Python Code

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "f5f36ca3",
   "metadata": {},
   "source": [
    "
project 5,6,7"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "0c5bd25b",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "a6097016",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "   Employee_ID    Employee_Name Department  Experience_Years  Monthly_Sales
\\\n",
      "0         1001          adarsh         HR                15          41834
\n",
      "1         1002  pratyush singh   Marketing                13          38047
\n",
      "2         1003             Amit         IT                 1          46105
\n",
      "3         1004            Priya  Marketing                 9          95766
\n",
      "4         1005            Karan  Marketing                 7          35707
\n",
      "\n",
      "   Customer_Satisfaction  \n",
      "0                     20  \n",
      "1                     15  \n",
      "2                      7  \n",
      "3                     10  \n",
      "4                      9  \n",
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 100 entries, 0 to 99\n",
```

```
        "Data columns (total 6 columns):\n",
        " #   Column                Non-Null Count  Dtype \n",
        "---  ------                --------------  ----- \n",
        " 0   Employee_ID           100 non-null    int64 \n",
        " 1   Employee_Name         100 non-null    object\n",
        " 2   Department            100 non-null    object\n",
        " 3   Experience_Years      100 non-null    int64 \n",
        " 4   Monthly_Sales         100 non-null    int64 \n",
        " 5   Customer_Satisfaction  100 non-null    int64 \n",
        "dtypes: int64(4), object(2)\n",
        "memory usage: 4.8+ KB\n",
        "None\n"
       ]
      }
     ],
     "source": [
      "df=pd.read_csv(r\"c:\\\\Users\\\\yadit\\\\Downloads\\\\adarsh data sheet .csv\")\n",
      "print(df.head())\n",
      "print(df.info())"
     ]
    },
    {
     "cell_type": "markdown",
     "id": "a3ba1168",
     "metadata": {},
     "source": [
      "Question 1.Load the dataset and display the first 10 rows"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "id": "895db9d7",
     "metadata": {},
     "outputs": [
      {
       "data": {
        "text/html": [
         "<div>\n",
         "<style scoped>\n",
         "    .dataframe tbody tr th:only-of-type {\n",
         "        vertical-align: middle;\n",
         "    }\n",
         "\n",
         "    .dataframe tbody tr th {\n",
         "        vertical-align: top;\n",
         "    }\n",
         "\n",
         "    .dataframe thead th {\n",
         "        text-align: right;\n",
         "    }\n",
         "</style>\n",
         "<table border=\"1\" class=\"dataframe\">\n",
         "  <thead>\n",
         "    <tr style=\"text-align: right;\">\n",
```

```
"      <th></th>\n",
"      <th>Employee_ID</th>\n",
"      <th>Employee_Name</th>\n",
"      <th>Department</th>\n",
"      <th>Experience_Years</th>\n",
"      <th>Monthly_Sales</th>\n",
"      <th>Customer_Satisfaction</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1001</td>\n",
"      <td>adarsh</td>\n",
"      <td>HR</td>\n",
"      <td>15</td>\n",
"      <td>41834</td>\n",
"      <td>20</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>1002</td>\n",
"      <td>pratyush singh</td>\n",
"      <td>Marketing</td>\n",
"      <td>13</td>\n",
"      <td>38047</td>\n",
"      <td>15</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>1003</td>\n",
"      <td>Amit</td>\n",
"      <td>IT</td>\n",
"      <td>1</td>\n",
"      <td>46105</td>\n",
"      <td>7</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>1004</td>\n",
"      <td>Priya</td>\n",
"      <td>Marketing</td>\n",
"      <td>9</td>\n",
"      <td>95766</td>\n",
"      <td>10</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>1005</td>\n",
"      <td>Karan</td>\n",
"      <td>Marketing</td>\n",
"      <td>7</td>\n",
"      <td>35707</td>\n",
"      <td>9</td>\n",
"    </tr>\n",
```

```
          "  </tbody>\n",
          "</table>\n",
          "</div>"
         ],
         "text/plain": [
          "   Employee_ID    Employee_Name Department  Experience_Years  Monthly_Sales
\\\n",
          "0         1001          adarsh         HR                15          41834
\n",
          "1         1002  pratyush singh   Marketing                13          38047
\n",
          "2         1003             Amit         IT                 1          46105
\n",
          "3         1004            Priya  Marketing                 9          95766
\n",
          "4         1005            Karan  Marketing                 7          35707
\n",
          "\n",
          "   Customer_Satisfaction  \n",
          "0                     20  \n",
          "1                     15  \n",
          "2                      7  \n",
          "3                     10  \n",
          "4                      9  "
         ]
        },
        "execution_count": 6,
        "metadata": {},
        "output_type": "execute_result"
       }
      ],
      "source": [
       "df.head(5)"
      ]
     },
     {
      "cell_type": "markdown",
      "id": "c802b7ab",
      "metadata": {},
      "source": [
       "Question 2.Display summary statistics of the numerical column"
      ]
     },
     {
      "cell_type": "code",
      "execution_count": null,
      "id": "01721082",
      "metadata": {},
      "outputs": [
       {
        "name": "stdout",
        "output_type": "stream",
        "text": [
         "       Employee_ID  Experience_Years  Monthly_Sales  Customer_Satisfaction
\n",
```

```
        "count   100.000000        100.000000     100.000000             100.000000
\n",
        "mean   1050.500000          8.450000   72030.750000               5.060000
\n",
        "std      29.011492          4.349329   30111.719996               3.299281
\n",
        "min    1001.000000          1.000000   20854.000000               1.000000
\n",
        "25%    1025.750000          5.000000   44832.000000               3.000000
\n",
        "50%    1050.500000          8.500000   72697.500000               5.000000
\n",
        "75%    1075.250000         12.000000  101270.000000               7.250000
\n",
        "max    1100.000000         15.000000  118506.000000              20.000000
\n"
       ]
      }
     ],
     "source": [
      "print(df.describe())"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "id": "15ff490d",
     "metadata": {},
     "outputs": [],
     "source": [
      "Question 3.Check how many rows and columns the dataset has"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "id": "f7fc1190",
     "metadata": {},
     "outputs": [
      {
       "name": "stdout",
       "output_type": "stream",
       "text": [
        "(100, 6)\n"
       ]
      }
     ],
     "source": [
      "print(df.shape)"
     ]
    },
    {
     "cell_type": "markdown",
     "id": "3bf6e704",
     "metadata": {},
```

```
    "source": [
     "Question 4.Select only the Employee_ID , Experience_Years , Monthly_Sales , Customer_Satisfaction and balance columns ."
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "id": "98a49f0e",
    "metadata": {},
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "[['Employee_ID', 'Experience_Years', 'Monthly_Sales', 'Customer_Satisfaction']]\n"
      ]
     }
    ],
    "source": [
     "result = [['Employee_ID','Experience_Years','Monthly_Sales','Customer_Satisfaction']]\n",
     "print(result)"
    ]
   },
   {
    "cell_type": "markdown",
    "id": "28113a6b",
    "metadata": {},
    "source": [
     "Question 5.Filter rows where monthly_sales greater than 20."
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "id": "9e8b8493",
    "metadata": {},
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "    Employee_ID    Employee_Name Department  Experience_Years  Monthly_Sales\\\n",
       "0          1001          adarsh         HR                15          41834\n",
       "1          1002  pratyush singh   Marketing                13          38047\n",
       "2          1003             Amit         IT                 1          46105\n",
       "3          1004            Priya  Marketing                 9          95766\n",
       "4          1005            Karan  Marketing                 7          35707
```

```
\n",
        "..          ...              ...        ...               ...            ...
\n",
        "95         1096            Megha    Finance                12          93656
\n",
        "96         1097           Pritam    Finance                12          59384
\n",
        "97         1098           Ramesh  Marketing                 4          67254
\n",
        "98         1099          Shivani      Sales                14          41918
\n",
        "99         1100            Niraj  Marketing                14         105981
\n",
        "\n",
        "    Customer_Satisfaction  \n",
        "0                      20  \n",
        "1                      15  \n",
        "2                       7  \n",
        "3                      10  \n",
        "4                       9  \n",
        "..                    ...  \n",
        "95                      6  \n",
        "96                      7  \n",
        "97                      2  \n",
        "98                     10  \n",
        "99                      2  \n",
        "\n",
        "[100 rows x 6 columns]\n"
       ]
      }
     ],
     "source": [
      "result = df[df['Monthly_Sales'] > 20]\n",
      "print(result)"
     ]
    },
    {
     "cell_type": "markdown",
     "id": "a50ac643",
     "metadata": {},
     "source": [
      "Question 6. Find employees whose department is HR ."
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "id": "c11be5da",
     "metadata": {},
     "outputs": [
      {
       "name": "stdout",
       "output_type": "stream",
       "text": [
        "    Employee_ID Employee_Name Department  Experience_Years  Monthly_Sales
```

```
\\\n",
    346         "5          1006         Divya         HR                 9          41976
\n",
    347         "14         1015        Anjali         HR                 1          29474
\n",
    348         "18         1019         Suman         HR                 1         114856
\n",
    349         "23         1024         Manoj         HR                10          86199
\n",
    350         "31         1032        Monika         HR                 1          71005
\n",
    351         "46         1047        Shweta         HR                15         110084
\n",
    352         "51         1052         Rohan         HR                10         107092
\n",
    353         "52         1053           Ira         HR                 7          70859
\n",
    354         "54         1055       Krishna         HR                14         107455
\n",
    355         "56         1057         Sunny         HR                11          90467
\n",
    356         "66         1067      Niharika         HR                 7          54698
\n",
    357         "68         1069         Aarav         HR                 4          42671
\n",
    358         "72         1073       Sandeep         HR                 5         106202
\n",
    359         "76         1077       Prakash         HR                15          69811
\n",
    360         "77         1078        Mitali         HR                11          22811
\n",
    361         "79         1080           Jay         HR                 4          92082
\n",
    362         "80         1081       Vandana         HR                13          54754
\n",
    363         "90         1091         Sujit         HR                 9         108614
\n",
    364         "92         1093       Shalini         HR                 6          57504
\n",
    365         "\n",
    366         "    Customer_Satisfaction  \n",
    367         "5                       4  \n",
    368         "14                      3  \n",
    369         "18                      8  \n",
    370         "23                      4  \n",
    371         "31                      9  \n",
    372         "46                      8  \n",
    373         "51                      1  \n",
    374         "52                      3  \n",
    375         "54                      7  \n",
    376         "56                      6  \n",
    377         "66                      3  \n",
    378         "68                      3  \n",
    379         "72                      6  \n",
    380         "76                      8  \n",
```

