OJyaPN3W7N20T8ZicLIS7I2W7arOn9L02b+F6QltpWco2a1EJ0T49t
E3egZ27v1hsxSUojtNlsTOSHU40AJ+zHqMgulRKfsdbg9RUdp4WAuT35uwrn3W+p0IAJzjBIY
00FaUum7aVlhCtLUFa0ER0bbW+lf73v70kdbmftdaWwuKMOfmJaasmlzupKRgLBm3dWzcFk
wOR7+vjlj0+OwEINBj0Y2PIXbZlq3ywT0UCC1Crf0gRZCiCSSQG9Hcg49EnvFWMY9xWVa//06
RWdeSEKsSmnF2MRxZouFw047k5KSEXT/81VcU6bjnrpLv+e0I5Yz6nd3JG3L/tHRuNZrS9PvY
8tHUX71zcmPPfgl7CPKaX3rH4lt7umzcE2evtnPs6/MvQ/E2udDs2p3kLXfwfR8+zUA3r32xzVp
6oD06Zo8HfeM2es2qCqjrr8T61YUJBNiR2IrLCZjr/3JHDeJ0087A30fInsjKsYSCQZ3tqKY9rM
dQ9SVUovvZrWt/+Z2JR3/Kl0fP5fsg46HPfk6UIJosmbSf4Jpw0orozd98FWWp74WbHayD70K
LrmfE7Rzy6k9a3XKb3sGrfjo8/wRVLydjnoKQvCc3ZOdhy8k74vjENnv56JSxLP+knw5ZVwBb
0QhG33h30rbM/Q/FNWHKkFxfCCMaRdlQr3iTGT1RRcyYnEDLHWghhEihG0tN0RtuZs+ebcyd
OzfdYQyawKrlhGqqiNTXgKriGDuRaGsTXYZCVVsb/nCI7IliMnNyyGxqwGa345k6A1tx6QbPqQU
D+BcvILhmFzbcPNyTp2PNK9iseGJ+H77v5xFYuQyz24M50w/FZsVktHCqqcTKycA1ZTr2fu4c
ba5g1Wr8ixegh8M4x00CXSOwbDHmzCcxk2dgKyza9EnWE2qoxf/DaRSeLhwV43FNmiqVuM
WmbLKq3o403uiRML7FCwl3tNEYCIPbUI/F6cKdlU1bQx25hUWUZedgbmnECiVwjBmPrWw0w
aULCdVWYfZmgaqgmM24J8/Aml9AYNVy/EsWggGuiVNwjtn83tRrBasr8S/6Hs3vw1pQSLSJH
bm3Ey3gw5qbj2PcJEJrVhBavRJzdg6uKTPQOjvoWTAPlrxDVjwCW/GlIGvH/D34Fy0gVL0Ga34
RrsnTBt6a2tp4VB8HF61AktOLq7J07HIFw7Z9cV2Z0jHmxXXXILZ4yH74CNS9jW9+jyOinGUX
nwl3+w9mZFX3YS1IP7/ZqStHTW3XstuX23ZMi0hxHZh01WDxYBIFe5tLnzYQKSxDnNWNpac
PEJVa0BRUJxu2jtaUX0+zGYzmZomoHd1EW1uit+BHjmaYE0VoWgEe3EpWaEQHocDw+miNe
DHabbgLYnDUVxCzNdD4NMPe6csWkDxcIwjcjcnhBOLTuL2z9sA7K7nirBaJ4F/4HbHuTmylI3H
182HX7HKTudf+ZO61f2JbrKcb/9IfsJaUYXK6CFWullxXg2vC5KT2F4GaSsLVIQC4J0zG0s+H1
WhH07GuTpxjJmAfWZFlct1TZmzV624vGoG9aASGrhOsXIV/8QJsJaVJVXIDtdVEW5ux5BVgL
9nwFxBcDDdaKEiocjV6NlrPZicUi2EEA/i7OikoH0WGx0ugsZ5oazOqNxNPQqNTC4sJazHsAT8
VU6fTGovRHgqRN6lcaZCAFggQqlxFLBjAOXEqt73aLSjnVBNJbGuDuwjK3BWJEOPRAiuXkHn
nM9RnS4lhbCPqkh6/22IrbAlLeAn1tmOyeXB5HQT6+nGPWkajpEVAfH32Q1H+WjCDXUEVY3H
kpVD3o9P2eiXZGaXh4zd9iZjt72JtrUQqq0m1tmBY2RFYmqRhgkTrFyFHg7jKB+Fuc/ykXBDHZ
GmBsZuTjKRm700Ri6TqhQNbGuzvi4k1eAyWbHu8tueHfZbZOvgRCDzdjoFG4zRqTPFG77en
egZQq3EEIkQR6G+qeN4dlvzyTSFMDJRdcRmDFUjo+ehfr50moZ/4c06cf4H/rdWxFI9D0v4Tq
+25D6+nGkptH2VU30/riU2QfchSax0394w8Rrq9BdThxnn0Rf/rvhxx/1vm8/dyT7D9rFoHbrkW1
Oxh5zS00vvQM7ukzKfvVddgK+r97G25rofX1l6m+7/fo4RC2EeWMueMhMvfev9/j1wqsWUHVn
Tfs/v5bAGQfdgzWnDwaX3qajH00pOKWP+AYWUH3vK+of/Yx2nqnemfsdwgjr7oJd5/p2IEVS1
l+xYX4F85HMZspuegKis/+OZaMgfWp3hAtGKD59ZepvO069HAle/koxj/0HK7J0+j45ANWXH4
+sc4OzJlZjLv/SbL2P2RQritEOoWbGqh58A6aXnkeDIOcn/+aFd4sHr3lt4QCACZMm8Edt9zF6us
uJdJYj+p0UX7NrdTb7Khv/g3Dm8FnMYN/v/oIUq4zY6/90PvY48kCah/5A1pPF5acXCpuewDH
pKks/+WZiffwiluvIP+kn9L4Iyepe/Ih0DTc02aSfdDhrLj6IsY/9BzuydM2GHukrYXaR+6j4bnHQNf
xzNqDzH0PouaB27GVIDHm7j+Rued++Jcuomv05zT/7UX8i74Hk4kRF/6K4vMu2WSfe9+ShSy
75GxCq1egWK2UX30zBaf8DCMcovaJP1L/xEOJa4+58yGcFePo+upzll16dryHs8fLmLv+RM6h
R/WbkGihK1vvMbqm69CDwWxjShn/J+ew7OBZTZCDIWNTEFeW0TMMAY0nm5Ux7re7qrNhh
6WBFollfQSNdDbSKSlieWXnUukqQFLtI56JEzHR+8CoP3oGlyq1fj//XcwDPJPPI3KO36H1tMN
xCvbVt91AyXnXUJw5VlaX3iScH0NEG8x4XvkXo489DDuvupS9jv0cJ5/+S849z0EPeBn9U1XUn
zuRTS/8jxdX366wfj8C76j8o7r0cMhAMK1Vaz5/W8J9t4x3pD2995KJM8QR8pr8ngxuT10ff4Rz
a+/TKixju5vv04kzwBdn35A27v/SvysR8LUPnof/oXzATBiMWreCe+Bd9t/ou8Cf6lP7D6+ssTzz
FUtYZVN1xOYMVSlv3yTGKdHQDEOjtYdsIzBktWD9q1hUiXrq8+o+mvzyUqalfHT+H+a35NKBB



vRXPgOT9nzQ2/ldJYD4Ae8FPzxzvla6on9NWntlcw7/++gJ6b6Gu+V98ip6ZtC0Dt6P1dAHxn
vUrr72EaHNj0nu45o930fPtHOr+/AD0VsD2LfiWwMplqDY7q2+8gljvONefnm+/puGZRxJFwnrm
fUWoajW00WMJ11Wz+sYrCNVU0fDys3TP+TyePANoGrWP/AHf9/M2+tpo/h7W3HotodUr4jF
HllTeel2BxQvp/u4b6v/8YNK1G55/nHBDHcsuOTNRPE3r6Wb5ZecQ6D3H+gLLfPy2ksS60bDt
VWsuuaXRHvHGyHSYaNfXHoTaD3gB5MZ1Wpdt89ixYhGExXthRBCSAK9zUSaGxNjr3PCIKQ
PdgGLFXPVuvYphq6nVLWotrWihQLYSsr6bXXiikbo6ewg6PPx/ZwvUCb1FqPRdXS/H4DOz1N
bRK0VrqtO2RZYupBoc8NGn1fHx++nbPMvWoBjdLzIWfsHb2NElvTM/SrluM7P/ovWOxUs2tFO
x0fvpRyzOf2pN1e4piplm2/+XMJ11fEPCn1ovp5EQiHEcNb1v4+Tfm5uTm6DV+T1Eq5Nfm+4x
k6k65P3seQXsnx1amunYGN94ouotWKdHUSbk1vDmTMy8S9J7dve/c0XuKfuQs+3c4i2tWww
9p75qes9u+d+hWtKvEhhcOUylk0NoMXo/uaLIGMDy1NbafUVbWuju5+2V+GmhnXJeB/tH7xNt
LWZaFtr0nYjGo3XqehHqDZ1bPUv+UHa6lm02ngbKwt6JEKsqxPzel07FEVBtdnR1vubKYQQOz
NJoLcRc2Y25ux41efQmIXx4li9bNEYWvG69k2KxQJq8q/C5PZgsjuJdrQIValeK2yxYXc6sbvdVE
yaAtVrEvtUV7ztIWeX3TcYn6WflmK20pGYMrM3+ry8s1LP6agYS6j3A7I31z0xVDOufqZpuqfPw
tTbSrszcA9Y1a/MQyWvi3B1rKPGOM1vzD+mvehWG1YcoeuoJAQ24p7xq5JP2dl5yb93BWNy
slJ3hasXIV75u5EW1soL02tB2DLzYf17l6pDieW9c4d6+nGMaoi5fGuiVMlrl6Bo2Jc0rrilOP66Q7
gmjiFUG+/ZmvxCExZ2aDrOPupYG0vH73Bc008fZ9zWuSU7ZbsHBwV41K2e2fvgSkzC1Of2g4
AqCqW3P4LMVrz+xlB8oS3RWESAcjvEp3EY4TKyrA1M/bTNVp1MSaCGE6EMS6G3EXILK2L
v+hGK1Ea6vwVpUjL33Lq3960+JjSjDNj3eaqntnX9RcuGvEkm0YrFQeuk1NL/xCqrbTcm5Fyf1Y
3ad/DPe/eRDLrnh98z55APOPefnBN5/CxSFkl/8huZ/vopn5u5k7nvQBuNzTZpG4U/PS/ysOl2M
+t3t/RYS6yvnqBNwjJ2Q+Nk5fjlm15tYexu20pEUnHY2julSvLvtnDTayj5yNHk/Pjnx8nhpPzy3yW
+ZADIOeZEPNNSk+ot5Zw4heLzLlI3TZebMb9/ENf4yVTc9sC6hMBkYszvH8AxcsgyXVuldMna
50A8fQoGutes4LSLfpX4+U8P3M3l6+9c1xdZUcg79hSCE6ZgKRIBqd/HzD32ThxfMKIUI65TcsF
l68Yos5lRv7sdtaAw+T185HF4Zu5OVp9Kv5a8AjL22p/AssVU3PYAluwNt5Pz7LonmX1qEVgLi/
HO3B3fwu9QHU5G33A3ropx5J/8E7IPOTLpS6/sw3+c3L6uH5bMLCpuuS+p7VXhGRfgmjQV7
3pxW/MLKbnw1zhKRzLmzofXfenW2xbPObb/sdl1YQolv7g88bPqcDLMzoc3q4CaENuKHomkf
Am2lsnlJtbVQayrM/XLIuJ/rzW/b1uHKIQW4a0sdqGDF0nuHo5odoarLi5mLNzCFWuBsNay8j
C190FWH2YTSa8E6eg9HQTqq7EnJINNCOT4JqVWJxObJ4MjGCAaFchnjeDpkAAq90FYjLjzM
wiKxYmVFWJNb8Axe6ASBJHmPGbbNESbKwnuGwx0fYWHOUVeGduXnXYUG11vFUN8Ts3kd
ZmFNWEa/I07L3ttPRoIMCKpYQqV2EYBs4Jk3H1c4cnVFNFsHlIJqcLx5gJWDIlyB/Yib0LM7yO
4chmxznbsZaNwjlonYWurBlcb67EVFuOoGCdtrsQO08Yq0tYS//8+HCLkycQfixH29dDd1krx6Ar
ycvKJNNYRrq3CnJtH0JtNNByCWBRrRxtkZtMcDBINBikpLMTp6wEMTA4nkaYGrEULuCFpWozx
xN/Da1ZicrlxjBmPJSOTaGcHgaU/E03qwJKZTbSnG9eYcZv1JVVW0ox3/0kXE0tux5OahhyNE2
5pxjBqDa8oM1N4kPIRTSbihjmhbK5bsHJwTpmz2+BGsXE2oejVmybybOMeMTSU00s4PgiqVo
oSCO0WMT1fkNTYuP5XW1WPMKcl4dj2q1bfD8sYA//vp3tGErHYVjVAWKil1ERJlhHW8WnPwj
co44Fvek1Nlh/iULaf3365Scfyn1z/2ZssuuSdq/6obfMPbex/D07QsvhBhO5A/QINsuq3ArillJ9A
AaEDMMY+O3FdlSfWF9LqHIV0cb6eLuq3Dx0fwDN34N95GgcE6YQWr4Y3+KF2lpKMGVKEatej
cftRTeb0TUNraWZcE0lJrsDw2zCFw7R7c0iGPBR0N2FxdeDkZNLyYbIV2cnRW4PzuwchKEg0
a4ODC2GYRgYAT+x9IY0v7+3R6uFWec7powM3FNmYM3OJVRfg2/hfNA0zFk5mNxetFCAIrf+
gXP8pE3ehbaPKMM+ooxYTzeB5YtRUDB5vATWrAbdwD6iDNViwT1pallV7X7PvVqOvbScSFs
L/sULAAMUu7z4Wiy2ttKROHunhMYCfolrlhJpacI+ojyR9Gq+HvzLlxBtb8VRPgphXxiU3g/ZZpc
bz/TUu9qq1YprwmRc/UznFGJ7p0ejBKvWEG1ulFxfgyWvIN7fPTefcGM9voXfoUciaMUINDQ24
HK5yDSbiDmctNTXYfH1oLS3YS8olhQIEA766A5H6Gprwe1wUqRrIMUimHOzibY2oXgzUd1utE
CAiGEQ8vloWfgd7u50LloST0BVhe55X6Goajxp7miP13fQNAy/j0hLM/7aGix2G3ogQLS9DvtJK
a5JUzG71t31smRlk7nnvgDEFn0Eli9FD7kxORxJnwD0aBTN14MINw9rfgG+7+ehqCrOcROxbqC
/sqHrBFYul1xfg8mTQaS1ubf//AT0WlZg8iUYsRiOMeOTWtspJhPOsRNxjp2YdL5QXU28QJrZH
L9u79IYs9MlyYbYrhjRKOOgPnCbvZIEWlvid6CdrpT9qt0hd6CFEKKP7TKB7nWgYRitnz4svfzLF
tP15adU3nkDRiQMMP5JPYHS0kTnJx/gmb0nGXvsS+3Dd8cfYDJRftVN6IEA7e/9m1BtFS0vu4
3FI5+fKCSWd+wpOE44nefuvY2fHX4UDQ/8HmvFOFbsth9PPHgPEC/scfndf2TCrN1ovvgMQK
H8yhuouueWRPXXvOP+D8PQaX3jNQAY9j6A8qtuZvmvzkmsKbSXjybv2JNpeu0vFP7kXNbcdA
UTHn95k71Ko50dVN93K41/eQoA1Wan/MobWH3dpUx49IV+v+XekFBtNSuu/AVadxcZe+1P/
VMPx3eoKqWXXUvmAYfgrBhPwzOPUv2HWxOv4/gHniJzv0OoeehO6p+MP0axWpnw2EtkH3
DoZl9fiOGm45MP0Hq6WHnNLzGi8f6thT85l4KfnMuyi39GaM1K7Mf9H6/X1FlxfHJ75efyQV09
7739L64590KqHr47UW26+LJrWZGZw90330h7U7zg2MTpu3DuQYegf/A2Bf/3Mxo++YDsAw6
h+g+3xQNQVdwxXcFSxUth0/+g7MLLqP7DbUQa6ig8/Rz8y5fQM/dLAMxZ2ZRefAUrr7mEsX
94DP+KRirrvD4Rd/EFI1L6yysxu5OLF8W6u6j+4500PP0IOpve/vNLZO9/KN1zv2TRWScmigFm
HXQ4JoeT1rdexzl+EhMeeSEx2yTpdfvsQ5b+4qeMvPlmqh+4PdH5IOuQI3GOnUTdo/cC8anjK5
7+20a/YPMv/YHFZ58UL2pGvMbDuAef3mSPaCHSQY9GNziF25yRRayjWhXR7xv+3pUu10SaC



