```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {},
   "outputs": [],
   "source": [
    "#import required libraries\n",
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [],
   "source": [
    "#read dataset\n",
    "dataset=pd.read_csv('50_Startups.csv')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style>\n",
       "    .dataframe thead tr:only-child th {\n",
       "        text-align: right;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: left;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>R&amp;D Spend</th>\n",
       "      <th>Administration</th>\n",
       "      <th>Marketing Spend</th>\n",
       "      <th>State</th>\n",
       "      <th>Profit</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
       "      <td>165349.20</td>\n",
       "      <td>136897.80</td>\n",
       "      <td>471784.10</td>\n",
       "      <td>New York</td>\n",
       "      <td>192261.83</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>1</th>\n",
       "      <td>162597.70</td>\n",
       "      <td>151377.59</td>\n",
       "      <td>443898.53</td>\n",
       "      <td>California</td>\n",
       "      <td>191792.06</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>2</th>\n",
       "      <td>153441.51</td>\n",
       "      <td>101145.55</td>\n",
       "      <td>407934.54</td>\n",
       "      <td>Florida</td>\n",
       "      <td>191050.39</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>3</th>\n",
       "      <td>144372.41</td>\n",
       "      <td>118671.85</td>\n",
       "      <td>383199.62</td>\n",
       "      <td>New York</td>\n",
       "      <td>182901.99</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>4</th>\n",
       "      <td>142107.34</td>\n",
       "      <td>91391.77</td>\n",
       "      <td>366168.42</td>\n",
       "      <td>Florida</td>\n",
       "      <td>166187.94</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "   R&D Spend  Administration  Marketing Spend       State     
Profit\n",
       "0  165349.20       136897.80        471784.10    New York  
192261.83\n",
       "1  162597.70       151377.59        443898.53  California  
191792.06\n",
       "2  153441.51       101145.55        407934.54     Florida  
191050.39\n",
       "3  144372.41       118671.85        383199.62    New York  
182901.99\n",
       "4  142107.34        91391.77        366168.42     Florida  166187.94"
      ]
     },
     "execution_count": 3,
     "metadata": {},
     "output_type": "execute_result"
```

```
    }
   ],
   "source": [
    "dataset.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style>\n",
       "    .dataframe thead tr:only-child th {\n",
       "        text-align: right;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: left;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
       "      <th></th>\n",
       "      <th>R&amp;D Spend</th>\n",
       "      <th>Administration</th>\n",
       "      <th>Marketing Spend</th>\n",
       "      <th>Profit</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>count</th>\n",
       "      <td>50.000000</td>\n",
       "      <td>50.000000</td>\n",
       "      <td>50.000000</td>\n",
       "      <td>50.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>mean</th>\n",
       "      <td>73721.615600</td>\n",
       "      <td>121344.639600</td>\n",
       "      <td>211025.097800</td>\n",
       "      <td>112012.639200</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>std</th>\n",
       "      <td>45902.256482</td>\n",
       "      <td>28017.802755</td>\n",
       "      <td>122290.310726</td>\n",
       "      <td>40306.180338</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>min</th>\n",
       "      <td>0.000000</td>\n",
       "      <td>51283.140000</td>\n",
       "      <td>0.000000</td>\n",
       "      <td>14681.400000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>25%</th>\n",
       "      <td>39936.370000</td>\n",
       "      <td>103730.875000</td>\n",
       "      <td>129300.132500</td>\n",
       "      <td>90138.902500</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>50%</th>\n",
       "      <td>73051.080000</td>\n",
       "      <td>122699.795000</td>\n",
       "      <td>212716.240000</td>\n",
       "      <td>107978.190000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>75%</th>\n",
       "      <td>101602.800000</td>\n",
       "      <td>144842.180000</td>\n",
       "      <td>299469.085000</td>\n",
       "      <td>139765.977500</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>max</th>\n",
       "      <td>165349.200000</td>\n",
       "      <td>182645.560000</td>\n",
       "      <td>471784.100000</td>\n",
       "      <td>192261.830000</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "           R&D Spend  Administration  Marketing Spend
Profit\n",
       "count      50.000000       50.000000        50.000000
50.000000\n",
       "mean    73721.615600   121344.639600    211025.097800
112012.639200\n",
       "std     45902.256482    28017.802755    122290.310726
40306.180338\n",
       "min         0.000000    51283.140000         0.000000
14681.400000\n",
       "25%     39936.370000   103730.875000    129300.132500
90138.902500\n",
       "50%     73051.080000   122699.795000    212716.240000
107978.190000\n",
       "75%    101602.800000   144842.180000    299469.085000
139765.977500\n",
       "max    165349.200000   182645.560000    471784.100000  192261.830000"
      ]
     },
     "execution_count": 4,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#to see the statistics\n",
    "dataset.describe()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "R&D Spend          float64\n",
       "Administration     float64\n",
       "Marketing Spend    float64\n",
       "State               object\n",
       "Profit             float64\n",
       "dtype: object"
      ]
     },
     "execution_count": 7,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#check the dataset features datatypes\n",
    "dataset.dtypes # can see that state is categorical data and other are 
continuous\n",
    "# we need to encode the state to numbers since it is categorical data"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "R&D Spend          0\n",
       "Administration     0\n",
       "Marketing Spend    0\n",
       "State              0\n",
       "Profit             0\n",
       "dtype: int64"
      ]
     },
     "execution_count": 11,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "dataset.isnull().sum() # to check if there are any missing values"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAuIAAALICAYAAAA61JVvAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
AALEgAACxIB0t1+/AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uIDIuMS4xLCBodHR
wOi8vbWF0cGxvdGxpYi5vcmcvAOZPmwAAIABJREFUeJzs3X+cnGV97//XZ2Z/JLtJKhsTTuSHgQZyv
oIxwKJST62IkCA2UCnVnNrEaom2cBLLF06gco45trRQKC20HBSUmtieRTFY0wICUqweG1OWEhdoD2S
NASIcEnZRsjtkf811/ph7ltnN/Ljnxz33j3k/H4957O6OauT/XNdd93df1uc05h4iIiIiINFcgq7
AKIiIiIiLQidcRFREREREKgjriIiIiISAjUERcRERERCYE64iIiIiIiIVBHXEREREQkBOqIi4iIi4iIi
EQB1xEREREZEQqCMuIiIiIhICdcQ9q1evdoBuulWqUIW+gUq7r5vvIVOsaqbz1vDqSPueeWWV
8IugogvilWJC8WqxIVIViVcKijriIiIiISAjUERcRERERCYE64iIiIiIiVBHXEREREQkBOqIi4iIi4iIi
EQB1xEWl52axjZGySrPN+ZgPJUiXSElSfJC6iEKttTX9FEZEIyWYdQ6PjbOx7sf2DXPm0h5uaC
7s7SKUs7OKrqk8+LjIxIrqk8RFVGJVI+Ii0tIyE1Ns7HumkwLLsmWf+6Hy2rDmFvl3PNT1WNSIuqX
m2Lx+LuhmM7vXEr3LB3j3ZGH0dUcmJ2ZwYeQ6S+xoF1YVMQPpCpFpzo4x3V2TzRcGa79PjCWYmugEh6y
DO5tXGbeZyZ0YXmpmYkT4lTbHqI/SSabkwIg/R0gTkyOhrVhk0GM6hqswVTMlMc/zDQOzH2v8Wz8P9C3Kk
nNYr3OpgiWWtDbn3/YjL/UVC9wcmIuNgFGfp8w5ZejgLO8eCHRPfK/QlGxdtdZ2XGq4Y6lc6TMHtNN
X/c7Tb0FmqSOrnQX7qjYWQDR4QLaI6ihdWRMnH3rPTxV3awe64T4nJRT6vKPSR/
fwvGLxrFHddaOsVfpDPfXUjTBtfWyCsrXjGlTyzPL2efcXoy1+v9NwvLXwPL4uZxLS8j6vmbN7
zP/s/Cz4JeE4rNOzGfnlRt0blVBiPALENK3arQWx1ZJQ2Ka2oMLbDa6BK+d60REA1Fhcl9vD0/+5
91UDNUvm4pyf/C+gm7cX1pJmv2+A9ABiWlTLgqkkYjmVtTK2sGt+1nGU8WS+EDc1u1FZVXYy9MmexT
K/OUDAcmg8GhDZXzbBhedKtuXn7YPkD3K6B7kDV0/Wh+M/M3rN66+q8ZHk7/TZLpRxPq8T/uY5+M
O3AsK8aRgvrn6XJNzbGMNPxQRdE9AC3r9Dis9iZ/RSIXYmxe3QoH7SmU8pfttLvK2X1GoR7dVWM3g3b
vmbi0FJmaFcuHeJqJWCTaAUPYjrZwEO/wlfqiOk1hjQmMJuABWVfi9YrARh+HWRXHfMHuUORhI3Dy
HZpSvK0XoPdB9dBmXnrWqjB+A/ie9hFsbGnUVsPJ7uNtyu9hCIwnp7O5TUJr9/o4/uWWEOFq29dyMRtN
04cOMqbaUmp0PxVkEjqpFM5IZBRrEd/o38h7o8KjrzkWh5w3fg/lxdLvhbxXbATZodrpGDGPK6
g+n1xzZ3J9QJ9uT82vTYGiHPf8yvKYSwgjk7X69aTn9CAdtYH+Oe+P5xAgtlEWCmTw57KQ1qXt0M1JhXS
jxgA3+1eCXVBtSMFdRhZDT7b9r4d5EhHs7R6bBhPgdIOkAfXA8AWu7GFRFTlwCqrt5UqJFJ5f8cj7Zph0u3
iDfDbBvdqiLdyUD7xoHlJu7k1IbUpHChWT3dvpOGtTW1NzkDpJNBYqiKsEFpA17TqMGYvZ5jjaw4Q/SHp
9NOubkITaYJeyRRwCVX3sN+6BLoHn5UsP/RYy7qRw0bUo/G7HqgB9VRdPTu4tmKCHmlkLz6xBMMWNOiw
p1ODFyg3OLcZSZZLLyZs1SKtw4Gpf2ueuXJXLNzc5Ejn2gc9URqe5/s3pc02Ib2eJJOgK5gvIYsggHxgg
qjOyR0H6gaoGaqwu07uDLNCWn9z5pE/9p81ZPDjxCNDiuqc2rgeNHY4tvRRu6LIqEZlqpW9+NGy4Bi
uZeRUA0Uwmk3hlw1eBfpsnXh/gqqZZYHW8ufrlnjdI36uUrJIgj8YAg7YgPdDfZ1T0J8qYzrB9M7FB3
hgHlCj+kRcYvDkL2PCUxIvjKO6pnQVrIaOfjaibWG4P6tJ7hvxrHoqFIpSAZCN0mVKq3nyC2j/VVRabHM
MOe4ofR9ZB58k92nJOEl4MG/4y88ElPG5z80fbZ3NsmL7dN61R0rQnNF0blVR7m08+RSC4MJRDWnBLz
TUTq3IcmYTD4MBbIRHH/1+KBrnDgC9pvN5tQoYN5kjLqTNMT9vA8Xzfp6xdJpOCE15Vt47P3XqIBmX3+p/
ZSn08NGjd7x5b2TtGJkmkf4jPZMUHyaEwqe/nxf2EkWD/yxG4lq2F11VIGYl1Edq/rGj/PqY1lB6/7H
nx4D9Gr4WW0E79vbpiS2q22pjSfNz5w+NKq73u8H5EANe1E3Mm2/e+FDXEVUfpwoMz2SJBadfiCo2jU
SZ2+A6vKpsRvz+7Bj8xDC2Av8BzuNr+93psBwDlRq7I/zj77bPrJD3OLM1SSRUKBXFXqSrjP3/Pq1WD02BSu
lB3jh1MqXJH9uP6u9JlJD5bHrs04GN3XibmGE1obNeJIAp9JKtFLLVD6Vu+QgGV5nepsymtXPyPPz+nRM
EITqDPQPzZsQtvgYo3nc6JnPmGSPcgdk4QT/UEEV8R/G5ntvIzABv5yN7iqbrZx+a3qbAv7SakjtM3yJ
lbXYZOwq3gjyW5XV7GS6KpF5NTuOfXevBkM6qFTVuq4HzsiECHsHtU8k3zpYmLn9AEJmN0nkAEePs6
k1hRNcYcYczXwrhhnZRX4j4G5jHuCJNZPVbMjC+qVBuYSTBonR0aJVUZrLyRWSB4CozZOL/eyt5
xKeHAo8NWrVbwmeDUNfIjAODl5mh/8jCT/1ZUf4JqkJ8b3UtvZWBEzhEYOx5QJ0zCf
YjwiFmezWZs91GUy0ZSNmSrgxHyPGMxXpRbFDn1pX13i29iTUxEfmK0o9+j/1tiW2qb5PEW5qedA5A6SrW
H5KHMGhw3wY6R6nX/l1O8EzVi79nqmErRaJ1jpv86WY/r5iX5yhW0FHD3fAE79o7y9wIIVnzYxHmyvMdHxO
9lMIHC0BsNfd4CB/9OyPrhZVLNE1pA6nvVqdOX4Z94tBjDwQIbW44GbdAgnPwG2cS8G8M96znzf0e
5JKZSQhu1fAYTncsTsc5tYHJbIGaQ3LKDfHD7Yr/wAxoKAW73vdoxmhmRp0lqrDBIKaSAwkBTpUkv4Ra
lywzyAdq9TIPMwWOzc9EAMFm4QdDIjLuGqnBESsmp73EDULYpPi0hKxFkIvJYsPDtXtb2qEnfyNWfx2RYx
bLWPfYk7xw3D3VVbyGN8ZXDyLATmPBwLCNcBwEfUhjpOzxGgJyK3ukbmKQIIjxRvP10Uzfe5Lk8pQoq
9O/rgPV+VmmvGfJCUC61PINE1Y0QTIlcOeUNkItNxRZpCp4qh6jzBQIwjnIcH8Ib3CeH36u6jqOchtI7
nzy4uOdlkjFKQpbOqJKB1lZfzXYmBZbVBbXdtbuxoKpB6OG0UDNuYPZMl8U5vdz9R7nnqJpK9pVgRfwoeK
+CEbZvbpq/e7KMQX6LTBWH3K4rjltlKrNMoHj5WSG6N9iRnC8JrS/ZXj52PW+JM4Pbd/nMNWBZnmZdU
pG2zl6bvIeOomoq70nLNaU8jYUe6/gEBNOFwRTPfDXX0VtWp4bBf7G4NHn3qpINC6YlXnQ9WaiX5pP
dm1Pi0WEcHbDiVvKjXsCkbaeeRNrRdiHdjZJyTNKuqh4yjnOnRLTfnj1lmH5kG1hlbwLh3Odb9rfmFt
XiXV9+OzfwOtklF67xnU3xODKqq+SthmMn8mdHkrQgEPYh0Fu7B0nbSoEgt7VO/F5ifMFQY40gCLzh80
JFwVGzwNC5YRJmwn9HoxHaahaqr0Gw0zD8yGWGdh9h12LJjsVi8xOGvBKZMl3HyO9eRWhETBGO48YxVk
9bwgZ3GT8UzXIEJWlmFPfLXj2rHVHxtHgb3DTlY7eQbL20BnORIe6ZvjN7OMGvkoG5QW6M2b5CCQgdH9
AA8DuMbyjWCtu7hNz+ZV+Trc+/2/PTDKHJuC3MXh6U0S3VUYR+qDDs0ofdYnWMBeuH1FcOEsgvOgqdJ2F
RbxVBbPbGixzugCCcsh1dJT/JGgBRpWoCCUpJe4nzTKu6WkAo3cHdJ3RDyXG4mxUAC9RL5tiHR5fvXzU
ujKrrVqAy/zfpDqEKlmEYR/l33XR6xRjbTaDyHjc6m51+HoaNhYo3qldmq+TU4/qT+X4B7mP/Puh9ebd
9P/lq+P/tFd5+34/rtOXhbGzw8Ln1tgYcnxd3jHgxrzVvLNWfbDkKVOv5FmPtZPnsFDg3nVZMYYh84y
MKCZpnnLi7O6W4ZJLRmb8GdxNT9VT4J2rEWbazWb4gV54hbFVJd4ZzNqbR60GE2OfpLG1P0E6xkXxbnf
y/bNTC7P4tm+DY6fvNF2KvsBNaC3rdFBKYcf+P7zAZB8RyVyu7tL98ucxJ52/S2jz2nKmRUPQDBoZpvaDu
2ZtnlvaJUsiDlq/Wj2lA2ReSJvn64fHmJeW8pEukxAgxyyw3z3EdjUpgYu9FZS8cLsWa+L/w5nNtT8/gxG
KuqQV7ACJnvZOTfQBQOIDF7h57/o7oXSKFEqhrZJ2PQHhJTxCMOPlQXQA11BPTsANBW3mOyTTqpKy/G6W
Gp1Fg3E4SROHTPrQc3FaYOGw38KdNqIVAFwSjZzmNrfyqUC2MOGF6ynjZdlCtG/iGbxVMvbOCXhpSz05
cKOQkXLh9K3/NI0u6ss8Ex04fTVfDBI/s0nxaqP7Oku7YRAD0yd6tvIr4Z7GATzvB2Gk14dmdqqpdVqn+
WX4oXOeNOpNZm+kWvyoxcmXJvqdBz4aK3SCJbd7FJvcTr7yAD5M4/cpVVBTA6zV0MuOuUYq58TnVVvyY
PnzuEHbkXTl/ohVD5/5VSPhpAubHJNTGEHazc8xjs/Wsxtr7wvdYrrxBiL+fUbPL1XJ1IvVZXg3zeAD7cM
e4RX6ZI53y1ezgWCvgEPk1T1/8Y5OZOzInN1DQVKQTl7nBlsXO5aRvONUFdiTM93HHGPdFXrj8wB+oBs
+Qr4XIj4XUSlnqg4t1w41KCW9GSNwfCDvqoJNBjTvh2sgIhMwP0pYJqp5ApMYXh9qPH6rSBTOXj4T9SeX
3C6oqxQbQs1pXXFnvNkhGtPtQ4y3IO72iJeoHOEmMhgqiFp8RRnJJvwjTCyMdYVlHUvM4vCG+sYnUEMZXP
LsMMrVtrbGh5/ZvFkUM8FiTG8rm5CTFyUcIn1GLvaVZJqlAqVcvWXXePt2FRQjBYAq5jpeLa9VSjm5
emJ9xLd2GY0qcuD1qfJqEM4yKL6iLabxLNCqSXZQKp2Uv+1imM9ZVr8/rfkgH3vrv8YnFnDdBExqC5ulg
DoUNKPbhZnLKl4qR7ClqNwK8VRk6vuL13QtEm4pjYm8FdYsdcTPjiScQ4i+p8Xw8QVIM0Ab9bcvxKg
Zlp+HN8ucgjV11HBhDuz0cGv4NNjTBr/HmKYrw8wjmHaKfkrB0Q3hhvrHxoVmvOQahOyzKNBvKN2WH
LOtVhAuV8mkGEs9nOBgPgIgk51OiOCi9rGKjOe5zs0qhxzuYPsDa6RMiCc9/3Ie+o8XfunRhcIg0wWvTG
EJhwOGfQgHP6N3WpAEIvHiamjNywpydYu2tJ8bbBQcuoDwNB/P2edzgtiSiDsBJGrYO1Sbn8oGEFsUB4RM
KXbxLC/1ThaSs7mmOY9Xca2YtMwWjQxt27izc9vJsg9ldwOqe6DEvAJfgy7aqLVHtHvBHPPb7dL3YKgCTbH
LTQJ/7CwSLDfnuNvodnimQfpJbT9XHjvR2L7HkW3mqnm/SyCz0/DakeCmzK1N/R7BKHBWuXNGOwsvb3kf
b2BaHVpsbcMk7JYgrE5iV80yuIqLSSKMvC/ktHDtgWF4JGPbRi8OLe/nJNqlXDX8PRwOiCYjOl/zSIv
JIw1MEQx6ASNjFeIokGz17pTemaE2nVIqbd1Oi5tmWndqu1SVMJKvYdfWHmBoypw+1uRjNUmLyRCt0+
JMO/M9UNa+rqPJs4j+F3+GPMDqcWPj56ar7tpcLkFLFVPDTsNE6YOpbFERoW8npTf9T9IGXxa2VVCpw3R
tK8EwbHfVhFDNIqlrQbDLdPFMsCQWCndbXyAxU6SGbxePN5qBWPZGwUl6r/XaEVjeRBIxSfMw3x1Mf
WoPTY7JMgv3C4RChd2hbd6zuZ3TyeXodXP3rCxQNXuP+Vmh7RMY3YKwAL9QVtn1Zzq47Hn91VGLHBUt
Myo6uqKhuDLlS4OM8XNnAQm4WVFpFSXpkC8Um3bEUtq2zGr6JUYI6btZ9Nqr6+Md+c+AO6Bfft
4wXQ9n+PvuQPd0G/n8OFvdHDFxXM5J3pfG3/z2vfx6HSo3g9bDTtOu+HCSdYtH0uXw8Bc9bBo9gHNK3
ZffF2+KCN6Ffe+Izp70w0ho/SPdVQNnLLLeChd2f6qWXKWPbPAHXzL/6gO+wvbHe5D6RGUH2W0OJMoxM6
