```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 12,
   "id": "bbe0b8f7",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.preprocessing import LabelEncoder\n",
    "from keras.models import Model\n",
    "from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding\n",
    "from keras.optimizers import RMSprop\n",
    "from keras.preprocessing.text import Tokenizer\n",
    "from keras_preprocessing import sequence\n",
    "from keras.utils import to_categorical\n",
    "from keras.callbacks import EarlyStopping\n",
    "from keras.models import load_model\n",
    "%matplotlib inline"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "2e9ce340",
   "metadata": {},
  "outputs": [],
  "source": []
 },
 {
  "cell_type": "code",
  "execution_count": 14,
  "id": "78a9b584",
  "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>v1</th>\n",
"      <th>v2</th>\n",
"      <th>Unnamed: 2</th>\n",
"      <th>Unnamed: 3</th>\n",
"      <th>Unnamed: 4</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>ham</td>\n",
"      <td>Go until jurong point, crazy.. Available only ...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>ham</td>\n",
"      <td>Ok lar... Joking wif u oni...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>spam</td>\n",
"      <td>Free entry in 2 a wkly comp to win FA Cup fina...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>ham</td>\n",
"      <td>U dun say so early hor... U c already then say...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>ham</td>\n",
"      <td>Nah I don't think he goes to usf, he lives aro...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"     v1                                                 v2 Unnamed: 2  \\\n",
"0   ham  Go until jurong point, crazy.. Available only ...        NaN   \n",
"1   ham                      Ok lar... Joking wif u oni...        NaN   \n",
"2  spam  Free entry in 2 a wkly comp to win FA Cup fina...        NaN   \n",
"3   ham  U dun say so early hor... U c already then say...        NaN   \n",
"4   ham  Nah I don't think he goes to usf, he lives aro...        NaN   \n",
"\n",
"  Unnamed: 3 Unnamed: 4  \n",
"0        NaN        NaN  \n",
"1        NaN        NaN  \n",
     "2       NaN       NaN  \n",
     "3       NaN       NaN  \n",
     "4       NaN       NaN  "
    ]
   },
   "execution_count": 14,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df = pd.read_csv('spam.csv',delimiter=',',encoding='latin-1')\n",
  "df.head()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 15,
 "id": "e8514131",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
    "RangeIndex: 5572 entries, 0 to 5571\n",
    "Data columns (total 2 columns):\n",
    " #   Column  Non-Null Count  Dtype \n",
    "---  ------  --------------  ----- \n",
    " 0   v1      5572 non-null   object\n",
      " 1   v2      5572 non-null   object\n",
      "dtypes: object(2)\n",
      "memory usage: 87.2+ KB\n"
     ]
    }
   ],
   "source": [
    "df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)\n",
    "df.info()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 16,
   "id": "8783fcef",
   "metadata": {},
   "outputs": [
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "C:\\Users\\sathi\\anaconda3\\lib\\site-packages\\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.\n",
      "  warnings.warn(\n"
     ]
    },
    {
     "data": {
      "text/plain": [
       "Text(0.5, 1.0, 'Number of ham and spam messages')"
     ]
    },
    "execution_count": 16,
    "metadata": {},
    "output_type": "execute_result"
   },
   {
    "data": {
     "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYsAAAEWCAYAAACXGLsWAAAAOXRFWHRTb2Z0d2FyZQBNYXRw
bG90bGliIHZlcnNpb24zLjUuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/YYfK9AAAACXBIWXMAAAsT
AAALEwEAmpwYAAAZwklEQVR4nO3de7RdZX3u8e9DQECBAiUgJGioxVbAKzFitZV6I9VaGO3B4pEaKxr
LodWeYVWw5yhqGdLq0apVas01VrTC7WIYusRiaFeIiAlA5DERBKQqxeO4O/8Md+UyWbvPXcga
++d7O9njDXWnO+c71zvnGvt9az5zstOVSFJ0mR2m+kGSJJmP8NCkjTIsJAkDDTIsJEmDDAtJ0iDDQpI0yLD
QlCVZkeQPZui1k+RDSW5Nsmac6S9P4T4JBtnuh3SWIbFTizJDUluSvKIXtWcCz
wMWVtWSmW6MNNcYFju/3YHXztQtlSeSedtZ5dHHADVX1vVG0/R9LkdIud3zuB30/9uy/9gJSRYlqSS798ou
TfLKNvzvzyJP83yXuXMSS3JbkuiQ/18o7Nck+TFvWkrkpyX5J+4xXHae1iSPu1jmfvTnTJT
kBUn+s7XnTZO3Oe7AsyY1Jbk7y+73pe7dtc2uS/wGV
+5ryU5vjft5e39ubO9hy9t5T/d1un2tg3/ZorrtneSlW37Xt3e44296Ycl+USSre31XtObiTJ2rbcm5K8e6Jt
MCdUlY+d9AHcADwXX+DvgD1rZK4FL2/AAOOwSgae+yLc
+C+06e8FvtCmPQLY0Ja1O/AU4GbbgF7d24Fn0P1I2HmfGbgxgFx4EfAq9q6n
A5sAtKmvxB4DBDgWcD3gae0ace37fBmYI+2jK3AR4F9gaOBHwI/NcFrHw88POAd/DuwN
PBG4G3hcm34u8O/AgcDhwDeAjRO8zgAFcD+bT0eBxw69N606acCP9nem9cB39n2HgBnt/U/U7oU2/AL
ge+P3e9rh+4HP4JeAQYAGwBfgP4MmtLZ8F3tL3tfH5dJVfN4aeKWnL2ymzOcyzbB/A4KbDyN7uymCtyPpb1pL7/D4X+/gXfbcvVqHrDYa5y8ONXb4t4xq/cBU1A6BfZYtAP7LjmQ3+O2OB/ARqIzYtt2Oq15pv0wrjXvTNZ2M/Uua/xDkmPtzXhQ7v6V/H94Ih1C/SsGwOWn8ie0LNmS/69S0a2+PSnKQPBpx/H5Ehk+0Gvqo6ecT0g34ESHLuvWtGeaBPJ8iRXUbBjoD0Op/ClBdFgeLGknSKmbMnu6iVnDW
KvhNqlO+4CwrdZaQ+Yqy7cyjK0p5PmHxmyY6Qe5XzeeDhn2oXDyZAfNTNfdFW1+/FaSHHw7cIknO9H6dM
uJMTfSDkG0Yh0IRNHXY1KIUrG7xzWi96M9TrMo8E0ok2hIp3KUaxjD+HDd1VAFqomWFmLZVSxTr4b2Mw7+
r9eEH/71twcxZF+1b9Kgzv0yc1LgSG4wAK8b1tWGPJDRM0mydqUJb5FDF4Y0TaqZQKXGJA7vXc9+X2+MUgqjxGgNCSsVhnbrkmOswYq1B03oSTEzrJLjobaBm0GrdzMtKsOXEMKbNXVKoL7dkt+VLxmcwJzrCtvZxs9S9s0T8pIo7KSSLVtiJmBdaABxoHhmSbHMrcmTDqgd6Hd94l0Hn19pOyxDBQ6yCJUm1WMKwkbmX9WxCy+IntiB6L0ZRpFBGoJ7HEEYvtDrXFfsxEmymZiGfk+8y4P6b8eWmRDmw1oVHO2NqGcuzatxkAxuvo+E3rSzzAkXWEMzSFaFo14aR9zNeRn75ysnl6nZRu6t5nOZ7/71OMqQoWBsqpxrLT+n1LFyLpyu+Kp0t3sx2nrJ/Cv30thmgFgGq5o8eT69m5R+3AXZhHsWgE6uSdQA/q5KmHkqqBaTQ7EwYtM7JHv4Lu1Vl36oSvJkgU/KgltUlm4kzphW7MZRCc3AVyGtHp54QKoTNNXx9ctZmwFTpJjF4fP1PmupNCLRszpHn0Z0o+kjn/D2KH2PwsSP6WqA1fZjKS6ZJ19u+FjhNnaPn5uS/+S6wnrNAmo+LmeMDjxQoc6Hw5Shq5R8C61fdX6hUttuhP3AmnYBTBPXCHuRSmQVKUJK6l+PGpZ0uJXQWbSNi24ap6HcpL6y6wyNN+nBU5dI/Scy2bMe/UXPBGfMS6J5DyTvEGNUpcnvB9fwYkD+IP63Dcn1Yh1Ee3KtMDY/jnwL54DRP1a6hxsF6O0mcL4PXxvd/FjlFnX0JQ2YVSugu9TpMsBu6c6H7SYE0NMrxsS/UxCzIE89fI5Z+8KpVOFoFg6xlTP6mCRbqwPBx3DT7kYP7/LmvvE6d8OqbrUwOgDyO+Yq4dGPWL1dWgGA0fEqDSk+A/BM5l9npM3ejQ9zgpiRQJAF4txMZPU0jkSx4f4bnMGJl+OVvgOjWbF5qbrDBx2qW2Um4S0UPNU1Ztl6R11fsoZnD3OXDtc0LBRffHdzt1Un1iX2oBS2u6CpeZ1rdQGjJ5KrGmT6b7DWCvRA9vkXWy/2DQs8fhEufAsNWhjgBcAOwDQNxV6S3ioHGpZVrmTGrX5MmrkQ+mqvW4j6KqetFnSl6tHTxtXcSFNQQQoxyUkSOpNwGnQ7bwUZdt2d5MCi6IO0DgH2s+3ZHGIuONTnSxyMGZmgGmyqp8uo/OVsMNz2E9b43vAj2QyW0/zzcqfYvjhwA4ICWqtNBt49DQHB5OMlvCHsBILwfcs4f3lM+VzK+zSl5AhxFetAPc86NoQi6AqZB5wM5A5TwHiJ1xvP8cZWm8lp/jiYLz9P7BN/ZWwVXyp1bQB8PVqxsZBO5B0b9fwc3tCyHQmCd8Kb0NNNZmH+2qJp6RGx3CmjJxfD4xvHjDK0dfyvHCX43d8ghABVXBeeAr1p3wyz8SwMdYpf+YaHf2IgUgUj8C/+F/+H25IdYRsqANqYzO5+VjkzH3Gu1Ftr9Ks2NHlGTXjwIWFQvwtxRjvqb1QSyp0wv+ouZSXFdHb4bZ3ihxc/VwrY6QdwXaSUPffHu3cR3JdPkyL3tJ1ShDE7UpDdYGOVfVSPtMebRgnN/9CZ7BVWbyH9z6AjZyE5XMJXDd3zWLNIJ5Xn2n0dW/CYdrKptTgpGOsGjdL9dNYBUh/X4HqiEv06GK+y75PZ5jKH8L0NkADdrbwZR2xLtWZ38WtC7qXh+Hwd/YM+5mQV5AqAxSqSBIKz1AVzcb6GoH38ZzHaWVgeD5EHz9qhyuHzDdhA1u3wzuhq9kd6XNUOlwGeRvgXm0XBRo+fPNtvCk7cv8nrc/IAwS7gFbq0kB/BfKVgCi6cErmEmsJRDF9U2kQY/SwOjOL3xf99eWBuMkzDSR8Jxl9jdEKRjM6dk6HEfbhpfx5+V8wwK0w4j/fkNH1mnXT0lI1lX2TFzqSMaYZhk5UvCzHr0qTu59g7Ws7h/WY1SV0kb9oYzmWA4lI2ZxGHq1X2kemRbDxBYgXrrlvD5eMXIbzj3O3oO41bkPMHQ0M/EwHK2Vd5ZFhCLGJoOqD+NVw8aEKmhD19rlY3jMEHfw4aQxBEmN6u9Wcsdku7ApkK1G8J/Y0A9VbV8Su7FbNTddbE1O8l2+ADuGXk0VyCMj+O5UHRTfNLZ5Mc3IbaG0mdp7VWhgW4vZyzsNfYoT46Z6JPvmjoZ2bAbnOY1UJFAN0QBxq1DAppvnkB8C2NwwJ/uBr3jfDUcCW8klIQJRt8lZLbdx/9t74IPzm28Xhi9IgvXjWJ9H6zuvI8ELQAeD/P0iBjXDlaPyUTSS4cSh6fbmhpZ3BmiBlXEp0bwkCLfMVWx55PDuJM6gtWaGH0rMBhgDXeGsD7O8cUfJvfajP8F3XnC2XsRU3KhfzS3aM5Pp2tAGyP2WSdtnU1NCgH/Jqq35oHl0g0l+cKgYSQiT8sHPoF7WJbpQWtnSj8Nx7uH7Ppq/WQ0Ue2WwKoYnhC2ow1Zl2r0M2AkixQTS8LpQW0WxDeQp6e23OmtmgOVnHHOe9KnXpbkUI6fxlwdgvIJMNjN/r9NDb+vXDZpzLh6/xs4KJrdQWyMJLnv/+v8zgNXZGNdMFeVKmbmdJQ+WpXd/HJxbMyvk7t0aSKH8BAbL1W/O4fJlG7iOIiGsfTBOaKGYaU/Jl1DKXd8hLG8Nqx/YVjZBxGGbUhFxhLv6HEbAhE6xhaCHPQoZF1Btv4ktcp8z1vYgMpIMnFzhqHazuwxhaAz5Ds3TaSd76M0fORIXZ0Ll4dbFo9IzA2DwSmgp3iajDD3Tdv1Km8jTc/+Vs2ynrwaTNj6WaLHMwtmBs3RxAw8y+5BTf5ud8HctMjtMYV+CN2OSFXoW+3l7ZEL0F04kLYUiGU2cO5uXwXFg5U3Ht1xSHZV9sS5J4bXbkh3pXzr9sTGMfs3OadJjqcbF8KHIoSV3IejX2FCgKF6y4aNDt3q3oWa+YN0xtJ9AL6e8jdSBS+7mxNcQGQWnW8xz+Hx2AQ8DtFdxyBt2nZO4EuBQkmTiBnFnEt5HzqlV1nuVX2tPLQ8cdK/wXknYPwgmEDNu5aXaXAUZxAxyJlvbwENJMoxUwvEs+MN08xhVvHWthyvjxAYZN+MEhtbzvcn/K5QQPkj23rmAezdxbn3X3kJ1+brKbWjJwtwDxz5RGUy8+xeTo3LRhgIMSmuqc1vAgk/KbRq8IuDcI/aQq6sGiIR+t1aWV6Hvu61IsI/6Hmv/lEfnRmbb7+n2oiwwcYM4qFXRaFeIaGQYyVwPavaf2xW9/MjSmFBFVfuGyyj2ZcBTSHy0NMzGYOBiIX9u0UDtEIlo2QoFG7pgLkKBOU4JlmTNmvbJOMQO+bTv/NzOOvFVZX9PuaG1l9uv5++8oP7XbzF/NPxOD3gJGL88LvgrIpKhVfDfdN6xgb2Oa0nCbyXYphfoJpCbn1eudNGakcXXU3DaW+sBZ5Kw5oxUW8uWiN8JbONTOxb7wyMd7rw4Dzhr6xH8CZrYvYhy8DSIzRX3iIhZ1kIc6xVOV7oblD3sGIiXrN8eoUrVj13gMWI25D1+dctVJ7F5fMd5zKw+VzfZqAbdW0EXVR31FIMRxEmGmVQINZxZBeGhcP9ZSMmn6K3onVx4XL/MPn8+7YQeOjA12Ac24sdWz4m99iEpcKp96KG3TJhwGO9M0kXkYVnYQgqAjPw4lgm+OzWZI3Mq03ClKMKWqtR3rr6IbVnJ/a34dC6K8N/sKRVySCTjiZQlEhmyHFdQFgEp5B3ZA5HcQ+C8g4Ju5RqAvSf4rWbMUIk1eGv6R0VKCqBVXuv+EapTa96/SEDuYe8HiqsF3Z9CqDZJ1tNDcq2gyDlOeo6ZSSzcnvvvFfJyv+3U2Eo3p/4YfwoRoXmnVRZXLNgjX0MF2Ql0ueEIPVVaGDcAkMSR5EwV3RQtMXd7E2PWm9ntj+NRbEQsh+u8w53CL84NRH9jnMmFAbB89aX9UPCQtvRfSQLmzDjm0hQADsPpCHR8KI7d5Umk49EIixbvA9YT/zCzdtSdZE0JNIwwxDyM9TKq8H/e1AV+tabPHbSR2KP6RsVZwzyrutCyyp1qwjbk3m4bPGfeuYu6E44o6Q4VNjLnXXNxJlQS0bQWPbLKMoJcAJcgnRTwoBh5aDx/lpeQUz1X+ncsWBmzKhxLuKE+X4LMw6d8z8k2tTmWE+cdaZKjKOvGa8jWG0CJ/xeN+p4zDzKvfuL39y+p7pJjXOgM/iJl+Oe9uKN+bScO9CHrBbm+fKJ+l0f+FZkJQJ+vbjrmAemp1S91iGzY1bW+KkXTKwGp66+2e2O2Bvq/4BRQC9aW8a6vO2h0+pCoqDzV13o+FwZaNJrq9FX9G0o/nmfmhZpXeIuhW6zT+GbTmk4ZDSu0MXX1EivEBwyoxu+WJy7T1Yjtw1zOo7smKz6Kx0vFy9HkMczm7xX/+zmPgb4JOkJ3SrPTlp1krjq8tCKUG7cAVOgGS5BZ1mPdOCVeFfMpgLgdrd6vmRJdWSufaJyKPbOimnX54F4YXoVL7pOw2vb8MaUQuA6NjJ1W1ikb1DcxqP2QlefavbkOo8uCHpAtjCF5lp71GDfnIpd6bfAx6CDgHjqaLY/n6oLg6tKutZjrsj6JUsBt1cbNdKcUEJcmu8zTyVHEE37NwHCOKhMX9FXaHwQ+5I+AQpnlZvpIEcyOKmOIbHZ1kUaEEjU8TR+ovHp+YA0mfqZlBoxqFdPoW4QeiSo4Yj1uDUqa41qc3Xk38rlbNftrUWSd57cqqxvQc2R7dbjZqHw1RA/f9PA+MVAR+6MiFAnu4QMbY+Sr8yc4GxnLx7+4FqAt67XBRqlHU+qeRXL6vQd1gG7txQGjpKPOHmF2Kaf2f7MxEi6mL3gm4whtMjYWnVFTz6aBGnpvjLGiGvSQ6zeD8o+e9U7+eiNjbr0/jNn/6rKaDG1C53e1kKxOWi6QV6DWpPXqXYaEw2lKnLpN4BA+cjIp3/WNEAfpRoVUU/c0vVT9HKTZ1hY/9KNVLpaaPDdDMFcaSeoLBXqpaV5l/uYHvZq3BTAfNXpiBkDeCt3B8hggF1TMJM62ZkGAPTRb5Hrj+1bZN9vG+w1D0VY1WT8X2sDLUUbtLzAMnvcWNCNpV/SPmjOT1w6IMrTnnEx8hcBq/p7O2pwvAiqb39cyl12plPExIY1Nqy0r8DbgXYMtHHA1HoIHAFGvEt9Vv4fSt3uCPQQ1GAvqEHt4E3GXUFXEaGDShBB3vXKwhzwqG0L2TNEmlp0kQiamJXzx6dOUKO1yZHo8ucspe3b2AsW9dnRI6j2TSkV5MwD91Pqi3kkAnrLxtvSOGPG6Oca/f/L0RzXPbXAeEwcaYgmMt0M7zXhuRpSGZfQnA5OuHxMPB9C6VTyGM0rgc5tUqh0rz8wVaLqaqWSeuYC16dAalD1G5rwE4k+ipfAnzrB5BUR4qoUyYfDpkTK1TTd5PewINHAqnnyLJcWbdpXvcbCkTCF17LNnPWwSXnfnhIuYqxVE9t2lvR2hjx9HkzdZV5D3kLI5IU3bptS6g6s4GPr4ix2ixs9x09kaGtHXf0G4NaqzGwq4njIRJ6KQTCJq3ReI08pFU02MISTDTTFTCVeUVEQMhdm4Z5vkTaZeukFKtUGugvNZ9ggDxq1PoLM2d8UlIaemSc3fOrvwWtOA1VhtaEs2r04YxwpYH6HXNN0Z5xt3Hpcvc1f4EzRaRu6I5Y2bMIRJSqlb2GfyfXdrjNm0gvwwERxsZWhpqwsL1XqpvSNE+PFBN01RvhDwmGtp4MWi0nGp2c5cyvC52NkKi05akFwFHNwJT6DwUQqyXpi6qAOWSKWOWYWT3gaN/X4Lw/ehkOkm3PkXoRvYqHi6lqcuBcItvVnkfwQH+vd6Nc5W5Z+2KaLgrR4r14b6WdmCLqjnX+SJYlBm/T0hi7RsRyXCWxmf4T/lFL9kzUYa4vH9c8J+qR6iqUE2/Av4sgagyQdA+N3iH5jPB4KMIunbK3IzPnRsDLqAyZ+jGMBmu7Aa5YGbavqXVLLr0/bbJaUoQO+8nrbXb6v7IQkjewEx9cHoaEwRVH1nGONk9Wa5zUeypGVtcoOYAgCsGXYzjpIJhSH69Jkt7c/nD6a+LOTO6vcBXdVmpnvQKWLNuVtXmWw+oAoRlDSKpDCOipGSB6f1C47sO0ilGyrzd8MyIDI37HBFIzQDrFVmzfI9sk4J0qhSHVRcNs0Q8mZrC9q8WPx9WBF8PXBzYV6Q9T8SjqpwcQG8swmLrDVO+r44qFsGmMSXpvugSB/2P5rM6Rl6JpTdTdYuZdd6Bw1AqOLGRvdGVLSM5XZDOWBnnRMUuT7ApQo2Z0e2tXoW/ucomt7ThGz79R92cOpAQYuE/YBBrrcKV8ZrHfzdqW9LeKdINGe2Cx7SqCmf0a1mpbpmKn6SSEtWSoWbyvg/yQ14x1NGc/B3WVvEj9aKr6GV7A3TGDKXQtXAO4iYIU9+NI+5zDo6EtixB6Ft63vF+WMCDYw1BvwH2bRtnW0nNp0lsfY7GNQP6uXQTz3nIGk0bvuyUy6FgSMdL5AGT59nt9ESv8ZsbhIY/ZJBOW0vA0tkOtHQSr/jK1TH+CMGSeV4ceCOQkO4idbk/0xAHCmfq2yWIIIVAhMwYwXxtVfw/Cu0/AbazaRp6w+7t3j7J2GwYu1mlsEt5w13CpIZngB47KdgPuxjojR9NUcFq8kXpFaw0pO7thg6FzMC3rehWw6PEtjNuWj5cCX/6HJoOZP8fjrS8dBMPNaCxe/D3yJC9XjjCzJfv3GkSN/jYFaMHdgmivDeu4p3UnA5e5Nej+LSn1nMFRcsZqwHhUPyGJpvy+cuNMsbOVD17/WPOPfkv2DrscvjmNN0VUPz0SVXKW1oOJjXr6BMAMYKRRc7GGYGlDkL4jGSG4kLnxzKu98V1tKL4CN2DEm16IQHjlsZhTT3MgA9B4QpBzoD13IjOgPXqMzu6xZr8z5Jc7NjX7CgzkGwt8hUFvrGGTKPI/bYPSM2nQa8pPOXu/RGMW8HJcNmDq3KE4C2cWAF3YxBdhQNAEnRdcmOe3hHZgV+LSjaz5dWc+FJcxsXQCh7DjYrhr2+fvJ7UpILsFsUdD6R0q9q6GI+JyHAoEb8W7AOSybBKN3ztDHf0JlsoZWDSPwvB8xLzVM47pqj5Yql2ANWOjzqZnEsbbc/wc7+Dd6EE9HtkvHgIO3c/j4w8cuqkxDQQy8j8vS4LlfNxcsTmW02NBqd38GYmq6CiNgOi0cgzl/L7wLJjnAGOs08vONEC0r8Nbcqrc2cH4n5f+VSCd2sjzZwrrhLdPlx3Y9tlVZsHFvc6rvntFwMWDTp5k4q9zz6tVdmRMyEHcvEoGOl4XrzN/wJmvRqbA1a8ntE/s1CFEhjCdvBzgsq+Nf4Ce6YaA6x6XDJL9cfIOhG0dJcMLwGiHCN4VzsSs37mQaUHrvGdDwsDh2C9nJf7LKknXaiTNh0LZzFvDfk0FF3YJE8IIJi8mwdN9vmN2SQUwqFfN4/nSAdWnsKGKjdfTJ96vq7jUGuD5i2xsl0AVZbF9ZDKVSjEDmqOIUTB1krYT1uAg6HEK/HDeh6yKjmhfIJPCzc8xzPOXmE0Ml7G93Z30u4Yt5mzRFMxuAQ9XIJSmh7dB9c0+1rjMLZL2iT8NRfmGlW5sPp7xdYiNWKPsKfr0tF/l0jQnrN9d8ssmkIqJ0YQDeDF4JJcSHNKUs4uRXRnEGG4cnL8/2p/Xq5X5pP4rOXYQPRpMl4yzy1EgSHGfnwNG5RRZtB0Az9JQGapfXDsSnG6ntOEhZ0e3smQk0YcWx2FjoG9bnxtVSobe5xf16ntNA+BtrnGO0xtmPYh6Zuf8hydoN6Q+WEmqBKsBfdrd1pDE6CJL7Xe8jh6WkVmnXLDXhNaKVHFaHWjMSYxXeGcFaMgp4tYnPj7SHvI3xN+3EqHt86qFJeYSPoeRqp6KVi9M/8gbPOTXXOm1KDtRZHO85OtQdx15+0bMx3b6NTjH1U1KeBB/6PzqBLCaKpTxXehU9kSpzGtk0wK+Yd92OmWGsF7lnmKcQWTsTrHXqgJ5Dim9ERFOPUqB8DC2PIGMqUcz4hTMaY3p6nxq63EJFNAJfIHMt1mwq7CgqJHJrodFjF8ldm4wAJpPyvbTcQh3JVMDmOWfLKqfMJQPdmYnXEZ3YXOTn4Nf6lQlMIGGRAn42G2zUbD1UdXqBXT7atDxTGoT2h+mqhE1nnJLNUqC2fYVY5RyHLmwf4s6ZF0fbfWXDRK72vF9RxzwDY5bTIQ9L/g29xQ/VK7tyNWsKAnA9sXP/AJI5VVH+Wj0n7eV0+iyV6E2/O02omQ3idgRRmA53/kLc98i1xNT6/vB8A+hiNWDLb5MN7T8Yk7DUtRFbr5mhPN8/+VrynBOLm0+2NbOkp1odvGPn/1AvPNgYOHj5d25zGJLg0CnTtrDV6Mfgz2TnLr/+XG/EhCfJXhVW7eRdz7ipRVQ2nAvDajYtJdkuTAQcpMV5O9fLXiA1F+m7RNmKS+Y5DpNtdkI/uRLHOi9Oa4oKi9HRyOt9ApfBmeM2fTXS9J0MbxXW0ry+2Nfp4xXjvd0pWJFOpeVJNhsOyOgkhTl0T7ZrWP7a/2HR7EIeaJUtQd0H2nJmz8fKAn26z2d5sJQyyKdRWn05DqqgyjWcI7P40OaJOYyJlb0A5ntzgoawdKWdyKSPWnlkJr08AXMl6AkErcFK+wAU/DyEQ9Zbh7Y2nwpSHcb6bBxJvRhOXYSExrKz6zZVQOLcIgxbzGuuVMYnzbIYAG14o7phBiw8ms7PXjxZJuH0SwIqIZQNQNFkqrPAKMpFlrV1uIrgqSpp6VEPRr7L4t2nfYmqE9MwgiY7zRWo9MjKvZAvvk66P3NBzCBQe+3LsKbd+1lpOtOGHqiWz4p1mJL/3+VWp06Aq7hpq5KClcLMcPJspG1H4kqldNMdMPZoQn4KhbUVo6ptrUhwSrc0fjWpldB8kpsm7j5I7Y3rBRsnE5jU91B2wtcLZqKIxi6uhdSuk6G+IK0s3B3q/IsVoBbTYIOKd1mVwQd/EgKcT8COgEaGxKxjXMvnH6IhsjRqaK/cOYO3bkIC4QwpjWv1cOyPTd2e8A6lF6sjPYYzXV44+5DiI49bOJQ6Xm9mYe8MH4AT27nwCWz4Cx0fN6T4MpgLRyRd/Jy2Jt1wM8gS/rSGtwHR43yKmx+MfxyXmmqVFhp3DU66Y2ESYgfPmCFV3tsp/bHX0XRpRqb1ai+2BnpLv+UNr7AHl36S8Lw5hGqNYh7S9eryUKxMiHvWYBR3ZPQzw+bLYkG7/6pRb2P3i+DoLsSK2sO5LBy+gR5KtA0Rr4HB80ZhiN38IaL0yGSvW8fJuU7bvWpM5twxO1UdPItKLXJW7l5E4ZcnzXyjOr6mkkfbkAw54aGGHwFMzqfRLNKtrtNAlMR1Y0sZ2OhfOtvE1kXK5wE7eJ4EUhujhTQKmj0wxGL6iHeqP9u9BE3IpDW4v0HDNyF49dqkRoyzNi7NlGpq0hQo6PFzFDZ6mjfR4y3hWCD9QQLCjcBKQypv0tIsYAcU7TpukfHuADZvV3J/Ej6MtVcWO/JCwoOEfW1zQjTmE7pMm9owTAq4Ov6ZGT0ONYHZbnRIfBvKmG8LZqPzgSX8mCn5UZyOwgmFjJAHoK0w4d6U1BPDFuRCIuWQJDHSdx2uh8c+lj7xHxQ1/ey4vFdMkFuIdBAZG1zlWy59qrG0JkWBmNUl7YsTBU/iRpsIKE9uWDJNt0RKTaP2a3Q3O9PmFsQKTZWkZzB+w4aHNIJtY7GakODGYn4n2hqL3+W/7mnnDiIxG4/M6LtWnFL7sGQqkZlhOiYgb6Fg17ux8OE+dNQ2pFtkh5F/FBWe1h4WaxCvO6HrxmMOEPq3q2kfgZ9UrDvmiEnuHn6atP8APq4NsoNsKvu4n1IUBWjAGhMPmvgzELvLtnG0tkxT0hhf0RXyFpw6F4aaf3BSGYNMzlQnOVSbf0IC8cvtPvjV0rhxXLeM8RS+7FgVnJ6NJRA8YlmqhCYM3hK3BtpzfEBdAGkoMW5sPn0PfIQX/LlxGRsnmA2OIDoVp6xNVWnXJa2tGHRfgd0HX4S1xj+V4P5RlGvfyRzObTjM52KytkPqUPWTV4AwOnDPCQo1MS/QEp2kPgPuYoUU57xbcHY4mxUjdPaNuFSj4iiUxavpmDbEUI5wphAYDx2WO2zY1GvaRmq3CRpVdHdwqgo7URUtDH3vGXvx0IR/QVs7k1uZFp9vvWN76zgdwKnOZEVzBFjM2y7FTrsBEDpoXTtFLHkJqWVTo9KlU1XzURv6uZM6z7yC8C/fk5EjhwjzNn1DDJqZTLd0YUIt+XPPkXvDdP87kEgJwXwVBbQWAkhFRcVY3aDPTk0DHOz/nPWzG33aBjABXzlTI2wzt9Tr11g5qUx4zqgFtqgmHjeh8PmQjGkjyIMLk1dgnHwojbx4B9I4DeOlT38UGA79H2rLZL9z3Sb8XxS2YMR0tQ6G+NbRCJR+dp8M4PxPFo8wvI1KM4Ax3xlP2VPVkghGMu4S/Xj0Vg8XKh4JjCixqzw8qSa+iBB8y3FcMwd+p29nTsSS7cbNQJjLjYpuLFSBpwnA9LUZ4ADHSKpaTo8CZsXtpkVNSHmZDjvBRqjNnxp8kSV5R6I/6TJ3eCDdc9DYOvZ0AsR5evCSkWYH6ElF1gkCK8gxjvk1WhW7l0uKOUKi0hqhsM6EEYIhMshhHSvTW3hy/N+A9dg9EyHG3kNtY9N5h8A9VWgUxqPUV+DDIjpn+PypR2wQI+pWtUubSB7fxbYbSBeikRsURoGKLFl8D44kHWDdl64qPQzVPpJj/tUWuXLM3PxpSY1hcxOJPa3MkXb88ddZ4IZ+ZnQDyiAhzWzFCwzuMhfJ1dEN9Mx+bEUn87JZwPDZNDnB2/dhPFMRL8R/cDGTdn9K+XZQmyGVDBXUnJldMRWIvcUMk7m4XmUGfsAI4ZJhPT/5u48tk0RvRYaPKSCZo+IP11R0tmPJ7hCklEOgjR73xbbJ4uNXU4YzNFkc7wQTF01p6QZ7pN8E7X8btDVdqy+4QMYGkR+1YJWUCG+Wz9qn0/4C5/iY8ZgNKdglMuZCGKuDEnPVCqLEbzsYm5PV4QqbLjB/FJxl4Z4wn8YRfrLJSy3ztpe7+uoRM5lVEAtWTfsa3Xy/0JhPGg5RPKXV2YvhVFwozN0Dz/L/QTlpQd91hnOI+MF2+X4MDZY2HFaXzy91nrLJbxPIJaVlkxZ71l3fc0hxlEuwWkpjiNSKptV3n5rKpKOPYS4DTtfHUZ2PmDeFklP6o7Aqe4x2yVKbxtGrTXTTZtrqHWLf8RROaUMq8Vg3OLvBZQlA4gx8iVIeDz+QN18BldyN3Y1cfUjrD91PzvADhQvxL8pN24bTJ1ZSpSNnvO6W5VjHtVQvlBdRQzV5TufcEvjvZ5QSMIDctA7Pz6wy6m2lbRySmJGyUlqiVM+kUTZ0H4rQ9TBa4ZxgGA2wrZZh9hH/nZ/Ax/wJLdaYqY8l7aYXKhCxI2rCazLpbYqEa/f8Zo+mC9jgQtHTcWpT09oqOV2RUWZMsP9QqD3b2wlV6nHzxzQ46p4NYcHLXYNRzHBzMrgmMLc7QX1Xe+NqIiqqWZNd7mxRANw9bKkJwzZpr24VD1m7rtcn1lHiYQsqk6oe7m0xBYsjkXtX/e8z8pIsjuZOUL9GK1lMsJ7s6tpB2ksixC0LvYEOu9V8OWIRxgLtQH5dymPxx4ez5j+AUAs4bu2xOB2k5kXWR7Tx4dc7YaqOa5F1nn6OpJ9zGl4ddMXSsN5OT1XgPOGxDDVzOTh2OKIWIbHYz6hafSP2NqoCNEkIbamUbQmyLyIAfx0vLnyXrxDvvqGR0V5rGRX+H48FKOfDEIiCQRKrBA7jb8nZ6jcRuNCN7mz+LQ6RlbNcHRR6QKbSkaKqzo1iGSN4R9pPMmZb/gnYLPo5kVbIGM8Y0mkMEzXwKRNNcfDg5qOuCgPQ7aUS9aaN3DtDV5KYjf2Fbtf2vcrBAu0vTkR3ftjJcvRp8UCFvnSXGIn3EO7LEfXxLeIrC8lI8p4sXe8PGQfhETK84izoPdSOqTXqf9JiDx8DJQnMcgi1YxDRoJm7u0/6OfHAfdwybOkzwNSOpKjR5kcjCnJUszQ1vBI4R4xN9Mdwp2U7KoHxqK0oOu3lmuXpeLp9p5y5CEt+a5MHtH9T8Cps+DqHwB1eN5uQ0hrbDJuYrUqSqw1F9Vjqn4dVTPtJEw8RQH8WIaFZuI1CjYO1sgWHWg7jsH24ntg+e6xJ4oE7N5G0n2FkTD7iNvcKhhBE1lItP4U2hLXK2tzg4WYr2F9UzbYpO43Ns2RjbHd2I41Y6f9j6tnD8+MDyFGLPs6DuU0+fxMb9LZKm0sPVyCQefMaBE3A5PPo5YB5EcSFs4gvi56HxcGNiOKJBdx2zajg1WuQFWWc3n5jB1p4ar19L5+ntwRHHyo7qKTWJlFzWP4BS3Pw8MHMTCjdPd2+UjLIp5KPsKtDQvWLj3h1wQ3M25DjOuk9SxZOX8U7c3LxqknFiN+dLbwSSAm98dtsUx+sF2M2+PPN9TGntj+iUaVnrLU+DlM2hNbszmSXH8ztqpXTfVnW2ufy9Ea4IoaWHRe3N1Iuyk4EOzFrHXZoYPG0FydULFfG0nLaHIHhV1YzAhdfXU2HFNnwL5eThpmrvmItNDp8QrivDWjWIFo1Il2ZTtHZK9c0iojGWTttDqzvmdoOJRA6NoImWqVw0/QHrFXANoFOuxzgdiGpEfMBPs7Lt0jh0nttCixg/RxDJJ/mpXfUGBevVP/xxo8VX9YYxSv3iu3F2GWoXf8ZImN2DQXbKQP/Q5aWv6BqFeJ2pXBUnJnBUGPFOmvcpNBP+x0Rp62pTkzMDvkuo21R1CHAlb9Qx5M4hhJzv/9pPyE3kSR8q4SAKm68y3Rke8YDGvsPPmL0w7yP0y5nyeN4U7avs2uz/MqIwXDMd1ZJTKgQ7XbFUGjIeh89YiDWEOTU8b3bMvAOiz4LMe6VmGTBaSJ0IcdgPvP9hY6XFVmfVeWP4m5lp9PPgAImqdBxp7KKCQm55gDPK7Vs+SVuJXTs5mDIXz29mbv/+r+eM6eDLwvSW1dJvGp3Bb/e3jJXtmWTzcQrs4cdQ7yMxbbE0pmEX9Ew7AIQE6CqHgCVmUvUH3d0S1NIxYqI2Q/hh4T5Cz6C6QTypLswVMeWhMqoMJUUGm9DYM1oG3Lkbl0OiSHPYQ8R7Hs4hA45Gp/uG1jkGsBHFl4MEdtRbtvbOc7/iHwL0C73WVzEW3c1iRkJpxAcgxXJCfVWuEdYFT0x8U9rY6pbOiNpbI/nYS6kZyl2IuFW1TeVBKM3lKeL+ZDcWN98HGAA2+eWClpIP8ym1FfUtsY3Nvyn4Q8Yb/u2HvkcRZQG8t2JADpStqBlNnwc09Aa8lahpZE2xxg6nL1kcZcPmN0cGlrkIrCvh1b6NJG4p7SM35Y+jxMsyEvpmwQ9YB4Sq6SrPS+x1NYEm7ZDKK8SYYQ6mmKGKmMKzfQYfUrDuGfXUbWOlwrP0Ukq7vP9gXX3KgJ2uxX6lf6MXSjK5lU7rnxoSIKwSX6u7ZTkZJJZ2mJ1e+CRI96W7u2bZ/xPPgLvhE9H1ApAFawEojNQb9nR8iiMxdSMvhfsGgbnfuN0M7aJBuIHNyvt4MO9UdMjS6vzH3/0Zru0CUp7+QHWqzk79vd5VWtKZk8hYmX5x3zo8WnHyLGWcW6HDCtDcFnWnG8nLXCGW5bc4cNkdGyawkvyrC6HqgZN0D2Ux5dJwnzOWbVSS7/G1UtNQBt0oFmxB0ElMU3oOYfTV4L3F9VQ+bx9r43VOkj3xvR13UxMvmrC5cZFfnyx1P0H2rGdk/+mhuCQsCTCJu07sENw4qlMXh2dg2AkmUb8eSDx0QHZ0MWkp0HvyAmmYFPpJc9u+D6OpdIRiQuWAUXoshB7GbC86yfiepKCNOnfhcBMNzPWx3BILPSPZjwaSmyaf9YqKQdRk8nJy7xzK4NjkB8hA9Cqy0cEWr6vkG+1mbqlY3p4mFI6m6Wfpyt78r+S5yGNAS2SnV3bSbb/Zu0zm9b1mfjKlbcI8+ah/oAq5V/oXoQj5HYxYc1I9mHh5QHkXp3eJZ5Iop4zZ8J5tGwjG0Xs+ZuygvpmdVMD9gzh4H9GcQMQ3wI/fuNXH5RFqH0ZCtGe/lLdzVXs9HDVw/c8U+dpZdF2hiErmmLDe7HZNadaGq69LU7wDY+vhuTrOVUdqhbAYfZcDJ/2Y5Ss7pV92ESeFTiiBTiJa6xzhmT0d4/DBlJ3mjwgWP0yS0iRMFR0oByB6LdK3nhzoAetYWz0WoY5gRYEyLhyLmbbNxt5JOHvCxPwVKoJVLQ3TWs8jjEhyYDQw8/T6z6cQEnsRV2k3RNtbZ7MN0EmAJLxrNgIgCBNQy5rEiRdDKEIWEbMbZ2s0FlOkmp6gDylrhjGa1wfyeW7zErAQRvJyODsSuyRUXOyn29o2fVZgefE8srDkIHqE3lHqTlLOQ4/Ao5AcxNFrkdqwvlG3DfJrq4vqLb+4nvBbIh2x1Ag6dgMBYIP8xW1GGQH69o5vl7D+oRjqafH3Z8RP58YvHIYE4/DvzY04h00G8AQ47QSFCGRnghuVVGOQPlqaabowAkWpgMNr/Dm8E4azvKdo5XFFbsHvKjjo0IlEB1MoTgkgUxUbm+NQsFzlSZWu/MbgpCG1lYEhmdZ08QbQFOqKgQ8c0c6ZFhCC3Dzdv8Kq4cS4/PgcdmR9cngwmuw6U0cmnFMmUx9c+cIBO+EhXL9H1VXF2cLyAPQkCDF1rOXt3PtibRvqsUKSPmLJlH7gXJUCpfvSqYDFmX/x6DwOS3AbQ94BOKQu+yMNKTFTPGT5cBOmRnW1Rw5utaE1YdJ7pgQCChv7KUm/MNfOsqe4gm9Y3LlvMGcu1Zl4QpBB+YyydnTHkHbu6xZVCdJm2FNXddNR6Qa2nQ/sklgUJ3B+5fR9p0AKlUI8WkXVkdGqQCzhJINS2g5Z4s0eXoMtOoSCpVgSUeZjtyw6Sm0OTR+ABDxdXvbbDzzbTzQNjRE7gPDt4/HqvKbjLM57cyQaF4/e1Ugux6SzkcnqP27B3SrNKlrvOsgsrcoJ5Ua55WNxLiNm/WaaxOjQYZ4dV4YdUGwOXymmqpqROPpFrf+6OhUdGaACxFqI4IdVXVPFBiNqJ1Gh04mRuCdH5H/fS8dwcrwCkBqNOKZz41BbKUP7/iLGiOM/8+8b8hNJHcGeV2A11OhPxSPqDLXvuPfKyjk/iVA+XOMJ5H23B7H0DmL8dF5ebBC+yYUZrOuAQY3yJrQ4yTn8mVzD4TsRGHEBq5W8nG9L5zTBhX+LzzNrbHHSTyYoMCnyyXnjKCp5y8ro+7wOAKXthm5DGTfPvWO/Q2BKz9cB8PbC52ZvZJwq9udwSrzwvJhCRWqBVYQaq4VzIT7yp/GT1wWw8qRcqDBWXDqQMB21npnlWObK91q6bTlX3mp5OkhFqNF3eG7XYrqZdixTtdfKe+TDiD7jPHhZpnBl3m4R5h3F/eOnHMYBaXG+KnpDa7lOTMIQqyXb4b0RVa3Dtc8gdJqeTR4nu79M8kqtlpRvUCWSO/pUBBh5KHRntO/9pUXWlTyxTTN3gKT1iufgOMYwvtUGOaXXn4GA8dhU6/zu06dmH3nm0y2hSrExlNoqpsgHAdhXrlU0/uDR3xqYKzYrC1A/dzh8UtXwWfk9i7V4DdeeCDcXFjG43X1bdQsDIGjFTcbfpXBXq2pCNbnP92qWnzTwnsBiOGy/9KK5lp2SOjQjm1t2kyh2CDpGnffCXMQ94W96mWM04H2G0D61m1KDsaOHGBnrFbAKK2GgPy0d3eNlzbp68A0AH2UsknQCMpK2AoCibSyMRdG1zNsFd5Fbsj4MtZm2g1wJNKvgHj9ZI1xL3s3HVA3jwQ2+7MKJSmEgr4ZVe4jWvLYrmfhJFiWcb1ZwE2ejfMw+J2aPEEcCdxwj8uPo0d3xO+Xl4YbiLqL3r0NJ7pT8AtkMI1nM0yMXTAvvPrCKBWH/w8yD3CC9T0tGd2H1uEtdwnXOlpQVg5fSwjY7hM9u+PDEVJxOtNsDdCfyAg0I1Lg0pPb5zVgW9o/pIVyCdwAocsFeG8pAW4bAHvjsMUOMKFs9GlE8hzp0qgNE1Rk3pvIIwLkDxcJpLZHqZpbR0wdbnNm9K0U9s6jrHv2/bL1vAv/DGr0OFW/ulTpZs8CihOEzMA8QQbc0BuBdn2NcYGENbbU42T6bI9YJqEhaOkpSP8Ix1pW6eEDJzZa6yMumaEJNLrdCJkMQ7P37Rnr+bBrpzCm+y7WBWcgFtPOKWhTdHnrN3qnvAlTbVaLlhn27rTPKNXQ6I0gGU6tpwjVHYmwVtemAzcHZvG3nzUcbbyKRapcF1AXrpZqSGSpd/6L6OpeqdO4NGo6Ar8B7HN9+BHjRm/NE2bvjLiBqW0ikPK0yyFc8tbzDwwDvjZoMFOjWrMLiXHT9xuDuyBXNdQ9OOZMUY5Eyq20JmM+yRaL+hmXHd3wPvq3sNYS3qlwVRdkzK4/YMd5yt2FxiAUq2+pzyyZGxn72PaMfaMUZtmvgEPXf4yGZpwFjvFELaZ9ItAFB/V2eN0KO1Y/pjNwDjs86XV3cAKzA/7vj+P1BI9Xkc4ZtVZ59EGvVnWo5CTdHbkfh1xeNSXfezbm9jrmIVcUG6HY6fD7ZlqrklTtXlx6SiKCz1N7Tmw1c2bXbOvS3D2fdXPMwVUTY+Jss2ssZHnyuGBCgXrtN6Q2KjpV9+vOGB+HzXpDBTDL1+Z0Pv85g3XKA3SwrTc93d3eFWLxmTsFTB1p+WnnI+78u9Tmhf5JkDQmg9dxBGhTHjIsfExZtmA3vLpJh/dtDPRbCpkxlAVRhVnmCdiIkqgGgE/zhtdlR9tmZJBH0YgGtlnwKaxTv2sK2P/RDA3Hb7Um81+fS/Y0EP0c/uUbbq6ZHIHETiXGGb6gGkk1a3BGFVkjjzXPgCqb4xOJ3nUR5p8cYqZVhMtyR1WiwQY7Ak4X6ynr7NSQuoSTL1Kic7kMP44D+/5G70Ul8gHi0HnLzrhhr7p5P2Z8Nrt1Chvz9W6Ulr/kjeuDcTDcDp8pJM5ZD41lsNCTEpndtI3E3x0RjZRw9OuMpSq7dKbBE/XlTvwmqeT73Ut2shVGIg6U5N8DzdQD7AfyK7FXfG8zE7i8C0e2nw55sB7Wo7ugsovyNFUNOcOK3ykN6+QHhsYTcuZlxEZcUdfqd1bDoGPEKuxC5MyQzk4+b6mXKKKZ8KRz0hWNsrdcCh4hx7HPc2lwQ/dUxs8JWsAO9kO3OVThO8ybd7sP7HK73XQ1P8PPAuT8E7X79dFwkA/U9I9dzr6kZpj/1gUa4HOJFVYrE7gPXlU8lhm0lhOAP5eM07Bd2wR7hxdGBaqo+B1sDmyQWNDw08kRmqCLZcNDtj3BlcNA8owo25mMHKMNjlngtsxD93c9Su2etp4jVIqx0cIdc2UkfgECIJ4Kmq1BNNH4IflA9QBlwKsk6iqJUQ0eykm9OATd6QjJu9K0ixAvZ8OP2/Jb0QbxGxyH6cPZ0KUQpwqbmYwk1TqNwXpQXPeQxBa1QvYpBsPxHB8W+7+MJnJWjVHLEIAQs5IwvYK6XbABZIVVgR+RS1NaGsD6RTQwScuSCwqlNJaMxP0czDlYSmNmvPgMNVbcZkuDgg7ldBXQ8ijeu1YTAo0nRAYdxEfCbjkBJq1GPdDMpNuhEuXzdAxqqB1Yqq8RDgTPHq4YLx/bzOHqPkibWgLNfh24zXs2tzNUbrKTFe3HnPwY1BnOLCDV8eddRHZt0G8gDxdf5vsXNvFpHzY28Pme5AP9o02aqHyZ7p4tBJ9Sw9KEUCRPTNFkXoYfybkRAgzPT3nHqfo3uO6WLv/3ZXC3wc8Ebjmuk5JRUM1RDVpHFn5ypPNm8ngo2vn1tvbh0d7Jl7qNuUwFU8Q5ULQMlaOeVdgAdibjO9NzWK5cXiqSsy4iQnvLCoVslfJEqGnlb2rE9RmRdxUcoFEqUeJxDp/Gr+/B3SPfW6jFaeF/IcNQpxlNXzeJi0WJPs2qZ7p+wB/bGqk8CuTb5v4Zr1I6B4G1g8h1AzuPiuxwZHZDNKzNU2u/iw0bgN+cwcCx5PRvq1HKFT4MDyHlzDNkijyCZZ60W6pyhA+f8aO8a6O8SEfTWqsSq9IXH/fjYWL+aM8jxJIlmx9o3PP7E7ZCyjI2LDhTqtOunRHt6h68IN9MFxBlHmlJHxMwE7NKu3L0tJ1V5YYuFnKvrF5mxXrr3qqJiHgLNhnYFIwFNkmLXiz2oojFw8c/aJc0M3JpEMdGWsRqvPvhumc8XxXGpA7Tgnzz9t4DoGxyiK4W1/P5ap8vW/BvFrqfFxKp9j6cGO2sxNUwgb8xH5AKXbD+hF/Bue2D/TDx9Kw2kPrK4UWzBUM8TuvBCnXgrpKC8UNRfIxW24xCdiKl2EXGbSbS3Na9gvqeszKEdQw7atoPk4+0rgqH54WhR0lyYQEc+jMxqwEp4FKtmppiaTMrapC/jQZfJhLuhohB9TCQETRffBn0TpYxqjY1TQyA7bbtN/0WDiD36MaNsv3vzWy8S3Iszi1pMVzdtX/m8EK0/hQ31Lo7Y5iFjOeopmNv5Dm1bqupHO7Fwn0tVC8ERbFVXOvJgDQBHnS1QxJ07Y9XoPDUVG69ABYXZeDVcUxzqkhQ8JGh3YUcHRZNqx+6FYHkbOH/FWHwH1BQUXWFW1MpcgJTRbFt4Tc7qsvGspu3HVWqV2cJ9g9Mhqk/u9NlJ8C/VdhQP+WH6PIGn1ldmmuDtNnAm4I63zwM0q2HUiL8Ks+SZVH6QWfS7cV9u3WNrk0PjRWVIBZoxTA9EYmNEdtVU0uoodyaX1S1WZMzcs+zVOa8sJJNe5P5ZaeE9TA7xEbKGr6nAOQNZ5QGbl2hY1/9h6EEChbqkmTZ/uzf3SGumJCq0Za3D7knpdwWfSWFNYhsThhtVq3NgNDgbBL1yDz9qrIZXqNCxKtRO6gSQRqqJ7HgAZrrgU/5aSdcemdLnHt5e+l0kvy8D9/9/mpb6pEa8hIzWKqkkaWejKnIrhTZn/wF/JlxzkJQNRKpS31R8yTHvz8RBVf1m/GBDbD2pqk6Mmdna3zMNO7ygLgmKu5LKsHFg2tGaSHnZqGpQsgmj8lYldXvfGzRAZGhlKcZfb6ekYYr72LgNTfyKcFAR5mnz/d6Tl2xjXjbM8XlGrcEiK/yhEMk0z2aCVaSj6i5L5OmvrXSlx2qu4snENs2Yk3u1EinlS40dfQZrxYRhtUJ5mPh2vIRV30D5pm0GClJmDNyLKeMWhZe2uWp1ZcRp6t+9ldRsuBb8mdiwJ9sVMqGzCMRz9RxFDNs3aSfNYrbNTHy5NG4u2X0NIQSmVlgqEGj9c28ubIApf6RaodWYnAGvqCo6ybCDpEfQNcBxEPdZBEjU2hgOsUwZ9KTiOXpHAmM3TeyRp6lIsWfbJ/2RdT9CLq3E0G9C+c5Eq3bFzoRobXWp9NUPNNuZ1H19SylLVgqwfQqDMBgn6LsWdoe16oGZ3cHpfMt/7N6J2b92W6HXwJPdkf4YHlk+o8kENE3QNYOmBGovOXsLa2HQ5J+OW2w4yRGM2S9/0SpN0mPhG4iKqVh9ZKdB07W+XnFnu3ZM1X1xcL/sZ3rvFvaHV2XyNwtX+TVKm/2dQBYNO3aX3QskXchgU0/VX2hcnx1aSyZCR6YuZPtdEKvvZSs7K9IRA0U7+e2Y0g3WVmcjBuKiRWhq1ArDEwYs8u9p8ZL4FSjP+/bKqAgTPjM8eN6mx8b8KdXRFxyazN3tfpYmiGVn3qUmlvFMWbxi+8v7Y0BSNDWFWTkbgmSWjqdMo9pvn0rYFhXnvNOSyJKn9TWrcGNxgNDI8XTqmp5sdpsXhf4PZxrBG/x4HpKu6GRWfIW7dhMNLGBP+ruyTsg9KJ3cfAxdUJrjl1VFKjsmt9kR8hnd1S7UDzWm1rkEGQY4Hm9TvGrY7mZAUN1vWQDNYm6sAFu9rxRtuxYIjbgMyEzTTGQu0KQbOhdJx9QPE1U5w9Vn6KzEatQx/4L1cw1thThOzqxU5D6DL4/o1TGCqfDpNvOgHVnPqOrWg8DLw5pqjqbO7s4uBsbzaFHLeE6UDVDvo2GxyzY6Gjpyu1lfaO9xOCA7jEPugBJuA34vDUK5ydYThJPfHDEYVBVjxuZbXWr9IFbmSFnCkeNeqkujPz05vkMp0yMpB6GnW82Gc23jeGHfOUbsLsFukKSwcZ3IZPXbyJKhvP+jUH3IXZ3a2cEH5EapQ/5Ai3DAEFt1/zpTeCKRHa32+a9rKD2Ynex2nDpYmbvJ4IFb2tbhDxJl6jpEYQOPqdqG6/JMHbrX4/NiPnqSJ+Ke3UntHHxoyNUAdO8JgW7nBiHc3MZDHO4lCCbmDDD9aNsFXKE5/D6FNrlhD1FzLvrRKtxuyaeH3g0BBCiPhqiKT8Mr5T0rmTcJrUkAthmkUpD4G0ko1V7cNZL17FKMzBW5ZYD2AVw6czGkIEyTuRU6bQpHt9Rn1lcX26npNKzAxOIbtIjKKchUALJBqCfu1UBbDh6OO0vELdTdx8DgTJ+xiXnvdGuo6Nuv2ZU6r3Gj5DI1YMSV/+OKzrVKtHQ9gn5HlpanaoOPS0SqF1mXqnQRyOPOOgKYr6CuwCSlW1jVNPOb1RCgPXpjJNbQ2A8dIFdwUPm7jtr2+/BlAfApgHHuDk6L6lWHrt1mmXLtU2hsBf9vSDaXR6IhDY+SkN6jRmZJoxz/G4YGZg3FYDYlsFKfV9NTEMqT+O/7HJYeV4d47tTLpTaUhnSOb9/ZHf9q2F8D1hInhHMcm+l5DJ+2a5tSjP3xW+oK6E4C1zbAfgxcunqlrM7j9XxqHhWtWYJB7nsnmkW9J2cGbWpxNJCtdK18zcTyC0lrNnxnhZe+NxTfhx0hsVK0ScJ6bPOQ8NmEBOxZUW9zM4YJQeEatM4zxxhbSvLeA4ERX/ViLAxmuQ4r6dYx5ggHUxDE5cmqXgBqzzfkGJtkSWvlbn9kPxlY6C3VZqWhSyFbxxS1sBD4sQqbAqWPu9smSSr0S9vKqM7QJ0T41GP4PQFRzDwOTplhaJBsVLGcVZp5Mcdk6Y9bEfR6YXeU1sJfmWcGFhN8ZtcsaTcoiyS0c2uJdL7mHiBBpSzLVbuQ3zQiUxG8U3bRdJbHbx1Jn7EvtnB0+KsGBZtWbtpnd9MaFb1Fc4U/FWF/OmwMhnGvtfHzzbPmCNx6TtSlj+rUKTqDvixoMCZ+XmmrEBCS1ctA0vKtQPxVUjdMgxWvBqDobDZhh9Ef8JOEBDTPnpFFWbbJ3xsNm7iFGjAV0IZdw4GtBX5LSKWBRiIGNqyX6g2V6Ab9kWKwm4cSHyc8DfJDkoX0zPJGuBp1uXcKbkI0CfuZYXd8iXPE4OrOkkh6xnCUiOcXpdh1WVHJE69cMJ8S4fsyuuxYbbp+yR2lzh6x3CcSjYsLP7PFBaKU9tYBsEsIjZiVcBmAZz1MzbzrbHLPDKrA+oRfhMPZjRszEhvgKsnlFsTfePNjXCpppMj+AyJ9CWSvK+qm0U7QjB6lpzkaEiJPYTn8LQMvpBM4dIB0wAo34E2r6chA9b7U6lp2GMdGJ+pjUSZk2WkTxvb5IpNKPwoXr7BmeBtQh1f6JhZbdSE9dyYuAS6RqO4Ci9dLX6e0Zh03w+u9shRV/Sp6MNbfYpXx+S2D1CpIgEL2xkpFPR8sjwiSkCxsdnb30qhy2ewN7dFjGDvwYw3TyTfQZJq+8N4C7JKdOqIw/tN9oX5JdYsE3Pcn+qzadzR6nlcpINwiKkyTd8kOxbgMr0nHA3Cy0DGOFMHMpmSC1HEw59uPBv9dt3FzGN+jZxeIkfQB4B0FJNL0gqhbFP0Yb96dYbIyEUpnGBJvgOsHAIzhk/Z+21eGFNx+9RmUcQ2I16FFx1Is3A/GO2RHg9/AhB2DiS+8/vT/l0D/CvKd7zAnMpgJ+qmS5ZSDBhLgtc/yowvvWk9+YyUIQCoyTRp2uEvQvnlJLf/3s4JpJc8M8Z4bHmTOr7JhxhYZoVAZJIl8INkFPiRomIfqCPvsIwz3dU+TCiAWgRzeNK4fSj2kULTqkE7FkU/zc0LS2E1CnqDkpvEkX/w2a3zxNOjN5l8Mp3XZEQqD3B18lHzJ2uPbvRjnjo5dxxFDTXT7GQnMzBmKEXHJ4O2fQVk8bOlIfyh9cErV4n1/W4Pu+FDPqHXb8ddjAx/Pbu3b6iwR8dlRnyqMbnZPsp1xo8tHO4t7FGR3NjB0kPSUXcjBAdfPrUiYqd7qJ2ObMm9czvhyuNZeLF1LDhTJWdI+VWkwqfLe0IJnQ1IHtR6bpfHZ7GxPjrDB2e9yxUJNvj9/7kBD4wo6D0uSj7vGGIrVYH6RZ3SZEFK+p3yN9dBnXmVLIaZ0tfJ0wovL+cuw+NZp96EZDm0aY9G6yl/CSbzTTvJXXV1R76xn5/jqE2MC3rrWfgIFRg0yAg9Ypdxy+TBr2cRHFJE/qWAY0uUgYDn8xsMCn3GpnfUfm0R0etXc9OzStX6ka0/mA5jNfgGx3jIJUNZ2PSOJw7jOrJwtDRmFQ7aqxVsh8FbW+hd/w60g+EbLoiyw+XR27t3bVGtnPchjeGDCwnNm1MAAw9AxunmjsrxjwJNWBAvFvSmhjEaSzAE2kGPZ0kmdE/j3xSgY+Z5pAqKO5jBW+bn3qhRuFI+3jEabMvDlSYrzzIB2YMSnmvD6YlYqBfE9GXg1sqSLq1hXH0NgV9jS6nA1vSZrTbzmgMDXw/hk9utJJF8EH8pxUmEX7kdZcgm0eNxsgsGBCWs8rCnYuL2ZnxM0mOB2x3r0pi8BOTYdbaHxBxa8jBi17FEtrsMh4X8w3dJDoA1wRbZ1HqBkJYo9kCHoftADlJ+uGrUGPPGJtk/iWOuT4d+1iXZxpDI7ZYLYeCkOLIBQ4qLRHbqHnZv/4d2BdG3pRxfc+oPp8Fa96bZAk0Um9IW+VzeHvFJXFlfR6hHZ9oH4yP1hxjtXb3iFr+CYM+/f+xrxcqJ4qVwVT7CMEo4swkdkTv3sNVtUqjBzmhmcgpYRB6S2GEYE7Z7QWUDtB4Wrnm7Bc7CRA7ZMNrBzRwUB+WXOqZRrjnP2UNsyPFq8uLm6Xn1d3ZcEK93L5HB9AgsrZaKzZwD+dDtzbyUn3IWsOBHlOXmn34LqrK1Mgyi3KF4M3XoaW8uWDPyT8f5/2DLBF6XPmxGqvG4xiPE4EByP7GOoPb1cJJVP8+OZUqLMzI/Fz8vhPQqJXt+QJLZcO8K8+SLvDGF1SL16iHmYGiH1Jd5TrD4T1tSnMlp1nqbZo9cOxYfw4+ZFHmq3RNQtOYW2i1PkcH+nK4WUeBBxzX0vuMuxbf6N44wK4A5BaZWOAXTL8eQFT3SOB3MNd4SA+JKh+OjUZbMc5H0S58hGbJBuIDjUqQ7ttXZmGg8ETrrCGTvWdzxhUnKCcp5zX9xbEIt+eBvzwCThXEz5XzcyoWvnJq02TrGRt0jYOrgcnm/3PWaVuihbmOWy/2R6TKJ6C72HW8UjQIaGcdGYz3R4nDZeY/AJ2vr7NPnf+c7gFy/jRdgIPRG9o9v+F+0xClQGsfjNgB/bNHCm8mZWl4ouSNT9fpaRBHqNw98HZyJMKfbzhsNALcPiQ0TtQdZ+2WUEOCBCvuUB4qtXP/cRSYxxgH3HWAB4ZBKRw+wVvhvb6gvXkzN9mTefO7Dfc9D1ZGZfoNS4PXQBNG9/iyqM+Pk9Y7eFxaHjm0wBMjpNpYPswaE+tcMKwfhqsKu6yVKgAfJcIXQfw3lbNwTChF4eQK9lvbB5AT4XawFfSdfcEBQtZj3kMZYyBBPxzb7RnxhgqH6XlvB2PTB/EsnhBp9F7nFDO2y4Fn46jWFWJj33dyaFvXFthl+6ZKjPgQrdBBeVQvvu/EmsEJSCWbJUmvnLnUYE3ZoHs0Tj6r5rbJD9YmVJT8mJWLn+qq8h04GwBDCVXOV7sL7u8BwCjz1zu2ffUlkiwvSJldjjpoRZqmTbQGFFS9B7JsDQW8vEDzJSXbtYe06+SmLI9X/Lc1QO9I/58hI2yA4sGm6OLRYNnVnERbAzcbrzZmxkFSRBRhddHI/dwNaBTfEwYn+r9w2sUYBo7uSo1LhyNm9jCgpzQ2K2l7aXhuNxgYcs5tOoJtyx49lyYgHeq4pPmxMxHJ3TlWBKj5N3k8Gix4lHSJ3KR7u8Nf4MRQc4q4yRzcjLKjtbSGqOUDkD4TszUjiQbDMeBSyvRHFbKNpu6SHBp8XSEyi0w6mrumBwbEEXjNLQXVqRm+ZHdzEROf8Nl7oyl/HGsWoNlyBUeT9RshhhNyrMnzyUHDcmQmuSHWk4wz90m/lkGFnh0zcC1JF5Vxj5tKSCuXdrqYNsF29JcSEwxHlUPhE2Yd1WrLTa94WCuJ/w9UWnxTuHpjPpU0KqaJlTNxkddA0QQvEx19DhmmhApLxvSf4TaDCEnGm7AmzH2bBUoTKoUwNZ8m1yHqfFUmiuZ7Gw7LadEpWs3qwXamtnqCJgVSIf3hyAnQTZlG3Y+iVXVFPR90HeG+PiZVQaifRM8q+jz6zXuGwhqtPIcwOHXU6HLKEJNz8kvwPWhBUqfbkE1GkP09VEnDdpXf+NUVGwJYCglbzZ6SHVb8b4DxA8FRdZqwG/yxcXiQdYMnrq5o+NRHqhOvchmN4E4oJ4i2l0V+h1JlGQMxzXg5y2vT1g2bgaowR/ZEtOZhNPGtGSWbB9HIYRWnWaxs/6N5aeT5jM5ee2rdYfTM6Zx7LxiKNIT2hYY1xl9cQT7mC5cYGyy4YKEqGoopyyb6tSWIvbL2o8svYGQVdcw/aKGc2rMdG6GqZWUTkJrxO4gBXmDAGNU1M8Cmy6d9D2pPHyGF3xAhQUBx6YHnw7VxSP6Ha3FUiZSI+5KzDKhlSbd+VJOAlxoxm5XA3Ls5ZjXEdZ7tUXUP4KoKPTsuP0VXeEfGVw+gQwXOBXGvJjcR8dWGJ7nBOY3Uj6vYvcdHt3b3v6rdFpuqUYPvkHSYt8Gz4sa2vwZjdqsrKqfnhnjgP1BnSgQhnEGxoCcL4FbGhD0MzRfLEmGBPFgfZ2UZFKpB3qUF9Irqxp2Nd6wNRIfhOxjI7vuy0blfeBcgdMOuF7oB5c6yfrWiYEM9CfojoqYc84BBW6KRE3lA+C59YjGDRa4/f2e4VjIsP+LLB92JkV9THDjgVb+qApMDTeqldMXXgr8e9zpXYwK1HEW3cVEk9k5RqKvjyKZNnq2Bfkg4djSCiRbAsMAwZEDmwXKeTItzKcbZqnMDN2Kmn6Kcz3vRVO3S0zKeTvdUXV7/KGNlDo1IwQgjbltBUxIKuErlzZd0XYMZpgLHb6rlXdCQHGCwIizcUIEhNRP4YC7PRAHWzgWO2b6AdEJ4lOxzyR0HG9z5pNfL5RgGJAQ5d1NAgzDeU93kS+E0ySzBTV5NbzL2ky1yLBFvQFvCi2gRUU8UxaUdFwh4cmU/uN57xKq/zdbQAuj/wJvJ4dzZYXpG3QrahdeEvnpTUlTXsJnyqgoxfqgMjPbvAlv0V+ihOSsQxZ6R55cgXzLV2dKkDrN5DjqE93m2W5rqlLHQzZPSdPQKsjXDf4v9dWCNvtpVRrAYDXvmY0KG9kLxyZAZ8B3ks3nTbTFrmEVx0G+E/EtllvjGqtNPeEX/PL8qhtBCjM8yPomJGmSOf1qrHglx4RTr02/WNpATSYjxo2Pb7bqX7PUxNjTXRvbZUZ+Ak6wP8kwgc3P8d2ahJVshECx2lxaoYiArmXqUFibjsT/7KmNeqkaAH3/zrBm8IhPWEAVoaumjBA2q8RWD5HC+Po4CRK7EKq84YWS3cuxSp3vgk3LcwQ4Y40YTl8QBVuKNFN3KlIELpDRPmqNXaTmJUCyHk88CnGOKCk5YwGeqyZkOO2NBunwX6SHsHDFJXgXhqGF3p8x8jOC2eIMG+1aGBEG4J4ngFGBUpn1SqdIyThX9YFsvI/uTuo4TEP1I2MsiIMo6fIkz/Bn5lOCLXA/0w5J4lNgmKHUSzjiE6JD8L8RrvPjYUdlNydgmTbdeJxCjeFbxaS+hOGQX/S9mBv4eDF8d8pSvuOZ7CFpc3rxvPoR8Y3JDypfIPtlO4WgQ2EJfoN3e+3m3G3OukzDEmL1D1XtYSJ/SA5u+AxCt3jwfTy7T2JuzvJNhDvz/BmN2DQSInpCxqW4HTMM7iDMdsqVkBEo8HWd0f02xRi69i71bVfnDhWQiW2tTWqftYFJAQMsqAKr6HuohaLxY8ZpRpXqt+hAGlwJ+/GJcPgW36SVEOzTpBjKuJrT4Xu8zOfTBgUwn1cjEZeBC/RZyETHfYcOwYXoJTtEwEZyTEXHjjZRBDhLugc1Ah7zJiW6o9Nrz86ZD1QWjFrIbn0LfcqP3gxbMcZJ/jJAhlkwzOGyF1V8eyxgMcpD5PDYXwYGfXqYdfFFCFhpvRk9UVa+FvdCgizImUPtvH+nVFHnvI0HkPFjOwCF6O4TjgzBR9brEb3eNPyCOhYtMHzkN+7CMJy7jhYgqKtA3ZUr40Wx1+/MUeuB8bOunlNlzuLWm+O0ugOZ0zr5lxHbtpuDUY4jROpxX5eAjk6fTLhQIJlu2t4MzZe2THzdUNa3k5OUGb1m9gJV+uF99D7Z4QSqzyyokL6AlF5Q1Pmwwj4yvXJGzL6yw1IhmOW8FnSnOi+GqT8rO7HqdoPxSq3CzbFDUYKC7s5INkIgC5i6xzoZUoRq2lAzHrJJ/FB6l28nl60iMrtRu98dPXyvylx34qsfaOjmrBM71ksOqvROcb82LkWtINQT4yN1f9jXMt1tKFS/mXd3KQwDFbumwO6ScfH3nRXfRCTFWzQOaZ5S1RLzpnkeCOWXwvJxDsRQiZe5WybzPZXaDYNUTXT6O4DzOwFGBj6YVtrvmqcXkYhQXsWO8S3GymGQRG7w1wyEY69w7FhMVyDLWL62SvYlNUXp7FLcoQ+Mbxm4xjFmlr/Vx7i3FIl+c7lt+aGHR+QxB98H5zPp6m4/tOvbXWm4rR7tX8S4UWEDRRkJKDfL9Gj79tzjyOt1XnYVpxrsI2VLK3XZ33wr9ql7O5cgD2FkNfBzVwO+rd9wDN6bDfKa2xnAf+1w07pMV8OpyF8kdptR5bcK66W3ALgjx0vXXKYnnyGY1WCbPzKmOH9mYPgkwTkmcdCNXdKCTNkjcg0VYBjh0ot5GqSn1gMrLG8j04iDjbpuwsNg+IJEAI0mqVhK5TlDd4wKPRfgIfbQhpdSIqxq1zovSqkwFl01VhG6Aw38gsUWHrKHN3bP/7oKEdhaDCoUuhSpgJSbqlu92J74mRxVM5oW+ipEPaMbMv9M2kmlEqMZJoTTIWhD1OSb3mMSxLH6Fu3ldX3Unal1EDnzDGg4lRjpxiDKddfiwJmVSHMhNFI5OgXnt6JtZMCS3vzCDGRU8vF68mTDgyxCt7lBELmCIFA6D2hsDJEPX3NHzwbhhX1xXzXu28EQmwD/ZWOhO/n28m+mIZ7CHVwnrU7NvjYKTmDM8sO7bvdQ7+DAhEf5SNEPOMeHRrq9uhQ4GRQ2VKbYmHmPkKWw2LUc8R6vnSVEHgRR2E/d8l/I7GvJzCrgzbBsbKGjPdh7a0tmKs+/YkqBtoHqLCmLf1hbALAqzFWt2O0MDhqptAKwbEGwW2xjeuUG7vR3GmGaXNcmQ3ND8cw6U+Au5U04OxYr2tsDpBmPPkqH+bo0BVVVC54/Y/UljKgtvk7eHJgMrS4fJ4/lZgy/JtmbdhBBijYCUPSCrufp4hBNX8w+ROB0A63kQPFn8m8+qOJ1AkkbHc1vqL6t/EijoqT6Fr2mCpKT3izWwyvjROWftYtNbzx6Jx5A6xkMWdT95OvyRCQ1sFaiN7xpK1j24x1RyZ3sxJUyZ3FtRVZmvDgq6BMVAKBGtu2C4S3dc0G4yOAZp2HNPJ7u5fXHfSZfVprmlOiwGw7Zc1uc/W7LR4c36LRKMP/KNImSoxiFLWIYRtaGIQIfM+i/M+ZyMDf0QzXwUrPNmwrw1F3n9K35hUSEvbTfuXdh1vMXawQ7f7a6VcU0SAB+mbRQZkO9hg62tBPpmOVvT+s/dYP8EHbbS6VsTmDAvxaFSYRG6QyRSAq3Ex7qKtvbPgKPiQqt11dZ7vOBCFk4vDVhgLEWPjmtV3j0dBBKZ20sGJbVTJaYfsb47BOzMQ3R9RDqbdcpYwMaVTFZsDRVgQt1gGVTUK7cEKEqyWB0svrk0AhVR1HiSQDrqj+6f1oBZNTXtGZ9I3q8/L2ARFQlJ6LS6ZpE3I3WPEu1ZlNSkk+82NZ3G+hP2cEvRP94h5NsgOLIyuYs0LDDmKXhCsfWvRPOfsJwPPbTrQYd4BbbCqMBh2q9C6gYtp0gUw8uHu2ui9m3k4FJDnffjjsmbhAomTa4PwA/RwaGtKyVGcA1nmIPlSoZ4E1E7aJhBdlZwGlPVeEGdOhJ/H7fLNAwGG5xoVgSZ3M2Gj1GGkWGmb8WG6bbxsuOWeMMB9RzMnwfexGoSHeCRxvFAQ7sCuv4lH4mVpXTINFgxNZVvGYXmKkPuV2nTr4tO/Z1WbWHOWuanTNi+jU3SIu96wCWvvoqdgV51qdj+6bMMvZaiX8kI8flxbbWlYjBp8yP76ahtDGOlqnKsj99xIgHKaQWfFWOBqQg0EJBbqbzzzK4bPoP0FmgBRhOo/AvtgcKB1zR2tULlBZ1y8UmpNsZgnkthUJlW/EwhjEu86w9bDPl/4g6I5Ps1Bwnn/Wmv5yM1rLb4TUJwY3WlqxhRX4/l3CXnMkVB73nG5ktbe0MGmKuZiYvRzVIDW7GSbYzw4v9ZDB+8DfGtnc3MtTZLdr5bETtgqjI6kjzrNKK1lYGcnT7hh6LOm4CuAOxO6a7xWIhCp2fPvx3hoNCRjUjc21NJQvm46J09+WcC/8/LRh2aFwnW5xh+FjIxVG3bUNPLG7zK5QSBNGLQMyG3GvEnH9mPDB1uBeKlcGXRr/RjmKdwRTqb6j7DQ4v6xn+qL1MrdXXDbjHMqI8QrpJHAZHTZM+ueQJdRy54UpmJ9qYmdNzR3SW6vBhcs7ntAm4zZY41LKE7x3f1SyYzP6O+nZ6JD44L4sDuQZdM/8DNvkqP2U9yg0D8QjGNrqjWwc+R8cdfQE7EqA8qqL8XTKKC0zvZXrgN6Us8iAOGQVd08SNPSaESMVWthiyxmfAm4p7C1Mb8HVlWwVbhB6pEdnqPSDSk1/IfxQ1AbKPHRnm46FjpS0T7OXbJdJDHqoT1sfpnLvW7rjnJX1+LlubmL1fA7CgMIK5p8Dr4n4Tn+GNCnDJxUOiWW1BlmUVnNlrzTx9FO5IKsKkZAl0YKmfKrvkwyjThBt1R4/WK/bH4tHGPZjWC+sdyzfMmVHNbYDwWmVnmVF4iCYbO4IjPZ5PX7VkzDmuHNm3f+2TntAgVgKEGP5U3/f2L2Ka5KbLrJMWAJa3TiGkrV8HKwPmUFH8ErXD4WOB7xiYrWJTl4Ogb2OJbuDaBFJfRt+uWmsSYnTWPGHoD9qYNMW2mGNB5GZGo9pyJ4m/hmN3rL/q6VPNv/e9rxnJ+1vqnoOYeP7m0n0BupbG13VdRAzPH7kPnKhG+lUzN9VdgtnVMpo/qs3XZKjgOnSUtFfkfCnCJpzAhvBxxgyqk4RqHLUe7SQ1HU+B3QkXJ+B9NGMCxSKAh9jvwTKrcFgbPrprV8HqjpV9D0lGu/3QWKPGhVnQdvzfqWJENszgNeE+ZBd25ARzaZaYZl6S1NaU9dfE1vzDlJAQX4n4eijljaJBvyvwzwmnrWEgC7gkAWBgiPGAtT4PHMVV1cNm3j9IZGWB/oK4iy7YH0ogFnLeiLYJnjmqUG5lRWsxbc/CtqiHF1ZkGg2ITXYtXH0eh33fyhaIv0j9LSC5uNw1tzt6NWbFzJxbCPW8ER3SQHuGLcHZmQLvpo9PSdnUU7EhRbTzcRHWJaZWhZ8iafjFEtF1nQpjtSsBoX/3P5GKrHJdvsLyd6pmA8f+1cVAcFpNdfd8cd7+o6Q33YeTnZ/qL11qaFzbY/vQ45l2w3r5wXB+Hpzy1jY2H4KRtLkTgzEIDQ6Y3/5B3K8n6+JV5AbzpaHNe5ndmtxQMojyjrxxDNlvvPRamO9NBw1f5rz0Ug9DjC3fLbjf4G/7MHdBkvLHu5bXo/NdeBSGhYGc+eNO1Ot61ynuNyVKqUgxdGN6GZu4jm3YHQVqZqZfx8O4jC3gtS+sz9+7CpsefcNjVxyn7F3c5hXsGwRrz25a9+wvY0N7bE5q+7wixsbUmIgdr1E4ETn2bdUHbz6eHZeSVnN2PCJa1ktzSpVwJ+dNcQm/qxd8Gc5+1N2DjnSJ4WdQR3n2Y2J6IrDg4s1V+2GBofsC0Z2PYv01lLGhqDWJ9bHyy1dExH/zi8CTSdWe4FAmnIRfQmV3MjZPlu2SzfJHzNtHqhOnTEcGkMz25oXnZZeh1GhMsp0p7zVHxRuB3IqbIaVpqAsDF7uk8M5pTxxdf4HFxTOHbMc6r5ekVg9TdjkBpFG6B3OiuNhlE0aEd23ljSLkGT3XZ8BaX+Qb/Kf2fmMbNyN8hMsEjdS7cYTOEGdYQ4IgCSSCQXSwKxBj8rYiDmhnfQF+L9fnE6V2vvFeWnHPJqNlkQRBHrK4UDWyHDKJl0vT8lWt3fB6WP8jbhFuRZN5ilBY8iwG8vvEl33+pLlZdHmU4nYAOi/vjErhp8jzPPQZ3X6tCxs7/d72cfL7DJwVFyH0mgG20Ar8P4VzYhFXlWyZ7+c77qzv0FrXjSMMLyWGYYVUtqh0CyX3oBIk9fGZQKgmYjCNvPzA4H1T/Gp8wLeTAiyOjSzh4vYDfihkRDf3LJZAy4xcaLnK9Cm5GDLgxwJ7YrEzn1QnR8oTAMSUFBbXnD57QNPJtxJzi8qQ2TuA57YjFLjoVX1XSWd3/tzZ8W3OU2j5j4hT1KWpzKKIL28xTnfm+m+QpPEPaeqgQlsNdC2Ufv86DX8Zcs7eHdT3F30i7+gf66J/FFK7m+zkQ+QTRTsA+Yzwyx/OVA/YhNufg7LF9M0+fAJEEP5gpxJaxJ5XZkQyGVFGXl87zhjJlrjfk5PXNrIeu9eVufRRpqmvYbRt+WYHQ06XlALU3MJ0rHlmzpK5IGVGRHk0uIN3YXcPvyd2mLyEsbFLDYp0EWNTKOrmWBCfv0MRUXMBTs+3vhm4Ysv5DyB2LPhHsgJhXLBjBTCQBHI47iHjI7DpZOC2kpxGMf+GQH1GcQOgpq2+PKt5zmgJfWyi0hqmwsG7LQz/6UPlRrVUHjYSIgl5dP98xpZ/zpJH0ukR8VwDRbZ3JldBmHbd/uBpCJ3zsrxDPxHejT9VTskYX+y8nWvaMT6dYRlf6Ex1r+cGAXNBOaF4+VxzMdE0hk/M9HCQJVF8vfUYQ4fL5+dxnq6iUpjoW8yTNAPaDGhuHn8jS3t9jV4J0G91XU/E0xZo0bxdi3I8ATWyepklpOc49fy2uZ2q2Nt3G1Mdu2ifnhXEUP2uCVyx2MFpEN/PoU7zQF7EqnK+/YTBNMv5uPu0/rtRyjFDSkUVXTr0FgL6JaVu7UmWMP10jycfzRGOuSNVvK9qNPMEHYs08xwPY5wSlRnX07zN6nBKxe8w/xCFbqLEOV4qAwFkjXEvLzpd7FRXzvyzOvoL7Io6xlCw8Z6pvvs6u25iYFdRS6R+ZUDa1O9ZuyrYKbd4SdRl3ivJ+WQ2ZMo7VGiCOEQ0Y7k29ulL4tDWDcZZVMy6Q9vvXvyT0b5O+QCZO5zrFq3jJxnF1P84YUK1cUKZNbdi9ZaF7SXO2WPgNDdB7HDc6O+oWAypcMqjBq7GwpE4vlB+YeSDnbjh5jS5IkRQR8eFfQkI3HEJfvGEqAXCYUgkeqqFSyMLUPHbNMSPqJLtnwDJ8Au+AHu4hZXRXxBVRO2XWIfxtQy86Cs7ixS3AAJ8W0AAdn5tGkMsU4HnQfSrLbV/+HbJcXa3V2j6kGSq+6K+7sBNoJlbD3h1X9WpjxR9HASw3gumQ2NNj/MvKmkXJrWYOJxmxSMXZbXFIJlTqA/CjHlA+NNAbQVD6JQ7AmWxVKGd4dD48qwEFgwtGMlq+oMHpKOMJAxSJkMDnFJqUpl61o2aEJ8RMkhJOaTyN3nvxuFWlUUBxAiF7ksJp8NR7HT9C33ODKiqfg6JQmSdFFYwFlpdfmexfgzqdPmUWhiJcVAOAtlVbNLmZtn/k5NFMgCHTvW4d6qsQRNlMvhXsNkGDQIHOQOz5Vt9J5EFxDdxuXr3JR6/p62VU2Q5KNf2ySRA5HbppOxYCkRNmkzhujeb2hTVpIrgiyUDjc5+VIA2Sml0QLlcM6zdi04t73RSkA1Lk87HZ+J/ZrWtrkPOyFDaIzekJrwbObwxErlfWiWMUvRD6Y5A8NSZqr2xpaWktMXzrbI1U9H30rt7n7F1tbffYOB3rfEi3S/54ZQZ4VdT4l9zpq5yF9HnEuOD6yoo1UXjsPv5TvfG0LbRT1IVYjPEuNkJnNNi9CTGMl9n1bcGqVSOwd+T6rdJ9pZ7Q1lR8mNbgEWf0hO5O0qtFOuCcI8bq5oggMLTS/nAkeLYqnQZ8g/rKwXlbz58Ui0zUbQVbKTpUW41tQVtmTNBuEBd3I8ll+BbGkRKqR/TYqoShrWkfhkQVpMcT/JWcoN7FZYXBe1OLsl7uYPL+lWRq52T7+6uL6v0LuxuKFWwrYMYrghk4DX3NH81P9YqQ38YAzkvAPCLmdeSG8T2NBXTINjhqvzHzhYiWKWVeTkAzRiv5cQm5G2jS76WR8TF4lBXUaS7fIhbB/jPUi/vSdRFtLvPtHz6Nw+fhCkHJQsPPPkVUu6zVWmF0pPzEQrtJa5J/gSTzXL1+9nMDozRpVYPNyk7O9e1hhOXVAH2fnzfN06ZMJXVr0t6UCXMlyJnsAv1JHzh0RgqxB3f3YGhm8WQpgBx8xfnFJbzTSdG+p5TmOUWzNeLfF+xlTDF0CNbPOUnkmp8Grj60ZcjcDQDpRt46yDPqO7e+BfgVCE+8hjYNSBUdlRsMohQHBp+OsY+y6ZW62x1VzLsHNwtUvWgnJkAExRkD7cI9VfpPutbvr5Hm86wkeRoGKEe/3QxjyTPs2tQ+UvVWX+tjEijFr9MFhxP0iEjXfnkqdvsV1b1R9NZf2r5Q2zdhazhTagg5wBVlOHqh4nwtWzxYsZd36lfb2FMJrCZPALVXjgZ5xK5CZmrfw2IYkm8y9+fxjyngv0EDOlu4yt7+GmG8IG9lamQqW7h/GRZlu8ivWNzbb2oIEVU/8jNgxD70Y6MdbUpNYqLBa/B1Ox7LrdeN4KA4D6qFgEfRqxaf/hwIgnt/MbrvmXSvmC3swXUf7vqEkhAbJbB4EucD1D46TZr/TOQq8yKULq/Gr1jQ0L6vJ6E9K3GC4Cpr2Ddd4rCvGdVpAUINQX08jNPDLKVwI+JN/p2Hp+nCpVHtgIZrq0s3CAcUYtpaFbCZbfNDMcqELQZY2srdHDzbrbKZ1zMhM6gQIXpYfhCqEsjIjN8Nw6gE8p7BDVnjLOU2qoINoH85lAW+2QSDmT6mR3Us8ovoANyhB8n6xzGsrnbbb3j0ZI4mDhJ2wfdN3RKXxj3O0ofdA3NDFYrHkPexhnbBv4t+ST4oUR40/HAFy+CUjzkSl2/dL2DK4ShMBhGT7B2VY5b8nJqGR4sllpFTRdx4O5+yPGb9nIhmoT1rnPeUoe6yjYR3WvPV7JujS4DV/i8BdG5gaKy72ujd8hXCE+RGhaMgO+RahbRXRJdFhDl11Z0WGwTQM8r3TG7ww5bvomr/E0dsbzNWK0lFwqfvG7pYq4pR8RWd5AvZUEBjUHKuY/tkSVtefcDU6bYgbXLEa5yq0qyxZhQw5kFutWjWiRNmTcxRaRvVjmfUfxk1xHkzDP1YMbt9Nn+iHp7HV2VJw3tbQF+JIeq+9SowVfKc3VCeSW99Jva6oOFtbY+iT89H6pLVafqOHcHGOpfrrIwpa1ypcRyLkYcI0g1XzyJX1oHzDAUTfYNG1lmlQrW0gzyzS/KQyVh5lgw7HymfK5ZfW6PNbA1h5oeJbYVAu+j+6WNmtiC4Pe6eBtkdgPj05VmzxMWRVbgu