GE6EPWQG8Flxaje95X1D35UCJ5Bmj+24t4Z8aLbWXte9C65Bni7VYevgf7qAr8SxaSd8xJ1Dxw
R1IV7pY3XiXcWMcuk6cRfvphjFgM49BjeOqhP6y7tmHw0PVXEg0Gcex3CDIHHkFDC08kkmeAl
n++gnP0umntwRVLaXvnjUTyDBCqWk20vQ00jeDKZZi8GdQ/9Uj8j+1G+JcsTCTPAHo4RP2zfy
Zjz31peObRTT6+r+65X9I953OyDzua+qf/tG6HrIP36H30fPMLgaWL1iXPva/jyusuxb98cSJ5hnil
0VXX/JJw08ariQsxXAWr1qB1d1J5x7okFKDxxaclFLlaM1KANpLrVLI+++w78zZdGoGL/z5YU
4+5ScEHL/Xqgmgb1/8/3/PkkkzwBLvv+OKpOFaFsLodoqco86jur7b18XhK4TePIhtKAf0/Gn0f
XFp0Qa6sBkwpKbl0ieAWId7XR8/AHuabtQde8tBktWJcVd//gfCSxbkvl8/UsXJZJnWPvevoRgd
SWrb7k6qZJ+x3/fwTkufnc4sGwx7f99N+V8kaZGV11zCZI77EvrW68nkmeAjg/exmRfNyU70lhP
48vPYPR5nfoynI2GF55MJM8Avu/n0fXFJ/0eL0S6GeEQqsXa7z7V5UIPBK0N2JybeAOtC81g
a68+ybCDXWDHqsQQmzvttcE2gDeUxRlnqloF6y/U1GUCxRFmasoytyWlg23RNNWtFAw3v6ovj
Zln96bUOt9EuvE43q60brj/VTNWdLJH8LWirS3ke31EmtvA6AnHE70ZU0cEwrR3d4KZaOwFZXE
11dvlA4A24iyfvukBlevwFZSSmDIMuxlo/D/MH+T3zb315IIXFuFNa+A7m/noAcDG318X4GVy4
D4FDPWW50vh4LE0juJNKdeT+vpJtaROkkh0txlrKtzs68vxMZsL+PNWrH0djCbE32J+1o7lihm
M+2+HgD0rk56eltQuS3mpC/ZAeyl5Syb/23KuapqqrHk5BFYvgSTw5WUdEP8velSVfwoBHvfw
2a3N/6F3HoCK5ZgLx9NqHI1Zk9Gatz9vL8j/YwxkaYGYh3t/bbK6jvW9Xz3dcr+WFcHkaYG7G
WjCKxlbZen+X1J1fm7v/ky5bVKXCsUpGdeaqst+5IF/R4vxObaVuONFgyg2u0buibmzOz4l+iu1
CJiqs205u9J2mbEYjQ8+xhd/bSFE0KIHD32mkDvYxjGTOAI4GJFUfbru9MwjMcNw5htGMbsvL
z0tR4yuz1YcvOTqk2vtbZqtmKxpLRMshaPwJyZBUCoag3OieutE1YUHMUjWFNXi21kfP1vpknF
7nQmHZArK0t2YTHGgnn4Fy/A00+LKdWx7jGBFUVJ0vCwlGPc02YSWLEU76zd8S9aQNYhR2
DeRDEee9moDZ4n98gTkqrcoqnT+XcvtXGgd71jYU4Ro7u93W0FY2A9YruCZN3eAaSCEgan
sZb9ayFhaj+XqSquEDoCjYR8Ur/RuxGIWZmfHNUflk2WyYLRyau7qxrFcNWluxID00PYL1TRg3
nnBjHd5ZexBta04aSwAsObm0hsJ4wsHE2BPr6sBakNo+LmOPfemZPxfP7D1TE2xVxd7PtGd7
2cjU9/bkadhKSSnuZxrg1/2QYen7Lfkf+KaPI2e+XPJ2GPfIP0mb0bSnfHco0/odz0oxKsW5xx1
fMr2jD326+doiTbftbhp9GAg5e9rX+bMLIKrV2Dur41VP2ugA6uWo4eC+BZ+N2gxCiHEcLfDjTC
GYdT1/rsZ+AeweeWh08A7e0+Kz7ko0UPU5PYw6oa7aXv/LQC6533N2HseSyTMtplyyq+8kY6
P3yf/5J/S9sFbFP3sAhy9La5MHi9lv7kee34hituLdvq5WEtHEV3bX7j6xtvJyot/+M0tLuG3f3oai6
qiNzbQckH5B51As7x8d6oqtNF+bW3EaqK35VWrfAKz76lrlMOI++E0+MfTBWFnCOOI9bZgXe
3vUFVcU6cQsEpP9tkxVjnxCMufsRTL09px1jxpP345MwDIP8k04fUMVZ78zdGfHLK2n+5yuU
Xf7bRI9aa1EJZb+5now998U5dgLjH35u3es4opzxf3wG59iJjLv/iUTCbh89ljF3Poyl9zghdjS2giJ
cE6cy8uqbsY+M9z02eTKouP1BXJOnkX9i/P3tmvMZ5119Ay/+9QVYr5Rxc1c138M5b/0S94FeJ
3vKWnFyKLvoNxaNGM/vAeN0Ak9nMST87l7xVy8g+5EiMalSeBd9Rcccf1703C4uxXHqIpWXlx
P75Cp5ddiP7sB8D0DN/LsXn/jLxhZd3172wl41EUUVGXHGZrinTsfcWBjR5Mhh3/5M41/8yAHC
Nn8y4+x5PvLcdFeMYc8fDWHpZKLvyBty9RbpUu4PSy66l46N3QVUPON8MvY+IOV8lswsxtzx
EJrfh3vaTny9xQUVq43yq2/G5HAM1ojmHH4seUefuNHfQ96x/5d4zphMFJ9/Kd5d99yM36AQ
Q08LBFBtG64cb87lJfXxg+rqJ4G22VISaN+CbzFn50gCLYTYKW13bawURXEBqmEYPb3//T5w
i2EY7/R3/PbQVkbz+wnVVBltAEN1urHk5UMkjOb3YysegdnjxbdsMbHWZszZOaguN7G2VjCZU
AwDPRzCnFtAtKUx/gHOM4lf1wmEw4SjUdwmFVcsiu5w0RmNEPL5yM7JQ3c4cOkaZr8PAN5
Mbi+KGm/RpNrt6MEQisWMEY1hZsjEOXositmMFg4RWLYYPRrB5HJxGIA6KEQzrETsWSkTrH
sj2EYhKorifV0xf8wx3Ts5eWYnKI/gDdFj0YJVa/B0DQUq41oUz0mtxfbiPKkeEJ1NcQ62rEUFGL
rrXgLEKyUroVuwIzUkviQL8QADas2VuHmJqKztUTbW7Dm5OHqncmihUMEV62lf+D1emnz+T
EAj0mlOxgkpbtkOhxYAj5sGZkEu7sxPF66lmeiWRBOq5UcixlFi6FarOiRMKrJDA4HRjRKpLuLqM
207nTi6ulBNZkwOR2YetvtoZpQbFZibe2AgWKxovV0YSKowgiHUSxmNJ8PPRzCXIKGvbR8o89
z7XvbWlyCNXvdezvW1UmorhqTw4WloIhWTSWKqmlvH7XRFIPR9lbc9XWobjd6lIDJ4cBeNgp
D0whVrcbQNOxlld497mvtWO/YjZjLx8trfDEQAZeGNoGI+Mz2XyC29s8Mvtusf/SMfH71Fx2/
2JL/TXan3rdRSLIdE33JXYtvK3v0IPBmh779/ssaA20flCCLFdkjZWg2x7rMJdAPYjd5A3Ay9tKHI
OF9+iBfgXLYDW1YFj9FgMk0po5XKsBUBVybDaCK5YQXLkc1eEg1tWO6nQTqashVF8Tn67d2kK
oeg2KyYRj9Fhi3V0EV63AVlpGuLWThpYWqutqcXgzyC8ewYJFC5g6cRK2jg580Si19XXUu9wUj
ijFrOsEq1ej5ebR1NRETk4uFouFFUsWM6qgAHtrM6qm4RhZAWYTwSU/EPP5cE2cgh4IEGmox
5ydQ7S5Mf7BsQ4G1WbHVlhEoHIVut+PY1QFkdYwIs0NuCZNI9JQR6yzl/5B1eUmUl+Le9l0XB
OnJF6jaGcHvoXfEaQuXfPQiGp3Eqpeg3P0WFXtZ2Be71tu1WLB2bdPdO+dtfXZS0qhT3uZtTZ
W+Tawajn+H75Hj4RxTZqGe/LmVwgXYnsQaqilZ+5XRJobsRYUxXvF22xUNzfT1d70JNWM77
OPMIXDNEcJVNBWoJhMjK4YQ6HDQXDnkqLdXerOmoejRjTqiPW3oZSVEwwM4cV383Dk19
Ac201qqpSMXY8+XYHajSK4esh1NJEzOfDXD4ai9uDpa6GaHsLVlXHNWz8C7/D5HTIGDsB1W
olsPA7NH8PtulyzLI56LEY3XP+I3j/a6Egvlzf3PG5gzs3BNmoZqsdAz/xv0cBhbUQmKasLkzc
BWWk6kuYGe7+fhGFWBc/xkgquWE1i5DLpbjpyo333NebsXGJd7aAoOMeMJ9xQh2/Bt/H2d
mMm4J46A5PdgsU7F0t26pds8VZV8TvhofoaOj79kFhHG86xE3FNmY6pn+mvJpdLWuGJ7Z4
W8KPa7BudGbZ2pke/d6DtzpS6lr4F88g/8Sd0ffkpoao1OHpnlQghxM5gu0ugDcNYDaQuKt509