```
        "77                    1  \n",
        "79                    3  \n",
        "80                    1  \n",
        "90                   10  \n",
        "92                    8  \n"
       ]
      }
     ],
     "source": [
      "result = df[df['Department'] == 'HR']\n",
      "print(result)"
     ]
    },
    {
     "cell_type": "markdown",
     "id": "67e80b4a",
     "metadata": {},
     "source": [
      "Question 7.Count the frequency of each employee_name column value."
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "id": "3ab32155",
     "metadata": {},
     "outputs": [
      {
       "name": "stdout",
       "output_type": "stream",
       "text": [
        "Employee_Name\n",
        "adarsh             1\n",
        "pratyush singh     1\n",
        "Amit               1\n",
        "Priya              1\n",
        "Karan              1\n",
        "                  ..\n",
        "Megha              1\n",
        "Pritam             1\n",
        "Ramesh             1\n",
        "Shivani            1\n",
        "Niraj              1\n",
        "Name: count, Length: 100, dtype: int64\n"
       ]
      }
     ],
     "source": [
      "counts = df['Employee_Name'].value_counts()\n",
      "print(counts)"
     ]
    },
    {
     "cell_type": "markdown",
     "id": "2eccfa9a",
```

```
   "metadata": {},
   "source": [
    "project 6,7 fills "
   ]
  },
  {
   "cell_type": "markdown",
   "id": "dd07b226",
   "metadata": {},
   "source": [
    "6) Create a line chart and a bar chart using Matplotlib library "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "68bf60c1",
   "metadata": {},
   "outputs": [
    {
     "data": {
```


"image/png": "iVBORw0KGgoAAAANSUhEUgAAAjMAAAHHCAYAAABKudlQAAAAOnRFWHRTb2Z0d2F
yZQBNYXRwbG90bGliIHZlcnNpb24zLjEwLjcsIGh0dHBzOi8vbWF0cGxvdGxpYi5vcmcvTLEjVAAAAAlwSFlzAAAPYQA
AD2EBqD+naQAAVExJREFUeJzt3Qdc1OUfB/APewkIyhAZgnvixj1yrzRzW+60MldlasOmWTbMykzTHm5Usu4R5t64Ebe
iKDhxsjf3fz3PcfxBRQEP7ve7+7xfr8vjOOC53w+7j9/n+zw/M41GowERERGRSpkbegBERERERz4NhhoiIiFSNYYaIiIh
UjWGGiIiIVI1hhoiIiFSNYYaIiIhUjWGGSAWuXLkCMzMzLFiwAAGoY57fffmvooZi
sQYMGoUyZMoYeBlGRYpghMjARUEQAOHLkCJQsJCQEr7zzyCnx8fGBjYwNXV1e0bt0a8+fPR3p6ukHtHHjRnzyySd5fn6
LFi3ksX7SrVKlSoU6Viq9iqPJaF+L2JSE/8/PyQmJgIKysrgxzTuXn4vvXXX4eHwdeffVVl1C9fHrGxd2bRuGDh2Kmzd
v4v333zdImJk5c2a2a+Ao23tzemTp36OPOzs56Hh0RFRWGGSIVUJg5UmNvbgstF3rEhbeb1JGHdF1d5kdUp6o9NOwB
OnSrSMcXHx8PBwaFXMx8Ci6gwEZHx4DQTkUp7+ZkXn7+f/7/Rl9fKvHFixXD9+nV9VmsdQdy7fvnkyGgm5s7p+dGYcx
hGTFiBB48ePMn/3pp5/Kn71kyzEpaeBsZ29b/bd/8pkk4/VKhWc4+C5/Fj
u3cvxPFF1ET//zJkz6NevH1xcxcZsr3uu4tMq3hdB5e1ke1Tqq6rKr/wq9fb/bqq4fDhw/n+AD6ySjXZLTv3L
i54o34WnTpqF79+64fPly1s/asmWl5j/asmWL/FiMbs5nz59B9atWkM6d27i7Yd
dvXv3scft7OxkQBB/Lly4EI0bN3yA77//yJXfbzuZDMNzjuQLbhxMnFK7hJ+gSJhbOJ5Q5/Kn78u4r6Z3dRe4NlVw
7Nl544QV8/fXzIfZGj2uQKjh5EwkdlhrJNvjgbq9Ia9pwfmK5a
rqmsXd4BB/Lly4EI0bN8YHH3yA77//5mtN+x2ZGfvKhPNMwbWSCUqjPEZ9uT
W5c2Lq6nFyhTh4HRrCmNTuR2VVB/gbVS2pGHvYwvv7/+66IP9+Lmz+EzUhK8xVy
eyYxyfM3sMkYhhPbqdSlhxhnAL9+I2NPbEqGDsvSd36Cc8xXmv3AzfJKhZ77dLTpB1IrVKDo28ZwqeJmRRXbYMaQj8K
IPj1Ev2SdS+hLf5jGm++SSc1n4B7bjIXu9u3E0Ws7zNidLTNvL2jB8izQRzNIrrs
8KFqJEhWzMW/aXgC0mxZSATqIakt4PrBIRg6Xs3X4NFUW7PpG2ORHLlDlrWeZsE
P0jJ9Gr3TS3HvHr2oKLZMtLJ9X9rQHkDOhDW7RSrmDG3LE8U99rHWBCIZUPA1zR
s4FZk+rZF55LxlBXiVYx1QH7SdXTrFF+ejRlhQ+BfcDyNx8CSYXpCtptdVu9F5eV
9xVSvBxZ56dukTKWovOHo1vS9CCtiwTlvNdh/DOqaEk2zSztmkBCiLeDvbJ1IuqfR
KBT6T/6+7eBHt8TqdXjuFudeJT3YGIY2CLBsUB1abWjSv1tN3Ie6KmLt8jkBjT
4ELlQMXJuPS+xaQyYs8bQh/Ejgbg0aASIjJNdwXRvrWi1PuJm1oX8XmeyrCZ9wa
bn3+WbpQgAnYdQtwQIoT8rSuVpsBvlv2aR6+L9Ub2U/M0ojmFfZV7gqS0iUA2lzV3
x0gRoCLYcCEaM9OzZjp+PQ2xZEXzEMuU67h8bIOiMaNGi+U07nZDcVuN3sYsLDs
y6F7Mph9fgn1Hkf1MNkSmgOIaiUB3BPEgiDmfWq18Z+V4T4/0XfufhRVyb/9JO1j
sL79NXIe5dHP6qvjsPTl+Iw/FGrF4rhcCqU/2uKn/XE7sYehAK2rjB6tdm6Xhk
qyTdiRgpqrtxvzgFDVZrhqmZz3YJPbd9pN4HWHP3+XLsiJfXl9J1Tdmc48aDgCv
qk7MhiNM1yDl4QC5a0/Pk16iaXXdsg43UHoWT+Q2gNjx7k25vnuRrSzUVpw7vqyk
g/VK/rQWoDzMkWd2F49uFK+2FQWM4oLPvG7YvDOCLFZPhoX4ATIlm9zKpGTaiSbOS
ExsSwlWK8a3uNjtSSHJKJAnMM+VlltNNcCO6dA3ZdW5KVfwpVwo6fR1OR/XyJgMO
6vDzz6kcaH6gcvo4QZMCSxdjpoJWsVzMjJmT3rMdQ6J+bAAV3m5NIGsgIkjGAqA
z2uXDNOlnLj9nVMD/4ftLxM9U3kv6W4BH5nXwaBUb8bCsL/ERBnmQ7sLc6BwN+FO9e
nQaEb1+gRZN4K8mpB+cqnOsSeyTDqhpEFDXg+gYFT1rJHT3q4/7h1mO9KRmI8vE6
jwMlPFH7cO9J2c7GyPFIEBUWc1Tg2H73b/L9m4RFVNoZp3tZBDdCRrGHSfsQJHP2
vv8+1GdbvBbbY3+7G3UzZBIbJCk9lSZVg/zr60jXqRCVPE/JIazZmd2x4bLzRXB1+7
nHzVERWhJ8mC79CCNx8uEvmw/WW9bFXb5Tfa3Z7qxHxDmXZnjE1Y6ltp+FDhy4zX8
+G31u0fz+9v/oVGWH6ySXV3eo8G6OFn6c89rDYl8aa0WETgZD3bfx0/SKOgb2qA
SOdOY5sO7ftO6NllLcQ5wrFKbodW8r5Uvw6Xg7aqYgbyvQVSnIZmzJ6Ilwe44wg5I
1ttiTbtH9UGlTQkMj6AYuKsbRGiJ/ZcGyPwKF1OuQFikpuiGjSkCXbF9RvOhKhgdb
HPr/e+rUUpq3wYgWmqJUA26eRMYIzmQ7fMxMcGRWhSFsXOE6KRy14O6fOn61Qx3e
Kkw/kwqO0TkSO76rXlJ9LKF7SvWg3DO4sEqrlO6G3Q/N8zdeonrL2MGtIhRjK/VPBu1
MGRJ2uDPHn3NH/Ybp7sjc7vs1/fo/zv1KSZRO/OjiJX7/HfgI+OeiBTGsnv2f8pV/
Jkn76/ffjrUswR6D+fNrMz7gtG4N/fzxgzefi79dEA5tYnF4Z85+gkqPMNHx5LIGDOX
Z7bkSxzVLOJoWbhDM7x2ui0BfBu1VSRKxI60BJ4hzBUG8IyXnFDdwAwHk4sK72mj
AHS1b3/DQXwK8twB/IPW6J6vmBXV0D6BCVXb+Pwy3e3ohVUnbnDnNfpz+v8ZnGAv
a5c3HxWKjKgPYxnDh/4SQGcKORYDK65L76RwmS+mBOJ9O0ekuLE9x7tU08uLMVaQ+
2FuPRgFLXjxZqd/p3ZdKc2xBo97KZRXddYgZGjdv9DFXzKdx4a/TmZ3MnGmdpT73
/+g+uydVbujyfs5VjmlV8Z62CecPOX/QjJRHRxzTUQPk1tNsJmH/96Y2eEwWbp5zF2n
j9EwJ3MZVh6AEUDPIVj2Gd3U6eLmq8p/Pno5SKIeFJc31s4RRXIsIGeOQxORLVm3
0IBkAy9dVcGYHhZhjUMgRvyYtmSmXYdqBy20NGNrjfOoLWHPrwJLgFTrK1sF2q0gjY
WIXLjvDnEbbBRFFbqIpJWz4YDr6Zxl4UtQpqK84p91Z7gvF1HZfcTSb4lz0h7fyJlK
0c3qD8kWu5RC9o1jrmdEi8v6t7uWrNLnT0z3DPxQjxdhOtqpzM89xSMPkjK4yt5s+V
1HwqNm00GdI/TW0pP93blW3XfAuVZE3gT4VD2JrKnZtRW7qKEhSO/bcwrtImdSU5w8J
0l0Q96Yp77sivH4S9tRc3vnvn68SNoIRT+vXyTnfhLU8BhG/v+oyohl/zu9x6zBhI6DI/
ZJwk+Ojst7jvrhqSxbwH4+XoqZkgdNvynLOz0aZuDw9BQd7P1VWd8yV8fvcjCoTUHHvY
IsVtbe0LUgTRP3dlI4A2ydxxyV43bQFVSxjKX98PNzfy5/lMwKaWXeL7lzDJOaW06W
hOoqGcnjsOFw+88bqSOjfvDzuNbS3MLaGz1JABpyTfrONEfDA5Z8oQ63yCXzKonWQpTR
wDjMsJgF/yUUlMUC0b/5HoeE1LHbnE4Ubcg4mqvz1QTS/FbmPHTGWAZvexbtPDH3Y
SRsNKy0LvS0c6+fwDJqb+YUFgBIeTg2OIXXn3sQ6yN5mehbmMfXz52kQWtKY5w+Wh
gzcBJfcVJZezBUX5AWRW2zV9Fz8Gkc5ZC9Pn+ZY0yBkLz6+AyKrZMWRkgA3fwmW+
EJJGS1oyqqNZMpzyMbBYMXnmYZC+I+UnAXyVKuCvWIpGUEpqLrNbYlDIoEcsTHBMr
YLe4vy4xAQz6r7YcxJ6l3w3DpDZ4u+Sq2tEqgaC2VVM4hgovJytpu3kSuk+Uzvf9gXd
zLLjb0N6xN2V8L9mR7C5vIP2zU/dTeGwUKYlgNWDQLWHV7n+F51BjKb2UBnQ2vrwB+
S9zA6nCy9pDWD+bMGkgHQZXYIKHFprhUiV9ROq61yxkxYs6vW3OQqr1LF0AYILM4
HCTqbAPdXS7Ax7/xNaagDq6Gs+7tMuN2uYSB+GBpqhEH+XXpApq5Qqm7y7+/3jcs
MNsGovYWdqgAvc3Qu7ZmRZKQ3H2cDpqfWnOwWcl6gAnrKSRbB1bGfnkTUlPZPcDNi
Lh3ueVVzy0c6k+ZPDQOxgoKQH9nmDbs66eC4/KPlBXCuT7DxW/TyYtPuhNjr0+oRe
UYAQLZJCXmBnk4X9bafhnDebI8UTQ+j7rQEnv7hdb7RUlDG0Pn0svjr/0i6xJgtlJxbUlE
vW1L57rIIj+hv+3zRBrmrCEFmrNMpNeWuWyLcA/K/d2CwUGyVG5wnKHFcT6HXQbLRLQ
uQ1wEq91ER1AuIHyNPSl7G+FZ8CL4sLHrjzIzlPn+JKLvFeE8cjvCqKAaSj9DHQzw
/EcAdSyESTjWTwsqnXWoH2uHgEZB0ftlIbAeKiQ4mHHOAJqkfGh2ALWs2Qxk0nIX
OOZf4HjxcL/4g8VAJGdu6RW1dwzU+8d1TRfPZ4QsSAMRnGdMf3hMNgOjMCGO2ul19
bTkGGnx+9yDs2NIJ9m4RqnL6OGTeFt+ZDk04PpDdH/Bqf+Wz4xqsh+vtWYA3KM
xfPGnDI/0fyc/LQ3ACQcnhq7Dr8AtSp2EyK2mrT4ISpkHmHWYBZOVBRpGSdfZ9d
l8OdjqTjGzEOQWf9PHygzpbZqNgAbRXjEPpN8sv3xrvZjpBy8JzlF+/fLUL+WMU/q9
TxXhWYKVpeIoMhRr4hIcWS3Znd0pTrJ3Qp9czD9qqi+R0tbLGrSXfq0a1HgmDi5a6T
aSzIyaEHxEbn1Ln+6GhfEDjS2cvz9AIK8Pk0IUcDjyf7i9gmfAGXJPV7N3C/TmWpMG+
"image/png": "iVBORw0KGgoAAAANSUhEUgAAAjMAAAHHCAYAAABKudlQAAAAOnRFWHRTb2Z0d2F

