```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 10,
   "id": "91f4a1b5",
   "metadata": {},
   "outputs": [],
   "source": [
    "df = pd.read_csv('STOCK PREDECTION.CSV')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 33,
   "id": "4014ce41",
   "metadata": {},
   "outputs": [],
   "source": [
    "%matplotlib inline\n",
    "\n",
    "import numpy as np\n",
    "import pandas as pd\n",
    "\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n",
    "\n",
    "import warnings\n",
    "warnings.simplefilter(\"ignore\")\n",
    "\n",
    "pd.set_option(\"display.max_columns\",None)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "id": "d2db2101",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 15000 entries, 0 to 14999\n",
      "Data columns (total 22 columns):\n",
      " #   Column                    Non-Null Count  Dtype  \n",
      "---  ------                    --------------  -----  \n",
      " 0   Patient ID                15000 non-null  int64  \n",
      " 1   Patient Name              15000 non-null  object \n",
      " 2   Age                       15000 non-null  int64  \n",
      " 3   Gender                    15000 non-null  object \n",
      " 4   Hypertension              15000 non-null  int64  \n",
      " 5   Heart Disease             15000 non-null  int64  \n",
      " 6   Marital Status            15000 non-null  object \n",
      " 7   Work Type                 15000 non-null  object \n",
      " 8   Residence Type            15000 non-null  object \n",
      " 9   Average Glucose Level     15000 non-null  float64\n",
      " 10  Body Mass Index (BMI)     15000 non-null  float64\n",
      " 11  Smoking Status            15000 non-null  object \n",
      " 12  Alcohol Intake            15000 non-null  object \n",
      " 13  Physical Activity         15000 non-null  object \n",
      " 14  Stroke History            15000 non-null  int64  \n",
      " 15  Family History of Stroke  15000 non-null  object \n",
      " 16  Dietary Habits            15000 non-null  object \n",
      " 17  Stress Levels             15000 non-null  float64\n",
      " 18  Blood Pressure Levels     15000 non-null  object \n",
      " 19  Cholesterol Levels        15000 non-null  object \n",
      " 20  Symptoms                  12500 non-null  object \n",
      " 21  Diagnosis                 15000 non-null  object \n",
      "dtypes: float64(3), int64(5), object(14)\n",
      "memory usage: 2.5+ MB\n"
     ]
    }
   ],
   "source": [
    "df.info()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 12,
   "id": "7b2841ed",
   "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>count</th>\n",
       "      <th>unique</th>\n",
       "      <th>top</th>\n",
       "      <th>freq</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>Patient Name</th>\n",
       "      <td>15000</td>\n",
       "      <td>13818</td>\n",
       "      <td>Ela Sarna</td>\n",
       "      <td>4</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Gender</th>\n",
       "      <td>15000</td>\n",
       "      <td>2</td>\n",
       "      <td>Male</td>\n",
       "      <td>7622</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Marital Status</th>\n",
       "      <td>15000</td>\n",
       "      <td>3</td>\n",
       "      <td>Single</td>\n",
       "      <td>5156</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Work Type</th>\n",
       "      <td>15000</td>\n",
       "      <td>4</td>\n",
        "      <td>Private</td>\n",
        "      <td>3863</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Residence Type</th>\n",
        "      <td>15000</td>\n",
        "      <td>2</td>\n",
        "      <td>Rural</td>\n",
        "      <td>7529</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Smoking Status</th>\n",
        "      <td>15000</td>\n",
        "      <td>3</td>\n",
        "      <td>Currently Smokes</td>\n",
        "      <td>5011</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Alcohol Intake</th>\n",
        "      <td>15000</td>\n",
        "      <td>4</td>\n",
        "      <td>Rarely</td>\n",
        "      <td>3821</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Physical Activity</th>\n",
        "      <td>15000</td>\n",
        "      <td>3</td>\n",
        "      <td>High</td>\n",
        "      <td>5060</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Family History of Stroke</th>\n",
        "      <td>15000</td>\n",
        "      <td>2</td>\n",
        "      <td>Yes</td>\n",
        "      <td>7592</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Dietary Habits</th>\n",
        "      <td>15000</td>\n",
        "      <td>7</td>\n",
        "      <td>Paleo</td>\n",
        "      <td>2192</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Blood Pressure Levels</th>\n",
        "      <td>15000</td>\n",
        "      <td>4458</td>\n",
        "      <td>96/70</td>\n",
        "      <td>14</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Cholesterol Levels</th>\n",
        "      <td>15000</td>\n",
        "      <td>5952</td>\n",
        "      <td>HDL: 50, LDL: 185</td>\n",
        "      <td>9</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Symptoms</th>\n",
        "      <td>12500</td>\n",
        "      <td>5786</td>\n",
        "      <td>Difficulty Speaking</td>\n",
        "      <td>268</td>\n",
        "    </tr>\n",
        "    <tr>\n",
        "      <th>Diagnosis</th>\n",
        "      <td>15000</td>\n",
       "      <td>2</td>\n",
       "      <td>No Stroke</td>\n",
       "      <td>7532</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "                          count unique                  top  freq\n",
       "Patient Name              15000  13818            Ela Sarna     4\n",
       "Gender                    15000      2                 Male  7622\n",
       "Marital Status            15000      3               Single  5156\n",
       "Work Type                 15000      4              Private  3863\n",
       "Residence Type            15000      2                Rural  7529\n",
       "Smoking Status            15000      3     Currently Smokes  5011\n",
       "Alcohol Intake            15000      4               Rarely  3821\n",
       "Physical Activity         15000      3                 High  5060\n",
       "Family History of Stroke  15000      2                  Yes  7592\n",
       "Dietary Habits            15000      7                Paleo  2192\n",
       "Blood Pressure Levels     15000   4458                96/70    14\n",
       "Cholesterol Levels        15000   5952    HDL: 50, LDL: 185     9\n",
       "Symptoms                  12500   5786  Difficulty Speaking   268\n",
       "Diagnosis                 15000      2            No Stroke  7532"
      ]
     },
     "execution_count": 12,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.describe(include='O').T"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "id": "89e2ca57",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<style type=\"text/css\">\n",
       "#T_f5598_row0_col0, #T_f5598_row1_col0, #T_f5598_row1_col1, #T_f5598_row1_col2,
#T_f5598_row1_col4, #T_f5598_row1_col5, #T_f5598_row1_col6, #T_f5598_row1_col7,
#T_f5598_row2_col0, #T_f5598_row2_col1, #T_f5598_row2_col2, #T_f5598_row2_col3,
#T_f5598_row2_col4, #T_f5598_row2_col5, #T_f5598_row2_col6, #T_f5598_row2_col7,
#T_f5598_row3_col0, #T_f5598_row3_col1, #T_f5598_row3_col2, #T_f5598_row3_col3,
#T_f5598_row3_col4, #T_f5598_row3_col5, #T_f5598_row3_col6, #T_f5598_row3_col7,
#T_f5598_row4_col0, #T_f5598_row4_col1, #T_f5598_row4_col2, #T_f5598_row4_col4,
#T_f5598_row4_col5, #T_f5598_row4_col6, #T_f5598_row4_col7, #T_f5598_row5_col0,
#T_f5598_row5_col1, #T_f5598_row5_col2, #T_f5598_row5_col4, #T_f5598_row5_col5,
#T_f5598_row5_col6, #T_f5598_row5_col7, #T_f5598_row6_col0, #T_f5598_row6_col1,
#T_f5598_row6_col2, #T_f5598_row6_col3, #T_f5598_row6_col4, #T_f5598_row6_col5,
#T_f5598_row6_col6, #T_f5598_row6_col7, #T_f5598_row7_col0, #T_f5598_row7_col1,
#T_f5598_row7_col2, #T_f5598_row7_col3, #T_f5598_row7_col4, #T_f5598_row7_col5,
#T_f5598_row7_col6, #T_f5598_row7_col7 {\n",
       "  background-color: #bde7db;\n",
       "  color: #000000;\n",
       "}\n",
       "#T_f5598_row0_col1, #T_f5598_row0_col2, #T_f5598_row0_col4, #T_f5598_row0_col5,
#T_f5598_row0_col6, #T_f5598_row0_col7, #T_f5598_row4_col3 {\n",
       "  background-color: #ffd4ac;\n",
       "  color: #000000;\n",
       "}\n",
       "#T_f5598_row0_col3 {\n",
       "  background-color: #b2dfd8;\n",
       "  color: #000000;\n",