Cycz6rfXob/h/mJbaNvvY/aR+/DOW4ihT85j+WXnp0ohuWoGEf51Tez6obLyTvmJFSbg+o/3lr
W04VtRDm5x5xI3aP3MfK3v2fFIRfRc+YvuOP6q1g7M2DkhEmcfffk1dNx7C80/OoZ7broucd0x0
2Zw7JnnU/j264S++wb3RVdw74P3cNGvrsle8NN23aWJlmSqzc7oW++j7Z1/EWmsxz19F1lffk
Lx+ZfS8o+X6fjovcR5S86/IJqvPiX30KOpvu82TC43pZdeQ7S5kdUvP7OuireqMur6uwHvRVL1D
b9hyotv4pkxGy0cou6JB6I79P7EOXOPOZFicPdcz5n9C1/oPAn5w5omveW8i9bwqKfHk20rTX
xOkx+8U28M7fbVQFCJAK3NbD80nPomTcnsW3MnQ/zZVMTf779Ru64+U6Mjja0pgaql1Fu/v0
NBHriBX9cXi8333o3wZuvxLvb3kTaWtD9PhpfXfDB333k8bDH/tx85slEe7sBZOUXcONjz+Kd+x
VdX36SVHyw5la76Hj1BUK11Yy59T6W/+aCRIExe/loyn7z01Ze+Qugt73dVTeBocdb/cVijP3Tc
2idHaz+7a/WxTB9Fu7pM2l8/gkg/mG+90lrMefk0vDMI7S98y8ALLn5JlJ4CtbcfXSY53jJtL15
WfkHnU8a266mskvvMGa266l++v/JY4be+9j5J9w2qZf78Z6ll9yTllhsnF/fla8o0/Y5GOF2B7pw
QCqY8PrnwHM3vhML3N/RcQcDrT6mnXni0YJrFyOY2QF9pEV+H74ThJolcRORebcDJD/h++Sk
meAmgdup+hnF1B0xgVU33drUiXp4KrlBJYtlffoE3BUjCoweAFaT7wCd+5Rx9Pw7GM4J0yhZ9
4cbPsftNP/Im+0+orly4m5PdjmrbTz36x6Trrlwn7a2VrS9D4pf64Un2H+/A1m2eCFFgZ5E8
gzxNINt7/yLrIMPJ/vgw2l65TlyjzweIxxOSp4BGI54gqy9DyTa3oo5OwfN78P3w3fYykYmtcBC12
l84XHc02aiBwM0/OVJDMMGuHoldY89kHT01j/Tsae8cl9IXfeQKimckCv+5bq+OS9RPIMve22
nn4EvXfauhDbO/+ShUnJM0CXzc4Td93CIUcT6ah0fPZBwRWLuV/879NJM8A/u5uPv/6K+xlo
8jYaz9MNhuNf3026Vx6cwMfv/G3RPIM0NHcxPvx5mDOzEqp3N/04J3knncl35xObWP3ZdUn
TtUtTr+fuudzqmHQ3R8+gG+ZYvJ3Dc+TpldHmru/33SOX3fz8Oava5gktbTTWDIUnwLviVUU5X
YnnvMidQ8cEfKY20lpYRrq1CtVmK+bnyLvk9KngHW3HYt4YbUjgnr8y9ekFLFe80tVxPp0+ZLiO
FE8/s3WkAM4mugURTU9YqVwto2VuvGIWhLE2a3B9Vux1pY3G8HECGE2JFJAj1AsX56IUY72
rFkZqFYrf1+yNLDQcyeTBSLIUif1jOkxYleDGDJzom3bMnKTurFulYw4MfweOlSTW1pocVihHqT
Qa2nC7fLRWdnB0Zvm6ykOFubUc2WeiKu6ygmU2L9c1K8oSCqxUK0vS3xrXS0pRnFILq+L9LS
nCIE86pajRGLxfuz9re2XtPi5w/4N9geZrCF+2kxFqpegxGN9H00ENSfracnZVsUCPp850Tmgt2
JohkY0RiN/YwfjY0NmDOz4v3kDSPxPlxLzcqhua4m5XFtjQ0oltRJSrHuThSLDWtOTr+tpjRfT1J
F7GhLM+gG5qwcABTV1G+rK0NLHosiLU1owQCW3scBmJwuYt2dKY9d+5y0YBDVakPzdaccE
uvqRA+lthXsL/71Rdtb0cJDM2YJMdi04OYl0CaXu9+1zKrdnlRELNLShDk7/r60ZGYR7qcVpxBC
7MgkgR4g5/hJKe2Ucg47htZ33sS/eEG8Am5fqoptRBldX35CtKsDz8xdE7v8ixfimb0HPd99Q+Z
e+xP77EM00/ak5lebTOSPKMP46jMOOuq4pH0msxl3Rgbero54bHsfwJdfcHkiVPQRiUXAQHI
308QQnU1RJoacIwaQ2DIMhS7LVFJey33jNn4ly3GNWEyoao1AHj32Cf+oXS9adc5hx9DrLMTg
MLTzka1WLCXjcJentzmyppfSKwn/qE2Y+8DsBwnrmPeFvprZ1P4k3PjFXeFGAacY8enjDmeYI
DZ+x/ER++/Q6SjHbW0HGvxCPbfZ/+Ux++39774llymM95/v6U50DFgrtnlpB5/wfymPm7nvAWg9
PSjrFeTyHvAjAt/Oofkfr5B3zMnJD1IU7OWj0Pt82M7c/xCsObl0/e9jAMKNdeT86Jkh1ksKBZr8
nVm7o5rwhT8fXo2d8/9isz9f5T8WksVxRRP9C3ZOejBAM5xk1CsyeFLOfXyREXFKc9zfY4xE1Je
77xjT8ZWuOnHcRE90v1+TLb+e0CvZc7K2WD7SZPdidbn5kGkqQFLVnb8cRIZRjldoYQYuciCf
QAeabPZtwDT+GcMBmTy03+iaeTe/SJ+BZ8S+dn/yX3uFMO/NkFmNweHKPHMub3D+KsmIC
1qITmf/wVx5gJlF52Ddb8QnwLv6X43EviOuhwApWryTvkSA6sqODY08/C6fFQpN4Sv77/Ud557
SUcx5zE0bvtxtGnnl7D7WbUpCn86r5HmFhUTOg//8R12l9p22M/DjyxxSOruDvH7yL97JrsRaV
YM7OoeSiK8jYa3/a//s07R+8Q9mVN8bvPms6Y+55FM8uu6l6nGQfciS5R59A5j4H0fPdXCw5u
ZRffQuxrg5a3niVitsewDF6LCaPl4JTzyT7kCNpevU5Rv7uDrL2j7fBseblM+HRv5B180GoDicZex
9A6W9+R8u//kbe8acy+uZ7MbtT11ltk9/XrN0Y+4c/x3tvZ2Uz8tpbyT44NakWYnvIHD+ZSc++jn
PSVEwuN7lHnYBVVbnw6huYMGMWc9vbcEyYjKmgIHG5ufziyt+SIV9AdkEhl9z+B6bOmlVj1Bh
a//13PDN3p+TCy8jc5yBUhxPvHvvi/dXvaK6v4+SLL8eblU1BaRm/vvdhRmdn45oyg1E33lVr4lR
Up4uc40+l4LxLiFwvwbfgWyzdo+Pd54M7KMqGHvf4yh2e+L9VnzuxdjLR6M4HMR6usk7/lS8
M3ej+ILLyDv+VEwuN87xkxj/p+ex5Bdiyc3HklfAiF9eiSkjC7Mng3EPPIF3t71RHU6sufmU/epaCk
49E9XpwjlpGiOv+z3tn3zAyGtvpePzjxhz5804p89m8rOvx4t/udwUnHoW5Vfe0G8hsPW5Jkxm
0jN/xzlxCiaXm8KfnkfppddutLK3ENszLeHsW88gbaVIDLqt7/vd5/qcKAFkhNoc0a8VaQ5M7Pf
mShCCLEj2+7aWA1UutrKRINBjEgYxenC0HUUQFFVDMPA0DT0nm7QdUwZmfHWTCYTRjRCz
O9HwcCIRDC0GlrDhWpS0bq7Ub0Z6H4fmslETyyGxWbHbjlR8Puuz2o0SiKct2hECbVhNNkw
WlxEfP10KPpWCwWXC4XZquN7p5ujGgUt9WCEY1iKyzC5HQT7erE0DUsgVno0Qix9jZMbg/Rr
g50nx/V6cDkcmPNyYsXHlPUEFQVRVWJ+X1Y8wrQqYFine1Y8gsgFsOlxrDmF6S8RlowSKyzH
XPva6D5erDk5KGud2doSH5fbS0YmoY1v3Dlry2Gje26jVU0FMIwNBTVHB9rADSNWCyG2WYjF
gxghMNgaPj9QTCbcVkt8SmZmo4RCWPkYCLi60YNh9GiEUwZGWB30d3SiMPjxdfjQ8UgOzML
xWYjGAXiUQwohE0XcOanYcSC8fvRmkxFKsNk8tLtLURk92BOTsHizeTaFsLWjCA6vZgychC8/
vQgwHMGVvmJ978WDBKsXIXJ48Exlt7KKtLSjBbw8dUiwVb8QhUVSXm60Hr6cacnYvJZotXE
m9tRnW60YPxCSNGLAYKWHPz171m3V3ofh+W3PwBt5iKdXei+f1Y8wpQzNtdvU0x/A3ZeNP6
nzdoeP5xyi67ZoseH+1oZ9Xvfs3u8+JrnauvZVlcpP5J5xGuLGe6ntvYfbni7Y6TiHENiNtrAaZfC
rYAMMw8C34lrZ338Q1eTr+RQtwVlWjuHo5oao1eHfdA5MnAz0YwNB17CPK8C9agCW/gNCa



