```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "# SALES PREDICTION\n",
        "\n",
        ">Sales prediction involves forecasting the amount of a product that\n",
        ">customers will purchase, taking into account various factors such as\n",
        ">advertising expenditure, target audience segmentation, and\n",
        ">advertising platform selection."
      ],
      "metadata": {
        "id": "Anoaq3ENWp61"
      }
  },
  {
   "cell_type": "markdown",
   "source": [
    "IMPORTING IMPORTANT LIBRARIES"
   ],
   "metadata": {
    "id": "EaTrrCP7W3ht"
   }
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {
    "id": "PxHL5IcCWgu_"
   },
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import pandas as pd\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns"
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "LOADING THE DATASET"
   ],
 "metadata": {
  "id": "1OYC9EeaXHll"
 }
},
{
 "cell_type": "code",
 "source": [
  "df = pd.read_csv(\"/content/advertising.csv\")\n",
  "df.head()"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "bW5BtmCoXD9H",
  "outputId": "796fd3bf-cfb4-4d4d-de6c-60e77d41232d"
 },
 "execution_count": 8,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "      TV  Radio  Newspaper  Sales\n",
     "0  230.1   37.8       69.2   22.1\n",
     "1   44.5   39.3       45.1   10.4\n",
     "2   17.2   45.9       69.3   12.0\n",
     "3  151.5   41.3       58.5   16.5\n",
  "4  180.8   10.8       58.4   17.9"
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-06e4a93c-de20-4d73-854b-78f17dba1bff\" class=\"colab-df-container\">\n",
  "    <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>TV</th>\n",
  "      <th>Radio</th>\n",
  "      <th>Newspaper</th>\n",
  "      <th>Sales</th>\n",
  "    </tr>\n",
  "  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>230.1</td>\n",
"      <td>37.8</td>\n",
"      <td>69.2</td>\n",
"      <td>22.1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>44.5</td>\n",
"      <td>39.3</td>\n",
"      <td>45.1</td>\n",
"      <td>10.4</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>17.2</td>\n",
"      <td>45.9</td>\n",
"      <td>69.3</td>\n",
"      <td>12.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>151.5</td>\n",
"      <td>41.3</td>\n",
"      <td>58.5</td>\n",
"      <td>16.5</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>180.8</td>\n",
"      <td>10.8</td>\n",
"      <td>58.4</td>\n",
"      <td>17.9</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"    <div class=\"colab-df-buttons\">\n",
"\n",
"  <div class=\"colab-df-container\">\n",
"    <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-06e4a93c-de20-4d73-854b-78f17dba1bff')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"\n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 -960 960 960\">\n",
"    <path d=\"M120-120v-720h720v720H120Zm60-500h600v-160H180v160Zm220 220h160v-160H400v160Zm0 220h160v-160H400v160ZM180-400h160v-160H180v160Zm440 0h160v-160H620v160ZM180-180h160v-160H180v160Zm440 0h160v-160H620v160Z\"/>\n",
"  </svg>\n",
"    </button>\n",
"\n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    .colab-df-buttons div {\n",
"      margin-bottom: 4px;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"    <script>\n",
"      const buttonEl =\n",
"        document.querySelector('#df-06e4a93c-de20-4d73-854b-78f17dba1bff button.colab-df-convert');\n",
"      buttonEl.style.display =\n",
"        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"      async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-06e4a93c-de20-4d73-854b-78f17dba1bff');\n",
"        const dataTable =\n",
"          await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                    [key], {});\n",
"        if (!dataTable) return;\n",
"\n",
"        const docLinkHtml = 'Like what you see? Visit the ' +\n",
"          '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"          + ' to learn more about interactive tables.';\n",
"        element.innerHTML = '';\n",
"        dataTable['output_type'] = 'display_data';\n",
"        await google.colab.output.renderOutput(dataTable, element);\n",
"        const docLink = document.createElement('div');\n",
"        docLink.innerHTML = docLinkHtml;\n",
"        element.appendChild(docLink);\n",
"      }\n",
"    </script>\n",
"  </div>\n",
"\n",
"\n",
"<div id=\"df-e5caa507-b295-42ee-a869-56f851e20f44\">\n",
"  <button class=\"colab-df-quickchart\" onclick=\"quickchart('df-e5caa507-b295-42ee-a869-56f851e20f44')\"\n",
"            title=\"Suggest charts.\"\n",
"            style=\"display:none;\">\n",
"\n",
"<svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"     width=\"24px\">\n",
"    <g>\n",
"        <path d=\"M19 3H5c-1.1 0-2 .9-2 2v14c0 1.1.9 2 2 2h14c1.1 0 2-.9 2-2V5c0-1.1-.9-2-2-2zM9 17H7v-7h2v7zm4 0h-2V7h2v10zm4 0h-2v-4h2v4z\"/>\n",
"    </g>\n",
"</svg>\n",
"  </button>\n",
"\n",
"<style>\n",
"  .colab-df-quickchart {\n",
"      --bg-color: #E8F0FE;\n",
"      --fill-color: #1967D2;\n",
"      --hover-bg-color: #E2EBFA;\n",
"      --hover-fill-color: #174EA6;\n",
"      --disabled-fill-color: #AAA;\n",
"      --disabled-bg-color: #DDD;\n",
"  }\n",
"\n",
"  [theme=dark] .colab-df-quickchart {\n",
"      --bg-color: #3B4455;\n",
"      --fill-color: #D2E3FC;\n",
"      --hover-bg-color: #434B5C;\n",
"      --hover-fill-color: #FFFFFF;\n",
"      --disabled-bg-color: #3B4455;\n",
"      --disabled-fill-color: #666;\n",
"  }\n",
"\n",
"  .colab-df-quickchart {\n",
"    background-color: var(--bg-color);\n",
"    border: none;\n",
"    border-radius: 50%;\n",
"    cursor: pointer;\n",
"    display: none;\n",
"    fill: var(--fill-color);\n",
"    height: 32px;\n",
"    padding: 0;\n",
"    width: 32px;\n",
"  }\n",
"\n",
"  .colab-df-quickchart:hover {\n",
"    background-color: var(--hover-bg-color);\n",
"    box-shadow: 0 1px 2px rgba(60, 64, 67, 0.3), 0 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"    fill: var(--button-hover-fill-color);\n",
"  }\n",
"\n",
"  .colab-df-quickchart-complete:disabled,\n",
"  .colab-df-quickchart-complete:disabled:hover {\n",
"    background-color: var(--disabled-bg-color);\n",
"    fill: var(--disabled-fill-color);\n",
"    box-shadow: none;\n",
"  }\n",
"\n",
"  .colab-df-spinner {\n",
"    border: 2px solid var(--fill-color);\n",
"    border-color: transparent;\n",
"    border-bottom-color: var(--fill-color);\n",
"    animation:\n",
"      spin 1s steps(1) infinite;\n",
"  }\n",
"\n",
"  @keyframes spin {\n",
"    0% {\n",
"      border-color: transparent;\n",