       "}\n",
       "#T_f5598_row1_col3 {\n",
       "  background-color: #4a4fa5;\n",
       "  color: #f1f1f1;\n",
       "}\n",
       "#T_f5598_row5_col3 {\n",
       "  background-color: #4167c7;\n",
       "  color: #f1f1f1;\n",
       "}\n",
       "</style>\n",
       "<table id=\"T_f5598\">\n",
       "  <thead>\n",
       "    <tr>\n",
       "      <th class=\"blank level0\" >&nbsp;</th>\n",
       "      <th id=\"T_f5598_level0_col0\" class=\"col_heading level0 col0\" >count</th>\n",
       "      <th id=\"T_f5598_level0_col1\" class=\"col_heading level0 col1\" >mean</th>\n",
       "      <th id=\"T_f5598_level0_col2\" class=\"col_heading level0 col2\" >std</th>\n",
       "      <th id=\"T_f5598_level0_col3\" class=\"col_heading level0 col3\" >min</th>\n",
       "      <th id=\"T_f5598_level0_col4\" class=\"col_heading level0 col4\" >25%</th>\n",
       "      <th id=\"T_f5598_level0_col5\" class=\"col_heading level0 col5\" >50%</th>\n",
       "      <th id=\"T_f5598_level0_col6\" class=\"col_heading level0 col6\" >75%</th>\n",
       "      <th id=\"T_f5598_level0_col7\" class=\"col_heading level0 col7\" >max</th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row0\" class=\"row_heading level0 row0\" >Patient 
ID</th>\n",
       "      <td id=\"T_f5598_row0_col0\" class=\"data row0 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row0_col1\" class=\"data row0 col1\" >49716.000000</td>\n",
       "      <td id=\"T_f5598_row0_col2\" class=\"data row0 col2\" >29001.000000</td>\n",
       "      <td id=\"T_f5598_row0_col3\" class=\"data row0 col3\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row0_col4\" class=\"data row0 col4\" >24562.000000</td>\n",
       "      <td id=\"T_f5598_row0_col5\" class=\"data row0 col5\" >49448.000000</td>\n",
       "      <td id=\"T_f5598_row0_col6\" class=\"data row0 col6\" >75112.000000</td>\n",
       "      <td id=\"T_f5598_row0_col7\" class=\"data row0 col7\" >99975.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row1\" class=\"row_heading level0 row1\" >Age</th>\n",
       "      <td id=\"T_f5598_row1_col0\" class=\"data row1 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row1_col1\" class=\"data row1 col1\" >54.000000</td>\n",
       "      <td id=\"T_f5598_row1_col2\" class=\"data row1 col2\" >21.000000</td>\n",
       "      <td id=\"T_f5598_row1_col3\" class=\"data row1 col3\" >18.000000</td>\n",
       "      <td id=\"T_f5598_row1_col4\" class=\"data row1 col4\" >36.000000</td>\n",
       "      <td id=\"T_f5598_row1_col5\" class=\"data row1 col5\" >54.000000</td>\n",
       "      <td id=\"T_f5598_row1_col6\" class=\"data row1 col6\" >72.000000</td>\n",
       "      <td id=\"T_f5598_row1_col7\" class=\"data row1 col7\" >90.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row2\" class=\"row_heading level0 row2\" 
>Hypertension</th>\n",
       "      <td id=\"T_f5598_row2_col0\" class=\"data row2 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row2_col1\" class=\"data row2 col1\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col2\" class=\"data row2 col2\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col3\" class=\"data row2 col3\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col4\" class=\"data row2 col4\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col5\" class=\"data row2 col5\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col6\" class=\"data row2 col6\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row2_col7\" class=\"data row2 col7\" >1.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row3\" class=\"row_heading level0 row3\" >Heart 
Disease</th>\n",
       "      <td id=\"T_f5598_row3_col0\" class=\"data row3 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row3_col1\" class=\"data row3 col1\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row3_col2\" class=\"data row3 col2\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row3_col3\" class=\"data row3 col3\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row3_col4\" class=\"data row3 col4\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row3_col5\" class=\"data row3 col5\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row3_col6\" class=\"data row3 col6\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row3_col7\" class=\"data row3 col7\" >1.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row4\" class=\"row_heading level0 row4\" >Average Glucose
Level</th>\n",
       "      <td id=\"T_f5598_row4_col0\" class=\"data row4 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row4_col1\" class=\"data row4 col1\" >129.000000</td>\n",
       "      <td id=\"T_f5598_row4_col2\" class=\"data row4 col2\" >40.000000</td>\n",
       "      <td id=\"T_f5598_row4_col3\" class=\"data row4 col3\" >60.000000</td>\n",
       "      <td id=\"T_f5598_row4_col4\" class=\"data row4 col4\" >95.000000</td>\n",
       "      <td id=\"T_f5598_row4_col5\" class=\"data row4 col5\" >129.000000</td>\n",
       "      <td id=\"T_f5598_row4_col6\" class=\"data row4 col6\" >165.000000</td>\n",
       "      <td id=\"T_f5598_row4_col7\" class=\"data row4 col7\" >200.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row5\" class=\"row_heading level0 row5\" >Body Mass Index
(BMI)</th>\n",
       "      <td id=\"T_f5598_row5_col0\" class=\"data row5 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row5_col1\" class=\"data row5 col1\" >27.000000</td>\n",
       "      <td id=\"T_f5598_row5_col2\" class=\"data row5 col2\" >7.000000</td>\n",
       "      <td id=\"T_f5598_row5_col3\" class=\"data row5 col3\" >15.000000</td>\n",
       "      <td id=\"T_f5598_row5_col4\" class=\"data row5 col4\" >21.000000</td>\n",
       "      <td id=\"T_f5598_row5_col5\" class=\"data row5 col5\" >27.000000</td>\n",
       "      <td id=\"T_f5598_row5_col6\" class=\"data row5 col6\" >34.000000</td>\n",
       "      <td id=\"T_f5598_row5_col7\" class=\"data row5 col7\" >40.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row6\" class=\"row_heading level0 row6\" >Stroke
History</th>\n",
       "      <td id=\"T_f5598_row6_col0\" class=\"data row6 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row6_col1\" class=\"data row6 col1\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row6_col2\" class=\"data row6 col2\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row6_col3\" class=\"data row6 col3\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row6_col4\" class=\"data row6 col4\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row6_col5\" class=\"data row6 col5\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row6_col6\" class=\"data row6 col6\" >1.000000</td>\n",
       "      <td id=\"T_f5598_row6_col7\" class=\"data row6 col7\" >1.000000</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th id=\"T_f5598_level0_row7\" class=\"row_heading level0 row7\" >Stress
Levels</th>\n",
       "      <td id=\"T_f5598_row7_col0\" class=\"data row7 col0\" >15000.000000</td>\n",
       "      <td id=\"T_f5598_row7_col1\" class=\"data row7 col1\" >5.000000</td>\n",
       "      <td id=\"T_f5598_row7_col2\" class=\"data row7 col2\" >3.000000</td>\n",
       "      <td id=\"T_f5598_row7_col3\" class=\"data row7 col3\" >0.000000</td>\n",
       "      <td id=\"T_f5598_row7_col4\" class=\"data row7 col4\" >3.000000</td>\n",
       "      <td id=\"T_f5598_row7_col5\" class=\"data row7 col5\" >5.000000</td>\n",
       "      <td id=\"T_f5598_row7_col6\" class=\"data row7 col6\" >8.000000</td>\n",
       "      <td id=\"T_f5598_row7_col7\" class=\"data row7 col7\" >10.000000</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n"
      ],
      "text/plain": [
       "<pandas.io.formats.style.Styler at 0x29b18021450>"
      ]
     },
     "execution_count": 13,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df.describe().round().T.style.background_gradient(cmap='icefire')"
   ]
  },
  {
    "cell_type": "code",
    "execution_count": 16,
    "id": "c4862eb8",
    "metadata": {},
    "outputs": [
     {
      "data": {
       "image/png": 
"iVBORw0KGgoAAAANSUhEUgAAA1kAAANiCAYAAACTvANdAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjcuM
iwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8pXeV/AAAACXBIWXMAAA9hAAAPYQGoP6dpAACX4ElEQVR4nOzdeVhU5f//8feIgIp
KogKSuBu55o6IuQtWZGqlpuGSW+GSW4uaZpZa7qWlZovmkmZmmZqhVprhXrimppk7LokggoDw/v3Bb87XETTrc+OIPB/X1WWcu
efMfeY+M3Ne577PfWyqqgIAAAAAMCKPsysAAAAAAPcSQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAA
AAAAwKK+zK3C3S09Pl9OnT0uhQoXEZrM5uzoAAAAAnERV5fLly+Ln5yd58tgLLy+Ln5yd58ty8v4qQ9Q9Onz4t/v7+zq4GAAAAgLvEiRMnpGTJk