```
btpSVFFZRiB7HnhkiIiJSNYYZIiIiUjWGGSIiIlI19swQERGRqrEyQ0RERKrGMENERESqZvSb5onrkty4cUNufFWYW58
TERGR/oidY8TmnGInbnNzc9MOMyLI6Ot6MkRERFS0xHXhvL29TTvMiIqM7mCI7ceJiIhI+WJiYmQxQvc+btJhRje1JII
MwwwREZG65KVFhA3AREREpGoMM0RERKRqDDNERESkagwzREREpGoMM0RERKRqDDNERESkagwzREREpGoMM0RERKRqDDN
ERESkagwzREREpGoGDTNTp05FvXr15HUX3N3d0a1bN5w/fz7Hc5KSkjBy5EiUKFECxYoVw8svv8svv4zbt28bbMxERESkLAY
NM7t27ZJBJTg4GFu2bEFqairatm2L+Pj4rOeMGzc069atw8qVK+XzxVWwu3fvbshhExERkYKYaTQaDRTizp07skHs/uYm0NVQsbz3DT+mjNyNSknIrx0vvKgZkckOqqrO1OurZS0+xKsp5Wyv+1TiXCKxmfIUWnk9PKjL+wBHs/wWp1s1ygxTqJB6u9z9uUEdsWJHVi7j+dcKjq
zZs0QHR0NNzc3LF26FD169JDPOXfuHCpXrowDBw6gQYMGebrqprOzs/xevNAkERGR/sQlp+HszRjUK+MKfcvP+7eiemb
EgAVXV+1BOXr0qKzWtG7dOus5lSpVQn/vrwwzT5KcnCwPPYFbERER6d/C/VfQ89cDmQkLQ6FIakmDCTkZGBsWPHonHjxqh
WrZp87NatW7C2tkbx4sVzPNfDw0N+Lrc+HJHkdDcfH58iLyIIlRbkMxZzdl+X9+v76r8yoMsyI3plTp05h2bJlz/V
9Jk2aJCs8ultkZKTexkhERET/r8pEJ6YiwM0BLwaWhiFZQgHeeustrF+/Hrt374ebnHDhw9zVGfEaib
xuSexsbGRNyIiIiocMUmp+G1PuLw/plV5WJibwWQrM6L3WASZNWvWYPv27fD398/x+Tp16sDKygrbtm3Lekws3Y6IiED
Dhg0NMGIiIiJasE9blSnr5oDONbwMfkAsDT21JFYq/f3333KvGV0fjOh1sbOzk38OHToUO6kEmLyu
ZiIiISL9EiJm7R9srM1oBVRmDh5lZs2bJP1u0aJHj8fnz52PQoEHy/vTp02Fubi/vTp02Fubi43yxMrldq1a4dffvnFIOMlIiIydfP
3hSMmKQ3l3IspoiqjuH1mCgP3mSEiItJfVabJ19sRm5SGn/rWQpfAwgszqt1nhoiIiJTr973hMshU8CiGTtVLQSkYZoi
IiOiZohNSZZgRxrSqAHMRMwQERHRM83bexmxyWmo6OGIDtWevD2KoTDMEBER0VM9TEjB/uuyPtjWpdXVFVGYJg
hIiKip5q7J1xeVLKSpyPaV1VWVUZgmCEiIqJcPYhPkcuxhbEKrMoIDDNERESUq7l7LyM+Sz0jmD5tLhYIVJeBv
6ovvxKfLSBUquyggMM0RERPREv+3RVmWqyKqMB5SKYYaIiIgecy8uGQv3/78qY2amD/TywkL+eGDL78q+Q
T2ii4KiMwzBAREdFjVZlF+6/K+2NbVVB0VUZgmCEiIqIc5uy+jMTUdNTwdkaryu5QOoYZIiIiynJXVGUGUOXFVFr4wOwww
RERFlmb3rkqzKBHo7o2VF5VdlBIYZIiIikqJik/BHsK4s5GQ+xeGR2GGSIiIpJm77qE/7Lwkqz J1/FzQtHx
JqA3DDBERkQm7FZ2EpYciVNkro8MwQ0REZMJm7QxDSloG6vq5oEk59VVlBIYZIiIiE67K/HkoUt4f10Z9vTI6DDNEREQ
m6hdRlUnPQP0yrmhUtgTUimGGiIjIBN14mIhlmVWZsW3U2SujwzBDRERkwlWZIH9RlVFnr4wOwwwREZGJuf4wEcsPZ1Z
lWleA2jHMEBERmZiZO8KQmq5BgwBXNFRxr4wOwwwREZEJufYgASuPZK5gMoKqjMAwQ0REZEJm7rgkqzJi9VJQgPqrMgL
DDBERkYmIvJ+tKtPGOKoyAsMMERGRCfXKpGVo5E6/9cq4wlgwzBAREZlIVWbV0WtZ12AyJgwzREREJuCn7RdlVUZcFbu
uEVVlBIYZIiIiIxdxLwF/HbtuNPvKPIphhoiIyASqMukZGjSr4r4IY6fi4wNgwzREREruzK3XisPq6tyowzsl4ZHYYZIiI
iI/bT9jBZlWlR0Q21fI2vKiMwzBAREmp8LvxWHP8mtH2yugwzBAREmpn7ZdRIYGaFnRDTV9isNYMcwQEREZoct34rA
2xHhXMGXHMENERGSkvTIZGqBVJXcEGnFVRmCYISIiMjKX7sThbxOpyGGMBhZvfu3ej
SpQu8vLxgZmaGtWvX5vh8XFwc3nrrLXh7e8POzg5VqlTBr7/+arDxEhERKV1YVCz+OXHDK/BpMgwEx8fj8DAQMycOfO
Jn3/77bexadMmLF68GGfPnsXYsWNNluPnnnn3+KfKxERERqMGNbGDQaoE0VD1QrbfxVGcHSkD+8Q4cO8pab/fv3Y+DAgWj
RooX8ePjw4Zg9ezYOHTqEF198xPvWdUU3zPWWdfahpGWXYW5fv06NBoNduzYgQsXLqBt27a5fk1ycjJiYmJy3IE
iYmJy3IiIiEzBjG0XXZVUPVPUyjaqM4sPMTz/9JPtkRM+HtYn6ekkmjVrluvX
RIVy4HYsNJ2+azAomVYWZ4OBgWZ05evQovvvvuO4wcORJbt27N9Wje3+n5fk1ycjJ
OJnUSDNoz8zSJiYl4//33sWbNGnTq1Ek+VqNGDYSEhGDNyDG7r6m8hFrA
mypu5ec4hWlhYICMjw2DjIiIiUmJVRuhY3fSqMgavzIh9ZMLCwrI+Dg8Pl5UXV1dX+Pr6onnz5pg4cSLs7OxQunRp7OrqqqmjbhpJdDzCD0Qmrp4tz4Mu7nvzwf4GD1NjN9Fsmw+VbHWr94ncC19/sbOy+/Zs3KBE7lR7o3Lf3Q4a5F+Eq0w7h0m1Mt
CokWL8P333xty3ERERIpx9mYM/j11hJhWptUro4gwc+TIEbRz2TLHvjTjTPA90/fH/fv35eBEZsq
UKXj99dcNOGoilVmVKoaKnI0yRQcOM2D9GLLnOjaenJ+bPn1+kYyIlKL0zeisem0tioz1gR7ZRTfM0NERER5q8p
0ruGF8h6mWZURWJohIiJSkNM3orHp9C2YibOdmvCrsmJM+uYtuFZnxaiJS
DRESkMj9svSD/7FKjFMq5m3ZVRmCYISIiUoiT16Kx+cxtmJvorkyigwzREREChs5SFZIRNmYkJKWOfvVkmwHk2W21yMN/WZUWz8jE7hHaHR7aR8kmDSj
Qzi4+hrYy5fcnkmIdLTRrJNEkUwQ5NUqPLK6cLcm2Mtv4
jaR6/sRaRt5QYxfOYuJ2o/brGKFYh5zMu3zHaU3JoDqH3nr93ML0U2pRR0wyRwP5e/AhBFxbKFJBLfZmZfcuwwJ7OTWh03e3brD0pq2hoTM07ZNcfy+fTwd5y8sH62QQKNmspnSHvTNvG5fLaHr2kOBazsgCjoixc8CFeQ2Hd9yWNZNfsmRW+yHbFgG/RWRo8DPdMBO6ekUnAnRYRE/KcBW4HRmoNVAiRxumdFnpFLYZMh
```