"      border-bottom-color: var(--fill-color);\n",
"      border-left-color: var(--fill-color);\n",
"    }\n",
"    20% {\n",
"      border-color: transparent;\n",
"      border-left-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"    }\n",
"    30% {\n",
"      border-color: transparent;\n",
"      border-left-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"      border-right-color: var(--fill-color);\n",
"    }\n",
"    40% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"    }\n",
"    60% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"    }\n",
"    80% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"      border-bottom-color: var(--fill-color);\n",
"    }\n",
"    90% {\n",
"      border-color: transparent;\n",
"      border-bottom-color: var(--fill-color);\n",
"    }\n",
"  }\n",
"</style>\n",
"\n",
"  <script>\n",
"    async function quickchart(key) {\n",
"      const quickchartButtonEl =\n",
          "        document.querySelector('#' + key + ' button');\n",
          "      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.\n",
          "      quickchartButtonEl.classList.add('colab-df-spinner');\n",
          "      try {\n",
          "        const charts = await google.colab.kernel.invokeFunction(\n",
          "            'suggestCharts', [key], {});\n",
          "      } catch (error) {\n",
          "        console.error('Error during call to suggestCharts:', error);\n",
          "      }\n",
          "      quickchartButtonEl.classList.remove('colab-df-spinner');\n",
          "      quickchartButtonEl.classList.add('colab-df-quickchart-complete');\n",
          "    }\n",
          "    (() => {\n",
          "      let quickchartButtonEl =\n",
          "        document.querySelector('#df-e5caa507-b295-42ee-a869-56f851e20f44 button');\n",
          "      quickchartButtonEl.style.display =\n",
          "        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
          "    })();\n",
          "  </script>\n",
          "</div>\n",
          "    </div>\n",
          "  </div>\n"
        ]
      },
      "metadata": {},
      "execution_count": 8
    }
  ]
},
{
 "cell_type": "code",
 "source": [
  "df.shape"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "nCFpnkznYzjD",
  "outputId": "53a861f8-76cb-433a-e227-bd2ed6103720"
 },
 "execution_count": 9,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "(200, 4)"
    ]
   },
   "metadata": {},
   "execution_count": 9
  }
 ]
},
{
 "cell_type": "code",
 "source": [
 "df.describe()"
],
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/",
  "height": 300
 },
 "id": "yy560mxWXUuL",
 "outputId": "0bb91614-139b-4143-b101-11ca7425e926"
},
"execution_count": 10,
"outputs": [
 {
  "output_type": "execute_result",
  "data": {
   "text/plain": [
    "               TV       Radio   Newspaper       Sales\n",
    "count  200.000000  200.000000  200.000000  200.000000\n",
    "mean   147.042500   23.264000   30.554000   15.130500\n",
    "std     85.854236   14.846809   21.778621    5.283892\n",
    "min      0.700000    0.000000    0.300000    1.600000\n",
    "25%     74.375000    9.975000   12.750000   11.000000\n",
    "50%    149.750000   22.900000   25.750000   16.000000\n",
    "75%    218.825000   36.525000   45.100000   19.050000\n",
    "max    296.400000   49.600000  114.000000   27.000000"
   ],
   "text/html": [
    "\n",
    "  <div id=\"df-08dad05a-fc03-4156-a921-40949d299ab2\" class=\"colab-df-container\">\n",
"    <div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>TV</th>\n",
"      <th>Radio</th>\n",
"      <th>Newspaper</th>\n",
"      <th>Sales</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>count</th>\n",
"      <td>200.000000</td>\n",
"      <td>200.000000</td>\n",
"     <td>200.000000</td>\n",
"     <td>200.000000</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>mean</th>\n",
"     <td>147.042500</td>\n",
"     <td>23.264000</td>\n",
"     <td>30.554000</td>\n",
"     <td>15.130500</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>std</th>\n",
"     <td>85.854236</td>\n",
"     <td>14.846809</td>\n",
"     <td>21.778621</td>\n",
"     <td>5.283892</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>min</th>\n",
"     <td>0.700000</td>\n",
"     <td>0.000000</td>\n",
"     <td>0.300000</td>\n",
"     <td>1.600000</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>25%</th>\n",
"     <td>74.375000</td>\n",
"     <td>9.975000</td>\n",
"     <td>12.750000</td>\n",
"      <td>11.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50%</th>\n",
"      <td>149.750000</td>\n",
"      <td>22.900000</td>\n",
"      <td>25.750000</td>\n",
"      <td>16.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>75%</th>\n",
"      <td>218.825000</td>\n",
"      <td>36.525000</td>\n",
"      <td>45.100000</td>\n",
"      <td>19.050000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>max</th>\n",
"      <td>296.400000</td>\n",
"      <td>49.600000</td>\n",
"      <td>114.000000</td>\n",
"      <td>27.000000</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"    <div class=\"colab-df-buttons\">\n",
"\n",
"  <div class=\"colab-df-container\">\n",
"    <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-08dad05a-fc03-4156-a921-40949d299ab2')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"\n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 -960 960 960\">\n",
"    <path d=\"M120-120v-720h720v720H120Zm60-500h600v-160H180v160Zm220 220h160v-160H400v160Zm0 220h160v-160H400v160ZM180-400h160v-160H180v160Zm440 0h160v-160H620v160ZM180-180h160v-160H180v160Zm440 0h160v-160H620v160Z\"/>\n",
"  </svg>\n",
"    </button>\n",
"\n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    .colab-df-buttons div {\n",
"      margin-bottom: 4px;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"    <script>\n",
"      const buttonEl =\n",
"        document.querySelector('#df-08dad05a-fc03-4156-a921-40949d299ab2 button.colab-df-convert');\n",
"      buttonEl.style.display =\n",
"        google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"      async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-08dad05a-fc03-4156-a921-40949d299ab2');\n",
"        const dataTable =\n",
"          await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                    [key], {});\n",
"        if (!dataTable) return;\n",
"\n",
"        const docLinkHtml = 'Like what you see? Visit the ' +\n",
"          '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"          + ' to learn more about interactive tables.';\n",
