```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "58a7dc8d",
   "metadata": {},
   "source": [
    "### LSTM ARCHITECTURE"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "f40f6596",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "62beca64",
   "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>CONTENT</th>\n",
"      <th>CLASS</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>Huh, anyway check out this you[tube] channel: ...</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>Hey guys check out my new channel and our firs...</td>\n",
"      <td>1</td>\n",
 "   </tr>\n",
 "   <tr>\n",
 "     <th>2</th>\n",
 "     <td>just for test I have to say murdev.com</td>\n",
 "     <td>1</td>\n",
 "   </tr>\n",
 "   <tr>\n",
 "     <th>3</th>\n",
 "     <td>me shaking my sexy ass on my channel enjoy ^_^ </td>\n",
 "     <td>1</td>\n",
 "   </tr>\n",
 "   <tr>\n",
 "     <th>4</th>\n",
 "     <td>watch?v=vtaRGgvGtWQ   Check this out .</td>\n",
 "     <td>1</td>\n",
 "   </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "                                          CONTENT  CLASS\n",
 "0  Huh, anyway check out this you[tube] channel: ...      1\n",
 "1  Hey guys check out my new channel and our firs...      1\n",
 "2             just for test I have to say murdev.com      1\n",
 "3   me shaking my sexy ass on my channel enjoy ^_^       1\n",
 "4            watch?v=vtaRGgvGtWQ   Check this out .      1"
]
},
"execution_count": 2,
"metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "Data = pd.read_csv('YOUTUBE.csv')\n",
  "Data.head()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 3,
 "id": "43b4bf60",
 "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>CONTENT</th>\n",
"      <th>CLASS</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>1951</th>\n",
"      <td>I love this song because we sing it at Camp al...</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1952</th>\n",
"      <td>I love this song for two reasons: 1.it is abou...</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1953</th>\n",
"      <td>wow</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1954</th>\n",
"      <td>Shakira u are so wiredo</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1955</th>\n",
    "      <td>Shakira is the best dancer</td>\n",
    "      <td>0</td>\n",
    "    </tr>\n",
    "  </tbody>\n",
    "</table>\n",
    "</div>"
   ],
   "text/plain": [
    "                                            CONTENT  CLASS\n",
    "1951  I love this song because we sing it at Camp al...      0\n",
    "1952  I love this song for two reasons: 1.it is abou...      0\n",
    "1953                                               wow      0\n",
    "1954                          Shakira u are so wiredo      0\n",
    "1955                       Shakira is the best dancer      0"
   ]
  },
  "execution_count": 3,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "Data.tail()"
]
},
{
"cell_type": "code",
"execution_count": 4,
"id": "59b21d72",
"metadata": {},
"outputs": [
  {
   "data": {
    "text/plain": [
     "CLASS\n",
     "1    1005\n",
     "0     951\n",
     "Name: count, dtype: int64"
    ]
   },
   "execution_count": 4,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "Data['CLASS'].value_counts()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 5,
 "id": "15eeec04",
 "metadata": {},
 "outputs": [],
 "source": [
  "Data['CONTENT'] = Data['CONTENT'].apply(lambda x: x.lower() if pd.notna(x) else \"\")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 6,
 "id": "7537b6b5",
 "metadata": {},
 "outputs": [],
 "source": [
  "from sklearn.preprocessing import LabelEncoder\n",
  "\n",
  "label_encoder = LabelEncoder()\n",
  "Data['CLASS'] = label_encoder.fit_transform(Data['CLASS'])"
 ]
},
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "92676667",
 "metadata": {},
 "outputs": [],
 "source": [
  "num_classes = len(label_encoder.classes_) "
 ]
},
{
 "cell_type": "code",
 "execution_count": 8,
 "id": "753f4b88",
 "metadata": {},
 "outputs": [],
 "source": [
  "x = Data['CONTENT']\n",
  "y = Data['CLASS']"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "6369e19b",
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.keras.utils import to_categorical\n",
  "\n",
  "y = to_categorical(y, num_classes=num_classes)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "id": "526d9637",
 "metadata": {},
 "outputs": [],
 "source": [
  "from sklearn.model_selection import train_test_split\n",
  "\n",
  "x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "id": "10b2f33c",
 "metadata": {},
 "outputs": [],