VYRqKsna/1Csl8roWbOK7k8/QB05BsvMXTHFYgTf+gfqrD1ot9nJMnSyRYseYXoPh/hlkYanW
6+nPs1Y2fMorW+hp62do7cdz9Cn30IzjOZ+x1Cw2cfYfV4yNxiX0ytzbT9910CFEOWFZUQ7Wz
HNXEqsZx8ur76BN+C78g5/Mfxnqy+bgJLfyDW00P2QYfR9cUnxHw9Z013MCZvBl3/+4RwbR
VZBx+Be8ZsluU1tPzrNczeTFyTp6EF/LinzKDtnTclF9KzhHHkrHHPlgys5NeQ5PDgclRkvjZvIH
pYUPBkpO36YOEESCMtGKB77lf4VyzBOW4SsfY2euZ9GW8ftef+fPL9dxxw4KFYlv2A77tyvD32F
DCZITTU0v31/7CNGY9j5u6EvvgE1y670WB3Yu3pwrpoPtGGWjJ22wdLbj7R7k4Cvh66C4p545
UXGT95KrvvfDKt18TXLKQ7qmzeGPu1+w7azb25YvRmhvJ3PsAApUrsWbnYcnJRTFbMGdm
0fTS03j3OgBrbh5t7/0bs8dLzuE/xpKZTfM//kqksY7sQ44k5vPR9eUnZB3wl8yZWbT84xW8s/c
g68AfJZJniM9eAfAtXkDb6/8iXfDn7tEn4t1tT2xF68YS1WJZ9/NG+tZbvBng3by+9uszezMxez
O36LFCbE/0YAB1E3egN0Z10OK1TXqFG+ux5MTXQJszs4i2tWIYxpB01hBCiO2B3IHegJ6F3/
HDKYcz4pdXUPfYASd/Qsa//lksc6OxDeJLvoN4fY2PNNm0vzXZ8nc9yCa/vpsUtXn0I9dR9PLz
xDpLbJhKR1J+Oxf4v7g3wS++BjX+Zfyt2+/5aSiAvKzmECxO6gOBbn+uis45uwl+ebDd2moXM
1dd9xH6K7rE+dVzGbKr7yJyjt+hzkji6KzLqTmwTsBsJWOJPuQlWitWUnu8adSecvVRNvbGHntr
cQ6O2l6JR5j+dU3U33/7RiReGGd0kuvpuG5x4l1rXuOZVfcQGDZYlrf/BsAqsPJmLseZs0t1xDtU
xBt5G9/T8m5vxz034MQQyitd6DbP/gPSy44lQmPv0xw9Qqq7/t94r0JUPzHp2l77AHCixfgnDC
ZjP0PRYIGqH/6kcQx5rwCiq/7PY211bSHwmS88Ge0PkW3Si+9GtXmoO2DtzAKi3l+1Rrmf/U/y
sa057lHU5oRbK3XH81V1x7ExnPPZJUUKvs17+l7smHyDv2FBS7DTQD1WbHnJFB1V03Jo5T
7Q5Kf3UdVXeuG6/KfnM9tY/8AT0YIP+kn+BftAD/koW4p89i4hN/Tbpr7F++mUnH57oVgCb34
JKiGFkyMabhhheoPPzjyg++xdb9HjDMFh0xrHstbQFwZmhz00JXOv/fHsEq/psvick9l1zvK0fk
kuhNgo+XZrkMka6A3o/PS/8enY0Ria34diMiUlzwBNrz5Pxuw9iDU34lVwLYrZnJQ8A9Q//Qg5R
xyb+DlaU4mprQn18Pi28GsvMGBMWHrKRtP2wVsEviymNRRBi8XwZudQt3olk3fdA9PnHyad1
4jFCCxfjG1EeTzh7fNFSLimErM3g46P3ydSW0XBqWfhnjaTzv99jGKJx2jJzSNcW530AR1ISp4B
GI94gqxDjkj8rAcDBFevRA8Gko6refBOQv1UvBZCbJoW8FPzp3soOP1sOj/7iHBNVcp709rUSHj
xAgCyDzoCRddpeuX5pGNIU1o3V2saGwk39CSkmeApr+9SKi+hqlzzsf/zr845KB43YLqFcsliar
gg/nfoSgK+bFwSjXq5n+8TPZBh9P82l9QTRZMLheeXXal9d+vJx2nh4JEmuoxedbd+W3512tk
HRAv/tXyz1fI0AwIN6CKrhYedLjfQu+TUqeAarvz1lBVCbB4tGEDdRBGxjVEUJd7KqncddKS5
EXPWuhln5uyceLV9IYTSUGCvQFGLNq7prm3x2k/d+qNWAxFUTZ6DFosUSF23QONRDVrlxb
DbDLHk/VYDAwwSD6PyWxG6afdlBGLoZhNKefsG4uh6ShmS3wNtqYlTitq789rbWDqlaHFUqZl
GZqW6POaFMswN80gRNoYBkY0imqxYuhaSksnAKNPv2WU3pol/Y0Luo6maWDoKfvoHbMU
VQXDSPpK2KahtnZ/P29lI6ahmM0Yerymw9qxr79Y0TQUU58xos9YZeg6qOuubOjJbbUMLTVu
Q4utG2eFEAOi+X0pVekHyuRwovnj07ijLU2JKtwAlsxsli1SiVslsfOQBHoDsvY7GCMWxeRwolht
KKoppd1T/omn0/X1F1jyCnCMGQ+KktlGovCn59H27puJn815BcRy8uGT9wBwnHA6CxcwtNv
SSPYhR2lfNYZchx1FUQj6feQWl7Bwzhdo+x6CHKCi4Jo8jVDlalSnK94CpjeBtfQW+vLuuhe2klKa
XnmOnu+nkbXPQYkYI82NOEZWJHo4YxgoJjOq05V0mYL/O5OuLz5dd1mLBde4CaAk/69TcuG
vsPZZoyiE2Hwml5uSX/yahuf+TobeB+AYOWbde7OXVliCdVQFAJ2f/RctFk1pm2fOyMKSnC04
ESNos9iT+jED5B1/KuacHBpfeR7nfofy2RefA5A/ohRHXTV77bobsWiUdocz5Y5V3rEn0/7hf8g7
9hRiPh/Rzg6CK5eTe8RxsccpZjPWktKKGtu5x5lx8fvA/F2UmtbWjkqxuEcMz7p8e6pu6T0rB1
x0RVJ07yFEJtPD/g3qWL9xqgOR7wYYCSC5utJmmFizpBK3EKInYusgd4AQ9Po/nYOzX9/icx9
DqLzi4/xTJ9N97yvCFWuihfDycrBiQxdA17SRnd877CMXIM3d9+TahqNblHHIt9/CS6F35H9/tv
oVSMx7b/oSiGQfC5R1H3OYg2TwbZqkQOApacloXwGH9NJY3uDN778F1mH3wYaxYtJBrcw9
wBB+H/999RLFZyfnwSnR/8B8VsJu/oEwjX1tDyxis4J0zGNXEq4doaPLN3x5ybT/tb/6Dn2znk/
99ZmJxONL+P7jmfOYXC5B11HC3/fh2tp4fcH5+ENSEp1n//nXBdNTIHHId3930Jva2k4YUnsW
TnkLn3gYTbWsncdQ9a3/w7/qU/UHDyGWQd+C0seamVuUyRtK6Bjrm66bz848J1lThHDOBW
Ecr7e+/hebrwXro0Xzw/XwOP/wo+OZ/+ObNofCMC8BklrRmJZ2ffYh9zHi8hx5N97tv4t3nQBq
dbmydHSiff0i0vjB+Hi0agREJ0V1bg2/UGJ760wNMnj6TQ44/GdeqZXR9/y2ds/bgk0/+y9EHHo
z5q8+INtaTc/gxhKorMXszsY0oQzGbseTmUfPAHWQfdgzW3Hya/vYXzBlZFJ56FqaMTBqefoR
lc2M84fb7aP/PP8k58nisRcU0PPkwGXvtT96xp6Qk0AA98+fS+NLThGqqKDztbDL3PSjpjpcQO
4AhG29WXntpvA3doUdu8TIW3XAFY+95BGtuPt8ffxDj//h0YI/Dc4/j3W2vLV5jLYTY5mQN9CC
TBHojwi1NaDoYkSBKNlpi6JHovEp2YaOrhuova2rMAxUqwVMZjCZ0XQds6qCrmGYzahmCzF
fDypgRCModie6YWCy2tCCfkWuD1pXO0rvFE4FAxxuwoARDmEYBlZTfPqkbrGgRiKYzBYwdAK
hEBazBbvVjMmTGa+GqWvofj+q04nFm4Eei6Ga100VV0wmol2dmKzW+F0qw4hPy9Z1Yn4/GA
aWPTvtjVgMeqeBq31auqw97/qi7W2oDicmx9Z96y3EEEprAh1qrEfXNIhGAdAARVFR9Rg6Cor
VAtEoqtMDRoxoTw8mhyPenskUJR4aysjEgHVFJ+yaejx1R2xGlRnIHloeux+HgT9BGNRFFdbi
KRMA5dx+J04td1LMRbNCi9YxsoqHZbfHyLxYh1daBmZGJ1e+PrI61W9HAYLRJOLPnQoxEUK
xmzy40ei6EFFgysIBUNTFuRNvbUJ3Ofu+OxQJ+9GAQa07uNnm9hUizIRtvlI12lRYRZWtTe9AW
n6Pyrhsp+/V1mD0ZrLz2EkbfDE9iX/M/X8WSmcXla27Z6liFENuEJNCDTNpY9SPa2UH3N19iLi
pB7+kiuHQRjs8+hep0kX/c/+FfsQTPLruimMyEqteqKqtb/4N1e2m5NxLCLc24VdUvOvJTJ48