"        element.innerHTML = '';\n",
"        dataTable['output_type'] = 'display_data';\n",
"        await google.colab.output.renderOutput(dataTable, element);\n",
"        const docLink = document.createElement('div');\n",
"        docLink.innerHTML = docLinkHtml;\n",
"        element.appendChild(docLink);\n",
"      }\n",
"    </script>\n",
"  </div>\n",
"\n",
"\n",
"<div id=\"df-677aa786-950a-41f7-bb23-566e62ec71e2\">\n",
"  <button class=\"colab-df-quickchart\" onclick=\"quickchart('df-677aa786-950a-41f7-bb23-566e62ec71e2')\"\n",
"            title=\"Suggest charts.\"\n",
"            style=\"display:none;\">\n",
"\n",
"<svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"     width=\"24px\">\n",
"   <g>\n",
"       <path d=\"M19 3H5c-1.1 0-2 .9-2 2v14c0 1.1.9 2 2 2h14c1.1 0 2-.9 2-2V5c0-1.1-.9-2-2-2zM9 17H7v-7h2v7zm4 0h-2V7h2v10zm4 0h-2v-4h2v4z\"/>\n",
"   </g>\n",
"</svg>\n",
"  </button>\n",
"\n",
"<style>\n",
"  .colab-df-quickchart {\n",
"      --bg-color: #E8F0FE;\n",
"      --fill-color: #1967D2;\n",
"      --hover-bg-color: #E2EBFA;\n",
"      --hover-fill-color: #174EA6;\n",
"      --disabled-fill-color: #AAA;\n",
"      --disabled-bg-color: #DDD;\n",
"  }\n",
"\n",
"  [theme=dark] .colab-df-quickchart {\n",
"      --bg-color: #3B4455;\n",
"      --fill-color: #D2E3FC;\n",
"      --hover-bg-color: #434B5C;\n",
"      --hover-fill-color: #FFFFFF;\n",
"      --disabled-bg-color: #3B4455;\n",
"      --disabled-fill-color: #666;\n",
"  }\n",
"\n",
"  .colab-df-quickchart {\n",
"    background-color: var(--bg-color);\n",
"    border: none;\n",
"    border-radius: 50%;\n",
"    cursor: pointer;\n",
"    display: none;\n",
"    fill: var(--fill-color);\n",
"    height: 32px;\n",
"    padding: 0;\n",
"    width: 32px;\n",
"  }\n",
"\n",
"  .colab-df-quickchart:hover {\n",
"    background-color: var(--hover-bg-color);\n",
"    box-shadow: 0 1px 2px rgba(60, 64, 67, 0.3), 0 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"    fill: var(--button-hover-fill-color);\n",
"  }\n",
"\n",
"  .colab-df-quickchart-complete:disabled,\n",
"  .colab-df-quickchart-complete:disabled:hover {\n",
"    background-color: var(--disabled-bg-color);\n",
"    fill: var(--disabled-fill-color);\n",
"    box-shadow: none;\n",
"  }\n",
"\n",
"  .colab-df-spinner {\n",
"    border: 2px solid var(--fill-color);\n",
"    border-color: transparent;\n",
"    border-bottom-color: var(--fill-color);\n",
"    animation:\n",
"      spin 1s steps(1) infinite;\n",
"  }\n",
"\n",
"  @keyframes spin {\n",
"    0% {\n",
"      border-color: transparent;\n",
"      border-bottom-color: var(--fill-color);\n",
"      border-left-color: var(--fill-color);\n",
"    }\n",
"    20% {\n",
"      border-color: transparent;\n",
"      border-left-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"    }\n",
"    30% {\n",
"      border-color: transparent;\n",
"      border-left-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"      border-right-color: var(--fill-color);\n",
"    }\n",
"    40% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"      border-top-color: var(--fill-color);\n",
"    }\n",
"    60% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"    }\n",
"    80% {\n",
"      border-color: transparent;\n",
"      border-right-color: var(--fill-color);\n",
"      border-bottom-color: var(--fill-color);\n",
"    }\n",
"    90% {\n",
"      border-color: transparent;\n",
"      border-bottom-color: var(--fill-color);\n",
"    }\n",
"  }\n",
"</style>\n",
"\n",
"  <script>\n",
"    async function quickchart(key) {\n",
"      const quickchartButtonEl =\n",
"        document.querySelector('#' + key + ' button');\n",
"      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.\n",
"      quickchartButtonEl.classList.add('colab-df-spinner');\n",
"      try {\n",
"        const charts = await google.colab.kernel.invokeFunction(\n",
"            'suggestCharts', [key], {});\n",
"      } catch (error) {\n",
"        console.error('Error during call to suggestCharts:', error);\n",
"      }\n",
"      quickchartButtonEl.classList.remove('colab-df-spinner');\n",
"      quickchartButtonEl.classList.add('colab-df-quickchart-complete');\n",
"    }\n",
"    (() => {\n",
       "     let quickchartButtonEl =\n",
       "       document.querySelector('#df-677aa786-950a-41f7-bb23-566e62ec71e2 button');\n",
       "     quickchartButtonEl.style.display =\n",
       "       google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
       "   })();\n",
       "  </script>\n",
       "</div>\n",
       "   </div>\n",
       "  </div>\n"
      ]
     },
     "metadata": {},
     "execution_count": 10
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "\n",
    "**Basic Observation**\n",
    "\n",
    "---\n",
    "Avg expense spend is highest on TV\n",
    "\n",
    "\n",
    "Avg expense spend is lowest on Radio\n",
    "\n",
    "\n",
    "Max sale is 27 and min is 1.6\n",
    "\n",
    "\n",
    "\n",
    "\n",
    "\n"
   ],
   "metadata": {
    "id": "YfUIQ8j_ZKas"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "sns.pairplot(df, x_vars=['TV', 'Radio','Newspaper'], y_vars='Sales', kind='scatter')\n",
    "plt.show()\n"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 268
    },
    "id": "CoHHKCzNbCBz",
    "outputId": "ef945815-7458-453c-f7af-478e8cf70cf7"
   },
   "execution_count": 11,
   "outputs": [
    {
     "output_type": "display_data",
    "data": {
     "text/plain": [
      "<Figure size 750x250 with 3 Axes>"
     ],
     "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAuUAAAD7CAYAAADNeeo8AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90
bGliIHZlcnNpb24zLjcuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/bCgiHAAAACXBIWXMAAA9hAAAPYQ
GoP6dpAACVoklEQVR4nO29eXxTVf7//0qzNE1LF1rKom0ppMoORRahLSAyIgIDyOiIfD7TAupnBuoy6iioyOaI
y4zjIOp8PiObv6/ijAug6DijIFtRFChL2aRlKUqhtDTpkiZpk/v7o9zLTXK37En7Fn8z3mf93kvKoZ
hGBAEQRAEQRAEETZiwt0AgiAIgiAIgujokFJOEARBEARBEGGGlHKCIAiCIAiCCDOklBMEQRAEQRBEmCGlnCAI
giAIgiDCDCnlBEEQBEEQBBBFmSCknCIIgCIIgiDBDSjlBEEARBEARBhJl2r5QzDIP6+npQjSSCiCxINgkiMiHZJIjw0O
6V8oaGBiQlJaGhoSHcTSEiEkIggfJJkFEJiSbBBEe2r1SThAEQRAEQRCRDinlBEEEQBEEQBBBFmSCknCIIgCIIgiDCjCXc
DCIIILGaLHTWNdtRbW5AYp0VavA5JBl24m0UQhEJIhqMXenaEP5BSThDtiIumZjz98RHsPl3DvTYmJw0vzRyE
HslxYWwZQRBKIBmOXujZEf5C7isE0U4wW+weCwIA7Dpdg4UfH4HZYg9TywiCUALJcPRCz44IBKSUE0Q7oab
R7rEgsOw6XYOaRloUCCKSIRmOXujZEYGAlHKCaCfUW1sk32+Qe8giPBCMhy90LMjAgEp5QTRTkjUayXf7yT
zPkEQ4YVkOHqhZ0cEAlLKCaKdkJagw5icNMH3xuSkIS1BF+jorqRpRW1qHiSiP5ORKEjwRDlpTIMBGZtMd