 "source": [
   "max_words = 10000  \n",
   "max_sequence_length = 100"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 12,
  "id": "b94b922f",
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.keras.preprocessing.text import Tokenizer\n",
   "\n",
   "tokenizer = Tokenizer(num_words=max_words)\n",
   "tokenizer.fit_on_texts(x_train)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 13,
  "id": "8f4e9c46",
  "metadata": {},
  "outputs": [],
  "source": [
   "X_train_sequences = tokenizer.texts_to_sequences(x_train)\n",
   "X_test_sequences = tokenizer.texts_to_sequences(x_test)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 14,
 "id": "6f831140",
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.keras.preprocessing.sequence import pad_sequences\n",
  "\n",
  "X_train_padded = pad_sequences(X_train_sequences, maxlen=max_sequence_length)\n",
  "X_test_padded = pad_sequences(X_test_sequences, maxlen=max_sequence_length)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 15,
 "id": "490ac782",
 "metadata": {},
 "outputs": [],
 "source": [
  "embedding_dim = 100  \n",
  "lstm_units = 128"
 ]
},
{
 "cell_type": "code",
 "execution_count": 16,
 "id": "e8491acc",
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.keras.models import Sequential\n",
  "from tensorflow.keras.layers import Embedding\n",
  "from tensorflow.keras.layers import Bidirectional\n",
    "from tensorflow.keras.layers import LSTM\n",
    "from tensorflow.keras.layers import Dense"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 17,
   "id": "52b2d9ee",
   "metadata": {},
   "outputs": [],
   "source": [
    "model = Sequential()\n",
    "model.add(Embedding(input_dim=max_words, output_dim=embedding_dim,
input_length=max_sequence_length))\n",
    "model.add(Bidirectional(LSTM(units=lstm_units, dropout=0.2, recurrent_dropout=0.2)))\n",
    "model.add(Dense(units=num_classes, activation='softmax'))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 18,
   "id": "db721ff6",
   "metadata": {},
   "outputs": [],
   "source": [
    "model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 19,
 "id": "34a45025",
 "metadata": {},
 "outputs": [],
 "source": [
  "epochs = 20\n",
  "batch_size = 32"
 ]
},
{
 "cell_type": "code",
 "execution_count": 20,
 "id": "9a459487",
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.keras.callbacks import ModelCheckpoint\n",
  "\n",
  "model_checkpoint = ModelCheckpoint('YOUTUBE.h5', \n",
  "                                   monitor='accuracy', \n",
  "                                   save_best_only=True, \n",
  "                                   verbose=1,\n",
  "                                   mode='max')"
 ]
},
{
 "cell_type": "code",
 "execution_count": 21,
 "id": "1a806964",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "Epoch 1/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.5730 - accuracy: 0.6880\n",
    "Epoch 1: accuracy improved from -inf to 0.68799, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 17s 281ms/step - loss: 0.5730 - accuracy: 0.6880 - val_loss: 0.4441 - val_accuracy: 0.7771\n",
    "Epoch 2/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.1811 - accuracy: 0.9417\n",
    "Epoch 2: accuracy improved from 0.68799 to 0.94172, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 11s 253ms/step - loss: 0.1811 - accuracy: 0.9417 - val_loss: 0.2495 - val_accuracy: 0.9363\n",
    "Epoch 3/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0767 - accuracy: 0.9773\n",
    "Epoch 3: accuracy improved from 0.94172 to 0.97726, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 10s 239ms/step - loss: 0.0767 - accuracy: 0.9773 - val_loss: 0.2378 - val_accuracy: 0.9490\n",
    "Epoch 4/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.4789 - accuracy: 0.8714\n",
    "Epoch 4: accuracy did not improve from 0.97726\n",
    "44/44 [==============================] - 11s 242ms/step - loss: 0.4789 - accuracy: 0.8714 - val_loss: 0.3012 - val_accuracy: 0.8662\n",
    "Epoch 5/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0437 - accuracy: 0.9893\n",
    "Epoch 5: accuracy improved from 0.97726 to 0.98934, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 12s 263ms/step - loss: 0.0437 - accuracy: 0.9893 - val_loss: 0.1807 - val_accuracy: 0.9554\n",
    "Epoch 6/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0227 - accuracy: 0.9936\n",
    "Epoch 6: accuracy improved from 0.98934 to 0.99360, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 12s 286ms/step - loss: 0.0227 - accuracy: 0.9936 - val_loss: 0.1952 - val_accuracy: 0.9490\n",