jd9nJD1DwoVKiQiGW9k4cJFnVybfy81NVUiIyMlJCREXF1dnV2dXIk2cD7awPloxMfHi7+/v5URboa
Q9Q/sQwQLFy6cY0NWgQIFpHDhwjlyR74R74X0AbORxs4H23gfLSB89EGzkcbON+90gb/dBkRE18AAAAAgEGELAAAAAAwiJAFAAAAAA
AYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgA
AAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDC
FkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAAYRsgAAAADAoH8dsjZu3CiPP/64+Pn5ic1mk6+//rhcVWV0aN
Hi5+fn+TPn1+aNGki+/btcyiTnJws/fv3l2LFiomHh4e0bt1aTp486VAmNjZWwsPDxdPTUzw9PSU8PFwuXbrkUOb48ePy+OOPi
4eHhxQrVkwGDBggKSkpDmX27Nkjjj3lvz588e9t5sNAAAALflX4esK1euyEMPPSQzZszI8vEJEyb
MGbJ9+3bx9fWVli1byuXL60yAwcOlOXl8vixYtl06yAwcOlXL8vixYtl06ZNkpCQIGFhYhYZKWlmaV6dSpk0RHR8uaNWtkzZo1Eh0dLeHh4dbjaWlp8
thjj8mVK1dk06ZNsnjxYlm2bJkMGTLeHh4dbjaWlp8
thjj8mVK1dk06ZNsnjxYlm2bJkMGTLEKhMfHy8tW7YUPz8/2b59u0yfPl0mTZokU6ZM+bebDQAAAAC3Je+/fcIjjzjzwijzzySJa
PqapMmzZNRowYIe3atRMRkXnz5omPj48tWrRI+vTpI3FxcfL+xx8sWrRI+/W8/+/ePl+ZJvH/fWl+SWreWWYITM3v5pvwWUlIRcW0WlZ3Q
mWLBAYGiojInDlzJCgoSA4ePCBAQESGRkp+/fvlxMnToifn5+Ii++Mn0AePfm6desmY8eOFC8vChQvl6tWrMnfuXHF3d5eqVatv
KoUOHZMqUKTJ48GCx2Wz/6U0DAAAAgJv51yHrVo4ePSoxMTESEhJiLXN3d5fGjRvLpp9ofKOLCSPgCpqaKOZfz8/KRq1aoSF
RUloaGhsnnzZvH09LQLClohI/fr1xdPTU6KioiIgIEA2b94sVatWtQKWiEhoaKgkJyfLzp07pWNaRdCMi0R4tb2Vh+BQvp4IzYuPnUOZ7h1J8emmIK9aXVRd3bScu9+hL8kN1q1C
Y/PXXX1K2bNlM25CcnCzCzJycnW3/Hx8SIikpqaKmpqWbeqP9BYmKiHDx48LbJyQl
0XcwL7f3A37T25FGzgfbeB8tIHz0QbORxs4X05vg9utt9Q0bORxs4X05vg9utt9GQFRMTIyIiPj4+DiU8fHxkqNHj1ph3N
m9Xt7ezuUufFvIpMKL+8pC7uUJmQbN5OdFC3d46C4nzC6hkRypkWGA0nOhIBj4bT5iQjIiD3qHB5ZTF2pTvNeKrxmJtCFxe+fU0KM6CnTrcx1P2j2WI0mD3NBlYMHIUvZUGAGLn9Y+gV7xjvUq6sTaQnrcXLG7D7ap/4x6J5v57MJiQpydlvcCUCAYpm7ZsQ7xRaACh3DCyVCSivDEvGS4u3ETF0xkYaV
9lj1BwGXz8kbiJ6rgNBT6qKHdgFxQTVxf+40x7GAkJe1k7jsuRGX3tMxSymF4O2egtbU2QrUN6+PVNB0dLVCs1Rt4sO9SSqIqBMFsYCZZDY+W9zE/WoxvGbEoy2qw7GwLXC7ovNCM5M8SEqx+ZrgzXs6nmeXr8Sm+P9RNRkRs4TkgSn+sbdwlYBIhiXKYeHJ/a9UdcfqEXs+qe
AAAAAgEGELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAh
wAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAAGAQIQsAAAAADCJkAQAAAIB
BhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAA
AAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAAYRsgAAAADAIEI
WAAAAABhEyAIAAAAAgwhZAAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAA
IMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhkPWdeuXZPXXntNypYtK/nz55dy5crmJmDFjJD093SqjqjJ
69Gjx8/OT/PnzS5MmTWTfvn0O060lOTpb+/ftLsWLFxMPPDQ1q3bi0nT550KBMbGyvh4eHi6ekpnp6eEh4eLpcuXTE6ekpnp6eEh4eLpcuXXIoc/z4cXn88
cfFw8NDihUUrJgMGDJCUlBRqd96BWbNmyYWzMM+T3yCXCRMmyMSJE2X69lWmQkTJsiUKVNkxowZ37dvH
19ZWWLVvK5cuXrTIDr2a2TNmjPUkG1s2TNmjUSHR0t4eHh4uLiIiNyYTBHTZ4ZuNEpaWny2GOPyZUrV2TTzp
2TTpk2yePFiWiWbZsmQTQwZMsQtIGGtC+rUTEZF58+aJu7u7VGFGPv/ONz8Y4eIZPRiTZs2TUaMGCHt2rUTEZ
58+aJj4+PLFq0SPr06SNxcXHy7rvvyvz586VFixYiIrJgwQLx9/eXdevWSWhoqHz+VUe33Yy3XHuq6ddy+Q6ivffKbT2pVBs1ooHDHA
jwoAQEBEhkZKfv37ZcWKFeLj42M9rVa/X8Ilc2m5//4X2JeGWEiNEV1mcAu+fmZfiC
NmyYiIkePGPHp9WmBgJCQmxnuGuMfm6ngZ0tNSa3TKgpmnu6ZN0qFZUkrXYWEzBvGmlRJ9S7TdrXpPVv36xkgvtsaf3b4
0RFRUlAQIBs3rxZqatmrRxbBOJBIfEDMm0xqrNJ8oq0SmwFHB5mDB2enp8Nb2rDfeuUQX7Ndv7JFjaTZRdXDIhbmatqsR3V6ki9gRPqrKYeE8uOPlkNnSgU3Ke7mzWU5y4J8DnUOqUBTYoepKb2L9zvrjxcqpUKtd+XsfrJx42CyNxUNTEf8fi
d9995aAPnow2cjzZwPtrA+WgD58vpbXC79TYXR4Ji4uTBx98UFxcXCQtUS3MTEiIiIj4+GzkeKqJOjsDO3zcsskXmFrbKDkzbUVHPPL5JwHpyf+NzG0Df3hTuB
LFjVhk3NzcpUqRIjL258fEm1/f29nOyoc+PrFClSRNzc3BKO5SP0LsJhoMOqyvJuPtYaYFp4ZXnqDWjR8/xb8RtkW+Pw
k1F5n1yZ3W7t2rbOrkOvRBs5HGzgfbeB8tIHz0QbOl1Pb4FOLeOkBr3uqoGDB0NRiv8/jPHNIl9/aAz9t2FyukzNRsCxQ+ZJlRqnR5fmd9oD5bzK8VV
c5mszk8T1UzLbvRjWWyKv9fylxv2LBhMnjwYOv+Ph48vEhXhC3aF4kuk6fuf1nGX5NZfX3/ZjVJnUmvowqtqaL94G
rq6uzq5Mr0QbORxs4H23gfLSB89EGzpfT28A+yr66K63DMlwU2vT29uSo4vK3HK/mbLIkVC2LUuH89y9vd3bt6d3AjhFA
oT1vHPnzlm9Tr6+vpKSkiGSyq+VcoSKKsDbOxVPNnzp2TBg0aWGXjn6O6pK36Hw7MmFKYj4vagyKr9ADZwz6Ayr6kmVD0XMTQhWSHdZeUH2ngQ4Sk+Z7wO3kWZUy0zXEV2CgndMfOltbrBkKnnr8KwTuaV
j5M2b1/o3J9b/XpJT96F7CW3gfLSB89EGzkcbOB9t4Hw5tQ1ut87GZxdMTEyUPHmF4mJN4V62bFnx9fWVtRKwCWtRmDdDLSxE3lnxY4ziSCXqRMMnjqM0KdnM1GZ04RSZsqKeLpCJ3gkZ8DHfkn7M8NdyjFBPZwLF1
```

```
AGqdu3a4urq6lDmzJkzsnfvXqtMUFCQxMXFybZt26wyW7dulbi4OIcye/fulTNnzlhlIiMjxd3dXWrXrm14ywEAAAAgG3qyHn/
8cRk7dqyUKlVKqlSpIr/99ptMmTJFnnvuORHJGL43cOBAGTdunFSsWFEqVqwo48aNkwIFCkinTp1ERMTT01N69OghQ4YMkaJFi
4qXl5cMHTpUqlWrZs02WKlSJWnVqpX06tVLZs+eLSIivXv3lrCwMAkICBARkZCQEKlcubKEh4fLxIkT5eLFizJ06FDp1atXjhz
6BwAAAODuZzxkTZ8+XUaOHCkRERFy7tw58fPzkz59+sioUaOsMi+//LIkJSVJRESExMbGSmBgoERGRkqhQoWsMlOnTpW8efNK+
/btJSkpSZo3by5z584VFxcXq8zChQtlwIAB1iyErVu3lhkzZlwhkzZliPu7i4yKpVqyQiIkKCg4Mlf/780qlTJ5k0aZLpzQYAAAAAEcm
GkFWoUCGZNm2aNWV7Vmw2m4wePVpGjx590zL58uWT6dOnO9zE+EZeXl6yYMGCW9anVKlSsnLlyn+qNgAAAAAYYfyaLAAAAADIz
QhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAA
AAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGEL
AAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGBQXmdXACJHL1yRK8nXsmXdR85fsf7Nmzf7mtvDPa+
ULeaRbesHAAAAcgpClpMdvXBFmk76KdtfZ8iXe7L9NX4c2oSgBQAAgFyPkOVk9h6saR1qSAXvgubXn5QsK3/aLGFNgsQjv7vx9
YuIHD6XIAOXRGdbbxwAAACQkxCy7hIVvAtK1fs9ja83NTVVYoqL1CpdRFxdXY2vHwAAAIAjJr4AAAAAAIMIWQAAAABgECELAAA
AAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGEL
AAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAA
GEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAA
AAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwh
ZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAA
AwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAA
AAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGE
bIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAA
AGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZA
AAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAz
KlpB16tQpefbZZ6Vo0aJSoEABqVGjhuzcudN6XFVl9OjR4ufnJ/nJ/nz55cmTZrIvn37HNaRnJws/fv3l2LFiomHh4e0bt1aTp486
VAmNjZWwsPDxdPTUzw9PSU8PFwuXbrkUOb48ePy+OOi4eHhxQrVkwGDBggKSkp2bHZAAAAAGA+ZMXGxkpwcLC4urrKd999J/v
375fJkyfLfffdZ5WZWZMGGCTJkyRWbMmCHbt28XX19fadmypcMbt28XX19fadmypVy+fNkqM3DgQFm+fLksXrxYNm3aJAkJCRIWFiZpaWlWmU6dOkl0d
LSsWbNG1qxZI9HR0RIeHm49npaWJo899phcuXJFNm3aJIsXL5Zly5bJkCFDTG82AAAAAIiISF7TK3znnnXfE399fPv30U2tZmJ
lrP9XVZk2bZqMGDFC2rVrJyIi8+bNEx8fH1m0m0aJH06dNH4uLi50OPP5b58+dLixYtRERkwYIF4u/vL+vWrZPQ0FD5/fffZc2aN
bJlyxYJDAwUEZE5c+ZIUFCQHDx4cKIDAwUEZE5c+ZIUFCQHDx4UAICAiQyMlL2798vJ06cED8/PxERmTx5snsnTr1k3Gjh0rhQsXNr35AAAAHI54yFrxYoVEho
aKk8//bRs2LBB7r//pomIiJBevXqJiMjRRo0clJiZGQkJCrOe4u7tL48aNZfPmzdKnTx/ZuXOnpKamOpTx8/OTqlWryubNmyU0NFSiE
Nm8ebN4enpaUtEpH79+uLp6UCSlRUVESEBAgmzdvtrt9uhmuXz38tKBqg9dNM9U9OTpbk5GTr7/j4eBERSU1NldTS
UVHNv1P937do169/sWL99ndmxbrvs3oac7k60AW6NNnA+2sD5aAPnow2cjzZwvpzbBeh68Mh688//fjw4bJt2
zYZMGCAuLu5cuXSQmJkZERHx8fBye4+PjI8eOHRMRkZiYGHFzc5MiRYpkKnP9+TExMeLt7Z3ptb29vR3K3Pg6RYoUETc3N6vMjcaPHy9jxoz
JtTxSjzvNUVG4o0ZNvwzHPumVP8NG3Mmu2yuKuNDjmLYJsxdG39+kuXqazvunXkDfkeVtNJWOGc0ibHyQDtRd3K3WQmKDlr1O/dTbhPXJnOYX3Bte+hZMrKXa8G2QCxNo+PQqgs9WKhBQEAJ5F39
fQkJCsmV44b7T8TJpzxZp2LChVPEzv/7U1FRrjfTmMZPqbSHnQkLm9tLi5qaqK/WMP+hMgz/t5HHRMg5KPtd5VeSJyAC8k+IoYQ+3/nMtGXrWf8fFYhVXB/zMeIGfQ7MZDHoimiaHXTkzSGXGmGl/z8P7xCb7Xc8SPNAyAEt9gzn0X85zcsbvm3MOq+BCR/
2dZyNMJt+ppEv9tlw+xR6b3/9LQr1Nn2QCZnyM3UG/+0zOTZmf7KiRrdg5B2bjT/b2bWkCDEtaO9SjHoYhxDJEoMZEsiacH8FG8BgVn6IHN4t3C6o1mXr/wwNNsHI4W/qSbKoSDaZFoyLjHMT4ZY7lL7/
yt4zaKzfaqAkfTZmt4OPgM9p9e8dH2YyU/mtrz/xT4b/
```

```
Jqvx/KXO9YcOGyeDBg62/4+Pjxd/fX0JCQqRw4cK3rN9/se90vEzas0UaNmwoVfzMrz81NVXWrl0rLVu2FFdXV+PrF8n+bcjp7
kQb4NZoA+ejDZyPNnA+2sD5aAPny+ltcLuXHBkPWTt37pRz585J7dq1rWVpaWmyceNGmTFjhnW9VExMjJJQUcIqc+7cOavf
XV1JSUiQ2NtahN+vcuXPSoEEDq8zZs2czvf758+cd1rN161aHx2NjYyU1NTVTD5edu7Z7Z1ru6uqaLTtC3rx5rX+zc0fLr
vqL3LltyOmysw1we2gD56MNnI82cD7awPloA+fLqW1wu3U2PvFF8+bNZc+ePRIdHW39V6V6GLMCU
lRTZs2GAFqNq1a4urq6tDmTNnzsjevXutMkFBQRIXFyfbtm2zymzdulVxcXi4uIcyuzdu1fOnDljjlYmMjBR3d3eHEAgAAAAphjvy
SpUqJBUrVrVYZmHh4cULVrUWj5w4EAZN26cVKxYUSpWrCjjxo2TAgUKSKdOnURExNPTU3r06CFDhgyRokWLipeXlwwdOlSqVas
mLVq0EBGRSpUqSatWraRXr14ye/ZsERHp3bu3hIWFSUBAgIiIhISESOXKlSU8PFwmTpwoFy9elKFDh0qvXr2yZegfAAAAAGTb7
IK38vLLL0tSUpJERERIbGysBAYGSmRkpBQqVMgqM3XqVMmbN6+0b99ekpKSpHnz5jJ37lxxcXGxyixcuFAGDBhgzULYunVrmTF
jhvW4i4uLrFq1SiIiIiQ4OFjy588vnTp1kkmTJt25jQUAAACQq9yRkPXTTz85/G2z2eTWT06NEyevTomz4nX758Mn36dJk+ffpNy
3h5ecmCBQtu+dqlSpWSlStX/pvqAgAAAMB/lm03IwYAAACA3IiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAAB
gECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBA
AAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIi
QBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAA
MAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAI
AAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgE
CELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAA
AgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQB
QAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMA
gQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAA
AAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYlNfZFcjtklLTRERk76m4bFn
/laRk2XFexPdYrHjkd8W1zh8LiFb1gsAAADkRIQsJzvy/wPy/PxvzZXvVn8HBndwIAAAA4KnaykCq+IiJS3rug5
Hd1Mb7+g2fiZMiXe2TyU9UkoISn8fXbebjnlbLFPLJt/QAAAEBOQchyMi8PN+lYr1S2rf/atWuIV7QhYAAACADEx
8AQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAA
MAgQhYAAAAAGETIAgAAAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAI
AAAAAgwhZAAAAAGAQIQsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGGQ9Z48ePl7p160qhQoQxE29tb2rRpIwcPHnQoo6oye
vRo8fPzk/z580uTJk1k3759DmWSk5O1f//+UqxYMfHw8JDWrVvLyZMnHcrExsbGZeHi4eHp6iqenp4SHh8ulS5ccyhw/flwef/x
x8fDwkGLFismAAQMkJSXF9GYDAAAgIhkQ8jasGGD9O3bV7Zs2SJr166166Va9euSUhIiFy5csucsUqM2HCBJkyZYrMmDFDFDtm/fLr6+v
tKyZUu5fPmyVbWxVbgwIGyfPlyWbx4sWzatq9aLi8XQjqjUUkWq6LLVbSMfBRGtbKAJHD9eTL3KnmLTk8JaXMScPbcnzMDWzNbRjmEEQbD5Mbx6DsUEhr1wMtLu0j1E1AGeLaXXrm+xeWt2fVpjbmtNyxlwCbTxAjVgvsqy5ClXOfqOEYhdL8PudLJ53z1mpAb2XglQwaMtsyJqwSltnUo/ymSyxntNOq7uqq1tq3Al5t5cGYqONV9dSh3WCg03aUVTvlXPFBW8kKn0bEhCLL8ItLwwjNoTzXDUNnE/Q1zgoTHA3xMLm64r3ioCjmeSXAe9zW4MfQ1eL8X+uw3/cVgIWAMfdGqVS9Pvd8KWqIqHwHS6knAoWFzSyQzl/YuqCMcN8lpLu5nbMyq26cDWBP8qvPaUcVGQvy0oW3ufTIRNeXmSz3QgHQFhsHVD6c6+5mGrtsPFz6JkcMWLUtMG+9sY+jQpfeYnl0EoKPaQSrVX9DtNLqR8xhX9HDa+9T1V+mO5QGiw3rtOGsEOpDgsHMwfh6iaZYg7gZt2WiCxpZBKxZgLHVbNQkyKk/DbEaFRbDm6t9TpfRVfY7KD/gc9wgMLzwl/wuXfj+Aj3M4hE1PRN48m0oPK1GYQjCM64MAXBq
kXSp08fiYuLk48//ljmz58vLVq0EBGRBQsWiL+/v6xbt05CQ0Pl999/lzVr1siWLVkMDBQRETmzJkjjQUFBcvDgQQkICJDIyEj
Zv3+/nDhxQvz8/EREZPLkydKtWzcZO3asFC5c2PTmAwAAAMjljIesG8XFxYmIiJeXl4iIHD16VM6ePSsNDtEEakWKvFvjpaxgGuS3s8M6kgpYgq6vgIzlI6Z3kNowMMwcq
Zv3+/nDhxQvz8/EREZPLkydKtWzcZO3asFC5c2PTmAwAAAMjljIesG8XFxYmIiJeXl4iIHD16VGJiYiQkJMQq4+7uLo0bN5aoq
Cjp06eP7Ny5U1JTUx3K+Pn5SdWqVSUqKkpCQ0Nl8+bN4unpaQUsEZH69euLp6enREVFSUBAgGzevFmqVq1qBSwRkdDQUElOTpa
dO3dK06ZNM9U3OTlZkpOTrb/j4+NFRCQ1NVWpVS0pNmhDzvXJT7/xXa7X2wsLJc2iJt/f/+R7fFBMdTnbGA+55jptmVBNafgQq7ZhXvG5K+IqjCVr7X3XoB0QrJZ9PkO9zXq/4G4t4/
bmWa9YnmH0JAnW4vUT57w/Pk6wX3ZzGZ12wZtN1Xj0uOMA39Ef9Y7GoNVgVfMofYCkLk9c0zjswvD+QrcFTtaY+i2L46jUh2nSr46oBs1SgHnqFIc0Zph7+a8Ezw8Q+XmWHJ0Xf3CWNcjkb5T0bm2jmMH4IJSeWQGCZVnCtM0ARBu9
MLq/5S+WV8iDNCl6k5QkNDzvW/ydPfK0SiZiQgdZtqOFdgfdE+lp4AEcjLZN+GtBFMu67fbTQIsjJZcSE1s/MLjmHJw4lKwXntz+JZvmpJPwEZXmJH4UmBAZh1bBcR4K1t/zlj4Yp2cfW6s9XDKR
Ku3+wH12fWkLzTNuQTwLQOaUzuOZ+Dou/x6GUqk3OSVbukymLQJuY+ITmN5F8hfm3rRVTN4eqQ+u0hKvNvWvunQvXXO3Z6RYQaRHxRXCYXtYG8GvqZWj+bxXLiLoSJLdqsLjN7uk/3Ge7y5EB8Tt2qZhNxHXbn9bbm3jI+
YqW3vHNt8I2jfXzYvYn4j76dHNXWLw0v2O+RBgdKuPbjhjO1GWH2gzSJLXHh/ZE6WVbRvv5a/fi3Yi2C8EoTgdF7z//jPX8zYrFh/fhPU/hj0hZlqmX2Z7dZt/cGKrJlqC1OKsUl4lfZHOm7KvsgfVpwd/l8qR8ijvG+lNo8BsmP
```

```
hdlExMT5eBjT912+f/6e5BbfgtE+Bw42799/0VoA9Oy+zMgkvPbID4+/rbK3dM9WadPn5b7779foqKiJCgoyFo+duxYmT9//m1
dVxMfHy+enp7/mFbvVqmpqda1KK6urs6uTq5EG9x5t3tNlgjTud8pfA6cjzZwPtrA+WgD58vpbXC72eCe7skqVqyYuLi4ZOq1O
nfuXKbeLQD3DlV1CFo3BqvrywEAAJh2T0/h7ubmJrVr15a1a9c6LF+7dq00aNDASbUCcCf8U4AiYAEAgOxyT/dkiYgMHjxYwsP
DpU6dOhIUFCQffvihD9+XJ5//nlnVw1ANruxR+v65QAAANnlng9ZHTp0kL///lvGjBkjZ86ckapVq8rq1auldOnSzq4agDtAV
XP8+G8AAJCz3PMhS0QkIiJCIiInF0NAAAAALnAPX1NFgAAAADcaYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhYAAAAAGETIAgA
AAACDCFkAAAAAYBAhCwAAAAAMImQBAAAAgEGELAAAAAAwiJAFAAAAAAYRsgAAAADAIEIWAAAAABhEyAIAAAAAgwhZAAAAAGAQI
QsAAAAADCJkAQAAAIBBhCwAAAAAMIiQBQAAAAAGEbIAAAAAwCBCFgAAAAAYRMgCAAAAAIMIWQAAAABgECELAAAAAAwiZAEAAAC
AQYQsAAAAADAor7MrcLdTVRERiY+Pd3JN/pvU1FRJTEyU+Ph4cXV1dXZ1ciXawPloA+ejDZyPNnA+2sD5aAPny+ltYM8E9oxwM
4Ssf3D58mUREfH393dyTQAAAADcDS5fviyenp43fdym/xTDcrn09HQ5ffq0FCpUSGw2m7Or86/Fx8eLv7+/nDhxQgoXLuzs6uR
KtIHz0QbORxs4H23gfLSB89EGzpfT20BV5fLly+Ln5yd58tz8yit6sv5Bnjx5pGTJks6uxv+scOHCOXJHvpfQBs5HGzgfbeB8t
IHz0QbORxs4X05ug1v1YNkx8QUAAAAAGETIAgAAAACDCFn3OHd3d3n99dfF3d3d2VXJtWgD56MNnI82cD7awPloA+ejDZwvt7Q
BE18AAAAAAgEH0ZAEAAACAQYQsAAAAADCIkAUAAAAABhGyAAAAAMAgQhb+J6oq6enp1t/X+/z8eL1YVLly7J0qVLJT4+3tnVy
bWSk5NFhOOHrDA3Xc5GyMJ/kp6eLqoqpa0tDTre4vvr3ufzWaTLVu2SIcOHWT//v3Ork6ucf1
na+7cudKkSRO5ePGiw/EDMthsNmdXIddIT0+XtLQ0o+tkj8Z/kidPHrHZbLJz507p3bu31KpVS7p16yYhISHy9ttvWwEMN3ft2
jUO5MA+cBewt4GLi4vYbDZJS0vj+yuXaNWqlRQvXlx++uknuXbtmrOrkytc/9lq166dXrp37y7Hjx93Yq2cL6uevL1798q
8efOs3j787w4fPiyxsbGOzgzx58oiLi4vhAkIW/hVVlbS0S0nNFmwYIHIHKhQ4fKk08+KeXKeRI0dKv
3795MKFC86u7l0tb968YrPZ5OzZs3Lp0iVIcnV2dzmMh3vwbnjhgfNGNbQxmuT1V/4KtFgwCBfdwyx9dR8Rg
qqohkHOgvXbpU/v77b2dV7Z51Y3BIT0+XIUOGyLx580RErpHDhwjJ+/Hi5du2a9O/fXy5fvuvuyMajrN9b3n1/fk2d+3MWPGyBdff
CHu7u6ckDPg6tWrUqVKFZk1a5a1zN3N4pcODAAXnxxRfl6aeflh9++MHI6xycFvS09Nlx44dYJ88cU
X0qlTJ+nZs6csXLhQhg8fLsuWLZMFCxY4u9bpTqWpn0zuj7CJRA7jeFyyRS8VfT0Y5i4KbH3zxS6NGNCi9IsT/f53F87H9y2uAetK8wMn56P10MdfpdpV7SfL6K0vdt4bC9GP2D7z1MXH2O1MIwHmfsYj0e
ZVbwn5cmTR44ePSq7du2y/v7jjz9k1qxZVm+/Bv7+/TJo0SX799Vf55JNPPnFndO+L6a9ntvefx8fGyfPlyWbVqlYiIdTKgQoUk
piY6LS63mvy5csnzz33nEyaNEly5cplfffzL5/wt/Nlb+tp0r0WQmfmXdk7lWHjtT3DbPUzduPBbdmz4wLnfmsgc3l3oGmJX5BnIn+yt9vGsuPbSWNTUX7K
ewXo954UHfq1CkZOHCgNGrUSN5//32ZM2u3bZj1VVDkJXlrzD7FzHhbvRdVZaUqxLb5Cgu6utxnvC2urbPxlVRpF8yfpxG3L2rjUT
1544QXZv3+/jBkzxnrM3d1d3nnnHHQkODpYDBw7IxIkTZf/+TJ8Xfy9UyFMDYOy4b9/B4kNjUZ9frjyA+sRjyn+/o5JcucXDX
8EhUVJf369ZNPPlvEXF1d5bXXWhMRkTYiICDl+/Ljs3LzjtY9p8nqQD8hIUG6d+8uZcqU3VuWScU/gtZ8DWj1Ncxg8QPLEyOT
OuRO3bsKIsXL5akpCTrIDgoKEi8vLzk6NGj0sMWSlsFh5vNz17HR7fb30cuXT/bs2SNLvy6VhIQE2116RklSoiIvLuu+9
KfHy8DB48WFTPU6VChQpy+vRG3uDFJxn9AsrzuY/Uno7iQ8lsGbOGiIsiCmA9Wk8ywF8NzvpaXqe+vOi2T1MtMVBMNLfiZqj/ZaNz++nc0l47CvX4hHn4IMdvvyDRisVQ/fvQbGvGMSgThjAJj6wGzOnCA9Ua2ckDvzPb3nkSpO9AuXn4cAgOA0uiw
nfuLN27d5epU6dKYGCgPPHEE7J3795M6/43aLFcLiYmRgYMGCCTJ08WkYwvvxt3VBERHx8fqVGjhhw9elR27NghIhlfiEl3SdZ
ZJ1WVvHnzioIl5eXJCUlSUBAwJ3cnDvm0qVL8vDDgAYNDw8qxGAP8t4aAuPj5e+vbtK5s2bZKIiAhJTEyUfPnyWRf6XP744
w8ZP368XLlyRZYtWyb79u2Tt99+u99+WypXrixJSUkiQm9GTuhDh4SEPPBAph+/nTt3SuPGjaVkyZLy7LPPPSvv27eWnn34SEZHWrVt
LyZIlxcPDwypfr149KVmypGzfvl3Bi4uKYDOM/sB/QiIhs375dfv31V4cz4ZuebOnSs+Pj4yaNAgEIkKDEVasmFSt
e+++046dOgghQsXlgsXLsjcXPlr7/+utObg3/h+l4l5ZUaIkl5qfJKnOqeXx/1dk3E36DxNBwNBJAV9bkZVjPdmYO6TQvTxeWB2S6dnl9uZhQlfI
efPFFSUxMFG9vb6lbt6dp6Nq8AEW15iuFmtFwh4F8LkF7IlCwc73KxmhFEXA4NKY1CngKzjZUDWnlAL+8r4nDAafwkg4v+7
VfrNTw8PKRo0aJy/vx5SUlJyeatMiPNBPp+ejVmKN1psihIsJIyBWLa+wzIeLMy27bDjhwZrE+ETehT89dqKIEr/L8uKeRvRudG1
2bXn//ffljz/+kFdeeUWOHz8uNWrWZsnm9BYaHB0FR7CXTd5m2d9beq5RdaPaAo6qTBjD6x+8pLvcZa9ic+iKIeLJTEoMRCPYYWQuKY3L1FCj8Tj5yjlawdkkbNDLPz8iPbQuTFXsPlXLwc/G8rXjfJZjTXw+rYRuhEMB1Zy7bRs3bqMjN7lHB+l4V70tVCUynmqnTVAJ4kDb4nSeJLj5G6ZEPDhUaVz3ZOFT2Z6ObzfZVG3+dMT2rn5GMcj54qI29q88jg23m6DCWcEOm6WXmDmZmUp4ZKEeWSWzaDlhlqbAXAPRaTgBOlZUEGG5vPYaYFLyN4aWgoqxkL23QRxsnXhnW5YM1Thg3NuDtxXTJSCC2QJdDsubgI3uHsYSjtA5TWmX9AtsYRsgW4znbHBK4J7nhM60rtHAU9mAPaVAThjKr6fcm3XijD6CKfH7Wzc4UaonSt8Mn1vxTuQzmQlbcfXk+zTqwsFv5f6Sr1IKwjhYRsYWH4aNXZCP6JBUIDSRVlC8BjJO5FFxGwGEPfwHhEZFZyM4Xrn3X37lzIJm4++ZjtsMtW9j/bfjrS4ZoT6zSYvo3sFxRt8ncyx8/Ra0ri2CkqpuQ1tC9WsSvzOyAjnZC7fzT9KWI3m55M3+Btmz2eFaC0hU4HhWBmzswVfYMPbYrwi9lxpVddf1H9cOiL+H4jugtYfWG6tWgcKZZf4X8ItRkztMwBxPCfovz2f2HHK5jwy+4Fqjpj4Hf0d79tMzxA4ynxnrBvq+96JmmgCLSuWRvQX7KMzP75Er+DoNz8ZuTdMJhLvD4Ee2jFGZ8J5R6o56dHEsdwuc4v/7qvxKd2DxXNk9cjdq+5NgbQeAGxaB3zOJSjJlSXqU7v+HwSTqPuEdeYvPFbfHwxfnFUjuiRNkoVE/ulR9RqHVG8LXCaG37dP3Wdyt+Y/8QXv8mv8nmXOpoZWmtyY1k7WhHxrTv1dZv1lKFk9+m2qVAeAk8iWqwRnyehp8XeqLd/cdRgc32zkvXvSSdeFvqMsQ9lT5ZryjOa+3BQ4zPxbNyrHkNpHt8LxzgNzSd93Ztw96I9HcLlUgb2YPAEeGdHaDRYOjMtPRNGgbcFTAw8Cj7QTBO/Zc5UpfWjUlZBJgUkhFz0Cyn93yGq+fGiBc4IpAUhD5oPjAuhBYcb//AJTwLkZyp48evNtG7HkMvSWqU+OsVyqvn/SVS0bNRtnc0mO9khHcVl+0bSmtgjB3/1rsBX96hu5ooc1TIqVlf3Ol+mLCRpQGh9nf6X7dSDWiK7e+XEj2j/Wt8mN8+BAh+4lbHTQSERNfl5APyCDSv3DRUvVhUrRtvX+rbcZvLrsBYkBVdY+TbQUpT9F1uP2gVqDP9OQ1kaqwcoh2IDtYDKyW3lqUHNLpb0ZTZ1tQ+EXBn/PkGbZtWD6/7MZdMrKDmtW5ezoPxfK5KUtkjHwMlnEDRS/xMPB6E79G72hhfNQzCtn8OO6IKWN8kJpkSJS08ZP1QwZZrZZ5sxc8f2/r/ExH5Kq5cQWn2sTwIrzIXahVr5mxvBPeaDjfxZhUn5ZBtMd7FWJZY3eTbp8dJBmyJyxWeuYQH/wRfwZd/lWhhmyTV7/TgkpTS6x3kBRdRTvMD3wbYhw9vHUDz+0jbhdCG6UQgn0GPZAf1a5/JrQ0ef7+bfvZDgyLwyaZ6SqpuS5eEr5kcbemt5D0PmSxlVRv3XdYfVxoIyOiVG0c9CAHgiCTa6UqH1rVfVsIYTJU5hAwQvBWJtlVgIpt/ApTBw6hJbvDzCE3mJwWuIvcfgxJ+AYXlXV5ClNBJiO3GbI9znrPTE6i1X1kozKlELg3z3yNzQjHmKlE+mdKVgfqGB0W1W19uqJWhzQlCj9vIt1HZaXGbVrf3i87WHQqylgkuLMgLEAq4m/H+bxbpO0Yrz6JCIVq4uqAYNs3OCyqgW45HdcjwD/1hKzN81HFYdTDnBbB8C5LDUTO8d/7OIx6wrcHxhvz9r2TukNrLbNiEQIdYCd1tTKvp5Gn7FcBIzyzbSjTbDb8iq3IEzLt5PtQUl83j9jELNNwMtWI+j1IXXzSJ/bKH5pE3BcR6p+KyFmOH6YNTfmaUOkz+CpaMuFEF5Xe8Tf9BCsk3zz1Dtu+OF6+Yx6+kR37bDb6mmyN1lWpHXxg4oNJNK5sQ/+9zn5m0JNs2fZDF9WxgPbLwLCzRhhWL5K3tVZnHLvRBWRNO9xYSfA8V9kUS6/gC9iz3aEJ3OIxHC6/7ssGzwAcOC5iM7Ggr4B0TdX06OxlcJZpLlK17E1DdUPUxQOF9SOVpgSONn4i3cuG06GfgeGlEMDiBm0XpORyZ7IEi+z4/8OLi+V2Ji9gJXrydnPywMKJczKaKFrNK5aUcOXRJvWwk5jhdQ1rQ3WnYFkzVhoUuS0gqP38bf7PLmqIONnjvRtMJhT4zS8pPrDdPHbUP/WZgBhJ9WRhdYqVsQcgTMsuJhTpJYfRqtUQTxNlSQDAeVn3l6F4m6dmO/xEuV4sA/F39ShPyLQoWHFg0V5QHsGr2JFXSWTgwGV2nMEDqj3WWNH5xPB3tEQ1WmpNPZWEQyk0+7mSzrCPoQq9Cqq+/2f9ltbQEiw3iJRKk5B/CEcMBZIXCFYYgMOWfXMpA5GYntEiXt/XFNvO3HUU1N06cgcEu3QX8tJBmJd9GzEgaFxMEcMTTL+KMKMUzKCNkJIICOCI5CGJF13i1yRSrkRsqB3e2n88YqHQd9MbtZ3k44KnbrUGXS79QKIu9Gd5AOu5Sfa/uHdfjrw5ezXH+qsVBqsMM4K7/Ka7dvTrrbIDrXsQzyV/W66aOsvWdvLVEjnzZFVrfGrcHDgkKZKqjlW5UNgxXuK4+7AY/9F5fgHV7OKtpNqyYn+88pMY/iMKqVS/X+I1U60vkdaRWKyHdV8vDbFqMwhZVU7q2kOsj2TZvuMRijHdKkjdGGrA8FLTzo8/Ys8C73A3QZPfy0qsv+RlmTr6OKNKSmbJ+QeGsTGa5kd1P5lNr6p2lbgdNnFX5VFcG2EbCYFnTuG4nbj3z3E3TvbkWJ0GvNpMSWFZz+ihHuqr6xXJ6qtZHX7G+JHOdXqdG+U8RaSN6GShDzaGIUK82rKufOJgxQXPkbMW8VOxlVDT8HkPf5bI1CXxqfzR/Y5XyaQz63pOsj8a/tR6+1sQqOnnJlYxH5jj4TfOL7CbEoECvRGTU0TyLP+cJxWCZiVYzDmCkOILfj+P+rxF3r8+PRzfF+5BvPBMMsI9AtWqDjTCqZPs8LTxZRDyz2mAbKB/0YrDvkvTXGRUCFwVHzewFdoZxPyuhrvAJ4OBDTe0Jrv0MAh+EfGXo0amBL4S8f+2eIVu3tUd9UEpm3+e6aLqBrQeNNmP2YJBv1IhtfNa9J56IFVkFMr7k8hZP+bPlhPYXffoIQK4Z5CUl1akmnWhUDZaUjf3zMmVZn7QAgJe6tD7OX8NbgnLSOD2MG0q7Wp/LQYaUGl6m0o/SKFN4V9PWA8wzbSm8hk22jIqWuiOgK6UK5QsmtqcDs9dKqHlVMhZKO5HuTyFX4D1cKpVk3nCsnz7yi0fRwpUs6s7yHxRJ3SR7KRTjUYP3MKlf4JmhG22SDMhNX32TR9D2sGBaLLxEd7xuyPaSf8ON9XD+2DFmxHlrnWr7fVeQPDPSiLIuROBTpGPXkJ0V9v+6cnwaOZfq8Rx1HB6L3xTGMrIfkQXvQSqo6cgQOhTr68P9CjAUJiwnDw1rWdnCqlVXDJ12o9bQr3R2OB2e78jM0SkYivgjH0r5bnixmuG8rbvgoWWj2F6E3Ku8VYDU+TS4ZNjd8yKuepnaRLXWJAu0xbMeudq2Aou3X3/cL7Z+HxhPCOB4cdiNNk/q9Wa4/fSzDB/qLxwwLqH0A9Wt+LWXy8eByYgv7Vz8Ydu5JjHuNqbzNSPv2K1hPDE8c7aAUS3ZhdRzqjsMq2mDz2M2ZyaD/wtoIvpzZvZhD6D4TIKDHGlKHgtQXVzNAIyZgRgGz77nBLIyf7wD1PRJ1FbH/5NiUiSqMqF/cAGVWSY+pHs8jXxHLRsBGbLnx71YJbVy3YwJJnBzflV0/ww0DFGvzvRwKavZp9U0tT5dVfMsLlOiqUncD/IY1uOkN3nqCVXgm7efxRDjfjFDMi2ZEz3h+4BFrUJO1+8wV8URGs7K8G6DrkzxGKxOSa9J5PUA3Bz2V7ezY3xymS6LfhrgCWd1k7znmiqVGsuzDZm/2iIAYfQe/bOXQGU58+v8q9C2fR4+2N4O2YujnZ7pBuYKi/vB3WgjcKPXDuIhp3+v+HnKJ2l5j1j5gEMpTM9/PYm5Mm7jzFY6QgaX4yUgAULpHlUDRyMzoeb7PZKhV7JmBqMCGmkq9KBQ/ThhOSvYuVOvpC7nOYqrz2MyyYzF69a95bH8QSvgL1GBSyjYDnXm8m9Ad5A/vXkK9mjOa+zsW6Ep9gNdYJqwsRpuSCw6f7MurhnixAAEgEGYtCJmkMgdnOdLPA48+aJLQKYC7OsZmpW6NZhHKZt3hODsAK7p8FEGwZpA+Lhp9ITSaDqySxTNgvBhDRx5b4rdR3Y5Z5eSmLZvWd5WUvpZQgTjsFsEBnd2wMBgnjfjgwnHWGQbZrsKGEaN0x/0WhOBQ3BD6S1TjLS4ILqrUK1ZQD+6WrnKJXSOk6eoTXw9Y8RWYOPsgHObNAj9+
Zs+yfvx49emj9+vVVVXX37t1at25dq1fY7tlnn9WRI0dqUlLSndqcHGnx4sU6depU6z1Wzfhes5+hvHjxon7yySdavHhxrVatm
kZFRTk83/7+rl27Vj08PPTw4cOqqnr16lXdunWrLliwQGNiYu7Q1uB27dy5Ux9//HEtUaKE1q1bV/v3768nT55UVdVevXrpww8
/bM14O2/ePK1WrZoeP35cVVWjo6PV399fe/XqpQcOHNDLly/rDz/8oJ06ddLBgwdbvcq5vd137dqlvXr10t27d6uq6pYtW7RAg
QLasGFDq1f46tWrOn/+fM2fP7+ePXvWem6FChX0rbfe0r1792poaKgOHDhQX3nlFe3cubOqqs6ePVsLFSqkly5dUtWM31Gbzaa
LFy+21vHSSy9pz54979Tm3lRcXJympqZaf6elpWXZkzRs2DC12WxaqFAhfeGFF6ye8Q0bNmjBggU1ISFBVf/vd6BJkybasWNHV
VWNjIzUBg0a6OTJk63XuFGTJk101KhRWT6WG1w/c7X9ePL8+fP66aefWiOl7O/tAw88oB988OZrSLq6urent7a9myZfWVV17
J1EN9/vx5LVy4sC5cuFBVVefMmaMtW7bUJ598UletWmk/oSsXCI5OVkPHz6sVapUUU9PT61Vq5ZOnz7derx3794aGhqq58+fV
9XM4ci+k+/YsUNbtGihr776qqpmHMy0adNGg4KCdPr06blmmtH169drs2bNdNu2baqa8YX62GOP6cSJE/X8+fPaoEEDHTJkiFU
+JSVFW7RooTabTX/++WeHdf3555+akpJyR+uPm7t+ytfr7dixQ1u2bKn58+e39nf7wUJQUJB27dpVVVW///57rV+/vjXMwP5Zm
jlzpvr7+1vrfuKJJ7Rp06Z67Ngx6zVyy+fnVtLT0zU1NTXLEzRLlixRf39/LVeunLZp00bz5cunH3/8sV65ckVVM4bv2IcMBgY
Gar9+/bRWrVrWULKUlBT9/fffNS4uTnfu3KkdOnTQrl278vm7C5w/PFHVc365Nzly5f1iSeesIZ4rl69Wps3b66tW7dWV
dXTp09r7969tXTp0hoXF6e//vqrurm5OQxZX7FihZYrV07r1aun5cqV00KFCmnv3r31yJEjDq97r54cvBX7Nq97du1bLli3rsLx
Vq1b6wgsvaHx8vLV8xowZGhISohcvXrSWvfnmm1qmTBlNTk7WvXv3apUqVbRQoUI6dOhQTU9P19On6uXl5fOnz/fek7r1q31g
w8+sL4XP//8c50wYUJ2b24m13/nL1y4UAMDA3XLli2Zyl24cEGnTp2qI0aM0KtXr+pnn32mrVq10tDQUIcQ9u2332rNmjV12bJ
lqvp/3+2TJ09Wf39/VW9dOmSduzYUVu1apXpO8i+rlq1aunw4cMz1TG3uf7uf7WCkuXLlVvb2996aWXrGVHjx7VZs2a6cSJE1VVd
fv27Wqz2fT99993eN+SkpL0u+++sz7zDRo00GbNmqmqamJiovHPPiHrHmE/MLlRWlqaLliwQEuUKKFvvfWWTpo0SY8cOaI9e/b
UEiVKWGd4Z8+erfXq1dOlS5eq6s3H/qalpenTTz+tQUFB1pfrvXrd1a1ERkZqnTp1VDXjWpBRo0apu7u4uri4aFRUlI4YMULL1
CmjU6ZM0TNnzuiXX36pPXr00Fq1alln9W58j2+83gBgB3zu0cXHXr1k3btWunJ0+e1OTkZL1w4YL12MyZM9XLy0vj4uI0Pj5ew8L
CtE2bNg7PDwkJ0Y4d0+rly5dVNeNgYsaMGdaZTru0tLR7/iAzP1pT1dExMT9ZNPPrHuvZeenp5pu6+/Zu3SpUvarFkzq8dYVfWDD
z7QatWqWYF27dq1+sADD+hXX31/bA++OCD+uijj+qVK1f0yJEj2qdPH61cubIWLFhQ27VrZ70+nCctLU2jOnjkObX2/
ixIn60EMPWX8fPXpUn3jiCXVxcbFOdqWkpW0R48eOmvWLA0ODtZf/3VYT0XLlzQH374Qb/99tts256cbNWqdqiRQvrh
Kuq6ltvvaVNmjTRXbt26alTp3TgwIFaoEABDQsL0y+++OEObvv6pU6e
s67mc5eTJk5lOOMfGxmq3bt00f/78GhwcrD179rR6ff/55dbb+mzzzz5r9che/
5qTJk3SgIAAXbNmTfZu8F3siy++0+MaNG2vDhg315Zdftj7Pixcv1pIlS+r777+vqhnHTjrpz505VzQh
qH374oXbu3Fk3b96sqhn77PX7umErBzm77//1okTJ+qmTZtuWmb//v3W6+p6qqZ86c0aZNm+qTT
z6pqhlDnkJCQR///43Xa/9oGXbtm3WcMN7/WDZg4ePKg2m02rVq2qrq6uWaNf3ss880PDxcW7ZsqUuWLNF3331y5Urp4U
LF9YSJUrookWLdOTIkVq7dm1nVx+3cODAAX3ppZd02rRpevDgQVVV/e2337R69R69erWF3pMTIzDvn/u3DnNly+fLlq0SFUzzrrSVK
VNGQ0JC9JtvvtExY8ZoQECAfvPN3d+g+5Su3bt0pCQEOs9touPj9e3335bH3roIW3RooXu2LFDVTMOZBo0aKCqGd9xw4cP1xI
lSmi5cuWs4UYjRozQMBAKwBv3bpVy5QpoyVLtSvvvpKVTMOIpcvX57lSSncefbflY8++sjhBvfXn31OT0/XUaNGXd69bXX69
On6wAMP6H333adt2rTRb775RlNSUqz2XL16tbZu3vLoLFy6sjRs3zjS50I2uH46Um9m/zyZOnkihoaEaGxtrLdu1a5a5dWrVpVvb2
99b777tNHH31U33//fY2IiFCbzaYjR47Lg4Vc3oGQgPD7/p6xw7dsyhR0z1zk7okNUxS1pamn7++eeZJrDYu3evDe5c+dqY
GCgFeivXLlirevgwYMaEhJiTeJh9/7772/772uhQoV0ypQLq3jx4+/Z3vaT506ZZ1kzKp
d5s2bp5UqVdJXX31VFy5cqE899ZSWK1fOej9ee+01LVOLVOmjHWZRaVKlXTatGnW83/66Sdt0qSJlihRQp966in18fHR0qVL63vvv
WeNfshuhKy7WFZndnft2qU3PUUvvPCCChjs3TkuUKKGlS5fWoKAga5zvuXPntHjx4rp1/0qK1aMqSN4SSJ2klXPd8+fR
76aWXtGbNmrp37169fPmysHq2FQLiOvUC6Cn6e8PKaTY/HRyfKIEvYNgz+/ZNhWi8DZCLAkeLU8/p0dpzuAPm2epGtNjSarCwvM
rlxohuHA6anp+uVK1d0zZo1atWtW0YcOGWqNGDS1VqpTGxdrYmKitm/fXitu/fXitWrKgNGjTQjh07apMfTZ5+1rj985JF
HNCwszFrv+vXr9ZlnntGTyXMV4Xl+Xe/lExb/dvj///FObNm2qderUe0ffee0/XrFljnbnbnbR4cLaq
lUrXbRokcMJpSlTpilSpV09OjRum7dOu3cubO++OKL2lEF2/fr3x7sYD1b59+2ZZbuQoKOPD1L
NPMz2559/VpvNpmXLlrV6GCYVhHCmkxDt27fX2rVrWyc97VF5f5++WW9//77/eR24Px74Y/Xw8HCYeTanNeJ
0/W0qVLW70eixcv1ooVK1qBB6NVVX9WSJuqasbMc0eOHNFr165tqyZ7j1S1G169qr7/+ulapUkVL8u8Iu0q2sIL2T
Zc3L9b1JuCP3z58/XypUr69dff62qfeRs2fPqre3t8O1UT/88OqZDAZ95513VcjC9+9e3cNDAzUr7/+Wp9++m9l++23Hdbz9
99/67fffqsjR440dp3Vv0HIyiH+6Wzb559+rrVr19YvvhCz507px999JFaVL0p9J9Hv9Phy/OPXv2OFz0t3HjRm3VqpW
WLl1abTabPvnkkzf9+efa1pamv7999999uxZPXihL722mv60EMPWRMk661edn8+bNarPZPZN
Dg42PoiPnv2rBYtWlRHjRql6enpev78ePzZZwy5YtU5vNpn/88Ye13tTU1Fz5Gbz5GbrxQ6
PP/7Qn3/+WS9fvmy1w7Vr1/T6dM6atQo69qHBQsWWJPD3OiXX37RkiVLas+ePR2GKenp+uePXv09OnTevHiRX3rrbe0QoUKW
rx4cR00aNBNh5/hzvg3PUUDBw7Uk2aWAeq6enp1nOXLl3q0JtsX75//3597rnnnrH3Ovtw+lAv/zoEDB9TNzc0aTmUPsDNztT
AwEBrmKV9+QcffKA2m8260/Wbr1q368MMPW9W9f5HDhwQB999FGrd2rHjh3q4q4vt1I/vxqGRkpI4cOVJLliypvr6+Gh4
ertHR0dZ6pjaa49q61bt9bGjRtriRI9Msvv7xj2I7CFl3scuXL2v37t2fmtmfzs7GOiV6xYoaoZX2oZWqGpGpGN/ZHH32kb
m5uGhYWpklJSbp9+3b18+3b18/PTefmOay/PTefPmOay/ffv2WrduXVXN+HGKiYnRJUuWOIWPRmY7d+5Ud3d362zqc7U7U7dq1S8PCwvS+++7TmjV
r6pdffpllDyWyR1Y/9/auvxr9dBgwbpjBkz73OXnwwQcdZssaOnOn1qxZ06Hn+HrPPnlnmm+/eXxfNsUjQxMVErV66c5VDem
81MdS87ffq0Dhs2TMuxL2/1AHp7e2vHjh2tg+hjx46bvuVbJEDx06pCkpKRoWb16tjXYh2a5evp7qq+umnn6qmaGmS6evm+ekhOmTJFt23bp
WFqb16tVzOLmxceNG7dq1q27YsmFa9tdf93RbcU/u3jxojXT342jhBs16evLhOmTJFt23bp
h988IE2btxYn3jiCYeereU/X3Pb5+56Wf3OHDx4UGvUqmQJFpmK5dTeyMNmjQQHv16qWq/Hv16UmXp/xejmhd/rnXr1rXu2aSa8F1
S5Y4Nv/qv7PvY4MGDtUKFC69X6DiNGDFC0ZmgqldSokXXd5ubNm3XVqlX63+qapVq3TTSpEnq6+urmzdv1oSEBB05c
qT6+flpkSJF1MfHJ8JuXnbdOjcNJFftz1qxZo15eXurh4aEsG6cOHCm16rFxdrXXq1Mk0U+XdgJB1F7t
27Zo2btxY27Vrp7/++qs+88wzOmvWLD179qyWL19ehw0qZoZXxL58uXTyZMna1hYmBYSWFBr1KihU6ZMcTi71KFDB+usu31H
vTTT7V48eIMX/uXIiMjNTg4WH///fcsH79+IoUffvjB4QAA2cM+mcL777/P5Vnsv//+WWb9+pWWtLz26aK1atbRRo0bWyYq
hQ4dq2bJlHQ7QDxwJBGR0fr3rr17dezYsDEG4m126dEmHDh2qNpW7RooZ988okeOnRIN23apPPmz
dPChQtrs2bNrKmyw8LC9LC9KmmnntKLFy9qXFyc9uvXnvx5ffHHFFzU4OFi
9vLy0V69e1okhTmA4R1bDcFUzrogrW7eu+vj4aIMGDTkyJFWW904NfOZM2e0devW+uyzz2b5GklJST
AGdPHlyruwtzkpWJ/CuXbmMAPgG2+8oUFBQTe9FigugGJUUnTZqkhQVvyjRksEePHtq6dWudNWuWtmnTRgsVKqTaxmNmjXT
6OjoTOu528Lt9T2cpUqV0jlz5jgbg8vnr1aq1WrZred991ontZcuWaf369a369ykpqba3RHz9e0XvhN9mg
528Lt9T2cpUqV0jlz5jg8vnr1aq1WrZred9991ontZcuWaf369a3e0xtFR0dr4cKFrSHjqhmT79zsuCC3uP7kyM0eV80Ymmgfg
aXq+N199epVfeCBB7RPnz6ZnvvBBx9Y12HZ1/X7779b++Hd9BtAyLpLXH92w/7vlStXtGHDhurm5qaFChXS6WR/enj17alh
YmP7xxx+anJyszZo1U1dXV33zzTd1x44dDjuZ/Qt27ty5arPZHMKZE67+/m2w2Xb2sm
bff1ocffti6QDsxMVGffPJJLVeunKpmXA9ks9l05cqVDuuuU6eO9u7dW9PT0/WLL77fixW56nVVun1AhKSlJ+/Xrp1W
rVrWWXf99NHPmTC1atKg1rfonn3yi5cuXt2Ygsx/vUy7b/JtSpUsO7XLo0CH95JNPtHv37jp69Gf9G+3i6H7zPcebc6oNq/f782a
tRIhw8frn/88YcuWbJEH3/8cWuEn/gmsHiZnPu6vFa8UmkBA6vt/ruVroaLa/osXL2qDBg2
0Vq1a1omiDh06aPFu3bNcj92FCxe0RIkSmT58T8mdEmRIkSOmrUKIcwtnjxYnVzc7NOb0bGTVU+Nst9oP7I81btxYe/bsqatJFtXz58
```

```
tqtWzdt0KCBPvfcc6qaEfrbtm2rzzzzjKpm9KZ+8803unbtWn3rrbe0UqVK2r9//ywDa3p6eq4/Tjh8+LCuXr3aYSZe1f9rhws
XLmhQUJAOGTLE4QSN/X0bOXKkVqlSRYcPH66HDx/Wo0eP6htvvKGNGjW663qsboaQdQfc6sN24xfC9Rfw2ofz5c+fXz/++GOHc