```
JTZs2Yd26dXKZNhERkRKtO3EDl+7Eo7i9FQY1ZlXG6MPMkSNHZEjRefvtt+WfAwcOxIIFC2TDr+iPEU29o0ePRsWKFfH
XX3/JvWeIiIiUJi09I0dVxpFVmSJhphHXJDdiYmm2WNUkmoGdnJwMPRwiIjJia45fw7jlJ2RVZu+EF+S1mKjw378V2zN
DRESkvqpMWFZVhkGm6DDMEBER6cHfITcQfjceLvZWGNiIvTJFiWGGiIhID1WZn7Zre2WGNyvLqkwRY5ghIiJ6TmuOX8e
VewlwdbDGgIbcPqSoMcwQERE9h1RZldH2yoxoFgAHNv0WOYYZIiKi56zKRNxPQAkHa7zKqoxBMMwQERE9V1VG2yszonk
A7K25FNsQGGaIiIgKaPWxa4i8n4iSxazxSgP2yhgKwwwREVEBpKT9v1fm9eZlWZUxIIYZIiKiAvjr2DVceyCqMjboH8S
qjCExzBARERWgKvNzZlXmjRZlYWdtwWNoQAwzRERE+bTq6DVcf5gIN0dRlfHl8TMwhhkiIqJ8VmVm7sisyjQvC1srVmU
MjWGGiIgoH1YciZRVGXdHG/RjVUYRGGaIiIjyKDktPasq82YLVmWUgmGGiIgoj1YcjsTN6CR4OtmiT332yigFwwwREVE
eJKWKqswlef/NlqzKKAnDDBERUR57ZW7FaKsyvev58JgpCMMMERFRnqoy2l6ZkS3LwsaSK5iUhGGGiIjoGZYdisDtmGR
4OduiF6syisMwQ0RE9IyqzC87db0y5ViVUaDnDjPp6ekICQnBgwcP9DMiIiIiBVl6MAJRsckoXdwOveqyV8YowszYsWM
xb968rCDTvHlz1K5dGz4+Pti5c2dhjJGIiMhgVZlZu7RVmZEty8HakhMaSpTvs7Jq1SoEBgbK++vWrUN4eDjOnTuHceP
G4YMPPiiMMRIRERnEkoMRuJNZlelRx5tnwVjCzN27d+Hp6Snvb9y4ET179kSFChUwZMgQnDx5sjDGSEREVOQSU9IxK7N
X5q0XWJUxqjDj4eGBM2fOyCmmTZs2oU2bNvLxhIQEWFhwqRoRERmHJQev4m5cMrxdWJVROsv8fsHgwYPRq1cvlCpVCmZ
mZmjdurV8/ODBg6hUqVJhjJGIiKhIJaSk4dfMXplRL5SDlQV7ZYwqzHzyySeoVq0aIiMj5RSTjY2NfFxUZSZOnFgYYyQ
iIipSi4NFVSYFvq726F6bvTJGF2aEHj16yD+TkpKyHhs4cKD+RkVERGTAqszsXZezemVYlVG+fNfNRK/M559/jtKlS6N
YsWK4fFl7wj/66KOsJdtERERq9ceBq7gXnwK/EvboXqu0oYdDhRFmpkyZggULFmDatGmwtrbOelxMPc2dOze/346IiEg
x4pPTMHt3ZlWmZTlYslfGOMPMokWLMGfOHPTv3z/H6iWx94zYb4aIiEitFh24ivvxKShTwh4vsSpjvGHm+vXrKFeu3GO
PZ2RkIDU1VV/jIiIiKlJxyWmYs1u3gqk8qzLGHGaqVKmCPXv2PHFn4Fq1aulrXERERERVq4f4reJCQCv+SDuha04tH35h
XM02ePFmuXBIVGlGNWb16Nc6fPy+nn9avX184oyQiIipEsUmp+G2PtldmdCv2yhh9ZaZr167ymkxbt26Fg4ODDDDdnz56
Vj+l2AyYiIlJbr8zDhFQEuDngxUCuYDKJfWaaNm2KLVu26H80RERERBqjKzMlcwTSmVXlYmJvxHKgM92cmIiKTtmDfFUQ
npqKsmwM612CvjElUZszNzeU1mZ62qR4REZEaxOTolWFVxmTCzJo1a3J8LJZjHz9+HAsXLsSnn36qz7ERERERVqvl7ryA
mKQ3l3IuxKmNKYUY0AD/pWk1Vq1bF8uXLMXToUH2NjYiIqNCIqaV5e9krYwz01jPToEEDbNu2TV/fjoiIqFDN3xcuqzI
VPIqhU/VSPNqmHmYSExPx448/yotPEhERqaMqEy7vj2lVAeZcwWRa00wuLi45GoA1Gg1iY2Nhb2+PxYsX63t8RERERee
CTGxSGip6OKJDNU8eYVMLM9OnT88RZsTqJjc3NwQFBcmgQ0REpGTRCamYr6vKtC7PqowphplBgwbp7Yfv3r0b33zzDY4
ePYqbN2/KlVLdunV74nNff/11zJ49W4apsWPH6m0MRERkWubuvYzY5DRU8nRE+6qsyphMmAkNDc3zN6xRo0aenxsfH4/
AwEAMGTIE3bt3z/V5IuQEBwfDy4ubGRERUcE9TEjB/H1X5P2xrMqYVpipWbOmnFoS/TFPI56Tn03zOnToIG9PIy5oOWr
UKGzevBmdOnXK8/cmyouk1HTYWlnwYBGZiLl7whGXnIbKpZzQtgqrMiYVZsLDtXOLRU1clfvVV1/F+PHj5T42eZGcnCx
vOjExMYU4QlJziHlr6TEcuHQPcwbUReNyJQ09JCIqZA/iRVVG+37GqowJhhk/Pz8YwtdffW1LS0uMHj06z18zdepU7kR
MT5WYko7XFh3B3rC78uPxK09g07hmcLK14pEjMmLisgXxKemoIqsyHoYeDhn6qtnCmTNnEBERgZSUlByPv/jii/oYl2w
KnjFjBo4dO/bUa0E9atKkSXj77bdzVGZ8fHz0MiYyjiAzbNFh7Au7B3trC7jYW+P6w0RMWX8WX/fIe78XEanL/fgULN
z//16Z/LyvkBGGmcuXL+Oll17CyZMnc/TR6H4x9HWhyT179iAqKgq+vr5Zj4nv/c477+CHH37AlSvaX8pH2djYyBvRk4
LM0IWHsf/SPThYW2DBkPry8V6zD2D5kUi0r+6JlhXdeeCIjNCc3dqqTLXSTmjDqozRyfcOwGPGjIG/v78MGmKjvNOnT8
sl1nXr1sXOnTv1NjDRKyNWUYWEhGTdxGom0T8jmoGJ8iMhJQ1DFvw/yCwcUh/1yrjK25DG/vI5E/8KlbuCEpFxuReXjE
UHMqsyrSqwKmOE8l2ZOXDgALZv346SJUvKDfPErUmTJrJXRfS2iCto51VcXBzCwsJyNBqL0OLq6iorMiVKlMjxfCsrK3
h6eqJixYr5HTaZMF2QCb58H8VsLLFwSD3U8XPN+vy7bStix7koXL4bj8/Xn8G3PQMNOl4i0q85ey4jISUdNbyd0aoyq
6/GKN+VGTHV4+joKO+LQHPjxo2sJuHz58/n63sdOXIEtWrVkjdB9LqI+5MnT87vsIieKD45DYPmZw8y9XMEGcHO2gLf9
AyEuDTLqqPXsO3sbR5NIiNxV1Rl9l+V99krY7zyXZmpVq0aTpw4IaeaxCUMpk2bBmtra8yZMwcBAQH5+l4tWrR45t412
eXWJ0OUW5AZPP8wDl25D0cRZIbWR23fJ19yo46fC15rGoDZuy9j0uqT+G+cC4rbW/PAEhlBr0xiajoCvZ3ZE2fE8l2Z+
fDDD+X+L8Jnn30mp4aaNm2KjRs3yitnEymB2BRr0PxDWUFm0VOCjM64NhVQ1s0BUbHJ+HTdmSIbKxEVjjux2XplWrNXx
pjluzLTrl27rPvlypXDuXPncP/+/ceupk1k0CDz+yEcufoAjraW+GNoEGr6FH/m14mdgEW/zMuz9mPN8evySrpted0WI
tWavesSklIz5N//FhXdDD0cUlJlZvHixfKaStmJhl0GGVKC2KRUDMwMMk62llgyLG9BRqeWrwtGNC8r77+/5pTcMZSI1
CcqNgmLD7JXxlTkO8yMGzcOHh4e6Nevn5xa0te+MkT6CjJHs4JMA9TwznuQ0RFNghU8isnGwY//Oc0TQ6RCs3ddllWZW
r7F0bwCqzLGLt9h5ubNm1i2bJmsxPTq1QulSpXCyEjsX///sIZIVEexCSlYsDvh3As4iGc7ayw9LUGqO7tXKBjZ2Opn
W6yMDfDPydu4N+TN3kOiFQkKiYJi4O1VZlx7JUxCfkOM+JaSZ07d8aSJUvkxnnTp0+Xq4xatmyJsmW15XmiIg8y8w7he
MRDFLe3klNL1UoXLMjoiIrOG5nTTR+uPSU33SIidZi16xKS0zJQ27c4mpbnRWRNQb7DTHZiB2DRENyhQweUL1+eS6epy
Ikde1+ddwghkfoLMjqjWpVDJU9H3ItPwWRONxGpwu2YJCw5GJG1QpH9nKahQGEmISFBVmY6duyI0qVLy2slies1iUsbE
BWV6AQRZA7iRORDuNhbYemwBqjqpZ8gk326ydLcDBtCb2J9qHaDSCJSrlk7LyElLQN1/VzQpByrMqYi32GmT58+cHd3l
43AYpM8cT0mcUmCzz//HJUqVSqcURI9Ici8Mu8gQq9Fw9XBWvbIVPFy0vtxElWeN1uWk/c/WntK7ltBRMp0KzoJSw+xK
mOK8r3PjIWFBVasWCGnl8R9oqL2MCFFBplT12Myg0wQKnnqP8jovNWyHLacuY2zN2NkoJn1Sm2WrokUaNbOMFmVqV/GF
Y3K5ry2Hxm3fFdmdNNLDDJkqCDTf642yJRwsMafrzUo1CAjWFua47vM6aZNp29hXShXNxEpzc3oRPx5KFLeH9umPP/BY
WKeqwGYqCiJDez6/XYQp2/EoGQxa/w5vAEqemovelrYxBTW6Fbl5f3Jf5+SG3IRkXL8suMSUtIzUN/fFQ0DWJUxNQwzp
Ar3RZCZexBnbmYGmdcaoIJH0QQZnTdalEW10k54mJCKD9acytdFUomo8Nx4mIjlh7VVGe4rY5ryHGZu3OBKDjJgkPktW
PaslCxmI4NM+SIOMoKVhblc3WRlYSZ7aNaGXC/yMRDR42buCJNVmQYBrmjIXhmTlOcwU7VqVSxdurRwR0P0CLFZnQgy5
27Fws3RBsuGGybI6Ij+HHH1XeHjv0/LPS2IyHCuP0zEiiP/r8qQacpzmJkyZQpGjBiBnj17iBnj17yqtkExU2cW0k0SOjCzKiI
lPOvZjBD/yIZgGo4e2MmKQ0vL/6JKebiAxclUlN18jVS0HslTFZeQ4zb775JkJDQ3Hv3j1UqVIF69atK69JkRkUnTBplgn
L8dC/fMiowSgoxgaaFd3WRtYY5t56Lw1zFONxEZwrUHCViZWZXRVUzJNOVrnxl/f39s374dP//8M7p3747lSvLazVld
```