    "Epoch 7/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0106 - accuracy: 0.9972\n",
    "Epoch 7: accuracy improved from 0.99360 to 0.99716, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 11s 255ms/step - loss: 0.0106 - accuracy: 0.9972 -
val_loss: 0.2302 - val_accuracy: 0.9490\n",
    "Epoch 8/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0212 - accuracy: 0.9943\n",
    "Epoch 8: accuracy did not improve from 0.99716\n",
    "44/44 [==============================] - 11s 258ms/step - loss: 0.0212 - accuracy: 0.9943 -
val_loss: 0.1807 - val_accuracy: 0.9427\n",
    "Epoch 9/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0086 - accuracy: 0.9972\n",
    "Epoch 9: accuracy did not improve from 0.99716\n",
    "44/44 [==============================] - 11s 256ms/step - loss: 0.0086 - accuracy: 0.9972 -
val_loss: 0.2118 - val_accuracy: 0.9299\n",
    "Epoch 10/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0033 - accuracy: 1.0000\n",
    "Epoch 10: accuracy improved from 0.99716 to 1.00000, saving model to YOUTUBE.h5\n",
    "44/44 [==============================] - 11s 261ms/step - loss: 0.0033 - accuracy: 1.0000 -
val_loss: 0.2377 - val_accuracy: 0.9363\n",
    "Epoch 11/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0059 - accuracy: 0.9986\n",
    "Epoch 11: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 11s 259ms/step - loss: 0.0059 - accuracy: 0.9986 -
val_loss: 0.2198 - val_accuracy: 0.9299\n",
    "Epoch 12/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0035 - accuracy: 1.0000\n",
    "Epoch 12: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 11s 243ms/step - loss: 0.0035 - accuracy: 1.0000 -
val_loss: 0.2188 - val_accuracy: 0.9427\n",
    "Epoch 13/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0013 - accuracy: 1.0000\n",
    "Epoch 13: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 11s 254ms/step - loss: 0.0013 - accuracy: 1.0000 -
val_loss: 0.2358 - val_accuracy: 0.9427\n",
    "Epoch 14/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 0.0011 - accuracy: 1.0000\n",
    "Epoch 14: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 11s 242ms/step - loss: 0.0011 - accuracy: 1.0000 -
val_loss: 0.2289 - val_accuracy: 0.9490\n",
    "Epoch 15/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 7.5091e-04 - accuracy: 1.0000\n",
    "Epoch 15: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 10s 237ms/step - loss: 7.5091e-04 - accuracy:
1.0000 - val_loss: 0.2381 - val_accuracy: 0.9490\n",
    "Epoch 16/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 6.1475e-04 - accuracy: 1.0000\n",
    "Epoch 16: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 10s 235ms/step - loss: 6.1475e-04 - accuracy:
1.0000 - val_loss: 0.2418 - val_accuracy: 0.9554\n",
    "Epoch 17/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 4.4207e-04 - accuracy: 1.0000\n",
    "Epoch 17: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 10s 232ms/step - loss: 4.4207e-04 - accuracy:
1.0000 - val_loss: 0.2406 - val_accuracy: 0.9490\n",
    "Epoch 18/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 4.3629e-04 - accuracy: 1.0000\n",
    "Epoch 18: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 12s 263ms/step - loss: 4.3629e-04 - accuracy:
1.0000 - val_loss: 0.2434 - val_accuracy: 0.9554\n",
    "Epoch 19/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 3.5652e-04 - accuracy: 1.0000\n",
    "Epoch 19: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 12s 269ms/step - loss: 3.5652e-04 - accuracy:
1.0000 - val_loss: 0.2518 - val_accuracy: 0.9554\n",
    "Epoch 20/20\n",
    "44/44 [==============================] - ETA: 0s - loss: 4.3136e-04 - accuracy: 1.0000\n",
    "Epoch 20: accuracy did not improve from 1.00000\n",
    "44/44 [==============================] - 11s 243ms/step - loss: 4.3136e-04 - accuracy:
1.0000 - val_loss: 0.2841 - val_accuracy: 0.9490\n"
   ]
  },
  {
   "data": {
    "text/plain": [
     "<keras.callbacks.History at 0x21ce3521710>"
    ]
   },
   "execution_count": 21,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "model.fit(X_train_padded, y_train, epochs=epochs, batch_size=batch_size,
validation_split=0.1,callbacks=[model_checkpoint])"
 ]
},
{
 "cell_type": "code",
 "execution_count": 22,
 "id": "cb25308f",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "13/13 [==============================] - 1s 19ms/step\n"
   ]
  }
 ],
 "source": [
  "y_pred = model.predict(X_test_padded)\n",
  "y_pred_classes = np.argmax(y_pred, axis=1)\n",
  "y_true_classes = np.argmax(y_test, axis=1)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 23,