kOAf4Xdx/iDu67gZz53vW6C2pHbvq1YJM1lUB7nxDJ6ZqZ7gGHzGdHM9j2OFKmbrlrJQEcYxaB1zNs9s
RMTSLhg8aKrTj6J9AhOQr6DS+OTq3Pol/Bd0UaFzfdtKP75zgj9DwP8+JeAxnBiqkDcN/11aCkSCnUb/
aHhNMPiuaMt7DwJehCK0jmLvbtDPXLRszcbvlfwgSnzbHnGc3VobSZz6TqWewOqtE5XKmJbpyr+t2dp
yO85qjKfgVh2DBySQHzY1pycHpYD8SDJA2rtL3p3p6s6tvk+CNXfO6qjYBrjLFg6QAxvF0Jdwht0sSL
XXtcN9u8IKcDf6bdGR3gVv+lP1u/BXo04vjc4eSCI0EvtL9Rmsyc1sqSI5UpRs5H9UwHRlg6NWvYlwo0
GZefU3K0GO4n3uK/kLmMvB4kYqEgIBcD3ee6KwY3m8kkU5MwoyOSN/bQsVczxhhHhlcQ9QKJdCh3Zm8R
C7zFAu4B8l4awNIOVGNtIOFZdgIAmnwJgQ48RGzyXzsvwz7kR5GjcJs9HA3KFELHhHO1i5f/m6CBrT0
ERLeYC4YN+YB5YqqzEGs8SmG3nE+j3w1zAB5RZ+L2Py+ZUj3ze+tfErESXFRYzmM/5n3J4fiwHoFTiE
YLyxVAYqAM9f0A/jngdoQnd8d8ruXvC1TAeA1YhWkQPbAvBmGvXdm/Xgb4UfjrO5uXSMczaHz0ueYQtdv
8BnCbOa9Wx0YyEaPDQP+SPJXB4M3dYLB9t8Z4JmaqPCQ6pbG13cBeg1SjgupJ5J1aCMJTw01nz59dwbY
lBsZOO3tRpxr1pBE+/qrnaShsw5GaeFqFNWt3JjzZLGhu/yWBS0BxRlHnG9/rdEq9Bhxre/hUl6z9fY
6irAlpiXp08E0JivlHzYLr3vQ/7aYmRdrchWNN88CVYIU60uxi5L1krXJWgnhU6MT6ZzGVLN8DYFehT6+Lwp
VTCJ0DRE8oT5fFNvhdzJufuWpIgL0sd/V48tsf8ZEQNwBkwfMFi2R3d4W6/g5zQ/ccsWpM6K8t7vHndq5Bk
4nAvRotOPzJQJ1yDRMZGIXYwVfIOyOhjZSdM4hzPFmRpzAaIvJ8mtUvJPrY3MGLGPvBNhc6nO0pdf
k9zBKB56vtEqNbbAzthgXVVJ+CkT8vAXZc7Uq6yf6fzS/JbXxBMT/DwOxdJW4OJd8SQfOO/6+ExYAvW
rPMyk4zrLxcwrf7bWrSV7mX7yhoQy9f3AvJ4LY6V9L7NCIpAD4Y3WeugNewIGM8FEB2qdovnhRfY/pLG
q2spSyWI7qwDiqrhG32WmYjnA/Jb8hLxt6E5trwWNyIdAXxNtC6Rn98ZCGkh0zdPaPHRbHgBGRtBJIUmXO
/ZBazbDIGJ1NfUiHsbn/Ta9HHE0TCgj+jVdkz+5HbdGtGQWhJgZx1tJ69rzpLKmu/Bm8n/IpLuqjxI8AmB
UuRKLNUBl6aZxn4CxxUX8ZtXyUMgYrvu1mG4RGUVGejhTu2J0RE/bvuEOAZyzs0Rkb5DGLGaWQ7zxtjHM
ly1NjkO/6JZ0Nb/ol3jmBN29nr9jj9Ao5G5K3LDlQq+hTnMrZL4s8eBu0SsbLrJyW7fyp5Gp7EixVqz8XqC
7xmi0Wkyt6nJbHqzGgRA3PGyi6ggkgQ0Lq5LBJoCJ9YdO5XMA5TqCbbKwVHjNtEyBkFXQqahBvf44I4/
b8crYNa4n/1TrzBR0/bTk+TYIo5+uFBwDsqNRCjTTbRBq7Gmbx2tuw7P2YfaqMIWeg6qhmXF+o6vp7v46
F4Ok5WpSOWGJfWkq4o/WfJ5ReuvqKSSuXjC8b9eKSgtvxgnxO0qLuOGJ6yyfFEg6Tx9WDxfLjb0LwFf
AtgZk5dnqt8UbNDTdkX+r+YSi3v7JmXHn/SNVrQSelXgP0NKOd0e2L1+hYqHxnlWojpOcXtvWQ3VzcnmZ
CVrAMU8sPQ1Ky6lHvZVn+FQkLkFJ5QKbT1nlbnZXQhHL/iWx/z3nFvK8kKZRIZsCjsMdwDNpcmU+Q2jMn
T/cJzwsSWdsaHLFfeVJRq+35RdeRYNEnHKYnwGlXnk2pSLx1ifA8Kla0oKYbaUMpUm5qF4aAdl6/+PWhkvx
Q8XS7fEYnFXM5A6eTbvjmYEFSlq0cOtZr1sYQWgqhMr5tZK9/Nt/7r39cwa18JVWDPdv8dzkr+JuI2Uw
mi1a9Lhs8mZojw+8G/mUKmDMDLKPbDwQgcrnd8dsGKD5K+4/Nks13hMo95Y8PwY/k/fJDfVi8gsXSqf6c
SCjK9D76GS86FD6E7XcUsXW2BBoNIeNHhbr7HQlmnqOQfbeo4eRVXPK/UhHD+7BLWYuUfeYp8cCP/d1ht
LQoPVVdyRWqMMTvVHwpQyAkEXqcBFdzcU3nQZWgoQfKwWhmZSkldu8qEMpKxafvI4vMBsBmOGx4rMwV
ce2ueUxv1NWjuPXXEkJTvNV3phKmDzXo3MIGF4uBO9MD8HWPO0+BH9hTZ1xmvJfcMrRvaNl+vGd1fqiq9B
8zTdDmcGl4tWR+g4r2OGjGqS8K3gWi/ozfEOixenMy9o9nC9ph58zS7TMV8HXxwyr4DtdT4TMEv4ODl36pgy
GUMTgdzHCrEj6Pn8lYUGt2Fw5MnHcsHK/0aE0YT4Gn0GZNhNqgnVZWA7NUX00kQb2o56qybIX/h/NO
0vE9YDSDIJBrH/IVX4BM0bvlUMtQXflUZMMn2sxNKUU3iNrqe9Y2VBrwWnyS7PaZyzZh9YNMvGt3BPnqZ
jDLsVXbV5Fu/pS8hMPTFtdqChIF5d6RHevEiRlsbeYk+o8XyhQMGrzY9+b4Vq7W0gjDNiQIg7tw8BGWcr
A/UTkiPbKe6Dl8xqEyA+X+XaNB/VKjR+e0t3Wlr+Eq4o1vR3y9tfYfVOr9xIgd6sRwMUjJGnkH+iY9b+X
vGgYn8JqpXAQ5q94FiEH4IUX3Pve5K0r9RKQC1bAZCXzQrG8Nea03zxfkJhvH6yVPIgSo09Ff/eAvPQ
VrQ+ZxWGfE0i+rUw1SxQpKYJBmWCE+tRtwMSQcpwvh9W0DROXKL/sDDZh1S1wtxFrGwaRtRx0+4o+ue
nlFrYfvlZKwXDJJBx1lDxMEDy4QUw+DP4pUVrLLY0+6HEGX20bsMcyMNz6eHGbT+mzETvsEmB5dV
ABhSmqD1EaDrVJ3MDL5i0qOHvJGH3FoLLUmTJnWq4XYgzJkXAjzKNJYHqoWlxwzvIiwNTbLpkuP
cIrsqvfEa4ZEy/lSvRUF5KWDNT1gLCQAPgfP6F9pZ1hUEBomIvHf8a1bfLaNYSBYMAUimhWrYd7xWoMVg
Ea8xLWT5ZtLJVQhCj0BjBQcgRVEv6eF/QhxqC1TKg5OJ9p/D5kUmEB6f5R2lAKEVbu9eF5E8b8TDlVIbD
XZCAtOq+YHAzRczU/SrYWPpTZZsCzrQW8U1Q+7EekoFx5fBrLS8aZmeT7YjbN1h3eHPOJE6bXiZ8mhx
Xbh/BRv1L2hnaN3Zk8RpN5fgTEPyNXwCNfQRlOwRAQ+Z9PwmZbXSfLGL1e6mhS4rPqYYmQ5Z6Gk
T7NeqoVUmghiO8QJELQK7kSJZEhGs1RJXrkLOXgC/vkgeVFsL0U3vZkdMqiLO4lvvkIEFKfdfMI3BRVNm
ld1LiFf8T9T0X6uMdUEjE6Cw5GpWM0lmUNx3wMpd8kK5nA+bN4xfGH1rgxkiJ0UOKaDQpIa39Nm6ay3kj
KZbvWRa41MXiwTkUuN4BPqSlCQNs5vHUSz4YMG6CahVO3cV1mlU0Y4EhSMTZTC5M9xE1Yd7BRTi2flDH
Hz8mW9YbCUnVg5ZvtUF7Sn6wcA33fJQzlL7ZP3Fa/7xfl3SpOqR3DQ+ygqS7wMCzgJPhbuSX3OEyE1l
5+PUe1UgdeJpE/dcoLp7Ukb1Ia1sNTGDl+54wPXrbZlavvbk0zKnH9RGTDG/ut7C7x7ZiE6qDrrWXCtW4X
pGkRRbQl9bvYsjR4WbQHSB+d1lcUbuE7h+oRwKEWkcO10Y3ll9xhUXRIOt1+yISfEEd+lpr9iLfsWSBB7
Wzs17jZFcevbi9xg4IgkyjRfkdA+hmgQXdGYkDHxEkPn6HZ81YyYZ4bn5qb/mVyyqe6XcpcI2EMLy8jbO5
kZCqFWhh2Iy8yjCCdhoe6JJYY8OfHTnLJbIBiFFqcSU1BqKdiXPKKIqMYj+2dQQAX7hkdMifAkUWC2QZG
JTNjt/5brv88XpiTQdnwdB1jtUe1D7xyYafzCHMkMoRG6j0Ff59Bo6ewBTTqZDZoI+BoVw/ANRedSdmVRqW
rqtQ0+ba7Tj0a8qbuq8dBvlzgzG8aVpVS58QfDKNDemTKR5ZqO0zrO1FQDLBmmhczyF1K1c10Xks63tv
E8IOM6bKiPsAnWCQPa5YobO73dZ9O9DmDb1u0cOd4NZQVB5ZPDA0NwBxWxH9FqKJA5jnjg1H6o0Us0Ym
QGtQsJRa2xHdkOYHvDHWRqKT1nH6yBdQ6vzBB0RPHGnOJVUzWqs0lZSTYbjNzvDgKDf8s5v4HuNDPqeIE
OX4N8nxpowrWtOeKeEXSkMSrKUFd8ab9s6yE+UmwUWtlhk0d0eDBYfZfY3NDnXEBwdTpY8dFjETTIgzIaz
mEZ/LZlPqaA2wNHT6PxIqOvB8WY7ldPJ9RZJCPfrDvTJyArFeF+FhxajU/Ixaq6tpibUoBRBZoOfpZ3X8Kd
lo5jVSUUZ9kAA5a1LOe3W1DyS2pmjG3EWPMSn0n12jHi10SfPE3PhbXFEUmQSrGCSL/JIGBaPx1OWGmTN
wqg/4FFpnnBpKYiqmMnl0Qw0J4v6zdvhz8KtGnKbGWW0ONm4tnDVtTQLobrcmgtUXCxfpnhDkKG7ZmpL
yyJwdFKPJtHwAoNHAlnbOyFcNZIzwMszWsA5ppJvu2UpFsdS3krtaA0X5tZ8JrE2ZsXjQ8qTrd0LSVUf1
R/cNbeKq3XxFg70WlhOCrZXcBk6dUmsWlqi8+eV4YAFXwgqGkuaItORdoHc76xLfGxZdL9TGxV30UqxXcT
AjHqLwxmDtr+mzD7XwCgkS0ebuV2VH7LU1nnNsHxZlgIXGwYm9eOfOVeo3xMbLdcgETw8Z+qL7O2sLC
wR8WY1fhvW0aZZ8CgxKiOJjhX3SBLlSlg9yO9y7rMLMzRB3bMU4r/fOeeK3avDFexuJ7fZOsBVtuHalEMm
5xhs/kWUA1H7vpt+sLVYzzNMdrVNd2mFHI69GJRIGhTK1a2QMWLfM0fYS+hJX6BXRZgLHiBEcCRLW5oJ8k
GFo4RsFGjrSjK/zEsPSvr5x4VLWzamQnk3BEcrzgjRPgzGn98dVWXtvrgoQAvPoTq8qMVNyHVVyZjGa4RZ
c+F0FYBPjCUNJU4Fz7qzLnYSdMzvbFU0p5Y5Aqsr3Sxx7BJ8Mz7lzxUyy4fEcqwcsYaCtjbOhQKUxMQgXHzEj
T9O7a4q7QsSv9RTFeWyNrH6Gtf3qlmoGbVOWrvzm0sdrG9mjO+PszJ6eJcDwKVAjYL9QWYa9eqDzPUzfDf
vjgI3o5BsAp5+K6y2xRqS9bJuWYhrH5GyyZLbRX6a4BV2X0lZ+FGi3vE4fWxJnLuGIt7a5w1Xw5CyT7sx5
XyJSXSV3B6gBWLCJIdSsvb2KiVhvu5OHUHWPvJ6mE4XLpHxdtRxt5yV8VRvRVepMm5ENfhcbTUt/rP4zo
LnQnA6rNGcBs7HKvAZ19TmOW3GO1CT9Ca+x3AVhT9ju1Zvg6jL9NxgLGzqkIEr4B3VfQmxU+U06ncWd
E4Sg2idFx/1U9Qzp1ZEjb4SRx2GLl6N/3Lhs8Gb5SKvwMi8mVlWqbbMcAaUGDUxOPIEVD2dNkbvE4fCnYW
JQp2y5u3rvEjv6N/F4k2JKzcUeR9aWaTEg6qoyHVyXbPfdWcXZ8Pp4tByFo8cTfk/c2qhP4bYIo1bsRKmD
MBDCj5NjAPv2vQFn3X4Bg/kt8PEi9gTfVfC3dAMjhlW6xv5IevYQUyTn1HQfOdyL6VIXp4k6KNDCJ3oh0sOa
T6HbhL5k6cyTtyyDk2eG9OlR97lvAgtFvBpdfLE4R4OYKv0jCHx6VlI1vGp4MOD3L8IwLw7WBTWQzvT7H7
L8LuKMd/mY+cZVRTV8AXG7VoGdHmnV5U+8DkGsxRq6UeRxsfXMelQZ+9NlSLdyqPs0jD+x5OmX8k5NHM4
PRe9B3ahhCY0V9jPc5fmSGR8RlPcvbWt0kXdaZM5w6rApBwfzKEIvh+ma2WsTmJCu5SBa5KeJQF1GkXIu
pwjcwr1PLnFtmJyOfVn0qGEjPZ1bhBZaVR30UmI5EuoGGd2SFwVIYp7mfqILzZn5CcMR5GhPgCFc/dk2u
ElKbKJqCnBxHJXtbmf46PdEMppRjOBrcqnodXh9IF20Qrvmb3wRlE2CDcD7jyqZGZk5CD8tfKf5XuYu7rGw
xOFUPeYA3Jm31lBoxa27oC6CKx35Qm9BdoWNGwItkoGBoymqB9ODj+wJ2ie0GsJ+uq7l0smt0aEwFX8h
2QCuUz7R3ZhnlTBfKvqQV/fBtx0ZEbIwMx3/JhCDvB+uOd9hyy2mmbF8bnQLx9VcQdsCeGBJQbr9q3oUHu
JDrjOEeYtAdPzgZT0j94jRu6UYVDehqKlMJZczRnfdHMQhI1BQMd3Ip3IzC2tFmTaBajnhLrF6MOx0xwGY2
H7fsgBnBqtD2DM5RxSMBCtyQUDR4yCFeDmWI3ZmOzHfTeaZ2x6pYkHa7e9MQWu4ZaIawa1mUXbFUPBr2
jfFvHMvDhDJuzE19zRr7BGYDRJX4wo9SNU/gDIYhXSeDRxT2Be5kYiz8DL2fW1SDQJe5i5vkbk9S5u8rHl
IAKXhmpSqtKXKrUjsl3eA62+HRkujP8WMvi3JtvsZM7Qn21gg1oTNh2CyFGOtEzGl6uw/g7jRLYYeT9M
K1mZC0YYEukzFX9dpYe8k9RHDTvPWqZTYLz8rCKZwqZdBI7N3UKyV0QTH8Ks1FLmbLXnrvLVT8kHMO1n
WhoHGAmbSUNEMZmHgmVj4cc7QK6l8zMyWXHHzKEF0/IX0KO8rXq9Bpa1rzhf3E4ySLwV51ZRI3yDiRJqX2K
uTFwPNQ6kRyBb1sN4WGK3nxvDGSvdSUz8XlpA6K+FIZpEmZ+SpvVS0CAwZyF9vI+HMbTs9b19Jb7G3Gs
Nn1b3PnKNJTD9iAm5GUsJGXPaQ5g6WEB9VhRBV6Wh9F1mEF6/rJpjcqGTcHLw2Mg99xCuPxTGpxZbRHB
w0MFbq0ZbjzyqoMlKNUJ6DxlF1UpO7ovNBoTM1t8FWx1ZTFCv4KCBQYkZtZgEQRP9oJPwzyNH9KZZCuSH
yrHo6f8Ja93JU8pntt4hOINPBYPHSi/rfuxlZb9UJWFyWtdTNv3/YAvXoKUMIeZWXf4kLmefi+vNhrx4bAD
7m3UPMnj92gI9HMVFCRt85CrPGYkJYcH76v6pftTQIKpDk/tvhEHBzIjO+XbF8bY7s2DH0vkRr8kj2QhXnRr
zswtfXLu7e4DFy8YdMYOGPQBpQgBDqAxxNkpbH+i4zlAr3qOFgbNzmxNqkrfnK9Q18AuXxM8Jp7tyKzMYb8
xDz3KdoKRWh2OVnw4TcXW4uPGk05CGVuROjgb5TXsZSmdWDnWwZlyd4fo4mN0y+pd8c+eD/jmS3CF+Q
xXeNQjBq2bC8pmQ4GiWjy1iFSGdfRpuKGb14yYoYhhQbKNsTd0aUcyhvLuHFNLwQJ8jpBpgRMs/FaBhFN
Hy8KwI/z9ZHJLf4N2Dkvic+6MYW6sjrXqHBqVSAhChsRvFc0CfMhZiXyUiiqd0cQRV4I4pg4Jq/0RAnB5vFaD
f0s7a1FHyNmw3SDufNHbtiVKIxBqgzWRn3JflUQ6bGwG9rIhi9gBUbmc8EOOkf0y7+4YSgx0M6aWS4QgiN
Lf6iyR9nEPxMFr/A1WaySkTV8V6PP5Gj1OZ8rnmLl9EIORgCEHrKXCK7hz5hQPYgjJk9RbPKyHSr7DDr0ppd
BETkP6Tt1XWSLw/h1Gm6/uqkjbO87Umcx5S4a+7xK9GiNf8PN8mR0hPDe2YPZcthT+bZDHNKcV4TPQ+5ejf
JymMeDeLlbyjl+vBZVjsTfNx2fqDsaE5pvlYA3kb2MP9bbrLUTlSTXVmkHoQV7Lxn2BTpymRbi9k+MRGkZVKg
sddV7eKPzuz4LlmSUZxVv8m9SjAyd98sd6ymwZKexv3iEmqcVBiZe64EdlMBLOiT3e4/0YFq/c1DHJu7qvt1VJ6
Qnr6I5wEF4vNgHWvjbvMoOyW9Zs9X75Lfq9MqYdRdeSCtU4NQ6CWS+wWB2TTr0Epd7Ye+GuvcXu+fS3xZL
9DYuhNAUlOUZCnPjajz4mc78nN3mFy9uwbAqz5JnjAq5lBxWu7f4ikVFF4F9UGeH8C8EpT5yC7btd46pGr/W
T48qOtj+2L5Uy8ZaK+M4BDGdqtfQQR03Pu7t8I2F0gGOHIWvSsxdm+9VJXwMTzTvdU0q62E6V5kJGKAUj
4bTMnrIJHxGFxVfpvLsLfD4OexKIsX9X+TqZe2R/iOjXNVO2NHC2C6NwTbTb+E0p/UuE7PjOZ0Rh0HSG8j9W
F/vWE0KDXdmWnoEyDnU6ZHA6lYwvK0EEOrBpUnTWcs7GK8YWahdx3Z0JJLmGaKBO5y0EAVXTnOvhpKt
gZZ3M9Q/DRa7S+YMzCUBl+wDbMnMCakmo7sNb7KygmRNAiFhwJ0W7zcakp1W58v9NJjPNKWyF0q8Ifoa
Dnrqmy3dwh0i8pXkDCClh9G4oQ3DXgtuUq13sZCprY8w0Gw0B1Hqlj7dI3qrtO8KUnGjUNKxbaqg/Wa1G
8I1IuYoB4rEGO8dodbiPLRNQe7Qk3sOFCAaXvKpiNKNK3D5MnpTDAqW76Rqiz/GxzjNppMc6pb9XcHvfmv
Gjn7CzhSZDA1MZwcuPSBXqZ8U8DZ8gNmvgQKU3HfbeEupm/yxTlCqjlKBrkPbw2GHiefDCvp6WWE0S
hN5ouo4d3n+VRiiBL9ptzEG6lXMSAymCxqRTO8tJNNwnV+SXRJPX0WfiSCa4wf63VAZRfBXLmfszT2S4sO
NYq4/QudADJKJmuB5wGbU44DbRovmX1+YHMAR0F2FqMX9VFpYF2dKBQYH4OgJsm7qKk3A8fPQgc8q
iSf6mfaetZ2U5WLCLTqwbLH6jqQrBoXK4WJRzkVFJS8gS9sHdNUbzbzXlW5VoEQKCaOZULXD8E2ZKuNOJJ
/pU/s2NpK8hb9c8yPDoNVvJbfaRUt5W5VmiH+cJzkVxXoNOAsMYfIGfz0Bqb4pcyPpq9wVUGO7bh9uDRT
1dyDIl4QdSfjEz/ucKGrqgemfbNGmC/QvCvFA/kiKhk1SOMSAjPkNCrFG2pLM0RMHqzUVQ39I5jy4dG/Cq
Vm3+o5KSOnJU1O29wU4d4MnSIaBN7Q3WJbmCR7Qq/Gah5u8MJkb49TAGb5mKbrw+HDT4oPxjvXAB0zPXK
PrDTZNh3A0x6Td6m+oNU93mPN8NebgFbl/DI/y8tNUH1gCB3k8J2+KJ9x0vLxfFrOgRRQnQdR1xCeyWZFGy
hpyeG+FU0t5wprmzFmbkBLVtpfX4tcHvqy2PDuDkyufFXHfOzbJnbHtHlSxKHERYyivbH5R3BbMgXdQRjm
4mkhXjlk5w+DpJNAT0HGRx8bhjtECqp2XIoYebRN1rn9/fNfTR8sx+bDfNW6x9wMW3nRP+KEVyhVctCWM
m9Vbz7qNdjQlNDLx9jpN/Wl+IJQ1HdYHxBcH8hx9K1uaMMdFlYtuzEfVwTCXGgJz4JhiA9aBL8n5cu2LNH
5N3Eg4pv3QMbEHwKb8PRa13ZhrUnxHQdf53l0krNqXFsE6WrfBbD9uUm3+pPYv8VCoTAFnlQtB5uAiaA2Oq
BNHGQ9Vhs3Lp4jlmP3xH2Uh7RWMpnMi/OPq6ApwK/v1ByqVs1rqGJcZkLOeqpuJUN52Vza9QdwH3oXEXlc
```

KQ5FSFK+SRIrr6vjpiL9iZtea2VIze6uZfRYYqvgoEZEma8X5hRJfildJIsV1dfyM+a8FPgd80/v7e
942EZFIiWLKQ5FSFK+SRIrr6lTsiDvnhoFNTSiLiEjdWm1+ocSb4lWSSHHtX8VPx8xOBq4Elhbe3zn
3/uCKJUFYevV9Vd1/3/UXBFQSEREREfFzmHIP8AXgS4Bm2ouIiIiINICfjvikc+72wEsiIlJCxPPKi
pSk2BUpr9XriJ+sKX9vZr9nZkuqvbKmmd1lZgfM7KmCbT1m9rCZ7fF+HuVtNzO71cwGzWzAzE4veMx
67/57zGx9wfYzzOxJ7zG3muSVJZ6DRFpvmzWMTI2SdZ5P6u8ymXQV8sUCUqp2J2aytZVJ0SSIl8X5
ran2fPyCHd9f2/Lte9+OuLrgauAfwYe925+Lz/1FWD1rG1XA484504CHvH+BjgfOMm7bQBuh1ynmlz
WlncB7wQ+V9Cxvt27b/5xqyu8hog0USM60ZmJKTb2PcHOvUNMZh079w6xse8JMhOaKSfRVix2+3Y9x
1BGB5Yi2axjKDPOp776OMuvfYAtO57motOOpW/Xcy3VvlfsiDvnTihyO9HPkzvnvgcMz9p8IbDV+30
rcFHB9m0u54fAm8xsCbAKeNg5N+ycexV4GFjt/W+Bc26NEJEmakQnuhlXyxQJQrHYX
XXqEjb17daBpbS8zMTUEXVh8/YBVp26pKXa94odcTPr8i7oc4f390lm9qE6XvNo59xLAN7Pxd72Y4A
XCu6339tWbvv+ItvLvYaINFEjOtG6SpvEVbHYXbZ4ng4sRSj9/bBs8byWat/9TE35a2Ac+CXv7/3AH
wVQlmIz810N2/2/oNkGM+s3s/6DBw9W81CRpoprrDaiE62rtMVLXGM1CMVid3RsUgeWEaFYDVep74f
RscmWat/9dMR/0Tn3p8AEgHPudYp3gv162ZtWgvfzgLd9P3Bcwf2OBV6ssP3YItvLvcYMzrk7nHO9z
rneRYsW1fGWRIIV11htRCe68Cptz153Pneu72Vhd0fTVtXXu9i01cQ1VoNQLHa7O/zVCcVd8BSrjVV
tzBb7frhl7Uq6O1ora4qf9IXjZjYXb7TZzH4RGKvjNXeQWwB6vffzWwXbLzezu8ktzPy5c+4lM3sQ+
OOCBZrnAdc454454bN7JCZvRvYBawD/rLCa4Ge/ue4LF9w5y5tIdb157W1AM
BibdisdvT1c4d686gu7NtevSvMJ4UdxI31cZsvl1eOK+DO9adQVdHmtcnsi2XuhD8jYh/Dvg2cJyZ/
S25LCT/1c+Tm1kfsBNYbmb7zeyT5rDrH55rZHuBc72+A+4G4wBC/B7AM65YeAPgce82+e9bQC/S+5
CQ4PAj4EHvO2lXkNEmizfEUmZ97PORraaURc/WVvKPZ8ytkijZbOO4cwEf/2/f8Kel0fo6mhjdHyKq
ans9H1mx92i+Z2Mjk2CodFxiaTMxBR9u55jy5pTeOaPzmfLmlOms5/MbmOnprIz2uUN2x5neHSiJTv
h4GNE3Dn3sJn9K/BuclNSNjnnXvHz5M65NE6tSX+dU6R+zrgshLPcxdwV5Ht/cCpRbYmakddC
js0wHRH+s71vczrbKv4fMrYIo2W77BcdNqxbN4+MB13t6xdyZu7O4+IuzXveAtXnrd8xn01Oi5RM7c
9dURM33DxCua2p45oY29Zu5K7dz1fsl1uNX7f8a8A/4nc9JR24JuBlUh8W3r1fWEXQaSpKnWsZ6vUk
a70fPnFRPn/wxsL61rxC0Pq19WRZtWpS9i8fWBG3G3q21007i47e9kR923lTotEU2Z86og43bx9gDv
WnXFEG7upbzdb1pzCzd/ZM/34Vh7g8JO+8H8CnwaeBJ4CPmVmtwVdMBGR2aodoa6aXS8A6UtaXS8yljizRaZ
nyqYgrDwrhTukOJg+7OtqJxWmr7ssXzZmxr5cxBfuaI/wqwyjn31865vwY+CLwv0FKJiBRRbTrESh3
pSs8XdsYWSZ6u9nTFFIaFcfe68uhLDJRLRVhquwY4cvyc13oGOB54zvv7OGAgsBKJiSQ71jPntNdq
gGvlLXFz/NVythST1aWVtaqn1sqZXR3pLl17Uo29e2uGHfZrKsq5kXCMLctxRd/K5cJaPDACA8+9RJ
r3/XWkm1sd0e67mxaSeGnI74Q+Hcz+xfv7zOBnWa2n0ryYaowomIFKolHWK5jnS96RWVZq42rf65p
dMp3tzd6SvuGpUCVCQo+UxAsxdk9nS1k06nSsbvvHRuUkarr3Xw8+7/e+ClEBHxqZ6c4o1+vmoXj0q
OPrfq4q7RMS/SSMXq8/Ti43RK8VuBn/SF/wRgZguB9wLPO+ceD7pgIIiJRp/SGtdHnJpIcqs/1KblY0
8z+wcxO9X5fQi5j5yieAr5rPb7ZZ5pUhPhGRyKp28ajk6HMTSQ7V5/qUy5pygnPuKe/33ywYEjC/2Xrs38
i8JKJiESc0hvWRp+bSHKoPten3NSUsnK1VnsS2Zxcc4fMLFv8ISIi4Wl2Jl4aa0zVJv5aTn9Rj
1pI54MWIM7kJ25bNcuMREO5+Jtdn0fHJnP3m1Cc+lGuI/6Cmf0XYD9wOvBtADMbI/7qmiIidWlk5yK
MTBzFsgW0UvaPekRtVHV0/cdvqWWYkPNmsy9WHzjSvHBrjL77zLC+/NnZE/KVSRld7mqERxWm1y
k1N+SRwCvBx4CPOuZ95298N/F3QBRMRqZbfNAjVXKPU9KTsW/C+nX3UaPXlnNTYsq7Xyt0X+U2HqZI9ZKxHZ0Ky4vJUNJqJcI4I5xNcTalaNBCC6pEMx6Hd7xKUFwxxZHN8VNoPfyKGGnVnTd+4KmB3kBu5+B1/xLXRV1ZvFwSzWZfZ
4F+VDTtxGnHNIu1vfOBq9pvQsZ2g1Ww7ZevlVOh7ltvLjN3dvFL9D3DMJFbxG2RY4V8Gw/i81qZ8HWk7lcqjUxVH8JOobacKTNfVfL/FnO8fzv1yX2S0mK+QfRM83zygfd1NStn5G68CkFcbHn68zrMU
sh6Ojz+CZkRlK1tpF3tqiOnYvD9DGCbk6HnRuz1ebHrrKSJiGKs5vYOaZkF5eS5N+fijYJQPpAzF6DEt/ZCl4kRJlFtHfSP8j9PGlR4c5OMdh8fSwP5EgbXzqLbQqC8YzCLLHv2cw9r74nVp8Kd1
Y97bN8wxy/sUntYpfBXyIhIS8p3QGZP+yjXaOc77/npLPBG531eZxvz5rQd0bE6ekFnbkSxb3fVC4g
qvZ6yf8Rbqf37wnCmaR0IPzH0+kSW+UViu9yovRZ4Sq0OT0xx5arlXHXPwHTs3HjJCl4ZGeOGi1dw8
8PPcPNHVvLya2MzHnfm0h4yY1NqD6tUdkTczM42s3vN7Gnv9g0zeIyiYiCVbYAXn2uvO5c31vxU5