6bhRo3XWXCm4n5DWBSqSR43hrKMRcSXOcEEtUF0i6NdTvRRJDNlvOnmUm3ZqOZf6nC4dg8DotWhZzUAWeXmiCbKJcdmZwUqsrC4QoGqNPEqHTpLqcOtqXrmnK09IaBUTFEEopPy07HvPJwhOZXqi5MJhmuQi70AJCALZz34xeL18/mb+zXVpkG1zVdfezwfm4pYwDOTFFZR9pJhtNi8X9DaSvA2J+V/1bZa8TRGaERWDP1Bas9+gn/zkNvbjoC2Pg0iBRSJTXBYlZqjM4cY0SYr1BQDRj5RmMhQ/JqE8pCUIrHaS25UAzRyKqU8tn3bKUkVd1Vi4vnRrJ5jvt5Gs9BVxHe3hc0YVu7yP5rdNgFX7hTd2V8kC1l5aLJ2cF9fpbqCkvrHp5PnIZGuMMutmVovmrxhw8oVV1zRJW2N9bW8IL2ytnA/oTuWcdm1XmnMVZ5ayTK7Aj0Y/E6uekaeEsJ4VEsq6fQWA7YWkGOPknyjgxZ64gbOAYtpM5qaLbRZ3aPjJVzaYdTSvlD9KbqY1VrmIL3AQFVtd+Ye2iGV8fEu6vPSEJh1Z4r06L8jSG0ckTpwJ0P4rrEMe4w0ez0M1xLaGwddJjRpFc9XRZSdRhPeWJeZW1fUS3Rj6wwkrkW3/cmbamb6lK1RDchPTMYrbFeEdBXRvJnpRt+ceYBj7g5b66/wCPLIxVBChy7W8/9VjDwcbU4jOOjZKWfhBXyWAj4PE4v6F6Xjs5bNbSo6j8yBfbd/Tsy6zWaUrd4hZ4zNBeDmPhbjptq9AzSZnMGdsnOpXJJGfUx8iD5onmA7ay8k24nn6YnjZc1Qrhlh/YdhFIY9Il5vGrnCy4EOlhTlgT1m6lk/C0xYy/YOGIGaEw3vVZ8dw8mWg6Hbu6P6fTG0/Z1F7jYZ3iiwE6VDrVVyxsU8xTt06xNQubO0xRpvunuJ0fXDCebXo4uCKdmOZwMr3+JpYvJThn3f9PU5fAIyzHuzRe8JIbKmBG3ttF6z7/dMh/x0wrlkx/R07+tJqW1vW5Nn1ZPnGgUS1k4hZiDU4jwc6Nd90adfK4oJUR3+HNQbv1Kh+A7QtC1ih7AWqOS9IGeV6RnpFpB1AyiWSY8G5ImXxx/QSJ5ZuYO7NRE5oq2AH3sCP3uvzE0bSWGJ9BoNeavi1j5OX2Hp8ux4bEAM0JUqePdbwe8P9tG+zo8uArQIrVaMsF5uIS6G/tsYGXxzzvJ2Mz2A20qzDRSC2eN6SiEdt/eWnIIN+W/ABs1grL4uTLlT3XaIlahHy1AoE4nlWk/9VtkInATi3nDAaTPNuc1yHQzxQo9AA+K2ICVxAuHgO4Ao3p1e86nFLXAV0wqBD9VYqc8T3TNo0DAnGDeI4XNj6DtLvUsnIRrP97IXkmHHT7xD2P6+4ZWa9NcXBF5qoUzMwwhTrJA1EUqLaKL4/uSUB+rQPG8E+TEW4fHRX3wLFD9ErImzqAJAKQ2WfTZyfv03qFnEC0TnWiFX/KemHvJ6xnL2/2WkwOmDQ6tPHm7R1fqu6Bxn+XTjoqxArdkflGshFvNvIebUJzCBDYS6nTWaJgDJtNp7xLXGOz0D2U6YN6f3JZEFqNrPv3GQOHnG3Ox5pd3WbB0cvMs61FW1OXX57+tfgiBsuC52ozUzLFB1KtQyKbKnLq3uH0/V5TnA1e6DYzX3tdiuwMUa1O3Sq3CSOl+Z0ijpHZBQZlYXTeuoPUsuoQAKV8l6rjhYm8WwkfnlA4YFc4PWuKAehtCqSBJXx9h6YtRVxCBFjRJ8lYIlN8d3h0ATm2bzwj6J0K4dmPfk4qiv0YRZDB4Sj2EQNDS2a96KkN8KyhW0yFcKIUj53eyFS8IhQRWqIt8vQk1mlXvQWDPDtdbgLUgWuEVBMIHvRDMlh5YXDRHaoa0+bMBx8z0KXYHY3p2fCJ2WHUPgHHtyTvpvbh91DiZmW3V7MTgVgm1H2HGJfGPkrG9+FBU5Ge7Ocx3+RCBWU0g5Ahjea6Y8ZK5GkPdNYDAbwbQsJETu4ZmQTZm17IHY4RbnmWYh6sgULwnlm3BrKKSnTFkmvcxAuNeDnJdpkLXSeYxvKMAJJIQwl4BcXnL4X/vE5I+hhYfFFlJpLWIG+e1xKaxoDwL/+MDkz1sEC2XaEvwP0MHXnrHYdEgjEAqfmgC7fqkfdvWi0iXu27/AJj+DH+r/uBaqdvlkZbQJS1otK2cnR7C0w7nkP/i9m3gNzdFw5fCWyCjy8VU1Wxt4cfxVqoYSDRwfb7A9GJ9AXCaSTIOKhRMwyJHpkN9eVqlH9EaXjz2SRR/PHyQ5dvDK7X4sUmp3L8Tc24YGzkdVPhD1eP3ZCG5KOrU8M04pSxzFPBKeSflqQRIJy6MMffldCEnwj1ZGUrlkvdjdqA3MElwSvh+rl7TBC9UFzW1LEKZgKfRzyxxYP8F0PMZYv2SLbVxchfIr2ec7z9UGq/+6qTHqgXdMPzo8+SLHyYDX+YO2bmFkUBSLzuX6vb35gTnvXdIlB0GRqq6uiMKEC4gDlJ6dHFHuHZtv2hPJOxdpGm9J9AUvRB+wTe2U1Yf+jCHDGnvlH5V4BoUjH66U1nADSfaBc6HU/0CXB4WuMcfNJvs04KH0AafxOrJ0C23Bb1CWOEt0m9eDsR4w8HNbIxO+Q8lUPWdZp9D3R9A4gfhWF2wd61D0bvvMjiHfEwGmO0bNYLjd6GANuGELOUiN3p2DzgSGVgPjTm+ebUeD5/bLa2vn1TQ3+D6Aiy3PPsrROf8gbfmo4JJWtc1gqlw+Nl2Dvg6ClSjTJV1a64chGuKLYD0jmlwS6CS/PC8ou51uE3TAsa6T8dNgXLoxsF6PLa+Ra1mpcPx5wFdBEAt6yqQRxHYsGs3WOpvBvqY7TfgsOb/g7Vn/R8v6Gaqqt8tg2BxiIyPhZBWRXIC1yeIzKjsDUTWxKQ13+lB1aJH0dpkiC2KSF9yGnm9WJtORb+P1wD3h4eU6hyqvtN8SJDT4rFUzNFVr1pZbN1cmdcJ7OF+K/K1q3bqFEHM8JDhlRHLfC0kuPDLaqzfVbzKoiawIWtNCfg2jXPOyxXSxDqgUGGsDx/BX7HSg5KdqM3Z4cHL5fM2h3HfIcUWC+u8PmS7D31FOH5pe8UYX6Ch3jcVRLYcrEKv4Z0V7UE3SUGSpyMKLi8kqfwr6Qld3jDtNZBgYMOzXRnBTdZl9tIrRhcYhW1zBFk3xKqK5uTZjRpKl9wRXYGnnpnR5vFZ8oqrAs2D27ndLW/lv7+UW0T/0knM5SASTUCHjVs4LZmDuftRjwfNzc37y+uCaNHRV4qT5CKdQeDEZWEBR7Z9qpAjClzXNZQ6ue3WXKlD8+S+m/QrQeOLXpBfFUr03ZtKBLrMoQ7OlvCxdx3mq+EjyWEKvFyuKTnaNb8p0KTdzLcdwXRlY/IU9A+CugYJmdt2lOTtB5txKqeK7jaJDGHjqUmW/NuWMQ5wXGpSqzE3z9RUgbsFj8H6gvvyw0qV5wHGWRv1Do4Fz5T6wPFrGbKAsF/Qp8CrJd7i2PXnwBK24VmStCtsJ8sEpkaakvxudsmV2cFhH8IW/ssNhIFRP1q0LAfVIHwWmGvkjZ6SeItbqTNdBbbCIlf6cGk0qwnpzqTbhAQwRxhQKvNjfEJLwrXZnnBtf2eTr01cCSXAqIQtmlsbvFt6i2wDEKDw0bTNntjP/pAr5qmDd7QOP+d1IRuQ3CaJ9Z6b/nvxvHKPzT6B3EbnDJcqzwV3Q0H2R7tFtsrOiw0YuzPXdmbk5nVIiXfTzPq7AGg1bc0mJcGW/4u1tVDDJWmwxXQNcLPTx7E3dNGlqKkPhoxAwedS1yuCDllHpjMvqyNrcDR6GOLFpUCZyr4FMS1nU7CoAmLiaz64JbbfqvyxcWf0DG3+GaZmiFXP8aYWG0Gt4nt6gkPHgz05V2sCQWQbYvWqOKaxyJKVpumoX6bvAOUpBcEF7Qgyz6qWiA8E+GR0QBDV8OiPKBtVKV6XgqDg4tUoAgeY4o+0ARXbqAOeS8wSnu9mC7CutQlfyCldJxNGqpjp64ePvDp0Wl1HzjyAPLbzw/mfXSumEnDBw/uVH0Q1q9lF5n4JJwOe0q8cI0y6BbXZJD01GUR9UE9x4pXHNP6sMDP1nKUQ/MRyeAnLk7DFvIN0LyrXxXE85Z7FHZ3VDaIYu2yd8rYLcYaXzlH3sE8sXI4I2nL2jy6y1SmnBGMJKsh59Z1Fqdzs3SGQAJr9lPhTc4YOcyrdDW4TBq+8OvY2xCaJ8MTXsZgtAVqtLpbsMjmX9Ihtp9XJ9+4R2YhV/pCxmiOAR4e4N3Z4OfIJLnJCd9MF4T3L51lCz7SqZ5u4p7Y3fPX4rexOsWuJIiPYvKdwYiQV9kLfL9jrqnwYdFEIeukOsm0xPcrFd7XQiBSxnsu4jG0UxIN6m3yUHUOpX7HOBgDfA+pNDw1ZBv6OBC1D6O7F2qXIxzSlSR8gy4r6Y7MHq0Gp1mMJPkcVbqWaRsqo07zt6r2DfVfoZr8t5nE1mtCeemo3EhPWWHm6CMHkkO9hIo9ExKFTLhhsnMO00uWoT6eQQqAGLgntWBGHNfPpZDGTCpK3BoYV8fLTZ5phu3SsfIsGq1OmLsBxZOzv6qvchRCgYCL02wMvFkjvBI/uZCqfIFlp0JVpYMmcF9tIjRObFpYyimxlA+0O3dNmmXsdR7YrqgDqg5rRbfRlRx4eTzqs3+JIlIRlKYU1vfXCr9pYzpFfXSapYX8QeYKBaeNUpZC1SHtxAfU+45uBdU/JDIbwngVdeoxa9wX+6YqY+UJk6W0qCoDNkXlmdEPKgAzNPGvbZ+aMh6oeFBCLTl6Apwx7FXywzGmIS4BVb+AA94nVzV7nVBZ0I7QUoT9iCrNddNhffv3T2+2kJBc7oTGsBSxcqkj3IaDM5d1T8+fg5nmK7kY4wsjbwRoS+DJLzD0pgk8Rb7gQBMK/+tDkhtNJuYBKIJp3GYfafcb/mjbqp5eNUs2kXqVTgQcvjXmuUGzWBxXhC4W+oC7qLiYqAj+Bd2HXYFRU9dEajEmAYxGiy7LqsPhVX6JjBpw08rNDXA0iu02aGKWU/3Yf3X3dNl5oQlLfmFtXV0OsS7rEmRKSVNe6BfaaMBGUWgJA3PoLcJcfuEXXZMmClM1oLmTFbuoJFvbgpuXxn79+W5d/MvlqRdx5zjApOhOsTb6nsnfjbVhhtj6jSCSsJgN4xMQeVXEUcWjv0qI7d3wUmHQO5hTrxKGcy+o1bjgRa9XJRCSbZKDbc3ijLEh17k+EBjm2a7txR5H7Ub2O39rkzm4kKO9s4jrHVJR0OO4Cg7l1/gNdjX9HgvIUM2GIprj20L0E2pCS07Lx6C9OrXFy2KdJfj55ZlnQt3eMu5bBnoRR+Ie5sR3gCP7hwBXxI6mz9oVvCA7E1M2FG4y4C/xKPxyYuFXVD3vG8V7sD2Ipc5RBNHqXGrVKs3kSJySDjb1lOWfsmZkWYnhl6sYbeI/F5FvtaT+j6V7Dt4XO0DXYlvX1jbvS7Fff+FJDDFuIX+HT6Wq/h4RyMxjU6vZDkvBzzrYNJIUE7b3jq9WSyDsYc3R7A7njc3ftOgz2sX1hLRmmeGZPhvSYEHuMzyPVdhHW7LPT7tqqxnt9aQUPWJ3RAjkZzKcdQFUMH9Sn0iTHOc60mf5ZjhNohhMtixoiMRxeY21iP4ZlHbgfVCv4mxkv94+Rt0w24mtD59bOd1mXV8R5Ywhtz6j37IHCWnNNNxQiQuuk8eDnGFhJsztaMBdQ07L3F3XjVeQ1aBpBS1TE0uHXzwo6Lq3WKWAPp5hRG62KhULFvPGcL1xXvmUFVjNyd8GOM3kbzbKR4hzIO9h+CFkcqFkXTbM3qMVjLWfyLqk7h69aWc5j+tDEnUb4jTK1cGcNn7s3n8CY98xeeIDQwmjLZfdRn5/CxsU3VRgXO2CXubgCnjbYzQ6tsj+T9H3+3iTrg12Kvzo9oQr+K+8PwKHXnqNAzgLoBPVoUMM0yBDd8/gK53FXaXFVb7RiVAyGUx0Y0DcTPuAYsWZQ6lGAPKRYfdmufQoCjtrjEKbOGUM5WaZrDf78gNIbJM07kWBN2wcgWbRhtKx0Tpq3JvOMaWR46IiOT3WpVgMWGhmZD2q9Yy2sp5tq4PZnbjyZZkpfk4NkMXOqkaRvh5ImeIO2NVBzXoKp8qqb6h0DGqbXm0NXaXm7J6d/bEAFbg/3cGTt5zVk8Gzqs6DHIhoYiY/3y0r9a3Ls4ux3oOHs5kwIImxhuSvfxCGz4Eeo/hFPgVPEkXWymHBlJXSsMZTPaYjGIRZtgk6gJR7h7YVs5G/0eclu4l57fDkDlTrRxXDWBzLMRvaTVBQ8pBXA3zqGKdK5lQ/hl2vLEjmpRlzfX1Ai7h44TH9f1RlXwYgLTZxDMWjxE+NEUgoZEfgYpMqGsPqT2W/LEHE1J4GxaHC1NI5PoaKHp+fdScs9ruMfOHl7Nak7Mk6A5+y9Oj+cuMaWqW7cc0nXxw4HJIfYm0IVIo9wu2mLuP1oL/LZx70Qb8Ks5bzG/JEkfz3s0pCZOvAzu7NjN0cbhVgGlv0UkqGMuj8fW20VujUY3vYQAcMxtIE8pMB0Ylp5z2e5mf04ZXeJfThbMwnhgG7Tb3Vgx9G6M7ABdrjbPbzFZeDbQy3X2Iz6Jm1dpmSFAnwzVeaBhO2JtB5NGkk1vV+aOGKnKIbBbkZTIVlRMHm2wi6UTY9U1wDkGNczvl/HaQ1e5S82vAsmYc9iVqOfJR6v3hX0xxd4Gfsmh7bCPP+i0uhswMtsd4ba1gtQj3ZGMFctpfkqpGfnbqqZ2G/eHadxWwG1EQ/hqxFuSVdENd6ldc8Rtd7DjWDnS3e+h8XypmfbUEQNUe6RWDK4A9F6FCOHrJdgzXq8TTdFUlYWbGYEydDTH3LqnCrXSvaGEvjvRU85CjWdhIQ8u9M3OU7FiUzmLEg0YnSzFOQA7WmNP4YBP8K8bn4K1MV+8xVwFbXdE6+KVlO+9fHa7aFqpuO/PNj2JkvsIk+mTxvVfrbk3sKPDt5hb/pt28TZ5Yq5AJp6Lw/FtkR4RRG0n+fbOrXxv5Qp3smxt0MGXZHQmO2Lx7jRqXdN71Zvx7q3cCBV2OyHCxn2Oq4ubBbjY9FB0LBpXsTfFXmtr4QaAXOqAqdVyjMHs3rx5LBhl8zcm3wLq0NKj5X4IBnPDf4vx6B3d0RhE2BJpkO52JDb0FpaQ+xMpY5pl+ubNRlkwQUsNBxvMxCc48kR4cv8UNclwZ/FwKs04I2Qsxn5PlxyBTGHCNslVGqF3cF5RoV6W16x1WfJEWL2Nx2CYvuH9Hhvn6eOObBlX+XWeVQ+IETXSAeOKe6kn6ziHwm+cT1PqvCxdh9MnxzhaANoqjr7QyIg9MLVIvJH4OtHyrTf9N6Mn7VmnucYVPOqYTAZEv4+u9gLGaWpvRQ8zSXT2nQe02BwnD3+17mbg20upFSMLgwYpcHqmRAXqimd2+O4C6jC07mqbuRSBglnJRTSyn2lVa7jf7CE5DhOmrqwZcMG9CqWmoNmGRQT6UBupOKSyDAQnAS0VdAPwOdnqKj0fv7KrTxJzKq9KQOjf0pGUT5M8yZmWELF/tSXxunjNPmEH3O4G9dGSxStH3lEZx9H3QFHjOIm0VVWo7XOlXmYBhUkcsNOjgbqZOhhmL5S9pdqkx0tUJ/kHMpw8T4tyB6gt5YUVsWujcVprZ79aYjKJbWMZK37IVWImrPtojbPZxrRZ6+5GUNf7tCVy5iStKTFDeMvAJG0e01xqmJn7W1tfFiasTkSqs42NpH3tVtbC7G6v4cpHdkQOSfTXGLDIYkqcH0Gm89CV5bvCuxTR5DG1XnBmGuZFWBgXj1XYLGlRjJwmbIm7RF5iiQXwSI20RhZRoD0rn77zdFzHNpxxgxpSBwLvSBRimwbm7RYSmlDEozz2mZCwlXQ8F8O2qt+ciDg/VcmD6mzo2PXUWzl7DmbOUUhyjKIqVIEEaTIqZ9X0W2DXFYgClpeuWvAqVIt9uT1vSC8ckJkLu3uUBfqlzUe8fvCt7XO2E15Gr0bnXP0bavrt7DFHxUP5b3sCBdo5+QBDuM40OsHbk5wpVCLPlaiaXf+m0OiIG8RgFAPoU1zl4OWtc0/KRTHSlWbJLTPZm2CqpzDYCwbc4M0jFMLLXPV8hCpdlIj/KsfA/eQmDafLpAz2KkGJgD3XVgn3iWeKT5p5V0fdxmQaSsHtBumTG6yZUVEOcEOwkFHhArE8Bk7B7uFwDrt4JGKwxAYLKxvjRV/S/0G9M3D2UNTcSPKHd+PNcrtCxJQGONlrBXa4cxtVfm5exlAUGTgcQRkoD9bBEhFVUNxRdmyTEjLmWWTqafvJgC6eKxpOCFUqzFgE7DtW0ROY0mXOZQpuvfCJ3ObIYL5F6uIqGAQ/m+bSQKXuj3DLF6FhLtIdkiExJfkWiPPEHwzIpOpCp7Z0EE5gsp+2Gd5b5L4kTE3ed4GAOfTlfrnrQ0Hp/OzSFbfR3bIXPvWpuKaEUHS94H+TnM2BcjXdlVH+kjJp7lfyPcnqNJoK6NydrOXAFttf/XnDKMyLWsGqCA+kiUe+pO9tU5oGpmDl6SOsFQZh8hbECUf9HX0JQVYgm4RATqlqjaHQChVGcMJ1BnRhxWc1/g/FKBJEpgiTJYKE0VjG/ExCd8W+xLmxbNaikA1gAlQSU7n8fDjxWb4tsF5q0IpbyeE+3uE5CiAg8hRzxQgmwgKz6LwjWD6yVMU1Q6g81aH3Sf4FW5ZWMjbDDs6wkLwYzE9v+PImB6b2k9J9mS4YFiVRhjiXbP7qZwWvyV8oU+mPM2Dmtn0dM2+6ZkbxyDtbUW2GPIm37ufpEkVqqZkZr92q88VHZ16nC1X1J+FH/ddCxI2q1e4L8mdZkGUnc/s0RjwFNyhYd6EjRYFq1JuYMYA6K9xaiA7FZcXiqnZZgJDuPJ55jNohKyuwk7uXvPQ0QjWO8QdwD58Nd6SZ3/8sz9WdOOUdt8h3ZrPP5HvaAe1xCpdXQqi9kmwr4jdOjsi9iPVR+nsc6XXgdyGhdfrmXk0V76Sz1SfgbN8jKKKHWbWHaFNNg4R9MmZixwbkRn7RgMc7b+TzeMsS5HzMnPOOJhK8WYMhg0RhvfzvC0xw3n1hmhRyA2E4J2YN1Nfx5Jd4CZlRgP/7vGMuaLS+JK+yrbMw+P+FfSq8vaQTmlnAW43MRC+1n8XlNqQX/xHE1HHNwE0jyOZT4DdeT9Zml2lqwEc49hcL55Ul6lF6uF1rJG9ZPl/8TcdcqyBQdUQFzCrRnmgbM3rBqIs1Nj2mhD0DO1Nkgym/5rIEhzFfsZ+8fi7/2dVhuqpgzrQOTcgCZadOgHlDoDpbRWPY3iL/fXcJbxu6C0ATrPAM1f8UxZB9vemVDnuvdmRFDu4Jo11OhmkLc2/R5dmRrYJA5bEBktqkhbYhwhMZdnNQs4C1zqnIGLpOv0n3TgEHmlB7VNVG5F9BU0wJk+NXH/cqrObH0oPYO4vjp9b6HjJsmWNUGENUxwmMWBrTCRrcIC1IVh8jBnEVLrdZCBv3WtuVR+73KzadlKYU5VBG7UHm2zmCuj0cCkvv6tlzSGAbHfs/1iFJdOM6kOj4Kl+I+vMYm4Bm76a4S0sn1Fy8oBgD89HjqUZh9PIcBl/ML5g2jp3S6GmTFNj1sPL/wP2JV3VKGCUKrP8X5V1+j4aJf12wuXSQxNOJMhmRNdX+HkCFLMqMlhzNj+3e2q/ONoovNYE3TN7gv7xeT3ydMR7K/SS6+e/gqKuAVCNfk+4oF6GyUAb1lWpPMJgqMUSgQh6wMzoXKmA/m1njVBxBH1ZmJ4y2WEdcgPp1MjIPbvGUaChX+l2+zxvoGf6kqYkJsb7xB3vgchH42prVzG5Rl0TPmc+tFhz4ItvREvEw8lwFXfkrtDa5/MQOiRjmp3M3ybBcJLX9b8cPnsJZxOh6Y+KtEwLnNXKC2DTY1tz/s9xWGFU4WEaRKN+f16lIUmGxowIgD5FEoWNAp0exFVI5oPwULzZndg6Yhw1FRAUB6DF8bVhPPQqHOdQkpI/NdAiFqNHBSpbBXaTzbvtGyO+yGMOwb62I42U/7NoVUdtCVYRp+IajaGKzOq0MKiTkkb2EoMj4jzVUnYNhpGUNlTAuWt8lBQ3GUsyb88j+MGdNw+MgHi/vpqEXs3eYDPxqzUH+2AJDqUkTwy3vpyIMJNYMhjd/VAdG1oWWQCnUk0Bxec4EGaF6nOSaDIp6iUq5hiR8c+5gyJ42hkHMQoxzl/gPvAFuLn2EY3ZJ+dlBfOyibUmSmHqE4FjqBGzwrAV1ifIadPjt9LLaP3TmtIvgdpu0Y6R/RMD2QH4Dp+O21xKE3G1xj4YEyOKE43lh8k6Ht2uiUtz57E4R0nDwyfPOuyL+k+mXffkEGHu8mT93AsoKj4RR3kFqpHt5/LzTLXb3bRT/Y26gsdkL58/eDrzO3vFNYd7oy4ZMvmqdO+YEM+1dK0EwU4xFAz/1lXaEkZp5oTTlz4+AcnpjELHrG9NCKEqMUUNJvRb93hL1iatS5Q9QtaPjk4mZmU4OMnlOG6nWsEH2BLySgLyLcPVCo+3mYFlWsPbJzj7/YmGT/HpMj24Ao45vKpE5KGGS5qsJRZs8DgZxYJXh9mAMuq0BYDGgbN7DOH1Fy0A2fcO6ptVdWbx6dpGV0l8V2Ej0Us4DFmS5ZbW9v9/sNQaT3kWhTqsDNiFwX6Eu2FHoSgVzCotu4Wjr4IaMSu9NzjEmQYTFSqHOzV8hKdwXgBOIZxWxpLwvUHKr+ht0p1TTYrNVzcmjKT23FbKS5mcTdKLnH5nQUZOd4q6IIsb/hZi2+2MPKuUT6z5zAFCzGJaO9N+lZBtqZNO/1K8m0aUgMzO5qxMnLfNtnRN56KyLDSsz37kqdpsEP76Pq4sHzoSMpHtk+eHq21mOo3gqN1pCNVR/2nxcGjjhvnZbq3uFgOTt0V2CkIWqk+pqXnK5pzFXr1rWQH3U3R4nBp3Fz+i7e0bHLkwqcu8k9Hb2VYqfJ1SOsmfhcWKbuSBmZ3WSxa5drsK6pUtdQTs/xbuUT2Kw6dLKxXQW3uy6rgJmPGW1YqDPbWp9cpVRxbXGoNTsfYMV7f1K1LEQtmHYjklLFlvcwDy0cDk9MyvBv9ccFTYWgDQ+sQ8wNWw4XlLRN/9zSONx/3xN+zy4T5r3MuRdrPPfc5kiDiKYVqI18Wp1D66e+Qk8XUFKdDDPZaZ0adjP6xeU3mh4h9KVCDuxdZ34cUpvIa9HO7mSMxO0CPuh4xfX8mjnfYnhdDjmrp4kMnpjmS6qAPk2TDTgz/nqZkQx9zLsNb+6M7UBGIBlxWqAtFa0qeZ01mWkqzzejo7+mcmH6k0cqjnC/GC4jC2zF0MAxJBQBaR61wW7hsHv0lBLsFi4dvPHXl1nBWHZKeNRyt2ggBJCfHwCSrjeudeI9vbg1s8h5uX5xGjXgjJ5q/cUt/0MVqa6cM4A8FgiXbp3LYyzdqkBdzbWNSEt7OVQuiGEhvW1QGwsIYn2jtLzPIkyh5PwfEFvWKwLEuyP3XBsf7lHdYwpPQJavChjnXo2JbZUMq7Ra4X5gpiqCTmC28OUEEAkBk4MZ2/Ni6pIgIuT/ZuX6vo+0NubVaIozm+aqPeuPd/NdnlqeK2mnp7jTzg3oGqbZV5lcdYL4ZqIAShiF6GjdW9v5HOJC3tnJ1HoKXAHzX1vw1mmN18vk7+k/mYCmKLYhqG4rOpMMw5z8aVmq6n3UPFOKOAhtj/QItZ8/6nB1hMOmb88Uzk6P8pbCLEYh2NbKG55ehPLRvX2V9UDjMHq+ar+ExlKhvr6+GUgoohH4wpAH6mXV28tdS2ZMe2sAyoq2JGD7lSBSS35jItF6ke7A60YIm6oFL+25B4zKhpknYcXQFj/FA+Dn+YvNCdR6T6m5FwkCz37+6wpSKYBmObCb6LPFPehjVqzK2nO+t3o3u+7JS2hzDTyTkZVoeolcE5VP33eSZeFnd3PumoOPXVOJgBsY+6bxIpfthkgBjBfEzf8WYCXFjumgHh7cGL8QF33cJVfWgGB5V7yIv4OAwfHK8h0K4hb6iZOH/1eJzEjCGrr9AXQCfvB6VNIx2pPoVBk42jl2Jh1DRaI/yZF9MvEvaAdvbOuD6xBSuIVcKrGcH32MybtVQ6gjdcAtIs/5gbHnUvFuxnZv1dMDBXHcmjlxgYHXGWUoOlgMUHmbqmH6UdMBfuhZ58hdsdvvmpZO7pR3DP5/f9Dr5vhJd01bHQwnhcVsJjgVO7JIzbOEm+lqGWL0hR8zEQuoR3Hxmb8kfPUSZ3DV64zFcBgMKQYFP1N7DBv54hzN93dRhdApchOc6CzJsbCaE+EQ/mBaYRMDyadmnHRT5MeYrlRkQTp2ZlL1dvBzE9gf4zfndOCD+bE0Ysm8+1OBsR7xmLh2Kvnm+BJE/n99jD31OWkVMebRA4YCr3Ly+dyOMr9Kw+JrdN8yvxlPk73jZuGO49ZVSf0W0OT6uUsI+ISX3hOeoJXN7nXDbbvK5afxAgnkdFOXR5a8c3w8v5Bnj8Id4Vhb1K9bsJ9SXV0xE/bJ2U9lE2FOZsw8MVsoudNnCcHXm5SLuIKvLSmhr6DOR9oVntTg1nk2X3nmE9xZQ/D7F3nzYvhD+wUVd+wmfn98hnWgfPdt9Qu1B/O2QdEOnzX+v9f1ddxjX6WQWiBEEwRA8xqnmBJyj+8BWCvyWOh7/2l5gxjVvttyg/rmdS2XdaRKNtUkqzSVbXPOGHpqZ7Bl9W6QfO5dR8rD5DvTNNK4+7P1jDCbcgG6pWvAF3B15Ns6GMVgcTa8aewIk8+BHl15hxUdZDdy0hFO2o/VnGUT4vvwIb3P7UYAnuEJuFPl02Oqb+SFiGyfA0i+HBIGp7xAobQvK42LLs/wIXOMRVGvL0RmdsugpVZD7AwjMt6xr3TzjlRlw8KxRBzcmx6bX2Ru0XPqrcrlb9cJONHPyWu2Hhx/d9LQufZOKLuP8iPtNpAshVdNrcrNnOOzFkNFZ/9tOxZwDUVSyu6DGsX3aa9sqoawEUdWYLdzmAewg9SN7sE02ht61eiH/uaGPN2Y8sZjjkEoMTLqYJyXmkOnQ5cPbVb/R6GtY5OO1oR7rqRN9SXsNc2lnRQpsl+ZG3a0nGQeg3Yw+XBWcbPYFJ8Km4Yu7VoxJtI5jNZzVoCVubLD4pFqjYTVWzJewQS+2mVnuzL3e4WEh/2XxRVjcmqtStrsMPrH+gmqmqRjHAQ7EwZ8w7N3NPFYX2QQNLOEY4PcowRfRG8bhjRrvUNZhoIi6pFXUswLvmpGwwBhd8cHIjdQ0ooB8NEt/jWvfyVxGsH43AUCpbUibh7DrWsqMvQAaOH26Z27kJRKr64mGJwXkRnKMEZ+YnnmGuaMDzXNjtW2CJKkqlmqAVrcZqqaKi7IXRfhJbgVFGUdr0Rpj/ZbpOKrBjlo/GbvVVDP/VSrxFMSLf5aGajX5rZHTjqRY7ibxS4ab3Ge0M3rnj2ALSE2RX1lUexl2MWYWpIQ4KVtjGhYKnS8VCFW6MKy0RkaqVhXt6Iet5uOKwjYgEOJ3u0lRfBALPgOlgt0O6iA9U8JM7X9U9fhfKjv7p6NAMbCqQKTzXT7Brg3HPmTbCjoo5y4SlkWt6W+NxRRRBH8CE2Pk7M+rywOwtVoaTXIeDBTc1Kjo9EIZHBGW+3kKimRdzoc9EgNHr6Sxbk5Q2i7YF3HxE+xWqhjqnjVPJyPkDtG8yNTLlCuCFl2jgHTdwhtz0Ki9/5wiJh06CzCqW/gVSMGEX5pIzLqPDqGhHfS0cGtbf7atMHRkjDJUZY+ADL/2iT37wgOuzrgMnMvl4ARcTgYjZdYFJy2vFyJnNI5eqxrJtKeSxyk3b2+aGnpc1zoMgBVDbwk1w6YwdqBbgNBu3RSLeqx/BgKbehfT4DhXLJkZvN0GDgT8VuIvR57RZyKaf91T5xa9jbPEs7hPuzhCpBe1TCgoRe6ZF7j/X5f4FFoEpGnsE4PrUXDYW+c2bFwLMaTj8x6TU+o8Jub4nuSG/ijc9KGmuuJh2Q5FBbxdyZfWRJ96d7lEcEuxuOe2GoA8rwRwhbd91nkdjTgNiEm3q1ooXn26ZuDBQIgsd7vKdGdhnyxzKBq7C2f0KAFSgjIAUwc4bHdo+ZrLiMcUtuvESoL1Ly6MQo+9iqv2GMD6ztg04OLdOZRPhvvxxnUM1vvJMMdTjRU1uR0bSHoBLMSVO8DGTKFBdGHN5ZvLFzBJMHw4DJF8gcwghtNfGS+kZe3OqN7L36bCBvDBGF1ffSSwUKahmA4GVYRzcEwLqnW4z2ZMCk+kBEd1F2G6tMlwGJYgAyEeVmjzrvc3PQmQPCUaMD/5yiNwojNW9LlB5nsMAvDGOHHz5iBnxwyUtpP9MDRaeySLwWySzm1D7SP0NcuXvhdXi2EgHTVAhRQSbGmrtwOSXSrbdQOB4Zhp9Kzx5ImHmT0TL1FUXaW1DzXUD8xchHPE6nD/lKTzHNBYhnE+ReWMiSq0RDaDvkTwHmhcyMXjf9JR0vpjeiSFXdz3TfXzBJ+wzBOrDSvdXsW5PrsAt7o3Q+h+NzbVzAYNRHvYd+yPRKDjArOEFOwIJs4c16sOvNPJXM4HY+xB5vypMHdKHuXm5UVH5OcoF2+UHLJk30P/6MEQ8d3xPMCm70TRpxHE/NSLyBcPOqGCc5GBM5aTcJ1GLtOuJN1ZMzfdc1OnryvWkiAuBIAZ/NFWh/SfN66S2hiJMjdsx46Nlxb0ACfizCjRG8nl2xTbkDFaGPrknGvrwMXcPxMu3wrrPkGv2SijB1kuhrb7NAGCe9U2BvlvvfrjnpKl+v84t4rlIdyDp7mpeNd3tphTBmPCO0BUrXSbZw2BUkzHxvO3Fa7cTiFtpd/4Ns8+UOGcxGeBKidUyDdkdWgr0pDP22JK+53R5zoqVL+d1DxcyoKVJ5P9bcRsCVoK0IZdQ+ekj/XnaWtTKamkL44bc4sEH8SJQjK8TIfebPRBp1J4nfUUT4mEKqCqGB49+W/ZrIrFepvw2pzNzlpU1J8tFqvHZAefc2JUmbsK3IG8Dk5X/2pp1ywebKM5CbnFgR2YnFR5tJ2gvGpZWbmnRi/bEmHV7+3T5txD/DDh37RqKnmQI9ZGBTP1KhHT9bo/AGsvqtkTDwgqBO+CNNrzOhyF0yXB04fqwCDTyhbY7qwu5YAyzk8Co5LlkNibR9h9fpymYAR7dzvANdP4R1xPt+7iQbl0aEItu70ke6gJ0dN6dNoHtppPJYt9Wd7KyLR8PD3fx6/O8cmDTHkq6tkUVEnlEjOyUsOdg9SQqaB8OXWPdMjVyhqxXyHCD4PN82fuHMu8V+DH+NJp8qnx9ufw1eCK+Q3HfnTpj5HIDMwVHNxhwsMXpBgvNGJ+fYp6fYXvx7hyrjObBkuwJKAQCW5Y8gIVfVLM5AQiLbTmUKA55kz3ODYTZ3gtJJH9IDGxtyVvF71PaXeVAxyQ0VlG1rULpM7JmPGN2aMzKXGGNzNm7I6/g7kWOPnqfq/0Un4K4x9Lx5BmqFkVPhtlsbOtNJ2JCGgnrVuUCQjpAg5kt3Mk8n4d0k76x60C8MqMZhZ1L7KVxZhbuhqCnOVg4eWUQDUn1eWZ2gyqSPR+eIrLtlvaPAnS3KvDuBbjZc4iOAR+ha01X5J82fJxd7oS4EHJhS4XxvpqBTzw3lPgHZvK9RhsVqLcZa7fiBGdTQMV9EK/D3uKf1Ie1QBjMcFowYs2gK9SXKSA1/Mq7RK9Axf8PVoaqjE6iLkxH7WU+jwRfnbb9s+UmQwGgKA0j7IJKN2qXmA7EK/bFLwZX8/KUkVyl9dUhYJh2hxi6DqzS7Zf8jvBtqr/Ypn/m2P8hUv/YKYOYsmWVLeAy4SnBb2PT5/jdyhoGvY9RXkGxwugXSBIexfNWBuiUeD3DN8q+BiWt6BsYqkJb1oWYnpeV1NPo29X7kcCIQOR7nBVGnlI5GnXeNKEJzdV9idw13hDuaGH2vUYnvR8h0S8ZXJsSeAK1OFaNR+3ZHGDp1nGCmdUhGNGSUezyFDuxt3yL3bLDK7Fs/9nYvMODq4Yjc1fGtrnNFRkdtsbc6Nv7Tg3Vg/J7mIHaUJaaBt0K6ISkfOJfkYp31HFnqQMdxPPqp8wa6KJf2hX8/t7+I1MONPJ1dQhqKaLCIyBTwdXZyiFJHmhKrkeVKp6tB8d0VWuUF4uVBfXwsPx0SDTXK8TPjzZnB7a5qr7bm0hH5QTtvQN5W93G+ErKDBrvDLMp2pNHsq2m/K4y4VpUPqqGmuNfqJfRuMrEKmOXgKDbLjSpLz/3e5yq0LiBozFcI8KeR9Xo7vhXHgc79ZNV3W8E7BUFk0EHdeqo1i1E0sc1PWiMQC8e3oQi0JsBnc0yZy1YcvB9HJEZuBpGXDzrFRRY95XCFqwhMrmOaHTfJN1ZA+g5XkeULTcq4n8U2rmNKF+oyqQXhbCFSkTBe56Q0rqnN1q3QFd53E6zx11fvI59q+Vc8Fj8oqQu+gkb9oLMAHyM7n9OyO4sPfnAuKjWbrx8X0X0wgs6CcPT/lFb9UpYQvK4rHQwujVJLOr6hEeJEymdQN+J53fJhQKKK7eyu5nwc3PwR5cnmTf+t+BAsuDTYuMrYdTLH/hRl5IUz2kuq8sJVIIatzh2ZoLNh5Ab6hqp5rj9Wr4yOPm62F+JA/Zbhqn1sYkGH5PhNJbXMb2cHfD+3idjJyVtGRFdwfTTnJZh9tU0SUP/j3OYKVXfM1amYXUhEBsUJkhsHZ5/bwP3EmOtqqrx13mCp9IAcqLHhmW8fF5A7fSt6lfGHDpZIwFr6d8SV0uPpJrfJ7kfjQOu9cnAZ0hA7Y4rFTqRTrRjnhhBIpSc8MUvE/zqkTTKeV6MTsXpaMJDXb/7IhEIZyySM9IgYA7wk1cTwlWDJZVbSWSXNWNfD0R1h9hNlfuBlxEsVaMSu6L+WvVl92V15qTi3d5vpX8xCLf08Xh+Mh4ZWEJx3fpfvSiomJM7KqltZhvg5xtcwV9S6Fg09DlXgWbGPc4wGVVtLB30JXMBKvCSdTgvGC9/YZg2bR7S4Sg55/cEaILWmnb8pOuW8Ijq+BZ6Q6Bv6pK5IvJtW82QvA3TcALWEKzq3emqfgsdEk1mHD3dyF0lBzIXn8rOBjR7Nj9MOyzkb1kctJWlfMKnDGSNaJ/WxuS2J4ylnUEhXzrpPqHBgWkk2UWmQJfyHq9ubwu1tkMgnW3o46fw/QtxZ9K/ZBtKtQRhTsLLu8l9c0/lLnRUqHKHs5k5kPHFMPZ82JUbW4y8QmgYfr0FmAZLUZc8v/zKYkt6UJ3RwaZCNbmpCOqyTuyiDlJlnzYvimvJyzAD9ZjLxXktW59Stt4QKPFbV6Sn4BxaFtaLrQF68/6HOxu6cV/AhN2MMcQoEvVzYL43hDYGo0Lei+hxz+fMQ77JzmVt7Y14z+P0t7Lb+DEddndkRrZWJc9X2C6b7GngaNvvgLWzGOMsUUjY8D+zb5UtD7eG1YDC7uN1d17yhNPhNNyl40VqZMEGwT03B+f7V8HsahXZsG7OV7wSH9RVFMXg1vHYAvsk6ROYp40tSo7APwoxTN4rQWEW5FJsqvTfsQj9A+bGUDOAE2glCW1MOzs2ZwafCu5T6rXqvXFDXY8oPBOyUn9eBtNb4PZYfVRAsnPC6W+lNQHXqWTiXsLxOwMBSYEFwRoUYjDbKlJ/pgtxVIonrvKHOV1sHYu2NhH6PGBbwc7b9Xmd5iUk9FdIAo0ym3+Nu3nHKcKUVsWNoNawBvlR4j8L/I/xS5MZ6l3HNQZbGJyG8CfpFiCp8KoJ0AZJIx7v7ruXb7DkKnCHLr9Mu1GdyxWS9+ScEb3PO9nLn3sM0AOjObuRPUp8JwR0B1PHyTs92TTnkaNbZ9xrn88U7CI5VZUFFmwsFDdwYEKm2L7XZlS3HTPP/AQtrqiMfBxUTcPt1+sAbQGnp7wkMlJjvn+VIHPjxdD/AnCxqSmchXawQf86zCJr/EcTKcGF/PsOlwPJ3Td+6Z2PCl+oB1QQJBfh7SXLlXUqYhLt7Zb9tb5HiNxPmNfY1TmjnuOtb7dNv/Y8gklVF1GwoDWIjDa8bnmKpt4DCRKn9ZB/ltrXlrHH8yt7oc0HYGX7Cl68v6uK5uXcAJLQ0Ko3rUvaKJd7hSS39kgM0RP0UvOsVIfAl+dNrhAcDQy5SfAc9jdYkZe/dB1G8pZfXz/fpT4unOeYvd+Qhw7vsxkmHH9L4Kv3Oc4qB6Ku0vdFcSo8fXH3p0HGQUfUOSUGYTuq81cyHz5xSngZljFaoHpBebT2Zx26vJLzjP6i3AuYxTJ91GQAcW59JkWFoI1ZRMyLAOuwCMK8bbYKGtJaNHLQFGkLc5Kvz7Y4KHjjCPk1+4A3IYGD7xg0BpcsJUUi02SAE2+p2BlgMSd0EEedRl6i0jpyH4TMuAw+TVzUnmYnrL8/Bb5GGzRGtXZ6ZvRhQv8+KgkVRc+gVNaUlHpAJEC6bPUmzRUGEpxlWFTa9SytIMmn9HzLKttD6A0MuG2VgbFvSAewpWQ3o5gTMHdbN8IyJamuRZAPaxyU8fpAtqSaIdcgkhRprnlUYeVAklZ4FxBYDtAB+XG+bZ7Av0c8yzPhDXSZhq3+Vo2q4u6rAhdEMqJjyT+Cwpi+UmkA2Zh7xz9MIyukXcQ7RZ4X1nhiVoTuxUGIg/2aelXx3A53b7AnxaHxlumuQ7sQ/0pbsjZEx7+UgXwyXY2bsFY5BXcg0YP6GwMBz3ZxuTcqWpg1LFlexmtFtx93GXhXvOnaG6Ej33cYsq49eowrpVC0gL5RTxlw/F6iz2GF8vUuocpDpqwV0b6Sj5YLW/OHHwWyJeqHOX1qxK+KX7QlDn9v8ka1Skho+xc4PwSMbxhu5AOeCE8AkaU7v3l2DUwGdZIw2MtqVTqqmhdRrcf4lhcnG2yrC7mKIyMhqHmTb06xw1yThXTzaIM9VN2H2lzBd4pgiYYDm4fzRhbBQCcUFWsaUU6WKhWPaV2ZS0uVEv/fr7VSmCl2qkCwmhUJOwOhRgmKR1Xqey9BhjmJu0BsV+vJWtaZIcS7KkKUd7Iq1XBtsUSiOHrOWZGNa7w5lKfF4y/1eBE0GcqtsR0UBTmdTlmnhI6Q2f+kO8HXYZbwf3zgAXA4NMsN/Ll7R5XZVBpoXAaSmTAy+6ADtoL8Y5d/R3iPaWtPZDyfw4H6bX5xJ0ILxRbnWOQFQZRK2h6RZyc+j1LM1Ze0RALxbHHP1E/5YVMyDGPDfb5uZhHYQ+DJyv1cjDFhtMtRDoOq3i5Zy1S0jnmSJiGbr8OwIQJJ9kz5aW85gB0vodCwfpRhLzgXUqiqHrU3uuTIWzzakbtc7e6KRBGkSVJHtgkbAWgoFHFPmyaBOyCdAGQsNfVjR/L1jjhJ8mQKQt8P6oyrDaw6T6AtsqxvzKvGT89p2NB17aiM8ujxL/bZFh3T4B5wMXdPxn8ybMG28wHEPbLbl5R/l/kzUwZPpqcFUB6ayP5QBp2Db6zvL8QGjlSVJdhKGmo3ZZuaW+ALTkBlIA/6vbbCgT0tvnRzCz24T9zjLBXqdqvYTNAhSQ2O0ZtAwL98W0Yxk4HSAzT0XN/i7OujqthYmBVFhUaqLkwM8HK