GXc4RKhyFXo4TMsbr6KYLRSeehb2sROof/KPBJcvJevgw7Hk5NEzbw7ZhxyB74fvce97IKtDEd
7728ucdPhR6P/+G7bCYjy770PTi0+hAHk/u5CXP3yfqqo1/OQXI1KRkYHV7SGwdBFNryzHEytR
csFI5PzoGFRv/MNupK2FljdepfGlZ7CPKKP00mvw7ronWk837f99h9pH/oBitVH2yyvJ2O9gzE5
Xog/q+v1Q10+eQ3U1NL/+Es2vvYhj9BhKL7ka76zdh+pXJ8Sw41++BN+Cb0FRUD0eYq1NNDz
3ZxSLfwTT6d77ld4Z+9OtLWF1v+8QfkVN9Dx8Xt0ffkp3t32JufQl6m6+xayDz8ae0kZDc8+h
p3UHz+JViy86j54x3Eursp+/V1tLz5d/zfzyPzoMnwz9iV2vtuwzF5KrnHnsrKh+9GdTopOf9SgsE
Azf98hdxDj6L1368Trq8l77hT8M6Ozzzp/OITPLvsRtFPzyNj972JdnbS/t+3qXv0fhSrneKzf07nF
58QWL6Ysl9dR/Nrf8G/aAG5x5xlwalnopjNNP/tJZr//hKOirHxcWlmbkB89k/X1/+j5o93Emmop
/CM88g95iRs0sNdiC2i+X1bVUQMwFpQRHD1SsyZWVjWm21mycwK0ixroIUQOw+5A92PtvffQ
rE7MKIRwlVrWHPnUn7y6++hZoH76D0179D93VR89DdSfvHP/ICq2+4HPWsi6jRdCatWoJz7E
Sq/3Br0nEjr7uNyjuuT6xdzjroMDSfD//iBRSD9XPqM3P47eUXc/Ntd6M+eHv82lfckNQyBsBzza
385rorMHSdO/740GUOOyuvTJ5KNf6R58k9/FgMw6D6/tupfXhdzlrFwrTXPyRUU8myi36W9Lh
Jz7501n7rrb/eAD0apfL239Lw3J8T21S7g2mvf4hrwuTNOocQaTTkd6D1SiTqh+6K37W1WrF
m5bD6+suTjim/+haq7r6R0kuultrWStdXnxFcta5ytX3kaDL3ORhLbi41D9yR9NiR195K5R3XU3z
eJbT885Wk1nOeWbtjycmj/b1/Y87MouDUS6h77H5QVcqvvBEjGqXm4bvjd7R7Ff70PLq++ITg6
hVAVGXehD89T6hqNcsuOSv52tfcSqy7i4a/PJIUDbzonF9ihIM0vvhUYptqdzDtn//FNW4SPd9/y
4KTDoE+RQ7Lr7yREb9IfI2EGOaGbLxZePrRZB1wKJ5pM7f4HG3vvokWDGL2ZhdR6SL/2FMS+
3q+n0fHx+8x9eW3tzipWlcQ2IXegB5kUEetH99yvMLs8oOs0v/5yyv7gyqVYsnOxZGXR/uE7Kfu7
vvqM3GNOxrrwOzItFiJNjXR+9t+U43q+/TpefKxXx0fv4d1tLzRfD4rFwvcl5mNzOPB0dYcm4Zo
whe5vvkg5j/rlp4ybMQtd11IRW0P3nM9Tjml44SkMTSPS1EDDs48m7T0iUcJNjTS88ETK41r//f
f+X6R+RBraXzp6aRteihYpMszT6HEDuTYOUqYh3thBsbUM0WWt54LeUY38LvclWeS/uH7+A
YMz4peQYIVa7GMAqCjv++m/JY/+KF2MtGYXI4kpJngJ55cxJfbMU6O+JTWQF0nWhbK0Ysmp
Q8AZT/7UWYd17X1i5cU0mweg31fb40S8T9w3xUpyOIlZbJbqPpleeStsXHiaXxmBd9n5Q8A9Q
/84jc4RJiCw1GETHbiHCyxfjX7wAe+nlPH3W3HzC0sZSCLEtkQS6HyaXG733uxpzZla/+7WgH
8PQMHSzUvabvRnxfsoeL+FYLN6qqp/jTN4M9N62EACqzY7Ru/5RUVQ8LjdaLlbr235CC/gwe1
LPo2ZmEeZt2epyuZKqY65lzcshVUW1WDD1E4tqMWPNTS0CZhiA5VvFYkmpVA7x6p1CiFSqx
QqGhqlq6NElluzUQllmjxct4l+PIRtoN4cCZm9mymaTxxv3bpeRe/4TINya6w+x6hWG6ipjzF7M
9D8vuTT2Ozx8WX907s9KdX6lX7X3eTypGxX7fEpputXDo9fNxFakt9DkKITdKCARTb1r1/7CP
KCK5aQWD5Yuxll5P2WXLziDQ1JI8nxGeldc+bs1XXFUKI7ZEK0P3I2G1vwjWVqFYrhaedlfTB0u
TxYskvxJyZTbSlhexDjlhvfaeWXvQ/uE7dJVXYM/JI3Ov/cnYYx+UtXd4iH9IdE/ZhXB9TWJb4U
/Ooe0/b+CcmJlwQx3Tps/E7nRRqxIYcvMJrVmFa9K0pLVMitWGb/IMKpcuJiMnl9E50XhmzE7
6EKpYLBSDcR6KomDJyWPkNcITya0lZdjlLRIH0swtSYsw94tjNft1shcUp53ZUjMM1aepmn00In
YljVAXuKTnxjBmPxeUm77hTk9+DThf2kRVEmhvJPvhw2t/7NzmH/zjpHNmHHEnHF98j64BDk
moUmNwe7CMriLa2EFy5HO+ueyY9ruDnyTuWrnzSRUXQnEPwyrduarwfbipKkx5RceBmtb/
8j6dr28IEunf2LpGurThf28lGEqlbjmjlj6Ry2kjlKr74p+XUYNxHXxPg44ZkxC2tBudL+8qtuwtLPI5
lCiE3TA/6U1nADZfJmgAJ6MlglJy9pn2qzY3K6Uma5NL36PatPPYJQbdVWxVslbY3sgZ6A7r
mfglOD0o4SKyjHd+i71EtVmwlpC6urFkZaHhThFbVUIWLkak80Be8oMgrXV9Dhc+KwWRmR
mYvL50clh0GKE1qxEMVtxjB2HJTObwIqlhOtqcl6fTKy7E3Qdc0YmWsCPKSOLDk8Gi+bPY0LZ
SDztzejhMI5xE/EvXojJZMI2cSqff/8dJpOZSZOmKgcYcnJdLcgH/RAhSTiYy99sczY3aiOq4W
CuL7bi5dc7/Aml9lxq574xg9BkPX8S38ju6v/4diNuPdbW/ck6cP6HWL+XvomT+PnrlfYssegXe
3vXCUjx70348Q20BaqnCH29vwL5wPhoahqCgY+BZ+h2K24BhVQXDVCpxjJxLt6iC4ajkZu+9Dt
K0F/5KFOMdNwlpYRPdXn2MfPRZbfiHdc79Etdrw7DlBxe6kZ95XRNvbyNznQMj1NYSqVuOY
MBnD6aJ7zv+wjxqDpagE3zdfYmvJxT1pGtHuLqLVa7AVlhBpbSLa1opr0jQs2bmEa6vi1x4zAef
EKXhznE6MHV1zPkdBwTlhesFVK4h1d5Kx94GEa6oIva3CPX02npm7oZhM9MyfS8+8OdhKSv
HuumfS0BFYtZzur78g2t6Cd9e9cE+fhWkriyAJsZ0ZsvFmzuwKKm67f6u/hFpz+2/BgFG//X3Kv
tU3XcmY0x7Cs8uuA0jhEHP3n46tpBTX+MmMuf3Brbq2EGKryBroQSYJdB/Rzg6CVatRHU6M
QAADI34nXh/A5HZjGPG2VZqvJ94iRtdRHS4MXcOIRIGdTnS/Hz0YwJSVjepwEmttBk1DdXtQV
AWD+De4CgpYlXTGYqhWK+5wFNWkYMRiqHYnuq5jBHwoThftkSihSJgcpXM0Db9iWuVYQWc
bWK14i0cQbWxESvRq/b54qylvBqrVSrS1GdVixZyVDQbEursIRcJEbA7yioqx9v5BDbc0oXV3Yck
rwNLpFO+1QnU1GJEI1uIS+UAr0kILh4k01KJYrNhLSgFz1EOaQPuWLCGIRTDClUyeLPSgD8ViQ
Y+EwWwFtxeluxN0DSxWiMUS446hayiKih40ojicYHfiCwawRcKYLRZUiw0jGkEPBFVfy8jBUE9H
OdrRwCN3jJarrRCMRPHYHtqAfK80BoahoCtiycohGltDehhGNoLrchFSVju5uLGGZfBzy+4iaLeh
uN86YBloUk9uLAcTaWIDMFpwV41C3ctpouKkRzd+DraAYk8s1KK+7GFqxnM4izY2YPV6sUkm
9ryEbb76cXMj4Pz2/1eug6599DFAoPuvClH01D91NwalnkfMSQA0v/4yJS89TckFI7Hiip8z69M
FWDKII7sQaSIJ9CCTNla9uhd+R+s/X8E9YzaxjnZqH76HaGsz3l33luvAHxFYtZzcY06k/omH6fr
fR5izcyg+6xcEq1aTc+hRdPzvlzL33J/qe24muGYlJrETGPXB26l//nE6//sOmfseTNFZP2fNbdCR
WrMCx5jxFJ1xPsFoFJPVhqJfAXjmMULVa3B0nMqlX1x07dv/YHFeES8++SiRUIgfhXcSh5aXY



1YU3unqZva0XVD/+gy5v7wCpyeDznfeoPm1v6CYLRSfczGK3UbN/bdj8mQw5o4/0vnFJzS/8lz
8w+0pP2N5+WgmztodW2c7K6+7IEh9Le6Zu1Jx6/24JyZPu475fbS++Xcq77gezddN7tEnUHb5
9TjKR6XpNyZ2RqGaSqofupuW11/G5HRRfuWN5B33f5g93nSHNiAd//uYcHUIjS8/Q86hR9H93
dd0fvx+YlyJtLdg9mRQ/+TDGNEIOUcej9mbQbhijqwDD8NWUEj1fb/Hv2QhttKjRlR+DvyffsjqI57
G5HRR+NPzseTk0v3NF9hGjsE5eRo1t/+WSFMD7tI7Ej76JK779S8oKhvJz39xKY7XXqDkgkto/f
Ad8o46gXD1amofvR89HCL38GPxzN6DRfX1jBldQcfTjxBcsQT7qApKrrqZxW1tmJ5/jBE/vxz/k
gU0PPc4YFD4k3Mp+sl50EaPGfDro0fCtL/3FqtvpvJoWyuZBx7GqOtuw1kxbvB/GWKb8S9bxKr
rf0PP3C+xFhZT8fsHydrvYJT+1uSLbCLQdfRwOF7XYCvIHHbMBuswmLNzCdetW5LW8emHeH
bZDbM3A8eYCXTp/YqcQ47c6hiEEGJ7IGugAS0YxPft13TP/QojFGLNzVcl1vJ0f/MFHR+/j8ntp
emlZ+j630cAxNrbqL7vNhwjK2h49jFyDvsxq667IOCaQAEVyxIxrU/p+gn5wDgmjCZFZdfQGhN
vP1LcOUyah6+B3NNJRk5uVTfeyuh6jUABJYspOrOG3AcidixPPXgvlb8fXdN45++vMN8fIPT2P5
g1opR77roFddYeNN15I8EIC2I6+VmMWA9Fkt2kXtRdAPFZEJRFxq+/ZqmF59K7Pc9/2c8nR
34a6tYcsGpRHoraPq+/YYVv/k50Y72pNfI9/23rLruUrSeLjAMWt/8O40vPIGxXrVclbYVwzBoevU
FWv72lug6mq+H1TdeQc/8wV/CsS31LJxP+/v/pv6ph1Htdnw/zKfz4/eB3nHlgtxjZ9M7UN3o
wcDGLeyrf96DcVsxvf9XPRwkDW3Xot/yUlgXgl7+WxNysnlTLwudY/dhx4K4p48HYvbzarLzyfS
1ACAb+6XmP/+Fw488ljq1qzijlt/BwcfwcpLrYX/mBMJV6+h+g+3oQf8oGm0vvU6oTUurmaqC/v
eXCK6IV9UPrVIF5dW/xOV0Yhx+LKE1K6h/4iGMSBgjEqHhmUfp+PT9LXqN/IsXsOzSs4m2tQL
Q+dG7VN17K1oouJWvvhgq0a50VI5zCT1zvwTiXRqWXngagRXSIWEoaX4fqtWGom79xz1bUQ
m2wuJ+91lz8wjVxNc6G4ZB95zPE/VPnGPH0/3N1t9fSGE2F5IAg3EwiHa3vkXOT86Kl7sYr1p7
d1f/w/vrrnvQ/uF/Uh6rh0N0ffUZkcY6Yp0dSfuirc3owfgHPsVqjVfm7ru/pQmTx0u0oz2lsm24vo
ZoT1fK9T7+8D0sM3fDsnOZFqsNn8eLc/QY2t/+Z8qx/qU/YB9ZgWfGbLr6aW1IXr6lWHdXSqua
wNlfcDfWJZ+r98N6Xy1vvEK0vTVluxDbQrSjnZZ/vJKyvWf+N2mlZsuF66oxezlrl5Bxm57p4wrt
plyer5L/VKg64tP8MzcHa2zM/Fl21p6MJByfKShjnB9LYZhpLSFCi78jmmT4i2sAj09dCpqpPPn2
+wlvRU45V+fnH2G0t+laOyFpu9bdiTMUJJJaVS8/381le1/buWxt4FTYuuHplyrb2994k2tLcz9Fiex
RprMe33v8TRixGsDL1/y+x7YRqKrEWFm36wK1kyc0n3FssLFxbhR6NYO1Ntp3jJtL99f+2eQxC
CDFUJIEGFFXFOX4ywcRv/baJshYWE2tvS2ndAKCaLViLSjBnZMF63/AqZnOiGrZisaRMfVLM/9
/enUfJVVCJHP/e2rp63zvd6ez7BiGELbKasIQ1oCC4AgMylo6go4wcUEHloKOAoiLoiJkIMogga
FgEZZMoYiAQspGEQMjs3Umnk04nvVVvd935470kvS9Jpate5X7OqdP1ln5171t+/7/2qfu/3Aog
lgZzuj3SRUAaBHh4JNWb8BLsmmnhZBY0NuwmLEKmpInPytO5xV1QSq6sIUINFeFT3ptbx8sp
OPXrv48/N6xZTxrDuJ+Dw+MI2X6lZMv6sLDInTe02Ptylp+hUF8gvRNvb8GfnOMdml9sg2up3E
h7ZPafw6HFEqrify8rusUfdrk00A4VF+PMLemwuGygqpn7PgS/oskJOz98SDPb46Lrw6LHg9xP
dXt15ggialY0/0trjdsjqYXsNRKCwuHsMo8ZYeeMh/pzcHh8DGexh25rDp/XDD7r1an+oNq1fy8L
vfpPvXH0Fv/jmzWx8dzWh0rL9TbgbXv8b2d003t9xad4yTSvW20tSlwxacMqOEaOL5/icxfQ8
PdXyJw4hYLTzzww0e9n+FXxu73w54y66dZ0j5jJO/4jtG7dROW/3kQ8GmH4Vdd3Wu7IL//H/
mcg1r/4HJVf+Eqn6RVXfoGmzZul10+k7BOF7TSt8robaa/eyugJB+75y87L4/x5Z902q47NPj9X
fP4G9PnFhD5yOsUXfaLTxUrGiNEE8gpo211P87o15J98eqfpwcpR1BUUEywfqfzT13T67HG3/
4jwyDGdxuXMOo4ct3dNAF84kzFf+xb+rO6VfGMOB384k5H/djO+rAOVqOzpr5M3+6QkRjV4W
ZOnE6wYwfBrv8TOP/2Ris98vIO5kj31KEklwwh36JXan1dA3vFzaF6/hkBhlaO+elunZVZceR0N
b75+4DMmTXW+wAsGaV6/hqILLj0wswhjq77lk7/7PwAuu/JaAn97icK55xDdVkoGsljMCZMPf
HZuHgWnzKVI6kxiYyd2+zyf/ki71VXI6/8maypMzpdqAdLyii96OMHtY5yps+kc078AyEHAoy74
26CRSUHtTwz9MKVlXn/nXs6fXfcdtlnyJrS/ctec/i0bvqAUNngKtCxWlz177zFspdfYMOK5bQ00
S1cdtRU88i9P+CBO26hqLyc8z53DZXjxPwu7exZMlfiW6vIbqjll0vPUf25On7l+cLhwmpGkvjire
6fVZkWzUR9xYyY4zxCuUf26Xt7exd+TatVVsJIZUR21ZDvLWfYHGJ89iq4jlkneI8726itdvxZ2Uj
GRn4Qhn4MsJEaraQOXYi0doaYjtqyagcSaB0GJEP3qO9sZGMkaPxZWbTtnC3sdptBPLy0XAW
9aaqUIAirS20NzUSKivHI5FBpGorLcWlBn61k2iklcpH5eS2NrM3GKbF56OwaS+BUljQsHJ81VU
ES8uluM9yDY8Zh8bjtLz/HoGcXElJR6MtLTrvqyLW3k5bidPb9vAp09G2KE1rVhGrqyVz9DiyPkzr
scORyPYamteuor2piawJkw/61yVjDkXze2tp3rAWXziT7KkzyCivTNSih6xX3KYP1hOr20Hb7nra
W1vIKC0nUrUZf1YOEGwQb20hNHwk0aotxKMRQqXDiO3aSaCoCMnlhPY22hv30LZrF8HSYfhL
SmhubISakvx+P6GSMjQeJ1pbQ+a4iURVaHvV8M8kYNYY68bNt84eUFJdQ2h4jDPgLCmnB04B
UjCDU3kakajPxIhZC5RW0ZmTy7tbNIA+rYFhbjGj1VgKlZbTkFxLY00Ao0uo8GzbGJ7Jpo9NR4a
Sp5Ew/+qDXUbSulqa1q2lvqCc8bhLZk6cl5D5OM3Ti0SjN69bQ8uH7BitLyZ46g2Ch9cTsGpLyZ
v3XridYWETRvPn9ztvS1MSSp57g1cW/J7+ohNzCQhp311NbVYUIBEIhps4+kVmnfZSMDq1gG
vfs5vIHuS4eBuT5p5D44vPMumeX+LPyto/z7ZHHyRj+AjGfP3b+8c1rV3N6isvAWDGw0+R1e
GLO2NMQlkv3AmWchVoEZkP/ATwAw+o6vf7mv9wPQfaGHPEScpzoI0xR6QhKW9WXHY2xf
MvImf6TAAa9+zhHud/w+qlrxFpaSG/qJjCsnKirS1sXv8uoyZPY/bpZ1JYduBWjng8TiwalZQR3t
8su6v2tjZWP/0Ek9auYHNOATsnTCEUDIM0rIJJs2ZTgrLtoQc49i9LnWXGYrw1bzYIF34c4nHq/
vQHZr+wDAnYw2GMOQysAp1gKVVSiygfuA84C9gKvCEii1V1TXIjM8YYY4zxltbNG/ffWuHcu3