CpdHDYqPsn/nAyWzs213TAiI/o5X57B8ps+nOucRD8f27ksULOpvaYa0UQ13taUbGJqs6g6QFnlKrb
Bauumfm2Zer7hlgTnuamx56hpdfGyMzNlW8bexIqz2sUskRcTO7APgr4PPA/wCMXBrDu8zc/c0
poogkVakUdqVSyVUaPSw2yn78wq6aF1RqxDvZ4rJ/UymjuyPN7R87nZ9lJjiup4sXhjMc1dVectReC
4mlVl2dxWNn/pw2li3q5r996G256YAd6cjXnTgoNzXlKuAi59yPCrbtNrN+4C+4+C8BdcRFpOEqnVIvl3+
6WMcqM1Z+eklUct3LY0Vl/1rZkxMZbnm3idnZKkopdK0KpFSSsXO80MZPvqu4+npap/ucMeh7kRdu
akp/2FWJxwA59wAcHRwRRKRlbvKfXZp/m7OoJdUNnMBXFafFebGZ/b4Uky4/H7DHP1YvYUq9KLNbW
QWKqVzx3e1ZHmznW97PqDc7ho5Vs468SF3tzwZ9nUt5vXJ3Vx9UYqdwgzWuP/RERq1uhT6tVMP6j26
orNXBCnxXe1Kfa53XjJCm568JnpKwT2dLXz+mQ20qfYq60XcZl2I9GQqydjMxa133jJCv7HmlNoT6d
4NTMOaHpTEMqNiP+ime0ocvt74MRmFVBEWksQubnzl1/OjHud7ImpI0ZCa7nSZzMXxLX64rtazwYU+
9yuumeA333fsunPcHR8quorvDZbLfVCC4mlkny9wmB0bIpF8ztn1JNXMxO88OrrXHnPAFeet5yN71+
mvOANVq4jfiHwZ0VuNwEXBV80EWlFfkn+pV9Mx89PJrqWj28wFca28+K7ag6TC2MBR9Gp/yxbPm/69u
7Mt8gc4mmoijTa7Xl1z75P8tw/9f/xg89msecdbeGzfMMf1dLFs8bzpi/p8/D0nKOYArNwl7v+p1P/
MbGEwxRGRVufnlHq10zQKO9nAdGer8BLmtXR0D09M8Z0rfoXjeroYPDDCbY8ocvDQWCAL4lp58Z2f/
Zc3NZVldHyK7s429rw8woNPvcSVq5aTdbDjRy8Cuc9t8MDIEb/nRfEAR1NNpNGK16vd3Pwb7+DK85a

```
zbFE3LwxnGPPmhD+2b5h5c3JnWKRxfLfeZvZj4D7gb4CvAG8LqEwSEUuvvq+q+++7/oJIPb/EV6VMF
tV0zMBfJ7vajm426xgdm5yRxeLGS1Ywv7MtkBGjWi6AlBR+D5KyWcdQZpxNBfNcb7h4Bfc+vp8rzj2
Z+598acYc8bNOXMgta1dy967nZzxPVA9w4pLhReKhVL06+hfm8FeP7GH9e05gfGKKmx/OXUEzqvUi7
nxf4t4594vAT4CdwPWBlUhEpIJqR6/9zK+t9tR/sSwWV90zQDqVCmSUspYLICWF3/nRmYkpNs3aJ5u
3D7Dq1CUcv7Ar97mt66Wnu4ObP7Iy9xl2dbD2XW/VlA9pOaXq1fNDGVaduoT5c9ro+5fnuf/Jl1QvA
lTugj4PAZc6557z/n438GngU8CHgG1NKaGIyCzVjl77GU2u9tR/yYOBzuC+qFp1RNTv2YBS+2TZ4nn
TsTFvzhufW/4z1JQPaUW5erVyRqaU/CXs/+w3VpIZm+ITv3wil59zkupFgMq15IsLOuEXADcCv+qce
9bMPtWU0olIy6gmdWC10zT8drKr6ei28pztZvO7/0rtk9GxybL7pJEHONWmwBQJSyplzO9s4/aPnc6
Cue289voEf/fET3n5tbHpOjP7wj3SeOU+2TEzWw8cB2wETnPO/dTMFgDdTSmdiLSEahdf1rJwrdGjy
a08ZzsMfvZfsX1yy9qVdHc0pzOsXO8SJ9ms49DY5BG5wy867Zim1Rkp3xH/TeBqYBy4AdhqZt8jl9b
wS00om4i0iGoXX0L40zSUxSJ6wt4ntcSxSFgK17kA0+tc7lzXSzrtewmh1Klc+sJB4Hfyf5vZPwIfA
DY7577ThLKJSIuIa47ssA8G5Ehh7pO4xrG0pjDWuciRqsma8oRz7kbgUTP7zXpf2Mz2mdmTZrbbzPq
9bT1m9rCZ7fF+HuVtNzO71cwGzWzAzE4veJQq7h3v33eFNp8tvP8J5/0HushqlEIiqIq2mKNJviWOJE8
RoNJTviZrbAzK4xs78ys/O8zvDlwF7gNxNr0+mc751Y653q9v68Gfuic6wE8HnHOnQQ84v0NbfANlAzulbE
H+BzwLuCdwOfynXfvsHsKHRe6QWUkQbTVQMlCRTHEieK12god97uq8Cr5PKG/w5wFdABXOic2x1Qe
S4E3uf9vhX4LrDZ277NOeeeAH5rZm8xsi5zwzu30tm8DLgIeCKjcIlKHsOf
2ijSC4ljiRPEaDeU64ic6594OYGZfAl4BjnfOHWrQazvgITNzzvgITNzwBedc3cARzvnXgJwzr1kZou9+x4Dv
FDw2P3etnLb9xfZPoOZbSA3as7xxx/iPckEog4xGq9aduSMt+61dPXxSFWg5DNutz+7kyDg8MTrbf
v46ZVY7WQc47cGGfh74rZZio3R3wi/4tzbgr4SQM74QDvcc6dTm7ayWVm9t4y9y0WV3CTaWipdz2mRucu
8M51+uc6120aJGfMouEIuqxmk/bdunWfk7+7ANcurWfodFxstkjql2i6XOIfqwGIbffx7h0m7fft/U
zPDrOocMTLbXv46YVY7XQ1FSWodFxNmx7nJM/+wAbtj3O0Og4U1PZsIvWUsoNO73DzF7zfjfAf
c45t6CeF3z3bj9vej9PGBm3yQ3Q54d99PLp953a83rHAi/+7FF7h9pS6++L+w
iSMyFNRqrtG05+hxaTzbrGB2fpKe7ky1rTuG2RwfZ8aMXueqeAf7kw28nnU5p30skzP5+SAGb7p6Zv
nDT3bu5Y90ZzFf6wqYpl74wsNn6ZtYNpJxzh7zfzwM+D+wA1gPXez1/5T1md6zcZxXgnt8N8bjn0oP2v2uSKHtaA3iSEZ8YGSdW+AlqgPaZ/9T1kB3C5md1NbmHmz73O+oP
AHxcs0DwPuMY5N2xmh8zs3cAuYB3wl0G9H5EoCPNiIkrblqPPobUUq3M3XLwCgPuffInjerpQvi6Jg
uLfDys5ekHnjPs9tm+Ybh04NlVYhzxHA//bzH4E/Atwn3Pu2+Q64QCG8vgR7gXO9vgPvJZWsZBO4Efg/
AW6T5h8Bj3u3z+YWbwO+Su/DQIPBjtFBTEq5wNHYy66ZHYzMTwaeiUhqsHH0OraVYndu8fYDLzl7Gm
Ut7eGE4o30vkVD8+2E3n/nAyTPud+bSHkbHJkMqZWsK5bDHObcXeEeR7R7UPAOUW2O+CyEs91F3BXke3
9wKl1F1YkJsIcjdXl3nP0ObSWUnvu2eJ53HjCuZ3tmnfSySUitXjF3Zx1okLp9urWz66UjHbZDr/I
JIQ+dHY/Hw/eGM0Nsg5qvl5hwvndXDHujPo6kjz+kS2JTNGKB1YchVbf5GZKFXnJunp7mBOm/a9NF9
VsTo2yR3rzqC7s43RsUm62tO6vH2T6dMWSYgwLs4O0vIhm2PMzw60dKdz3waxpR5P1v0c0iSUtlw5
ralita57o42ujq076X5qo3Vro425s9pJ2XG/Dnt6oSHQCPiIgkRxmissoRIKygX5zoDIlGiWI0ffVO
KJEizL4qjLCHSCsrFef7MB8T7QlSSDIrV+NE5CBGpmbKipnlAAAAgAElEQVSESCtQnEtcKFbjRx1xE
alZGPPSRZpNcS5xoViNH52bEJGaKUuItALFucSFYjV+1BEXkbo0e166SBgU5xIXitV40bkJEaoNF40R6Shll69X2
xfv59118Q6PNLMIrlzC0c/an0fxE/ysWRYkyiQu1h/GiOuIg0RDbrGBmbJOu8n1nXlNcsljM3/9qV/
i9SSTbrGDk8CQavHBrjiq/tnhFHijGJiqmpLCNjk8xtT7Pn5RHu+v5etYcxoI64iNQtrAa+MGfuZNZ
N58zNTEz5+r9IOdNxvS0X19fc+yRXnLucRfM7p+NIMSZRkM06hjLjfOqrj7P82gfYsuNpLjrtWPp2P
af2MOLUEReRuoXVwFfKY64851KPPYnG9efsAl529bDqOFGMSBZmJKTb17T4iYsT2d5u4idT4ifVledukTtYcSpIy4idau
mgW/kFJZKOXOVU1eqMTs257anisb1ssXzpuNIMSZhKYxXHBy9oHPG//OxqvYw2tQRF5G6+W3gZpEV/3
VLO3Gbl1A1jfnySRWm9wcb3L5txvzOX9vDCcGY6jpS3WZplRr04PMmhwxNvxOu2fq5ctZw173jL9P3
PXNrD6Nhk09tDqY6ypohI3fIN/Ma+J3hs3zBnLu0p2sAX5Vy5jYjp26+Azf7v
S/s7lA2ghqE9XkWi81Nfbv54m+dwc69wwVlWUl3Zxtz2t6II+VtlqAVqxc3XrKCRfM7p6eiXHXPAH/
y4bdz/5MvcebSHm5Zu5ujua2h1I9dcRFpO6UVn4b+CDmKFbKmRt0Tt1GH1y0urA+z1KxOW9O2xFxn
S9n4TblbZYgFasXV90zwJY1p7DjRy8Cux8fmEXz153PpmxKVIpsJQx4o2Kp1KmHOMRpKkpIi2uUdN
F8g18yryfRTrySZyjqAVQjRXW51kuNgvjGlAKOGm6UvVi2eJ5039Pt6UOXp+Y4pNGgTriIum
RlPkjhHMYkHF2EK6/P0G5tKASdhKFUvXhjOHBGvitF40XkJkRbXzBHIJM5R9Ds/XvwJ6/MMc3qVSCX
F60VuvcKz150/I14Vo/GijrhIi8uPtOTnHsLMU/KN1uw5kFf0jmJBxdhmv15jo5N5j7XieA/Vz+x2
ez6IgJF2hlvDvic9iPbHMVovGhqikiLS+LS9y7LdbaeTOBtRA9Xpkrm13OjDsWLrPs3SPZ66u97Z2
6ItE2faBYYQ64YjRedGgk0uKSPKKrjCbxFdV9l+T6IvFQqW4oRuNF30QiktiUVporGV9R3ndJrS8SD
37qhmI0PjQ1RUQSSxlN4Y7kUdcRP44Y1xdUh3skvUdcRBJLcylpHuWh5Tjq
dcRFJNM2VjC/tO5HiVDeSQ1NTRERERERCoI64iIiIiEgI1BEXEREREQlBYjviZrbazJ4Ezuzrs8
og0SzbrGBmbJOu8nyFfiTAo5d5nq3wGcTY1leXQ4QmyznHo8ARTU9mwixRZiud48rPfqtm3ioNkSuQ
MfzNLA7cB5wL7gcfMbIdz7t/CLZlIsPKdzfk5Vpy5tGeuhzT9Y5VzAgT5FxpuWVo7nAF2Rx5HhOX+q+u9
eJ5jIxN0t2Rxsyq+gyWUdmYkrZB5poairLUGacTX27p/fRLWtXsrCrg3T6yPGhVt5HrVKno6yW+Cu
133q62nl9Mpt7rrEpprJZPv03/1px3yoOkiupI+LvBAadc3udFCSJFx9ezLrpSx9nJ
pJ1oYfMxBR9u57jotOOZcuOp1l+7QN86quPM5Qz5/Ck/88g/+V26dZ+Tv7sA1y6tZ+h0XsMzH
Fpr7dM/bRpr7d2kdFtEqdjqpa46/Ufhsdn3rjubb1c2hskkXzyvuW8VBciW1I34Mrf9zr8ELB3
uXLgjdSV0eaVacuYfP2gSM6c9ksvj8DfbmFo7uzreg+6i6Shq3V91Gr1OmoqjX+Su237T0
DXHb2shn3KbZdJB+3ccwn6uaOOuPvPCFsYPIrXxG6utcunjzPgUyxbPK/7BX9K/j7WOD
Rnoy63x/MRqZqxEnI5pH83WKnU6DH5itdb4K7XfBg+MHPFcyxbPm3GfWdJmvWcZBcS2/J
```

F2Xdyzt3hnOt1zvUuWrSoaYUTqZbfWG2VSx93tacZHZss2Znz+xnoy63x/MRqyuDGS1bM2Ec3XrKCY
lNdW30ftUqdDoOfWK01/ortt1vWruTBp1464rleGM5U3LeKg+Qy55I3z87M2oBngXOAnwKPAf/ZOfd
0qcf09va6/v7+ss+79Or7GllMiZh911/g526hr4qpFKutsrCt2IK//OIlwNdnkPAFUKG/gVKxms3mM
qW8mpnguJ4uXhjOcFRXO/PntGuRWhEtUKdDfzPlYrXW+Ju93+a2pRjOTMx6rpV0d7Yxp73yvm2BOIi
Dhn/gicya4pybNLPLgQeBNHBXuU64SJK0yqWP0+kUb+7u5M71vUW/mPx8BqmUsbC7o+RzSDBSKWP+n
HbS6RRm8Ob5nSU/d+2j1qnTUVRP/BXbb+Weq9K+VRwkU2L3pHPufuD+sMshIsFpxBeTvtzCUc3nrn0
kYWpk/CmWZbakzhEXEREREYk0dcRFREREKgjriIiIiISAjUERcRERERCYE64iIiIiIiIUhkHvFam
NlB4LkKd3sz8EoTiuNX1MoD0StTo8vzinNudQOfr2o+YjVq+8CPOJYZol3uOMQqRPszrFZS3kuz30d
cYrUVJCWGG6HYZ9HwWFVHvApm1u+c6w27HHlRKw9Er0xRK08zxPE9x7HMEN9yR0mSPsOkvJekvA+pn
vb9G5r1WWhqioiIiIhICNQRFxEREREJgTri1bkj7ALMErXyQPTKFLXyNEMc33McywzxLXeUJOkzTMp
7Scr7kOpp37+hKZ+F5oiLiIiIiIRAI+IiIiIiIiFQR1xEREREJATqiIuIiIiIhEAdcRERERGREKgjL
iIiIiISAnXERURERERCoI64iIiIiEgI1BEXEREREQmBOuIiIiIiIiFQR1xEREREJATqiIuIiIiIhEA
dcRERERGREKgjLiIiIiISAnXERURERERCoI64Z/Xq1Q7QTbdKt9ApVnXzeQudYlU3n7fQKVZ183lrO
HXEPa+88krYRRDxRbEqcaFYlbhQrEpY1BEXEREREQmBOuIiIiIiIiFQR1xEREREJATqiIuIiIiIhEA
dcRERERGREKgj3gKyWcfI2CRZ5/3MBpKBR0Qk8tQeipSm+tF8bWEXQIKVzTqGRsfZ2PcEj+0b5sylP
dy69jQWdneQSlnYxRNpiKVX31fV/fddf0FAJZEoU3soUprqRzg0Ip5wmYkpNvY9wc69Q0xmHTv3DrG
x7wkyE1NhF01EpKnUHoqUpvoRDnXEE66rI81j+4ZnbHts3zBdHemQSiQiEg61hyKlqX6EQx3xhMuMT
3Hm0p4Z285c2kNmXEe4ItJa1B6KlKb6EQ51xBOuqz3NrWtP46wTF9KWMs46cSG3rj2NrnYd4YpIa1F
7KFKa6kc4tFgz4VIpY2F3B3eu76WrI01mfIqu9rQWXohIy1F7KFKa6kc4AhsRN7O7zOyAmT1VsO1rZ
rbbu+0zs93e9qVm9nrB/75Q8JgzzOxJMxs0s1vNzLztPWb2sJnt8X4e5W03736DZjZgZqcH9R7jIpU
y5nW2kTLvpyqViLQotYcipal+NF+QU1O+Aqwu3OCc+4hzbqVzbiWwHbi34N8/zv/POffpgu23AxuAk
7xb/jmvBh5xzp0EPOL9DXB+wX03eI8XEREREYmUwDrizrnvAcPF/ueNav8G0FfuOcxsCbDAObfTOee
AbcBF3r8vBLZ6v2+dtX2by/kh8CbveUREREREIiOsxZq/DLzsnNtTsO0EM3vCzP7JzH7Z23YMsL/gP
vu9bQBHO+deAvB+Li54zAslHiMiIiIiEglhdcTXMnM0/CXgeOfcacAVwP8yswVAsclJla636vsxZrb
BzPrNrP/gwYM+ii0SDsWqxIViVeJCsSpR0PSOuJm1AR8Gvpbf5pwbc84Neb8/DvwYOJncaPaxBQ8/F
njR+/3l/JQT7+cBb/t+4LgSj5nBOXeHc67XOde7aNGiet+aSGAUqxIXilWJC8WqREEY6Qs/APwf59z
0lBMzWwQMO+emzOxEcgst9zrnhs3skJm9G9gFrAP+0nvYDmA9cL3381sF2y83s7uBdwE/z09hERGpx
dKr76vq/vuuvyCgkoiISJIEmb6wD9gJLDez/Wb2Se9fH+XIRZrvBQbM7EfAN4BPO+fyCz1/F/gSMEh
upPwBb/v1wLlmtgc41/sb4H5gr3f/O4Hfa/R7ExERERGpV2Aj4s65tSW2f7zItu3k0hkWu38/cGqR7
UPAOUW2O+CyKosrIiIiItJUusS9iIiIiEgI1BEXEREREQmBOuIiIiIiIiFQR1xEREREJATqiIuIiIi
IhEAdcRERERGREKgjLiIiIiISAnXERURERERCoI64iIiIiEgI1BGPkWzWMTI2SdZ5P7Mu7CKJiIhHb
bTEjWI2fIFd4l4aK5t1DI2Os7HvCR7bN8yZS3u4de1pLOzuIJWysItXs2zWkZmYoqsjTWZ8iq72dKz
fj4i0plraaLV/EqZm9CsU45VpRDwmMhNTbOx7gp17h5jMOnbuHWJj3xNkJqbCLlrN8o3ApVv7OfmzD
3Dp1n6GRsd1RC4isVNtG632T8IWdL9CMe6POuJN0IhTP10daR7bNzxj22P7hunqSDeqmE2XxIMLEal
fHE+XV9tGq/2TajW6XgTdr1CM+6OOeMAadUSYGZ/izKU9M7adubSHzHjjjlyLVfAgvxCTeHAhIv6Ua
3PiOIpWbRtdbfsXx4MTaZwg6kU1MVsp/or9X9/x/qgjHrBGHRF2tae5de1pnHXiQtpSxlknLuTWtaf
R1e6j0T48SWa8fAUqVsGnprKBfiEGfXAhItFUrlNRb5tZrEPQjE5stW10tZ2gOB6cSOMEMbpcLGZvW
buSOW2pqg6OS/3/8IS+4/1QRzxgjToiTKWMhd0d3Lm+l2evO5871/eWXFBxRKXY1s/w6DhXfG130Qa
8XAUP8rRStV9cjaYRJpFwlGtb6mkzi3cIxjh0eCLwTmw1bTRU1/5F8RS/2s/mCmJ0OZUyerra+eJvn
cEzf3Q+W9acwt27nufFnx3mru/v9X1wXOr/2Syhfsf7EYU4DixripndBXwIOOCcO9XbtgW4FDjo3e0
PnHP3e/+7BvgkMAVsdM496G1fDdwCpIEvOeeu97afANwN9AD/CvyWc27czDqBbcAZwBDwEefcvqDeZ
yX5UY+de4emt+WPCOd1Vvfxp1I2/Zhyjy2sFAA79w5x1T0DbFlzCqv+4nts7HuCO9f3Tj9HqQre3dk
W6Gmlwi+uZq+oTmoWGpE4KNepqKfNLNb2bezbzZ98+O2zts1sAxvFbxudv6/f9i9qp/jVfjZfI/sSh
V6fzPKprz4+43l37h1my5pTputJpfgr+f/ONF0d6VC+4/2IShwHOSL+FWB1ke1/7pxb6d3ynfC3AR8
FTvEe8z/NLG1maeA24HzgbcBa774AN3jPdRLwKrlOPN7PV51zy4A/9+4XmjBGfUtVimWL503/XtiAl
zpFOjo2GfhppfwXV8q8n00K/iiOMIm0inLTMuppM0u1fcf1dB2xLQrzVP22f1Gbxqf2s/mC6kuU6y/
MPjguVBh/5f4f1ne8H1GJ48A64s657wHDFe+YcyFwt3NuzDn3E2AQeKd3G3TO7XXOjZMbAb/QzAx4P
/AN7/FbgYsKnmur9/s3gHO8+4ei2tOVjVCqUgweGJn+vbABL1fBgz6ICOu0UNRGmERaSbm2pZ42s1T
b98Jw5ohtcZqnWmou79z2VCin09V+Nl9QfYly/QW/B8eN7is0q18QlTiueD7DzBaRm06ytPD+zrlP1
Pial5vZOqAf+P+dc68CxwA/LLjPfm8bwAuztr8LWAj8zDk3WeT+x+Qf45ybNLOfe/d/pcby1q2a05W
NkK8UhadbbrxkBTc9+EzRClLuFGmQU0fCPC0U1Gk+EamsUttSa5tZrO27de1KOtIpzjpx4Yx2JkrzV
CuZ/XmNHJ7kKz/4Cbf+42Aop9PVfoYjiL5EsTpzw8Ur+Lsn9hc9OC5VXxvVV2hmvyAqcWzOlT/SMLN
/Br4PPE5u/jYAzrntFZ/cbCnwDwVzxI8m1yF2wB8CS5xznzCz24Cdzrm/8e73ZeB+ciP2q5q5xzv+Nt/
y1yo+Sf9+6/zNt+HHC/c+7tZva095j93v9+DLzTOffGJ/1G+TYAGwCOP/74M5577rlKbyk2ZlzNamy
KVArmtEdrjtbI2CSXbu2fUQnOOnFhybmbjbxCVx2VPZQPLsmx2ghLr76vqvvvu/6CSD1/QFoyVou1E
4CvbVFoFyuptt0MQgCdpZaM1agorDOjY5N0daR5fSIbSp0oGd/resGIwmBgwz8QP7W2yzm3uREv5px
7Of+7md0J/IP3537guIK7Hgu86P1ebPsrwJvMrM0bFS+8f/659ptZG/ALlJgi45y7A7gDoLe3N1FLv
mccOc95YzdHabSi0mmh2QcTU9ksn/6bf21Iwx/mQtFaJDlWJVnCjtVSo4aF26KySKsW5drNkbHJprR
jcWs/Swk7VqOisM7Mn9MOwLzOcJLqlVv4+fxQhqO62pk/p70hsRaVOPbzSf+DmX2wES9mZksK/vw14
Cnv9x3AR82s08uGchLwL8BjwElmdoKZdZBb0LnD5YbxHwV+3Xv8euBbBc+13vv914F/dJWG/aVqjZj

DVW6BR7EUjIfGJlk0v7NhiyqivIhERIITlUVa5ZRqY0u1m3teHmlqfnG1nzJbkP2CPS+PcM29T3Job
JLDk/FPGDGjDD7us4lcZ/ywmR3ybq9VepCZ9QE7geVmtt/MPgn8qZk9aWYDwNnA7wM4554Gvg78G/B
t4DLn3JQ32n058CDw78DXvfsCbAauMLNBcnPAv+xt/fYsmqpOI26uES5BR7Fviivu
meAy85eNv14LQ4SkVoUjrqtecdbePAz7+Vvfudd4IhEPuxybWyxdvOGi1dw26ODkTygkHiqtm8QZL+
gML6vumeAbLaedxY9FecpOOfm1/LEzrm1RTZ/uci2/P2vA64rsv1+cvPFZ2/fS26++Ozth4FLqipsi
6r19GzxXL3V5+Utd1qoUgpG0OIgEalNftRt0fxOrjxvOZu3D0RqikqlNraw3dzz8gg3PfQMO36Um52
pAQqpVy19g6D6BUXjuzNZ8e1rEpCZrTGzm7zbh4IulDRHradnG5nyp9RpoXJpyKJ6hS4RiYf8qNsV5
57M5u0DkZuiUqmNzbebmfEptux4erqTAvFLzSjRU0vfIIh+QWasRHyPJSu+K3bEzex6ctNT/s27bfK
2SczVWnGacXGJ4tNWVrJ4QWfT8rGLSDLlR92OX9gViTzCs/ltY8O4YJwkXy19gyD6BV0daW5du/KIf
kDY9bPR/Jwv+CCw0jmXBTCzrcATaO51LJRL+VdrDs3iuXob2/hXWs2s6SgiUo9Uyhjxrh58RBs4NjU
j21Sz+W1jo5L1QZKlZN+gTL0Iol+Qi+/OxMe335bmTbyRAvAXAiqLNFileV61VpxmNf7NvhCSiLSWX
Bu4ko19u2dc/GwqmyWbdaF94VfTxqqdlEarpV4E1S9ohfj2867+BHjCzB4ll8j8vcA1gZZKGqLS4ol
6Kk4rVA4RSbZUyujubONPPvx2juvpYvDACH/67Wc4eGisqRfIKVU2tbEShlrrhWK2Nn6ypvSZ2XeBM
8l1xDc75/5v0AWT+vmZ56WK09grdopIvMxpT/OBm/+JyYI0a21e5iapjdrU+GuVehGFWC25WNPM/qP
383RgCbkrVr4AvMXb1jIakaQ+DEEsnojrZ1FKo3Kfikh1otKWNGPxeTFRef+Npja1vLjs90bWi6i+5
6jEarmsKVd4P/+syO2mgMsVGVHZUbVo9Ir6OH8WpcThCnsiSROltiSMzCNRev+Npja1tDjt90bViyi
/56jEasm5CM65Dd6v53sXyZlmZnMCLVWENCpJfRgavXgizp9FKY3MfSoi/kSpLQkj80iU3n+jqU0tL
U77vVH1IsrvOSqx6ueCPv/sc1siRWVH1arUBXNqEffPopiwTkuLtLKotSWNbCf9iNr7byS1qaXFbb8
3ol5E+T1HJVbLzRH/D2Z2BjDXzE4zs9O92/uArqaVMGRR2VFRkMTPQhfEEGm+JLYl1Ujy+1ebWlqS9
3spUX7PUYnVcucFVgEfB44Fbi7Yfgj4gwDLFCnNuHhNXCTxs9AFMUSaL4ltSTWS/P7VppaW5P1eSpT
fc1Ritdwc8a3AVjO72Dm3vYllipSo7KhaNDotT5w/i3KUwlGkuWppS6KQZqxRktqW5qlNLS6K+z3oe
hXF91woCrHqJ4/4djO7ADgFmFOw/fNBFixKorCjqlXpqpq1iuNnISLRU01bElR7Fia1pa0pSvu9WfU
qSu85iiou1jSzLwAfAf4LuQv6XAK8NeByxVoUcmaWS8tTb/mi8P5EJPnybQ0Go2OTLJrfmYiUeM1qQ
9VWJ1cj9m2Q/YRmlD8p/Bya/JJzboWZDTjn/oeZ/Rlwb9AFiyu/R5hBnw4qt1J5aKT2I+AkjkyJSPQ
Ua2tuuHgFADt+9GJkMi9Ade15s9pQtdXJVc++LYxVHBy9oHPG/xvRTwiy/EnkJ31hPod4xszeAkwAJ
1R6kJndZWYHzOypgm03mtn/MbMBM/ummb3J277UzF43s93e7QsFjznDzJ40s0Ezu9XMzNveY2YPm9k
e7+dR3nbz7jfovU5TrwLqJ0F8MxLcl1qpPDo2WVcC+6gkwBeRZCvW1mzePsBlZy8DopN5odr2vFltq
Nrq5Kp13x4Rq9v6uXLVcta84y3T92lEPyGo8ieVn47433sd5huBfwX2AX0+HvcVYPWsbQ8DpzrnVgD
PAtcU/O/HIzuzENHHeDzinDsJeMT7T7G+D8gvtu8B5fN7+nUfzkzGxGEJZMy1NnT
s8o5wQVkfio1KaWamuWLZ4XqZR41bbnzWpD1VYnT77OdHWk2bLmlBkdaD/7tlisXnXPAFece3JD+wm
VKDZnKtsRN7MUuc7uz7zMKW8F/qNz7r9XemLn3PeA4VnbHOLOTXp//pBcasRyr78DWOcMK66XuKAa
CLv3xcCW73ft87avs3l/BB4k/c8NatmxMNPzsxmBGHhSuVnrzufO9f3srC7g9cnsnXl9IxyTlARiQc
/bWqptub18anp9iwKp7Grbc+b1YaqrU6W2XVmy46nufK8N0az/ezbUrF6/MKuhvYTKlFszlR2jrhzL
uvNCT/L+3sMGGvQa38C+FrB3yeY2RPAa8C1zrnvA8cCzsrnvA8cA+wvus9/bBnC0c+4lr/dvB5zZlPZ2Xx5E3//hMFEuuxbWPwPg5sKj36TLXjCgIGzr51BERaNUVVgrHDhsiRS8uKnnZdDNvqHLGiTNI7Pg6uybGpu2OMgzJakQo8jVkWsG7HqhI1Fd4LD1QOozIaiTEkbe
uvNCT/L+3sMGGvQa38C+FrB3yeY2RPAa8C1zrnvA8cA+wvus9/bBnC0c+4lr1wvmdlib/sxwAtFHvN