kuXLtWaNWtaO9qwYcPUy8vLoUxaWprOmTNH58yZY1230LlzZ4cbeOJ/k9u/RJ3p+us1rle7dm0tUqSIuri46EMPPaTbtm3Tq1e
vateuXa17uU2cOFGrV6+unp6e+uSTT1rDz6pVq5bmp+JEydqgQIFdN++fZqQkKA//vhjph+N3OhWBzNfffWVVqlSxeGWEPbPy
unTpzUsLEzLly+vqhlht2zZsjp27Fjr+y81NdVq288++0ozr1q2baSpu3F12796tw4YNcxjq99prrzlMBvPTTz9pjRo1NG/evJl
u+mtv73Xr1mmjRo20b9++OmzYMPXz81MXFxdt27Ztph6Cm9265F4VGxur1apVy3QJgd3u3bv1nXfe0a+//to6mfTHH3/osGHD1
NPTU3v27Kn+/v7W0NqsPsP2ZXPmzNF69erpp59+aj129+aj129erVTDd0vRvCxO3WwV7mk08+0Tx58miRIkW0S5cu+t1331kz1r3++ut
arVo16+T0mDFj9OGHH9bffvtNk5KS9JVXXtFy5cppUFCQvLJJ1n+BuWWwJ/VMPi0tDSdP3++dY+0jh07ZjmE3v4e9e3bV0NCQ
qybX1/fGxsXF6fvv/++li5dWmvVqqUFCxa0psrPKe8xISub3bgjbN26Vjpg/nll19qaGioBgcHW3dMV83YYe1nV1T/7yx
5XFycNmzYUAcMGKCpqal69uxZvf/++7Vp06Y6e/f/++7Vp06Y6e/ZsXblypXbo0CHTdKHAvWj37t3atWtXbdq0qQYEBGjevHl19erVDmWqVq1q3
YC0Tp06OnXq1Exnw6dPn64lS5Z0uAYrNjZWX3vttUyTnOTG66wSExN17ty5Dtn1lFk4y5DurUyWG53dixY7Vw4cL
WAVt4eLiGhITo0aNH9dy5czplyhR9/vnntWrVqlqkSBGdNGlSrrmO4W51s2tJ9uzZo4GBgVq4cGF99NFH9fXXX7fO7AcGBupr
72mgwcPVl9fX/X19dWIiIhMN2S/3uXLl7Vly5Zqs9m0bt26Onfu3GzbppyoWrVq2qNHD4dl58+f17CwMC1cuLCGhIRo9erVtXb
t2g4ng77++msNDAxUm83m8N14s+8+msNDAxUm83m8N14s+8w+7Cs0qVLZ3mQfP2JkLvJ8uXL9bvvvrO2/frvHvv/X7p0SUuWLGn1pqv+X8hfsWKFVq9e3
bp+PTIyUgMCAnTUqFGqmjGD8I3XBuWUA35A35Tstpe+2d+9+7d+tBDD+m4ceP07NmzGhMTk+UEYfb3+7vvvtOaNWtaJ2dUM/bnJU
W6Ny5czU1NVUTEhJ03bp1eubMmWzaouxDyLoDDhw4oM8995zWrp0aS1durQ++uijunv3bk1OTtZevXrpAw88oK+++qpOn
DhRGzVqpLVq1VLVjJ15xYoVevmidPHuv6KvuX4pAhQ7RZs2bWlKo///yzRaLapUsXh7HCQE6V1QHe1atXdc6cOTp
t2jTt1KmT9unTxzoYsN+bXper169tFixYtZYbrszZbrszZ85YPS6XL19Wm81mTSDDPbnnO+++07Ll7C8gmkj8+fPM6WM8Zqz6hlPT
0/Xt956SytXrmz1WFzf6zB79mz18vKYzsivXbtWS5cubd0dkZ6JE7ZY6K5lvmD6S6fvZ47ukIKoxP0/dL8q8ivXbpj8uxu9E9KfSKZFC0qU1-vuxw3oFwFqpP3j5KMQjLoYBIaTVgWgZKMMoGMfQWJAqAAMABoAAABNVHJDwfaAvAAAAqamA+FH7dJPNmNEYNR2+a+0sVTA1m6tKXtBC73hkjQ2RYqwbFvXHUpVtWW0Pa4ZUNpVYqWPJy0d4CxXUS27AGh7J7DmPZfGNrFW3yhGMAKtAkHtp/Ap0p2IgHVOeREXtgR3iGUwmLCqOxEJyxzLcuhK7NUR0CwZ3hCIX7dDfoCFbpLECrHJTxe77zaMwwd2QPBf3G9BCS4CY0IK0o53p3TiWY0ZJ6PrvmTIyV5O7N87IY1XYlFYlIPq6Q97m1nzZWVt6gwBxnL7Hp0U1dpLmo2B9JVu3+WQSqHYF6f4uj+AP+6DR3jGq1GkOp58IgZuSRIa9SdE6YHgF6ZDeqwn9sLfUcV9IlUY7RXOWVz8VzQ8D1WUUzG5qhX9t/cGDR8PVjg/55VJN4pnmu0UZ4R7+C3sO5sw9hMYiMrOc3nJg1jm9c2GJVT5n5Fn6rfzKVDdq2fA5HjN8ypcHmQrxYwHL7qo8NnG3qhd5g2QjNUB96jCHyGUlYH7gaLQ7TYY2DjV3/BMMgSChArGOsY7BHQT2Vu7bVUT+HQ8ZlYJdpn1xPBNOLf9mKyaDY/Ap9ZwqtLAoEQxXmo6MoCNJZApPrywbFoK2Wu2/0Ix6qbYmqJ+hN8k65bO3j8HuSgnEJvbK3Hn1lk4U+P8Vz7b2F3n3yCdxPq4L2lyA9KbDBFGcNN3/H2JFmF9mpBZNpt7cBwsF2Qp/q2Thd4U0X1B2bAFG3v6OHi6mRnR+3GQD6lXnDDMB8vVeE2z5b3c0FWBl9m1dKWoNYy5X/Xxp9DBnV/Xx+HgK7DYZDWIHM/Aj7Ft7lxhFe5h05R1hk4GjuxdHKqL3pN4i0I0dqPqnfVoXlKfmVONoTp+y9+bnVNv/pX7Jo/hH+CAXgTF/6kTyqBNBWHn2jWNyh6J6wOQqeo1wRNqz8DfcYy9VnBDXq3YnZMu2ZcAIAZIpSmQp8qnVRJ1zdEHh3qt5Y0u3N+cDnIOhJdXhP2bGU3TVPF6qhmWWgjSyRbCl+hMKqG9PKrmUzCqr+O7V6LHzUmP1fXPbyuhL/kUFAe1IamoSNQ7BGDY9FfBU+k9Oqa1vQM4HbL9F3X0GNx0pC4dEfb+S3JV1MeHYq4NhnUXU3JHoe/Yw1YuQzyMVX+GSNM2WeLMWk6cy5dzrHBXUT6VcPFUyAXSJ7ND3aPWtRKb5jTI1BGp1jT8mf99U9XOqeZk0u8Zd3zEq7aGIgOUIC8fYKDRCf5JOJ7+FaWd6Zo4R+O1XP1X0+CUK9z3q3o4JEeOaO+9qsaG7TTLWNH8yB9Rw0gK2+3XrN6OWxN5s5ZHSzJmDdr+YZsQQtKyMfRSe5+8SpDQKsvT37JEnM5MuW3/0XsvQ7Ybr/35AQq2bqIABuD0Ij5p9WgUx3qrfVv/SaS3UvC6Nk4jQ2zqnWbWc7C5W4F2/x7H3p8QdIj1XAtCgwoIJFxRFAG0mlvQlc3Hm8Y2cRxc27WlLwbAaKEuhaxpoELSWLbmSsW5GXmmRbyLFrp+xxHgD3Q6sxYkzjOl7EAETLR/UY+9oqGaeWLvC0W8qxuEW2JOJ1n+3UlmD6D6o/Cxc6QzmIrIyKKwkhbX3YX0UEAJhnOUPH0AyflsGyh+JhFmPHxm9ZOEwS5bu8SLyYpKD7bvPrXmjQsr/bQyvqXzgNUV+TcTTH2zZgeAyAYJkhWGIFSnV0Iq3EHXLyk3i5nOPswzwWMrFSC47v4TH8/C5bBKpG79+ah/V3xnCf4z6AnsA/mLGm+3Ys0PBtkWbPD/1T3/2u3q/rYF/fzd7d0WzYeGk9xm6ZJsAn3ZM3w8zAWKT7oJDcpDeoa7kZqFrJvYW3WJlHqJg+Qk4ee5KC6dFGXpCgMZzIOa3IbdOJTHYkJ5ZkWtlWAgPNmlyEgLxJ9fBJr+fnD6hCRU0dkvyrh5eL3W5ISKVoIWx9PcmKwDePh7b4aJ/W8PGDiTMlBz7UbHQOzWAJ/m9NzGfE+xyaKeh2yfz1uHqzoPQPOmjbXH3U9uyYyyaDC+zWtc/8M8y1y4y0OE7A4cRT7rpqKkNjBWcLTzo2xj0pH/eWpaXzCH5uKWuWw1HgQ6HoaCR8m9Cgk8yGZ3XOSk1W9L9mhP9tbmcQ/c0SS6V6ZPxgN1qKOPHmGd8FNN5z1t3UV+EJrvzSg5+fbjyVtZh/WJBl3ZD9
```

tssqvW7dOhw4dqqGhoTpy5Mib/h7kBhs2bND33nvPIYQXKlRIn332Wa1fv77abDbds2eP/vzzz1q7dm0NCAiwZti9cuWK9uvXT
202m4aGhupnn33mcN2afT+cO3euvvTSS9qhQwf18vLSNm3acPx6A0LWHXLy5El94oknMt1EEMiJbtYj9Ouvv1o3zs7Kzp071cf
Hx5p6d8CAAVq0aFH9448/rDInT57UsLAw7d+/v6qqRkREaNmyZXXp0qV6+vRpPXbsmI4ZM0YDAwN1w4YNqqoON93Mzew9CNcfU
CcmJuqUKVM0ODhY27Ztq5999pl1j50DBw5oWlqanjnp1SkeMGKEVKlRQm82mDz74oPWj+tVXX6mLi4s1+5+97ZOSknTs2LHatm1
bnTRp0j05M9Td6J96IyZPnqxVqlSxDuJnzpypjRo1criWd8OGDRoUFGQNzZ00aZI1TCgrJ0+1CZNmmiZMmU0KChIPTw8tH379
lneHDS3TruelpZ2y6H99vdkzpw5WrduXeuat19++UW9vb2tm9/aBQcHa+/evfXq1at64MABDQgI0CpVqujrr7+ubdq00SFDhuj
777+f5bDMOzkUOqtbPahmbFelSpWs/dDum2++0aCgIB00aJAOHz7cGhJYt25dHT58uKr+30H8smXLtESJEvrrr79qYmKiDhgwQ
OvXr6+qGb3s3bp105IlS2qVKlWs34Ib973cNGvshQsXdPjw4VqsWDEtU6aMPvroozp79my9fPmyLl++XG02m+bPn18HDhxonRT
5+++/9dlnn9WyZcs6rOuDDz5QX1/fTPdaXbt2rU6ePFlTUlL0559/1kcffVT79u2r27Ztu2PbmZMQsu6gQYMGaUAJY3yAPuB
bt27VKbzaarVq3SyMhIHTVqlMOsmpGRkVqpUiXr3iU//vij5smTR3fs2OFwZr5Lly7apk0bTU5O1osXL+pzzz2n5cuXtyZRaNC
gga5YsSLTj+e9OLb+Rjc7YFi5cqXabDaHm2zeu3ZNe/TooodWrV9dp06bp8OHDtW7duta4edWMC5zr16+voaGhOnfuXN2+fbvmz
ZvXulfJmTNnHO7Td/1BS245eLkbbd++XqaMTJkxQVdXVq1drjRo1dObMmdZzU1JStG3JStG3bttq
hQwdNTEzU3377TQsVKpTpgD0qKsqahOHChQsaGRmp48ePt24hgqwdPXrUuhefnf2zcurUKa1Xr57DjKg1atTQAQMGOEwM8sYbb
2hwcLB1/6E9e/boyJEj9aGHHtLevXtnCi/Odv78eZ2zZo1DvWw2m44bN05feOEFLVasmB4+fFhXrlypAQEBWr58eT18+LT18+LCqZkw
R3qVLFw0NDVVVdbgljbu7u37zzTeqmjG07cEHH7QCqmrGRA0//WaNWqUBgcH67fffquXL1/WEydOWJN+nD59W
j/77DO12WzWfevs78+HH36o5cqVs2aVXvZq25Nm6d+9enTBhgtapU0f79euXKyex+C8IWXfIhQsXdNq0aRkTX1/dFe/tL1d3cVzUBgcM4hWx/cTxgOAsIWaw2m9L8fgv7czLyMj8Y0l09FC5AsR4ayWZCbYp
OmJd5swWSMhEZrFxw1Z5iJ4R2yp4kg8HCTBGRQL4MOYPXGU0UgGLOPEXBcyVtcJoyMqyxS6cdROhrFbO6wQKUPHCLJrdvYs5vDnOc
3w5M6HtDubHrGg9tT4R8ndMOdMZQfaMHmlCPQmxRq2QFGyOXQPBbHX8BcXmbPVnsMOaKuSr9hWK/ZZ2h8Cvkhlrbcnx/sFhcpBi05

tssqvW7dOhw4dqqGhoTpy5Mib/h7kBhs2bND33nvPIYQXKlRIn332Wa1fv77abDbds2eP/vzzz1q7dm0NCAiwZti9cuWK9uvXT
202m4aGhupnn33mcN2afT+cO3euvvTSS9qhQwf18vLSNm3acPx6A0LWHXLy5El94oknMt1EEMiJbtYj9Ouvv1o3zs7Kzp071cf
Hx5p6d8CAAVq0aFH9448/rDInT57UsLAw7d+/v6qqRkREaNmyZXXp0qV6+vRpPXbsmI4ZM0YDAwN1w4YNqqoON93Mzew9CNcfU
CcmJuqUKVM0ODhY27Ztq5999pl1j50DBw5oWlqanjnp1SkeMGKEVKlRQm82mDz74oPWj+tVXX6mLi4s1+5+97ZOSknTs2LHatm1
bnTRp0j05M9Td6J96IyZPnqxVqlSxDuJnzpypjRo1criWd8OGDRoUFGQNzZ00aZI1TCgrJ0+e1CZNmmiZMmU0KChIPTw8tH379
lneHDS3TruelpZ2y6H99vdkzpw5WrduXeuat19++UW9vb2tm9/aBQcHa+/evfXq1at64MABDQgI0CpVqujrr7+ubdq00SFDhuj
777+f5bDMOzkUOqtbPahmbFelSpWs/dDum2++0aCgIB00aJAOHz7cGhJYt25dHT58uKr+30H8smXLtESJEvrrr79qYmKiDhgwQ
OvXr6+qGb3s3bp105IlS2qVKlWs34Ib973cNGvshQsXdPjw4VqsWDEtU6aMPvroozp79my9fPmyLl++XG02m+bPn18HDhxonRT
5+++/9dlnn9WyZcs6rOuDDz5QX1/fTPdaXbt2rU6ePFlTUlL0559/1kcffVT79u2r27Ztu2PbmZMQsu6gQYMGaUAJY3yAPuB
bt27VKbzaarVq3SyMhIHTVqlMOsmpGRkVqpUiXr3iU//vij5smTR3fs2OFwZr5Lly7apk0bTU5O1osXL+pzzz2n5cuXtyZRaNC
gga5YsSLTj+e9OLb+Rjc7YFi5cqXabDaHm2zeu3ZNe/TooodWrV9dp06bp8OHDtW7duta4edWMC5zr16+voaGhOnfuXN2+fbvmz
ZvXulfJmTNnHO7Td/1BS245eLkbbd++XYcOHapdu3bVd9991/pMQZYsWX/MjJtdNjlH4b1C3
hQwdNTEzU3377TQsVKpTpgD0qKsqahOHChQsaGRmp48ePt24hgqwdPXrUuhefnf2zcurUKa1Xr57DjKg1atTQAQMGOEwM8sYbb
2hwcLB1/6E9e/boyJEj9aGHHtLevXtnCi/Odv78eV2zZo1DvWw2m44bN05feOEFLVasmB4+fFhXrlypAQEBWr58eT18+LCqZkw
R3qVLFw0NDVVVdbgljbu7u37zzTeqmjG07cEHH7QCqmrGRA0//fSTVd7ubnpnQ404aNWqUBgcH67fffquXL1/WEydOWJN+nD59W
j/77DO12WzWfevs78+HH36o5cqVs2aVXvZq25Nm6d+9enTBhgtapU0f79euXKyex+C8IWXfQ1q1bdfXq1bn

localhost:8888/files/task 2 stock predection.py 13/27

```
D+XtbU1rK2t+7sZDNyTxRj7H/Fwj/5TWlqKlpYWvPDCC0p3I/2woF8p+Z2fn4+ZM2fCwsICfn5++Oijj2BoaIitW7fCysoKcXF
xyMjIgKqqKjw9PTF9+nT4+/tj8ODBv2vbd373pFIpRCKR0n8W5D0loaGhqK+vh5aWFoyMjJCeng5dXV0sXLgQdXV1SEhIgI2ND
UpKSuDq6oobN27AyckJAQEBSE1Nhbq6OjIzMxEZGYn3338fsbGxyM/PR3JyMoDe4YTR0dHQ1dUVtk29ue5KdQ4OHDiAuro6ODs
7w87Orr+bw9hDQ3muAoyxP4X8Jk8qlfZzS5SPi4sL/P39leqG7mEhn0dKJBL1GWBdv34dq1evRkBAAEpLS/HPf/4T27ZtQ11dH
Q4ePAhVVVWYmJjA2NgYOTk52LlzJ1577TUMHjwYDQ0NWL9+PW7dunXP7dP/z2El7ykGer+LyvRZkAc0d5L3HIWFhaGhoQFz585
FU1MTUlJSAAAXL16EmZkZrK2tYWpqiry8PABAW1sbBg4ciCeeeALq6upob29HeXk5jIyMsGTJEgBAYGAgdu/ejcrKSrz77rvQ1
dVVaIMyBrnPPfccZs2aJQRYPT09fZ4XxpSNcl0JGGN/Gu7NYo+72x8kyIeB1dbWIiEhATExMQp5O6qqqvj2228xbdo0AL2Tt/r5
+SEgIACbNm1Cd3c3xo8fDyMjI2RlZeHMmTPo6upCXV0d0tPTsXnzZnR0dNyzLfJA4l5BnjKQF45oa2sD8EuREfnxeOml9DZ2Y
mmpiYsWrQIhYWF2LBhA1pbW9HZ2QkAmDJlCrZs2YKrV69i8ODB8PX1xbJlyzBjxgyMHTsWV65cwbZt2/DNN98A6D3vBgYGICIh
mHichwP+XrcHvKqqqnxMGAMHWYwxxtg9yWQyyGQyAFCo9nfp0iXMmDEDXl5eqKqqQkNDA/z9/ZGZmQmgt/iIubk5ysvLAUDolZ
oxYwaOHj2KyspK2NjY4F//+hfq6+sRFBQEDw8P2NraorKyEvPmzVPqvLqzZ8/irbfeEnqZ5Mfv9h4SqVSKf/zjH3jzzTfR3d2t
cGPf09MDAAgNDcXnn38OV1dXLF68GJmZmdi7d68QZP31r3/F+fPncfToUWhpaWHhwoVYunQpenp6EB0djdzcXDg6OsLNzQ3AL/