 "id": "a6f0f26b",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "THE ACCURACY SCORE OF LSTM ARCHITECTURE IS : 95.91836734693877\n"
   ]
  }
 ],
 "source": [
  "from sklearn.metrics import accuracy_score\n",
  "\n",
  "AC = accuracy_score(y_true_classes,y_pred_classes)\n",
  "\n",
  "print(\"THE ACCURACY SCORE OF LSTM ARCHITECTURE IS :\",AC*100)"
 ]
```

```
},
{
 "cell_type": "code",
 "execution_count": 24,
 "id": "f6ced910",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "THE HAMMING LOSS OF LSTM ARCHITECTURE IS : 4.081632653061225\n"
   ]
  }
 ],
 "source": [
  "from sklearn.metrics import hamming_loss\n",
  "\n",
  "HL = hamming_loss(y_true_classes,y_pred_classes)\n",
  "\n",
  "print(\"THE HAMMING LOSS OF LSTM ARCHITECTURE IS :\",HL*100)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 25,
 "id": "7f1c3938",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "THE PRECISION SCORE OF LSTM ARCHITECTURE:\n",
    "\n",
    "\n",
    " 93.98148148148148\n"
   ]
  }
 ],
 "source": [
  "from sklearn.metrics import precision_score\n",
  "\n",
  "PR = precision_score(y_pred_classes,y_true_classes)\n",
  "\n",
  "print('THE PRECISION SCORE OF LSTM ARCHITECTURE:\\n\\n\\n',PR*100)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 26,
 "id": "608b7c34",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "THE RECALL SCORE OF LSTM ARCHITECTURE:\n",
    "\n",
    "\n",
    " 98.54368932038835\n"
  ]
 }
],
"source": [
 "from sklearn.metrics import recall_score\n",
 "\n",
 "RE = recall_score(y_pred_classes,y_true_classes)\n",
 "\n",
 "print('THE RECALL SCORE OF LSTM ARCHITECTURE:\\n\\n\\n',RE*100)"
]
},
{
"cell_type": "code",
"execution_count": 27,
"id": "2f650580",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "THE F1 SCORE OF LSTM ARCHITECTURE:\n",
   "\n",
   "\n",
   " 96.2085308056872\n"
  ]
 }
],
"source": [
 "from sklearn.metrics import f1_score\n",
 "\n",
   "F1 = f1_score(y_pred_classes,y_true_classes)\n",
   "\n",
   "print('THE F1 SCORE OF LSTM ARCHITECTURE:\\n\\n\\n',F1*100)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 28,
  "id": "2c8d5164",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "THE CONFUSION MATRIX SCORE OF LSTM ARCHITECTURE:\n",
     "\n",
     "\n",
     " [[173   3]\n",
     " [ 13 203]]\n"
    ]
   }
  ],
  "source": [
   "from sklearn.metrics import confusion_matrix\n",
   "\n",
   "CM = confusion_matrix(y_true_classes,y_pred_classes)\n",
   "\n",
   "print('THE CONFUSION MATRIX SCORE OF LSTM ARCHITECTURE:\\n\\n\\n',CM)"
  ]
 },
{
 "cell_type": "code",
 "execution_count": 29,
 "id": "91a18df0",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAf0AAAHwCAYAAACsUrZWAAAAOXRFWHRTb2Z0d2FyZQBNYXRwb
G90bGliIHZlcnNpb24zLjcuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/bCgiHAAAACXBIWXMAAA9hA
AAPYQGoP6dpAABSxElEQVR4nO3deVxUVf8H8M+wzADCsMmqCO67SJpImmKiqKmZmoqaauKRpairpo
+YC+tSDZqZZptWTSwZa5q6pmbuF5kZmGb8wDFRwZ5dt5vz+8GFynAEGZmDA+3n3uq+ce8+993tnLvO
dc+6558qEEAJERET01LMwdwBERERUNZj0iYiUJJJn4iISCKY9ImIiCSCSZ+IiEgimPSJiIgPSJiIgkgkmfiIhIIpj0iYiIJIJ
n4iISCKY9ImIiCSiXEl/w4YNkMlksnGxgbe3t4IDQ3FqlWrkJWVpbNOVFQUzV7z7t69qzV/z5496Nq1K9z
d3WFnZ4cGDRpgyJAhOHDggkbMtWvWvXtPZnaWmJevXq4eWXX0Z8fLzW9mQyGaZMmWLwsXzyySeQy
WQIDAzUmj9x4kTI5XJcvnxZZ52ioiK0adHGfn5+yMnJKX7evzeK0adMGfn5+yMnJKXX7KpUK69evR3BwwMFxcXKBQKODn54cxY8bg3
LlzOuV/++03jBw5EnXxq1IFCoYC3tzdGjBiB3777Tads8edgeY2ODDGzdu6CwPDg5Gq1attOb5+flpvZePFs2awtbWFu7s7s7OnTogNmzzyM7O1tTbvTo0bC3t9fZphaCmzZt
QpcuXeDk5AQ7Ozu0bt0aixcv1vseBwcHQyaToV+/fjrLis+X9Ds9+9tdee03v8nnz5mnKlPR+DBky
BDKZDLNnz9aJwZDp2rVrOHbsmM457u7u7ujsGDB+PKlSs6+3zyfbx9+zZcXFzwwgsv6JQtLCxE69atDTpfAe
DevXuNWsYNWsWmjZtChsbG7i4uCA0NBR79+7VKHUaFsDkU8Wc6TXtvVuwDR+t4zRxl38HNqrOHpRfG8T
SqWzPyEEfH8IJPJ0Ldv3zLfr8d16NABMpkMa9asKfW4SYcbeXFTFpVy3fBdqZ5zkvyXZ5aioK0adMGfn5+yMnJKX7evzeKhYtWgR/f3/Y29vD1tYWrVq1w
uzZs3Hz5s1yf4bl/Y568jy1sLCAi4sLevfugbi4sLevfujbi4uBLfL0OPryayqshKixcvRv369VFYWIi0tDQcO3YM06dPxcff
IDdu3ejTZs2pa7//vvvY9asWvmzLfr8d16NABMpkMa9asKfW4SYcbeXFTFpVy3fBdqZ5zkvyXZ5aioK0adMGfn5+yMnJKX7evzeKhYtWgR/f3/Y29vD1tYWrVq1w
mD//v04ffo02rZtW5FDQExMDPz8/PDzz/PDzzz8jMTERjRo1AgAsWbIEnEu3btwsSJE3EnNyQYAK1aswK+//o
p9+/ahVq1aJW774cOcOHDwhwIA4cOIAuXbrg7bffhouL65l8g7ul0ku4LC65du4ZvG4ZvWvWrQsA2L59O8LCw
uDi4oJx48ahfv3+232LJaLyafhG7k44Z8k8FIrwB4rZtrjR/MG+8eybc+bIuYzXbSQoutLBIgcGvGGzgqwOF+NAPo
uDi4oJx48ahfv36uHbtGr744744gt8++232LJlC15++WWd/eTn52fLDDVr/
Sffv30f79u2RmZmJsWPHolmzZrh37x4uXbqQEPjQqMVT2G+nawjbZ+wDPPP4+oqCjY2dn
h5MmTWLRoEbZu3YoffvfvgBHh4eOuvu3bsX58+fR7t27SoUOwDY2Nhg27Zt+OSTT3Teg82bN8N8PGxkbzg+
hJmZmZZ2LNnD/z8/z8/LB582YsWbIEMpkbM5nqdLRpk1bZ5uX4/v161ixYoMfbXfc3N1y7dg0A8Oabb+LZZ