SrQWt5hKtfnJm5oMw/3zwRhA2ckVxsZXKxcu3kpTlKnqlL7Z6c4ImKQWZ+Lf06vvCLoJEiJ82tWRb0
xGtNqPa9nz2+9r4/mV8/D0n0NWRzo12Vphf7rf9jHL+ZqlesTqzefsAW9acwsFDY772baVYLVf3blm
7krntqYoxmlcuVhWbM/np9T1kZhcD93qj0nUzs88Ck8DfepteAo53zg15V/P8OzM7hVY3
8xsA7npKxx//PEln7CaEQ8/OTPDDMLp8q3rpaszzfNDGa677995+bUxblm7cvpouFRFqycnqBZo1M5
vrIqEzU+s+mlTo55/OK/a9rzwfc1tTzE0Os5XfvATVp26hGWL5zEyNkl3R5p0euYJ62rbz7h8fmGKU
7taqs6cdPS83D72sW/9xurs2Bk5PMlXfvATbv3HwYpxl826XKx1pnnl0Bh/8Z1nefm1sRmPUWzO5Ge
O+BXAPcCYmb1mZofM7LVaX9DM3gNcYa64z3xcvGhxEtUhLFEgsb1/dOHiVUhoiQJvCZeN7NQKUHsdQ0AbC36
s3DvEpr7dDB4YrbjYqNL7K0ULNGrnN1ZFwuYnVv22qbW2Nc1US3uef1+vT2S5e9fzXHTasWzZ8TTLr
32AT331cYYyR7a/tbSfcfj8whSndrVcnfG7b6uJ1XzsZMan+NRXH+fm7+ypGHfT02e25abPXHPvk1x
x7nIWze884jGKzTdU7Ig75+Y751LOuQ7n3ALv7wW1vJiZrQY2Y2A2ucc5mC7Y/QaggpXmb+6YYq73IHFfHTB3cKkRaS7nNX7T7N9gpKRtxVa8Gy+ROIZG0wIiNFD4VT6DPYwoT
8zs3V62lHXAt7yH7QDWe7+vn7V9nZc95d3Az/NTWGpVcuFjHSPYYQdhuYVQQXWQtUBDRCCYNjVMtbb
nXR1pVp26hM3bB2Z0sDf17T6i/VX72doaVWeqjdVq4q5cpiPFamkVp6aY2SPOuXMqbSvyuD7gfcCbz
Ww/8DlyWVI6gYe9LIQ/9DKkvBf4vJlNAlPAp51z+T3/u+QysMwFHvBuAuANcDXzezTwLPk7vQEMD9wAe
BQSAD/Hal91hJEKdRwp4rXWqu2OCBESCYBj7IufFhf54i4l8cT00H0cZkxqdYtnier45OEO2n2s348
FtnGr1Pq4m7cgN8QayDq0aUY73kJ2Jmc4Auch3po3hj7vUC4C2lHpfnnFtbZPOXS9x3O7C9xP/6gVO
LbB8CjjgY8Ka7XFapfNVq5CVaozBXuthcsRsuXsFNDz0DBLN4NKi58VH4PEWkOnG67HVQbUxXe26Bp
p+OTqPbT7Wb8VOpzgSxT6uJu1Kd9heGM6Ge8Yp6rFup9Zdmtgn4DLlO9095oyP+GnCnc+6vmlLCJun
t7XX9/f1F/9foI6mRsUku3do/I1jPOnFh0SwsQSp8X9UsxmjUazbqqLTJn2fotXkx0MsAACAASURBV
LZcrEZV1LKm7Lv+gqruX235q33+gCQmVsMezQqyjZmayjKUGWdT3+6KnYRGfg5R+R7yRDpWw44/v4L
ap37ff/EO70q6O9uY0xbeZ9bgz6Xhb6JkCZxztwC3mNl/cc79OVwbgWaAgMNtiRtg5Zb7g3HbBQ4Ci58VnH4PEWkOnG67HVQbUxXe26Bp
p+fijP76PoTv3wil59zUqANTRCjYJo7KZJctYxmNbrjFGQbk06neHN3Z9kpBzMuSw7g6m8/1W76E/R
oaiNjtZ59Wq4cfr+3ozrlLOqx7qcn+X/NbD6AmV1rZvc2+7LxYSq2+GBT325Gx6dqviR90Mnsq73kM
vhfwOH3CqPNpIsDiCRXtdlCamn/KpYh4DamXPubez9js97PWN1tr9pNf4LM9tXoWK11nzaqHFE9cxD
1WPfTEf9vzrlDZvafgFXkrmLZkMvGx0GpI6nuzraaK2LQGQOqaTiq6VgH8QXXCEnLwCAib6j6ypUBd
JzCbGMy41Ns7Ns96/3sJjM+WfnBZajd9CfI0dRGx2qt+7TWcszoPxye5NDhicj1DyD6se7n3FZ+T1w

```
A3O6c+5aZbQmuSNFSLrvISUfPq+k5gz5947fhqPaUWzVXGG2mqJ4OE5H6VX3lygA6TmG2MV2dJd5PZ
5uvqyGXonbTnyCzfTU6Vmvdp7WUo1j/4cZLVrBofueMznzY/QOIfqz7GRH/qZl9EfgN4H4z6/T5uET
oak9zy9qVM46kbrh4BQ8+9VJdpzWCzCPu9zRMtUfBUZ5nFXZedhEJRrWjWUGdhg6rjcmMFX8/zw9l6
p4eoXazsiBHU4OI1Vr2aS3lKNZ/uOqeXM7wvKj0DyDase6nQ/0bwIPAaufcz4Ae4KpASxUhqZSxsKu
DL/7WGTzzR+ezZc0p/N0T+1n7rrdG5rTGbH4bjqpP+VaorFGcPy4i8VbtlSujdhq63nYxlYJbiwwG/
cV3no1MJyfJgrwSdlRitZZylMsZnuf3oKLV+w7l8ogvcM69BswBvutt6wHGgHjlTqtTOp1iXqeRmZj
ipKPnccxRJ0bqtMZsfk/DVH3Kt0w+0ajn6RSR+Kom21KUTkM3ol2c05bGOfiTD7+d43q6GDwwwk0PP
cPBQ2OhXiCllQSV8z4qsVpLOcrlDG9Lme889+o7lB8R/1/ez8fJdbwfL7i1VEccon1aoxg/5a32KLj
cyEC9i05a/YhYRBonKu11IxbjpVLGnLYUXZ1pPvalXVxw6/c5eGiMW9auZG5by8wSTayoxGq15Sjef
1jJ4gWdVZ05KFlHxqdapi9QLo/4h7yfJzSvONJMtRwFlxoZqDd/aasfEYtI8jRqXU06nZqeItnd2cb
ggRHu3vU8a9/1VrWTEopK/Qe/Zw5K1ZG5HbkDz1boC/g6nDazY8zsl8zsvflb0AWT5mjU0Xg9i06Cz
NMqIhKWRi7Ge30yy6e++ji/+Af3s+ovvsfN39mjdlJC1Yj+Q6k6MnhgpGX6AhU74mZ2A/AD4FpyizS
vAq4MuFzSYEFP/ahn0UmUs7GIiNSqkYvxujrSHL2gkwc/815+/Mcf5MHPvJejF3SqnZRAhdF3uOHiF
dz26CDQGn0BP+cOLgKWO+fGgi6MBKMZUz/qWXQSZJ5WEZGwNHIx3uGJKa5ctZyr7hmYkbf58MQUXR1
qJ6Xxwug7PD+U4aaHnmHHj14EWqMv4Gdqyl6gPeiCSO0qHbE2a+pHraepopLCSUSkXrPbY6Ah0/+yW
bjqnoEj8jZns40svbSacv2HZvcdcNDd2cbBQ2Mt1Rfwc4iRAXab2SPkUhcC4JzbGFipxDc/R6xRn/o
RlRROIiL1CHIEsfQVNqPRjkv8VIrXZvcdWrUv4GdEfAfwh8A/MzOFoUSAnyPWoK4010hBpXBSWkQRa
ZYgRxCb2Y6r3WwNleI1jL5DM9M5RiXOK3bEnXNbi938PLmZ3WVmB8zsqYJtPWb2sJnt8X4e5W03M7v
VzAbNbMDMTi94zHrv/nvMbH3B9jPM7EnvMbeamZV7jSTyc8TaqlM/8kf7l27t5+TPPsClW/sZGh3Xl
4qIBCLIEcRmteNqN1tHpXhNct8hSnFesiNuZl/3fj7pdYxn3Hw+/1eA1bO2XQ084pw7CXjE+xvgfOA
k77YBuN17/R7gc8C7gHcCnyvoWN/u3Tf/uNUVXiNx/ByxVrpEb1SOCkuptXxKiygifjWiHQxyBLHeS
637fX9qN1tHpXhNpYyernbuWHcGz153PnesO4OervamTxUJoo8SpTgvNyK+yfv5IeBXi9wqcs59Dxi
etflCID+ivpVcVpb89m0u54fAm8xsCbAKeNg5N+ycexV4GFjt/W+Bc26nc84B22Y9V7V7HXSBy/R6ylT
vdE6aiwmHrKF/W58SISDY1qB4MeQaz1tH0170/tZuuoFK/ZrGM4M8GGbY9z8mcfYMO2xxnOTDS1fxB
UHyVKcV6yI+6ce8n7+Zxz7jngVeBQwa1WRxc890vAYm/7McALBffb9tgZv31m1
n/w4ME63lJ46h0lidJRYTH1lC8Oc+P9SkKsSmuIY6w2qh2stz0OSjXvL0ntZiVxjNVGqhSvUegfBFW
GKMW5nwv6fMrMXgYGeGOhZn8AZSnWUrkatvvmnLvDOdfrnOtdtGhbRQ+tWpDTP+pZ3BClo8Ji6ilfk
ua3NTNWReoRx1jNtzNr3vGW6QvmbFlzCnPbfV18eoZmLjbzq5p2NEntZiVxjNVCjehXlIvXKPQPgip
DlOLcT/rCK4FTnHOvNOg1XzazJc65J1c65l7zpzpJQe87fuB4wrudyzworf9fbO2f9fbfmyR+5d7jVA0Iyl+r
aJ+MZ16yteqqZBEpDqZ8Sk2vn8ZF512LJu3v3HBnFvWruTN3Z2xbzOqaUfVbsZDM/oVUegfBFWGKMW
5n8P9H5PLJd4oO4B85pP1wLcKtq/zsqe8G/i5N5m3X91PVKulnPVew1Q
hGF0zulROmosJh6yxfF0SkRiZau9jQff88JbN4+84I5m/p2R6Kdrle17ajazehrRr8iCv2DIMsQlTj
3czhxDfDPZraLKi/oY2Z95Eaz32xm+8llP7ke+LqZfRJ4HrjEu/v9/9XMi/zQ/Q/9/2XmfYzP4QeMy73
+edc/nzFL9LLjPLXOAB70aZ1whFFE7vlBKlo8Jiol4+EYm/VMqYN6ctsu10vdSOJk8z+hVRiJsolCF
ofjriXwT+EXgSqOpius65tSX+dMpw85lXCOZ4HoTS4e9Wt2j+lgzKxcyaG1Ii3D9/gzLJp2fpDUY7
09E4i/q7XS91I4mS7PiNQpxE4UyBMnP1JRJ59wVzrm/rvaCPpIThdM7QQsrF3nUc6CLSDw0s51Weyn
1Skq/op6YTEo8+zm0eNTMNgB/z8ypKbPzg0sJST+1EtZi1CgvghWReGlWO632UhohCf2KemIySfHsZ
0T8P+PNEyfY9IWJFpVFAUEIazFqlBfBikj8NKOdVnspjRL3fkVd1wlJUDxXHBF3zp3QjIIwDC2WvmMx8
EvkMqcAnE3ugjr3BlcsiZP8opHZc7WCXjQS1utK/C29+r6wiyAtSu2lSE49MZmkeC7ZEXfO/TaAmf0
D8DbvAjp4V6q8rTnFk0LZrCMzMRW5hRlhLRpJwmIVEUkGv+2z2kuJkjD7FfXEZJLi2U/WlKX5TrjnZ
WB5QOVpCbUEftRXCIeV5zPp+UVFJFx+2utq22e1lxKEavsWUehX1BOTSYlnP1lTvmtmD5rZx81sPXA
f8EjA5UqsfOBfurWfkz/7AJdu7Wdwf4ErGrPdSgvlzVkXfTWT8EMHDRxHwl3Fs7+2dpHcPh0vAsqHxOn7Zdrj/rjD/rrBwr1oZu0ywhDHqAmDrFo6gKoDCPwdGnXdN0ocngHC4Q47cHUqe2bXGrw2fcNx85DyD5PuBvwTQI/ieUdIVO5PRajmr2vXQqvjvR+gMgcIvT8NQJ+mo8rPg2YO2fQ+gMgIqIpMJtq2bxkeIy1ZbTo0ozAyvG0MODcJ0xXOnuM5+oT6yrAjXHJzh3lbRuN67FDQXpZwq4cPIwd3q1pPcyDXNDZdF+g3UNIMaJ0e8BZuBOTqWMiBsAx5djlQDVHQjqXPAGhDMV3BZPGkPqsgfj8jRcV47rjrMxmwhLBtmz2LZp7Jg7XGV8mmN2nyxQbUFzCAG1NsgiVFtc2MOjmGi4pw2ehAHnlQ98yTjI2FU+Rb9Sn5XUb4mn+IcwejBHGw6GP0BBzAY4D/ykQv+mlfQcbCZTyupOR6OCgS0vL1hsAoIZFPfJGWsKHpcg1ew3yzYDoqgBzw1XSgrvdPLz2cn+s0Uji+tbFFQnA7GsuFXDZkcHHuWuhAxvJGodPi7cAwuQ9SWdDTFvvo3jnyvyXfXvD0zyz2JYI0ipF1NiLqSRfdSFfGAppGHCqVyE5jYeuHJIaH1kP/Ug+NVQ7nnyq+pMBvPH+9U8Zy8YHD82+BwQ9HhpvB8qM/m/xQfGmp5vYJfA5AVT2wnOW5/L8Y9D8NwnyvOfT87vWpzEGIQmHddZWZW2Gf3zX6V09ReRNqXptlyFvPf5LjYtOh6rmMPp6r26bVEeOIuh8bJgwMX8nXjkU/yyuwcTzRxqRR3uAb3vNfRlKqOeqVpCl6TUTqySZnX0/0yuUzkD3k1kHYs1Io3d92R7fHAnN1Hr0Ozc/0+ueuQEv5nmIbuLszdLvPrGk7kWz7rfsdPY+nHLqJsiWNE4OfkcEqzjvy6rWWKbqcXqi6WlzPZ+ZvFmZV+9pCnD1+XDX0gHq7sXlCs+7PrYDzJsTtRjuqGAmrc+bJqV/Ucfs/wqB8uZtOiW9qxzHCXDBI7OIdH9c+dAMs8Tlja+wk5/gJtl0Ea+Kb+QMmU/ZsbwYrBcezLeGDEGLnOOQqUMXQJ9ALWsiaPuVtNJn4ZNhjB46jQFIi4Onm9X51WnrsZNgY5Hhjk5+8Anb0c+pxfbZ6v40HEmIqeLWZTuv4EBvRELtDE8AQ0FF3B6Ui3D5mQ8EeaqptKnq35Z5mzOb12dUkkHEGfUtbiIAfDm9Gt0LG9o1ssbJiNT1PJGqBpHa2CKg8BdtvDh7pdV/QvdHDNq7fOwYfQ2kPGkbb3XHbsyYLwNHXmqJvVE0iCMZiQwcp8xj/uEYR4Ze4JMGcRPp/SSsTxqsSPzVBgrL/cqjIaKuBVrXtTiNUEuqIvJNtpj/ZU4E45MoeHvBTvVDxUMfbJcZhpeB4Z3DPmFN3jZ6eqCGXgz+TTcakqhRMp4RMmOcWXrXzNWo8L5QLa9WfNNdvsjO7/cTOTq8BdVvFtZeGrmbbGWXTp6RQm0xWdvH4Gfv4O3QZjd7SjPvA3MD7/UKvhIVY2RnbdAkPWLXKmEpZNw7GmwZeKBa4ODNydOCMzfQxj4m9W7e/cTDKSNKNW3L/MX5fS2IO8d6aXsxJeP8UGUBmATq1lTbTCZE6IZoA4dswvpvqZYcwZwFNovmZ0zDHhLnUnDBj3pC27iYNcN+6WVQH3TnYmIutHs8OKjFi3dYWhDrLwIKKXwVtgyC2DmYGGEs9ljBpkMMiaPEwgiRVQx+4bWfg0khJJd5GFK9+tl3MHdcZWP0X8nzmM3g8wqD1mwPT2SdKddz0yqDPGk29/XeXZ8WUr5MNcRbKbYAc3QB7dz8NvX+9tdK0bHhT5kCzz2qPfM2aRVHEErDt/ZiDYBzU0EMzaD1F9LkiyK8JJbDmBGp2RxyHp8HZZxTRwHfhzjJdgQvw2KaD4J2n+gi97L3wmmNQ7fX1EEyc2BjNYmFp+6TFFAeKo1xBwnFcbYiMk5BR5+2lXYjaZS5DN0zLnffPhQFXE3Hq1BkAVNK9wqlxLxU2YaSqGkhJzEHNBQ0aMnG0n7BaeptNbDh+mvXwiuVvXCI6uFrWeOCEvTBn0nS9ShpkDxYOQkGjzDJWg8RoJvJAy55NhKN8E+ytAjAkgK5qIeLGBWmcWgSDD8p8ZT1kVqZ7ZiLvmQO0rvZkI/mWxFiGjGuzTXWq6lzrsXkxL3KF95i+ovG6VZSdGM3hjkNOq9YZujH+qkeNs4nmc2kvl9Fq24gjc89Z7Z9KoFrsUuxSAnwO79g9hzxRMg3NsnybV7Gr/3NMz/m9hm2pO6xSpAfcYDl9M1pV5j77bu9F3Pb3jpkk6wZf+bZ/Ys2gEuGSj5xjJG7OX24Vfq4eEwqQS1Br0MFC0cNOZErhtNRCWFjnxoyKAtr0u5BRMvB5ZPfoJ6TEXUqBzTjGRP4NOo/tHl0RXZvsdLkOJf2Zyj7HcqtEymM/Qtz1tTESswL36EC1nhu+NI1bg+d/kdszvcVKk9KhqGHpSbvmVtQKUmXj5V1fRjTsHyH2FhzC57ugrOFABXbtUuPFJNBLvq9b/LRmTaDqE0ZZtimhBJNK8q1Oo4ab8xewOQhaDTbXiGwn/Ux7uDDHkBb6Xt8xiqWgJRjIyevzuYDxfmiIpXkmDoLhfYE9V6u+7xkAuzsYOqd4N/5bt5ArfyERYc9mQdd7T5T2OM7FBlNAmfYwfjk/sGCJVWhXpGOjzMk0ZBS7SKgLZ6qwaMhmmFmI7Vw9g/5w3jZvpo6vnPyO05hP8QUe1+8+8SJhk0JKXx4e1H6NqSXyhUOyu/Pr2ZsEAjYZ6rJP6mRXv1jjv9MW4ZnMJO7xYJ8mpQFczmJz5UP7w/JuftnBcWO7Lp0UJqH+Qj3DqMYhVJ1S+p4sWr/r4G5QL3+cSoPqZxm/cGpb7wAF5ZZEWyeNGo+66UeJ+EUXLTU+B6c/CG3GKM3PsOn1hqmM64U/MIP05MGzZlRTNQWnRGSkWtBEQyiB8qBVbCwkAyHTO/dC+7vLdrSPN6IqTCMJRHGPCBYDqbPaIX5w9lKYJZz0g9PATcl/C2xgGg2W1hk1SQTdJbgwaqy5Hgx/JHT11BkU7vgaArqYsekrtGC/Tg6OcKP/KPpZ+CK1CUiAH03wJlATPSrOgEz6dcwmQC/pKOIqFhULIcwqg4UYvK89hd/TPGRWnyezCRGVWhHV9zvm8XDs/6cxA87Mc4Ag7YYjmx/oIvkmvWrmbF+aYUZzAJyxV1KPZAcOVIg0gDJRC9YiPgCdLlI7ugM0/B3C9bHcrE2M1G+bf81NxiOA9sRlnWzEsjUz7eu6brrmaaZjV0Q6cgnvRnGnDLqzZH/7dqLUjGnVkKuMvk8iijZ3bHErxF2IgHb4lr6tWAlsohZU/i1QLJ6ZoCgb0wFafjB/6P9X3i5xg6TyUWPVkdqKJU48ay7NfNgoi3nGpj1Ut/HgAihe6oG2L+5e9HsVTVUSp0qaZ22OV9zIGhgLNjFY+yT0YDmh27UrOaUBXw+wuLtqqdVkhW6c+hNkFUmO9hX6gmCYS2RiKyyROLP4mx94L7HkJwUxwLbwW+9MUL7Q2OZJ8NU+NU5RbbWTY6ZBHb3MnX/fcIyLFDmHm9bRB47vD4yzctCA1XOVVYRDU6oN39HCxRbW1kc/vUC82B/aZ4RkfjDCDkqeuqw/f8gGZBBGeIaPzR5z1n2JHxX5SjtCU5gB+2DW9j0bcnHxKSa/Kvl1Yd0a8tR4aiOWD0c4pvMcqHRRZRNwZRs4mIa3vBC5vZNj/Hdt/5s4vcKSsgeBAjoTZK+UwJ9Xsg3T0PMQkwx3sWsWbvUMemWq7MPk6KhLj3dJx4PXgbCkfMR0KVzMGbyaNTwEPAxjyhj+xjwe+gvdz3x7UtDTqTYWW4T5aBmzdAb/i+RCQxFbRNKypvubNRn6MMrBTVuHIhpUyqPGn2CqUSqs5lE6iHKPaYESXTm2uJNW0H0sDgcLH3r5uqgb4v97bS0hZ1JXc3J3p3KvPV8qqkb2pPVWDEmjZAjgfmqGmBoSCYhUjDv6kTlJwjwlHgTwMUWe9gB2ra8xPrVJT9jsL0b5TPWavvWvNt5tCDr7qCqDzSdPLL8T1yWnzUKPkxaLk8OW6Q8+uCxiMZCj8BdzMEvM7V5DhBOcnOjUM2/ZnPYljMs2lx7Q9/WN82X5dGNfWNi2JfdVsYY3J7A4z7lZcD77k2q0K/bkyUFLsTHS1t48g05yK97EdNUY1cAUcH1Em4ZN9Lr8KGRMP3fGuGg5Ag+/t5QY7iuH6a6ks2WMTMFXFp9/Uh7o8bEYVvzYy/8ewVQXPcm3aIOFDXHKP2ZZ0wPgpxVfV1eNY5PRwxIlJe+K6vHoRu0zLcTEpJg8dNhOC4/5p4DN7XGQOHP0Ewl/j/lkJnn1kSYkmhFHNVlDVrbnu7uBzR0r3a9ZaVHwQgr+PKfYXwDqKt2U+2aeDdeOtr6dO6hR4AwHdGS5ksJLJpsRD9DQ5TJkvphtLDmUshBs5SmQTxfG2kyCU3FiJd5d1Ybm+DLuQfCttbJjGGEhwqu4Pn2Ofn7n7B7l0a3cF7s7LxnqBX5uNVMGrSHXnOC9kRUrNHrOTYVDQ0oqnI6sNQkjQYeUT9MqJYLTO3W7gcU0+NvxFq6F3CfOIwX3Q5mHKBFFV6E3eQQvUaSMuvMwcmSqfYK6D9kWXuOAwCjNGEcwLZ1khaQZXcfe0JSwMSC3q9iKHVuK9/ivGFDcE6d7pupvi0tsHENzHUZuE1POFBvPeKrcUqXdBUlJRnkFZ0a2yGX6H7h8oyGwcL/F5l0Wk6IbrcOd0d9H9+nR8y77TtDT+yrR2XTyBtM7xhnU/yL/jUZfY4fA8e7q5eR0wmsGfcjj/WUSmzeJXu2V7LXKA4YrOFu/ys2xXW2AOV4Dw+pK2Ay7n/xUBy7mBjPZbTknWnv8s5HX9dXp7TJA3DVEa1zNKNkIjqDo0PHm5CtAA/xTpC4ZpPYt2W8iOMs4LPVJ9+q2aKoGZ3ShOvDiikzZ1MH/AzRBSP0cQgSE6ofQ26lb7BW+kYtZyeKDDYZ4LvIFOqfRBmHKHJLIMDD+rhrt9AP3DpXDc/bkDdCDv8QO0BwvscVf29kWWwFGXCG8pyMjXQ/pGe7gVZE5PZKmZZ9HJ+m1kRUfw8MBTo91Pfc7QhO0B8bQCIbpAl8R4JOSpu1bEoHoFUC+x9UfB8xsrNHMo/4hD8WdUGRiikJbqWgS6OTNbzuMTfM0z/BeL5lqzjxhgHJ2RQjdXHNlNiIF/KbaBlvLxRM4RGfFNcGRsn1W5JbURn0H8E73/jtJO0QeBG4N8dm+NqDSi9Q4PO78eB+hGDAMy8iQD2wP+AVfNMBVdjvHn8DsUaHwBPxC3wAq3AKZdYqZyyugo8r3FNdv0FWMK/rzPYOwlq4kfnfYBxPJYaXmYOaJMS1cRNjmdE5Z+5Vyeeb+HF8NRdWAwPTWGnDNeNMffZ0PK8Z2bQGm4NxR/fW2zoZsmk2B6QUc1PPUItMtaQuWyjUtLcKQMx2D6hm7JuJb2HnXeWyBbjI8oFVW9xqOU/TIyGzaVijZMnnwgd44knm5zLpbE+QrgGsuoUJQhrK5Zy2GkTvs4Fz/WlnrFhKR5HOrmYwWOoFzkm4hqtBTG9bcyKI0T5sRwJ3fH7bFBuMFBpdD6HaNdQcu7Ru2Wg/lI8XOV1r/UmAFLUiUH3HZbxGyqeOHTrwj2jUd+0Lvky2fCK/w5DutoMUqkN5pbM4jKM5fBsjRBNZ4c6gz9bR3nBGjaGQ6nsPgOz8oaUYbaaqlZzzh+lQgUVtdB66HFt1Gr1NFPbsO3n6MC6klDsyLGGBkNTnTgVGM4Nq8FLAyEMQyjg4nmWctjXvpEwUr5XkpiOJfmdx0JZLUH8o+hCbs1hNoC2zmwHnpWIwr0ivvxa4o8t1ljtb84OuC9xBAqTAKyeMlKOW2lcMv68RPjzBIQHA3XK9cg9BQzqsjGJ+QEqH+r9IDyP3NPDyMdpJqXkMX0FM6aXBsGjd6KiCn54CkOtRXBH/pVbAc7nJGwLyYLWcTxmTuDaQc2pkXWTDPD9FKrqD0bs5j/VTZrI2RIh1FaAtpRUN9GeVuCzvDDpbGdVDjdUwrCZt1DWuU1VZ+DRUXmhLYcVn5idVZVd6J/K+zBx+UnmV9dHrzQjJVGzVFGNcaUiXw7LCSDhLQgtqNA9I3WeuyjfdMVO94C7JB8bTwnMv/Ucc/ulyDwH08LkfYYOR+JpqNGSwjhq6JdvSsdmu4aNTtuFZwrWgBrHgz/iZvFK2PXNWzYJA8vGOQIvC49T+b9dEUTBjhzwmSxQJ7W9JzWeh4Y/xFQBlt6dhw9Cvt4FgAJxF/P4tLOmA8z65jTBxBATkpqF7w8vUIgm+4SOc6nNyrJrKhBS+sSuvqvKSFEfYDkfS/+OtEmYhzM3zuuYcsRxKovqNctOAJSUN/jvtLR8q4r9RBtYF3VyFbAGYRmDQ7RIpR0LyYnRL/Pn/0z7kHSxDLrrCoz4nupNd9E3C8Vf+a7zmdoBR88VEVvbfYsHAvYUqA9RENJELBIuNQjJxBxAVMPAj5HrL/RHTzSk7NQr6Kni5fKaDUfWcvWX1QafVYCQv8qhRzTQSB2KfGmaI1s8ta9ffdCA2ZlDnY/qu4lRfuGOXTWqb6zRuw08ExeS8CHYBFy8mSN7HgiqhHzjdPdiUchRYUsSI0lg/z8RbjdtXDP/AdHxK6sFWm+cqDazkqvmmgHgjtuWySnu+bYG1nYU7EWQSM+WQ5oRbpsDoKP8i5bKlUzDrr0e0X7fUz7e0C40/UUg7kDp5lxj6bjg9EfAPxRQIjDIXm7NvfpCxoCzvgvJEJGZH1XdZSKr95I0P0E3I8VFbh3bUfFH7QDwIHEl4ih0HFPn4A0pL88BLfVSrWvYcKhpHHLqi99M5VHPj/cK/pbz+lS7H3zs3DbKN4EBBjZqXhMrk1z6tAfDBSlGgGy0ef6iqOmF/1R5u+9E/9vO2ONkyH1P0F8uaEpHsU5Kvv1YkJNjRWJKwLKSuzgZQqYEhRS0zL3lARGlyWBHi4b/xFmKr3BGgSlMrvyP7dbPFG5gQUQNxtbyZMUJ0DuRKcF9k5gcoEFq0hg1VbSdgmZvOKbNyEYDzqjxQpb1CJmZlkUi4cqrYvGfo/1VZgKWNDbe6n1PEGJ5T0Lh8JaLfP1hT5ZTDDGtQfuD0UYgL+BvKr3YT/TtfUk+8HfkvjDbAzVDXE38w4FpcPPFfBJcWpBVPGAtaW7+Gm5hbGxOsYAGqyNjh2YmkhrRkEJJMdvpMy0zsA3ZwEBSrt7f9WxbxJUKEmw+gr79gkibyeXK8G48J2/+cUhjHU46H8PCkYUG7jDc4c6k2VWHl/2vH91fQ7M/KMd69SiuYpdOXyaeSPS7v4+dKDlegTfh12PXSWVQEEMFOpMO2dOjBYbkOhlUeRHqP5H16Ns+MgSRFy4oZPi4FhfzM2Gc8J9z26cRQR+qHnlyEmJRNf01mp1X5M2B7PeHdzxRs9JpvfJhzmN6lJT2AUUIWx8pVtQ2D3F3cfuxd5JSx59dBBhAnNWtKlP5ujqzkmSqVJo0H3OdsbA7iXEzxlnYnM4PZb6EbJ/7WHlUPMy5EXXGQ2mNmyI9UY4lJb2gJDlvFEJ3mv59s7D1TAdnKLtJ2e8Qv+FTUEntRqnO2xsstDjUq5G7bmf2Z4IamAdN+QxqAshh/t1+pjAmfjPtRG3PF+p/1bJBk/bAvtHc+G6+YvB1JxIMcWw9DwFGxsHhpyEx8/wq0fOmJvXxu5lf+L2b8WngM3d2r3xqV5bTr5BXa7AaG55ChJlP9/IfvA6B/uBx4oxoqrz4ujVlkQLGZy4XqnwTKYw85Yb+KNB/AqeQYOIm9BNlkFk7NufcUhlPq0ajlwDuAW7DUhQlKs4AJOKD9CdgSgzUa2g1blLYLzF7h4/j6rB2a0/KaTQ2JbbJIbjy4Mll6OpePeoFLkfFG84eQy6/R7J7MowifTWV4rnTMFH3VHRaABHGsSGQWDLqUeb7b6wYhTbuLApVPXRzp2KuzgqH5Y8YNX3Ym7TftPXEH+E3VWrCBuUPefRm37m4LC1ZmIs14PlZ1w5Eq+HRDWcqPDwfUKhWv3YQtUqFQZ6HYsE6Cb0tiDHAtKaCOT36dNbyVYmgUq3ndBbVs37fvZPQoC9LNJp02FXovD8+nm1/4hbOgBZpjMuNs7TKUd6dFiiVXtMb3i/VjfZ7ni7XEwHuNMOySKKAD/R6GsM6JC4fMXuyvZTutzMAw2rhoCfjFQBVZfpx+7mZPAEkfwyNj7UYadw5JFlbO/6grPuoiBsjM8uZ8Nk7vP9pCMYS5TvJPmjAi2vEYUt3K/OrSN5UzwgD0nSlE6m14Umw+r39ioxDWHsMeBWbqKShK8FUNZhDcKvuOFOP8nd54Byl6zX+9+Cq45j1B0wQ/yg3J/iFfuF6U6Pz1Z7t9ZmkfAdTQNS8Mq9/xIPXLPN+RN8kaytqI0ThjeKnIbnuTTmR9CTpebhjlUGG4sSFq0GZk5FqETpOKnaVVHAxsNzmCJNBL9ZbXEqT8b7AT1ivfNl5rAXuMZ9ZmlzZvr+WZJjX3YqwL8hexAsKLKqaaxdA6zFKDpNAswiQPbg8rrgvPp53PnzBl3NTZMh7qCkpyo+dDR4drbhMB86vXbUvZK/KXFZ5pmL+vdVdzNiTsNnd9xpQYfUhQVXyZ0dYvRpfQkp3hnWFxT7EkIPUTa8QDPxb/FVU99Gmr9VNunnHHzY6pqUyjyxt7yJgXU3nSpAO2cX2FWH5yRuo2cJWl9xIgbfOwLPTL5XmDYuV9IWZv53IOzyoWBPt1nVGpDMzV7HFJiTEAhNUJqWk4GaDPi3uuCrfB8ooyYGK+JAuZdsHXwOgTPfkqrZyvv1nNOjIxkUY5Y+zi1ZRt8WDFqOoQ5RGxMv0vQ1C1p3vcp+RzmbkUFHvH6usUVR7qfT4EufwXrBEcmfIvCYtmsKhmjJrhdQkjBCzg9JbhIzbSMdnC1EMNqPRtdT2QgbPQhMGfQmX/iS3OoaeRz2PqfY/ZmpvS3aXtxOoYbo9Ia1XYkxDCSpGfkrywwnP4rnOHFWoQBG6Pdj36Amg2GI3DHVfxRKfoHlH5vhpAKUoBTqK0QPLuPTzGhJX/gC3nnhnW+sx0eApT9OjdwXbLUq4rDYxGK4o3bPQ7JuiNXbGkdjlFgGw2C8hJSg3uHFT5NZuMWpIzeZ6/o2IhVxY4k8vIkeTIF3OHvdxs4K1fP3z9Jc0fNx+aClUuFDgVOdD7HS2kRoZq2UNR9UB0rFE5V6NB01zN/rnEWUZTtGR7KHbNH0/qSPTE0HuwVmXvGBKkE0XDPjcZwHJrYHPBXSZIfE/SYz2FwIiN4Y5YFJQLmuAhK3qwX0w7XsN9Yd+dGaTZ5pbDgz8TUHEFgAhTJVKT5CRPpWlEHCjsLH4yZ9cXf5ZePL7