```
+zYMX2PkUyM2JhOBJmLUXHwcNJWZALclBFkdMRUl9gq/etN5/DputNoXK4ESjnbGXpYRCZZlRF//8QqJjJd+d407+rVq
1i9ejVcXFzQtWvXx8IM0fMQS57F1FJYVBw8nWzl8mv/kg6KPKivNfXH5tO35HWhJv51EgsG1+PeFkRFJPK+qMpck/fZK
0P5SiK//fYb3nnnHbRu3Vpeh8nNzY1HkPQmKiYJfX8LxqU78SjlbCsrMmUUGmR0001idVPHH/dg14U78n+sver5GHpYR
Cbh5+1hSMvQyKti1y3Dqoypy3PPTPv27TFhwgQ5xSQqMwwypO8g0ydbkBE9MkoOMjqij2d824ry/ufrz8iVFURUuCLuJ
eCvY9qqDHtlKF9hJj09XTYADxgwgEeO9Eosb+4zJxiX78TDKzPI+JVQfpDRGdLEH3X8XBCbnIaJf4VydRNRIft5x0VZl
WlWwU3+3SPKc5jZsmULvL29ecRI71e5lUHmbjxKF7fDsuENVRVkBAtzM3zTowZsLM2x5+LdrOvDEJH+Xb0Xn7WCcGxr7
SVGiHg5AzJokBE9MuFZQaYBfEvYq/KMiNVW77WvJO9P2XBGNicSkf79tD0M6RkaNK/ghtq+rMqQFsMMGewKt33mHMgRZ
Hxc1RlkdAY3KoP6ZVwRn5KOCX+FIiOD124i0qcrd+Ox5ri2KiO2RiDSYZghg1wUTkwtXbmXAG8XOywfof4gI5ibm2Faj
xqws7LA/kv3sORQhKGHRGSUVZmWFd1Q06e4oYdDCsIwQ0XqemaQuXovAT6uIsg0hLeL+oOMjliBNaG9dnXT1I1n5aoLI
np+ooq75jhXMNGTMcxQkW49LqaWIu4nwNfVXjb7iikmYzOgYRkE+bsiISUd41ed4HQTkR78tO0ixMxtq0ruCGRVhh7BM
ENFQjTEiopM5P1E+JUQQaaBUQYZ3XTTNz0CYW9tgYPh9/FH8FVDD4lI1S7dicPaEN0KJvbK0OMYZqjIgsy1B4kokxlkv
Iw0yOiIVVmTOlaW97/695xsXCSi56vKtK7sjurezjyM9BiGGSqSICN6ZcQ1lsTUkqlckLF/fV80KlsCiamcbiIqKHGdt
n9O3JD3WZWh3DDMUKERza+6IBNQ0kFea8nT2dZkjriYbvr65RpwsLbA4SsPMH//FUMPiUh1ftqurcq0qeKBaqVZlaEnY
5ihQtulUzT7ZgWZ4aYVZHTEkvMPOlWR96dtOofLd+IMPSQi1QiLis1WleFuv5Q7hhkqpCATjBvRSQhwE1NLDeDhZHpBR
qdvfR95Zd/ktAyMXxUq98kgomebsS0MGg3QrqoHqnqxKkO5Y5ghvRKNrr1nB+NmdBLKZgYZdxMOMoKZmXa6ydHGEkevP
sDve8MNPSQixbtwOxbrQ7VVmTGtuIKJno5hhvS6qVXvOQdwKyYJ5dyLyWZfd0fTDjI6YvXWh521q5u++e+8bGokotzN2
HZRVmXaV/VEFS8nHip6KoYZ0gvRCyJ6ZG7HJKO8ezHZ7OvmaMOjm02vuj7y4ngpaRl4d+UJTjcR5eL8rVhsPHlT3h/DX
hnKA4YZ0suGVqJHRgSZCh7FZLMvg8yTp5u+erk6HG0tERL5EL/tuczfPqIn+DGzKtOxuicql2JVhp6NYYaei5gu6TsnG
FGxyajo4SgrMiWLsSKTG7HHzsddqsr73/93ARdvx/I3kCibc7disOHkTZiZsVeG8o5hhp5r2WTf37RBppKnI5a+FoQSD
DLP9HLt0nihkjtS0jPwzsoTSEvP4G8hUaYZWy/KPztWL4WKno48LpQnDDNUIKKi0GfOQdzJCjINGGTyMd00tXt1ONlaI
vRaNGbv5nQTkXDmRgz+PXUrsyrDfWUo7xhmqEBBRlRk7sYly/lsMbXk6mDNI5kPYt+dT7tqp5t+2HpBltaJTN2MbRfkn
52ql0IFD1ZlKO8YZijfqwxEs+/duBRUKeWEpcOC4MIgUyDdapaWW7Snpmvk6qZUTjeRCTt9IxqbT99mVYYKhGGG8hVk+
v0WjHvxKajq5SR7ZBhknm+6acpL1VDc3gqnrsdg1s5L/G0kmHqvTJcaXijPqgzlE8MM5YmYBumbGWSqlXbCkmFBKG7Pq
aXnJTYV/PTFqlnLUcW/TolMzanr0fjvzG2YmwGj2StDagszu3fvRpcuXeDl5SX/lbp27dqsz6WmpmLChAmoXr06HBwc5
HMGDBiAGze021tT0TblieXX9+NTUL20M5YMbcAgo0cvBnrJXU7TMsR0U6jcVI/IlPyQWZURfxfE7uFEqgoz8fHxCAwMx
MyZMx/7XEJCAo4dO4aPPvpI/rl69WqcP38eL774okHGaspBpv/cYDxISEWgtzMWDwuCs72VoYdlVESQ/+KlarKJ+uzNG
MzcEWboIREVmZPXorH1LKsy9HwsYUAdOnSQtydxdnbGli1bcjz2888/o379+oiIiICvr28RjdJ0iSmP/nMP4qEIMj7Fs
WhIfTjbMcgUBrHR4Oddq2Hk0mMyzIjG4GqleZVgMn5iNZ+uIT7AjVUZMoGemejoaPmv2OLFi+f6nOTkZMTExOS4UcHms
Pv9pg0yNX2K44+hDDKFrVONUnJJqna66QSS09IL/WcSGZK4/tK2c1GyV+atF8rxZJDxh5mkpCTZQ9O3b184OeV+rY6pU
6fKqo7u5uPjU6TjNJYgIyoy0YmpqOVbHIuG1oeTLSsyReGzrlVRwsEa527F4qdtnG4i4w4yo/48Lu+/2sCPVRky/jAjm
oF79eoFjUaDWbNmPfW5kyZNkhUc3S0yMrLIxmks89di+bUIMrVFkBnCIFOUxOUgvuhWTd6ftesSQq89LNKfT1QUNoRqg
0x6hgbda5XG5MzrlREZbZjRBZmrV6/KHpqnVWUEGxsb+ZzsN8qbE5EPZbNvTFIa6vi5YOGQ+nBkRabIdaheSq7qEP+jf
2cFp5vIuKw7cQOjl2mDzMu1vfFNz0BYiHkmImMNM7ogc/HiRWzduhUlSpQw9JCMVkjkQ7wy76AMMnUZZAxO7D0jmoIvR
sVlLVslUrt/TtzAmMwg06OON6b1qMEgQ+oPM3FxcQgJCZE3ITw8XN4Xq5VEkOnRoweOHDmCJUuWID09Hbdu3ZK3lJQUQ
w7b6ByPeIBX5x5EbFIa6pVxwYIh9VHMxqAL3Uye2Fn5y5e0002zd12S54hIzf4OuY6xy44jQwP0quuNaS8zyJD+mGlEI
4qB7Ny5Ey1btnzs8YEDB+KTTz6Bv7//E79ux44daNGiRZ5+hljNJBqBRf8Mp5wedyziAQbOO4TY5DTUL+OK+YPrwYFBR
jHGLQ/BmuPXUdbNARtGN4WtlYWhh0SUb2uPX8fbK0JkkOld10deNd6cU0ukx/dvg/7zWwSSp2UpA+Ysk3D06gMM/P0Q4
pLTEOTvit8HMcgozcddqmBf2F1cuhOP6VsuYFLHyoYeElG+rDl+TfZ+iSDTp54PvnyJQYZMrGeGCs/Rq/ezgkyDAFZkl
Epc/0r8K1aYs+eyPG9EavHX0Wt4OzPI9K3vyyBDhYZhxgQduXIfA+Zpg0zDgBKyImNvzR4ZpWpV2UOu+hCFSnHtpsQUb
qZHyrfq6DW8u+qE/L3tH+SLKd2qcWqJCg3DjIk5LILM74cQn5KORmUZZNRicpcq8HCyQfjdeHz733lDD4foqVYcicT4z
CDzSgNfeakO9shQYWKYMSGHwrVTSwkp6WhSriTmDawHO2s2lKqBuCbWVy/XkPd/3xcuQymREq04HIkJf4XKIDOgoR+DD
BUJhhkTEXz5HgbN1waZpuVLYu7AugwyKtOyortcCSLeJMavPIGElDRDD4koh2WHIvBeZpAZ2NBP7pckrqdHVNgYZkzAg
Uv3MHj+4awg89uAulziq1IfdK6MUs62uHIvAdM2cbqJlOPPQxGYuPqkvD+oURl8wiBDRYhhxsjtv3QXQxYcRmJqOppVc
GOQUTlxwc+vM6ebFuy/IituRIa29GAEJmUGmcGNy8gtBViRoaLEMGPE9of9P8g0r+CGOa/WYUXGCIhQKpa5CqLJMj6Z0
01kOIuDr+L9NdogM7SJPyZ3ZpChoscwY6TERmuDFxxGUmoGWlZ0w2wGGaPyQafKKF3cDpH3E/H1pnOGHg6ZqD8OXMGHa
0/J+8Oa+OPDTpVZkSGDYJgxQnsvaisyyWkZeKGSO35lkDE64tpZ4iJ9wqIDV2UVjqgoLTpwBR/9fVreH94sQAZsTi2Ro
TDMGJndF+5g6EJtkGlVyR2zXqkNG0suvzZGjcuVlHt4CONXhcpNEImKwoJ94ZicGWRGNA/ApA6VGGTIoBhmjMiuC3cwb
NERGWRaV/bALwwyRm9Sh8rwdrHD9YeJ+HLjWUMPh0zA/H3h+GTdGXn/9eZlMbE9gwwZHsOMkdh5PgqvLTqClLQMtKnig
V/6syJjCsQVzr/pEZi1omTPxTuGHhIZsXl7w/FpZpB5s0VZTGhfkRUZUgSGGSOw41wUhi86KoNM2yoemNmvNqwteWpNR
cOyJeS+HsKEVaGITUo19JDICM3dcxmfr9cGmbdalsP4dgwypBx8x1O57eduY8QfR5GSnoH2VT0xsz+DjCl6r31F+JWwx
43oJEzZwOkm0q/fdl/GF5m/V6NeKId32lZgRYYUhWFGxbadvY3X/zgmg0yHap74qV8tWFnwlJoicdVzMd0kdo5fdjhST
jsS6cOc3ZcwJbMfa3Sr8ni7DYMMKQ/f+VRq65nbeH2xtiLTsbonfuzLIGPq6vu7YnAjf3l/4l8nEZ3I6SZ6Pr/uuoQvN
2r3MRrDIEMKxjCjQv+dvoU3lhxFaroGnWqUwow+DDKkJfoY/Es64FZMEr7I7G8gKohfdobhq3+1QWZs6/IY16YCDyQpF
sOMymw+fQsjlx6TQaazCDK9a3JqibLYWVvgmx415HTTyqPXZE8VUX7N3BGWdSFTMa00tjWDDCkbw4yKbDp1CyOXaINMl
0Av/NC7JizZI0OPqFvGVW4tnzXdlMDpJsq7n7dfxDebtUHm3bYVZJ8MkdIxzKjEvydv4q2lx5CWoUHXml6Y3iuQQYZy9