lt8twiDiZ6PU6VEhm7Xzj7kDHGGLuH24d+7d69G/Hx8UhPT8ewYcNgamqKXbt2wcHBAQCQnJyMjz/+GGPHjoWFhQXc3Nywc+dO
JCQkCFW+1NTUcOvWLRw8eBC2traws7NDWVkZysvLUV9fD39/f6HSoDJrb2/H0aNH0dPTg5CQEKirqwP4pQdPno/2/fffw9bWFm
pqagpVFOX/RkVFwdXVFSUlJQgNDQURIT4+HuPHj4dUKoWpqSkkEgk+/fRT+Pj4QENDA9HR0Qpt+bUy/Iwxdi8cZDHGGGPo7f1Q
UVFRuJnu6OhASkoKhg8fjtOnT8PT0xNWVlZQUVHBnDlzoK+vj+3btyMzMxMHDhyAqqoqiouLYW9vj6CgIOTm5iI3Nxcvv/wyND
Q0UFxcDENDQ2zbtg2zZ8/GwIED8eSTT8LHx0fYpjzfSxmHAcrPwahRo+Du7o7Kyko0NDRARUUF48aNQ2hoKBYuXCgcGwsLC1y8
eBGAYhAkD7KcnZ0hkUiQn58PZ2dnvPLKK3j55ZcVKtzl5uZCR0dHCOQAxYImHFwxxv4IHi7IGGNMqd2ZS3Ljxg3hvba2NlRXVy
MmJgYDBw7EqlWrMGzYMAwZMgT6+vpYsGABkpOTYWNjgzNnzmDs2LEoKChAU1MTJk6ciIiICERFRWHcuHGwtLTEuXPnsHbtWhw7
dgzd3d13tUNemU4ZAyzgl3Nw8+ZN2NjYQCQS4cCBAzA0NERMTAy++OILrFq1CkBvafaWlhaYmZkJlRVvJx8yOGvWLGzbtk0Ixl
RVVRWGHbq6uirMowUod64bY+z+4CCLMcaYUpH3FAG/zN1z/fp1LF++HGPGjMFrr72Gd999FwAgkUgQEhICsViMsLAw4f8AwOHD
h5GXl4e5c+figw8+gEQigYWFBU6ePIkffvgBAJCWloZvvvkGzs7OiIuLw4YNG9Dd3Q11dXVcunRJoV3KlNdyZ/U5+d87d+7EiB
Ej4OLigoKCAtTU1KC4uBhAb/5UZGQklixZgsLCQqipqaG5uRlAb76cPHdOTh4kzZo1C6dPn0ZjY6PwXl/z+THG2P3EwwUZY4wp
ldvzrOQ322+99RYqKiowbdo06OrqIjExEZ2dnUhISIC3tzd0dXVRWFgIOzs73Lp1CxoaGujs7ERLS4swEXR1dTV++uknnDt3Dt
9++y38/Pygrq4OV1dXuLq6AujNNfrkk0/w0ksvYdSoUQp5RMpAnkt1e5AjD3Tb29uRnJwMf39/JCQkoLy8HKdPn0ZpaSmqqqpg
Z2eH+Ph4nDlzBsnJyejq6oKzs7MQPN0ZOIlEIshkMpiamqK1tRXa2tr3bJeyBLeMsQdHea7sjDHGHlvyHJo79TV/m1QqRWRkJH
bv3g2gt/ekpKQEGzduRExMDKZNm4aJEyciOzsb+/btg5mZGQIDA7FhwwYAvcUrAGD8+PF48PPkZiSlTpiAgIABRUVHYuHEj
IiMjhRyfxsZGfPLJJ5gwYQKMjY1x5coVREVFQSwWP1YB1r3Owe3kvUtHjx7F9u3bcePGDSHAOXDgABobGzF58mQYGRlh4sSJwq
TAe/fuFdaRlpYGBwcHREdHo6SkBLa2tpDJZH0GSmKxGEQEbW3t32wbY4zdT4/P1Z0xxpjSkk8C29XVBeDuOZPk5MUM8vLycOrU
KQDAoUOHYGxsDJlMhsmTJ0NPTw/ffPMNYmNj4eHhARUVFfj7+6O2thanT5+GWCwWSop//fXCA0NhVgsRnp6OiZPnoxXXnkFxs
bGwjYNDAxgYGAABwcH7Nu3D9999x3c3d0fxGF5IOTBi/wcNDc349ChQ0KZ9Nvl5+fD1NQUQUFBiI+Ph5+fn1CmvampCaqqjAz
MxOW9/T0hKGhIY4cOSLkWOno6GD58uWwsbHB4cOHAfQGU/eaEF0efD1OAS1j7OHHVxzGGGOPvLa2nDHwwONjY3CjXVxcT
HmzZsn5EiJxWLU19fDxsZGCA5sbW2xa9cueHh4QF1dHRs3bkRtbS0SExMhkUgAAI6OjjKLsjXb2Njg8TERKxfvx5BQUFC
e27P8RGJRAgJCChQvh6Oj45x+MB0we4GRkZMDe3h6Wlpb429/+Bm9vb5SVlQHoDcs5vx0UcfYcKECWhsbMT69ethZWF8P
BwdHR0YNy4cWhoaEBdXZ2wbh0dHTz55JMoKysT1nXr1ii1oampiZIlcHFxEYYLcqEKxtjtDhIMsxhhjj7zz589DW1sbc3NSE5O
xunTp4X3KioqMH/+fOFviUSCgZZGQEPT09pKWlYd26dXj++eehrq6OxsZGrF27PfXmlTAyMoKnpyeqq6sB/DJ0kdDJ
4mkyn0pChTjk9eXh709fWRkpKC2bNNn48cff0RWVhYGDBiApKQkAL2L2B2MmTJ1FcXIy3oqbs8cHyVbqtXbnTu3DuU5YnfmVz3LI
0dERGRkZQkGL9vZZ2XLp0Ce3t7cJwTXn5dT9T09NH9s+OMMfYrOMhijjDYMH2yJL3GDU2NuLGjRvIy8uTFgrqxDZzs3ERcXBwDw8f
HBunXrUFFRgdTUVDQ3N0NdXR1aWlpoaGgAADg5OWHChAlYvHgxvvvzySzQ1NaGiogJJSUnYunUrxGIx1NXVsXTpUiGP607KXHZd
Q0MD5ubmSEpKwrx58zB8+HB4enrCz88PP//8s7Dc5cuXYWBgIAy17OrqgoaKDg7F9+3YAQFJSEn744QeEhoaioKAACxYsgJ
GREWbOnHlXoZCrV6+iqalJmOiZMcYeJhxkmcYYe2TJgoIjU1FTt27EB2djak
UikkEglu3rwJABg0aBBWrFgBd3d3LF68GH5+fnB3d0dLSwuSkpJgZWUFANDS0gIALqBwBw8PD4wYMQLffvut8NqxY8ewdetWxM
XFob29HUDv8TM3NxcCVXlQ6ujoiAsXgCAsXLgCAUHBEW1sbkZGRqKmpQXx8PJYvX46//OUVOUvQoBVXV0Nc3NzGBoawsPD40HuLmOM/S4i
4skhGGOMPeL279+P1NRULF26FM7Ozjh06BA++OOAD2NvbY9GiRejo6MBXX32FmJgY7Nq1C2+88QZWrlyJl156SaF35Tp02hoaI
Cnp+ddQwLZvS1btgxbtmzB008/jdLSUjQ2NmLEiBFCuftNmzZh6NChePPNN1FTU4PS0lIAvRMG+/j4wMbGBh9//DFUVFQgFovR
3t6OgQMHKvQOyoNbsViM7u5uFBUV4cUXX+yX/X8/WMsd/CPVmMMcYeeVKpFNeuXXYOzszPa2tpQVFSEvXv3YnR5dtSHl5eaivr0d/dSUhI0NDNDQwb948
6OF154AWlpaTh37pzCZMDy540WFhbw9fWFmpoapFIp9r9r9Tr6+vpDJZNi/z8WLVqEU6dOoaSkBHl5eaivr0dSUhI0NDQwb948
XLx4ES4uLnj//cRFhaGpqYmTJ8+HWpqakKwO2jQIKioqEAqlQq5breXvFdVdVeUAizH2UOMgizHG2HG2CPPxMQEZWVlwsOzw5AhQ7
Bt2zZkZWUhLCwMycnJUWUrxjARwlAmzbaGcn7J+h08gtPmrIZ2Z2zMgX0fI2zMzGQAICIpSV8fX1x
6tQptLW1wdHREfn5+XjhhRewZcsWaGhoIC8vD+7u7uhrYI2KikqfuW7KVFiEMfZo4uGCGjDHGnnyea4nZ2qyzBfPuZtlcAZqxh4jTECGKlZfXW7KVXFPVfD0P7NnsJkvgaxh38MbgmNPPWqSTKvs0
KFC1i0aBGysrJw9uxZDB8+HO3t7Rg0aFB/N/mxk5mZiU8//lRQLFixAcHAwAKC1tRVBQUEwMDDA5s2bOWhljCkNvtoxtoywxxwh75vMKmr
q6OtrQ2pqakYNWqUHHO0NAQaWlp+Oqrr2BoaAgAQoAln9yW3R9eG0aNHAQAbNmzApEmTcO3aNbz99tt9tt9Blg8JJMx9r
jinizGGGOPvPLycri7u6Ourg4SiQRExEPK+kFkZCQ+++wzSKVS6OjoYPbs2YiOjsbQoUP7u2mMMfZqfZ3AxhjjLH/VUtLC5yc
nHDt2jVIJBIOsPpJYGAhg0bhsDAQDg6Ogqvc9DLGFM23JPFGGPskSeVSpV2MuCHcDLRUVDjAYowpHQ6yGGOMPTY42Op/RA
Qi4iIXjDGlxkEWY4wxxhhjjN1H/JiJMcYYY4wxxu4jDrIYY4wxxhhj7D7iIIsxxhhjjDHG7iMOshhjjDHGGPsPuIgizHGGGOM
McbuIw6yGGOMMcYYY+w+4iCLMcYYY4wxxu4jDrIYY4wxxhhj7D7iIIsxxhhjjDHG7qP/Azup5J5ckxEEAAAAAElFTkSuQmCC",
       "text/plain": [
        "<Figure size 1000x1000 with 1 Axes>"
       ]
      },
      "metadata": {},
      "output_type": "display_data"
     }
    ],
    "source": [
     "df.boxplot(figsize=(10,10), rot=20);"
    ]
   },
   {
```

```
   "cell_type": "code",
   "execution_count": 30,
   "id": "f849bc72",
   "metadata": {},
   "outputs": [],
   "source": [
    "pd.set_option('display.max_columns', 50)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 31,
   "id": "09f6340e",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Age', ylabel='Count'>"
      ]
     },
     "execution_count": 31,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Age\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 35,
   "id": "f91534d3",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Gender\n",
       "Male      7622\n",
       "Female    7378\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 35,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df[\"Gender\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 37,
   "id": "2f7a98c1",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "0.249\n"
     ]
    },
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Age', ylabel='Count'>"
      ]
     },
     "execution_count": 37,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "print(df[\"Hypertension\"].mean())\n",
    "sns.histplot(df[\"Hypertension\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 38,
   "id": "641960e8",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Hypertension', ylabel='Age'>"
      ]
     },
     "execution_count": 38,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Hypertension\"],y=df[\"Age\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 40,
   "id": "2294937a",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "0.5029333333333333\n"
     ]
    },
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Hypertension', ylabel='Age'>"
      ]
     },
     "execution_count": 40,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "print(df[\"Heart Disease\"].mean())\n",
    "sns.histplot(df[\"Heart Disease\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 41,
   "id": "a0240e8d",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Heart Disease', ylabel='Age'>"
      ]
     },
     "execution_count": 41,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Heart Disease\"],y=df[\"Age\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 42,
   "id": "963db780",
   "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>Hypertension</th>\n",
       "      <th>0</th>\n",
       "      <th>1</th>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Heart Disease</th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>0</th>\n",
       "      <td>5607</td>\n",
       "      <td>1849</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>1</th>\n",
       "      <td>5658</td>\n",
       "      <td>1886</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "Hypertension      0     1\n",
       "Heart Disease             \n",
       "0              5607  1849\n",
       "1              5658  1886"
      ]
     },
     "execution_count": 42,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "pd.crosstab(df[\"Heart Disease\"],df[\"Hypertension\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 44,
   "id": "87dc91e6",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Marital Status\n",
       "Single      5156\n",
       "Divorced    4980\n",
       "Married     4864\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 44,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df[\"Marital Status\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 45,
   "id": "242e7027",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Marital Status', ylabel='Age'>"
      ]
     },
     "execution_count": 45,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Marital Status\"],y=df[\"Age\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 46,
   "id": "544e6d5d",
   "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>Hypertension</th>\n",
       "      <th>0</th>\n",
       "      <th>1</th>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Marital Status</th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>Divorced</th>\n",
       "      <td>3735</td>\n",