wbY6Z0Y96lNyKcncXe+nqaGholhIKcfP4CMnvoe8Xn83X6xbkn/kCAoxZcRiQUJHP4KApVaYtG
qf7wff7xp8V5eYxv3UvzZs2kjV6Lnt/9xDB4hIKT52LqrJ7yYvsePr3lF18+WFZD8YYk0gpVYEG
TgA2qOoHACLyW2ABYBVoY4wxpgBam9uoq2hgWBRCevfpmH//O7nHHxZYdOmP/PDI5B
Qn7PBEhfN4lJOWyfs78C9m4ZiWbn1/M388+ifC8cznm1T/TePJctj3/DEXIFZSceZfBfnlXJed/DF
+HPiGiO2qpe+YJ8o6fs/9XdGOMSBZUq0BXAls6DG8FTkxSLMYYY4wxntS6aSMZ5cN557VXee
y+eznnk1cyfMy4/v8xwXwijJ9+NO0lpYx4dBHRJS+wPa+A6ppqYps30bCzjrrqKs4vyif69S9ywt2
/wO/3s/O5xbz3jS+Rc9Qstt73l0ou/Dhjb7vL+klwxiRdqlWgB0RErgOuAxp1aSSozHGpDMrb4wx
QyWR5U3jlk3sbGrk2f/+GRdedR0lFQnrcPGg+ldVEP70tYQ2rCX/hFMY1+F+6khLCXuXvkbWU49
x5+InmDRiJMN31qlXflzghCnknnU+9Q//ithXrmXCXf+F330aQ2z3LvYuW0ps5w6n48Kjj7UKtjH
msEupTsREZA5wu6qe4w7fAqCqd/X2P9apjzEmQawTMWPMUEl4ebOnfhfLX32Z7Vs2sfL1v7
Ht+ae40KiA8ILLyejQl3Yqk9ptxJe9TnOsJ3hL0pbmmltaqK1uQmNRpkTDjHc720L+MgDCIF2i
4+oz0dBvJ2gKnWlFCRgJCJUWEw44CcUieCPRRGfH19OLlJUjD+/EH84jAASiUBrM7S1IYEgueU
VFI4eiy8cBIXlLS207d1DvLkJbWtDMjI50bjz85BAGHi0QhtDbuJ7aqjvakRQfDn5hEsKSWQX4g/
KwsJBj2KvUinR7uhCqpoezvaFkNjMee9KuL30f8XCCKBgNOBW9f/H6ih+hwzeG49TONxaG8n
3hZD22LQHne2T8CPBIL4AkHw+xGfj8wJkwnk5A7mU2xjJliqVaADwHpgHIAFvAF8SIVX9/E/O
4BNCQyJBKHL4PKSLZ3ySadclL3ySYdc6lS1z2e9JLI8SYd11pN0zCsd4L0zMsR0SW8vBkfpMj
UhgzfN1zu9/GZnEx8aXbpnNvhF+aIKtEO17G59uuzOQl8uSe67e6drVUDnL0EWNtfeWMGJ6Uq
0AAich7wY5zHWC1U1TuH+PPfVNXjhlzD6d0yiedcoH0yiedchkq6br00jGvdMwJ0jOvdMxpsLy
2DrwUr8V6eHgpVvBWvF6K1UtS7h5oVX0WeDbZcRhjjDHGGGOMMR1ZWxdjjDHGGGOMMMWY
ArALd3f8k04AES6d80ikXSK980imXoZKu6ywd80rHnCA980rHnAbLa+vAS/FarleHl2IFb8XrpVg
9l+XugTbGGGOMMcYYY1KR/QJtjDHGGGOMMcYmGfWgOxCR+SKyTkQ2iMg3kh1Pf0Rkpli8L
CJRGS1iNz0ji8Skb+lyHvu30J3vIJT938VojlscnNoDsR8Yvl2yLytDs8Vkt+6cb8qliE3PEZ7vAGd
/qYpAbeAxEpEJHHRWStiLwrlNm8vm2+4u5nq0TKEREJE3n7JJPXypreiMiHlrJSRJaLyJvuB73
8VQmlgtFpFZEVnUY59ljdZ9e8rpdRKrcbbcbfLfvmm3uHmtE5FzkhN139LxvJdIXipbeio/Uslgy
oVkg+yxnkyDPYZTNNaUW7fuNdISEXnHjfUOd3yP12nm0FgF2iUifuA+4FvgGvBJEZmW3Kj61Q
b8u6pOA04CbnBj/gbw0qpOBF50h8HJbaL7ug64f+hD7teNwLsdhn8A3KuqE4B64Bp3/DVAvTv
+Xne+VPMT4DIVnQLMxMnLk9tGRCqBLwPHqeoMnMfMXYG3t09SeLs6ctHVfWYDo/J6G0fT
2WLgK7PyPTksdrFlrrnBc4xe4z7ehbA3QevAKa7//Nzd19NNel43kslj5YtXcuPVLKlgZcLybalAR
7rKWCwx3Ay9RYrpN66jQBzVXUmcAwwX0ROovfrNHMlrAJ9wAnABIX9QFWjwG+BBUmOqU+
qWqOqb7nv9+JU0Cpx4v6109uvyYvd9wuAB9XxOlAglhVDG3XvRGQEcD7wgDsswFzgcXeWrr
nsy/FxYJ47f0oQkXzgNOBXAKoaVdXdeHTbuAJApogEgCygBo9unyTzXFkzSL3t4ylLVV8FdnU
Z7eVjFeg1r94sAH6rqhFV3QhswNIXU0q6nfcSLN3LliE1yHlHqQZ5rCfVQRzDSdNHrCnHLEma3
cGg+1J6v04zh8AQ0AdUAls6DG8lRQ+SnojTRHYW8E9gmKrWuJ02AcPc96me44+Bm4G401w
M7FbVNne4Y7z7c3GnN7jzp4qxwA7gf8Vpkv6AiGtJ0W2jqlXAj4DNOBXnBmAZ3t0+yZTS23qQ
FPiziCwTkevccB3t417jyWN1gL7kNmde2KGZpOfySpPzXiJ5Ldeeyo9U57XyradiPWUM8BhOCV
1iHRct+LcBrkcqAX+ArXP79dp5hBYBTOniEgO8HvgJlXd03GaOt2sp3xX6yJyAVCrqsuSHUuCB
lBjgtVdRbQRJfmSF7ZNgDuyWEBzhcDw4Fsem4uZo4sp6jqsThNRm8QkdM6TvTSPt6XdMnD
dT8wHqeJXw1wd1KjOUjpcN4zfZcfqc4D+1lKH+teOoZ7iDUI162qtqvqMcAlnBYpU5lBfgyCvQ
BVcDIDsMj3HEpTUSCOAf1w6r6hDt6+74mau7fWnd8Kud4MnCRiHyl0+xsLs49xAVuk2HoHo/
+XNzp+cD0oQy4H1uBraq675vKx3Eq1F7cNgBnAhtVdYeqxoAncLaZV7dPMqX6th4wt2UCqlol
PllzWu5tH/carx6rfVLV7e5FVhz4JQeaaXsmrzQ67yWap3LtpfxldZ4p3/o41pNukMdwUvUUayqv
WwD3lsGXgTn0fp1mDoFVoA94A5jo9lYXwulMZXGSY+qTe0/pr4B3VfWeDpMWA1e6768E/thh
/OfEcRLQ0KG5TFKp6i2qOkJVx+Cs+5dU9dM4BcCl7mxdc9mX46Xu/CnzbaWqbg02iMhkd9Q
8YA0e3DauzcBjPlL7nf78vHk9kkyz5U1PRGRbBHJ3fceOBtYRe/7uNd49VjtU5f7fy/B2Wbg5H
WFOD3oj8XpdGvpUMfXn3Q67x0Gnilb+ig/Up1nyrc+jvWkOohjOGl6izUV162llplgfs+EzgL557t3
q7TzKFQVXu5L+A8YD30PQO3JjueAcR7Ck4TlxAcvd1Hs69pi8C7wEvAEXu/ILTQ+f7wEqcHp
WTnkcPeZ0BPO2+H4dzEbcBeAzlcMeH3eEN7vRxyY67hzyOAd50t88fgElvbxvgDmAtzonilSDD
y9snyevSU2VNLzmMA95xX6v35dHbPp7KL+ARnGZ4MZzWI9d4+VjtJ6+H3LhX4FywVnSY/1Y
3r3XAucmOv5ec0vK8l8D144mypbfyl5VegyKXkv0a7LGe5FgHdQynaKwpt26Bo4G33ZHWAd9y
x/d4nWavQ3uJu3KNMcYYY4wxhjTB2vCbYwxhhjjDHGDIBVol0xxhhjjDHGmAGwCrQxxhhjjD
HGGDMAVoE2xhhjjDHGGGMGwCrQxxhhjjDHGGGPMAFgF2hhjjDHGGGOMGQCvBvPE5FFInJp
/3MaY0xiichVlvKzZMdhjDHGmKFhFWHjjDHGGGOSSESyReQZEXlHrFaJyOUI8qGI3CEib4nlSh
GZ0sf/3y4iC0XkFRH5QES+3GHaV91lrhKRm4YklWPSmFWgjeelyOdEZIV7knnlHX2aiLzmjR6/
TVaRM5wTy6Pi8haEXIYRMSdNk9E3nZPUgtFJGNIEJLGPBQR+fq+i08RuVdEXnLfz3XLjKtFZL2I
LAV07mdZi0Tkp13LJ3H80L2gXSkilx/2xlwxqWw+UK2qM1V1BvCc0750VY8F7ge+1s8ypgDn
ACcA3xaRoljMBq4GTgROAj4vIrMOSwbGHCGsAm08RUSmA7cBc1V1JnCj06kCOAW4Aph+P
4uZBdwETAPGASelSBhYBFyqqkBAeD6RMdvjPGEJcCp7vvjgBwRCbrj1gN34FScT8EP/rTU/