59FYWEhl26hLVr1+LYsWO04fPkyWO4fPkyPyPkyPD09SzwxCDDvd3d3DLF26FBMmTMDGjRsRRh6utd/Lly9/Jz549pZ6vAJCQkI
Du3bvjjzp07GDNmDNq3b4/09HTExMxMSgX79+mDlzJpYtYtWOMMLOI0Wbr8DYZvvrvNdz586Fvb095s
2bV+K2bWxsEBsi86dO2vNP37OK5fvw6FQlGuW8KXMSvIDMwc6ZHMxTr1ubLeuGWb8AGDhc2/Y2FPhFIrcE0ZvTRcHN9f5ihN5TTT3Tai3
2PDhg04deoULl++DBsbG025yMh7F68GI0bN8brr78Vt3D9999x0GDRqEmJgYDB8+vFxx6tOlSxe
d9/C1115Dhw4dMGHCBM28J/+ep16xZo/dv28nJSfPvv/76CyEhlUhOTsYrr7y7yCEhMmQC6X49KlS/
jiiy+w
Y8cOnDx5skKfYXk9kfYXk9nkP+7//+D5988gm6deuGs2fPonXr1rjlrDTm+GkuUw/r16wUAcfbsWZ1lhw8w8fFra2tsL
X11fk5uZq5kdGRgoA4s6dO0IIQoLC4VSqRQ9evTQu49bt25p/p2UlmmV2b2tAgJkyYoJkH
QEyePNmg4/jrr78EALF9+3bh5uYmoqKitJZ//vrvUatWLTFkyJAyAazDzMHNNPNfP+/vtvUatWLTFkyJAytz958mQU
JnWVFRkVi2bJlISUkRQgiRmJgo7OzsRLNmzcTt27e1yt65c65c0c0a9ZM1KpvS1y9elUzv/hzaNu2rVAoFGj
Rta63Xt2lW0bNlSa56vr6948cUXS437yc/qSS1bthRzge/qyEcfPDxe1aRkxaW9ZtrN0rmaVpovnp87o
tXSKvOf//xHABAzZ87UWX/37t3CwsJC9OrVS+fY6RbMGBGAJSS9Rrf+fY6tWrJ5ydnUW/fv2a0lpV0vugDQAwYMEBYWFinInTt3ai3
```

78ccfBQAxaNCgEt+PdevWCWtra3HkyBEBQBw7dqzEfb344ovC19dX77KjR48KAGLr1q1a89esWSMAiK
VLl2rN1/c+qtVq0blzZ1G7dm1x9+5dIcSjc9zW1lYMHDiwxLiKFRQUiFatWgk7Oztx+vRprWVFRUVi6NCh
AoDYsmWLZn553usnlfY9Uqw859bjnjxH9e134MCBonbt2qKwsFBr+fjx40W7du0M+lt53MKFC4W7u7v
Ytm2bkMlkIikpqcR9P3nMs2fPFgDE119/rZm3detWAUAMHjxYFBQU6GzrwIEDYs+ePUKIsj8HfX/T+s6hx
9WqVUuEh4cbvD19CgsLhb+/v7CzsxMnT57UWZ6RkSHefvttveuW9hmW9zuqpPdn//79AoCYNGlSubb
/NDDZNf0XXngBCxYswN9//42vvvqqxHJ3795FZmYmOnXqpHe5u7u7QfsCgKSkpArFGhMTA2dnZ7z44o
sYPHgwYmJitJYPGTIEffr0wZw5c3D79m0AwNSpU2FtbY0PP/yw1G1fv34dn376KXr06IHp06frLLe0tMT
MmTM1tfxly5YhNzcXn332mU4NqXbt2vj000+Rk5OD9957T2dbb7/9NlQqFZYsWVKewzeZq1evwtLSU
m9zrlKp1Kq5POnhw4dYtmwZmjRpgujoaJ3l/fr1Q3h4OA4cOIDTp09rLXNwcMCMGTOwZ88eXLhwocLx
16lTB126dEFsbKzW/JiYGLRu3VrnEsmTZXr06IFu3bqhefPmOueQsZ5//nkAj97jsshkMqxduxYZGRmYOX
MmAOCNN96AlZUVVq1aVeb627Ztw+XLlzFnzhydy12Wlpb49NNP4eTkhKioqPIfSAUZc26VJSwsDPfu3c
OhQ4c08woKCvDtt99WqPYcGxuLwYMHo2/fvnB0dNQ5n0qj73NesGABXFxcsG7dOlhbW+usExoaWu7L
D1Vt27Zt+OWXXzBv3jydFhXg0Wf47rvvmiGyR8rz9/W0MWlHvldffRUA8P3335dYxt3dHba2ttizZw/u37
9fof0Uf1Curq4VWj8mJgYDBw6EXC5HWFiYpnnucZ988gkKCgowY8YM7Nq1C7t378aSJUtKbWoFgP379
6OoqEjzXpSluIm4+CR8UpcuXeDn54d9+/bpLKtfvz5GjRqFzz//HDdv3ixzX4WFhbh7967WlJuba1Cc+vj6+
kKlUuk0zxni1KlTePDgAYYYPHw4rK/1XmUaNGgUAeq8pT5s2Dc7OzkYnouHDh2PPnj2a/gdFRUXYunVrq
V/+N2/exNGjRxEWFgbgURL59ttvvUBQYFQsjytu9nd2djaofMuWUWLTFz5kxs2LABb775Jg4cOIB33nkHder
UKXPdPXv2APjn/2AMjXnpo6MjzXnrpJfzxxx9ITEzUWpabm6tzThUWFhoUc2mMObfKEq/+/cjIyzb//+/
cjIyMCwYcPKta0zZ84gMTERYWFhkMvlGGDhwYLl+AD75Of/555/4448/6f6W5/4448/MGDAADg4OBi8HX2fg7F/36W5
f/++zr7S09M1y3fv3g0ABn8PVrWy/r7KOr6azKRJv27dunB0dCz115OFhQVmzZqF8+fPo169euj/+/+85
//lFpjKz6hb926hePHj2P06NEAgFdeeaXcMZ4/fx5//PGH5o+7c+fOqFu3rs4sfqqv+vLxYtWoTY2FiMHTsWn
Tp10rrGVZLizlf6rhM9KSMjAzdv3oS/v3+p5dq0aYPr16/r7Sg5b948FBUYenSpWXu7/vvv4ebm5vWpK8
FwVBjx46Fm5sbRo8ejebNm2PSpEnYvHkzMjIyylz3999/B4BSj9LEtOnTze6tj948GCoVCrs3L
kTwKP36O7du5qErs/mzZuhUCjw0ksvBHCAgQBw8e4LvvvqtwHFlZWbh79y5SU1Nx8OBBBTJ8+HTKZ
DIMGDTJ4GwsWLECDBg3w0UcfoV27dpg8ebJB6/3+++iXWVK+iwiIyN1zqkff/zR4JhLYsy5ZYj
hw4dj586dePjwIYBHFYGuXbvC29Nv56quuv4OPjo2m5HDZsGH7//XedjsbFMjIycPfuXVy/fh3btm3Do
kWLoFAoNDX38nx/PE7f5+Dm5qa3H4YpNG3aVO4znk1KlTGDNmDIBHf/
/6lHV8NVmFOvKVxt7eXm9yfUxt8HHXJyiRYvQrFkzfPLJzh48CD279+PefmPmISAgADExMWjevLlW+cjSGRmpe
K5VKLF26FAMHDix3fDEXMfDw8EC3bt0APGoKTp0KL766issK0xX74clpaWmrLTp0/Hl19+icuXL+PTTz/V6t
RXkszMTAAw6Fd68ftUVnipZmZmYZ5syZAy8vrxK3ExgYiHfeeUdn/n/Yry8PDAL7/8gs
WLF2PHjh1Yu3Yt1q5dC7lcjvnz52P+/+/PklvmeGHPvjx63PtGnTsHLlSixatAi7du2q0DE4OzujoV69e2Lx5M0a
OHInY2Fg899xzpSbAmJgYnYiEmJgYDBgyoUBxjx47Vel3cIfDZZ581eBtyuRwWFhaIzVZAy8vrxK3ExgYiHfeeUdn/n/Yry8PDAL7/8gs
2udy6XJysoq1zn4uAkTJuj8+C7rR6whjDm3yhK1xY2hBnRLrTw+ZIISDzTOCDpD+2I6V/ASq7ggmK/fuRa9evb8Pz/Pux2s9ugV7MJU6eA3+WhwNpOE91PN1rLNK3enaQVyjB1DcF7AF9QPPjeQXcnSBV8N2S2E+HSNAjawBetgcDm3zZB2WfPPWA/uaPWqDr0RhjnLADEBNAakRzeJD7uyRGwNrR64LYQ3sPvEAbh9b0FCAatc4dlegEJMMe8SbUqxeLvgSCTCMUDCCSJHqBwxVzZTGz2l+6M+hcsl+xs8+h5cNg7Y2tEJKuUmxTb+42OGGAvqNyJEWAHDr48JSDO0JhX2Uyd+0zMG5Xaw1+2cA3mlyZ37kx05rKDCqgBLHQ2nrSCFHqKJDy7yAK0dOhnqrlDGJPEDBiImDkCJJHDOdkhP2Vr1lkKqzOuCBE6zMC3ZM3HQ8LT+4YY06zx+zF8/H3oCvH38LvUmavK+s8CySjmj8x5HV3LQ2Gz5W1MFZxz0GN4yfBBEfU0tBBg5AICGhIsPUzwEe5fDoA==

gw8+wIwZMzBnzpwyx4p4Gpk06Rc3I4WGhlZo/fbt2wOAUYmrNDExMXB3d8fWrVt1prCwMOzYsaPEk
8lQvXv3hqWlZal3MDyub9++SEpKKvEX+cmTJ3Ht2rUye+vOnz/f4Gv7JSlu0k5ISNBZlpubi5SUlFKbvYs1a
NAAzs7OpX6OnTt3hpOTE2JjY0v8xf3ll18CQKnHXlzb37VrFy5evFhmbPrY2tpiwIABOHbsGHr06FHiL3wh
BGJjY9GtWze951CbNm1M1ot/yZIlyMvLM7iHc/HAVKtWrYKXlxfeffddfP/999iyZUuZ6xa/v8Xv95MyMz
Oxa9cuNGvWrFw/6CuDIeeWoV5++WVYWFjg9OnT5e61n5OTg127dmHo0KF6zwUvL68yzwVLS0tER0fj
5s2b+PjjjwEATZo0QdOmTbFr1y6twa1qmuLBswz9HjSUqb6Hs2bBwcHB8yfP9+k8dUEJkv6R44cwb///
W/Ur18fI0aMKLFcbm6u3pGQgEc9aIFHnShM7eHDh9i+fTv69u2LwYMH60xTpkxBVlaWptdpRfn4+GD8