       "      <td>1245</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Married</th>\n",
       "      <td>3653</td>\n",
       "      <td>1211</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Single</th>\n",
       "      <td>3877</td>\n",
       "      <td>1279</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "Hypertension       0     1\n",
       "Marital Status            \n",
       "Divorced        3735  1245\n",
       "Married         3653  1211\n",
       "Single          3877  1279"
      ]
     },
     "execution_count": 46,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "pd.crosstab(df[\"Marital Status\"],df[\"Hypertension\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 47,
   "id": "5f2f267a",
   "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>Gender</th>\n",
         "      <th>Female</th>\n",
         "      <th>Male</th>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>Marital Status</th>\n",
         "      <th></th>\n",
         "      <th></th>\n",
         "    </tr>\n",
         "  </thead>\n",
         "  <tbody>\n",
         "    <tr>\n",
         "      <th>Divorced</th>\n",
         "      <td>2429</td>\n",
         "      <td>2551</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>Married</th>\n",
         "      <td>2439</td>\n",
         "      <td>2425</td>\n",
         "    </tr>\n",
         "    <tr>\n",
         "      <th>Single</th>\n",
         "      <td>2510</td>\n",
         "      <td>2646</td>\n",
         "    </tr>\n",
         "  </tbody>\n",
         "</table>\n",
         "</div>"
        ],
        "text/plain": [
         "Gender          Female  Male\n",
         "Marital Status              \n",
         "Divorced          2429  2551\n",
         "Married           2439  2425\n",
         "Single            2510  2646"
        ]
       },
       "execution_count": 47,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "pd.crosstab(df[\"Marital Status\"],df[\"Gender\"])"
     ]
    },
    {
```

```
   "cell_type": "code",
   "execution_count": 49,
   "id": "4ec57771",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Work Type\n",
       "Private           3863\n",
       "Self-employed     3855\n",
       "Government Job    3710\n",
       "Never Worked      3572\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 49,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df[\"Work Type\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 50,
   "id": "194e724b",
   "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>Work Type</th>\n",
       "      <th>Government Job</th>\n",
       "      <th>Never Worked</th>\n",
       "      <th>Private</th>\n",
       "      <th>Self-employed</th>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Marital Status</th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>Divorced</th>\n",
       "      <td>1229</td>\n",
       "      <td>1190</td>\n",
       "      <td>1315</td>\n",
       "      <td>1246</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Married</th>\n",
       "      <td>1234</td>\n",
       "      <td>1143</td>\n",
       "      <td>1243</td>\n",
       "      <td>1244</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Single</th>\n",
       "      <td>1247</td>\n",
       "      <td>1239</td>\n",
       "      <td>1305</td>\n",
       "      <td>1365</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "Work Type       Government Job  Never Worked  Private  Self-employed\n",
       "Marital Status                                                      \n",
       "Divorced                  1229          1190     1315           1246\n",
       "Married                   1234          1143     1243           1244\n",
       "Single                    1247          1239     1305           1365"
      ]
     },
     "execution_count": 50,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "pd.crosstab(df[\"Marital Status\"],df[\"Work Type\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 58,
   "id": "1188ccd1",
   "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>Work Type</th>\n",
       "      <th>Government Job</th>\n",
       "      <th>Never Worked</th>\n",
       "      <th>Private</th>\n",
       "      <th>Self-employed</th>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Gender</th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "      <th></th>\n",
       "    </tr>\n",
       "  </thead>\n",
       "  <tbody>\n",
       "    <tr>\n",
       "      <th>Female</th>\n",
       "      <td>1812</td>\n",
       "      <td>1788</td>\n",
       "      <td>1895</td>\n",
       "      <td>1883</td>\n",
       "    </tr>\n",
       "    <tr>\n",
       "      <th>Male</th>\n",
       "      <td>1898</td>\n",
       "      <td>1784</td>\n",
       "      <td>1968</td>\n",
       "      <td>1972</td>\n",
       "    </tr>\n",
       "  </tbody>\n",
       "</table>\n",
       "</div>"
      ],
      "text/plain": [
       "Work Type  Government Job  Never Worked  Private  Self-employed\n",
       "Gender                                                         \n",
       "Female               1812          1788     1895           1883\n",
       "Male                 1898          1784     1968           1972"
      ]
     },
     "execution_count": 58,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "pd.crosstab(df[\"Gender\"],df[\"Work Type\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 53,
   "id": "46dfbc57",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Residence Type\n",
       "Rural    7529\n",
       "Urban    7471\n",
       "Name: count, dtype: int64"
      ]
     },
     "execution_count": 53,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "df[\"Residence Type\"].value_counts()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 59,
   "id": "00bebdd8",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Marital Status', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 59,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Marital Status\"],y=df[\"Average Glucose Level\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 55,
   "id": "47de0667",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Marital Status', ylabel='Age'>"
      ]
     },
     "execution_count": 55,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Average Glucose Level\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 57,
   "id": "2dbecf82",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Hypertension', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 57,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Hypertension\"],y=df[\"Average Glucose Level\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 60,
   "id": "eaabb0c4",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Heart Disease', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 60,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Heart Disease\"],y=df[\"Average Glucose Level\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 61,
   "id": "0d487d09",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Heart Disease', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 61,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Body Mass Index (BMI)\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 62,
   "id": "abb73d92",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Heart Disease', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 62,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Body Mass Index (BMI)\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 63,
   "id": "3cba8d45",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Heart Disease', ylabel='Average Glucose Level'>"
      ]
     },
     "execution_count": 63,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Body Mass Index (BMI)\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 64,
   "id": "c9cf4194",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Work Type', ylabel='Stress Levels'>"
      ]
     },
     "execution_count": 64,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.boxplot(x=df[\"Work Type\"],y=df[\"Stress Levels\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 67,
   "id": "b5ed8800",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "<Axes: xlabel='Work Type', ylabel='Stress Levels'>"
      ]
     },
     "execution_count": 67,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "sns.histplot(df[\"Stress Levels\"])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "fee46c93",
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.11.5"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```