n0MeAYYCZwJvBDEallUPzGG09ZCZwlj8QkVNVtcEd/4T7dxkwpp9IPKQqEVWtA2qBYThlz50
q2qSqje7yTu1rlcaYvlkF2njNXOA9+SAQu5yx/9BVeOqugbnhNGXpaq6VVXjwHKcE9JkYKOqrn
fn+TVwWqKDN8Z4wjJgtojkARHgHzgV6VOBGPCkQu5Q1Sjw6ACW11P5dArwiKq2q+p24K/A8
YIOxBjjDe71x7E4Fenvici33EkR9287zpf7fYl0eD+Q+Y0xB8Eq0CZddDxpyCDmtROMMaYTVY0B
G4GrgNdwpfH+KDABePcgFjmY8skYcwQSkeFAs6r+BvghTmU6EZYAF4tll0hkA5e444wxB8kq0
MZrXgluE5FiABEpStBy1wFjRGSCO/xZnF+EjDFHpiU49xu+6r7/AvA28DpwuogUu826LzuE5V8u
ln4RKcVp8bL00MM2xnjUucBSEVkoFbv4XilWqqpv4dyithT4J/CAqr6diGUbc6SyX96Mp6jqahG
5E/iriLTjXNAmyrmtlnl18JilBIA3gF8kYtnGGE9aAtwK/ENVm0SkFViiqjUicjtOs+7d0LeBHIwngT
nAO4ACN6vqtkMN2hjjTar6PPB8l9FjOkx/Eziji/+vcvwjA7v7wHuSUCYxhhAVDXZMRhjjDHGGG
OMMSnPfoE2xhhjjDHGA9zWcj2Gf13Vb0hGfEYcySyX6BNWhKR04CHuoYQqOqJyYjHGJPeR
ORWut8P/Ziq3pmMelwxhhzeFgF2hhjjDHGGGOMGQDrhdsYY4wxhhjjBkAq0AbY4wxhhjjDE
DYBVoY4wxhhjjDFmAKwCbYwxhhjjDHGDIBVoI0xxhhjjDHGmAH4f2PSeQYekWs1AAAAEIF
TkSuQmCC\n",

```
"text/plain": [  
  "<Figure size 981.79x680.4 with 12 Axes>"  
],  
  "metadata": {  
    "needs_background": "light"  
  },  
  "output_type": "display_data"  
},  
  "source": [  
    "fig = sns.pairplot(data = data, hue = \"target\", palette = [\"#C4210E\", \"#410F01\"],  
height = 3.15, aspect = 1.35)\n",  
    "plt.show(fig);"  
  ],  
},  
{  
  "cell_type": "code",  
  "execution_count": 15,  
  "id": "c39a74f9",  
  "metadata": {  
    "execution": {  
      "iopub.execute_input": "2022-10-25T06:57:30.794299Z",  
      "iopub.status.busy": "2022-10-25T06:57:30.793865Z",  
      "iopub.status.idle": "2022-10-25T06:57:30.826635Z",  
      "shell.execute_reply": "2022-10-25T06:57:30.825111Z"  
    },  
    "papermill": {  
      "duration": 0.050621,  
      "end_time": "2022-10-25T06:57:30.829424",  
      "exception": false,  
      "start_time": "2022-10-25T06:57:30.778803",  
      "status": "completed"  
    },  
    "tags": []  
  },  
  "outputs": [  
    {  
      "name": "stdout",  
      "output_type": "stream",  
      "text": [  
        "\u001b[2m\u001b[1m\u001b[32m\n",  
        "UPPERCASE LETTERS WERE SUCCESFULLY CONVERTED INTO LOWERCASE  
LETTERS...\u001b[0m\n"
```



```

    ]
  }
],
"source": [
  "data[\"text\"] = data[\"text\"].apply(lambda x: \" \".join(x.lower() for x in x.split()))\n",
  "\n",
  "print(colored(\"\\nUPPERCASE LETTERS WERE SUCCESFULLY CONVERTED INTO\n",
  LOWERCASE LETTERS...\", color = \"green\", attrs = [\"bold\", \"dark\"]))"
]
},
{
  "cell_type": "code",
  "execution_count": 16,
  "id": "d50f380d",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:30.858992Z",
      "iopub.status.busy": "2022-10-25T06:57:30.858616Z",
      "iopub.status.idle": "2022-10-25T06:57:30.884071Z",
      "shell.execute_reply": "2022-10-25T06:57:30.882326Z"
    },
    "papermill": {
      "duration": 0.043495,
      "end_time": "2022-10-25T06:57:30.886886",
      "exception": false,
      "start_time": "2022-10-25T06:57:30.843391",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "\u001b[2m\u001b[1m\u001b[32m\n",
        "PUNCTUATION MARKS WERE SUCCESFULLY DELETED...\u001b[0m\n"
      ]
    }
  ],
  "source": [
    "data[\"text\"] = data[\"text\"].str.replace('[^\\w\\s]', '')\n",
    "\n",
    "print(colored(\"\\nPUNCTUATION MARKS WERE SUCCESFULLY DELETED...\", color =\n",
    \"green\", attrs = [\"bold\", \"dark\"]))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 17,
  "id": "ff858dff",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:30.917743Z",
      "iopub.status.busy": "2022-10-25T06:57:30.916441Z",
      "iopub.status.idle": "2022-10-25T06:57:30.933545Z",
      "shell.execute_reply": "2022-10-25T06:57:30.932257Z"
    }
  }
}

```




```

    },
    "papermill": {
      "duration": 0.034753,
      "end_time": "2022-10-25T06:57:30.935854",
      "exception": false,
      "start_time": "2022-10-25T06:57:30.901101",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "\u001b[2m\u001b[1m\u001b[32m\n",
        "NUMBERS WERE SUCCESFULLY DELETED...\u001b[0m\n"
      ]
    }
  ],
  "source": [
    "data[\text\"] = data[\text\].str.replace(\"\\d\",\"\n",
    "\n",
    "print(colored(\"\\nNUMBERS WERE SUCCESFULLY DELETED...\", color = \"green\", attrs
= [\"bold\", \"dark\"]))"
  ],
  {
    "cell_type": "code",
    "execution_count": 18,
    "id": "36b313c5",
    "metadata": {
      "execution": {
        "iopub.execute_input": "2022-10-25T06:57:30.965857Z",
        "iopub.status.busy": "2022-10-25T06:57:30.965012Z",
        "iopub.status.idle": "2022-10-25T06:57:31.000242Z",
        "shell.execute_reply": "2022-10-25T06:57:30.998611Z"
      }
    },
    "papermill": {
      "duration": 0.05329,
      "end_time": "2022-10-25T06:57:31.003171",
      "exception": false,
      "start_time": "2022-10-25T06:57:30.949881",
      "status": "completed"
    },
    "tags": []
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "\u001b[2m\u001b[1m\u001b[32m\n",
        "STOPWORDS AND PUNCTUATION WERE SUCCESFULLY DELETED ...\u001b[0m\n"
      ]
    }
  ],
],

```



```

"source": [
  "stop_words = set(stopwords.words(\"english\"))\n",
  "punctuation = list(string.punctuation)\n",
  "stop_words.update(punctuation)\n",
  "\n",
  "data[\"text\"] = data[\"text\"].apply(lambda x: \" \".join(x for x in x.split() if x not in
stop_words))\n",
  "\n",
  "print(colored(\"\\nSTOPWORDS AND PUNCTUATION WERE SUCCESFULLY DELETED
...\", color = \"green\", attrs = [\"bold\", \"dark\"]))"
]
},
{
  "cell_type": "code",
  "execution_count": 19,
  "id": "826a9208",
  "metadata": {
    "execution": {
      "iopub.execute_input": "2022-10-25T06:57:31.033570Z",
      "iopub.status.busy": "2022-10-25T06:57:31.033099Z",
      "iopub.status.idle": "2022-10-25T06:57:31.588999Z",
      "shell.execute_reply": "2022-10-25T06:57:31.586788Z"
    },
    "papermill": {
      "duration": 0.57384,
      "end_time": "2022-10-25T06:57:31.591369",
      "exception": true,
      "start_time": "2022-10-25T06:57:31.017529",
      "status": "failed"
    }
  },
  "tags": [],
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "\n",
      "*****\n",
      "  Resource \u001b[93mmomw-1.4\u001b[0m not found.\n",
      "  Please use the NLTK Downloader to obtain the resource:\n",
      "\n",
      "  \u001b[31m>>> import nltk\n",
      "  >>> nltk.download('omw-1.4')\n",
      "  \u001b[0m\n",
      "  For more information see: https://www.nltk.org/data.html\n",
      "\n",
      "  Attempted to load \u001b[93mcorpora/omw-1.4\u001b[0m\n",
      "\n",
      "  Searched in:\n",
      "    - '/root/nltk_data'\n",
      "    - '/opt/conda/nltk_data'\n",
      "    - '/opt/conda/share/nltk_data'\n",
      "    - '/opt/conda/lib/nltk_data'\n",
      "    - '/usr/share/nltk_data'\n",
      "    - '/usr/local/share/nltk_data'\n",
      "    - '/usr/lib/nltk_data'
    ]
  }
]

```



[illegible]

[illegible]


```

{
  "cell_type": "code",
  "execution_count": null,
  "id": "68e18727",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.857891Z",
      "iopub.status.idle": "2022-10-25T06:53:19.858325Z",
      "shell.execute_reply": "2022-10-25T06:53:19.858150Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.858131Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "data[\"text\"] = data[\"text\"].apply(lambda x: \" \".join(re.sub(r'http\\S+', '', x) for x in
x.split()))\\n",
    "\\n",
    "print(colored(\"\\nURLs WERE SUCCESSFULLY REMOVED...\\n\", color = \"green\\n\", attrs =
[\"bold\\n\", \"dark\\n\"])"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "f863448d",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.860536Z",
      "iopub.status.idle": "2022-10-25T06:53:19.861125Z",
      "shell.execute_reply": "2022-10-25T06:53:19.860844Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.860817Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "data[\"text\"] = data[\"text\"].apply(lambda x: \" \".join([x for x in x.split() if len(x) > 3]))\\n",
    "\\n",
    "print(colored(\"\\nWORDS LESS THAN 3 LETTERS LONG WERE SUCCESSFULLY
REMOVED...\\n\", color = \"green\\n\", attrs = [\"bold\\n\", \"dark\\n\"])"
  ]
},

```



```

{
  "cell_type": "code",
  "execution_count": null,
  "id": "7ccb8a3a",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.862488Z",
      "iopub.status.idle": "2022-10-25T06:53:19.863339Z",
      "shell.execute_reply": "2022-10-25T06:53:19.863073Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.863045Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "data.head(n = 10)"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "32aeb4be",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.864859Z",
      "iopub.status.idle": "2022-10-25T06:53:19.865418Z",
      "shell.execute_reply": "2022-10-25T06:53:19.865162Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.865137Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "corpus = []\n",
    "for i in data.text:\n",
    "    for j in i.split():\n",
    "        corpus.append(j.strip())"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "7edc988c",

```



```

"metadata": {
  "execution": {
    "iopub.status.busy": "2022-10-25T06:53:19.867418Z",
    "iopub.status.idle": "2022-10-25T06:53:19.867978Z",
    "shell.execute_reply": "2022-10-25T06:53:19.867705Z",
    "shell.execute_reply.started": "2022-10-25T06:53:19.867680Z"
  },
  "papermill": {
    "duration": null,
    "end_time": null,
    "exception": null,
    "start_time": null,
    "status": "pending"
  },
  "tags": []
},
"outputs": [],
"source": [
  "counter = Counter(corpus)\n",
  "common_words = counter.most_common(15)\n",
  "dict(common_words)"
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "5c9eef54",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.869738Z",
      "iopub.status.idle": "2022-10-25T06:53:19.870348Z",
      "shell.execute_reply": "2022-10-25T06:53:19.870090Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.870062Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "text = \"\".join(i for i in data.text)\n",
    "\n",
    "wc = WordCloud(background_color = \"white\", width = 1200, height = 600,\n",
    "                contour_width = 0, contour_color = \"#410F01\", max_words = 1000,\n",
    "                scale = 1, collocations = False, repeat = True, min_font_size = 1)\n",
    "\n",
    "wc.generate(text)\n",
    "\n",
    "plt.figure(figsize = [15, 7])\n",
    "plt.imshow(wc)\n",
    "plt.axis(\"off\")\n",
    "plt.show"
  ]
}