+PH4/vvv9Q6ao1arNYO1AMCsWbNga2uL119/Hfu3dnaYNWtWqftt2LAhRo4ciU8/
/bTCo0Z1794dcrkca9asgVqt1lr22WefaDlB5C1799bMO3PmjN6R3n7++Wfcu3ev1M/Rzs4OM2fOREJCg
t5BOPbt24cNGzYgNDS0zA4006dPh5OTExYvXlzWIZZo5syIyMxIIFC0os8+OPP+LatWsYM2aM3nNo6N
ChOHr0qEF3UpSlYcOGDRoEDZs2FDm57ljxw7s3r0bixcv1nSceuONN9CuXTtTtERESU2Cei2ODBg9GiRQs
sWbJEp+ajVqsxadIkPHjwQOuaaGUz5twwylL29PdasWYOoqCi9IzyWZseOHcjJycHkyZP1ngt9+/bFtm3bkJ
+fX+p2goOD0aFDB6xcuVIzENShSiRYtw7949vPbaaygqKtJZ5/vvv9d7R0R0t1MnjwYLRu3Rrvvvvvuu3u/7rKysCg
2+U97vqJI4OTnh9ddfx8GDB0vsdPm0qtA1/f379+OPP/5AUVERbt26hSNHjuDQoUPw9fXF7t27S72HNjc
3F8899xw6duyIXr16wcfHB+np6di5cydOnjjyJAQMGICAgoEIHc+7cOZ2OasCjP6wbN24gKysL/fv317tux4
4d4ebmhpiYGAwdOrRC+y+y+2fPlyXL16FW+++abmh4azszOSk5OxdetWrbsHGjduI0bN2LEiBFo3bq1zoh8
d+/exebNm9GwYcMy9ztv3jzsWLLlyI+fPno0uXLujfvz/s7Ozw008/YfPmzejZs6fWl
+OmTZsQExODl19+Ge3atYNcLseVK1ewbt062NjY4O233y51f3PmzMHFixexdOlSxMXFYdCgQbC1tcWp
U6fw1VdfoXnz5ti4cWOZcTs6OmLatGlGlNfH7+/uX2QEtJiYGlpaWePHFF/Uu79+/P+bNm4ctW7YgliKiwr
EUmzVrFr755husXLmyxLEYsrKy8OabbyIgIABvvvmmZr6FhQXXrl2LwMBAzJs3r9RRG+VyOb799lt0794
dnTt31hqRLzY2FhcuXMBbb71V71V7tvZyrJu3TqtYbeLTZs2zehzy1CPj2BYHjExMXB1dcVzzzzz2nd3n//v3x+eef
Y9++fWV2OJ41axZeeeUVbNiwARMnTsTQoUPx66+/4t1338XFixcRFhamGZHvwIEDOHz4cLnGAqgM33
77rd4R63r06AEPDw9YW1tj+/btCAkQZcuXTBkyBB06tQJ1tbW+O233xAbGwBGwtnZ+dy36tf3u+o0hR3BF6
yZllOi1hZx1ejlWckn+JRpYonuVwuPD09RY8ePcSHH34oMjMzddbRNyLf559/Lgf559/LgYMGCB8fX2FQqEQdnZ2
IiAgQCxbtkzk5+dr1i3vCGSlTf/+979979Fv379hI2NjcjJySlxG6NHjxbW1taaUc2EKHFRWJ//73v+L5558
Xjo6OwtraWvj6+ooxx+6pS/dOmSCAsLE15eXsLa2lp4enqKsLAw8euvv+qULW1Es/DwcAGGgQiPyFf
vqq69Ex44dRa1atYRCoRDNmjUTixYtEnl5eToxz5o1SzzzzDPCxcVFWFlZCS8vL/HKK6+ICxcu6MSl731Uq
VRi/fr1olOnTkKpVAobGxvRsmVLsWjRIpGdna1TXt9og0dBA+Ho6Fiu86WsERwfP3cLCgqEq6ureP75
50tdp379+iIgIEBrXkVG5CsWHBwslEqlSE9PF0Lovo/Tpk0TFhYW4ueff9/pQpU4SFhYW4ueff9a7/pQpU4SFhYU4d+5cqXELIcTt
27dFRESEaNSokVAoFMLJyUmEhISI3bt365Q1xYh8JU0pKSnlOrceZ8ilfKWNBChE2X8rt27dElZWVuLVV1
8tsUxubq6ws7MTL7/8cpn7VqlUomHDhqJhw4aiqKhIM//w4cPipBmJfv36iV27dmn
KmGtEvpKmo0ePapV/8OCBWLhwoWjdurWws7MTNjY2olWrVmLu3LkiNTVV7z5K+wyLGfodVdb7M3r
0aGFpaSkSExMrdHw1kUwIISrl1wQRERFVK3y0LhERkUQw6RMREUEUkEkz4REZFEMOkTERFJBJM+ERGRR
DDpExERSQSTPhERkUQw6RMREUkEkz4REZFEMOkTERFJBJM+ERGRRDDpExERSQSTPhERkUQw6RMR
EUkEkz4REZFEMOkTERFJBJM+ERGRRDDpExERSQSTPhERkUQw6RMREUkEkz4REZFEMOkTERFJBJM+E
RGRRDDpExERSQSTPhERkUQw6RMREUkEkz4REZFEMOkTERFJBJM+ERGRRDDpExERSQSTPhERkUQw6
RMREUkEkz4REZFEMOkTERFJBJM+ERGRRDDpExERSQSTPhERkUQw6RMREUkEkz4REZFEMOkTERFJBJ
M+ERGRRDDpExERSQSTPhERkUQw6RMREUkEkz4REZFEMOkTERFJBJM+ERGRRFiZOwBTUavVuHnzJh
wcHCCTycwdDhERlYMQAllZWfD09ISFBX/rG4pJn4iISCJ4eZ+IiEgi/j/tCmSNFwoLbZMxfx8aR5eIZTlO
SkpKBu3bqVsu28vDzU97VH2m2VSbbn6emJpKSkGpX4n5qk7+DgAABYeqw9bOyfmsMi0rKjjvYe5QyCqF
EUoxCl8p/kurwwFBQVIu63C3+f9oHQwrjUhM0sN33bXUFBQwKRvDsDsVN+jb2VrBl0qenlJXM2twhEFUO
8eh/VXF51t5BBnsH4/ajhuHrR0dHY/v27fjjjjjz9ga2uL5557DkuXLkXkTpk01ZfLy8vDWy0Ly8vDWW29hy5YtyM/PR2
oKD755BN4ePzzQz85ORmTJk3C0aNHYW9vj/DwcERHR8PKyvAEVrHBP6YmxzcC0aNHYW9vj/DwcERHR8PKyvCcx458REQkKSqhNslkqOPHj2Py4Mk4ffo0Dh06hMLRyMnD0lmhdtnlCf
0Dh06hMLCQvTs2RM5OTmaMjNmzMCePXuwdetWHD9+HDdv3sTAgQP/iVmlwosvvoiCggL89NNP2Lh
xIzZs2ICFCxeW69hlQghRrjWqqczMTDg6OuLDcx1Z06en1jfNPc0dAlGlKBKFOIZdyMjIgFKprJR9FOeJtIR6
Jmne92yaXKF479y5A3d3dxw/fhxdunRBRkYG3NzcEBsbi8GDBwMA/vjjDzRv3hxxcXHo2LEj9u/fj759++L
mzZua2v/atWsxe/Zs3LlzB3K53KB9s6ZPRERUQZmZmVpTfn5+metkZGQAAFxcXAAAZ58+fR2FhIUJCQjRl
mjVrhnr16iEuLg4AEBcXh9atW2s194eGhiIzMxO//fabwEy6RMRkaSoTfQfAPj4+MDMRoVEzRUdHl75vtRr

Tp09Hp06d0KpVKwBAWloa5HI5nJyctMp6eHggLS1NU+bxhF+8vHiZodgOTkREkqISAiojr2wXr5+SkqLVv
K9QKEpdb/Lkybh8+TJOnTpl1P4rijV9IiKiClIqlVpTaUl/ypQp2Lt3L44ePao1FoGnpycKCgqQnp6uVf7WrVv
w9PTUlLl165bO8uJlhmLSJyIiSVFDmGQylBACU6ZMwY4dO3DkyBHUr19fa3m7du1gbW2Nw4cPa+YlJC
QgOTkZQUFBAICgoCD8+uuvuH37tqbMoUOHoFQq0aJFC4NjYfM+ERFJihoCqnIk7ZK2YajJkycjNjYWu3b
tgoODg+YavKOjI2xtbeHo6Ihx48YhIiICLi4uUCqVmDp1KoKCgtCxY0cAQM+ePdGiRQu8+uqreO+995CWl
ob58+dj8uTJZV5SeByTPhERUSVas2YNACA4OFhr/vr16Fhr/vr16zF69GgAwIoVK2BhYYYFBgwZpDc5TzNLSEnv37s
WkSZMQFBSEWrVqITw8HIsXLy5XLEz6REQkKeVtni9pG4pPpGakz
z4REUmKKXvv1zTsyEdERCQRrOkTEZGkqP83GbuNmohJn4iJEVlgt77xq5vLkz6REQkKSrxaDJ2GzURr+kT
ERFJBGv6REQkKbymT0REJBFqyKCCzOht1ERs3iciIpII1vSJiEhS1OLRZOw2aiImfSIikhSVCZr3jV3fXNi8T0R
EJBGs6RMRkaRIuabPpE9ERJKiFjKohZG9941c31zYvE9ERCQRrOkTEZGksHmfiIhIIlSwgMrIhm6ViWKpakz
6REQkKcIE1/QFr+kTERFRdcaaPhERSQqv6RMREUmESlhAJYy8pl9Dh+Fl8z4REZFEsKZPRESSooYMaiPrv
GrUzKo+kz4REUmKlK/ps3mfiIhIlljTJyIiSTFNRz427xMREVV7j67pG/nAHTbvExERUXXGmj4REUmK2gRj7
7P3PhERUQ3Aa/pEREQSoYaFZO/T5zV9IiIiiWBNn4iIJEUlZFAZ+WhcY9c3FyZ9IiKSFJUJOvKp2LxPRERE1R
mTPhERSYpaWJhkKo8TJ06gX79+8Pb2hkwmw86dO7WWy2QyvdOyZcs0Zfz8/HSWL1mypFxxsHmfiIgk