nR+tF6FExDMOEuxHBpL6+HklJSTCbzUhMTCbzUhMTAx3cwgiqFw0NWPhx0WZg+AEIiozAMkmEa0EM8uG
mAy/PHMQuodITkk2fSMSnl2goEwy4YGUcoJoZ7B5chusLeik13IWmuKNpYKBSGNy0vDGrNyQ59Il2SSiEbPF
HnRZEpLhUMonyabvhPvZBYJQjHFCGMpTThDtjCSD5yjQUd0omxWMwBBSGO
22FHdYEOrg8HaouGCcgQAep0aZot/lsRoL9Ue7e0PJN72RSj6Tu43KJNM+CClnCDCgL9BNEo/L3RdnjEVq2bl4
pGNpbDYHcg3pkKvpUMzghDjoqkZT390BLvLxeWIfW3rkSocuWDyOSAu2gPsor39gcTbvghF3yn5DTaTzC4Rn
/JozCQTLYR1JV65ciWGDx+OTp06IT09dcrhk3bhxUKpXLv9/+rdhajFB+I+/5ZiVfl7supLyWqwr
OYu5+dnIM6aiKC8bSz89RumuCEIAs8XuoZADrnIEtCnkc/KysXbPWZ9Lq0d7Zob38g8bvYvQtF3Sn8jyaDDS
MHeaR4ZDPJdNRTj1AQVkv5zp07sWDBgwfPhy7m+D48eOIj/nrnvwwQexl9J19MLF/V/z72GUXq7kvAXHRHmAX7e0PJ
hJEQPA3iEbp56WuKymvxcJJfQCAs/Q+SFLTaPdQyAEXOQLaFPPJ19MLF/V/z72GUXq7kvAXHRHmAX7e0PJ
N72RSj6zpvf6JEchzdm5UZ9JploI6xK+Zdffuny9/r165Geno4DBw5gzJgx3HmdTodu3bq52Bw4m49tXBw
aKLLdfMIIij4G0Sj9PNy1
1242ozV28sV/y5BdETk5OinumYAcJElFm9lKtoD7KK9/YHE274IRd95+xvtIZNMtBFRjqRmsxkA0LLzZ5fX33vvP
aSlpWHAgAFYtGgRLBZLOJpHEB74UvHM3yAapZ+Xuy5W4yr+FLxDEJ4okSN3WWLxVqCOOH1gHm+fZSiefTSPr45CxAR6Op1
bYdCpXa6J5PYHGm+fZSiefTSPr45CxAR6Op1OPPbYY8jLy8OAAQO41++++25kZWWhR48eOOKIJ5++mmcO
nUKn3zyieD32Gw22Gw27u/6+vqgt53omPgaROOvED8SSz0tdl2dMRc/2UGyJfK1e/6y8km0Q0EEEoHQJkJi/Gy2erzni
ylJehQkJMm6GJQEEQZ9Vc2lQSUd7QAQW/n+VAEV1IAZ+QTMZbyBQsWoKBqysDB988B+VvJdJIAmTpyIgQ
MHYvbs2Xj33XexacqzKXZwxEgdt1+byANG9+1
19INolIRerEi5U3dznKM6bi4fE5yDem4VSVqxLrj0wtuM2IPGOqx28tuM3o9XcpxR/ZVBJQ3hEDBL2d55Ve78
vprK9tIkKPimEYJtyNKC4uxpYtW7Br1y5kZ2dLXtvU1ISEh3AR8+eWXmDhxosf7Qjv+jIwMKhdMBJSK6kbc/tpO
0fe3PT4WvdMTJL/D23LM7rllE2I1aLK1or5Z+PMXTc14fksZ+nRPRG5GMmytTqQYtMjobIC91Sn6uWBBsklEI
kpPvNg85ebmFhh0asRp1WhlGKhVKsTLyKJSKqobMXX1HszNzcx+ZkNlYTg9ILJqzdcxafeFefLziu+4I9sys2FXz5ag
O5J+g6r8Pk6z7tfb7bYcaneip/qmqFSqbg8+cOyUrxOmehtm4jQEVb3FYZh8PDDDDD2PTpk3k3YsWOHHHIAOAIcOH
QIAdO/eXfD92NhNhYxMbGBKZvAmikVjIcenXXxXKT51quvT1S7vDcmJw1/uuYTzDcmJw1vzMoV/FwwIdO/FwwIdkkIg25E683Zu
W6nDwlGXRey6I31FtbYYLE7bYLE7BINGgeAFSvojm3JzobXF0aEVPm+DJYWul8uT7z5WA90mInSE1X1lwY
h/fffR6dOnXDs6FD6OnXtXPp0iXVcinVcunQJ1HiVcunQJzc1t0ewVFRVYsWIFDhw4gPPnz+PLL79EUVERRo0ahUGDBoWzqQThgzRZhKgbTOZ/zPJVbnvb/e/oGnn1t/j5J8zv5R9qnf3J8j4RFhTp4W9xaoqCMV4dFXYPXAAE/aTjcOVSb4sqdUTwQyQSqcDFJTrfP8kwAJTLaVZvk7N5FGsqmVB5sCXkrkzKNp/cqXHu5O7BDrrjNAWu1WPNwwKmY
VRkvqKIDo63spJsHNJR2MgXjS2OZrgxpxEnya09sPYbWUv/322wDaCgTxWyTxsk+KbUHKFj7pdpnxBPsrakorKolNwzZ5VMqgxnbfahfZsXmsw3CV
```

TEzIyMjBz5kw899xzYWgt0ZHxcB3Ra0IWMONL/lpvLflUFptoL3gzlr2Vk2DnkpYKxCvISYNGrYLZElm5vjti8GA
o50u5ehNz89o8DDpSuklviLa1LezuK1JkZGRg505xXzWCCBRSgit0XP2Lvul4YfoAPLe5zGUxCkbAjC+uMt5Yr6
gsNtFe8HYse2vlDXYuaTYQb+HHR1zmlTxjKgpH98Skv+72yYc4mIi1ub0GD4Z6vpQbc7ZWJwAgMU4bdQposI
nGtS1iUiISRLiQEtx4nVrwuPqra37ar94zGI3W1qAGzPhyPKzUeuWNTy1BRDK+jGVvrbyhcNVgKymyQX0AUH
rBxKUWjETZ7CjVH8MxXyrJk/+LvunQqWNQvLE0qhTQYBKta1vEpEQkiHAgJ7jVDTbRo8OvTlSj0dqK3ukJGJKZ
gt7pCUERclZxEELseFhp6ivyPSfaC76MZW9TxPkii75S22jHvA37MW/DfqzeXg6L3cG9F4mymWTQBX0uDDfh
mC+lxlyeMRXVDTYs/WV/LNp0NGixDtFItK5tZCknOjRygmtqDvxxtbdHjL4eDyuxXlFZbKK94OtYlpMTvrwmx
Wnx4oyBeGbT0aC6atQ02oMy97QHwumiEY75Umz+L8hJw4ppA5Bi0AY91iEaida1jZRyokMjJ7jxbmWi3fH2
uNpXHzdfj4flUl9R5gSiveDPWBaTE7F4kpV3D4S1xRk0V416awtiNdIH2R1RNsPtIxyu+VJu/j9T0yT5+UhVQINJ
tK5t5L5CdGjkBDdepwnYcbW/6dSCcTwcyuN4gggmgR7LYvL61YlqLPrkKNISdEFz1UjUa1F6weRR2ZOloAPKZr
DTUSohnPOl1PwfrQpoMInWtY2UcqJDIye4yQZtwMoSR6KPG5VdJtoLgR7L4ZTXtAQdTlXVY05etodinm9Mx
coZAzucbEbC/Bmp82W0KqDBJFKflRwqRi4vYZRTX1+PpKQkKuVNiHLR1Ozir2fQqbF4Sj8MzUyGxe4QLGmfo
G/729ys3K+xtLIOM97aK/r+5vmjMSQzJaD3ppRwlF0m2SSCQaDGMiuvBp3apey9XqvGwco63NE3HYMyxOX
VX9/ni6ZmLNlShpu7J3K/nRynRVaqATekGLy+H2+IRNnkz59CzyQjJQ45XTuFpC38MRYfq4FOHQNTsx0J+vClIX
Rfx4DrCmj3Dph9hSUca5s/kE850aExW+ywtjjw3JR+cDIMrHYHkg06LN5chkWfHOWu45fRvmhqxpMfHvbar
zGSjxip7DLRXgjUWE7Ua2HQqbFqVi7WlZzF6u3l3Ht5xlT8auiNop/1xvdZTHnvkRyHP90zOKoUimDCzp9iz6Tg
mgIaCt9ydowF28fdm41dR0lL6S3RtraRpZzosAhNqCvvHogvjlR5lDQG2ibbV+8ZjGc3HUUfnvWKtZydqqrHn+
4ZLDoBmC12PLyxVDQncqTmTQ0WJJtEJGO22PFF2SVsPXIRJeW1Hu/zZZavPHWO1+G5TWWicwhfzsMduCh
GJMomO38OykhGaWWd7DMR+45AZW4xW+weecGVtkMJkTo2iOBClnKiQyIWNJTeKVZwMQXa/BbrLHbc
NyJT0HI2Jy8btU12j0WaP/nLpTakimwE4T9CcgTA61SkQzOTXU7M+Ow6XYPaJjua7A6XuWRN4TDJOYRNTxe
txU3CBTt/nqtpcpl7+Uil/wu0khvMNIThGBtiaw+tSaGFlHKiQyI2obIli8VosLZiXclZDysN+/fSqf1lJ3+xI0ayjBCE/
wjJUUFOGhbcZsTc9T9wRXiUyBa/YI8QDifj8Vvyc0hbejrKLe09PZLjcMncLHmNUPq/YCi5wcyDHeqxIZb6c/GU
fnh2cxmtSSGEsq8QHRKxCVUuN3BCrEbw2BRoU8yFFmnANW1XkkGHtAQdOum1qLe2oKbJjsv1Vjy/pSys6b
4IItoRU74OnK/DmSuNeHfuCLw1eyjWFg3HoIxkLNlSJilbcnEgDifj8VtK84tHa3GTcJMUJ62MxsdqUFHdiNLKOl
RcaYTZYkdtU+AztwQzRiiUY0NMZm7unkhVQsMAWcqJDonYhMrmBhbzV9RrYrCmcJiLL/naPWc5i1qjrVVy8
hc67gbaLHmFo3tib0Wth3WOrGYEoQwhCyM/MPCZTWXc6+4uZ0Kwqebc40DYDE1OhsFbs4e6zAVycwibni
6SA78jGbFnArTNo/vP13kE6S/5ZX8YdGrRkw8x67qU24ZUO/xNQxjKsSFmlc/NSPbJTYjwD1LKiQ6J2IS6ds9Zr
C0aDrVK5eH3/cL0AVj+2TF8ffIK93qeMRWrZuXikY2lsNgdMOjUKB5v9AgCZRV3MUv67tM1cDIM5uZnC06E
ZDUjCHmELIxz87NlXc7EEIoDMejUWFs0HG9uL3dR/ti5YOHHR/DSzEEuvwF45kcOplLXnpGKzZl/zUWJz67TN
Vj66THRuRW4ruSyVnUGwNItZdjt9vz4bhtKYoR8JZRjQ8wqr9QNiwgspJQTHRKxCXVYVgp6djZ4+H0n6DV4dt
NRF4UcuL7ozs3PxpELJiTEalBaWecRBMoq7kLH3fzvmpuXLfheMKxmFMBDBItwjS0hC6OUxY91OZPCPQ4kx
aDDc5s9s6uwc8F9IzLxyMZSLJ7SD0un9keTrVUwPV0wlbr2jlBsjiZGhUmrdgtaw3efrsHvxvYWHAesksv6VQ8
Wye4i5H8uFSPkjwyEcmyIWeWVumERgYWUcqLdIjcpyuV15V9bUd2Ir09UC/5OSXktFowz4r5hGVj66TFRi9zi
Kf1gsbdKtlnIOhEMqxkFlRLBIpxjS8jCKGfxkwvmBFxzHVdUN4pmV2E31sOyUjDupi4uRVvMFjsqqhtd5iPKLe0
77vmnSyvrJJ9lrDbGY2ywSi4AbswWje7plduGUB7sQMiA0rHh7wZYzCpfesGEfGMq9si4YRGBhZRyol2idFJU
WlhALvBGr1XD7nDi65PiivvzU/pBEyNtfUiOc7U+BMMyQqnYiGAR7rElZGGUs/glxXln8ZObC5LitB73KTcfkbz5
j5wfdnKcTlTJrahu5J6Nv24bgZQBufUpEMq/mFX+VFU9XpwxEM9tLqOTnBBCSjnR7gi0YmC22BGnVXsEdPGt
MklxWpibpSfrZrsD2Wl6SV/B3ukJ2Pb42KBYzViLylWLHXPysjE4I9njPiiAJzQotW75awULtRtJJKT541sYa5vs0G
tjUGBMdfEPZinwweInp/yluClS4d6ohBJvx1sgx6cSP2wxJZe/0fLXbSNUMhDIcSVllaeTnNBCSjnR7gjkpChkiXA
P7lR6lNdJr4XF7sD824xwMIyLmwtbIrproh5dfSygJ7XAKbkPFgrgCS5KrVv+WsHC4UbCV24MOjXm5me7BD0
7Q1RAmlO+qhsxdfUerJqVCydcAy/zjKlYMW2A1wqGt0F4kbBRCQXejrdAj88kgw4vTB+AZzYddXG5yDem4oX
p0s+Zv9FSmj1HjFClM+SPKyFZM1lavBpXYhuWaCtTH+2QUk60O4QUg2GZKUgyaKFRx+CqxQ5caVRkxRGyRL
gHd/KP8qQW60S9BhU1TeikV+O5u/oBKsDc3AJriwNVZisMOrXP9yy1wMXr1KL3EQPgg4duxU91zdwpQKKX
x/mEcuSsW6/eMxiN1laYm+2wtToxOCMZB85f95VVagXj/477gn2+tgnqGBW6JuoDfm/siVK8ToMkgxav/vuki

```
38uu/n0Z1PgjXU1LUGHYVkpeGRjKebmZ2NuXjZsrU7EamJQ3WBDisH7se5+3M/27+heqYjVxKCmyc5dB7T/f
OSX662w2Frb6iwIBEc+/fERvDB9AJLjtFyfXK634lxNE2aNyMScvGzu9NGf0wOzxY7lW49jSGYK5vCec+kFE1Zs
PY4/3TMYgHBVV/5Ga+2es1g1KxeAdPYcMUKVzpAdV/yUn4GWNSL0kFJOtDvYSdGgU2P1/bl477vzGJKRjD/9
55THJCtllZGycJWU12Lx5H54MD/bJTBULGL+1ZmD0GR34I3tp13akG9MxeIp/fHkh4cxomdnnywScore4in9R
O9jd3ktihpsmP/eQa499w3L8LoNhDLkrKYV1Y24/5193GtCpxlKrKvs74RqwXbfFBaPNwpmsNjtp8uGt9ZVvkz
y799fv1j2uJ+fPs/9+9k2ted85JW1TVi06Sjm5mULugcBbc+8vLoRG/aew8szB4EB8PRHh12u549zX08PTJYWz