```



```

    ],
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "id": "e87641f9",
      "metadata": {
        "execution": {
          "iopub.status.busy": "2022-10-25T06:53:19.871901Z",
          "iopub.status.idle": "2022-10-25T06:53:19.872449Z",
          "shell.execute_reply": "2022-10-25T06:53:19.872203Z",
          "shell.execute_reply.started": "2022-10-25T06:53:19.872177Z"
        },
        "papermill": {
          "duration": null,
          "end_time": null,
          "exception": null,
          "start_time": null,
          "status": "pending"
        },
        "tags": []
      },
      "outputs": [],
      "source": [
        "fig,ax = plt.subplots(figsize = (15, 8))\n",
        "text_words = data[data[\"target\"] == \"spam\"]['text'].str.split().apply(lambda x : [len(i)
for i in x])\n",
        "sns.distplot(text_words.map(lambda x: np.mean(x)), color = \"#410F01\", ax =
ax).set_title(\"Distribution of average word length in texts where target is 'spam'\");"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "id": "75067b3e",
      "metadata": {
        "execution": {
          "iopub.status.busy": "2022-10-25T06:53:19.874149Z",
          "iopub.status.idle": "2022-10-25T06:53:19.874938Z",
          "shell.execute_reply": "2022-10-25T06:53:19.874653Z",
          "shell.execute_reply.started": "2022-10-25T06:53:19.874626Z"
        },
        "papermill": {
          "duration": null,
          "end_time": null,
          "exception": null,
          "start_time": null,
          "status": "pending"
        },
        "tags": []
      },
      "outputs": [],
      "source": [
        "fig,ax = plt.subplots(figsize = (15, 8))\n",
        "text_words = data[data[\"target\"] == \"ham\"]['text'].str.split().apply(lambda x : [len(i)
for i in x])\n",
        "sns.distplot(text_words.map(lambda x: np.mean(x)), color = \"#410F01\", ax =

```




```

ax).set_title("Distribution of average word length in texts where target is 'ham'");
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "6026593a",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.876816Z",
      "iopub.status.idle": "2022-10-25T06:53:19.877610Z",
      "shell.execute_reply": "2022-10-25T06:53:19.877404Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.877365Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "lb = LabelEncoder()\n",
    "data[\"target\"] = lb.fit_transform(data[\"target\"])"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "f0774d71",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.879023Z",
      "iopub.status.idle": "2022-10-25T06:53:19.879970Z",
      "shell.execute_reply": "2022-10-25T06:53:19.879757Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.879735Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "x = data[\"text\"]\n",
    "y = data[\"target\"]\n",
    "\n",
    "train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.20, shuffle = True,
    random_state = 11)\n",
    "\n"
  ]
}

```



```

        "print(colored("\n\nDATASET WAS SUCCESSFULLY DIVIDED ...", color = "green", attrs =
["bold", "dark"]))"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "c2027779",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.881248Z",
            "iopub.status.idle": "2022-10-25T06:53:19.882102Z",
            "shell.execute_reply": "2022-10-25T06:53:19.881871Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.881850Z"
        },
        "papermill": {
            "duration": null,
            "end_time": null,
            "exception": null,
            "start_time": null,
            "status": "pending"
        },
        "tags": []
    },
    "outputs": [],
    "source": [
        "print('The shape of 'train_x' is {} and the shape of 'test_x' is {}'.format(train_x.shape[0],
test_x.shape[0]))"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "5c977308",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.883582Z",
            "iopub.status.idle": "2022-10-25T06:53:19.884013Z",
            "shell.execute_reply": "2022-10-25T06:53:19.883809Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.883790Z"
        },
        "papermill": {
            "duration": null,
            "end_time": null,
            "exception": null,
            "start_time": null,
            "status": "pending"
        },
        "tags": []
    },
    "outputs": [],
    "source": [
        "tokenizer = Tokenizer(num_words = None)\n",
        "tokenizer.fit_on_texts(train_x)\n",
        "\n",
        "tokenized_train = tokenizer.texts_to_sequences(train_x)\n",
        "tokenized_test = tokenizer.texts_to_sequences(test_x)\n",

```



```

        "\n",
        "train_x = sequence.pad_sequences(tokenized_train, maxlen = None)\n",
        "test_x = sequence.pad_sequences(tokenized_test, maxlen = None)"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "1ce9934b",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.885739Z",
            "iopub.status.idle": "2022-10-25T06:53:19.886731Z",
            "shell.execute_reply": "2022-10-25T06:53:19.886511Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.886475Z"
        },
        "papermill": {
            "duration": null,
            "end_time": null,
            "exception": null,
            "start_time": null,
            "status": "pending"
        },
        "tags": []
    },
    "outputs": [],
    "source": [
        "GLOVE_EMBEDDING = \"../input/glove-twitter/glove.twitter.27B.100d.txt\""
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "43fa041f",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.888408Z",
            "iopub.status.idle": "2022-10-25T06:53:19.889225Z",
            "shell.execute_reply": "2022-10-25T06:53:19.889034Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.889013Z"
        },
        "papermill": {
            "duration": null,
            "end_time": null,
            "exception": null,
            "start_time": null,
            "status": "pending"
        },
        "tags": []
    },
    "outputs": [],
    "source": [
        "def get_coefs(word, *arr):\n",
        "    return word, np.asarray(arr, dtype = \"float32\")\n",
        "embeddings_index = dict(get_coefs(*g.rstrip().rsplit(\"    \"))    for    g    in
open(GLOVE_EMBEDDING))\n",
        "\n",

```



```

"#
#\n",
    "\n",

"#
#\n",
    "\n",
    "embeddings = np.stack(embeddings_index.values())\n",
    "embedding_mean, embedding_std = embeddings.mean(), embeddings.std()\n",
    "embedding_size = embeddings.shape[1]\n",
    "\n",
    "word_index = tokenizer.word_index\n",
    "nb_words = min(6012, len(word_index))+1\n",
    "\n",
    "embedding_matrix = embedding_matrix = np.random.normal(embedding_mean,
embedding_std, (nb_words, embedding_size))\n",
    "for word, i in word_index.items():\n",
    "    if i >= 6012:\n",
    "        continue\n",
    "    embedding_vector = embeddings_index.get(word)\n",
    "    if embedding_vector is not None:\n",
    "        embedding_matrix[i] = embedding_vector"
]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "1992a040",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.890281Z",
            "iopub.status.idle": "2022-10-25T06:53:19.891105Z",
            "shell.execute_reply": "2022-10-25T06:53:19.890869Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.890848Z"
        },
        "papermill": {
            "duration": null,
            "end_time": null,
            "exception": null,
            "start_time": null,
            "status": "pending"
        },
        "tags": []
    },
    "outputs": [],
    "source": [
        "lr_reduce = ReduceLROnPlateau(monitor = \"val_accuracy\", patience = 2, factor = 0.5,
min_lr = 0.00001)"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "f95b3782",
    "metadata": {
        "execution": {

```




```

        "iopub.status.busy": "2022-10-25T06:53:19.892486Z",
        "iopub.status.idle": "2022-10-25T06:53:19.893291Z",
        "shell.execute_reply": "2022-10-25T06:53:19.893103Z",
        "shell.execute_reply.started": "2022-10-25T06:53:19.893082Z"
    },
    "papermill": {
        "duration": null,
        "end_time": null,
        "exception": null,
        "start_time": null,
        "status": "pending"
    },
    "tags": []
},
"outputs": [],
"source": [
    "model = Sequential()\n",
    "\n",
    "model.add(Embedding(6013, output_dim = 100, weights = [embedding_matrix], trainable\n",
    "= False))\n",
    "\n",
    "model.add(LSTM(units = 128, return_sequences = True, recurrent_dropout = 0.3, dropout\n",
    "= 0.5))\n",
    "\n",
    "model.add(LSTM(units = 64, recurrent_dropout = 0.3, dropout = 0.5))\n",
    "\n",
    "model.add(Dense(units = 32, activation = \"relu\"))\n",
    "\n",
    "model.add(Dense(1, activation = \"sigmoid\"))\n",
    "\n",
    "model.compile(optimizer = tf.keras.optimizers.Adam(lr = 0.01), loss =\n",
    "\"binary_crossentropy\", metrics = [\"accuracy\"])"
]
},
{
    "cell_type": "code",
    "execution_count": null,
    "id": "860fc8c2",
    "metadata": {
        "execution": {
            "iopub.status.busy": "2022-10-25T06:53:19.894779Z",
            "iopub.status.idle": "2022-10-25T06:53:19.895750Z",
            "shell.execute_reply": "2022-10-25T06:53:19.895560Z",
            "shell.execute_reply.started": "2022-10-25T06:53:19.895539Z"
        }
    },
    "papermill": {
        "duration": null,
        "end_time": null,
        "exception": null,
        "start_time": null,
        "status": "pending"
    },
    "tags": []
},
"outputs": [],
"source": [
    "model.summary()"
]

```



```

]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "3ec12f81",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.897009Z",
      "iopub.status.idle": "2022-10-25T06:53:19.897804Z",
      "shell.execute_reply": "2022-10-25T06:53:19.897607Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.897587Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "history = model.fit(train_x,\n",
    "                      train_y,\n",
    "                      batch_size = 64,\n",
    "                      validation_data = (test_x, test_y),\n",
    "                      epochs = 20,\n",
    "                      callbacks = [lr_reduce])"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "a2df88c1",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.899248Z",
      "iopub.status.idle": "2022-10-25T06:53:19.900238Z",
      "shell.execute_reply": "2022-10-25T06:53:19.900034Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.900007Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "print(\"Model accuracy on the train set: \", model.evaluate(train_x, train_y)[1])\n",
    "print(\"Model accuracy on the test set: \", model.evaluate(test_x, test_y)[1])"
  ]
}

```



```

},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "d3e851d5",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.901486Z",
      "iopub.status.idle": "2022-10-25T06:53:19.903072Z",
      "shell.execute_reply": "2022-10-25T06:53:19.902837Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.902816Z"
    }
  },
  "papermill": {
    "duration": null,
    "end_time": null,
    "exception": null,
    "start_time": null,
    "status": "pending"
  },
  "tags": []
},
"outputs": [],
"source": [
  "epochs = [i for i in range(20)]\n",
  "\n",
  "fig, ax = plt.subplots(1, 2)\n",
  "train_acc = history.history[\"accuracy\"]\n",
  "train_loss = history.history[\"loss\"]\n",
  "val_acc = history.history[\"val_accuracy\"]\n",
  "val_loss = history.history[\"val_loss\"]\n",
  "fig.set_size_inches(20, 10)\n",
  "\n",
  "ax[0].plot(epochs, train_acc, \"go-\", label = \"Train accuracy\")\n",
  "ax[0].plot(epochs, val_acc, \"ro-\", label = \"Test accuracy\")\n",
  "ax[0].set_title(\"Train and test accuracy\")\n",
  "ax[0].legend()\n",
  "ax[0].set_xlabel(\"Epochs\")\n",
  "ax[0].set_ylabel(\"Accuracy\")\n",
  "\n",
  "ax[1].plot(epochs, train_loss, \"go-\", label = \"Train loss\")\n",
  "ax[1].plot(epochs, val_loss, \"ro-\", label = \"Test loss\")\n",
  "ax[1].set_title(\"Train and test loss\")\n",
  "ax[1].legend()\n",
  "ax[1].set_xlabel(\"Epochs\")\n",
  "ax[1].set_ylabel(\"Loss\")\n",
  "plt.show()
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "d71d3b3f",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.904453Z",
      "iopub.status.idle": "2022-10-25T06:53:19.904846Z",
      "shell.execute_reply": "2022-10-25T06:53:19.904677Z",

```



```

    "shell.execute_reply.started": "2022-10-25T06:53:19.904658Z"
  },
  "papermill": {
    "duration": null,
    "end_time": null,
    "exception": null,
    "start_time": null,
    "status": "pending"
  },
  "tags": []
},
"outputs": [],
"source": [
  "prediction = model.predict(test_x)\n",
  "classes_pred = np.argmax(prediction, axis = 1)\n",
  "\n",
  "print(classification_report(test_y, classes_pred))"
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "41dd7b09",
  "metadata": {
    "execution": {
      "iopub.status.busy": "2022-10-25T06:53:19.906337Z",
      "iopub.status.idle": "2022-10-25T06:53:19.906736Z",
      "shell.execute_reply": "2022-10-25T06:53:19.906555Z",
      "shell.execute_reply.started": "2022-10-25T06:53:19.906536Z"
    },
    "papermill": {
      "duration": null,
      "end_time": null,
      "exception": null,
      "start_time": null,
      "status": "pending"
    },
    "tags": []
  },
  "outputs": [],
  "source": [
    "conf_mat = confusion_matrix(test_y, classes_pred)\n",
    "print(conf_mat)\n",
    "\n",
    "sns.heatmap(conf_mat, square = True, annot = True, robust = True)\n",
    "plt.show()"
  ]
}
],
"metadata": {
  "kernelspec": {
    "display_name": "Python 3",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {

```




```
    "name": "ipython",
    "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.7.12"
},
"papermill": {
  "default_parameters": {},
  "duration": 89.243179,
  "end_time": "2022-10-25T06:57:34.236039",
  "environment_variables": {},
  "exception": true,
  "input_path": "__notebook__.ipynb",
  "output_path": "__notebook__.ipynb",
  "parameters": {},
  "start_time": "2022-10-25T06:56:04.992860",
  "version": "2.3.4"
}
},
"nbformat": 4,
"nbformat_minor": 5
}
```