```

```
VPH898I2avhMRERERkRkKpUXczOJkZjv5eE7y/zCzlWS6kRzMPubuz5rZQ8CPgCRwg7ungte5EdgJt
AEPuPuzwWttAh40s88Ae4D7a/6mRERERERmIJRA3N0TZAZV5qZdW2T7O4E786Q/BjyWJ/0AmVlVRER
EREQiKexZU5peOu2MjCVJe3CfzjtuVEREGoTqdYk6ldHGEeZgzaaXTjtHR8fZOLCH3QeHuWBJL1vWr
aKvu1NLzoqINCDV6xJ1KqONRS3iNZSYSLFxYA+7DhwlmXZ2HTjKxoE9JCZSYWdNREQqoHpdok5ltLE
oEK+heGcbuw8OT0nbfXCYeGdbSDkSEZHZUL0uUacy2ljUNaWGEuMpLljSy64DRyfTLljSS2I8RU+XP
nqpnSW3Pzqj7Q/edXmNctKa9Pk3L9XrEnUqo41FLeI1FO9oY8u6VVy4tI/2mHHh0j62rFtFvEO/SkV
EGpHqdYk6ldHGop9GNRSLGX3dndy3vp94ZxuJ8RTxjjYNlhARaVCq1yXqVEYbiwLxGovFbPJSkC4Ji
Yg0PtXrEnUqo41DXVNEREREREKgQFxEREREJAQKxEVEREREQqBAXEREREQkBArERUREREREERCoEBcRER
ERCQECsTLlE47I2NJ0h7cpz3sLImINCzVqdLsVMalHKEF4mZ20Mz2m9leMxsM0nrN7Akzez64PzVIN
zPbYmZDZrbPzN6e8zrrg+2fN7P1OennB68/FDy34pns02nn6Og4G7YOctYnHmfD1kGOjo7X7aDSwSw
izSTsOrWaVD9LPlEs4yqr0RT2LO8XuftrOf/fDnzb3e8ys9uD/zcBlwLLg9tq4B5gtZn1Ap8E+gEHn
jazHe7+erDN9cD3gceANcDjlWQyMZFi48Aedh04CsCuA0fZOLCH+9b313yi/OzBvHFgD7sPDnPBkl6
2rFtFX3enVsmSqlly+6NhZ6GlzfTzP3jX5TXKSX2EWadWk+pnKSRqZVxlNbqi1jXlCmBr8PdW4Mqc9
G2e8X3gzWa2ELgEeMLdh4Pg+wlgTfDYPHff5e4Od7ax++DwlLTdB4eJd7ZV+pJlyz2Yk2m
fPJgTE6ma71tEpBbCrFOrSfWzFBK1Mq6yGl1hBuIOfNPMnjaz64O009z9FYDgfkUs5zDwVpx
dIP5UmvSGI8xQVLeqekXbCkl8R47Qtw1A5mEZHZCrNOrSbVz1JI1Mq4ymp0hRmIv8vd306m28kNZva
eItvmu27iFaRPfVGz681s0MwGjxw5UnDn8Y42tqxbxYVL+2iPGRcu7WPLulXEO+rQIh6xg1nCUW5ZF
QlbOWU1zDq1mlQ/N7Za1qtRK+Mqq9FlmZ4bIWfCbDMwAmwA3uvurwTdS77r7meb2DvweC7Z8D3pu
9ufvHg/QvA98Nbk+6+78N0tflbpdPf3+/Dw4OAEYEPqmG6G4m2pXOXrbcNBmc0q1QSj5Pr2OW5UN0qHPK4cBFnm0lYACvwbsfYVrupcbNLy/ABfH9e7giQNpzSHYIMaUGXuRx2jbqDD6B3v6LOvjzDnUiFJULFH0IIS6AUJBmjRX/XBFZj1nVhRNZS0oxeg9HMz4vqfcrXYJmKbfHBOAaW6gVZjw2fR3B9AEiuqgcnuc/4c9fWKyEbv4AjbIevr+GZDKtDPAdXt6pWv9i6BX6mBGKBmGEV8RuoFtkxtyE7j3ChQ1D+KX0DnqqkiRJW1odjN6cI5l3gbNDaBX4xFbeU5w3N0lL2S0Qsg2y9Mx0N3uE4nA1SG4dN6CLb8XzpzMYUPo0Nr4CAVxHdWdI6ZdMMvJhYcfd8qfxCvVAo4VNUaKdWxoY4IhbcFPd9hX2I7bS1qCOOhW9aF5VQf1kPVdW1PtjmOmrhuPoDHdycmNBnJGJV7JGNjyrZeMtCBCCdsu18GC0XFPfn4Ru67Wcc2ebKyHxVPgHYx6uBJ2QYtPWWFCM2XlWGAxxGXtI8e0oDxb9a8f0Hu4jvGnHYIu7
```

9xok5ukVCF6VpiKS+9N2LiEgxUT9PFAzE3f0Fd38BeJe7/5677w9utwOXFHqeSLUVG4yXTjs4/NWG1
Xz31vdy5crTNf1dxJT6/mYz0DJKU1CJiEj1FTpPlHv+iPp5opzOMd1m9uvu/j0AM/t3QHdtsyWSUWy
QBZDnsZV0d7Uzp13T30XBzL+/mQ2gidIUVCIiUl2FziG98Q6GExNlnT+ifp4oZ/rCjwJfMrODZnYQ+
DPgIzXNlUggMZFi48Aedh04SjLt7DpwlI0De0hMpAo8tpe0E5kDrNXN/PvLPDYTUZmCSkREqqvYeWI
m548onydKtogHK2OeZ2bzyMyy8ovaZ0tqKZ12EhOpSP4ynK7UIIsoD8CQ6Hx/jVTmRUTCFpU6s9A5p
LurvWnO/yVbxIOFdH4buBG4ycz+wMz+oPZZk1rIXubZsHWQsz7xOBu2DnJ0dDyyi6AUG2QR9QEYEo3
vr9HKvIhImKJUZxY6T4yOJZvm/F9O15RvAFeQWUxnNOcmDaha3QHqpdggi6gPwJBofH+NVuZFRMIUp
Tqz2HmiWc7/5QzWXOTua2qeE6mLqM+nOV2pQRZRHoAh0fj+Gq3Mi+RacvujM9r+4F2X1ygn0iqiVGc
WO4c0y/m/nBbxfzGzt9U8J1IXjdido9ggiygPwJCMsL+/RizzIiJhiVqdWeg80Szn/3IC8V8Hnjaz5
8xsn5ntN7N9tc6Y1EYzXc4RKYfKvIhI+VRn1lc5XVMurXkupG6a6XKOSDlU5kVEyqc6s75KtogHq2u
eCbwv+DtRzvMkuprlco5IuVTmRUTKpzqzfsqZvvCTwCbgjiCpA/haLTMlArNf/lykWlQWRUTqrxXq3
nK6pvwHYBXwAwB3f9nMTqlprqTlFVsaXb/MpZ5UFkVE6q9V6t5yupiMu7sDDmBm3bXNkki05jGV1qa
yKCJSf61S95YTiD9kZl8G3mxmfffVNlvS6qI0j6m0NpVFEZH6a5W6t5zB/AfIG3mxmG4BvAffVNlvS6qI0j6m0NpVFEZH6a5W6t5zBmp8D/hbYDpwF/IG7f
7HWVLckQQAAIABJREFUGZPWUKj/V9TmMZXWNb0srj3vdL51y28ANG2fRRGRsGTjAoBv3fIbrD3v9Mn
HmjEOKKePOMB+YC6Z7in7a5cdaSW5/b9Om9fFzR84i8V9cUaOJ5nbEWPLulUn9Q3TPKatJZ12EhOpU
KfQys6pmy2nt15yNrc9vK+p+yyKiIQhlUozOp6iu6ud518dYeczr/B7a84mZvDqL8eaMg4oGYib2ce
APwC+AxjwRTP7lLs/UOJ5c4B/BLqC/fytu3/SzN4CPAj0eq27j5tZF7ANOB84Cvz8HwYdz
EeBFLDR3XcG6WuALwHo2BwtwF+4+11Bet597ZVnP2pNKF8AQ0QapCT7f81/5Qubvng2WzaPjW46Y13aB7TFpV
Oe+Y772rjtWNj/Mm3fjJZCZcKeqsdvOfOqYvDhm2D7DpwFGCyz+J96/vp6Sq3XUNEpDHk1sWJsRSxG
Mxpr825OJ12jibGuWlg72QscPdVK3jk6UN85sq3gdGUcUA5Z47bgFXufhTAzPqAfwGKBuLAGLDa0f
MrAP4npk9DtwC/LG7P2hmXwsXmH/A/d3AcB/NLO3AtcA5wCnA98ys7Of7C/H+W8/7C/AXLD8W2DdGT3M
jvc/UfBc/PtoyXlH3m8ks62GP/laz+YTPvCupX0dXfyxkS6LoU92//r0Y3vZtP2fUWDGwU5rSNfeb3
7qhV87pvPsXFgD/f/Tj9pJ2+gXatR9tk5ddPuJ/VZPG1eFzik3fWDsQJLbn90RtsfvOvyGuVERHJl6
tMxNuYExp+9egU9Xe2c0tWOmVW10SMxkeKmgb1TYoFN2/exee05xLvaiFlz1qvlDNY8BBzF4
pJ3nGSPBvR3Bz4H1k+pwDbAWuDP6+Ivif4PH3m5kF6Q+6+5i/xQYAt4R3Ibc/UZP1OGdSM0+Z
Fgjz2WZf+TxXl5PTExJu2lgL0OHR9mwdZCjo+N532M1P4ds39tlC3paYkCGlCdfed20fR83XLSM0+Z
1MTqWZMPWQW75m728dmwDEaOJydbwms5yj5ff/FbLzmbDdsGOesTjxc9dkREpotybJGpT/dOqU9ve
3gfP09MMDqe4tjxCTZsa37xFnmh
mbWa2FzgMPAH8X+Dn7p4Mn3Mnhdz4EnBH8fQZBgB88/gugLzd92nMKpfcV2UdFsq1sUwtc45xsCxXuM3vjJ
6UtW9BTMHip9ueQ7Xv70nBCAzNlUrHKeNOaf8vGgb2T3ZnueGR/pixuy5TFuR2xmv6oy5bZC5f20R4
zbvngWdz28L6mn15LRKov6rFFsdihu6v9pMa8jQN7OfzLsYrfS6FJGkbHkk3XLzxXOYH4/wW+TjCPO
PAN4BXglOBWkLun3H0lsUhMC/av5dssuM7vpgsPDdctGyyO9OUslj2
DhzMXMfIFL9X+HLJ9bxfM62LLupWTwc2FS/uackDGbJRbVptBsfJ62pvmsPvgMDdctGyyO9OUsljj2
XZy+4v/5M5LWdwX19WcaVqprEpjC7usRj22KFYXDx0eyduYd2ZvvOL3Mr2h48KlfXxh3Uq6O5u7u1/
Jjrfu/t+zf5vZqWRam2f0c83df25m3wXeSWY+8vagxXoR8HkKw2SHgTOCQmbnUWTL/21I
vuYnq97gXsB+vv7C76nRp/LMnfWh9w+4h1tMS5c2ndSP1w4Ebzk9s2uxecQixnxznbmtLdpYGYR5Zb
VZlCovHZ3tZMYK96dqburveaz7WT7i0Nm+sILlvRO9mmE/MdOK2mlsiqNLeyyGvXYIlMXrzypj3hXW
4wde39G18qpnQ1yG/Ng5u8lt6GjlWKBgmcKM/sD4CF3/z/z/BjBjCaPAyuBpn9n9trt/q9gLm9l8YCIIwuc
CHyAziPJJ4LfI9OleT6aFHWBH8P+u4PHvuLub2Q7Q/gr83s82QGay4DfpgZzLnQMlwR0oH
aMi2V+FjXqyLVS43Z0vX3s+PV3tHDueZOu//JTH9r9SsEW60OcwOpaku7O9KoPhQAMzW12xyjgxnuS
zV6+Y7M6U75isdkVebBaW/D8adDVHREqLemyRqYu7uO+684kHDSG/eGOcgX99kXWrF9M5rTHvs1ev4
H/8/XOTz6/kvbRiLFDsXf5H4NPB3+vJdGOZT2ZRn61kVtgsZiGw1czaguuHnh+vacjZ2iMukbWr
7g+3vB75qZkNkWsKvAXD3Z83sIeBHQBL4n+7+fccsc8a/Dr/8SuF7FBTZlJSUmfOeJBIGSd0d5LxQ4e
cJBXWUZsR+tftcnS5NaqSuFWXnBLFAsZkD2wSRCMyQFZeb8HT4YhR2OprWUR+6MbGwW4U1MqUD4UtQTkD6e1xZO0xPaRtj5kNAFxSbw22jzJGCDb8mMA3x3
EL+qTVbsQVHRGavkWKLQvXc5N9mbPuXn7J57TksW9DD0OERHvuZXF/nIu5fS06ZgvJhiZ40xMzsXeBW
4CLg157F4/qc0L51sM2Ixo7urnbM+8TjJnNHQUerXJs1tJj8eZyvqfThFpLE1S2wR72xjy3eG+Py3n
p9Ma48ZN75/eYi5agzFfqbcRGYu7v9DZnGcnwKY2WVkuntIi6r1zBQiUaGyLiJSmurKyhUMxN39KXf
/t+7e5+6fzkl/zN3X1Sd7Ui8zBQg3xRD5fZri/LiBdL40mnPLO7jmfvE+OzK2GzKuohIqyg09eC9C9
pjO9SU07nUQqZqZLgteab+2Wi0/LgKFl2M+paudU+Z0VFTGGqkPp4hIWGIxozfewZezvzYbGTo8woN
PvciVqxbx9T2HNKFDEQrBpawBdVNV0q+t0H5qPeBOWkPucszAH5HLMf/Sht9HWFqu4/+VMynqxqQ5FA
Jbc/mjYWRCpiTeSaT7+1aenTMe468Awm9eec+JcP8upjpuRhrLKjAakzaZrSeH9tEdyeV9pLMWWYy4
2uLJa3aWivly1iEgtFaqDswuwxTvbeW10jFQqHVIOo6lkIG5mH8pze7+ZLZg/ZLahBqX2yh1kS/QyB5U5
QQzhfYzdHgkksv7SnXUa1xAseWYRwvsv5rBc9SXqxYRqaVi5/js/U0DexKp5pCfPC8eOT5BKp1u
2L3k5LeIfBf4C+E/B7T7gFuCfzeaGuZN6qAcWEvMJ6Y4bE3bfu6+agJVfenJochtNDddc6
tlKnF2OObd8ffbqFbw53sFffu+nefdfzeBZUx2KSCsrdI7f+cwrk+f63Q8NXleuH7b0/z
s9eM88E8HWvIqYjmdJtPAr7n7qwBmdhpwD7Aa+Efgq7XLntRtvDScKNnHfp+RseS/
OX3fsqOH748uU2UlveV2atk/EGlTizH3E+8q43EWIqYwV/804HJuW2n77+awXPUl6sWEamlfOf4uR1

```
tXHLuwsnF/y5c2sfQ4RGWLeiZcl7YtH3fZF/yVlscsJwW8SXZIDxwGDjL3YeBidpkS+otOyAtZsF9n
sEUxS47ndk7dY2nQsFM7n66O9tZt/pXNTVcE6t3K3FmOeagHM9pZ06wyESh/Vdz7ltNdSgirW76OX4
4Mc7mHc/y2P5XprSQDx0emfK8qX3JW6vOLOcnxz+Z2d8BDwf/XwX8o5l1Az+vWc4kcuIdbXxh3Upuy
pke7u6rVvD1PYfoWnnGlG3LaQnU1HDNL+xW4lL7zwbP06fUrCR4VnkWETkhFjP64p1TpjT8+p5DXLN
6MQ8+9eKUbXP7krfaVcRy3ukNZILvdwEGbAO2u7sDF9UwbxIxhQ6qdasX09kW48KlfZPBzBfWrWRue
+kLLs2yvK/kV61At9JpAUvtv9rBs8qziMgJmaljjcREiuWn9XDGqUuZ2x5j3epfZdeB4ZMa9cqNHZp
JyTNFEHD/bXCTFpfvoIp3tOHuJ03krwn8pRqB7mwWgipn/wqeRURqJ18dW6gveSvGDiXPOmb2IeBuY
AGZFnEjE5/Pq3HeJKLyHVQjY6m8E/m32qALOdlsA93ZDvhUoC0iEi3ZenlkLMn121o7dijnXf4P4Df
d/ce1zow0Lk3dJrWisiUi0pxUv5c3a8qrCsKllGrOPiGSS2VLRKQ5qX4vr0V80Mz+Bvg6MJZNdPdHa
pYrmZFKB7JVUzVnn5DWVKgcq2zJbCy5/dEZbX/wrstr+voiURJ2/KD6vbxAfB6QAC7OSXNAgXid5Tt
ggIoHslWTpm6T2Sg1ILORylbYJzYRkXKk05nl5V9PTHBmb5zXjo1xaryDU+Z01K3OarT6vRZKdk1x9
w/nuX2kHpmTE7IHzGvHxnCH146Ncez4BMeT1Vuie7bKWRRImlc67YyMJUl7cD+DZYpLLTXfKGUr+4M
iu3Tzhq2DLblks4hE3/Fkiol0ekraRDrN8WR944dGqd9rpWAgbma/F9x/0cy2TL+VemEzO9PMnjSzH
5vZs2Z2U5Dea2ZPmNnzwf2pQboFrz1kZvvM7O05r7U+2P55M1ufk36Wy3fk36+me0PnrPFzKzYPqJkpkHL8WS
KY2NJ7nhkP2f//uPc8ch+jo0lwWn5gQ4SvtkGoLUcsDObHwgzVeoHhYhImHLrw1RQF+bGFccn0pk+D
1I3xVrEswM0B4Gn89xKSQL/1d1/DXgncIOZvRW4Hfi2uy8Hvh38D3ApsDy4Q/cA5mgGvgksBp4B/D
JnMD6nmDb7PPWBOmF9hEJMw1a0unMAbPwTXPZvPYcLnvbQnYdOMptD+8j7bT8QAc5oZ5BZ67ZBqC1G
rBT7xZqzQAgIlGVSqUZCebsfv7VEf7yez/l+ESa+ad0Tdbb2bhC6qdgIO7u/zv4M+HuW3NvZPqMF+X
ur7j7D4K/j5EJ7M8ArgC2BpttBa4M/r4C2OYZ3wfebGYLgUuAJ9x92N54A1gSPzXP3XcGiQ9umv
Va+fUTCTIKWbCBx/banOfv3H2fzjme59eKzWXve6ZkTfFdmoMOFS/tojxkXLu1ruYEOkhFmt4jZBqD
ZATvVLsf1bqHWDAAiEkXptHM0Mc7Hv3oilrhy1SIeefoQN1y0bHK7bFwh9VPOYM07gIfLSCvIzJYAq
4CngNPc/RXIBOtmtiDY7AzgpZynHQrSiqUfypNOkX1EwkyClnyLmWzavo/Na8/hyLExEmOplh/oIBm
zXfhmVvsOAtDcRRmyAWi5i+7UohzXu4VaMwCISBQlJlLcNLA3byyxbEHP5HYXLOklMZaiZ05rLKYTB
QU/aTO7FLgMOGNan/B5ZLqdlMXMeoDtwM3u/sugG3feTfOkeQXpZTOzclCRoKRR
ILFvQw5Z1K4l3tmnlwBZQTlkNs1tENQLQWpTj2f5AmCnNABBevSoyU61UVovFEi8NJ2iPWVBvr1RXu
jor1kf8ZTL9w48ztW/4DjLdRUoysw4yQfhf5cw7/mrQrYTg/nCQfgg4M+fpi4I8FEtflCe92D6mcPd
73b3f3fvnz59fzluasXx9dmdyGb7wpe4kfd1dLXWCb2XllNXZdIuYbd/y3AD0J3deyn3r++s+hWY+t
eryUkyrzwBQj3pVpBqauaxOr9OPT+Q/P4yOJVkwryun3lZcUW8Fm4Tc/YfAD83sr4PtFrv7c+W+cDC
Dyf3Aj9398zkP7QDWA3cF99/ISb/RzB4kMzDzF0G3kp3AH+YM0LwyXl+uAL5bYR
10Vmxu53FazQi2N3Z2td4KX4iptlS41h3e5onhlRi3UItJq8tfpK/nz//x2/svXfjCZ9oV1K+nubKO
tLdMmG5V6u9WU86mvAT4HdAJvMbOVwKfcfW2J570LuBbBb2Z7g7T/RiY4fsjPQmYYWa4wQ
2QGg34YIAi4Pw3sDrb7lLtnr6/8LvAVYC7weHCjyD7qqlSf3UJBy5QFQSZS9MY7FEhISZUGnbPtWx7
1BWyi+ANBRKRWjidTjI4l+drHVjN0eIQvPTnExoG93Hddv2KJCCrnrLSZzLSB3wVw973B4Mui3P175
O/HDfD+PNs7cEOB13oAeCBP+h6gL0/60Xx51Fr+uwWCl6Ktk6aKZCQoioJOmfTt7xaremzEfUfAiI
i9ZJKpUmmnDN74wwdHmHnM69w68Vn8/knniPe1UYsGKenWCI6yvkmku7+iyKDLKWAQgPFjk+kGB1L5
Q1ewpz5QlrTbAY0hl1eo/BDQJrLktsfDTsLIhXJTlF408Deyfrw7qtW8PU9h7j5A2fVbJC6zE7JJe6
BZ8zst4E2M1tuZl8E/qXG+WoKhQaK4UxeNnp047uZf0rX5NzGWhBE6m02Axpzy+va805n583v4WsfW
w1OXeYv10qWIiIZx5MpEmOpKbHFpu37uOTchSzui2sa1Ygq56fR/wt8AhgD/hr4JvrszWmaqWeTrszu
3PcZwYpw7Htk/5Rfr5594bnKbek63JjKbAY3Z8jr/lC5uvfhsNm3fV9eWaf1wFRHJNHyMjiXzxhbLF
vRobvAIK6dFfIG7f8LdLwhunwDeVuuMNFyvpU5m9kUyzMZhUP9uCt2n7vsnLRmFMtyZS6ZR72fJ6ywf
PYtP2fXVvmdZKliIi2auD+WOL0bGkGicirJyfR4+Y2W+6+88Azw+wJdQMF6RQi14i/vi4JpuTRpLt
rz29XSG0jKtlSxFRIrHFp52xRARVk4g/nHg62b2m8DbgT8kgV15aabk0aSSxmjIwlQ+l
SpR+uIiLlxRYSTSW7prj7bmAjmb7hm4EPuvtLNc5X08muchXvbOPL157PRkZeq2
a7iOV2rr2QpIq0pty7F4c//89tProMVW0RewZ9JZva/gdwzZBz4BXC/mVHr4SyD8/bk65aLJ5Ztr+aaBvZGZG1wLokg1lNOlq
uC849f1g6GyJyJShvx1aSauqObif4oPaq9g1xR3/wfge8D/5+7/MP1Wvyw2vjCnWCvVDSD7q3rD1kH
O+sTjbNg6yNHR8brMAS2tp9CxMLezLRJlr9rdZkREaqEecUUqlWZkLMncjjaef3WEB/7pQOh1dDMq2
kfc3VNAwszeVKf8NKWwplgrJ8g+nkwVXFxIpNoKHQtDh0dCX4yn0PGSPRkpOBeRqKh1XJFKpRkdT9H
d1c7Q4RF2PvMKV65axMBTLyg+qLJy5hE/Duw3s/vNbEv2VuvsNV2VuuMNZOw5gYvtepg7gIAZ//+42ze8Sy3X
nw2p83r0gAPqYl8x8LdV63gS08OAeEuxpPveBl46gWOJnTFSESipZZxRTrtHE2M8/GvPj0ZG1y5ahF
f33OIS85dqPigysrpRPRocJMKhTXFWqlLV7kLAACTCwD80YfeFsmVPNVXrfFNPxZePJrgc998jh0/f
BkIdxXZfMfLJecu5KZpx8jGgT3ce935dHdqhhYRCUct44rEROqkem/T9n1sXntOZpXOOtbRrXDeL/l
JuvvWemSk2YUxN3jBeUWDg6jU4kJRotk2mkf2WEinne6udo4cG6M9ZqEvxpPveFm2oKfAj9l2jo6Oq
/yJSGhqFVcUig2WLehhdCxZ1yC8Fc77JbummNlyM/tbM/uRmR3I3uqzNREZmdyeXHP7CcnTe/h//7h5f
x5WvPZ2575msv2MdsLBW5Ql6qm400ntwWnZ/ceSn3re8PtYLNd6l3NFioKFe2T3sY5U+DSUWklrJdV
p/7zKXsvPk9rD3vdCBT742OJenurF+LdKuc98v5WfOXwCeBPwYuAj4MRCtKkBH7xiMaM33sE1qxdz08D
ek35RFlwePIL9v8KceUZqJ0qryOa71Du3PXbSMXL3VSv43Defq3v5a5XWIREJR7465u6rVrBsfjfXr
F5Md2cbbW3lDC2sjlY575fzic51928D5u4vuPtm4H21zZUyxvJ9GRfr+m/KKPWIllMWDPPSGuZvkp
nW1uMvu5O7r3ufJ77zKVsXnvOZJ/2epe/VmkdEpFw5KtjNm3fx4d//S38SndXXYNwaJ3zfjlNUMfNL
AY8b2Y3Aj8DFtQ2W42XO1b2wnefxo3vXca2nS/yu+9ZxsGjo3z16Rd4+QzXX7z7NRWjIePb3nGL0XC+W1HGvIXNiB8pjOEGzfe2rCUj3x8Ab4sP1Esxnh1c8Ah4OGK2dKJUbMWzQdxbh4d6nIHQPHo1xUmSrjeMP3g3wT/tRm+JncE7uETJeeEUMMrcLJ0ktCwHMo3H/7JZo89/+nqB+6JlBP6yG1S5wLZ3TOv7zTnH3YDRhfdvKJiZc6tbwhx+g9OR4mK0f7r8XAE+9L27ZH/I3HpiwbAYGMZqkBNTYAp4ZTm3ZvKODsfyYn8oHX1WaHXXapXTZ5cPQ9k7W9wIqfNDn/4z6H8e7xA97PLWSxgRgaB6d4eLy/LmmDE94F8fR80Sg1uvfV/MkPaRDZx19D0Pd6xDUvMD7L8ZrlhA3mfPr3Mzb5drQnpjrPwfK4VFTcamrfTxD31HCu4yHz87/5kY5nJWuE5vM2z13MjBTTyDN4qRPdHIUMVWnZXGvdkcD5pffGbKeMFG56yBlA7rUJ7JTxf8K3WAegUm6e3Up7DBPcnd0hhFqdJTfOLb5JrWBAwzt5lj1cEnPWO6qbALhYy5rAzqWW5M0zOO6p8u9DGtMZlSY/ZYsqYYJlOa08lmyEHDMxRuXwFDtAxJOHwpdadmiyUl0iXNMP5MqmEwaWzb6Z1Dqbs4wtR3CMgbvOeL6PPSycnbzVW00fPEBzw4tdGavIQO7dp4YIBjyhQR+atk3Vl3mQXBH5Of7vzN13nM9ADBPySx4zXsBg9vzHMwePw23hxbN+R+hOSqu+LJzMMx45VWIMV0bgqTr67THjDBYSiHQsQ/9mzHD2MU/oLy/6v35dg1/JKaiWb2/fz6QF/M85G1zaY/xUvDB0kgoa5vbs9uxg7YVGFHlrCIADR1gQ9LlV68Ua9Zi2W6NlfxG5g0I21/Qcpfzls9A/fTn4/yRnUd1d8vVsZX/5g5Ej+vNYHK0Wg3d6+q6hOfONyQ3zb30DMgYTDHgSIE0uSX+31jyRqAbmMhBxiFPpVlDvh8AzSFVYeYm8Kqo5MgKk4fVQ0ZLZBfN/h3pvT5rGEfLW/3sWFl2HgCRCDkMkQJK8W2YRYqDiGxgHRNkSDVhl0Z/yDQH1VcfLnr7umVz0IW5CnP4/dXR3Qaaz8gP4qJsYf1/vQhT7dZSRdbsYsdZj9VjmXmzE2Xg8vMnq/LHz+GZTAe3Mmhb9eq+qZZMbnxkBjcc6r+Gu8QcN9MWfh4/uVJlkLBqoyG4ho+LB5MNKX86MNPnlQOCmqgRs8fb0LFEn9yBgGfMcLn+9WQjHIZJYOuzL/WxFUa6XdlZtP1pkR+1Y6HDsNkmCp7LyVZA8uchNnJ3vu5bKL/E9m54+9zyY29ebYTG1odLDGAsa/qm2J76B/tCtRQ2Gr4dXYlYuBGvwANyP8ARj/m3H/fKCtYl9Xp0yxiCFc5HhIG1npZeFeP95NMP2Tt94CM3cUXEqBBc7rC6LSwEnk3BJx4RWLC80r5TC+Uie4HnAz8F9vtQWS/jmhMzOwt1lA=