```

U7bighwc0BUbDI+XafdqZXoWX7cdhHf/ncha8ryrRcYZEgdGGZUYKMIMn8el0GmW00vfN+LFRl6OlsrC3zbMxDmZsDq4
9ex5Qynm+jpZmy9iO+3XMha6j+yZTkeMlINhhmF2xB6E6P+PI70DA261yqN73rVhIV4hyJ6htq+LhjerKy8//6ak3gQn
8JjRk80fcsFTN+qDTITO1TCmy0YZEhdGGYUbN2JGxi9LDPI1C6Nb3oGMshQvohVKOXdi+FObDI+4XQTPUKj0chqzIxtF
+XH4oKR4npLRGrDMKNQ/5y4gbHLQ2SQ6VHHW26IxooMFXS6Sfzu/B1yA5tO3eRBpKwgIyoyok9G+KBjZYxgkCGVYphRo
L9DrmNsZkWmZx1vfP1yDQYZKrBAn+J4vXmAvP/h2lO4z+kmkyeCzHf/XcCP28O0vxedKuO1ZtrfESI1YphRYJAZtzwEG
Rqgd10fBhnSC7G8tqKHI+7GpWDy36d4VE08yIil1z/v0AaZjzpXwbCmDDKkbgwzCrLm+LWsINOnng+mdq8Oczb7kh7YW
P5/uml96E3ZWE6mGWSmbT6PX3Zekh9/3KUKhmbuSUSkZgwzCrH62DW8veKEDDJ96/vgy5cYZEi/qns7Y2QLbXPnR3+fw
t24ZB5iEwsyX206h1mZQeaTLlUwuDGDDBkHhhkFWHX0Gt5ZeQIaDdAvyBdTujHIUOEQm6BV8nSUfTMfrT0l3+DI+InzP
PXfc5i967L8+LOuVTGIQYaMCMOMga08Eonxq7RB5pUGvviiazVOLVGhsbY0x3di92hzM/x76pacciLjDzJfbjyLObu1Q
ebzrlUxoGEZQw+LSK8YZgxoxeFIvPdXqAwyrzbww+cMMlQEqno5Y1TmNvViuikqNonH3YiDzBcbzuK3PeHy4y+6VcOrD
DJkhBhmDGT54QhMWK0NMgMb+smyr5m41DFREXizZVlU9XLCw4RUfLCG003GGmQ+W38G8/Zqg8yUl6rhlQZ+hh4WUaFgm
DGAPw9FYMJfJ2WQGdSoDD55kUGGipaVhblc3WRlYSav2yQ21CPjCjKfrjuD+fuuyI/FgoL+QQwyZLwYZorY0oMRmLT6p
Lw/uHEZuTSSFRkyhMqlnDCmlXa66eN/TiMqhtNNxhJkPvnnNBbs1waZr7pXlwsLiIwZw0wRWnLwqrzgnzCksT8md2aQI
cMS1+GpXtoZ0Ymp8neTq5vUTZw/EUwXHrgKMWs97eUa6FOfQYaMH8NMEfkj+KrsTRCGNfHHR50rsyJDBmdpoV3dZG1hj
q1no7D62HVDD4kKKCNDIxu6F2UGGXEZlF71fHg8ySQwzBSBRQeuyD09hNea+uODTgwypBwVPBwxto12uklcWftWNKeb1
BpkFgdHZFVketVlkCHTwTBTyBbuv4LJf5+W90c0C8D7HRlkSHmGNw2QF6SMTUrDRLnKjpvpqSnIfLD2FJYc1AaZb3sEo
ieDDJkYhplCNH9fuJy/1vUmTOxQiVNLpNzppp415KZ6O8/fwcqj1ww9JMpzkDkpV0iKIPNdz0C8XMebx45MDsNMIfl9b
7hcGim80aIsJrSvyCBDilbO3RHvtKkg73++7gxuPEw09JDoGUFGrIz881AkxPVov+8ViO61GWTINDHMFIK5ey7LzaqEk
S3L4r12DDKkDsOaBqCWb3HEJqdhgtydmtNNSg0yYjpw+RFtkJneuyZeqsUgQ6aLYaYQgozYPlwY9UI5vNuWQYbUw8LcT
G6mZ2Npjj0X72LZ4UhDD4kekZ6hkZdBWXHkWlaQ6VqzNI8TmTSGGT2as/tSVpAZ/UI5vN2mAqeWSHXKuhXD+HYV5f0v1
p/BtQcJhh4SZQ8yq0Kx6ug1GTxn9KnFIEPEMKM/s3ddwpcbz8n7YlfVt1mRIRUb3Ngf9cq4ID4lndNNCgoy41eewF/Hd
EGmJroEehl6WESKYNDKzO7du9GlSxd4eXnJCsbatWtzfF7M10+ePBmlSpWCnZ0dWrdujYsXL0JpZu28hKn/aoPM2NblM
S6ziZJIrcSb5Tc9AmFrZY59Yffksl8ybJB5d+UJrD5+XZ6bH/vUQucaDDJEiggz8fHxCAwMxMyZM5/4+WnTpuHHH3/Er
7/+ioMHD8LBwQHt2rVDUpJyNvWauSMMX2/SBplxrStgbGsGGTIOZUo6YEL7SvL+lxvPIvI+p5sMIS09A2+vCMGa49dha
W6Gn/vWQqcapQwyFiKlMtMoZLmCqMysWbMG3bp1kx+LYYmKzTvvvIN3331XPhYdHQ0PDw8sWLAAffr0ydP3jYmJgbOz
s/xaJycnvQeZbzafl/fFktZRmRftIzKmVTN9fgvGofD7aBDgiqXDGsBcdJ1SEQaZE/jnxA1tkOlXC+2rMciQaYjJx/u3
YhuAw8PDcevWLTm1pCNeVFBQEA4cOJDr1yUnJ8sDkP1WGLIHGdEsySBDxkgEF7GjrL21BYIv38fig1cNPSSTCjLjcgSZ
2gwyRGoLMyLICKISk534WPe5J5k6daoMPbqbj0/hXJ/E19Vezl2LIDOyZblC+RlESuBbwl7uXi1M3XgOV+/FG3pIJhFk
xiwPwboTN2BlYYZf+osg42noYREplmLDTEFNmjRJlqR0t8jIwtknQ6wi2Dy2GYMMmYRXgvzQMKAEElPTMX5lqJx+osKR
KoLMshBsCL2ZGWTqoG1VBhkiVYYZT0/tX97bt2/neFx8rPvck9jY2Mi5tey3wlLOvVihfW8ipU03TetRAw7WFjh05T4W
7L9i6CEZcZA5jg0ntUFmVv86aFMlZ3WaiFQUZvz9/WVo2bZtW9Zjov9FrGpq2LChQcdGZIp8XO3xfqfK8v60zecQfpfT
TfoOMqOWHsfGk7dgbWGOX1+pg9YMMkTKDzNxcXEICQmRN13Tr7gfESGuAGuGsWPH4osvvsA///yDkydPYsCAAXKFk27F
ExEVrX71fdGkXEkkpWbIDdzE/if0/FLSMvDW0mPYdFobZGa/WgetKrMiQ6SKMHPkyBHUqlVL3oS3335b3hcb5Qnvvfce
Ro0aheHDh6NevXoy/GzatAm2traGHDaRyRL/yPi6Rw0Us7HEkasPMH9fuKGHZDRBZvPp27C2NMfsAXXQspK7oYdFpCqK
2WemsBTmPjNEpmrZoQhMXH1SXpByw+im7B97jiDz5pJj2HpWG2TmvFoHLSoyyBAZzT4zRKRcvev5oFkFNySnZWD8Kk43
FURyWjreXHJUBhkRCucOqMsgQ1RADDNEVLDppperw9HWEscjHmLunss8ivkNMotFRSZKG2QG1pXhkIgKhmGGiAqklLMd
PupcRd7/bssFXLwdyyOZB0mp6Xj9j6PYdk4bZOYNrIem5RlkiJ4HwwwRFVjPOt5oWdFN9n6IqzqLnWvpGUFm8VHsOH9H
XpH890H10KR8SR4youfEMENEzzXdNLV7DTjZWuLEtWjM3s3ppqcFmRF/HMVOXZAZWA+NyzHIEOkDwwwRPRdPZ1t88mJV
ef+HrRdw/hanm54UZF5bdAS7LtyBnZUF5g+qj0YMMkR6wzBDRM/tpVql0bqyO1LTNXK6SexmSzmDzJ6Ld7VBZnA9NCxb
goeHSI8YZohIL9NNX75UHc52Vjh5PRq/7rzEowogMSUdwxZqg4y9tQUWDK6HBgEMMkT6xjBDRHrh7mSLz7pqp5t+3H4R
Z27EmPSRlUFm0WHsDdMFmfoIYpAhKhQMM0SkNy8GeqFdVQ+Tn24SQWbowsPYF3ZPXml84ZD6qO/vyt80okLCMENEep1u
+qJbdbjYW+HMzRjM3BFmckc3ISUNQxYcxv5L/w8y9cowyBAVJoYZItIrN0cbfNa1mrz/8/YwnLoebVJBZvD8wzhw+Z68
GOeiofVRl0GGqNAxzBCR3nWuUQodq3siLUO7uklsqmfs4pPTMGj+YRwMvw/HzCBTx48VGaKiwDBDRIUy3fR512oo4WCN
c7di8dP2i0YfZERF5lC2IFPb18XQwyIyGQwzRFQoShSzwRfdtNNNv+y8hNBrD43ySMfJiswhHLpyX154849hQajFIENU
pBhmiKjQdKheSk45pWdON4mrRRtdkPn9EA5feSCDzOKhQajpU9zQwyIyOQwzRFSoRDNwyWLWuHA7DjO2Gs90U2xSKgb+
fghHrj6Q16ZaMiwIgQwyRAbBMENEhcrVwVou1xZ+3XUJIZHqn26KSUrFgN8P4ejVB3LX4yXDGqCGNysyRIbCMENEha59
NU90q+mFDA3wzooQeb0iVQeZeYdwPOJhZpAJQnVvZ0MPi8ikMcwQUZEQV9YWe9BcuhOP6VsvqPKoRyem4tV5h2R1qbi9
NshUK80gQ2RoDDNEVCSK21tj6kva6abfdl+WUzRqCzID5h3ECQYZIsVhmCGiItO6ige61y4tp5vGrzyhmumm6ARRkTmI
E9ei5aUalg5rgKperMgQKQXDDBEVqY87V4WHkw0u343Ht5vPqyLIvDLvIEKvRctm5qWvNUAVLydDD4uIsmGYIaIi5Wxv
ha+615D35+0Lx+Er9xV7Bh4mpKD/vGCcvK4LMkGoXIpBhkhpGGaIqMi1rOSOnnW8ocmcbkpMSVdmkJl7EKeux8jLMvz5
WgNU8mSQIVIihhkiMogPO1dBKWdbXLmXgGmbzynqLDyIT0G/3w7i9I0YueHfn8MboKKno6GHRUS5YJghIoMQe7R89bJ2
umn+visIvnxPEWfivggycw/izE0RZGxkRaaCB4MMkZIxzBCRwTSv4Ia+9X3k/fdWhSIhJc3wQea3YJzNDDLLhgehPIMM
keIxzBCRQb3fsTJKF7dDxP0EfP2v4aab7sUlyyBz7las3Nxv2fAGKOfOigyRGjDMEJFBOdpa4evM6aaFB65i/6W7RT6G
uzLIHJRBxj0ryBQr8nEQUcEwzBCRwTUpXxL9g3yzppviktOKOMgE4/ztWLn/jQgyZd0YZIjUhGGGiBRhUsfK8Haxw7UH
iZi68WyR/Mw7scnoOycYF27HZQaZhghgkCFSHYYZIlKEYjaWmNZDO9205GAE9l4s3OmmqNgk9P0tGBej4uDpZCuDjH9J