BqRCVurEyqVCser6jn3PINOjatNdo+NA/858edudhO3YJwRsdoYJMfpFLtthCqdITuuxFJ++itrRHggpZxod7CT4
tCsFKzbcxaDM1MEJy05q4ychcva4vD4nArApIHdUTi6p4tFroVh8Ozmox5t2FNeixVbj+HlmYN8tpjJKXomGV9
3fmDTnvJaPLPpKE3kQUJuTLk/K/6pDF/pMzVLV9NjfycUC7bQpjAYhUd89aENVoYT9vPFG0sFLcRsm9prPvLL
9VYs2tQ2p80emSV5ra3ViV2na7Djxyv44kiVR3+5j3Nv58KLpmY8t/moqKI/Nz9b1JLPPqdAjZNQpTNkxxUV+
WlfkFJOtDvYSbG6wYbXvz6NojzxohG7Ttegqt7KfY6PtxYus8WOpwSUBgD4/OF8QR9FADhYacLSXxrQ6mBQe
s19xJvAPzlFL17GLcY9sIkm8uAhN6aEgsyE8te3tDpxud6KRmuroBsH+zuhWLCFNoViGSxYVw9bq0N2rCv5HR
a5ewmWX6ySNvVOT2iX+cjrmuzcnCYXHMm+n94pVjadJOA6t0q5K5ktdlQ32FB51YI5+b0wODOFs47zFX2lc
hCocRKKVJfsOneiql7yOvcNDtWniGzCqpSvXLkSn3zyCU6ePIm4uDiMHj0aL7/8Mm6++WbuGqvViieeeAIffPA
BbDYbJk6ciLfeegtdu3YNY8uJSKdHchx+NjUDkE9xdeZKE178/ITHEbgSCxd/govTqT18gFnqrcL5yVn3gmWfHp
OsHgeIH90/O7mv5P3Fx2pE7yPPmIrSCyaP16PdzzWUeOvj7O2zAFzH8Pg+XaDXafDkPw+Jjpm0BB0KctJCUp
VPaFOo13puBMVcaeTcyNj+rW2yY23RcMHsR0B4xqxSf/H2mI+cP6dJBUcWGFPRpVMs3po9FOmdYlE83ij4/
IC2cc4/PRCb8zg3GJngdamCbHy8HTtKZD4UAZI9kuPQZJOufcHf4FB9isgnrEr5zp07sWDBAgwfPhytra145pln
cMcdd+D48eOIj48HAPz+97/H559/jg8//DqdXdxrCZGMDqdXdxjNTGCR+BSx5
CvXPMRV5rRJEEvbK0Wcy9wb4/U0f2kSpOooveLvunQqFTCGV+MaZiT3xPF75d6fC6a/VxDiT8+zvznVZCThsL
RPfHIRs9nAVy3NOYZU/H4L27CkZ/MKMrLxqyRWS5jmT9mlv6yPy6ZrZLtD8RzFrL+MwzjUXhE6Vjn403WoH
CMWW9O09pbFotE/XX1QSw4ssCYhvm3GXHf/33HPSux5we01WlgTw+k5rzrbjDiQfirt5fDoFOjS6dYaNQxkil
tvRk7kabYpneKVeQe1ZFSc0YzYVXKv/zyS5e/169fj/T0dBw4cABjxoyB2WzGmjVr8P777+PFGUWzGmjVr8P7772P8+PEAgHXr1qFv37
747rvvcOutt4aj2USUEK/TIM+YirKLZqwpHIbV35S7WOjyjKlYUzgzM35+7CkD4CFzMwgVc8yVVmNHkSoNNs
Dqa3LFqldmKMzVNklb4FVuP44tHCvD8Fs8iD4un9MPCT47gQKXJM/tEvRVHfzZ7fF80+7mGksv1Vo+ANcA3
H+cEvQbPbToqaD1sy6UdizWFw1B20QydRo3Pj1Z5pPdjFR12DKvQFvdQYEwTdBnwJUe3O2aLHU6GwZrCYV
CpVJzCo1apUJSXDQbXFSVvXWmUZD9ivy9cY7a9+osrISVex81pFrvDJcMNAHRPisPhC3WYt+EHl3EtFidRcK1
OA5sVSMo1SM4N5vcTbsIH31fipZmD8OqXJ0V9zb1JaQsEV7H11a1EqQ97R0nNGe1ElE+52WwGAHTu3BkA
cODAAbS0tGDChAncNX369EFmZia+/fZbQaXcZrPBZrNxf9fXS/tbEdGH0skr2aDFw+NzUFnbhLe+Kfew0LUp0
Crc0jOFe03oGFPIwsWvAOfO7vJaPHpN8bW2ODC6dypS43V4ccZAjxy6cpypaeIyo4hZmCx2B+qb7S6KXpxOj
YOVJpRfaeIWJCGF6P0HRrr8HUw/1/YkmxdNzThX0ySaccIXH+dl0wbA2uppFS6+zYgYlQoqlQpjcrpgxWfHBM
cy0KbosGM4NV6HlV+cQGFeTzjBeGT9WTljoF/PWcqKbXM48ft/HHLZCMbrpJcbtt2sfNtaHZLZj+bl9wIQOt9s
sXmnPfiL+yKbXRP1gnOaXqtGrCYGDBj8LHJS4+5WwvYXP02nu2sQ/7TTIDOWzM0tWFs0HK98edJjvpVKaSu
Hr4qt3Jrlr/VdiXtUIFJzkj968IkYpdzpdOKxxx5DXl4eBgwYAAC4dOkSdDodkpOTXa7t2rUrLl26JPPdZ6Ugd
dXCJMeDN5JRl0yOpsQJxWjac+Pir4fbvLazAn//rioPQYU26Cq7e2YN6G/dzf+cZUvHT3ILx27xBcbbKj3tqKRL0
GMSqV5PfwA6jELEwAEB+r5RQ9s8XOWfHfmj1U8vv1WnXQKoi6015kk7WWzRqRKXmdt36q8To17hrYHU
W8zD1lF81osrdi1bVUmmsKh4luBFhFhx3DSQYdlk0bgCVbypCbmcIpx8lWmSlGnBDisGr9vGRsmLHqFRY/sv
+sNgdLuN0TeEwye/spNe6yLfc2E3Ua7D9ibFIDYFiIDfvRLu/uK+ymZkajz/fOwQmix3qmBgs+7TM4zRSzFUlKU
6LzfNHi/YX3zXIPR5BbiwBQIO1VdQAUlJei+fcUtoqwRfFVm7sBMr6Luce5W9qzkhz22mvSIdMh5AFCxagrKw
MH3zwgV/fs2jRIjpNZu7fhQsXAtRCIiIaIMNzu7fhQsXAtRCItzITV5mi2eauO7JcWh1Sge7adRtirE3x5hyVj/3CXBPeS0WbTqKWE0M+
nRPxIjszujTPRFdE9v8AYUQCvwrKa9Fbkayy2vu7eZbc+SyIiTFadE7PQFDMlPQOz0hqIpEe5FNtn/9Lcct9L2LPj
mKeRv2Y/57BzFvw360OBis2XPdD1sucBOAy1jokRyHP90zGDOG3IDUeB1u7toJ/Xsk+qWQs20VPSk6XYNWJ
+MxrtlgQCHG5KQhQa9xkW+5/rW3OkOikCuZd5IMupDJUTDwRza7JurRLVGPZZ8dE0x1uK7kLObmewZbph
h0kv3FugYBnvEIUmOJnTfNcqlgBVLayuFLRi65saPE+i6H2WJHRXUjSivrUHGlUXAt5PenO3Jrny9rL+EbEaGUF
```

xcXY+vWrfjmm29w4403cq9369YNdrsdJpPJ5frLly+jW7dugt8VGxuLxMREl39E+8DXyUtOgU416LDxwZFYeb
ey4/yLpma0Op2Si4JOQKHYLdBG9vjbfbLMM6ZiTl421u456/E9fMVM6Jicb82RU4RC6fPaXmST7d9A962QFS
43I9nF7UROUb0xJU4wC0SgFUY5i2GTrdVjXK/dcxYPj89BgdtYZ8dwk63VRb5LL5hQYBTfsO49U4vaJrsihcQfA
qE0RTr+yqZcoTU5Q4IQ/LnRXQ7W7jmLOXnZHvLHnzcDvWkGvFdslYwdf91KLpqaUbyxFLe/thMz3tqL2/+8E
w9vLMXFa9nHWMTWGiWuVvz7MOjUKB5vxJrCYXhr9lAU5WXDZKFsXYEirO4rDMPg4YcfxqZNm7Bjxw5kZ7
vupm+55RZotVps27YNM2fOBACcOnUKlZWVGDVqVDiaTIQRXyevmBiVaLquPGMq/nXsElZvL3cpSy+VF/fpj4
/gv27NwpxrPpHuAXdz8rJRJeJLKdRG9+NvvVaNrUerBI98AaBXWrziY19/S0YTnrD9K5pxwse+FbLCuVvGpVLP
jclJQzeeT24wUWIxFHPrWC3i6lFaWefyHWv3nMWm+aOxfOtxQRlb+PERzBhyg0fAdaCP1APhi9vekesj/jj2Rj
7YMfRjdaPL6+6BpQadBhZ7K0ovmLh5s/SCCQU5aYJKsa8GCW9jCOT6xdzcgqQ4391KvHV98dXVir2PUFUI7si
EVSlfsGAB3n//fWzZsgWdOnXi/MSTkpIQFxeHpKQkzJs3D48//jg6d+6MxMREPPzwwxg1ahRlXumA+OoTp4l
RSSrQbBq6Xadr8PTHR3DXwO5Y9Ml1H3T+Is9aDIpG98TDvEWB9QFmF4U3rilr7ui1asGiKfxJ0cEwOHzBJKiQ
j8lJQ/ckveQkys8IIZQVIbOzAemdYkkh9xF+/z7iNgaSr7kDdfVBORbK5OFu7YuUTZZU1pGCnDRo1CrOrUOoT
UKvucu3xe5Alcnq4g/Pl7G5+dlY+ql0lcZA9Ie/vrgdAbk+yugch7dmD/WQD6X5vjsLPEd+zMKawmEuMTwAcL
KqHi9MH4DFmz2zUvkjK94otnL9Ym1xcC6MvmTw8SXw1JfUnOx9hKJCcEcnrEr522+/DQAYN26cy+vr1q1DU
VERAOAvf/kLYmJiMHPmTJfiQUTHw9f0Y2wWCnZxj4/VoMnmalVhYRVuPvxFnu+6kJspnOItX6QITL4xFVuPV
rmkcWOVfX4QDWuNYBjGJVBJ6WLibs1hFy/2893JmuEX7v3Lf57uGST8+V6gbZzxU2nyN1kLxhmh16qRFBf6
wEK2rULZVwpH98Skv+7GsKwUryzWQvK9v7IOpZV1gnI2qldqSMqLd+S0h0qRK4r172OXuUwnrHx4Ezgotw
msbrC5vJZnTMV9IzIx8+29eOKOm/Hs5L5otjsCFoSrVLGV65e9Z9p87tksNt5uHkJ1isPeRygqBHd0VAzDMOFu
RDCpr69HUlISzGZz1PqwEte5aGoWPTqUUjb5n3tr9lAuzaAQYu9ve3wsAOD213a6HOO5uy68MH0AVmw9j
q9PVHOv5xxtTUXTNKs/fBIzJScOr9wzGkx8edlmc2PRfo3ul+qx4sVaoSM0IEe2yGaz+5X9vYpwWOnWM6IIdzg
2W2WLHF2WXkJOewAXVlV4wuRRmGZOT5pX1zF2+DTp1m+/qNxUueanzjKl4+LYc3Pf370S/a/P80RiSmSL6
vjf4Ou9EK77IplAfFeSkYdkv+wOAS1AuP0OUO2JjRuoZGHRqVDfYUHnVAsD/cRhIxFKHzuGtB2wWIW/nk4rq
Rtz+2k7R97c9Pha90xMCdh8nquo9TiT4BFLmOiqklBNRh6/KED//8V2r9ohe534UyirIdw3oBo06Biu2Hucs2v
wqoSkGLbJS42FrcaDFyYABAxWjgoNhcKXBBoeTwQGBanJfPlqAO/+6W7Q9gZxYI4lIl02zpa20e6uTgZNhYLG
1IsmgC0tu3kjcYLEKgZDrAJ9tj49FWoJOcX5js8WOn03NOFdrQXqnWJy50oiLZisG3pDk4sKS1zsVs/6+T/J3Ayk
3kfgMgoWvsqmkj8wWO1cUTazCptizk/r+UCqo3vLj5QZcuGpxGb/8e/ZWmWX7wdxsh63ViZKKWo8+DMZG
5PTlBvziL7tE32+va1UoiZg85QShFF/LVfPzeIsehRpTkd5Jj+LxRi7zCT+wxd21hD3K+0XfdCye0s/DBcXdki6Ut7f
e2irZbgokCz0XTc14fksZ7huR6fEMw5GbNxJLtLNH53JpGk3Ndiz97JiLpfAXfdOx9Jf9YW1xeijqSYY2BX7+ewc5
Odpy+CJe//o09/mCnDTMGp4RUreSSHwGkYZcH100NePpj454nHq4z4lic57U94c6INebQjpqlUpy4+pNXIKQ
5T1foEppMGJM0jv57v9OKIOUcqLDIRZBn2dMRWFeNn79f98iNzMZq2bl4thFs4tSxvfpnc/z6U3Qa1xcUMQ
CYtqqiAIfPHQrfqprhl6rRpcEHQw6tWBgJ0CBZKHEbLFzx+CP3n4TXv7yhMczDHQgYbTCBn/JpZ6ztTg9XLN+P
SITT318RHSzozSYtj1U0+womC12D4UcEC6K5sucF8qAXG8L6QQqLkEs28qe8lqoVCpsWZCHGJUqaKc47aWC
bSRDSjkRNQSyxC8bQe/uh8haGtiF4uk7++AvX512+SwbOLl6ezm+fLQA9dYWtDhdFQ+pgJjd5bUoarBxfusFO
WlYWzQcc9f/4KGYk/UhdLgvtGsKh4lWBKSgpuuKhlSaxoKcNOw94/q62IbVfbOjJJhWKhMGlQSPHFiXFXeFnK
WkvBbz8nsB8H3OC1VAri8VOH1VZt3HsNPJSBbtilGpgu4+0h4q2EYypJQTUYG3lgmlqbZqGu2ix4ol5bVQx6gk
23Wmpgnz3zvoURJc7kif/z57T4un9PNIxRgs60NHUFi8uUehhVbuGbLH4R2hL4VgFY0lW8oEU47mG1OxdG
p/TF3tGr+hNIOD0sVfyKWBSoJHDuyzmDUiU/I6jVrlc45/QFrxfWXmIABtfuf+yqkvaQgB75VZoTG8pnCYZNvq
LHYuFWkwIVeu4EFKORHxeGuZ8GZBlvdDlPb3Zo/u3Y/w5Y703d/ffboGz0/ph22Pj5WdsP1VAturwsLvl3idBgc