bdzzL7oPDU2YoqNdgolZpHRKRcMztiE0OyBw6PMKXnhzisf2v0B00TtRbq5z3y4m8biazvP1G4NNkW
sPX1zJTUj3HJ1LcesnZ3PbwvsmC/NmrV3B8IkW8M7qB93RhzTwjAoXLH1C37iKlBl+LiFQq2y0l29i
Q2y0lrDqmVc77Ja8zuPtudx9x90Pu/mF3/5C7f7+cFzezB8zssJk9k5PWa2ZPmNnzwf2pQboFc5QPm
dk+M3t7znPWB9s/b2brc9LPN7N7P9wXO2mGV+shXaRytKp+G2h/dNudR028P7SKfDztnMTe820GwHo0R
bvvJXz+4iYa1HICLNL3fKwtxuKb/zrreEWse0wnm/4E8cM9tR7InuvraM1/8K8KfAtpy024Fvu/tdZ
nZ78P8m4FJgeXBbDdwDrDazXjKDRfvJTKr3tJntcPfXg22uB74PPAasR4vso+WE+8qcDm7Sydvkdm
qZ3eRVmkdEpH6K1SX9cwJp1tKKynWIn4hsAj4J+BzwP+cdivJ3f8RGJ6WfAWQnZt8K3BlTvo2z/g+8
GYzWwhcAjzh7sNB8P0EsCZ4bJ6773J3JxPsX1liHy2nVQY7iISh3sdXK7QOiUj9KVYIT7FOP/8P8EF
gHfDbZFbXHHD3Z2e5z9Pc/RUAd3/FzLIDP88AXsrZ7lCQViz9UJ70YvuYwsyuJ9OizuLFi2fzniInd
wDZl689n6/880/Z8p2hph3s0OyauaxWKgorrrXKYKKZUFmVRqGyesLc9hhfvvZ8urvaGTo8ws5nXmH
d6l9t6bqsXgoG4u6eAv4e+Hsz6yITkH/XzD7l7l+sQV7ynUG9gvSyufu9wL0A/f39Ds+wDs+w
euJCc7sjTM8Os51/24JN7xvGW9MpHU5uwE1a1mtVJTm1J7b0cZfbVgdLIQFc9pb+/hSWZVGobKakUq
lGR1PTQnCr1m9mN54R0vXZfVSdBhsEIBfTiYIXwJsAR6Z5T5fNbOFQUv1QuBwkH4IODNnu0XAy0H6e
6elfzdIX5Rn+2L7aAnHkymOjSW545H9U2ZK6WyPaXYFaQq5gySByUGS963vD3355TntakGS8Cy5/dE
ZbX/wrstrlBNpBOm0czQxPmXRv7uvWsGDT73IR969lJ46zx3eigp+wma2FfgX4O3Af3f3C9z90+7+s
1nucwcnpj9cD3wjJ/26YPaUdwK/CLqX7AQuNrNTg9FSLgZ2Bs+9bmbvDEZLuW5aWvn20XJaZQaEFQV1QuBwkH4IODNnu0XAy0H6e
EXPHB1U0viiMEgyCj8GRERmo1A9NjqWVD1WJ+r804TinW1sWbdy2sIfK9VSJ00jd07tn9x5Kfet76/
7QE0tsCMijS5fPfaFdSvp7mztQef1pJ87TSgTpHRp4Q9patk5tQEtvywiUgHVY+FTIN6kwg5SRFqBj
jMRaXSqx8KlrikiIiIiIiFQIC4iIiIiEgIF4iIiIiIiIVAgXoZ02hkZS5L24D7dGCvhNmq+Rco1kzK
u40FEKtHMdUczv7dGoV75JRRaxrreU6XNVKPmW6RcMynjOh5EpBLNXHc083trJJZZR0f6+/t9cHDwp
PSRsSQbtg5Omez+wqV93Le+P9Kjixs13w0g9NqpUFltNTMp4y16PDREWV1y+6N1yo3U28G7Li9308i
W1WauO5r5vdVQ1cuquqaUUMky1lG41KPlt6XZlVvG02kHh699bDU7b34Pa887veC2IiK5onourUacE
dX31moUiJeQXf41V3YZ63yyl3o2bB3krE88zoatgxwdHa97MD7TfIs0mnLK+OTxuG2Qs3//cTbveJZ
bLz6bteedruNBREqK4rm0WnFGFN9bK1IgXsJMl7FOTKTYOLCHXQeOkkw7uw4cZeCpFxgdr28LuZbfl
mZXThnPdzxu2r6PWz54VlWPhyhcBROR6ovSuTRbz2AwOpZk/ildk/XaxoE9JCZmFkBH6b21MnUCKmG
my79Ov9Sz9rzTuXLVIq7f9nRdB0No2VppduWU8UKXXhf3xcGpyvGgAU8izSsq59J89czdV60AYMcPX
66oS0lU3lurU4t4GbLLv8YsuC9SSKdf6rnhomVs2r5vSotcJb9ca51vkUZUqowXu/RareMhX6t7vY5
xEam9KJxLC13du+GiZUDlXUqi8N5anQLxKpt+qWfZgh4NhhAJST0uvWrAk4jUWqF6ZtmCHnUpaXDqm
lJl0y/1jI4luWBJ75TpgbK/XDU9kEht1ePSa7bVXce4iNRKoXrmjfFUpn5Tl5KGpRbxGsi91NPd2a7
BECIhqvWlVw14EpFaK1jPdLapS0mDU3NNjWkwhEhz0zEuIrWmeqZ5NW2LuJmtMbPnzGzIzG6f7evNZ
noyDYYQqa96TyeoY1xERCrRCRfdAAAgAElEQVTRlIG4mbUBXwIuBd4KrDOzt1b6eqlUmjcmUhCcy4+
9McGx4xMtMVew5kduPK3+nc10sYvZZfl6t/nmLSH756oZK64uoLBZYiOrByjVlIA68Axhy9wPuPg48C
FxRyQul086xsSSvJzKr8531ice55aEfMjqe4niyuacni/qBLyfTdzaz6QRn+3np8xaRfPLXDWMcOz5
RUX0R5WlSVQ/OTrMG4mcAL+X8fyhIm7HERIqfJya47eGpc4H/14d+SDpdlbxGVpQPQfMlP39nMphOc7
eelz1tE8slfN+zl9cRERfVFlKdJVT04O80aiOfroHnSTzMzu97MBs1s8MiRIoKN7Zxpm98fwHQFf
4B0AtRfnAbzXllFXQdwbFF/GZbraflz7vk5VbVkXCVsuyWqhuOLM3flJaOfXFTOq1elM9ODvNGogfA
s7M+X8R8PL0jdz9Xnfvd/f+fPn532hxHiKl4YT+Q+AsfAPgFqK8oHfasopq6DvDGY2neBsPy993ic
rt6yKhK2WZbVQ3fDScOKktHLqiyhPk6p6cHaaNRDfDSw3s7eYWSdwDbCjkhfqScb0a8bvPpYCwqOLx6P1UOGB9ekMhnO8pggV8W8BKN0bTMyUDHBsbRmDwfe0wz/7Mj8dDezMvtC9ilmq0uZUONqOG9oT4IPNVqOcc1p5z+A8Hdm6Ul5ea0wk0KVAkr7ovQPlOLhT0UuomWS38uD4yQ7Ej2Ir/Zc27LZWbPyqVO
/so8CJwdbD5Y8BlwBCQAD5MZkfDZvZpHpYHew3afcPXsG/l0yM7PMJTNbyuNBeqF9tATNlCLNLgpzakf
hx4CIyGwUqseWLejRGgV1UqyP+KyvR7j7ugIPvT/Ptg7tg7cUOB1HgAeyJM+CJybbJ/lovn20isRYgSBlL
EXPHB1U0viiMEgyCj8GRERmo1A9NjqWVD1WJ+r804TinW1sWbdy2sIfK9VSJ00jd07tn9x5Kfet76/
7QE0tsCMijS5fPfaFdSvp7mztQef1pJ87TSgTpHRp4Q9patk5tQEtvywiUgHVY+FTIN6kwg5SRFqBj
jMRaXSqx8KlrikiIiIiIiFQIC4iIiIiEgIF4iIiIiIiIVAgXoZ02hkZS5L24D7dGCvhNmq+Rco1kzK
u40FEKtHMdUczv7dGoV75JRRaxrreU6XNVKPmW6RcMynjOh5EpBLNXHc083trJJZZR0f6+/t9cHDwp
PSRsSQbtg5Omez+wqV93Le+P9Kjixs13w0g9NqpUFltNTMp4y16PDREWV1y+6N1yo3U28G7Li9308i
W1WauO5r5vdVQ1cuquqaUUMky1lG41KPlt6XZlVvG02kHh699bDU7b34Pa887veC2IiK5onourUacE
dX31moUiJeQXf41V3YZ63yyl3o2bB3krE88zoatgxwdHa97MD7TfIs0mnLK+OTxuG2Qs3//cTbveJZ
bLz6bteedruNBREqK4rm0WnFGFN9bK1IgXsJMl7FOTKTYOLCHXQeOkkw7uw4cZeCpFxgdr28LuZbfl
mZXThnPdzxu2r6PWz54VlWPhyhcBROR6ovSuTRbz2AwOpZk/ildk/XaxoE9JCZmFkBH6b21MnUCKmG
my79Ov9Sz9rzTuXLVIq7f9nRdB0No2VppduWU8UKXXhf3xcGpyvGgAU8izSsq59J89czdV60AYMcPX
66oS0lU3lurU4t4GbLLv8YsuC9SSKdf6rnhomVs2r5vSotcJb9ca51vkUZUqowXu/RareMhX6t7vY5
xEam9KJxLC13du+GiZUDlXUqi8N5anQLxKpt+qWfZgh4NhhAJST0uvWrAk4jUWqF6ZtmCHnUpaXDqm
lJl0y/1jI4luWBJ75TpgbK/XDU9kEht1ePSa7bVXce4iNRKoXrmjfFUpn5Tl5KGpRbxGsi91NPd2a7
```

gWmC/me0N0v4bcBfwkJl9FHgRuDp47DHgMmAISAAfBnD3YTP7NLA72O5T7j4c/P27wFeAucDjwY0i+
5AWk047iYkU8c42EuMp4h1txGJ24rHxFPGuNl47NsaffOsnvPrLMbasW0Vfd+fkdlJ7xb6neuz76Og
4Gwf2sPvgMBcs6S2rDISZZ5FaKqdsF9qm0uNJpFVZZsIR6e/v98HBwbCzIVVU7IQAnPTY3Vet4HPff
I4jx8a4b30/PV15f6eGfiZptrIa9ol7ZCzJhq2D7DpwdDLtwqV9xcpA6HkuU+gZabay2grKKdvFtkl
MpGZ8PKGyKo2j6mVVK2tK00pMpNg4sIddB46STDu7Dhxl48AeEhOpvI9t2r6PGy5axu6Dw8Q728LOf
sso9j3VQ7yzjd0Hh6eklSoDYedZpFbKKdvFtqnkeBJpZQrEpWkVOyEUemzZgh4uWNJLYlwBVb2EfeJ
OjKe4YEnvlLRSZSDsPIvUSjllu9g2lRxPIq1Mgbg0rWInhEKPvTScYMu6VcQ7FFDVS9gn7nhHG1vWr
eLCpX20x4wLl/aVLANh51mkVsop28W2qeR4Eml16iMeUP+w5jPTPuJb1q2ku6udOe1FB92pL2OVRaG
/9UwHXkYhz2UIPSPNVlZbwWz7iGcHbM5wILPKqjSKqpdVBeIBHYSNrdgI/qKzpsx81gudMGpgNjOQh
DV7SQPMmhJ6ZpqxrDaTSurNUs+tkMqqNIqql9VaTl8oUhelWmeyI/Wnj9gv9pjUV6XfRZgt0yo/0sg
qrTezVP5FqkN9xKXhaQaL1qXvXqQyOnZEokGBuDQ8zWDRuvTdi1RGx45INCgQl4anGSxal757kcro2
BGJBgXi0vA0XVbr0ncvUhkdOyLRoBEW0vBiMaOvu5P71vdHeQYLqQF99yKV0bEjEg0KxKUpaAR/69J
3L1IZHTsi4VPXFBERERGRECgQFxEREREJgQJxEREREZEQKBAXEREREQmBAnEREREREkRAoEJeGlk47I
2NJ0h7cpz3sLMk0+o5EokHHokj0aL4iaVjptHN0dJyNA3vYfXCYC5b0smXdKvq6OzUXbkToOxKJBh2
LItGkFnFpWImJFBsH9rDrwFGSaWfXgaNsHNhDYkJLNEeFviORaNCxKBJNCsSlYcU729h9cHhK2u6Dw
8Q7tURzVOg7EokGHYsi0aRAXBpWYjzFBUt6p6RdsKSXxLhaeKJC35FINOhYFIkm9RGXhhXvaGPLulU
n9XmMd6iFJyr0HYlEg45FWXL7ozPa/uBdl9coJ5JLgbhEXjrtJCZSxDvbSIyniHe0EYsZsZjR193Jf
ev7T3pMoqHUd1TouxWR6pvb0cZfbVhNYixFLAZz2nW8iYRNXVMkstJpZ+R4EgxeOzbGV3cd5LVjY2A
wcjwz9VYsZvR0tROz4F4nlcgp9B1lZ3HYsHWQsz7xOBu2DnJ0dDySU6ply2LaM/eJcU391uiaZSq/c
t7H5LG2LTjWtg0yOqYuKSJRoEBcImn6ieOOR/Zz6bkL2bH3Z5MnkqgGbVKeRpnFIVMWx6YEMcOj4xw
7PqHy16Aa6UdgMeW+j0Y51kRakQJxiaR8J46bHtzLJecu1ImkSTTKLA6Zsrh3Slm87eF9vJ6YUPlrU
M0SmJb7PhrlWBNpRQrEJVSFLqsWOnEsW9Az5X+dSBpXo8ziML0srj3vdDavPYfFfXFwGq4VVRo/MM3
Wm/HONjavPYe1550++Vi+99Eox5pIK6pZIG5mD5jZYTN7Jidts5n9zMz2BrfLch67w8yGzOw5M7skJ
31NkDZkZrfnpL/FzJ4ys+fN7G/MrDNI7wr+HwoeX1Kr9yizU+yyaqETx9DhkSn/60TSuLKzOFy4tI/
2mHHh0r5IzuKQWxbXnnc6t158Npt3PKsuUg2skQPT6fXm5h3PcuvFZ08G4/neR6McayKtqJYt4l8B1
uRJ/2N3XxncHgMws7cC1wDnBM/5MzNrM7M24EvApcBbgXXBtgB3B6+lHHgd+GiQ/lHgdXdfBvxxsJ1
EUO5l1cvetpDNa8+ht7uT0fEkc9tjJ504vnDNSnY+84pOJE0id0aVn9x5Kfet7z9pue0oDKjLBDEru
XBpHzdctIxN2/c1fJeGVteogWk67YyOJ+nt7mTz2nO47G0L2XXgKJu27+OGi5YVfB/lHGsiEo6aTV/
o7v84g9boK4AH3X0M+KmZDQHvHvCFmf0dXk4HNwD3Ba20O0v8W+FMzM3dXk
1XEZC8PZ1sZN23fN2V+2954x5Rp7+a2x/jIu5dy4/uXa6q7JpGdUQWYvM/KtvxNn/e43gFEJojp4r7
r+ol3NXaXBsloxKlP8x0Pd1+1AoDH9r/C8tN6Mu+nwPsodqyJSHjC6CN+o5ntC7qunBqknQG8lLPNo
SCtUHof8N3T05Ln/JaweO/CLY/iZldb2aDZjZ45MiR2b+zFjfT1svs5eFCrYxvJNNTpr1ra4u17FS
FrVhWi10xqXfLeCxm9Mxpb+guDfXSKGW1XlOfVuuqTr6BmdmW8GwZbLV6cbYapaxKc6t3IH4P8G+Al
cArwP8M0vPVHF5Ber/Ojn5RQyXRg2cvDyxb0qJWxhFYsq9OvmGze8Sxn//7
jXL/t6dD6ZTdql4Z6asWyWkg1p0ksNoBdZbAyKqsSBXUNxN39VXdPuXsauI8T3U8OAWfmbLoIeLlI+
mvAm82sfVr6lNcKHn7kqo7nkwxOpbkax9bzaMb3838U7pK9p3NXh5OjCfVytjEKm0RLHXFJIx
+2eprKzNRaHrB48nUjI+JwldjkiqDIg2sroG4mS3M+fc/ANkZVXYA1wQznrwFWAb8GfgJWBbMkNJJ
kDnjqC/95PAbwXP98I+e11gd//xbwHfUPr6102hkdS3LHI/s5+/dPjOI/bV5XyVbtWMzo7mxXK2O
Tmk2LYFSvmGg1VylXvlbs0+Z1MTqWnPExUehqTHenyqBII6vZiA0zGwDeC/yKmgm818xbwOkyCPgm818xWkukqc
hD4OIC7P2tmDwE/ApLAx9x9InifjwPfAtqA+9392eAlPgU8aGa/D+wB7gvS7wP+PDjh8CiZ4EVmITTWTWFiTTy8iT/Zy+Hh5j+Rn8+zGqfDOhwCVj4Dd3MpH1vRwDOj2ZR9N5vA+Nx6zX+yY3SE9Qa1YvIrNA08aJc99+zdt9T1f+74WX5D3NDX41rSmm7NX7gQoL3a1W5u5CRyLqvaNLaM8f2X/6wn4yzxtF4ZMOPhnxHkHlxGRWcOd4SZ3kbaDBuzKzDPRbYvPnYNqPYwjwxx9kPsKCfb8oDHFj0Ho9dgPLgPAV
l4PUjmYIAcsFA2RNeyh35QvK9x+HgjNdHABfI3ggE6gJUeRV5XN1Xa3Zs1JoSDPAb9lFsufPbMYmUbyZkOk9xVS9Eh6+pl5wpETZEgfT5OQ+s37Y2TIHdo9xrIbA30NgLhZV1O+NYM9Ahk4SgEknK0jC+1y5pdGgMrEeBsimVcAZm97jW1NvcNAzOSzBGRP9D+lGSPVhzHOUpEu2THtu
Y8Fie9AOc6NqSm34cuHpGmZWy5Z/JYiWnzeuast3ug8OTC56UQyP6m9NsF07JvWIyffaUcq6YRGXmF
WlN2Vbs3PJ3Zu9crly1aMosUXdftYK5HaUvUKueFGk+OpJlRvK3cO7ljz70Nr6+9+XJ7S5Y0ktiLEX
PHBWxVpavRXAmrdkwu5bA2bTIi8xWvrKbGE9NjnmAE1cQ773ufE6Zo8WuRVqNjnqZkUItnIv74if38
dasJy2vWrOMVNovu9GXMpfGN73sdne15y2T3fphKNKSdOTLjBRs4RxLqe+inCTsfq3VaJEXqSaVSRH
JpaNeZmRue4wvX3s+3V3tDB0eYeczr7Bu9a8S7zwRXOlkIrnC7Near4/u9BZ5DeaUesqWyYGnXuCSc
xeybEEPo2NJ5rbrArVIK1LEJGVLp53hxMSUoOYL61bSG++oVKpFvtqDOVWepZRYzOiNd3D
N6sU8+NSLkBOMd3dCW5sCcpFWoiNeypZvcYqbBvbyRjJd0etVc9U5kUKK9S8vtOBKJYsFqTxLud5Ip
nnwqRe5ctWiyRVjP/7VpzmaUHkRaTUKxKVs1Rj4lrvK4uh4koGnXojEionSPGaykmc1B3NWM6iX5hb
vbOOScxeyafs+5p/SxaMb383XPraaxFiK40mVF5FWokBcylZ4ieXyThzTWwyv3/Y0V65axNrzTp/cR
jNayGzMtFV6tmU6l2ZokXIlxlMsW9DDafO6uPXisydbxe94ZD+jZS55LyLNQX3EBSivb2s5A9+KyTe
n86bt+9i89hx2X2/DAzB7lmD5DZOJ5MMTqW5GsfW83Q4RG+9ORQ0XnDZ1umc2k2DClXvKONkbEkN3/gr
Cmt4ssW9PDScILurhTxTpUZCdeS2x8te9uDd1lew5w0Nx3pUvaAtdlORVeoxXDZgh7aYzarIEgknXZ
Gx5Lc8cj+KSsWfv6J5wq2SldzesVqBvXS3DIrxrbRMyfOafO6uOWDZ09ZaXPLupXMaddAX5FWoEBcK
l590N1xnJGx8oKXwi2GSX5y56WaZULKUujqTaYc7z3pissffehtRVulS02vWO5MKGHPmS6Npa0txsj
xE63i01crvv93+kl7pgHj+ESKdBriXSpXIs1GfcSlaN/WKQPfjic5dnxiSh/vn71+nAf+6UBZs0MUW
mWxu3PmKyZKayrWB7zYqq+VtkqX0+c89xhJTARBkspzSTMZVNus4p1tLO6Ln1RuT5vXxehYkg1bB7n
lb/YyPDrOhm2ajUekGSkQl4ID1o5PpKYGIdsGOTaWZP4pXZOzQmzavo9Lzl1Y1uwQuS2GP7nzUu5b3
1/xfM3SmorNTFJw4OVYquIyVmomFE1ZWBl9bhmxmJEYO7nc3vyBsyav7vzue5dx28P7NBuPSJNSIC4

FW6rTaU4KQm57eB83XLRs8rnZPt7lzg5RbE5nkVKKXb0pVI5nM2tJqZlQNGVhZfS5nRDvPLnc5raSZ
+vXXJqNR6R5qI+4FOzbilFwcGXWBUt6GTo8otkhpC5KzUxS7T7apfanKQsro8/thHz1b7aVfNeBo5P
1q2bjEWlOOopbXO5ANAD8xIC1kbFk3hPAS8OJyVlO7r5qBV/fc0izQ0hdxDva+PP//HZeT0xwZm+cl
4YTnBrvmCx7pQZeVrK/YjOhaMrCyuhzm2p6uU2nfbLc3fPdIT579QoeefoQl5y7kGULehgdSzK3XRe
0RZpB69V4MqnUtIX5g5CVdHe185M7L2V0LEm8s42PvHupRvFL3Yyn0lOmKNyybmXN9lVqJhRNWVgZf
W7FTS934xMprlm9mJsG9hadYlZEGk9ZgbiZvcvd/7lUmjSWUtMWlgpCTpnTAUBPl1pmpD7yTVG4cWB
vyak2Z6NYK7umLKyMPrfScstd0uGmk8p96SlmRST6yo2gvlhmmjSQcvppanClREkU+xbrGKmMPrfyR
bHci0h1FP0pbWYXAv8OmG9mt+Q8NA9QDdDg1E9TGo3KrLQilXuR5lXqCO4EeoLtTslJ/yXwW7XKlNR
Hbj/N0+Z1cfMHzmJxX5zEWIp02tVCJZETxb7F5a68KVKpUoOURcK25PZHa/r6B++6vKavH6aigbi7/
wPwD2b2FXd/oU55kjqa29HGX21YzcjxJF/555+y5TtDGggkkVVO3+J6BsalBjyLVMt4Ks2OvT+bMnO
KuwMqZyKNrGgfcTP7k+DPPzWzHdNvdcifVMn05aRTqXRmZbttJ5ZQvvH9y3ny1vcy/5Sull1cQ+qr0
DLnxZY/L9a3uN4rNmphmuZRrMyFLTGRYuCpF1n3jsV0BdMWDo+Ocyxi+RSRmSvVNWVbcP+5WmdEaiO
d9kyrYFcbrx0b40++9RNe/eUYX1i3kgefepH5p3RxywfPZtP2fZMtendftYLPP/GcBgJJTRVqTe6Nd
zCcmKiolbnUTEDl5GkmrekaRNccUqk0o+Mpurvaef7VEXY+8wrrVv9qZK5sxDvbuOr8RYxNm7rzs1e
voLM9RrxT/cRFGlWpWVM+G9xf5u7/MP1W68zJ7Ey2Dm7LtA7e8ch+bvng2cw/pYubBvZyybkL2bTmb
NLufO1jq3l047uZf0oXm7bv4+YPnEViXK16UjvFWpMrbWWeTWBcqDU9lUoXbCnNDqLLlR1EJ40hNwg
fOpwJwq9ctYiBp16IzJWNxHiKN83t5LaH9005Lm57eB/pdNi5E5HZKPUzeqGZ/Qaw1sweZFpnNHf/Q
c1yJrOWr3Vw0/Z9fO7qFYyMpVh+Ws9JfcOzreGL++KgK55SQ4WC5u6u9oqD6UKzSxyfSJF2irZ05zt
eBp56oehCKlEcPCrlS6edo4nxKd9vdrXgS85dGJkrG/GONjDyHxddhfOogcQi0VcqEP8D/v/27j1Mr
qrM9/j37e50J91BoAPhJEJsmEQUISTQ3AZFQAk3GDSgwktEhqwEhqMAMj4gQOBw/hwdGIykV0lDwqlygPCWh
QBCRyMSDCoMcQkpgQQQAVCDJBJhkAil16RJJ939nj/2qs7uTl27q2q2pXdf0+z1NPV63aal1W73r171drrA
jOBvYH/7PeeA8eXIlNSHKmCztSDx3LhceMZP3ok6954h5FNDVx6Z9+mKKte28KCp9Zx2V0rufqfqTB9H
R2c3I4brdKaWTqdC8pbNrwEO1pSsY3/iZQ9jS2cWMHLMSpvthcOKBY7JOpFLuiWlUsBq8+DHcsq2LO
xa/vFNlxaypH2D86JEVMzxgXZ2xeev29OdFhmu1OhKLVIesTVPc/ZfufjLwbXc/rt9DhfAK17Gtmxn
Hj+c/Tn1/bwef7h5ne08Pe+7S1Ht787K7VnLhceOBqIZl3kjqmiqkJkqErVWg+ar9RNNQZR+03qrc2O
VN6LvGC8fPfOpk509upr6vrnY0zW1OXdM1Mxo8eWTFtwMvdvdEXUo6d8R85bfr+a9VzzI+fOWcfrkvZl
68NjeZZes2dQ7Kkkl3dlobmxg9rRJ/c6LSRljUR2JRapDXj/13f0bZjYVOCYkPebu9+Vaz57Gb
HD3A0NaK/BzA/yTu//dzAy1YA/yTu//dzAy4HjgF6ADOTTTV9MbPpwFfCZr/p7nND+qHAl4/e8C
OFT09Dg5f+sgE3t7axfwnX+X+5tfnLdWRO57KT9WfDUOmDHPx7IXsMiUkzZapPzqWXOVDvcf0r6Hve8C
tPpatNz1c6Xs9ZxsB1Ra0mf2Ojsprunhy/c/qMdwFnTf1A7/Uw9b23NFbW3YbovGjK++6LOhKLVIe
8prg3s6uBi4A/h8dFIS2XW4GT+qNAG+CH4E/xHXW4GT+qXNBB5x9wnAI6SGwmRH2n5jzSc0NYN
rXeSTn2URP6d9L8wm1Rrc8pB43G3p7kIpB43p7eCz64jG3uUPa2tl1YbNO2oedlutr
0lsbstfPlrHVUwSo/O8XGvKW83dmV9S7g+NEje7/f66dNoqWxnvr6vP49llWu8yJOHYlFqkO+1SinA
pPcvQfAzOYCy4BZxaNLxAoI/ezZmxTs9SnQmUo4Hvpe9ozoi9nqk:1cdmOsm7mf4swPiDrRm3azbOsH+HhpBM0oBNVb7VuiQCodPuncobtN4fjhKmzAW8RJrkjEn
4l7svCunzgNOBB7PsoyZEY86+xHfOmsiuIxppbqrn1Te39taCpzr4NNQZh7W1cv20STtqINXmVKpAI
bXDqZru+Ytf6jMZyoiGnQta/WvTgay18yOG1fW2J161YTM/fHQVDzy9viSF41qf5jzf9vFbu7rZ0tn
F7f96RO938uU7+9Z673QXcFsXz3/r5CHV7l4diUWqQyFX792AVHXMroPY7eHuamvH2m1d2d2SPI5ph4+js7uH8+Y
dhya0NatvS1adKz7aMPMzufqEadcePGDeIjlUd3dw8d26NhtlLtF+vr63b6BzW8oY5ph4+js7uH8+Y
tjV2EJzH14LG89nYnWzp3/sczsqnyaoAkUg2xOtiOhIWsX0jtcF2d0do8LOvoJ9mkK5yn8rtxyzZmL
Xi2T5OHkw/cK2ra0NRQ1IJdtRSsShGr+TYB6ulxtnR29RlrOzUSVKrgDTvfBWxp3FHLPJg8puZs6Oj
sps4Ag+ENyRTsy92RuBpVw3VVhr58rzpXA8vN7FGiIQyPIUdt+ACkuzpkmr83W3re3P1m4GaA9vb2i
u7xlJoJ86I7dhQkrj97Eq3Njfz9nb6Tn1w/bRIOvWPOQqrGcAVXf/IgWprqi/KPR8qn0mN1sG2lC12
/0Nrhd7p6so5+MhAd27t32uavlq/l7CPGcf68ZUVvM14tBatSxGq+d0Ci5VbsNArK1Z88iFc2dZTsL
mAUv519Rua57qyJDB9Wx/auHnYZPiyxwni6H5ESqfTrai5tM+9POgtSBDmrQEMnyj8ARwJ3h8dR7n7
HAPf5amhyQvi7IaSvBfaJLbc3sC5H+t5p0rPto2p1bO/mojv6jvxw0R0reKdr5zaqF81fQUtj+rGYx
41qZlRLU8X985bqNti20oWuX+jIKqVoX51ruMNStBkvpI3wUJLv95dpuXGjmhn9rqbetv97tDRRX1d