1IYTHuw6HtcS6mIapDmT2DE7sXgTeuFY4BC9iCPr4XGqUWqIomxhfdsQAANNj7nBJqIkKfvOEu3VLn8XLc2i7PzB1z5Mqnu0EstFngtKJ4aafSjRpCLr8TU0wWn2XZapKfTHbnsTkMT2upCzhmjvNKEHfMiWKYOu6hW5gSe6F7UIhCS0HrDIEadQeAj9aWxmpJSMQrYu0CXVnvDmmQVtSF7cy1jTF0rO38nyrmjUjf2/v+GHPMHPJjhRhOdbbWcDUM1I7dDJ1D8/XZlAKJZWlYwDAUhHbJJ7dq/nd4E1XtEsV1NZyO9MpN66QzpD3JdKxGOoeL6QpQuVPbFSlUUBTzY84jv9ESHMoTYvzt9iVNxHVxuJ7o2YBZYuMRRZk8sp0+2i3Ep6bZMR8BINwhE0y+nftVoNnyEW3W3KJBcBbiYWHo+JV2rbzp7VnSAPjuPiDqh7/EVnnKuJKH6/BNDDf7gkx8O7b01yjI9Sp0J5J/l36Q2Tbxw8jkDRFsjqF6Fhx+nEVMWgRlXAa5GDqylp0PZDuFuUm/NNBE3LHFMudqEDOXEP7I/JRE/YDI9pJtSELV1U3bSVCeadFfhLh9Fd4zyZ5Y6eAh4SAOtjPjGdsAN6GRSngVxR7ppxNzkLYbP2OODozx9XIPBg5upALXqY7cbOVljnSNUjcD5ekGcNuD3s7ybzazq0K6vAczSpf/Aj2JRTJa5BtHk+J5wLC8G28ZA+eiVaeH0vNSeuZLUQvt0qMhHM8HxaFSCpidWH9ANzd1a6tqaTtOUq/CG4SZe/l6tmnGUlSG0Jha3OoUgtDqvJ+qQJaGVHcF7vFaErI7aKpZZlsldz3y6pAqiDXmCvIShD5Q6H9nQ+/Khq3DiCrcmc6Rg0vrdcvaTNBjUGhJGgFcKDTWUr0xK4NV0ObT9Qt/QAk8gUMzg3+4nHXW0lmTmeLuQgNn94Zce3sKGWKSwtEsd6YImv/w3qB2D7oOTeKUdQ6sFyUqEsmi4t5qCHPt6Q8y1oSR6XZmOhrZhdmKN+rTVcxhHxpjMzN3oL7ThHBGjfxgB4f0dSMkzKF80EFTUKCnrVgJpe+yeIFhgzmhAFMqkM5bD1v0R6g7bV3+JsotKCt2W/yFcOSAUunFMJjoGY+MGxsxMhnoTl8vsqN4Bqt0lkfFqRgcxkwaEBlRDLjJZWZYVm2N5pvsM6jMOvw3p1pSqXLoUaPCBbaqG+x8n9xrjbE1C/cHpkXGxkSrbeHpYsZYMuCdPDPwWPO1c2M1Fmcpo1TeAN4tVHPhPYQGr4pbBvK7FAY96tZIgzb4gGPW1Qdu0D+qnNNQo5O2qnWGWrWQu/16K5rE8anV/jIkcGswZHhbQSUMqMj2Ui3nWu6ZsC2PTbwj3Dy5SSoLGyaCYTF6H9TIJH43AzBgCvbk0m7OPHjW4BPT0+OR2NMTXjBgYYePmlgdPpUkuUw7bBM2FhrnXhI0Q8aLLpE1cefl8z0F/nUmkVhgx2kz8HyF3TWFd0hqYVumQpx8g6v7wAsoSB4lZSvArEukLOsCiSoV8qQE31yFbK/NVn+RHCjPq+k+BbI/Mtza37zg4ty0BcqivUlDK8e/9r3BljDMOHNVR5jSxtKJ5Eh8LVaQ1P6bwPq4zqcPohXItUtmPdFWBkb1EZ2WXWr7B0ARD3UZZlMzJqRsXuGbNdAXsOy95SBkWYlXkwlxr8D4KWzrhOsBuMfnzYwMgzt8cdG8DHfhrpiOTY3DDM6BbjB7oxy9IytkVqlxCprZ6wxG+SDrbHKdmAKChgYaVqhG0cM4hGVNb6BC9XvG1IFmH7RVOBDfFzFzl4ZpnBmLOxMGNUdN7vyZCxCrIPO7d0QgJu2TpqtOG7F0LZ9Ff6HkqXO+uJ7qNYt8tBqXiTZ5Kb6R9nNmjmRgSSuQCbP9bDDi8U0Ui5O4ctSJNnyqAgrGNuh48YJEM6bafZk7+g1LzD73VRmvTBtTUEWRbR0Oc5a5f2DK7o2JrSmSO4g+uN3pY/e1Wp4smC4FjCS09ctBX6OXZnkTCcx5kb2xs1tAd/L+9wNj5Q4ek5wuJXj0Jbs94XszxjbL4pfq7wbQYKYfWnXXqRXOtRBPbwzbxcs6gF6zJkHwjsMe7KaMPuGSCoMyVUq5Qd8crMJ6I/yk7yOzhIVAWQFkBV/eJ/0YbjsMTQYYsgiv1RKIB77L0GhnQ6qoMK0Fhd5tihoQsKLMqrDxWjk0jIKxc3s93pOmY6Qhlx+pxeNSHoQ9OV6GX0+M0kWrdwhbHH37GEhSUf/32A0hRnd/UjsDzmdAYI3EoQMtmC05Fkh/ahP7j1oMhnGkSWuPbbtx8MWPtKu6cLnSwkd00Dd0GO+Pee48XjcQsMcfKLR2Zn2VW1H16P1SsrBxEDz/hM73eGKwIR+9j0mEuBSZCLSE4gzHAm+cLjGYOf/vB/bCI1EU/HPVV1Sr5Z6UKk+nNH/JvnbP6ZAsGX1Is7K/TU6qeSy0mUZWStNFJSvz9kaE4c4kQlzlsCDzROj0GcoPjbQmKtPiUCPf3EtEnIRDcCRy0LoXfBtb5NsBVwUcS8TPNgDHq4YGVxgv+PuUGNJ2ZjeK85X+4ZGYdmm8MUKuAocMLb6lxxfXMs5o0xfYrw1/5Y1Nqc2vkMwc/jcM48RFOdAobPnRqzfCqpRr8PrxLLeaJzvHUsS7Y7AQgBTK4v0f2mjA6D4fJfvAeB9Yd3Bx/rsnEqS1fmU7y7f9QskOIyDu9X1q/CJHh2Tk7KWTA+3tRDeZnMxZ8q+GpxQnzzi/hs9/YF0zIzkGCd58tDnR7Ccd0RUgB21FP+rhsUMRaGgevmQhgsb3RVAR1KDr2qd3J/aQhZcahH4ncJWi/ttgUQkS7T8JlzTxfaYu8YXKtiyhnQp/qHwQNsWmrtEw/AmkvzCWPyKaKnc3IEVjeSPLnLXgmjf5D25Mt7GMu3L6Q8tmP7rTiw1VMW3V1kEiFd/J8dmnMUlTmYoASzZaRLnxSbcZxF5HY7bpF3PaGJ3/T7wcJ0aTLdRXr9CUQNZvMhQsmuw2ooOlDtKyrtFhn2AnJqFOfmKK9cnxLzjq5n9vrLzsf9G7b+7B2gNO8WUoXG6rgkxT02aaN5Qjd60CnMDjddCpqNZe0VKd5QqFbq6J1D3xsWPsTD3OP9nsVw6BbgbAgWJFDCeY5jcdBJX6iKO+XCjWdOnNlN1pl4ILBmD6ofR1tL4pnxVmGdlBBpUPHDzhoyHnOf8+8tJqHfZLRR7cG3Du9/QT1E8UMcpkkUQYdtlM22XGB8EeZB8LLdx6PEAkpvo8Vb+ZRqOBRnfT20EbBKvXKz0jBSx4uILZ0RxrGI6IyHXRM7EJ6y3Ggb+p4RdDHr9wGWGVdd9hjp2tJhY5LRJuQaEV4D37pJldktPJYM9AaIAqsZ4y3lnzZRy6fMK8Sxc/rwexMa2mtwdlEZCnt+9tQvv9jcvyrBkzA1/FLv3pmB6vCV0zvrqCIJ1XDrxXIbM2NG3Mx0wq1tQcfGvrbxI9cp5MgGmB3Pt/bXT/yBjDxKnzTXZfpq/PfgqvaJIXx8nqGrfrNg9fyuaKmoCtLYDf+a+kRBDnvOAWTRWuPjxtyv8Nt3GFhVWpdeNWdSJUV1Jm5AglGrFjAAJtpkEktAkeqG3rT7kUIqy0Ha8HcRzNwEiyH0F/1OGn/5pwsuI1m3bGPbg8e0MTi6cqpAwRCNPE9mpQL+8Vn0PU2i/I5N15fRAEYhoZUFzqLZ/gWTBrF5oxPnyhAv+xMOMgGffx2rmJ/3V1cOMb9wlFbWPbXTpf+nI56POnmYTlsFzvbKxOuVCJzCnDsn5FcgV/72/nu7O9LvvjoyC0PXJsbSs3jHBrxLmwrZBQb4fLdjwRdQImQxLWa/MN5j3lL/0q1LBd2Am58UNIIg3c4ZBQe/UAH8rOn8MZ3Fuf5kUNfQpeWtp4btgNF6gB35ifvkbU4NZZxb1QlkPspjzaTxUCm3gb/YnqNf+UVwnukLTk5pN0pDotcB/vkBXdXRCG7H2edBI1tSoFrlAGxHRUjJHZtZSrL/jtP5NcrrdcnAKCMF2bXIEuQ+t+N4mOQYsHoqDAWoo6rM6RfZ2hXvkxYcqMZEQodkKuvUpJW2oP1lyMrfnCd2U4g9xQrVWUqC1+Fu6w+oltIVWQ8+4VUqXBxF5cNOfVr4OX2q5VlJ+CmJgMdymX3YrA4hZEwMmY+tSLbBHFNgMkCmqMiLGlMCkakZYQkR8ZIffHLfWR5Zy9x0wWnpYMGjv+bzV+8ARn8WUtKCugJEhOwm8SlGVrVrrpDmn4NSe5YCS5DVUxkAQsj3sG6yvb6E9sdkFtw1j3rNyhLT0cSXNlNqJ1HKcOTYeErHbj3aQJ7P4s+YyjRZrWbbmjRq4RIQxgTyh7qBZvdeB6Rk0ppjzvpeU+ASHI7RQ4jLv2B6cfdcpbpBsTrxjk7rOrR/NzEvKrsbE6FeDhvAZcJT0MzVL/3lWOi6iyZZD4crgOJAbkvkDkRcFSdhpgCnZuSmG3Xwfi1KYULEJrhVNvuRDAVjzhzsKl9lR6hZGY3HUeOtQ5sCy6F6lyebUhYf1FD+8TEFSJbxbGgLRdqB3Kt5wdZ0WqeB/HuDMxXPBPHSNvMXTm6y8N2+aoMuiiYA1NRWVW4dVM8Ly1uMj8d7bzLeN0Zi/T7kq/c4TdX8wyeCp9ZObXaFjh3Opm7bWAkZMYDrwtm7ULCuHUwlt0FdwfZkRT3M69WdFM47WpEdo7BNPqgTX1vzuAPFoflvCoIqS5XN/EyCw/YFyf7blvpYZ46Ck48j/qljBrCyBE/iHxs3ZoMNJOBjYwkkudTQUpLY9IptQw85zhtI2WPqrZwwYEDJ2y35Tf0u9b4Q1bOGfJHk7sFOlNQlzbVTWAxVHzP9h1qVUa5FDBNi+i5qFoLVmsUqIqjxlsUh7IuL4NA+Rt8YQH0O/SOvlJ2LqbQBujXrx7jq6C6wPP+C/1LkS6bnEYPQDYSRngdwn4LPiBJrfmMT6lDUdbJx+7QX1xhOcg+QR+1n4TaMR84lTqtcSvQ/SeFYVaIi8EtUZz3jHjRSxO0QnQsP9oB4TAE+oB4HS12I/SpoT5MrzHtGPrVKt+O3OsXXaWI2RSd42HdFI0p1BUqYZ3+0+Z4VsLPK2H5Y8rqq7b7M4gCQVw5c2Kcsb/rJwmfoTNuxf6ijHVU5uGLQ9X1QS3sPv8LzCAMeBbbr0LSpkuKyIi8nC31xLtedB97gIHpa8AD+Z3RgdoAeLyvaNNHPs4BjyJWSyR+0pdi/e1CNkzMtHBtmhKlLFgKKcWjgUvzt/wH+HLUVzpR2C4vGwsadGNq+tdwkPRfWrBsY8k9xX8RkpTT7dqafxgkzYYQNV2kGUjP30rh31+97u8tT8+0HgxE3DMlIX/4XwrPhyyvfmCxmb7vbmVzvHGBeYhPxPDk3kS/18n4gTm8dG1xmG6LkKY7kpntlTkBvvgEn5Mrq8WxvQwi2e9Eie0vbnF+2G+ld5sHN/eEd0vKOWWIA/g7CRsiNDTFaOORAqzoYf6wX1Y0oXw/C3h1bPXKcOMjHO2wg99BoXeAqdPjqwd41bPOB/Bw0ZjDlcNKt2NXlp40UX72/XwnnxYRt/n/lDTZO9HJRXy3c5ffFAMOK8aXRA1cN8spUzR2sIXb4NzimCPBQz7sfhMGV1fbSUoajtN+OVeCiWdO1h1R0lJ9DhHLyOM6m9uU8p9yD64BYW3Uwb2U5PjxIc+XsCRrNeXoidjSHPq7XSC61Uldgju7kO/B8ZWRF1NSnRsWqmaIHBGL0/4sL+VqrxlwR2qqHeZXOqQwuV7K9R0Vh5xDz8hGAvsT9HNGAoEhMlTUBHvJmVkmPzf9pLGBYbNMDkk+vqjjfWGtkQnsvHlPOY4AcJEyFpSvGMtVeS9ZXPaxCuD+CnRe0B/Nx8PZU/dWupbo8+zr8jnDjNpqnI5Pdkvi/zcy4GAZQHgB7BM0KBCJHoRnLrQbfFFzJ9BHzbrMcZU0nlzNvyRuwSNdzFYthr9HLlIZmrpMvMwNdnprCcjEiufm/wDXPpRUVZwcAAAAAElFTkSuQmCC\n",
```

R9/JuAT9Id77j4IbZ3W9teQ/fFcSvdl/nqh7rcnv8BvC3JnXQHS1dtR9230nU9XQ/8K/DhSebdj+6X9q2tzi1
0e0vbjPve0B0TuIjueMe36EJmsi7BkWnhdSJdd8/W1o7X030X7UZ3gHoT3To8i27bQneW2OVJ7qJ7715
bVdczvG5vAzbSbd/PAB+nCyJaN9uL6E6muJ7uhIy/oOvGAlgKXNle8710x5l+uMM2xk5m25kiknZiSVbQH
bj9XzPdltksyel0X/rPmum27Gzcs5C0y0pyaJJnJNktyc/Q7bl8cqbbtTPafXgWSdppPYzuFO4j6I47XEh3LY+
2k91QkqRBdkNJkgaNtBsqyQ10tyC4F7inqhYnOZDuFM9FdFd+vriqbm3zn0V3uuS9wGuq6tOt/Fi6K1T3
pjs//rU1sEt00EEH1aJFi3b4OknSruyKK664uarmjy2fjmMWv1hVN/fGzwQuqapzk5zZxt+Y5Ci60y6Pprvq
8jNJHttObzuP7jYIX6ILi6V0p8tNaNGiRaxdu3bHr40k7cKSjHsXgZnohjoRWNmGVwIn9covrKq72/nT64El
SQ4F9quqy9rexAW9OpKkaTDqsCjgX9ttg5e3skOqajNAez64lS/g/hfTbGxlC9rw2PIHSLK83VJ47datW3fg
akjS3DbqbqhnVNWmJAcDFyf55iTzjnefmpqk/IGFVefT3ZmSxYsXe5qXJO0gI92zqKpN7XkL3YUwS+julnk
odBfM0N37Hro9hv5NvhbSXfa/kfvfCGxbuSRpmowsLJI8YtsdRNuNwZ5P949dVtPdu572/Kk2vBo4Jcme
SY4AjgTWtK6qO9P9d60AL+vVkSRNg1F2Qx0CfLL7fmd34KNV9S9JvgysSvevMm8ETgaoqiuTrAKuoruL5
Bm9++mfzn2nzl7EwJlQkqQda5e9gnvx4sXlqbOStH2SXFFVi8eWewW3JGmQYSFJGuRdZydw7OsvmOk
maBa64p0vm+kmSDPCPQtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMN
CkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTI
sJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNGnlYJJmX5CtJ/rGNH5jk4iTXtuc
DevOelWR9kmuSnNArPzbJujbtfUky6nZLku4zHXsWrwWu7o2fCVxSVUcCl7RxkhwFnAIcDSwFPphkXqtz
HrAcOLI9lk5DuyVJzUjDIslC4IXAX/SKTwRWtuGVwEm98gur6u6quh5YDyxJciiwX1VdVlUFXNCrI0maBqP
es/hj4A3Aj3tlh1TVZoD2fHArXwBs6M23sZUtaMNjyx8gyfIka5Os3bp16w5ZAUnSCMMiyS8DW6rqiqlW
GaesJil/YGHV+VW1uKoWz58/f4ovK0kasvsIl/0M4FeSvADYC9gvyUeAm5IcWlWbWxfTljb/RuDwXv2Fw
KZWvnCccknSNBnZnkVVnVVVC6tqEd2B689W1W8Kk2vBo4JcmeSY6gO5C9pnVV3ZnkuH
YW1Mt6dSRJ02CUexYTORdYleQ04EbgZICqujLJKuAq4B7gjKq6t9U5HVgB7A1c1B6SpGkyLWFRVZcCl7b
hW4DnTDDfOcA545SvBY4ZXQslSZPxCm5J0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSY
MMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMM
C0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nS
IMNCkjRoZGGRZK8ka5J8LcmVSd7ayg9McnGSa9vzAb06ZyVZn+SaJCf0yo9Nsq5Ne1+SjKrdkqQHGuWe
xd3As6vqicCTgKVJgPOBC6pqiOBS9o4SY4CTgGOBpYCH0wyry3rPGA5cGR7LB1huyVJY4wsLKpzVxvdoz
0KOBFY2cpXAie14RBC6vq7qq6HlgPLElyKLBfVV1WVQVc0Ksj1mkWRekq8CW4CLq+py4JCq2gz
Qng9usy8ANvSqb2xlC9rw2PLxXm95krVJ1m7dunWHroskzWUjDYuqureqngQspNtLOGaS2cc7DlGTlI/3
eudX1eKqWjx//vztbq8kaXzTcjZUVd0GXEp3rOGm1rVEe97SZtsIHN6rthDY1MoXjlMuSZomozwban6S
vw3sBzgW8Cq4FlbbZlwKfa8GrglCR7JjmC7kD2mtZVdWeS49pZUC/r1ZEkTYPdR7jsjsQ4GV7Yym3YBVVf
WPSS4DViU5DbgROBmgqq5Msgq4CrgHOKOq7m3LOh1YAewNXNQekqRpMrKwqKqvAA08ep/wW4Dk
T1DkHOGec8rXAZMc7JEkj5BXckqRBhoUkaZBhIUkaNKWwSHLJVMokSbumSY9ZJNkLeBhwULsmYuwain2Aw0bcNknSTmLoAPcrgdfSBcMV3B8WdwAfGF2zJEk7k0nDoqreC7w3yaur6v3T1CZJ0k5mSqfOVtX7kzwdWNSvU1UXjKhdkqSdyJTCIsmHgccA64Cxq6rHbhcuSdrFTfWivMXAUe1GfpKkOWaq11l8DXjkKBsiSdp5TXXP4iDgqiRrgLvHCqvqV0bSKknSTmWqYXH2KBshSdq5TfVsqM+NuiGSpJ3XVM+GupPu7CeAhwJ7AN+vqv1G1TBJ0s5jqnsW+/bHk5wELBlFgyRJO58HddfZqvpr4Nk7timSpJ3VVLuhXtwb3Y3uuguvuZCkOWKqZ0O9qDd8D3ADcOIOb40kaac01WMWp466IZKknddUf/xoYZJPJtmS5KYkn0iycNSNkyTtHKZ6gPsvgdV0v2uxAPibViZJmgOmGhbzq+ovq+qe9lgBzB9huyRJO5GphsXNSX4zybz2+E3gllE2TJK085hqWPwO8OvAd4HNwEsAD3pL0hwx1VNn/w+wrKq+B5DkQOBddCEiSdrFTXXP4gljQQFQVbcCTx5NkyRJO5uphsVuSQ4YG2l7FlPdK5EkzXJT/cB/N/DvSf6K7jYfvw6cM7JWSZJ2KlO9gvuCJGvpbh4Y4MVVddVIWyZJ2mlMuSuphYMBIUlz0IO6RbkkaW4xLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSoJGFRZLDk/xzkquTXJnkNa38wCQXJ7m2PfdvI3JWkvVJrklyQq/82CRfbdPelySjarck6YFGuWdxD/D6qno8cBxwRpKjgDOBS6rqSOCSNk6bdgpwNLAU+GCSeW1Z5wHLgSPbY+kI2y1JGmNkYVFVm6vqi234TuBqup9kPRFY2WZbCZzUhk8ELqyqu6vqemA9sCTJocB+VXVZVRVwQa+OJGkaTMsxiySL6G5pfjlwSFVthi5QgIPbbAuADb1qG1vZgjY8tny811meZG2StVu3bt2h6yBJc9nIwyLJw4FPAK+tqjsmm3WcspqkfNvCqvOranFVLZ4/358Il6QdZaRhkWQPuqD4aFX9v1Z8U+taoj1vaeUbgcN71RcCm1r5wnHKJUnTZJRnQwX4EHB1Vb2nN2k1sKwNLwM+1Ss/JcmeSY6gO5C9pnVV3ZnkuLbMl/XqSJKmwSh/7e4ZwG8BX02yrpW9ETgXWJXkNOBG4GSAqroyySq626DfA5xRVfe2eqcDK4C9gYvaQ5I0TUYWFlX1BcY/3gDwnAnqnMM4v8BXVWuBY3Zc6yRJ28MruCVJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSN

```
MiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiw
kCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQ
NGllYJPmrJFuSfKNXdmCSi5Nc254P6E07K8n6JNckOaFXfmySdW3a+5JkVG2WJI1vlHsWK4ClY8rOBC6pq
iOBS9o4SY4CTgGObnU+mGReq3MesBw4sj3GLlOSNGIjC4uq+jfgu2OKTwRWtuGVwEm98gur6u6quh5
YDyxJciiwX1VdVlUFXNCrI0maJtN9zOKQqtoM0J4PbuULgA29+Ta2sgVteGz5uJIsT7I2ydqtW7fu0IZL0lw2
Ww5wj3ccoiYpH1dVnV9Vi6tq8fz583dY4yRprpvusLipdS3Rnre08o3A4b35FgKbWvnCccolSdNousNiNb
CsDS8DPtUrPyXJnkmOoDuQvaZ1Vd2Z5Lh2FtTLenUkSdNk91EtOMnHgOOBg5JsBN4CnAusSnIacCNw
MkBVXZlkFXAVcA9wRlXd2xZ1Ot2ZVXsDF7WHJGkajSwsquolE0x6zgTznwOcM075WuCYHdg0SdJ2mi0
HuCVJs5hhIUkaZFhIkgYZFpKkQYaFJGnQyM6GkjQ6N77t8TPdBM1Cj3rzupEt2z0LSdIgw0KSNMiwkCQN
MiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiw
kSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYN
2mrBIsjTJNUnWJzlzptsjSXPJThEWSeYBHwB+CTgKeEmSo2a2VZI0d+wUYQEsAdZX1XVV9f+AC4ETZ7hN
kjRn7D7TDZiiBcCG3vhG4GljZ0qyHFjeRu9Kcs00tG0uOAi4eaYbMRvkXctmugl6ID+f27wlO2Ipjx6vcGcJi/
G2QD2goOp84PzRN2duSbK2qhbPdDuk8fj5nB47SzfURuDw3vhCYNMMtUWS5pydJSy+DByZ5IgkDwN
OAVbPcJskac7YKbqhquqeJL8NfBqYB/xVVV05w82aS+za02zm53MapOoBXf+SJN3PztINJUmaQYaFJGm
QYTGHJVmU5Bsz3Q5Js59hIUkaZFhoXpI/T3Jlkn9NsneSVyX5cpKvJflEkocDJFmR5Lwkn0tyXZJnJfmrJFcn
WTHD66FdQJJHJPmn9tn/9tn7RpJfT3JDkj9MsqY9frrN+6Iklyf5SpLPJDmklZ+dZGX7PN+Q5FeT/FGSdUn+Jck
eM7uWOyfDQkcCH6iqo4HbgF8D/q6qnlpVTwSuBk7rzX8A8GzgGzgfwL/ALwHOBp4fJInTWO7tWtaCmyqqi
dW1THAv7TyO6pqCfAnwB+3si8Ax1bVk+nuF/eG3nIeA7yQyHwE+V1WPB37QyrWdDAtdX1VfbcNX
AIuAY5L8e5J1wEvpwmCbf6jufOt1wE1Vta6qfgxc2epKD8U64LltT+Lnq+r2Vv6V6x3vPT2/BC4NPtc/p67v85
vaiqftSWN4/7QmcdfK4fFMNCd/eG76W7UHMF8Nvtl9hbgb3Gmf/HY+r+mJ3kIk/NXlX1n8CxdF/q70jy5
m2T+rO15/cDf9I+p69mnM9p+yHzo7rvgjI/pw+SYaHx7Atsbn27L53pxmjuSHIY8P2q+gjwLuApbdKv954
va8M/AXy7DXs74BEzYTWe/w1cDnyL7hfevjPbHM0hjwfemeTHwI+A04OIz4GPA3smuZzuB+5L2rxnA3+b5N
vAl4Ajpr+5c4e3+5A0qyW5AVhcVf7PihlkN5QkaZB7FpKkQe5ZSJIGGRaSpEGGhSRpkGEhPURJrkxy3AzWv
8nujWr40KoaFJGmQYSGNwER3RG2emOSzSa5N8qpende3u/1+PclbZ6DZ0oQMC2k0Jrsj6hPo7nz6dOD
NSQ5L8ny6OwAvAZ4EHHBoLgb9JcijwMOD63rRPVdUPgB8k+RxdQDwTeD7wlTbPo+hC49+7EuEkUxm4HBeUAp01QZ2wk0Jrsj6hPo7nz6dOD
NSQ5L8ny6OwAvAZ4EHJvkF6a3ydLEvN2HNBoLgb9JcijwMOD63rRPVdUPgB8k+RxdQDwTeD7wlTbPP
nTh8W/T12RpYoaFNBrvB95dVauTHE93H6Ntxl4JW0CAd1TVn01L66TtZDeUNBqT3RH1xCR7JflJ4Hjgy8
CngVck2Qy89XY6Uh7llID93Dk2zsjb+bye+Iugb4J+BRwNurahOwKcnjgMuSANwFnApsGX3zpWH
eG0qSNMhuKEnSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA36/x/W84YoIJYvAAAAAElFTkSuQmCC\
n",
      "text/plain": [
       "<Figure size 432x288 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
   "# data distribution\n",
   "sns.countplot(df.v1)\n",
   "plt.xlabel('Label')\n",
   "plt.title('Number of ham and spam messages')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 17,
  "id": "93f91193",
  "metadata": {},
  "outputs": [],
  "source": [
   "# splitting data into input and output\n",
   "X = df.v2\n",
   "Y = df.v1\n",
   "le = LabelEncoder()\n",
   "Y = le.fit_transform(Y)\n",
   "Y = Y.reshape(-1,1)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 18,
  "id": "afe2d75d",
  "metadata": {},
  "outputs": [],
  "source": [
   "# test and train split\n",
   "X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)"
  ]
```

```
},
{
 "cell_type": "code",
 "execution_count": 19,
 "id": "4f7807e5",
 "metadata": {},
 "outputs": [],
 "source": [
  "max_words = 1000\n",
  "max_len = 150\n",
  "tok = Tokenizer(num_words=max_words)\n",
  "tok.fit_on_texts(X_train)\n",
  "sequences = tok.texts_to_sequences(X_train)\n",
  "\n",
  "sequences_matrix = sequence.pad_sequences(sequences,maxlen=max_len)\n"
 ]
},
{
 "cell_type": "code",
 "execution_count": 20,
 "id": "304aa1d6",
 "metadata": {},
 "outputs": [],
 "source": [
  "#layers of the model\n",
  "inputs = Input(name='inputs',shape=[max_len])\n",
  "layer = Embedding(max_words,50,input_length=max_len)(inputs)\n",
  "layer = LSTM(64)(layer)\n",
  "layer = Dense(256,name='FC1')(layer)\n",
  "layer = Activation('relu')(layer)\n",
  "layer = Dropout(0.5)(layer)\n",
    "layer = Dense(1,name='out_layer')(layer)\n",
    "layer = Activation('sigmoid')(layer)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "id": "44edc1b2",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Model: \"model\"\n",
      "_________________________________________________________________\n",
      " Layer (type)                Output Shape              Param #   \n",
      "=================================================================\n",
      " inputs (InputLayer)         [(None, 150)]             0         \n",
      "                                                                 \n",
      " embedding (Embedding)       (None, 150, 50)           50000     \n",
      "                                                                 \n",
      " lstm (LSTM)                 (None, 64)                29440     \n",
      "                                                                 \n",
      " FC1 (Dense)                 (None, 256)               16640     \n",
      "                                                                 \n",
      " activation (Activation)     (None, 256)               0         \n",
      "                                                                 \n",
      " dropout (Dropout)           (None, 256)               0         \n",
      "                                                                 \n",
      " out_layer (Dense)           (None, 1)                 257       \n",
     "                                                                 \n",
     " activation_1 (Activation)   (None, 1)                 0         \n",
     "                                                                 \n",
     "=================================================================\n",
     "Total params: 96,337\n",
     "Trainable params: 96,337\n",
     "Non-trainable params: 0\n",
     "_________________________________________________________________\n"
    ]
   }
  ],
  "source": [
   "model = Model(inputs=inputs,outputs=layer)\n",
   "model.summary()\n",
   "model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 22,
  "id": "57be216e",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Epoch 1/10\n",
     "30/30 [==============================] - 14s 282ms/step - loss: 0.3317 - accuracy: 0.8720 - val_loss: 0.1777 - val_accuracy: 0.9768\n",
     "Epoch 2/10\n",
     "30/30 [==============================] - 7s 226ms/step - loss: 0.0930 - accuracy: 0.9776 - val_loss: 0.0651 - val_accuracy: 0.9842\n"
    ]
   },
   {
    "data": {
     "text/plain": [
      "<keras.callbacks.History at 0x222d39d8d00>"
     ]
    },
    "execution_count": 22,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "model.fit(sequences_matrix,Y_train,batch_size=128,epochs=10,\n",
   "          validation_split=0.2,callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 30,
  "id": "ce5258f0",
  "metadata": {},
  "outputs": [
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     "WARNING:absl:Found untraced functions such as lstm_cell_layer_call_fn, lstm_cell_layer_call_and_return_conditional_losses while saving (showing 2 of 2). These functions will not be directly callable after loading.\n"
    ]
   },
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "INFO:tensorflow:Assets written to: my_model\\assets\n"
    ]
   },
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     "INFO:tensorflow:Assets written to: my_model\\assets\n"
    ]
   }
  ],
  "source": [
   "# saving a model\n",
   "model.save(\"my_model\")"
  ]
 },
 {
  "cell_type": "markdown",
  "id": "3e3b1525",
  "metadata": {},
  "source": [
   "# loading model\n",
   "# model = load_model('saved_model/my_model')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 31,
  "id": "cda2a0ca",
  "metadata": {},
  "outputs": [],
  "source": [
   "test_sequences = tok.texts_to_sequences(X_test)\n",
   "test_sequences_matrix  = sequence.pad_sequences(test_sequences,maxlen=max_len)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 32,
  "id": "38d64f40",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "27/27 [==============================] - 1s 34ms/step - loss: 0.0686 - accuracy: 0.9844\n"
    ]
   }
  ],
  "source": [
   "accr = model.evaluate(test_sequences_matrix,Y_test)"
  ]
 },
{
 "cell_type": "code",
 "execution_count": 33,
 "id": "26ad9585",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "Test set\n",
    "  Loss: 0.069\n",
    "  Accuracy: 0.984\n"
   ]
  }
 ],
 "source": [
  "print('Test set\\n  Loss: {:0.3f}\\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 34,
 "id": "cfe48015",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "27/27 [==============================] - 1s 33ms/step\n",
     "[[0.0052104 ]\n",
     " [0.00270706]\n",
     " [0.01115318]\n",
     " [0.00448523]\n",
     " [0.02838335]\n",
     " [0.00183202]\n",
     " [0.43712318]\n",
     " [0.00751833]\n",
     " [0.00228402]\n",
     " [0.78971624]]\n"
    ]
   }
  ],
  "source": [
   "y_pred = model.predict(test_sequences_matrix)\n",
   "print(y_pred[0:10])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 35,
  "id": "f0154ae1",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "[[0]\n",
     " [0]\n",
     " [0]\n",
      " [0]\n",
      " [0]\n",
      " [0]\n",
      " [0]\n",
      " [0]\n",
      " [0]\n",
      " [1]]\n"
     ]
    }
   ],
   "source": [
    "print(Y_test[0:10])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "a66043e8",
   "metadata": {},
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "ce1e195e",
   "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": "3.9.12"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```