q67Bi63GXDAxi9yi00CqpCNhe+1IpPZLj8MKMgTh7pRFP3nEzFk5Soc7SglYHg4OVdbjSYPU4/VG62QF8W/x9
sWQSwYH/LNxTzbqTlhCLF6YN8CubjZji22R3BKzIlFLfdbF5WmlSAqExLPu55hY8vLG0w8w/7RFSyomIh1WY0
hJ0eHnmIKQnxqLR6kAnvQaX66242nTdMuDtgiznh6iRqQbK5iN3P8KXOtLPE8lj3mRr5SLw2Qn9TE2Ty4Turx
LYXhUWsbRj/OAntjjU4in9oI5RuWxmhBZaObcMdQxwoqoec/KyMTgjmct+EGl9abbYYbK0oMneiia7A8lx2o
AWj2q0tuI+iSwo7lUVg1H+nI+vlkwi8PCfhdyc2OpwIjlJOse/0hNQlnprCzRqFZ7bVObhOrPrdA2WbCnDCzMG
otHaqtjIIbdmJMbJb9bl7kNsDCtZVyJt/iG8g5RyIuKpt7YgLUGH9x64Fcu3HvM4In9h+kDub28XZLniDnvKaxRV

A3WvtMj+rQJc/JLdP8eHVUbEJvQXZwz0yGLB3pfSSbg9KixiGw2hALLdp2tw4aoF8zbsd1kkhRZaseqZBcY0LB
hnxF2r9nBWYKENQLj6kl3wG20tSI7TocpsxRvfnPbIpy9WFtvbkxgpy+HaPWfx2cP5WPbpMdGCSHwC4fsb6i
wcHRGpMcJ/r9V5PeOymDyxc6JeGyOfuUWBQUIoNkTIl50NOH7yn4dcqsLKGTnkfNfjYzUedSeA65uAJVP7Y
9Gmo5L3ITaG2T6MUalE856zvxWNczlBecqJIKN0gZe6rqK6Eedqm7BWIDgMaFPM/3zvEHRN1KO0sg4z3tor
2p5PF4xGVmq8y2/pNTFY+ukxfMUr9jO+Txc8NuEmLq95eic9tGoVqsxW3Jgch69PXgbDgMudrNeqceQnE1Q
qYGK/brC2OJAYp0V8rAb1zS2ovGpBUpwWP15uwAufnxAM6Hzj2oIlVlSjICcNgyX8cf/9WAFu7iY9xuX6J5TFH
wIlm3L5iT8tzkNNox0nL5nRp1siuiXGwdzcgk56DRptrejbrRMA4OGNpR4LrUGnxuLJfTE4IwU1TTbE69Q4dU
n4GeYZU5GbmcI9n3AU0uArJMXjjeiRpMfnR6sE5UYoj7EvJzFC/c/P4Z8Yp0XytSJI5mY7EvSBL4jEnz/idGpsPV
LlkbeZxZdcyh0tbkBKNqXGiArAU7y0sB88dCuuNNi4OfLozyaXeTOjcxz+fewyTlbV48/3DJaM+SjeWIoD5+tcak
PotWpcrrcir3cqkuLaNtbu8ydbDM69rkRGigEvf3lCdHMoZORgx0GdxY4Wh2t+cHb8WuwO0fmoeLwRhyvrX
DYBQr8pNacZdGr865ECNNlbca7WIpj3HKBCPtEKWcqJoOGrZcP9urQEHawtDkHFAmizRNc12aHXxCBOp8Y
HD96K+Fg1ABW2n7qM/915Bha7AwadGolxOkHfwhdnDMSiu/qivrltUW9uceLlL096WHQeyO8FnUaFuwZ0
x/LPjrnkTs4zpuLh8TnonqR3mcwbra2Yt2E/Z+0fmpnsshC0WfsHcJOxVHS9lF/mT3XN6Jao9+voNRrTL8pZRn
+qa8s5/sFDt2LJp8InLT3T4gWDxHIzk9ElUY9f/W0vLHYH1hQOwzObygR/p6S8FnOvnaoA3velv4qf+4lBbkYy1
y4h3K1pl+utOFfThFkjMjEnL5sr6iJ3EuNuOeTn6OdvIN1l35ugN6m+UZK3md8Gby3xHT1ugI/QqZRBp8agjGS
cq2kCA2BOXjaG9UzBwBuS8MqXJwVPCh/eWIrczGTkZqagtLIOf5w+UHKs1zTaceB8neC4yjOm4pbMFKzYehj
PTu6HA+frXD4bq4kRHJPeZlcSGgcFOWn4tDgfDsaJ5DgdZxgSQ2mQs5Q1flhWCpINWjgYBser6pGbkYx+3RO
x+v6hLoWYonEuJ0gpJ4KEUt9lpdedutwg+Xv11hb88fMTHkUpim8zYtANyVjw/kEsntIPizcL+xY+s+ko3piVi15
dEnC53orl/zzkocyUVpoQp43B/nN12CpgfSwpr0WMSoXV1yzeLA6GwZrCYdCqY3C1yY45edl4IL8XLC0Ozsqx
Yutx/OmewbIKplyQnNyRZajShoUS/kbD3Rqm16qRlqDDK78a5KGQA20buuc2H8Wf7x3iEiRmbm6BtcWBv
WdqXRQ7pVl1vO3LQCh+7q5Jcm0FrrtytBV1OexiwSswpmLz/DycrW2CVh0Dk6VFcGy5Z71QmvKQH/QmFk
Mh1zfxOrVo3mbA1XXJl2xG7TUGw1fcx5jYBuzFGQOwbs9ZD6WXHROLJ/fD0KxkVJmsuCE5DimGNhkW23z
VW1skaz+s2HoMgzNTsHhLmcszB9rcpZ6b3Nfjs94EHIuNg92na7Dk0zLkZqbgyAUTXpo5SNLwofQ3kwxt8VM
7fryC9E6xLqcCt93UBUkGHRqsrSitrPPYoKyalYt/fF8Z0XN5Rzt58gZSyomgoNR3Wel1crv+eJ1GtCjF5IHdsXhK
PwzNTHZJOSj2W43WVkHr4tz8bLzxTTnm5mWLWh93u1lYLpqaseKzYy7KDmstevLDwy5WvJpGu6wlOzlO+
H02yKdLgnR+5vZW/MFsscN5bdMTo1IhNUGH17/+0WWhKjCmYtHkfrInLV2vnTLwn93fdla4PCOx/meJ1cR4
3ZeBUvzcN3RyAZVAmzWf+323/tldXotlW49xLjlSfuj8DY2t1aHIGsjii9LN9s3iKf1E54895bV4bnI/TOiT7nMu5f
YYg+EP7mNMTFHumqgXdNEA2ubl30+4CTPe2othWSmcrEiNg0S9Vr72Q16bMv5Afi8UjzdyG/M4rRpdE/Ue
J1zeBBxLjQP2hGz19nIs/PgIXr1nsKjhI0lm/uD/JgPgiyNVLuvamJw0jL2pC8wWOxZtOiq4QVEBET2X08mTNKS
UE5L4uqNVGmyl5Dqzpc1yKRYcVmBMg16rxmMTcvB/u864KFHshJnZ2YBGm3R6w9omG1ouOdEokgaRXRR
mj8yS/J6rFjvSLHYAEFR2hAIQ2XvNTosXndDzjanonqT3iL5nlfyFHx/Br4beKJv6q70UfxBzWyjKy8Z3Z65y42B3
eS0WWKTHWb211TVA0qCDvdWJxybk4JnJfaFWqaCOUSE+ViP6fApy0mDskuC19TRQip/7hq70ggk9BMYLv
70OhkFVvVVW4QDaNpxSmwR2QyN1fA8os0CySveKaQMk+8bULP1crS0Ov/xqQxU0Gi2WQ/cxJqYoy1mEn
QzwWXG+S6GnnT9eQdHonrh/ZCbSO+m5GIS6JjsMOrVs22ytThh0atyQrMf/7Xa1IAvl95bLrpSgv64esaedr
MWa7ybCv99dp2vQZGsVrRjdSa+RlcfTlxug18Rg2WfHBE91lWxGrS3yp2ThgE6e5CGlnBDFnx2tUt9lJdfVNNp
R/P5BvPfArVix9ZiHj2JhXk9MXb0Ht2SmYPX9uSh+39WP1NbqRJOtVfa3GqytuPd/v8N7D4wUfJ+deOUsLCo
AT3x4GH+YeLMiZYd/r2KW7IKcNBSO7ol/HavC5IHduZLjrPvLIxtLJd1z3Ce8aC/+IFluGsAHD92Kn+qauQWUv8
AKkRin4QLJVs3KxSv/PiVYAl6utLsvQYqBUvzcXZPW7jmLN+8fiuLbjAA8M8jMyeuJ+//+HVbPGir4fSx8BUvJJsG
buAW5DUmTXXojHS+jrPnrV5sQKz1uAuG3G02WQ/cxJqZ8y82RqfE6l2DbOksLvjp+Cf16JCE3IxmX661INmiR
GKdF4brvMejGJPxhYh/J74zVxGBufjaWfebppiaEVDaYwtE98dymo1g+bQAYQPC0kx+zwL/f+uYW9Lq2OWc
NH3qtGluPVmHu+h/w0rVCRu6xLYWje2L6myWw2B0oMKahMK8n9vKMCyxKNqORmmGITp7kIaWcEMTfH
a1S32Wp6/KNqdBrY1DdYENNox2z3/kOa4uG40lnm1UbAKeQWuyONmVU1RZA9jDPDzhWE8NZg6XSH7K5
w789UytolWcnXrlcsT9ebsB9IzJRZbKK9g/guqDx+0TIku1gGEx/swQAsGpWrkcmmoKcNMXuOe0Bqcl9d3kti
hpsmP/eQQBtz2Ta4B4SJy2pKD1v4jKWyPlDB/qkwZfgWzHLKn/DYLE7sOD9g1j2y/54YdoANLc40GR3QKuO

wY5T1Sh+vxRz87PhkEnA5a5gyS343sQtyG1IhLKn8InXiZ9c+BsjcdHUjP3n60RlPRAxGNFmOXQfY2LKt9Qcm
W9Mddkkmy12vLD1GO4fmeXhm55vTMU7hcNx/9+/w9GfTCgwpgmmN2Tn71G9UgUt90LtsdgdnDHjmUl
90WBrhUGnhupagoCSilrsKa/Bp4cuSp52llbWudSdYOXVPWbiyAUTahrteGRjm9zNzcuGVhMDtUqFb91iV3a
X18AJxuM0lUVuMxqn4GQhHFC6UnkCopQ7HA4cPXoUWVlZSEmhFDztAX93tEp9l5MMOrw4YyAWfnLEww
JelJeNpZ8ew1N39uHadKWhTRmft2G/4O/uPl2D347tzVkxcjOTUd1gw7CsFNE2ieUcV7nlgr1cb0WBMe16rlh
A0Fec/d2n75S37Aj1Cdsv/L8rqhu5CZs/qbPWcmOXBFy95jIjRnua8LwJiC0pr8XKf53AC9MH4rnNRz0y3yyfPg
CTV7WVgleaHSGQJw3eBt/KWValNgzuqdZyM5Lx7Zla8SN1Y5pHoSs567A3cQtyG5KkOK1k3yQbtEGJkWCV
ZfbkBIDHyUkg/Haj0XLIH2NOhvEoDgW0zaFrCocJzpFFedlo4rkS1jTa0adHkuBmuO3k6xTm5mdj6WfHsXl+H
pa51apg5933951Hfu80wTaL5fcelpWC0b1TseyzY9h+8orLd66+PxfpnfT4w0dHBL+zpLwW88cZMSQjmVs7x
DZq7jLBzjHvPzAS978jXHhL6DSVRWozmmdMxcFKk2wmrnDQHrN/BRqflPLHHnsMAwcOxLx58+BwODB27F
js3bsXBoMBW7duxbhx4wLcTCLU+LujNVvssLY48NyUfnAyDCw2B5LihC2KdocTQzJTMEfAJcNid+C5yf24CUh
JRglzcwve23ceiyf3Ra8uCcjqbOB+k7+g1DbZ2iLYeb8FXLegfPzb0bg61g4HwyA9IRZHfzZjTn5POPcweGRjKT54
6FYUXcvB697mkvJa2FudggsW0GbZzuxswLbHxyqysvIVN4vd4ZFq7o1ZuS6FOoTQ69QwWyJvkfcFucnd3YK3
/eQVPDzehjl52Xh6Uh802x1I1GuREq/DJXOz4uwqwfIhXnn3QI9c+UKKn1LLqtgzZuWazVITr9NIHuM/N6WvS
157pdZhpacJchuS9E6xskp3ksG79IpCuD8PXYwK8/KyMWtEJrQxMZibl415+b1gbXFAr1Ujp0s8mlscKK2s88s
HPFoth/wx9rLA87klKwXWFidu6dkZRQLz+vs8F8F6a4tMEGcNivJ6wmJ34GxtE3IzU7BoUl9ctdiRYtCi1cHAbG
lBvx5JYCA8B7Jz+r8eKUCrk0GDtQXxsRrOd5uvkAPXMmlBhd+O6y3ZD1q1CnFaHR4a0wvHfjZj+bQBouNASC
bMzdKGFKH5iN2MLp82AM9uPiq4QXlkYylG9OzM+etHSrxCe8z+FWh8Uso/+ugj/Nd//RcA4LPPPsPZs2dx8u
RJ/H//3/+HZ599FiUlJQFtJBF6vN3R8gU/XqfBgco6rNh6nFN2WEueYOGg5hbRCbntfTu3MCvJKBGriUFJeS3+
MLEPusTr0M3NL5NdUFqqnLj3f78T/A6L3YFWxonZ7+zDmsJh0GlikGdMw5Itbemv5uZlo9nuELXYA8CVRht
WTBuA57eU+e1/rNT6KGU92XqkikvbFWm+qt6SlqAT3fDwXZH4XKq34Xf/r82lhV9Ahh/YG+wS8IB0xVY2V76
YYumvZTVRr3VJY5ebkcwpLO6nL6UXTKgyWV1k2BvrsJLTBCXjWonSLfRbSpURqYBhNkMSq+w8v6UML80ch
Gc2HfWqCqQY7cFyKKRsamJUmLRqt6j7Ef++EvVaVJmVufppYlRYvb0cUwd1xzu7zniMmVnDM2Tze/MzvRSN
7omv3RRylt3lNXhq0s2S7Wq4VoOiICcNK2cMlJ3ThU5AJa93y9bCl4tz1zYoQvFFFrsDTbaWkMcryMlce8v+FQ
x8UspramrQrVs3AMAXX3yBe+65BzfddBPmzp2Lv/71rwFtIBEeEvQaUR8UXcd5id9cd59AIcEXKjsu5iMptzDFx2q5id9k
acFeyeP26wrZRVMzXvvPKVG/zJR4neQ9VtfbuL9Tr00uf7pn8PXgHRm/vSS9FikGbcD8j+Wsj0rccyx2R0T6qnp
LkkGHlSJuT3xXJD469XV3Ib5Fhm+9kfKFDbYPMT9Xvhj+WlbTEnRYPKUf5yqQm5nC3a/7xnhMThruGXojNs8