xRzN+zk5OfD398fYsWMxcOBAneWpqalar/fv349x48ZhO0KBBWvMXL16M8ePHa147ODiUKw4mfSIiokr
Wu3dv9O7du8Tlnp6eWq937dqFbt26oUGDBlrzHRwcdMqWB5v3iYhIUtT4pwd/RSf1/7aVmZmpNeXn5
xsd361bt7Bv3z6MGzdOZ9mSUvg6uqKgIAALFu2DEVFReXaNmv6REQkKaYZnOfR+j4+PlrzIyMjERUVZd
S2N27cCAcHB53LAG+++SaeeeYZuLi44a44KeffsLcuXORmpqKDz74wOBtM+kTERFVUEpKCpRKpea1jEwe
pvr1q3DiBEjYGNjozU/UIIC8+82bdpALpLfj9ddfR3R0tMH7ZdInIiJJMc3Y+4/Y+4/WVyqVWknfWCdPnkRCQgK+/
vrrMssGBGaiqKgI165dQ9OmTQ3aPpM+ERFJihoyo1vnFdO5FhZlrR0dxHnB8bvKgQ6PIWmNF/T1F0F3PJLIcX+7Ev
WkXnBxSGTmRwRQK85aYhYX9ddoBFcpIrTo/Y72+pGo7ojpS7K4kZvdHmQxiZ3GeX7rjkrc9HOEdQhl1JFHrmtqu7a
wFhdM1VPJcz5Xg5Y/bZgDpdqLk7J6Mno2nRE9M5YpI9ThpJRpQEDdNEt1HD5zx1prqNu+IHu2DuI8MQHGnbyPw
e4o/jwpsDyDCUN29MzbK2jFeDDaDVCijI2/LVz5prpDz9w8Zn3NRgTk0JSlPCZ7UmNjL1jjDqPtmzsDg/XR0Uw
mZWD7VwxDW7IUG7Z3EJy1bZEHW9I3+8TMHewnLrUbUS09wc0l2jp21bz5kw7t49H7drxTE0fXZJT9i1hxSNWMw
Ce4g4HHpRmh6Gfvb0TtUyX2hJuXHdf+RXsS3/3hNGoaR3jR7Jc7C8ZKr1pQaTavmtqfRf5vcZXmV2FmD7+gcEd
6dL5lW5mk8oB0wXPWhsUPDoxvMpcz0IYnSymLuCS2NuFcbjnGSY7gtUOzvbZvLT3kCVnqCGBxQ0kJfT8gS5OY
6M2P7SaZHr/ESJ5LunLxCBYsGlm8pI0jUaPP4UzPd5DOBp9nRjPCNRpD/Mm5J2zNxxHu/u3jGRIpWEc/u8OvS
QZXAlYlSu/BLMzIy+vLCpu6kHObBWPl+0ZTvu/XKr4z5rlffOF6+kSlz42OcnDK9CJGZYjUvJJ0MrFZuZJFGFlK
Z9DTmKKvMeU4kOEqCF5ZOdvl5xfmsWj08p08hbJaKnwjaMi8D7W2Y3w4pf+z+mLwnUx2ycxWLr9zEbUDGQL
RyW7gjlfFnP0e4bLaYJtNtpn+A9eNIlJP1DvC9sRvvPYPq1TY0uzR9pdyRY3EvX+CvzJBv++1OlGmk8H6HPt1vV
WcQsEPpV1RUSnZ0iQPu4D5TcjJH1ZpR6WOsumyB8pDX3TT9bvMmG7Pj1Yk7xX6bJMO+3OFk1eKFz5IzvXkaJcG
KU7zYgyZ3JG2QgNdzKohCfJIJpXtCBb1xPyUzOOjnCkv5D5vtsmfcvq5NdHhb0BoK5nsXKKMHs3TvbQwxCkBOG
mL+rwXiL1cMjj+cb5Fv3jx28Rz51RIHl/q+xmdbBHb12Fz7e4AcxdrGH0mpmdvTk9Gaz8xZbnmiugDrvW+qWo
WHw9nddwFBBRPORaRGBqNYuUd3BAJxZgPBsO+ZWkl5tsDr2XpoU3Sa0BBQj29smHy2lGm4O7pTR+D9OCo+j
cHhqUjttfm7PU5wnjqTbwT1cB7lMSAJlnZrKFxV1DmBFK+x8i0Jox/9+5/JZfrOrLBBKBHm5F2SlRZUU0cRnTS
LNxQ/PezW0rEeV9/ZvOBwUIXYZ4SMeRzxsy3+A8rBBbG3c+sHrDrA6dDTfHP+xKzFi5uMEH9PTAuJiKX2NGU
eWRdTZyUyX8VhgqEJYl5YfOA2EdnU4gAjW4B4QpCU5zvhp1RN9djKcN9CNTDPG1ZnJJJ0jhpbRoJtaAhbGd4
bOp7lg1uSy6SLHkPx1N9xTS8Y/JUS3xNp7VaNcArpqQjQmY+jU4t3VxUJj+yQFrPdUN/1BTWTWQ5u4ZS8+2m
bVSYaeS2cIHb1hS7cG2bD8T4BqHuKU8E6NnTDOJy2ffrw4wvlLYR7bTbpwGE1xWj01NVkOJ3w1AOTFaPHkHp
Y7pnLKoVGBqa9tVmzUhu7O4Jj4K2Ne8lW2Ry5uZGtWBZR15y8HhVdXUZxHRyd4XzWxjjD8pQ58EoJAqcoRKc
r8H/V7CmEj21+NRjh48v1LMjXrLzGVw/UuJ9LB8Hy2Kl+6k6WqWxBIh2SzFCh9e0Gl7q2BhsBz7lYd1Gbj4c5N
6PvGI4XcM8HgDT4RkBXl4DbXZR6HpCpT2sSwmuHTbqVb8UQoJVK8eWlUJ2tRtkr2O0oMTkjNgOD4rvl1mHNn
qTj0J9BeY4MsOn5oG0xS9cPvUD9VQnXwjCcZt3d4+0s3v6cfGFrx8t7Hi2xm8qxZ5HoXmn1h5KJz9vgW0v4Uuw
LjGYBHC2Nz0B6JJbv1K9sQe63DLo5y1FdfL2TiWnFU4bhwd1Vx5cM7VwZP/c5D/b+S/JxcBuO5JJA1/bDAk9wB
uTyQ2R5aSO4nU5hsSBFwpuf6p3sZMEo36eHLtmQzEMbjHkHE3AHuRS0v2OGBE13WgP7B4mm6vsDE8z5Y3KN5
Bz+WSBsk8PN5KX3hV68dk6YIm72LBKnJlJCqhIoqbOo6b4m5Xl7rLqz4aDxFTpKwrvpoX+W2P7jMHMXHoYvXnmz
Tp09Hp06d0KpVKwBAWloa5HI5nJyctMp6eHggLS1NU+bxhF+8vHiZodgOTkREkqISAiojr2wXr5+SkqLVv
K9QKEpdb/Lkybh8+TJOnTpl1P4rijV9IiKiClIqlVpTaUl/ypQp2Lt3L44ePao1FoGnpycKCgqQnp6uVf7WrVv
w9PTUlLl165bO8uJlhmLSJyIiSVFDmGQylBACU6ZMwY4dO3DkyBHUr19fa3m7du1gbW2Nw4cPa+YlJC
QgOTkZQUFBAICgoCD8+uuvuH37tqbMoUOHoFQq0aJFC4NjYfM+ERFJihoCqnIk7ZK2YajJkycjNjYWu3b
tgoODg+YavKOjI2xtbeHo6Ihx48YhIiICLi4uUCqVmDp1KoKCgtCxY0cAQM+ePdGiRQu8+uqreO+995CWl
ob58+dj8uTJZV5SeByTPhERUSVas2YNACA4OFhr/vr16zF69GgAwIoVK2BhYYFBgwZpDc5TzNLSEnv37s
WkSZMQFBSEWrVqITw8HIsXLy5XLEz6REQkKeVtni9pG4YyZDgcGxsbrF69GqtXry6+vL7777jD96sPk
z4REUmKKXvv1zTsyEdERCQRrOkTEZGkqP83GbuNmohJn4iJEVlgt77xq5vLkz6REQkKSrxaDJ2GzURr+kT
ERFJBGv6REQkKbymT0REJBFqyKCCzOht1ERs3iciIpII1vSJiEhS1OLRZOw2aiImfSIikhSVCZr3jV3fXNi8T0R
EJBGs6RMRkaRIuabPpE9ERJKiFjKohZG9941c31zYvE9ERCQRrOkTEZGksHmfiIhIIlSwgMrIhm6ViWKpakz
6REQkKcIE1/QFr+kTERFRdcaaPhERSQqv6RMREUmESlhAJYy8pl9Dh+Fl8z4REZFEsKZPRESSooYMaiPrv
GrUzKo+kz4REUmKlK/ps3mfiIhIlljTJyIiSTFNRz427xMREVV7j67pG/nAHTbvExERUXXGmj4REUmK2gRj7
7P3PhERUQ3Aa/pEREQSoYaFZO/T5zV9IiIiiWBNn4iIJEUlZFAZ+WhcY9c3FyZ9IiKSFJUJOvKp2LxPRERE1R
mTPhERSYpaWJhkKo8TJ06gX79+8Pb2hkwmw86dO7WWy2QyvdOyZcs0Zfz8/HSWL1mypFxxsHmfiIgk
xRzN+zk5OfD398fYsWMxcOBAneWpqalar/fv349x48Zh0KBBWvMXL16M8ePHa147ODiUKw4mfSIiokr
Wu3dv9O7du8Tlnp6eWq937dqFbt26oUGDBlrzHRwcdMqWB5v3iYhIUtT4pwd/RSf1/7aVmZmpNeXn5
xsd361bt7Bv3z6MGzdOZ9mSJUvg6uqKgIAALFu2DEVFReXaNmv6REQkKaYZnOfR+j4+PlrzIyMjERUVZd
S2N27cCAcHB53LAG+++SaeeeYZuLi44a44KeffsLcuXORmpqKDz74wOBtM+kTERFVUEpKCpRKpea1jEwe
pvr1q3DiBEjYGNjozU/IiJC8+82bdpALpLfj9ddfR3R0tMH7ZdInIiJJMc3Y+4/WVyqVWknfWCdPnkRCQgK+/
vrrMssGBGaiqKgI165dQ9OmTQ3aPpM+ERFJihoyqGHciHrGrl+SL774Au3atYO9vT3c3d3Rv39/JCQkaJ
Dt8+kT0REkqImoYKmr6hsrOzkZiYqLWvKKiIkRGRiIgIAB2dnbw9vbGyJEjcfPmTa1t+Pn5QSaTaU1Llixp1
FxcXGYGMWDGRo4cCWdnZ4PjiYuLw4ABA+Dm5galUolBgwbpxQJGfCSmqkgDxf8Rp/dyivhR+O/zAXXNJSXl6

Xh7exsUR7Vl+gAQERGB8PBwtG/fHh06dMDKlSuRk5Oj6c1PVcO9QwGGXNHfj6JYwyEP0XCI/mv5Tk2K