XtGMY/wEQn2xn89Zu/t6xXSOViEhGOQvioc32r9x9vbsvcPd73BHw6BBsFgmvmhKrGdzw5GhOuoWHVrsMnpFUyy5zsXZZ9mK7S9
ZqXLASmmNnuoZPmFGBheO9tMzsy/Fg4p9+20u2jamWa5CTj5CdN9Rk76tTKP28pn8EWdAtdP13Hymy
1zqXouFbpwx0OJfl+fxmX6+ymubF0P2Ayxe8+rc3s06rhYEUks3wbBT9hZocVunEzmw8sAvY3s7Vm9
nngGuAEM3sBOCG8oLgXkGYQhiA/2wdXAKmAO8EWA0EnzG8CS8Lgy1XETuAD4cVjP65cK9HgKomyTR8XoP65trrGAU8O
oxcWHV+ufvnlrF9edNXHnMWcrrD2pDA2DLegOZP18a4dTQ3n+9LwjeOzSYz190tiCxibPJF2tfKbzU
SNVDE6+d0AyLlfigvDW7d389pIP8+JVp7Dw4mOYevF4hkZFGFd46FzP4M7E8
0JvcWovbZ7u4TS5q7Mmpvb/elS5eWZV+Ftoe8NOgF10dzUwMsbO/j+b5/n1bc6M7YRnz1tMq3Nw+js7
qGnh97OQ80VNi5ulUr8AGaL1aRmXix3G/HB5WsSLar1ys8AABwGSURBVE0NRelE1/94j2ioY1PHzud
jQrMZVnSsFirf2C73OdDd3cOWbVEn+lUbNrPwfV88tC9GT4squcaVleXWBvxKpL4wSlnGaBYYqmnN+
JprTk06CylFj9V8C+LvSZfu7i8i8VOO0DEBTQ13aUVMqsQPpXVUVrVViwdyD5zWbAw7pA4vBN/qMzEiAx
XEJL4gc0Uq0lPaV3OUVPyVci5ViwVdD5WbKwOFT09zutbOvuMxnPtGRP5lfk1fPaD+1JfZ4mNmlJlE
j9A1RirKognouixmrVpipkNN7OLgS8TZbz3+7MzUikLaw6bvxLaCHuidcKJWO3BJX0lPaT3
YOCxFHCcxCY7Ox9oRHzknPlnQiQeOoaWpIWqXru9fRLLI1UZ8LtAOPE008+V3S56jGlBIe1jNpif5U
qzsTLMLSillOufGjx6pGBORvOQqiB/g7p9x95uAM4Ehf4q6rsNBhDkUKkemcS02

```
4JiKSS65GkttTT9y9KxolUAarkIk5qmU2PUmeYmVn1TIJjlSndOfc9dMm0aKO8SKSp1wF8YPN7K3w3
IAR4XVq1JR3lTR3Q0CmjlvZZjzrs872blqbh6kgITkNptA5mA6GFdQ5MS3NLijF0CfOO7upq4Phw+o
ZMayen5zbzvBhlRn/IlLZsv5XcvfarUorgoGMYpF1HU1JLzkMpNA5mNFWkh6pRaQc0sX5dWdN5DsLn
+PVtzqZPW0ywxvqdX0WkYLlO6GPDEC2USwyTeiT9MgXUnsGE3OVEK+FTo4lUqiO7d3MX/wSs6Z+gOe
+eTKzpn6Au5et5YJjx+saLSKDop/vJZSpR/3whjo2d3bR0tTAC69GE0BMO+I9jGpp1MgXUnaDibn4u
lMPHsuFx41n/OiRvLMt+rFZ6lpx1chLqUUzszpf+sgEXt7Ywf/5xQpefauTa8+YyNjdhgO6RovIwKl
GvITS9aifcfx4NnVs499uW8b+X3mQWQue5fTJezN/8Ut0bO/WyBdSdoOJudS6Uw8ey6VT9mfWgmfZ/
ysPct68pWzcsq3ktdOVUCMvQ1fqh95585bx3ise5PK7n+aSE/Znz12auOyulWzu7AJ0jRaRgVNBvIT
SDZ127tH7ZpwAormxXsOtSdkNJuZS615ywnu57K6VZS8Q6w6SlFK6H3qX3bWSC48bz5I1m9hl+DBdo
0VkUNQ0pYTSjmKRYwKIkU0NGm5Nymowo62k1h01sjGRAnGqRj4+hX2qdlId52Swsl2vD2tr5Z1t3dF
5o2u0iAyQasRLrHcUi9gd+t9e8mGmHjy293X/CSA0RbaU22Birq7OCmreUszOlbqDJMXUPza3bk8f1
69s6ojirLFe12gRGRRVGZVA/3GVRzTUsalj+05DX9UZvPpWpyaAkKqX72RCxe5cmU9tfqWPcy6ZlfO
7Sx+bk7jxM4fwhdv/1CetpamB4Q2KIxEZPBXEiyzdxfz6aZO4Y/HLvbfPF63eyJfvXMmcc9rBUMFAq
l6+zVvibW6B3rbkc6a3D7gpSa7JsTSqSnUq93eXPjZXMOecdjUVFJGSUdOUIkvXueei+Ss48cAxfZZ
bsmYTzU26rSlDRz7NW8rduVKjqlSvcn93GWMzdZ1WU0ERKQEVxIssW+eeOA13JbWo3MNzalSV6lX2H
20aOlZEEqCCeJ7y7WCWaezwjm1dvHjVKSy8+Bgu+egEdSiTmlSuzpWp8xXSd44uduFKs3sWLtsx6+l
xtnR28dw3T2bhxcf0fn8l/dGmjr8ikgC1Ec9DIW0V+3dam3H8eM4+fBznz1vWp814a/Mw3eKUmjOYo
RLzle58jXeOLnbhSu3QC5ftmAE7vXftGRMZv2cL0454T8kKxuWITRGR/sxdNTcA7e3tvnTp0rTvbe7
s4ry5S/uMVXzUfqMydjCL9/Tf0tnF+fOW5b2uVLzE/ytni1XJcr6WqHN0odeHMqrYWM12zIC07918z
qG0NKqN9hCV+JdajdfVtpn3J52FsllzzakFLV/osSlg+0WPVZUE81BoW8X4KA4tTQ1qoypSRtk63dV
Z8f/fqx164XIds3TvtYQOkyIiQ4naiOdhMJ141AFIpLzKfc7pHC9ctmOm4ykitUQF8TwMphOPOgCJl
Fe5zzmd44XLdsx0PEWklqhpSh4G04lHHYBEyqvc55zO8cLlOmY6niJSK1QQz1O22ftKua6IFK7c55z
O8cJlO2Y6niJSK9Q0RUREREQkASqIi4iIiIgkILGCuJmtMbOnzWyFmS0Naa1m9rCZvRD+7h7Szcxm
9kqM1tpZofEtjM9LP+CmU2PpR8atr8qrKsGhiIiIiJSMZKuET/O3Se5e3t4PRN4xN0nAI+E1wAI+E1wAnAxP
C43zgBogK7sDXgCOAw4GvpQrvYZnzY+udNJiMagprkcqic1JqhWJdZOhKuiDe32nA3PB8LnB6LH2eR
54AdjOzMcCJwMPuvsnd/w48DJwU3uVuy/yaOrQebFtFSw1HfN5c5fy3ise5Ly5S9m4ZZsuhiIJ0Tk
ptUKxLjK0JVkQd+AhM1tmZueHtL3cfT1A+Ds6pL8beCW27tqQli19bZr0AenY3s2M+ctZtHojXT3Oo
tUbmTF/OR3bNcGESBJ0TkqtUKyLDG1Jjgt1tLuvM7PRwMNm9tcsy7ecgHwPkA48aNy7h
zTWEtScs3VmuFzsnKpVgtLsV66ZQ6Vttm3l/Q8muuObXoeZDKl1iNuLuvC383APcQtfF+NTQrIfzdE
BZfC+wTW31vYF2O9L3TpPfPw83u3u7u7XvuuWfGvGrKZUlavrFaK3ROVi7FanEp1ktHsSqVIJGCuJm
1mNkuqefAFOAZYAGQvlkOnBveL4AOmGmnnIk8GZourIQmGJmu4dv1Bay1E2Au4JIwo2AD9z99+Y2RL
gF2b2eeBl4KywE7bzKzbwBLwnJXunvqHt4FwK3ACODB8ChYT4+zqWM7M+YvZ8maT
RzW1sr10ybR2jxMUy6LJKRY08qnOsLFz+/Z0yYzqqVR53cJ6bjnr1ixLiKVKZGCuLuvBg5Ok74R+Ei
adAcuzLCtW4Bb0qQvBQ4cbF7Hj6ytt0BmR2lGMadDTnd8z5i+Pzm9NrV4yO
u6FKUasi0hlUkkyB3WUERm6dH4nQ8ddRCSigngO6igjMnTp/E6GjruISEQF8RzUUUZk6NL5nQwddxG
RiBqb5ZDqKPOTc9vp6YHmpno6Ol5Wrp8fp2N6tjl15UEe4ZNTVGa3Nw7j5nENpaWpgS2eXjruI1
CQVxPO0pbNbPfyl4mk0isKpI1z5pRuNSnEqIrVITVPyoCmGpVooVqUaKE5FRCIqiOdBPfyl
oTkVEIroPm4dUD//UmLewo4e/bmVLJVGsSjVQnIrsrG3m/UlnoWIN5WOjGvE8qIe/VAvFqlQDxamIS
ERVD3nQyApSLRrUpyIiEriERXE86SRfgcRURERERJBEqiIuIiIiIJEAFcRERERERFcRERERCQBKoiLiIiIiCRAcRERERERFcRERERCQBKoiLiIiIiCRABXE
RBKggLiIiIiKSABXERUREREQSoIK4iIiIiEgCVBAXEREREUmACuJ56OlxNd20ePhb48nnSURQedmr
dD3LCJDlWZRyKGnx9m4ZRsz5i9nyZpNpHNbWyuxpkxnV0qhZ4EQSphOPzNuh7lkrRNvP+pLMgQ5BqxHP
o2N7NjPnLWbR6I109zqLVG5kxfzmd2zXyjkiSdE5KrVAsiwxtSc4LdTzpppITdO5WRv0PYvIUKaCeA
EciAjo3a4W+ZxEZylQQz6FjWjuKO19BlH7H7TeK2dMmM2pp7Q3AcqqtE8cQqwMYzIEcRzaB5Wz+xpkzlqv1E
01BlH7TeK2dMm0zxMtTEiSdK5WRv0PYvIUKbOmjnU1RmjWhZM72ZT2ESBKmc7M26
HsWkaFMBfE81NUZI5uiQ5X6KyLJ07lZG/Q9i8hQpYpIiIiIiIJUEFcREREREQpREREREEREQkoiLiIiIiCRABXE
RERERkQSoIC4iIiIiIgkgBz96TzUBHM7DXgpRyL7QG8XobVxA6GIsgVdd
T6qMc9Q2fmuhliFyj6GhRoqn6Xcn6NaYrUWDJUYLoZ0x6LosaqCeAHMbKm7T9yedj5RKYw9UXp4qLT/
lUI2fuRrzDNWb70oylI7hUPksQ+VzSOH03e9QrmOhpikiIiIiIiIglQQVxEREREJAEqiBfm5cfm5qQz0E+l5
QcqL0+Vlp9yqMbWUZSsdwqHyWofI5pHD67ncoy7FQG3EREREERkQSoRlxEREREJAEqiOf
JzE4ys+fMWnVuR2Ll9uRNCmbkU2RnR5g6d0BnG0N+Bn4j1Q2q0H6Ko9N3G2UwjPmy79gZtNj6Ye+G7
a8K61q//d9iZhvM7JlYWnmSN3n3n3nUF6fIzy8z+OXyjFwZ2Suy9y8+O2nzOzE2BJPaB83M9vAffVmf
DelN4vSq83zaQ77MYko6JAvJZUbEziDzXVHxVgkzHL4F87GNmj5rRKXdBxn+Tz1Zrbcz
O4LrwuOx2LFvJRPNV6LS6Hqzmd31yPHA6gHXgT2AxqBp4ADiryNMcCh4TsuwPPAAcANQEdmdQM22
cCSwO6a3A6vB39/B89/Dek8BRYZ0HgZP77esY4BDmrXLuP8s+0uVnFnBpmmmN3QPhOmoB9w3dVn+17A
34BnB2e3whcEJ5/bgbxcEqbnHNkgTBSQz4qKnUHkuabiKK+lHtuOXQF7GAIeE57sTGVc7sYLZgI
nuQT4GXDfQOKmxDGvR1njuPiceONXwCAd7Yez15cDlRd7HGnYvnYudD0HjIkF+1nPh+U3AtAt
P7LAdOAm2LpN4W0McBfY+l9loult/U7gUu+/0z7yJCfWaQvKPBoCF4TtL+72FE+d1oKH/95taNzx
vCMtZQnGXeEwUkNeKip0B5rmm4iivpR7bjl0BewgUu+Ql7ubU6ohjhOk/e9gUeA44ghHuPxYeTux95Pcoeu/2va
1UXwyU4JhV9PqtpSn7eDbwSe702pBWTAw+Z2T2E58pe0A8l+O/WfaRyZfCre
```

```
Obond8ik0P6OAN9y9K01+etcJ778Zlk9CJcZEvioxdvJRS/GVtHJcVwsWmmdMBhZTnXH8feD/Aj3h9
UDisZgxL8mqxhgummo4n1UQz0+6trNe5H0c7e6HACcDF5rZMQPIT6HpA5XU/m8A/gGYBKwHvluC/JT
ju85XNcVEvio5n7UWX0mruGNhZiOBu4CL3f2tbIumSUs8js3sY8AGd18WT86y72J9jor7LiWnIf8dV
8v5rIJ4ftYC+8Re7w2sK+YO3H1d+LsBuAc4HHjVzMYAhL8bcuQnW/reA8h/OfafaR87cfdX3b3b3Xu
AOUTHaCD5eR3Yzcwa0uSnd53w/q7Apkx5KqUKjYl8VVTs5KPW4qsClPy6WggzG0b0T/un7n53SK62O
D4amGpma4A7iJqnfJ/C47GYMS/JqrYYLopqOp9VEM/PEmBC6BXeSNSpZUGxNm5mLWa2S+o5MAV4Jux
jelhsOlE7J0L6OaGn75HAm+EWyEJgipntHm6rTyFqp7ceeNvMjgw9e8+JbSubcuw/0z7SHacxsZefC
McotY2zLRoBYF9gAlFHirTfm0eNtx4Fzszw2VL5ORP4XVi+rCo4JvJVUbGTj1qKrwpR0utqIUJs/QT
4i7v/Z+ytqopjd7/c3fd29zai4/k7d/80hcdjMWNeklVVMVwMVXc+J9V4vtoeRL1qnyfqMX5Fkbe9H
1Hv86eAZ1PbJ2p39wjwQvjbGtIN+GHIy9NAe2xbnwNWhcdnY+ntRAWLF4Ef0K+DGDCf6Hb8dqJfe58
vx/6z7CNdfm4L+1sZgn1MbPtXhG0/R2z0j0zfWzjmT4Z83gk0hfTh4fWq8P5+CcVb4jFRQF4rKnYGk
eeaia9KeWQ6fgnk44NEt5ZXAivC45RKj+Mcn+lYdoyaUnA8Fivm9ShrHFfdtbhEx6GqzmfNrCkiIiI
ikgA1TRERERERSYAK4iIiIiIiCVBBXEREREQkASqIi4iIiIgkQAVxEREREZEEqCBeZcys28xWmNkzZ
vZrM9st9t4t4EM3vSoim6f9tvvvWPN7E0zW25mz5nZ4xbwpZuH3uZZH1pn1ZnuUej9
SPGb2CTNzM3tfhvdvNbMz072XDfFmxZvbLPJZ7IB73ad6/2Mya891vbL1zzWxs7PWPzeyAQrcjyQixe
FvsdYOZvWZm9xW4nWMLWcfMJpnZKbHH7zLLtI81scbjm/8XMZhVju1n212Zmz+ReUqpNv7L
DnYVeI83srBCDj5pZu5nNDunHmtk/libXtUEF8erzjrtPcvcDiWZAuzD23kzgBnefCJyXZt3fu/tkd
98fmAH8wMw+kma5K4GH3f1gdz8gbFekv2nAH4gm9/RRg0d1/n7jkL7u5u+iru/kWWWi4G0/2TMrD7Leuc
CvQVxd/9Xd/9zrvxIxdgCHGhmI8LrE4D/LmQDQtmyEJMIhqjGAB3X+Du1wxgO0QVxKvbRg0GFxvKvbIuDdsdfbCNOuuvvfsq3o7iu
ICtxfSvP2GKLJAFLLroTeX76Pm9k9oab8xtSJa2ZTzGyRmf0p/NoeGdLXmXnNnXQ/rTqdpTMxtlZg+FG
vqbiAbUlyoRvt+jiS7OZ4c0M7MfhNi4HxgdW36NmV0idiZgvN7EUz+0JYrUrc2LtRO321mvzG
zF8zs2/22tYdFs4/eH+7cPGNmnnzKzGUSF6UfN7NGw/GYzu9LMFgNzWxJWOZO+GmoN
RphZo+ZWXvYrQQv8+Y2bWxvGw2s2+FPDxhZnuV9MBLLg8Cp4bn04CfpjzzGTzrmVE+ZSKT69x
uuGb9GngovjEzOywsv1+G+WNVOpvmcZ7Lq0j8sLHq4Jd4iXgYgNXHBwLUtCOwT1TX9bZ7Rva9Y76QGNZSHgA3T0T3
PfxAUt/92g00QQtuHt36oehmc0ys9vM7Hfh/OiteDGzL4c4bAedm0LnrTBi5nALPOYVo+tjSM=w0Ovl0jM2sjKtD/7xD/H0ro81S3pGeC0qPgGaM2h
RfSt2ZOj6PTA+Fhc/Av4E7JPuWmdmXyWaJOdGM7vOwl0jM2sjKtD/7xD/H0ro81S3pGeC0qPgGaM2h
7/1RLOXnRR771LgdeBjadY7ljDLWixtEtEUsP2XPRF4g2jK4iuAsbFtbCWaQa0eeJhoSuM9gMeBlrD
cZcBXw/M1wL+H518Efhyz44tcyrRLFh7JH189cg7Dj8D/CQ8/yNwCPDJEBP4RIXhN4AzY3FwQXj+nsQ8QFwQXj+P
aIZz3YB9gQ2hPQ24Jnw/FxgNbAr0ex/LwH7xLa1B3AGMCeWp13gaxpF9uY3npadasiEYv5u/b8HOoA7gOGCpz9+IYPiQwXgt41VEVx3enkVPqdpgRKCZrd3Sk
MRFD3BkeC/bta73+tjvvJoFXJr056vmh2rEq88IM1sBbARaiQo+mNkhRLdwJWmqbY
5bbq7LyQqbM8B3gcsN7M9w9tPuvtqd+0+Z/xfyNh14T2yTd4e/y4hOfIj+Odwe9nc/0
T8bqR7TgDvC8zvC62OA+R7V3K0juojHLQh/nwYWu/vb7v4asNU+bGUhsXm0q2/vKuH46nXqAHeGPuiWejmxtleS6c9xmx9ZwmZRS3hzaJcm83SNMWLDcHFaW1
5bbq7LyQqbM8B3gcsN7M9w9tPuvtqd+8mqm36IHAk0T+Z/xfyNh14T2yTd4e/y4hOfIj+Odwe9nc/0
T8bqR7TgDvC8zvC62OA+R7V3K0juojHLQh/nwYWu/vb7v4asNXSt/l+xN3fdPetwJ/pG1Op7XzUzK4
1sw+5+5sZ8toN3BV7/VMz+yzwFn4Ca3sRxfWD/Fcb5LhJaHQCaq7wz2lqWpU2AMIG67MSpOsb4VMtmO
5ZdgTvDXZfv0fd7f9jdN8Vev5+oQPtxd385pE0BZoZr3GNEhf1xeWTrV+7e41FtduqOyQeBO0P6/xB
VeqT7PFcS/Rh8CPhn4Dext+9193fc/fWw/uEj1OA5UQ1nO8DJoTl/+bRnVAIsWpmuwK
vYy5KTKDkuJCts/CekvufsT4Xm2a52UyEDaw0my3nH3SeECeh/RrcTZRDU8j7v7K2b2CaJCz43AA+7
uGcrik4G/pHsj/FP6GfAzizIxV6B9292kZ8twZ/nbTN+b6b0uqgJmNIirAHmhmhmTlQD7
kS1dtm+01Qc9MSep16nuxbFl+kfO7j782Z2KNEPvKvN7FcFgK9wi5m5dyEVLPzikWd34ZnyTN
kbza13UO1ULo8SiIWAN8hqrUbFUv/BvCou38i3FJ/LPbeln7bE8UF5OBdSHNgDPc/bn4gmZ2DjgGZYD
OPY+v3Nyd1fBG4wsznAa+H8g8zX4qvd/aZ+5/NpRFhe1+Ha8I5HfQ16hXJBPPbVRDQBqhGvUqH
2bwZwqZkNI6oBOc3MdnX3X3vxLdYvwuoda5PzObCPwH8MM07x1voUe1me0C/APRL2iAw81sX4X4avahn+Kq
LPeE8DRZjY+rNNsZu/N8ZJ/8HdG8z4dlj8Z2D3vd09+q8Q3H3K+m5KtvGbv1AvAz5HixxUyHa8ExX0yZ3
NcNLh7rcTFbwOCW+9TdTsZ1Uoft1i9q5x9vlZlpvMfBhi9ql1xPVtv5XmuWkMtwCXOnuT/dL35Udn
TfPzbGNN4iazF1lZseGtIXAv6fuMJrZ5JCeLd4y+QNwRmgrvhfRj4admNuTuaE4gK0KmOyqeZ2fB
QMD8WWBLy+Dnb0Ufn3WY2mgw86vT8ppl9MCR9usDPIUPLQK51A4l/iVHtTRVz9+Vm9hRwTrvfvZma3A
0+YWQdRweizwK2xDhQfMrPlRCNKbABmuPsjaTZ9Z9KNGIKl1EP9Z+7O5Lwj+kRcA1wEFEhel73L3HzM4
F5ptZU9jGV4jmf8n2ndk9Mg1Cb+ZGV4jaUGby9bD8n4h09JezLCuVZRpRDMTdRXQ7/wWiJiPRfRU/rC6kFEzbB6g01EbR4halLwo
Jmtd/c+Pwbc/Y1Qs/g0UVVyJbG3byXqjPQOcFRsnfVmdAR7X8just0b2k+kgyWu68Frk/z1reBuWZ
2CTs3m0q3nVfN7ONEsfQ5ohr17wMrQ+F4DfAxorhINVm5Os9s3gV8BHiG6FxZDKRrWvUvwPfCNb0L+
LS7d4ey+ZPA/UTNY74RmoOtM7P3P3A4vCMpuJ+nN0Z8nLZ4Fbwj4W5pl/GYIGeK37NfBLMzuNqD/Y70u
dz6HGdtxVFckuFMQvdfe044+LiEh+zGyku28qtiM3Ovh5vwVecqzQsgqEnNSrLatNOeN9xAq4N4wIE5eh43B8e1A4J0NxWD4A+4M+1t5yIKxNW
ZNyI1GNdl6FcBEZWlQjLiIiIiIKSAHXWFBERERFJgAriIiIiIiCRABXE
RERERkQSoIC4iIiIikoD/DwoWuJ3ZEhyrAAAAAElFTkSuQmCC\n",
       "text/plain": [
        "<matplotlib.figure.Figure at 0x260925a0eb8>"
       ]
```

```
     },
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "#do some ploting to see the distribution\n",
    "sns.pairplot(dataset)\n",
    "plt.show()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 40,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "California    17\n",
       "New York      17\n",
       "Florida       16\n",
       "Name: State, dtype: int64"
      ]
     },
     "execution_count": 40,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "dataset['State'].value_counts()\n",
    "#can see that there are 3 different categories for State variable"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 42,
   "metadata": {},
   "outputs": [],
   "source": [
    "#split the data to Input and output\n",
    "X = dataset.iloc[:, :-1].values\n",
    "y = dataset.iloc[:, 4].values\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 46,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Encoding categorical data\n",
    "# Encoding the Independent Variable\n",
    "from sklearn.preprocessing import LabelEncoder, OneHotEncoder\n",
    "labelencoder_X = LabelEncoder()\n",
    "X[:, 3] = labelencoder_X.fit_transform(X[:, 3])\n",
    "onehotencoder = OneHotEncoder(categorical_features = [3])\n",
    "X = onehotencoder.fit_transform(X).toarray()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 49,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([192261, 191792, 191050, 182901, 166187, 156991, 156122,
155752,\n",
       "       152211, 149759, 146121, 144259, 141585, 134307, 132602,
129917,\n",
       "       126992, 125370, 124266, 122776, 118474, 111313, 110352,
108733,\n",
       "       108552, 107404, 105733, 105008, 103282, 101004,  99937,
97483,\n",
       "        97427,  96778,  96712,  96479,  90708,  89949,  81229,
81005,\n",
       "        78239,  77798,  71498,  69758,  65200,  64926,  49490,
42559,\n",
       "        35673,  14681])"
      ]
     },
     "execution_count": 49,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "X.astype(int)\n",
    "y.astype(int)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 50,
   "metadata": {},
   "outputs": [],
   "source": [
    "# Splitting the dataset into the Training set and Test set\n",
    "from sklearn.model_selection import train_test_split\n",
    "X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 51,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1,
normalize=False)"
      ]
     },
     "execution_count": 51,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# Fitting Multiple Linear Regression to the Training set\n",
    "from sklearn.linear_model import LinearRegression\n",
    "regressor = LinearRegression()\n",
    "regressor.fit(X_train, y_train)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 52,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([103015.20159797, 132582.27760815, 132447.73845174,
71976.09851258,\n",
       "       178537.48221055, 116161.24230166,  67851.69209676,
98791.73374688,\n",
       "       113969.43533013, 167921.0656955 ])"
      ]
     },
     "execution_count": 52,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# Predicting the Test set results\n",
    "y_pred = regressor.predict(X_test)\n",