```
h0L9mURUOBhmiEgxGpUtiQEN/eT9CX+FIjYptVB+TlRMkqzIhEXFyb1uxNQSgwyRejHMEJGiTGhfCb6u9rj+MBFfFsJ0
kwgyfX4LxqU78fDKDDJlWJEhUjWGGSJSFAcbS3yTOd3056FI7LpwR2/f+7YIMnOCcflOvFwOLqaW/EpwaolI7RhmiEhx
ggJKYHDjMvL+xL9CEaOH6aZb0ZlB5q4uyDSAbwl7PYyWiAyNYYaIFOm9dpVQpoQ9bkYn4Yv1Z57re92MTkSfOQcQni3I
+LgyyBAZC4YZIlIkO2sLfNMzEGZmwIoj17DjXFSBvs+NhyLIBMsLWoql3wwyRMaHYYaIFKteGVcMbewv709cHYrohNQC
BZmr9xLg48ogQ2SsGGaISNHebVcRASUdcDsmGZ+uP53nr7ueGWTENZ+0QaYhvF04tURkjBhmiEjRbK0s8G2vQJibAauP
XcfWM7ef+TXXHiTIHhkRZMQy7+XDG8peGSIyTgwzRKR4tX1d8FrTAHl/0pqTeJiQqutzI++LIBOMyPuJ8Cthj+UJ
QYbIqDHMEJEqjGtTAWXdOT1lD755/RTg4y4vpNYCSWafUs5syJDZOwYZohINdNN3/WqKaeb1obcwKZTt54YZESvjLg0
geiRYZAhMg2KDjPp6en46KOP4O/vDzs7O5QtWxaff/45NBqNoYdGRAZQ06c4Xm9eVt7/cO1J3I/XTjdF3EtA79kHZJAR
zcKiIuPpbMtzRGQiFB1mvv76a8yaNQs///wzzp49i2+++Qbffvstfvrpp6QPMs8dTPbG/D6Xgz2zc3VVw8K1yaMCmPoTuA+vRqPQ06MHanfgJgXTbfN
hAA3B/w5vAE8nBhkiEyJmUbBZY7OnTvDw8MD8+bNy3rs5ZdflWaxYsX5+l7xMTEwNnZGdHR0XBycirE0RJRUQm99hA
v/bIf6RkaONtZIToxVfbT/PlaA7gzyBAZhfy8fyu6MtOoUSNs27YNFy5ckB+fOHECe/fuRYcOHXL9nuTkZBk+sfT6iLyLj
U8C6ON1top5tEkCnnXkxWZBhkiEyTJRRs4sSJMoxUqlQJFhYWsodmypQp6N+/f65fM2XqadVHz66adFOk4iKnqjXiiPszd
jkZCShhl9asHN0YangchEKXqaadmyZRg/fjy++eYbVK1aFSEhIRg7diy+//DDk4+PDaSYiIiIjnWZ
SdJgRIURUZ0aOHJn12BdffCH7Zc6dO5en78GeGSIiIvUxmp6ZhIQEmJvnHKKYbsrIyDDYmIiIiEhZFN0z06VLF9kj4+v
rK6eZjh8/LqeYhgwZYuihERERkUIoepopNjZWLpr3999/486dO3Xuk+PDJPpCbq3Zs0aREFwcvLC3379sXkyZNhbW2dp+/BaSYiIiL1MZqeGX1gmCE
iIlIfo+mZISIiInoWhhkiIiJSNYYZIiIiUjWGGSIiIlI1hhkiIiJSNYYZIiIiUjWGGSIiIlI1hhkiIiJSNYYZIiIiUjV
FX5tJH3QbHIudBImIiEgddO/beblQgdGHGXF9J8HHx8fQQyEiqICvI+LyxqY9LWZMjIycOPGDTg6OsLMzEzvqVGEpMj
IyGdeN0KN+PrUj+dQ3Yz9/JnCa+TrKzgRT0SQEReZNjc3N+3KjDgA3t7ehz4fozxF9AY/xLqMPXp348h+pm7OfPFF4jX1/
BPKsio8MGYCIiIlI1hhkiIiJSNYaZ52BjY4OPP/5Y/mmM+PrUj+dQ3Yz9/JnCa+TrKxpG3wBMREREExo2VGSIiIlI1hhk
iIiJSNYYZIiIiUjWGGSIiIlI1hplc7N69G126dJE7D4qdg9euXfvMg507507Url1bdq+XK1cOCxYsgLG8PvHaxM4eVd2
6dQtKNHXqVNSrV0/u/Ozu7o5u3brh/Pnzz/y+lFxK1SrC1tUX16tWxceNGKFVBXqP4nXz0HIrXqkSzZs1CjRo1sjY
ba9iwIf7991+jOX/5fX1qOn+5+fLLL+VrGjt27JP/LIIFPSS4udFpqm5D7rK2y3bNwOW/6s4