fHdQMPUqs6t66CylVRqQChhlcz5DEjoeXZw4SrCzsqw94e7Ecuj8fs8WOYVkpiu4rLUGHy/XyJ1/sZntMThq6J
epFx4yc0sd/5rNGZEr+bquDkYwfYtea3Tz5DdTzzzOmwtri4DLHZHY2IL1TLPf9CbFaSWNWUpwupPEKSmWu
vWT/ChY+KeVdu3bF8ePH0b17d3z55+23+ef++23AQAWiwVqdWQGGBDe0WRrRVFeNhh4HmnzfQLFFjWhtH9
iljylCxM78es0MchOi3f5HeBaRon8nih+v00hi9XESFoPuybq8eKMgXhmk6ef8fNT++P+v7dZ0W9MiRPMWm
K22EWttfnGVGSlGlw+FwjklBN2wqsyW3GmpsnDegJErq+qt9zY2YCXZw7C+VoLTM0tSO8Uix8vN3hUcQTgsq
AL+e+zC7mYK0ek+BD7a1lNMuhcAoLl7rd7chyyEC/5nYEgkD763gRPKsk/zf974aQ+iiukKqG9Wg69ua8kgw5
ZqQZRA0meMRWX662Yk5eNf3xf6RGTBLgWn0qK0+LVewaj0doqqPTxn7ncyZjZ0oI518aAmJsISyCfP7/yKZvL
3d0KL7du2h3OkMUreBuwHO3Zv4KJT0r5nDlzcO+996J79+5QPIQ3GwPdDv4lhyWrkrB+d0ijD1Ix6h2i8/rkhM4oVNWCZ4q8VhPrLD
QG8WNYNODSfDoKK6UTJrBOvvOrpXKmI1MahpavO7Y4W4e3IcGu2teH5KP9gdTjTZHNDEqLCnvIZLh8i3m
EtZDzNT4/HKrwajrsmOemsrEvRqVNfbcP/fv0NNY5vS3S1RL/jZJINO0J+SPcq8IcWgvMMDSJJBhzM1TVz2ESE
i1VfVW9g+Tm5ugcPJIKdrJ6wpHIaSilouj3BBThqW/bI/AODB/GzBxYBvvWmyteDF6QOvja3WgFpyAmHp9tey
yt+wCLmu9Ew14IbkOEX3G0n+qizebHy8raDbZJPOBuOLXLVXy6H7fSXGaREfq0HjtUqU/PFyQ4oBL909CIs2H
fUoY798Wn84nAxiVCr86Z7BHv0iZaHlp11k4T9zyVzlxlRcNDfjhc9PYG5+Np6+sw+abA5Y7J5xSCxKYq3c5cW9
n+JjNVyOdn4ud3fkNj6X6qVdggK5BkRjwHKk4pNSvnTpUgwYMAAXLlzAPffffffcg9jYWACAWq3GokWL9Gq3GwoULA9pAIvA
oWUgT9VqPgEI+rEVO6aLG+rEud8v36p41orbJDgbA0i1lHsGM/GOwGKgw7a0SzjfW3YqxeGp/LoWgfDVQN
SqvtmL1N6c9vmfBtRzPYkTqgtoefFWVYLbY8VNdM1ZvP+1WIj4Nm+fnASoGXTspy0IQCutNICzd/lpW3dvgLu
fbHh+r6HsiNb+2NxsfbwOG3Su6sgYEtnqkr8HU7dVyyL+vi6ZmPPnhYdHxcmNnA1Z7OZf6klKS/8ylgpzn5GdD

BRVuyWqraJubkQxAPPMXIC2/cvLCD8q8arEr2uRKrT9y6UQDuQZEa8ByJOJzSsRf/epXAACr9fpurLCw0P8WE
UFF6UKq2KUkTutS0ti92hm7qImVYxaaPIs3lnrkhXW/Li1Bh2FZKYLW/Mv1VvyrrAoWu0OR9bCm0Y65638Q
PBWYu/4HfFacLzkxRuKC2l58VeUwWVqwavtpj3G1u7wGK7YewwszBkbUswnEc/F3IxiINoQ7v7aUYcGbjY/S2
gVAW9+kxF+/nh8wK2VAIJSNFwBen7r4YqF1z2TFX0OAtjVtx49XUPx+KfJ6p+JP11xhnAzjc956JfffZHf4tMkV
W39CuQZ0FCNQKJBPZSGAw+HAihUrcMMNNyAhIQFnzpwBACxevBhr1qxR/D27du3C1KlT0aNHD6hUKmz
evNnl/aKiIqhUKpd/d955py9NJiA/MZh5ea5Zi9yYHNe8r+4WOZ06BqWVdZi3YT/mv3cQc9f/gNLKtty+4/t0Q
ekFEww6NSb07Srriwkom2T57Rt2zYrB/v7akrNIT9Tjf3eeUWw9rLe2cNZC9nvmbdiP1dvLYbE7onKXr/T5RTt
N9lbRcbW7vNYlH3IkEKjnkmRoq4g4JDMFvdMTfPJj9acNSuWUxWyxo6K6EaWVdai40ugy13jLRVMzijeW4v
bXdmLGW3tx+5934uGNpbhoagZwXRkRwl0ZEeuL/Gs+vWv3nOU+9zKvouuYnDRZQ4PQPQayH6IJufFyqd6K
4vfFn6kYvlho3Z85O/evKzkLBsBv1n6P1dvLMSwrBcunDUDXRD16pycgp2snjLupCx4en4M8Y6rLdxbIyI7c/Zs
sLYrXZiGExlUo1wBvZI6QxidL+R//+Eds2LABr7zyCh588EHu9QEDBuD111/HvHnzFH1PU1MTBg8ejLlz5+Luu+
8WvObOO+/EunXruL9ZVxnCe7y1KshZ5MwWOxZtOuqxKJVcK3X+4oyBqG+241dDb0TFFenUT6Zr+Vq9mW
Td25eg1yBWHYMGWys2zB2B5Li29G9ytNddfqS61gSSJpkjWrkjXG8JhA91JDwXf9vgjZwG0s1FqYXeGxcfob5I
0GvQZGvF+w+M9OgbfjC1kkJTweiHaMJsscPW6sBbs4d6nKSy/FTX7FGpc9fpGjz98RGsljh18XXu9vaZs3RPjo
NBp8aL0weiyd4Ki72t/gY/K4oQcvLSZG/12Sfbn2JigaK9BiyHA5+U8nfffRf/93//h9tvvx2//e1vudcHDx6MkydP
Kv6eSZMmYdKkSZLXxMbGcukXCWGUKgq+WhV82f3vKa9Fi4NBRud4FG8sRdHonpK/bWtxwmyxez3Juvss+
rLotWdXj0h0rQkk7j6+7rjn+fWHQCpVkfBc/GmDUjkNtJuLUsOCt8qIt33BBlNLwZ9Pfe2HSAyk9QYl6XKl2H26
BtUNNtF79mfu9nX8+/I5OXmRMy6Indb6W0wskESCsaE94JNS/vPPP8No9AyAczqdaGkJ7FH/jh07kJ6ejpSUF
IwfPx4vvPACUlNTRa+32Wyw2a7nl66vrw9oeyINdtI7cL6OCzg6V9OEjBQDuia67t4DbRFWquTvPl2DwRnJkvl
e956pRecEHVIMvk2y/iz+tMsPDcGQzfROsZJVU9M7BeZkLdw+1KFEiSKoVBkKdFYGbwwLwVZGvJlP+f3gERy
qVcNkafEqq0igLevBkE2l6XILctIEi31x39Ps+cz5Y/S5yf0Ei9UFYu4O1KZIr40RTfk4JidN1rggtjZHWtaTSDA2RDs
+KeX9+vXD7t27kZWV5fL6Rx99hNzc3IA0DGhzXbn77ruRnZ2NiooKPPPMM5g0aRK+/fZZ0XzuxLJlywWL1kuxLJlywL
WhkiGnfQOnK9TFHAUaIuwkkWJXUTFotwLjKkovJbvtV/3RGw6eAovTB+A5zYLV8H01WdPbnKiXX7wCYZsiq
WlDPSGKtIWv2ChVBFUupENdFaGSHI182Y+ZftBLDiU9Ulm+zjUm8BgyKaSdLljrqUrnfzGHtHvcXdBFBujXzxSg
PpmO+JjAzN3B2pTZLbYseTTY4J1P/KNqXhxxkB00mt8Wpsp60n7wyel/Pnnn0dhYSF+/vlnOJ1OfPLJJzh16hTe
ffddbN26NWCNu++++7j/HXu/xY8cO3H777YKfWbRoER5//HHu7/r6emRkZASsTZEEO+kVjzc
qymwSaItwgl4jaqV0n0jESnl36RSL+/7vOy5Ty1cnqgFAsviDEIGYnGiXH1yCJZuh2FB1hMXPW0VQqt9ZC2Ork
8HaouGCfsSA90p0JLmaeTOfspsJseDQ3W59HOpNYDBkU05mkuLaqh2bLS3IzUwWzRUep72eZlJqjD6/pSxg
m5VAbopqGu34+kQ19lbUCmb4sjucPq/NkbRJJQKDT0r5tGnT8Nlnn2H58uWIj4/H+Kzzz7DL37xi
0C3kaNXr15ISS0tDeXm5qFIeGxvbYYJB2UkvNyNZccBRoBSYi6ZmPL+lDIWje8LJMJLVD/npp/jtzDOmIjczBRa7
wyX92FcnqrFwUqtg4QcxaHKKfIIpm5HkqhCt+KIICvW7Uj9iX5ToSHM1UzqfspsJpXN1qDeBwZBNOZlJ4Y2dh
8fnAPDMFV6Yl43Jb+zBsKwUvDRzEKwtjpBsVgK5KWKfpVjdjwl90gH4tjZH0iaVCAw+5ykvKCjAV199Fci2yPLT
Tz+htrYW3bt3D+nvRirspOdedc4d9wmcL+T11hZAdf09Jf5zfCuC++4/OU6L3ukJ6Mqrgrl82gAs3lLmsUizZYqF
ShZ7u+jQ5ERIcbneylVtTYzTIMWgcxmjcnSE8RUIRVDKjzgGwAcP3Yqf6pqRYtAis7PBJwUq0lzNlGwI2c3EiSpp
X222j9vDJlBKZgpy0qBVq3Co8io6xemQmRKH6UNuwO8n3MT5kPOrZrIW6uem9JP8zUBtVgK5KfLmWfoScB
xJm1TCf3xWygNBY2Mjysuv7xzPnj2LQ4cOoXPnzujcuTOWLVuGmTNnolu3bqioqMBTTz0Fo9GIiRMnhrHVk
QM76blXnXPHfQIXsmQV5KRhwW1GzF3/g4slS8h/jm9FENr9b3t8LLomegahshlYbkyJg04dgx+rG/HGrFzBks
XeLjo0ORFiVNY2eaTuZH05M1PjFX1HRxhfgVAEpSyMu8trUdRgw/z3DgLwL2gxGl3NeiTHyebNZ/u4PWwCx
WQmz5iKwtE9sfSzY7h/ZBbuf+d7DMtKwYsz2ztIM/upv3wp+367TNXA6GcnfDNRmJZCbomA/y0jbpBL+oVgp
T0lJgUqlkr8QwNWrVxVdt3//ftx6//Xdo2effRYAMH/+fLz88svIyclxvv/9d99xQXFyM9/Y4xkaZ0FyPP0
tv54xXFVcbELFm7T9fAyTBcRDwg7j+nxIrg/jvuAaiv3jMY7313PqATFU1OhDuX662CufT3lNfimU1H8ed7h3h
YzMUyLrT38RUI5UFubuCf6rXHzDVypHeKVVYpuZ1sAnskx+HVewajoroRpuYWWzo+aNcLYWp3cmvPMpqOyl
nC2SnOwNyuBVKRD8SyjcZNKCKNYKX/99dcD/uPjdFHDcw4cPiyCqo0Dw4jvfP/9738H/DfbA+5KjGqV4LJiO0CuJiO
cj5D+nxIog55PXaG0NykRFkxPBp67JLlrxc095Leqa7C5KuVzGhfY8vgKhPMjeu0pcw2LVBo9b/q4vWw

CTRY77n9nn+B7/DVHiSU8KU4bks2K2HMqyEnD8mkDvP6+9vIsieCjWCkvLCwMZjsIhYgpDS/PHITVCoTeG0s
Wi7v/nBIrglxhjTqLHWkJCTRREUGl3irtLsB/Pxy5yCOtOIy/yoPU3MAP5ubTHjLXsChJo+dNH0f7JtBsseOnumb
Ja/hrjhJLeJJBF5J1g31Ol+qt3D2UXjDhrlW7ucBTb1yvov1ZEqHBb59yq9UKu93u8lpiYqK/X0sIIKU0PH1NaXDP
WOK+6CfESj9yIf90oSqactaKRL3d/Wtc29Xcgoc3luKlmYO8yrJCEN4osol66fHOfz/UaegCXX6+ptEOc7Mdhlg
NYlQqaGJUSPVByfdHeZDyI3YP5mYJlB9wuDc43mzqOoqCVtMovQ4ArmuOIVaNF2cMxDObjnqsLcunDcC52i
YkNNmRFq8L2LohN25e+PxE0Dfq4R67ROTgk1Le1NSEp59+Gv/85z9RW+t5NOxwSJeMJXzDW6VBaNFfefdA
0dziQpYsMf85FYBJA7ujcHRPLudqdcP1inBKLGYd0aeU8A9vFdmUeJ1oJb18YypS4r2LlQgUgbTKi6UgnJOXjZV
fnMCyaQMCXgFSCndLcHysBvvP1wmWVQ+UH3Aoq1+K0VEKTHlDvbUFpRdMKDCmYrdINWd2zckzpuLg+T
po1TFYefdAWFucaLC2IE6nxsHKNgu1XBICb5EbN6F4ppEwdonIQTpthwhPPfUUtm/fjrfffhuxsbF45513sGzZ
MvTo0QPvvvtuoNtIXMMbpUFs0V+x9TgW3GZEQU6ay+sFxjQU35aDtXvOcq+J+emZLXY89fERLPrkKOZt2I/5
7x3EvA37seiTo3j64yMwW9qsI89O7os1hcOwtmg4iscbYdCpOWWB/R12YiMIOeQUWXbc8emaqMeLMwYi
35jq8jqbfYXvTx7KNHRKFns5zBY7Tl9uwImqeszJy+ZkDGjz1V1XchY3d08U7RtvMVvsqKhuRGllHSquNEp+Z5K
hzZI5JDMFOV07YexNXTAsK8XlmkCWQvd2XASDjlBgylsS9Vqs3XMWc/KzUWB0XXP4awH7/xc+P4HO8Tos+u
Qo0hJ0yE6Lxwufn8CiT466bOgC8WyVjJtGWwuKxxuxpnAY3po91GUtA/x/ppEyduXwRvYJ//DJUv7ZZ5/h3Xff
xbhx4zBnzhwUFBTAaDQiKysL7733HmbPnh3odhLwTmkQW/Qtdgfmrv8BH/12FJ66k8FPdc3QqWNw9Gczvj