    "y_pred"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 53,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([103282.38, 144259.4 , 146121.95,  77798.83, 191050.39,
105008.31,\n",
       "        81229.06,  97483.56, 110352.25, 166187.94])"
      ]
     },
     "execution_count": 53,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "y_test"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 61,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "0.9485223547171526"
      ]
     },
     "execution_count": 61,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "regressor.score(X,y)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 76,
   "metadata": {
    "scrolled": true
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<table class=\"simpletable\">\n",
       "<caption>OLS Regression Results</caption>\n",
       "<tr>\n",
       "  <th>Dep. Variable:</th>            <td>y</td>         <th>  R-
squared:         </th> <td>   0.948</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Model:</th>                  <td>OLS</td>       <th>  Adj. R-
squared:    </th> <td>   0.943</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Method:</th>           <td>Least Squares</td>  <th>  F-
statistic:       </th> <td>   205.0</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Date:</th>             <td>Thu, 06 Dec 2018</td> <th>  Prob (F-
statistic):</th> <td>2.90e-28</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Time:</th>                 <td>14:19:02</td>     <th>  Log-
Likelihood:    </th> <td> -526.75</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>No. Observations:</th>      <td>    50</td>      <th>  AIC:
</th> <td>   1064.</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Df Residuals:</th>          <td>    45</td>      <th>  BIC:
</th> <td>   1073.</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Df Model:</th>              <td>     4</td>      <th>
```

```
</th>     <td> </td>   \n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Covariance Type:</th>      <td>nonrobust</td>    <th>
</th>     <td> </td>   \n",
       "</tr>\n",
       "</table>\n",
       "<table class=\"simpletable\">\n",
       "<tr>\n",
       "    <td></td>       <th>coef</th>     <th>std err</th>      <th>t</th>
<th>P>|t|</th>  <th>[0.025</th>    <th>0.975]</th>  \n",
       "</tr>\n",
       "<tr>\n",
       "  <th>const</th> <td> 4.122e+04</td> <td> 4607.941</td> <td>
8.945</td> <td> 0.000</td> <td> 3.19e+04</td> <td> 5.05e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x1</th>    <td> 1.339e+04</td> <td> 2421.500</td> <td>
5.529</td> <td> 0.000</td> <td> 8511.111</td> <td> 1.83e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x2</th>    <td> 1.448e+04</td> <td> 2518.987</td> <td>
5.748</td> <td> 0.000</td> <td> 9405.870</td> <td> 1.96e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x3</th>    <td> 1.335e+04</td> <td> 2459.306</td> <td>
5.428</td> <td> 0.000</td> <td> 8395.623</td> <td> 1.83e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x4</th>    <td>    0.8609</td> <td>    0.031</td> <td>
27.665</td> <td> 0.000</td> <td>    0.798</td> <td>    0.924</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x5</th>    <td>   -0.0527</td> <td>    0.050</td> <td>   -
1.045</td> <td> 0.301</td> <td>   -0.154</td> <td>    0.049</td>\n",
       "</tr>\n",
       "</table>\n",
       "<table class=\"simpletable\">\n",
       "<tr>\n",
       "  <th>Omnibus:</th>       <td>14.275</td> <th>  Durbin-Watson:
</th> <td>   1.197</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Prob(Omnibus):</th> <td> 0.001</td> <th>  Jarque-Bera (JB):
</th> <td>  19.260</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Skew:</th>          <td>-0.953</td> <th>  Prob(JB):
</th> <td>6.57e-05</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Kurtosis:</th>      <td> 5.369</td> <th>  Cond. No.
</th> <td>3.34e+17</td>\n",
       "</tr>\n",
       "</table><br/><br/>Warnings:<br/>[1] Standard Errors assume that the
covariance matrix of the errors is correctly specified.<br/>[2] The smallest
eigenvalue is 9.69e-24. This might indicate that there are<br/>strong
multicollinearity problems or that the design matrix is singular."
      ],
      "text/plain": [
       "<class 'statsmodels.iolib.summary.Summary'>\n",
       "\"\"\"\n",
       "                            OLS Regression Results
\n",

"=============================================================================
=\n",
       "Dep. Variable:                      y   R-squared:
0.948\n",
       "Model:                            OLS   Adj. R-squared:
0.943\n",
       "Method:                 Least Squares   F-statistic:
205.0\n",
       "Date:                Thu, 06 Dec 2018   Prob (F-statistic):
2.90e-28\n",
       "Time:                        14:19:02   Log-Likelihood:
-526.75\n",
       "No. Observations:                  50   AIC:
1064.\n",
       "Df Residuals:                      45   BIC:
1073.\n",
       "Df Model:                           4
\n",
       "Covariance Type:            nonrobust
\n",

"=============================================================================
=\n",
       "                 coef    std err          t      P>|t|      [0.025
0.975]\n",
       "----------------------------------------------------------------------
--------\n",
       "const       4.122e+04   4607.941      8.945      0.000    3.19e+04
5.05e+04\n",
       "x1          1.339e+04   2421.500      5.529      0.000    8511.111
1.83e+04\n",
       "x2          1.448e+04   2518.987      5.748      0.000    9405.870
1.96e+04\n",
       "x3          1.335e+04   2459.306      5.428      0.000    8395.623
1.83e+04\n",
       "x4             0.8609      0.031     27.665      0.000       0.798
0.924\n",
       "x5            -0.0527      0.050     -1.045      0.301      -0.154
0.049\n",

"=============================================================================
=\n",
       "Omnibus:                       14.275   Durbin-Watson:
1.197\n",
       "Prob(Omnibus):                  0.001   Jarque-Bera (JB):
19.260\n",
       "Skew:                          -0.953   Prob(JB):
6.57e-05\n",
       "Kurtosis:                       5.369   Cond. No.
3.34e+17\n",

"=============================================================================
=\n",
       "\n",
       "Warnings:\n",
       "[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.\n",
       "[2] The smallest eigenvalue is 9.69e-24. This might indicate that
there are\n",
       "strong multicollinearity problems or that the design matrix is
singular.\n",
       "\"\"\""
      ]
     },
     "execution_count": 76,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "# Building the optimal model using backward elimination\n",
    "import statsmodels.formula.api as sm\n",
    "X = np.append(arr = np.ones((50, 1)).astype(int), values = X, axis =
1)\n",
    "X_opt = X[:, [0, 1, 2, 3, 4, 5]]\n",
    "regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()\n",
    "regressor_OLS.summary() \n",
    "# we can see only the x5 variable has p value > 0.05\n",
    "# so exclude that variable and run the model."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 77,
   "metadata": {
    "scrolled": true
   },
   "outputs": [
    {
     "data": {
      "text/html": [
       "<table class=\"simpletable\">\n",
       "<caption>OLS Regression Results</caption>\n",
       "<tr>\n",
       "  <th>Dep. Variable:</th>            <td>y</td>        <th>  R-
squared:        </th> <td>   0.947</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Model:</th>                   <td>OLS</td>       <th>  Adj. R-
squared:   </th> <td>   0.943</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Method:</th>             <td>Least Squares</td>  <th>  F-
statistic:      </th> <td>   272.4</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Date:</th>             <td>Thu, 06 Dec 2018</td> <th>  Prob (F-
statistic):</th> <td>2.76e-29</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Time:</th>                 <td>14:20:07</td>     <th>  Log-
Likelihood:   </th> <td> -527.35</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>No. Observations:</th>      <td>    50</td>      <th>  AIC:
```

```
</th> <td>   1063.</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Df Residuals:</th>          <td>    46</td>      <th>  BIC:
</th> <td>   1070.</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Df Model:</th>              <td>     3</td>      <th>
</th>     <td> </td>   \n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Covariance Type:</th>      <td>nonrobust</td>    <th>
</th>     <td> </td>   \n",
       "</tr>\n",
       "</table>\n",
       "<table class=\"simpletable\">\n",
       "<tr>\n",
       "    <td></td>       <th>coef</th>     <th>std err</th>      <th>t</th>
<th>P>|t|</th>  <th>[0.025</th>    <th>0.975]</th>  \n",
       "</tr>\n",
       "<tr>\n",
       "  <th>const</th> <td> 3.686e+04</td> <td> 1959.786</td> <td>
18.806</td> <td> 0.000</td> <td> 3.29e+04</td> <td> 4.08e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x1</th>    <td> 1.189e+04</td> <td> 1956.677</td> <td>
6.079</td> <td> 0.000</td> <td> 7955.697</td> <td> 1.58e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x2</th>    <td> 1.306e+04</td> <td> 2122.665</td> <td>
6.152</td> <td> 0.000</td> <td> 8785.448</td> <td> 1.73e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x3</th>    <td>  1.19e+04</td> <td> 2036.022</td> <td>
5.847</td> <td> 0.000</td> <td> 7805.580</td> <td>  1.6e+04</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>x4</th>    <td>    0.8530</td> <td>    0.030</td> <td>
28.226</td> <td> 0.000</td> <td>    0.792</td> <td>    0.914</td>\n",
       "</tr>\n",
       "</table>\n",
       "<table class=\"simpletable\">\n",
       "<tr>\n",
       "  <th>Omnibus:</th>       <td>13.418</td> <th>  Durbin-Watson:
</th> <td>   1.122</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Prob(Omnibus):</th> <td> 0.001</td> <th>  Jarque-Bera (JB):
</th> <td>  17.605</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Skew:</th>          <td>-0.907</td> <th>  Prob(JB):
</th> <td>0.000150</td>\n",
       "</tr>\n",
       "<tr>\n",
       "  <th>Kurtosis:</th>      <td> 5.271</td> <th>  Cond. No.
</th> <td>3.20e+17</td>\n",
       "</tr>\n",
       "</table><br/><br/>Warnings:<br/>[1] Standard Errors assume that the
covariance matrix of the errors is correctly specified.<br/>[2] The smallest
```

```
eigenvalue is 3.66e-24. This might indicate that there are<br/>strong
multicollinearity problems or that the design matrix is singular."
      ],
      "text/plain": [
       "<class 'statsmodels.iolib.summary.Summary'>\n",
       "\"\"\"\n",
       "                            OLS Regression Results
\n",

"=============================================================================
=\n",
       "Dep. Variable:                      y   R-squared:
0.947\n",
       "Model:                            OLS   Adj. R-squared:
0.943\n",
       "Method:                 Least Squares   F-statistic:
272.4\n",
       "Date:                Thu, 06 Dec 2018   Prob (F-statistic):
2.76e-29\n",
       "Time:                        14:20:07   Log-Likelihood:
-527.35\n",
       "No. Observations:                  50   AIC:
1063.\n",
       "Df Residuals:                      46   BIC:
1070.\n",
       "Df Model:                           3
\n",
       "Covariance Type:            nonrobust
\n",

"=============================================================================
=\n",
       "                 coef    std err          t      P>|t|      [0.025
0.975]\n",
       "----------------------------------------------------------------------
--------\n",
       "const       3.686e+04   1959.786     18.806      0.000    3.29e+04
4.08e+04\n",
       "x1          1.189e+04   1956.677      6.079      0.000    7955.697
1.58e+04\n",
       "x2          1.306e+04   2122.665      6.152      0.000    8785.448
1.73e+04\n",
       "x3           1.19e+04   2036.022      5.847      0.000    7805.580
1.6e+04\n",
       "x4             0.8530      0.030     28.226      0.000       0.792
0.914\n",

"=============================================================================
=\n",
       "Omnibus:                       13.418   Durbin-Watson:
1.122\n",
       "Prob(Omnibus):                  0.001   Jarque-Bera (JB):
17.605\n",
       "Skew:                          -0.907   Prob(JB):
0.000150\n",
       "Kurtosis:                       5.271   Cond. No.
3.20e+17\n",

"=============================================================================
=\n",
       "\n",
       "Warnings:\n",
       "[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.\n",
       "[2] The smallest eigenvalue is 3.66e-24. This might indicate that
there are\n",
       "strong multicollinearity problems or that the design matrix is
singular.\n",
       "\"\"\""
      ]
     },
     "execution_count": 77,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "X_opt = X[:, [0, 1, 2, 3, 4]]\n",
    "regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()\n",
    "regressor_OLS.summary() \n",
    "\n",
    "#now the R-squared value is 0.947 and Adj. R-Squared is 0.943 and all the
variable has significant P values."
   ]
  }
 ],
 "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": "3.6.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```