ix2x8u1CvHAIfTNUzLlVRvDFqc

mvWj7awRxk
```

```
vjzJkzKFeuXK7fx9LSEq1bt5Y3sRqrePHi2L59+2PLz4nIcFiZISK9mjRpkqyGiBVNQpkyZfDtt9/ivffey7GK6EnEtJ
JY+bRy5cocU0yCmF76448/cPbsWRw8eFB+XqxAyo0IKGJ10ieffCIrLRs2bMB3332X4zkTJkyQVR7R8CuCjnieWH2law
AWlaUff/xRfu7q1atyRZaYgqpYseJzHCEi0rsi7dAhIqO2c+dOjYWFhWyifVTbm01L7zwgiYjIyPXrxfNujY2Nhp7e3
tNbGxsjs8dO3ZMU7duXY2tra2mfPnympUrV2r8/Pw006dPf2IDsLB3717ZgCy+pmnTpvJrsjcAC4cOHdK0adNGU6xYMY
2Dg4OmRo0amilTpsjPidfRvHlzjYuLi8bOzk5+bvny5c99nIhIv8zEf/QfkYiIiKBqeZiIiISNUYZoiIiEjVGGaIiI
hI1RhmiIiISNUYZoiIiEjVGGaIiIhI1RhmiIiISNUYZoiIiEjVGGaIiIhI1RhmiIiISNUYZoiIiEjVGGaIiIgIavY/vV
GiuBhhzksAAAAASUVORK5CYII=",
     "text/plain": [
      "<Figure size 640x480 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   },
   {
    "data": {
     "image/png": "iVBORw0KGgoAAAANSUhEUgAAAkAAAAHHCAYAAABXx+fLAAAAOnRFWHRTb2Z0d2F
yZQBNYXRwbG90bGliIHZlcnNpb24zLjEwLjcsIGh0dHBzOi8vbWF0cGxvdGxpYi5vcmcvTLEjVAAAAAlwSFlzAAAPYQA
AD2EBqD+naQAAMiNJREFUeJzt3Qd8VFXe//FfiiSUJLTQBEPvTVAjCAISmixSBBFRihFcF1RkEQkqqEJBXeGyIDwiuK0R
FpEmRsriIFGkiIFIEFhBIWAlNSKiBJ5n/63f2P7MZUiCaZCY5n/frdZ/M3Ln3zpk7rPN9zvmde30cDodDAAAALOLr6QY
AAADkNQIQAACwDgEIAABYhwAEAACsQwACAADWIQABAADrEIAAAIB1CEAAAMA6BCAAAGAdAhCAPBcbGys+Pj6yfft2zr6
HVK5cWQYMGD5h7UIQEABCRNplzJlykibNm3kH//4rbJnLkv8vmPN9zvmde30cDodDAAAALOLr6QY
AAADkNQIQAACwDgEIAABYhwAEAACsQwACAADWIQABAADrEIAAAIB1CEAAAMA6BCAAAGAdAhCAPBcbGys+Pj6yfft2zr6
HVK5cWQYMGMD5h7UIQEABCRNplzJlykibNm3kH//4R563Z/HixdKpUycpXbq7Q6FCpUSCpUqCCPPfaYfPvtt+IpH3zwgTl
Pt+vm85l2+fOf/5yrbQWQN/zz6H0A5LLx48dLlSpVRG/vd+rUKfOD/HTFd+//vd8KxRS9vd/ONdsjT4rTxJcAAgnA2zt2Bh3fSdrFT8AlKTEiW/DrYxvdNzs+9rHDBe44S6ts
HD5dy5crJyZMnTShq27atbNq0SZo3by6eCEAAayLLT49GuxT7vp169fuvU0pTyNtYVcpAQJCn0BVuMpVngWsjZEo
wlS+++CJHAlBqaqpcv35dAgMwIAACqnjx4lK4cOF0oePtvLFgD30+BEbbK3oCSP1CdfJNaZbWWnWBeAMsmjV+tZcnvDMkTRgFl5+2I5TdNAsRwjNwHbyBAHuapM5ZuQSgc8ZLtRvRwm8AQBoRFWSvMLWRTx5GEWWpPrSVMZTvQIQXP4GUvpjsIp8kd4cLBKRhZcbLHZGapdwgZaa+BpkXBkAyd1Fq17gTEUyvl+V3CG7tIMl/xRE////LQr/cJJAIK0+bD4JvGntqR7I1r9HaTBktXlngu2xEGOv+MDoY2kaS+ZPCBBQ8Do0Ihx6/yvVQMnzzeBo8d6hzO7/aGltMFQHHPbPFd+XIAXM1J0QGi/bGsyjlaxZIlBX1dtXc/7ulVTk70x9g4uUhycyyuu+BWsGThYS8Nm9ZHYAvnbcAJbaXEEkMHbPQDYLEd8LgxK4WK0/h5Y8hgw9QijAGaC
MmzfJQZvJ5wBM/Vk6GxGt2ou5vpGCyLHY3AywzxFf8Ml0ghVAw5jEBkuIfbaEpN8GUI6gYLEXDT1KSRnoEW5P
Ol+dBg2UhpYIT9fBWmsf8gfx9G/3S4jVh5+1/CDOXlwRBA7rXrC0dpbDuKWmPRGYMb9u1vgtbfm3a5Q/H+Bs+fGO8GzzmeKadBcd9
UL53iXyXN8T1UY3xWvvE1m29NGSQmYfHWNQcy+Q5WP0j7G5brR8sjc4G4vLVp1btzxCYC9YtbJsDXzVeP0vYvKySUOn9cWJ6OhgOKg09/t9V2K18N3Y9VX4fbtcM
y+Q79Y4Ed8E2BuxWP4nD3M5XFvX8pd09QVLu4lDxkqRPGoVVujv2Dmn4pXd5yLH7jBmpKCLKj9/UagYjLBqf4W64lh59+H
/B/6YTrsOG9HxaW+4HlJ1D63KUZ4AWrZyyl8kDuHi2L8mhNDx+sAtHDT7wXtMHL3EF+cQsdR8c2yjbiJe1C6sn6YtYvNK
Ti1JXaqt+YNSLX9LwsdTfFBUctvmxHJ3jX/LmoNmpxKzmb8bjVpGRCBgZf0ByEE1w4/yK1I0ZUyetBF93xtILNTkl4ACVWsVVFDSGwTRlt8h49Lkna8jdzqNDCs8cHypb/k6wz
8LKpkHbuRB91ArjNAB3Ac7v2mj6Z8WCvIzRddXBw1TLXP4tjc94MJ+DkD6MtSQf13XiBAt9bXFsGgwJ6gsMwZWvvM7Xb
Rt9Xd6H+ptOsG2Jx/mz0d/AE2qqA1eLvKO8Hi6A/nhmR1ek5LAVmY2mfmQQ7VkTjMvYDpf6dYj7bLemKRd/1tMsRq77W
jr2UMx+jWR+7/E6zPUqFP0u6dmt1xkuVbDjhqXl/0YwtRdsG8n2dsiPhaR3+9nMQAb+JJyD9SHSu6nS3b8iWP/01jYZadQwuv1gmmCG0+WbotmgBM1dAeVzKDDphFgiR4pwiqpjvqDGMu/gzxmC9f9Es65gx+hhu0Anu
jmQEc/JZgBOdq4M3b4KaNX0X3PatsoFPlX7HDyxQ5yJHCbn8tvlqK5ErKjd/Rl43pZP1efTzQK/NP05GFPXPzgGNcEGS8l5VHn3ad6bfxJZNjyPGUrRHqw3PPqUBN/xIB27dsGA9oxeoN5hhuTcLRCbB27JrlWnxTZycnDCSC4ZBKqZ4N6oKiSWpzGoG3rwMuvzFSFRkqxS70dgrqWwnAYY3hrgBo/Ys5WtXf+i5gxq7fZ9ujt28bjwlUa1uDZkL3DBbRAM6ke8a
GOG5fVeahMyTGOwLEvtg8WqEkFQu9LQc9dkgjHzfwAQjuS9cC2Yt4mhNwXvdPzNtdhx0A1FmL3D3cwhXKDC
```

```
AAGAdLoSYAb1kvN7TR+9snZuX0AcAADlHr+yjN2XWi5Dq/RCzQgDKgIafm++8DAAA8of4+HipWLFiltsQgDKgPT/OE+i
8hQAAAPBuSUlJpgPD+TueFQJQBpzDXhp+CEAAAOQvt1O+QhE0AACwDgEIAABYhwAEAACsQwACAADWIQABAADrEIAAAIB
1CEAAAMA6BCAAAGAdAhAAALAOAQgAAFjHowEoJiZG7r33XnPPjjJlyki3bt3k4MGDbttcu3ZNhgwZIqVKlZJixYrJo48
+KqdOnbrl3WDHjBkj5cuXl8KFC0tERIQcOnQolz8NAADILzwagNavX2/CzdatW2X16tVy48YNad++vVy+fNm1zUsvST
Lli2TBQsWmO31Tu09evTI8rhvvvmmvP/++zJjxgz5/vvvpWjRotKhQwcTpgAAAHwc2l3iJc6cOWN6gjToPPjgg5KYmCi
hoaEyZ84c6dmzp9nmwIEDUqdOHdmyZYvcf//96Y6hH6dChQry17+96Y6hH6dChQry17/+VUaMGGHW6XHKli0rsbGx8vjjj9/W3WRDQkLMftw
MFQCA/CE7v99eVQOkDVYlS5Y0f3fs2GF6hXQIy6l27dpy1113mQCUkaNHj0pCQoLbPnoywsPDM90HAADYxV+8RGpqqgw
bNkweeOABqV+/vlmnQaZQoUJSvHhxt221N0fy4hzvW5zu/skJyebJW2CBAAABZfXBCCtBdq7d69s3LjRI8XYdHRef6
+AICCo/KoFZ5uQr5xbFJnTzfBO4bAhg4dKsuXL5e1a9dKxYoVXevLlSsn169flwsXLrhtr7PA9LWMONfSPFMsq32ioqL
M8JtziY+Pz4FPBQAAvJVHA5AWLGv4Wbx4sXz77bd77bdSpUoVt9ebNm0qd9xh6afXR
IS2eDZbZPQECAKZZKuwAAgILL19PDXrNnzzazvPRaQFqjo8vVq1ddxcuRkZEyfPhw0zukRdEDBw40QSbtDDAtjNYQpXx
8fEwt0RtvvCFfffWV7NmzR/r162dmhul1hgAAAAxaAzR9+nTzt3Xr1m7rZ82aJQHdDDCPJ0+eLL6+vuYCifqorNfz+eC
DD9y2114h5wwyNXLkSkSHMtocGDB5vhsxYtWsiqVaskMDAwTz4XAADwbl51HSBvwXWRQG054ug8+11gAAAAPICAQg
AAFiHAAQAAKxDAAIAANYhAAEAAOsQgAAAgHUIQAAAwDoEIAAAYB0CEAAAsA4BCAAAWIcABAAArEMAAgAA1iEAAQAA6xC
AAACAdQhAAADAOgQgAABgHQIQAACwDgEIAABYhwAEAACsQwACAADWIQABAADrEIAAAIB1CEAAAMA6BCAAAGAdAhAAALA
OAQgAAFiHAAQAAKxDAAIAANYhAAEAAOsQgAAAgHUIQAAAwDoEIAAAYB0CEAAAsI5HA9CGDRukS5cuUqFCBfHx8ZElS5a
4va7rMlreeuutTI85bty4dNvXrl07Dz4NAADILzwagC5fviyNGjWSadOmZfj6yZMn3ZaZM2eaQPPPo49medx69eq57bd
x48Zc+gQAACA/8vfkm3fq1MksmSlXrpzb66VLl0qbNm2kWtWqZR7X398/3b4AAAD57Yrlw1KlTsmLFComMjJFZ7ZzltocOHTL
DahqU+vbtK3FxcXnSRgAAkD94tAco0Z755BMJDAzMcrqzs7CwMImNjZVatWqZ4a/x48dLq1at5JdffjH7ZyQ5OdksjqSkp
Bx/CAAAAoGgBAAcgsU6PDQIs8QE+AAsKkb/aw7fKEfHiVn7TFWUD6cSaZ0zHDvTyb+OPeNpVPqdMMqE1hknQh8gkvNNTnHBH7k7cU6J2EDVf5L5ZTGf1Sm8PHKV6K5W+HZ+54ZsyDdIcpyCgxLNVd1LhpgYrHfi1Vh/rWZbTgUBjP6YRbxqCUp7WWz
5Vwv4bHAOByhTmhZ5d4HHZqxqjsK7sqYp9mR3R6S5F1EOhqbHYQO+dqARu5ppJqH8AqUCqwsDTSMghwlXQW8bxs+ZL2xkCl2ovx59yCLRD7F2QyRG4aKCm+jL5/rTO9xXIs9bHfhSmyDmyOCLpVDmTpI2tbZhnXrgwYl0nCD7wr3Pk9UG8HVmSgpJvw1Tb+b5ykQ4cFDqNjHkWO5t7Dsl1oAR6ej7Ao5Ryo8cxZ3ecS5SWNG2MAw9
```