9Xizdm5cLW6kSvtHh0T9ILLphSCsX+83Wos7R4BJ0WGFOxeX4e/nWsysNi1hEXK8J7fLVaZabG48/3Drmep1y
vQUq8Z57yUKah81eBU1Kghw2iW7293G+Lnr/WvGAGukWKhbo95BYPNGkJOgzLSkHx+6V4aEwvPDohB04
ng5gYFXSaGFSZrR5pcW2tTpeNabCerdy4qW2yIylOh9LKOo+id6yc+ftMI2XsSkGW/NDik1J+9epV9OrVC0Cb/
zibAjE/Px+/+93vAtc6wgWhsvYGnRpz87MxulcqzM12VFxpRFq8dEU4i92Bc7UWvLfvPHIzUwSrjG2eP1p0Mp
D67rn52Vi8+ajHUeXu8los23qMq+DJpyMuVoT3+KPIdk3UuyjhrOXH3YczVGno/FHgpAr0AHBJa8oG0fmz8Q
2Uq00w/KjNFjuuyljtQrXpbw+5xQMNX6Ze//o0Xv/6NNYUDsO8DftFP8P6mDdYWyCdi0X62cr5acvNJw4ngy
WfHfPI3sT+vXhKP7+fqVwb6ix2lFbWhc3PPBzB7x0dn5TyXr164ezZs8jMzESfPn3wz3/+EyNGjMBnn32G5OT
kADeRANp2q3/8/DieuONmgAF2l9fAoFNj1axcrCs565EH/NnJfSW/L1YTI5gCkUVKKZBSKKTKSAv9XkddrAjvC
ZQlUs7yE4rMDv4ocN6kNWUVHH82vpFqzWOfI1uUTAy9Tg2zJfhtbC+5xQONu0ylGJRl6FEyZsWuUWLdlZtP
HE4Gu8vF5ez5Kf38fqZybTA3t3AbmHBYpyNV9iOBYAXn+qSUz5kzB4cPH8bYsWOxcOFCTJ06FatXr0ZLSwtee
+01vxtFuGK22PH8ljLcNylTq7b9iMGZySjK64nO8Tq89p9THjv5XadrMKnSJDnxHf3ZjOLxRnTpFIu3Zg+FXqvG
wco6rN1zFsOyUjilQGjgSSkUcvDTX3X0xYrwjkBYIpVafkKhwK28eyAuXLUgVquGRh2DuiY7dOoYZHSOk/x9p
WlNWQXH341vJPpK85/j4Ixk0eJpecZUbD1ShSMXTCFRaCgftTDuMiWXoYc/Zr2VeaUy7j6fsKfOuRnJAIAGme
qrzXaH34qZNylEw2GdjkTZjwSC6dLjk1L++9//nvv/hAkTcPLkSRw4cABGoxGDBg3yq0GEJzWNdvTpnoh1JWd
RUl6L7SevAADWFA4TjGgH2gI6v3ikwMO3O8+Yinn52VBBhXf2nPHwlVtbNBw9OxuQZNBJDryXZw7C0wIW
oRtTpAdkz7R4fPDgrUg2aAX9eglCjEBYIiPF8lNlasbeiloMvjEZK7Yec5FjucldSYEeVsH5x/eVfm98I9FXmv8c1+4
5i1WzcgHARTHnK3kWuyNkCk1HSXfoD2I5wB/ZWIphWSkuY9ZbmeePDb6ibWt1Qq9Vw2Rp4Z4R+937z9d
5nDqvKRwmeQ9xOjWKN5b6pZh5m0I01NbpSJT9cBNslx6vlPJvv/0WtbW1mDJlCvfau+++i0WVXZ0sfau+++iyVLlqCpqQnTp0/
HG2+8gdjYWJ8bFMmEK5dovbVF0C1EqNAPi8XuQH2zHX+6VuLYbG1Bl4RY6DQxYBgGf/q3p4W9pLwWapU
Kb8zKVTTwhCxCgLhlI9+Yii+OVnH3QcEihBLc5e7VewajydaK+mbvLZGRYPkxW+w4f9WCFocTy7Z6+qzKTe5S
1rWCnDT0SovH0qn9oY5R4U/3DPZ7jlJ6QhHK+ZH/HC12Bx7ZWIq5+dmYm5cNW6sTN6bE4T/HL7sElYfzuJ3
yUHvCKsbdEvWoabQjNV6HGUNu8JDneJ0ai6f0g6m5BQk6NQw6DZINWtH+czAM1hQOQ6uTQXZaPJZ/dsxl
7Sy4ptD3SI7jNgcmSwue23zURRZLL5hET2DG5KThYKUpIIqZ++mKThODL8ouCaYQBYI3R3l7Kt5RXU+DbdjxS
ilfvnw5xo0bxynlR48exbx581BUVIR+/frhlVdeQY8ePbB06PpbBdbB06VKfGxSphDMCOVGvRZXZ6vF6rCbGwxIQp1XDyTB
Qq1SwtTJotLUiK9WAn03N+MvXP6KkvFbSwi4X9W7QqTEoIxlVZiuaWxxIjNMiOy1e9mgy35iKIredPwWLEFK
YLXbOkqZSqVzcq5ZPGwCVCm0J870gGJYfbxUuk6UFb2w/jbl52SitNKF4vNHFksfep/vkzv5Oo60Fy6cNwPNb
ygSth90DPB8pOaEI9fzo/hwtdoeL4rWmcJhgbEs4jts7avYKs8UOk6UFTfZWNNkdSNRroI5RQaeOga3ViUZbK
ycvYoWApPouySB8/YrP2k6eiscb8e635zyU6t0Crmo1jXaPNVHsBIYtZHTXqt2CbfZFMeOfrlRUN4rGZQHBsU5
LVQqnOAlXgm3Y8UopP3ToEFasWMH9/cEHH2DkyJH4+9//DgC48Ybie7/HbneB2S4zwFseyiGW

```
sLh+GNb8pdhJhVgAvXfc+VLZ4/zojSShMAaQs7IB71LhVYKlVGWq9VY+tRz1SIAAWLEMJcNDXj6Y+OuARa8V
ORPbv5KJc5yBsFJ9CWH18UriZ7K0rKa/GbUT0F5Ym9zybb9cnd/XcM16yHz07ui2a7I+i+y0K+0gl6DZpsrfjxcg
OnCPHxd36U2ux444vLJ9TH7R01e0WVqRnnr1rwxvbTLgptQU4aim8zYs76H2QLAXnbd2aL/dqc0fZ7UkkH3
NcdIUWLfwLz7F19YW91cnJWcaVR0IrN4o9iFmrrtJJK4RQncZ1gu/R4VTyorq4OXbt25f7euXMnJk2axP09fPh
wXLhwwa8GRSKBKPQhhlBS/ipTM4o3luL213Zixlt7Memvu7nKhHwYBnjzm3IPS8Cea4VD5uZnc21845vT3N
/8ssZCxMdqBAfe3Pxszq+dDyu8l+uvW/P5xUOaW9qsWGKTWEcNFiGE4RYJt8wHJbxxXVJeywVkeVNog7X6j
nErniVVKEusaIavhT+arslB9yS9oDyx95kUd90txP13LHYHFn1yFH/8/ASy0+LROz0hJL7SrEzH6dR48sPDGP/nn
bhw1aLo5M0bLrrNgbf/eSce3liKi6Zmri1iz/Hh8a5F0PjvhfqfXWjtMOjUKB5vROHonvixurHdFWMxW+zY8e
MVD4UcaLNSr/6mHO89MJIrxLP/fJ2gvHi77l6qt7rMGUqMTyxiihZ7AhOrUWNIZgpn0be2SH+3UsVMaH7xd
o7yF6XuGKzsh2KuiWTYTZMQgZhjvLKUd+3aFWfPnkVGRgbsdjsOHjyIZcuWce83NDRAq21/jv/BOq4Qs7LNv
82IA+fruNcsdgf+e80+vDt3BFb+6yR3/cAbkvD616cFv9s9NRr/bylfuTxjKnSaGOi1arz/wEiYmlu4I/WhIjnNgbb
JtqK6EQ4n42HxoGARwhuUpvzjL7renLgozZARrBLc7KmXvdUpKIPsfdodbfcXKcGpLO6bBG+UH1++n8XdQir2
HC12B4ZlpUTEcbv72qH0tDGaqWm0I71TrOjY3n26BkWje2Lehv0up19V14w6UtZrPu4VrNmAURY54xN/3fH
GOl3TaMfeM7Wia2iBQsUsElKzApERZxNNBDv1qVdK+V133YWFCxfi5ZdfxubNm2EwGFBQUMC9f+TIEfTu3d
uvBkUiwfJDFVt4HAzjUgAEaJsIfvW3b/HvRwoAFdBod6C+WVlqNPe/pbIVzMvPhsPJ4MkPD3tkbcnvLbw7ZD
E1twgeK1KwCOENSlP+uS+63iwechky5I50F0/pB7OM/Im1x6BTI9+YiiuNNsnPN11LyebLomm22FHdYIOpuQ
XxOjXiYzVIjhMPjvMG902CN8qPL9/Px30TIvQckwyImON297VD6rQx2txZxNyL6q0tshs19n1+waszV5rw4uc
nOKVUbt3VaWK4YnlCpzFygZr8dYdVtJZsKcPN3RO5GI8UgxaZ17KRsdRbWyTX0GW/7C/7DCMlNStARjNfCO
amySulfMWKFbj77rsxduxYJCQkYMOGDdDprjdi7dq1uOOOO/xuVKQRDKXSmwIgLAadGg4Az25qixJfWzRc8
jfcF0v2b76v3PxxRjgYBi2tTpReMOHoz2as23PW4zi6pLwWC8YZZX9PyHJHRTUIb1Ca8s/dbziQi4eUfO4+XYM
LVy2y3yHUHrPFjiWfHkNRXjbitGpFn/d20RTzx394fA6yOhv8DgR13yR4o/z48v3uKNl8RUpaQve1wxs/50hGy
sqbqNfiapO0Ow5/beKvd3ylVC5u4IuyS1xMySO353iMQ6lATaF1p0dyHJZM7Y9FnxyRPMVI1GsFM/7EamJEY
xnciaTTLzKa+Uaw5hivlPK0tDTs2rULZrMZCQkJUKtdF5UPP/wQCQnCUdTRjLdKpZJsDEqtgXXenjkIz/LSNjEMg
3xjKvaIuKHwJ4iCnDRUN1y3zFnsDpRW1mFIRrJLAOb7D4zEX74SdonZe6YWBTlpgpMJ//fMzS0eJcypqAahFL
nF+HK91SOHr7eLh5SMKindbmt14nhVvdfKaE2jHV+fqMbeilq8MSsXBcY0waqB/M97s2heD3bz9McHgCmD
euCuAd0UyZ1YH7lvErxVfuRoT5Y797Uj0K4+4YBv5XXP/nW+tglZqfGobrCJu3eIBOKyr7FKae/0BEU5vHedrsF
vx/b2GIes4vzc5L5YPLkfztY2ITs1Ht2T9IJj8nK9FYs+OSIbsMyXR/cN1picNDyYL1wlm08kuYyQ0Syy8Kl4UFJSku
DrnTt39qsxkYwvfqgGnRr/M7YX7ujXFTWNdlhbHbDYHUiO06JzvA4GnVo0+FHoSLhbot5lktPFxODJiX2gwim
XRdg9/SArXAadGiN6dkaDtQVxOjUOVppcFPIxOWmSR9Fr95zF5vl5WLH1uIcVjj9JWlscuPvtvdz7fEsDCTghh9
giUZCThuXT+mP/uTqPcevN4iGV/osBFJVuj9XE+KSMsouxxe7AwxtLsWpWLpxgJD/PVv88X2txifE4VVWP5dM
GuPxOW2o36RM4JVY4KUuo+yaBbzVcMM4IvVaNpDjfN92hsNyJbTj4ryfFaREfq0GjtdWv/OL8tcPWKp6xA4i
ODQdr5RXzj5/QNx1LpvRDr7R4xAAuSm5BThrm5Wdj/nsHudcMOjWyUg1Qx6jw7twR6ByvQ4vDiTNXGpEa
r+P6rs5ih7m5hSsyxF87956pxS1ZKYLW6+p6K1qdTnxy4CfRvP0XTc04V9MkG7DsXnTIVyU2nBtPobFPRrPIwS
elvKPijR+qQafGm/cPhUEXA5OlBau/OeGRGmpd0XCX1FD89/hWbaBN4Jvs18v+GnRqpCfF4uUvT2JwZjKK8
nrC1upEUpwW6Z1icclkxfo5w5EaH+siXPz2d0vUY0TPzmiytSApTge7w4kWh3RBooumZhTl9cRvx/WGubmF
O7JjJ8l8Yyr2nol+f0kivEgtEp0NOgzNTPFp8ZDy5dzx4xV8caQKu8vlS7eXXjBJpkwTaw9/MRY6Au+V5mnJu2h
qxsJPjrq0uSAnDStnDPRwRVFyAidnhVPi7+qulFjsDhy5YMLsEZl+u8cE23IntOH4Rd/0thSTm8s8FE73DZMvAZ
ns2mG22KPeVYAdY2L+8V+fqEasJgZP39kHdw3sjiKegny53gqGl2/XoFNjbeEwLP30mMuJL2voWfnFCSybNgC
90xNQWlmHeRv2C7Zp7Z6z+PyRfCzeXOaRXnROXjb+8tWPHhtYFna8zxqRKXnffLnxV4kNl8uIXHAprc/hh5Ty
AML3E5ubn40qc1s0+OdHqwRTQwHA4in9sOiTo9zrQlZtVuD5BYTm5mfjha3Hsbu8FttPXnH57jxjKnIzUzBjyA
3onZ7ApV0yN9thiNUgRqWCJkaF1GtFG/iCWjzeKKmM7L9W2GTVrFy8t++8x0ajcHRPj9LAQHT5SxKRgdgm2
B9fPilfzvROsZyVWUnpduB6yrSpg7pjSGYKgOtpzpTk1+YXvRmTk8ZtXFlrloNhBHOA7z5dg2c2HfXY6Crxx5ez
winxd+2dnhBUy1qwLHdiG46buydi0aajLoGHwQjIbA+uAuwYk/KPz+6SgGc3HRW0POcZU7lEBs9N7os3vyn3
cMFk+z03M4Xrb6mxbbE7YG91YkhmCua4+Xgv/PgI7huRiQZrK0or6zxkkh3vcqdjeq3a4/PuedLF5N6dcIyDjp
```

ozP9ogpTyA8K1UbA5lAIIKLtC2sD47uS+++v0YNNlaPRYedwGxtjo5H3KpCZENykxLEK60x7dCLP1lf84KZ9Cpo
YlRYeGkPqiut7lUUbwlK4VTuIUsfD1TDdCoYzD9zRLKR05ELPXWFg8/WNYdpNV53YQnNMYzOsfh38cuexyd5
xlTcfB8HTrFagCVStYS5U11TCXVd5VmOsozpqK6wYZhWSmyfSQFK8fBDqYMxveLbTjc59NgBmRGu6sAO8ak
/OPl1qdn7+qLqYO6o7nFiWc2lYleNzevTXmvabTL