ELyBPfWp5lrxFvsHPe1M2Xv/yU7kxrRCHzt2DO7u7nB2dsYLL7yAd955B66urgCAuLg4ODk5aRI+AISEhM
DCwgJnzpzByy+/bNA+qk3SHzp0KO7cuYOFCxciLS0Nbdu2xYEDB3Q69xERERnDlM37T3Yij4yMRFRUVL
m316tXLwwcOBD169fH1atX8fbbb6N3796Ii4uDpaUl0tLS4O7urrWOlZUVXFxcSuzwrk+1SfoAMGXKFDb
nExFRjZGSkgKl8p/nMlS0lj9s2DDNv1u3bo02bdqgYcOGOHbsGLp37250nMVqbO99IiKiijC25/7jY/crlUqt
yVQdzBs0aIDatWsjMTERAODp6akzQm1RURHu379fYj3yPZj0iYhIUoy+R98ElwfKcv36ddy7dw9eXl4AgK
CgIKSnp+P8+fOaMkeOHIFarUZgYKDB261WzYYKDB261WzftERERPo+zsbE2tHQCSkpIQHx8PFxcXuLi4YNGiRRg0aBA8
PT1x9epV/Otf/0KjRo0QGhoKAGjevDl69eqF8ePHY+3atSgsLMSUKVMwbwNgw3vuA6zpExGRxJijpn/u3
DkEBAQgICAwKM71gICArBw4UJYWlri0qVL6N+/P5o0aY/48ahXbt2OHnypNblgpiYGDRr1gzdu3dHnz
590LlzZ3z22WflioM1fSIkhRzDM4THBwMUcoDSA4ePFjmNlxcXMo1EI8+rOkTERFJBGv6REQkKeao6Vc
XTPpERCQpAjD6KXk19fFLTPpERCQpUq7p85o+ERGRRLCmT0REkiLlmj6TPhERSYqUkz6b94mIiCSCNX0iI
pIUKdf0mfSJiEhShJBBGJm0jV3fXNi8T0REJBGs6RMRkaSoITN6cB5j1zcXJn0iIpIUKV/TZ/M+ERGRRLCmT
0REkiLljnxM+kREJClSbt5n0iciIkmRck2f1/SJiIgkgjV9iIiKSFGGC5v2v2aWtNn0iciIkkRAIQwfhs1EZv3iYiIJII1fSI
ikhQ1ZJBxRD4iIqKnH3vvExER0VOPNX0iIpIUtZBBxsF5iIiIInn5CmKD3fg3tvs/mfSIiIolgTZ+IiCRFyh35mPSJi
EhSmPSJiIgkQsod+XhNn4iISCJY0yciIkmRcu99Jn0iIpKUR0nf2Gv6JgqmirF5n4iISCJY0yciIklh730iIiKJEP+b
jN1GTcTmfSIiokp24sQJ9OvXD97e3pDJZNi5c6dmWWFhIWbPno3WrVujVq1a8Pb2xqhRo3Dz5k2tbfj5+
UEmk2lNS5YsKVccTPpERCQpxc37xk7lkZOTA39/f6xevVpnWW5uLi5cuIAFCxbgwoUL2L59OxIsEtC/f3+d
sosXL0Zqaqpmmjmjp1arniYPM+ERFJiwnb9zMzM7VmKxQKKBQKneK9e/dG79699W7K0dERhw4d0pr38c
cfo0OHDkhOTka9evU08x0cHODp6VnhsFnTJyIiaTFFLf9/NX0fHx84OjpqpujoaJOEmJGRAZlMBicnJ635S
5YsgaurKwICArBs2TIUFRWVa7us6RMREVVQSkoKlEqlZ5yZhAAwyEcMfrydJjwGRAZlMBicnJ635S
5YsgaurKwICArBs2TIUFRWVa7us6RMREVVQSkoKlEql5rW+Wn555eXlYfbs2QgLC9Pa9ptvvolnnnkGLi4
u+OmnnzB37lykpqbigw8+MHjbTPpERCQpphyRT6lUaiVmYxUWFmLIkCEQQmDNmjVayyIiIjT/btOmDe
RyOV5//XVER0cb/GODzftERCQp5ujIZ4jihP/333/j0KFDZf6YCAwMRFFREa5du2bwPljTJyliMrPihP/nn3/i
6NGjcHV1LXOd+Ph4WFhYwN3d3eD9MOkTEZG0PNYRz6htlEN2djRSExM1r5OSkhAfHw8XFxd4eXlh8O
DBuHDhAvbu3QuVSoW0tDQAgIuLC+RyOeLi4nDmzBl069YNDg4OiIuLw4wZMzBy5Eg4OzsbHAeTPhER
SYo5nrJ37tw5dOvWTfO6+Pp8eHg4oqKisHv3bgBA27ZttdY7evQounbtCoVCgdjYWERRFRSE/Px/169fHjB
kztK7zG6LcSf/SpUsGl23Tpk15N09ERPTUCQ4OhijjlI0JpywDgmWeewenTp42Oo9xJv23btpDJZCUGWLx
MJpNBpVIZHSAREZFJSXjw/XIn/aSkpMqIg4iIqErwKXvl4OvrWxlxEBERUSUz+j79zTs2oVOnTvD29sbff/8
NAFi5ciV27dpldHBERESVQhg51VBGJf01a9YgIiCffr0QXp6uuYavpOTE1auauXXXiu4LFouWyqug7OUxWMSv
offfQRPv/8c8ybNw+Wlpaa+e3bt8evv/5qdHBEREQmZ2wtvwbX9o1K+klJSQgICNCZr1AokJOTY8ymiYiy
MSMSvr169dHfHy8zvwDBw6gefPmxmyaiIiokshMNNU8Ro3IFxERGcmTJyMvLw9CCPz888z/YvHkzoqOj8
d///tdUMRIREZkO79OvmNdeew22traYP38+cnNzMXz4cH7e+PDDz/EsGHDTBUjERERmYDRY++PQGDE
CI0aMQG5uLrKzs8v1tB8iIqIqx5q+cW7vfvo2Eh4QAj4bhdXNzM8VmiYiITM8MT9mrLozqyJeVlYVXX30V3
t7e6Nq1K7p27Qpvb2+MHDkSGRkZpoqRiIiITMCopP/aa6/hzJkz2LdvH9LT05Geno69e9e/fi3LlzeP31100Vl
xERkckUP1rX2KkmMqp5f+/evTh48CA6d+6smRcaGorPP/8cvXr1Mjo4IiIik5PwNX2javqurq5wdHTUme/
o6AhnZ2djNk1EREQmZlTSnz9/PiIlpCWlqaZl5aWhlmzZmHBggVGB0dERGRyxR35jJ1qoHI37wcEBEAm+
+dg//zzT9SrVw/16tUDACQnJ0OhUODOnTu8rk9ERNWOTDyajN1GTVTupD9gwIBKClOIiKiKSPiafrmTfm
RkZGXEQURERJXMJIPzEBER1RgSHpzHqKSvUqmwYsUKFPPNN0hOTkZBQYHW8vv37xsVHBERkclJuHnfq
N77ixYtwgwfCffIChQ4ciIyMDERERGDhwICwsLBAVFWWiEImIgUjEr6MTEx+Pzzz/HWW2/BysoKYWFh+
O9//4uFCxfi9OnTpoqRiIjIdlSJphrIqKSflpaG1q1bAwDs7e014+337dsX+/btMz46IiiU2PSr5i6desiNTUVA
NCwYUN8//33AICzZ89CoVAYHx0RERGZjFFJ/+WXX8bhw4cBAFOnTsWCBQvQuHFjjBo1GERE
RCbFEfkqZsmSJZp/Dx06FL6+vjjpp5/QuHFjgJqMp/9QuHFj9OvXz+jgiIiITE3Kl/IZVdN/UseOHRRE
RCbFEfkqZsmSJZp/Dx06FL6+vvjpp5/QuHFj9OvXz+jgiIiITE3Kl/IZVdN/UseOHREREYHAwED85z//MeWm
iYiIyEgmTfrFUlNT+cAdIiKqntiRj4iIiJ52HIaXiIgkRQYTXNM3SSRVjzV9IiIiiahQTT8iIqLU5Xfu3KlQMKawu0
dTWMnkZts/UWU6ePOouUMgqhSZWWo4N6minfGBO+Vz8eLFMst06dKlIpsmIiKqXBJ+4E6Fkv7Ro6xt
EBERGerEiRNYtmwZzp8/j9TUVOzYsQMDBgzQLBdClDIyEp9//jnS09PRqVMnrFmzBo0bN9aUuX//PqZO
nYo9e/bAwsICwgYNwocffh7e3uD4+A1fSIikhYz3LKXk5MDf39/rF69Wu/y9957D6tWrcLatWtx5swZ1K
pVC6GhocjLy9OUGTFiLy9OUGTFiBH777TccOnQIe/fuxYkTJzBhwoRyxcHe+0REJCmmHJEvMzNTa75CodD77
jd+/eerclhMDKlSsxf/58vPTSSwCAL7/8Eh4eHti5cyeGDRuK1eu4MCBAzh79izat28PAPjoo4/Qp08fvP/
++/D29bjYobtb0iYiIKsjHxweweOjo6aKTo6utzbSEpKQKlpaGkJCQjTzHB0dERgYiLi4OABABAXFwcnJyc
NAkfAEJCQ

mBhYYEzZ84YvC/W9ImISFpM2JEvJSUFSqVSM7siT5hNS0sDAHh4eGjN9/Dw0CxLS0uDu7u71nIrKyu4uL
hoyhiCSZ+IiKTFhElfqVRqJf3qzujm/ZMnT2LkyJEICgrCjRs3AACbNm3CqVOnjA6OiIjoaefp6QkAuHXrltb8
W7duaZZ5enri9u3bWsuLiopw//59TRlDGJX0t23bhtDQUNja2uLixYvIz88HAGRkZPApe0REVC0Vd+Qzdj
KV+vXrw9PTE4cPH9bMy8zMxJkzZxAUFAQACAoKQnp6Os6fP68pc+TIEajVagQGBhq8L6OS/jvvvIO1a9fi
888/h7W1tWZ+p06dcOHCBWM2TUREVDmKR+QzdiqH7OxsxMfHIz4+HsCjznvx8fFITk6GTCbD9OnT8c
4772D37t349ddfMWrUKHh7e2vu5W/evDl5W/evDl69eqF8ePH4+eff8aPP/6IKVOmYNiwYQb33AeMvKafkJCgd
+Q9R0dHpKenG7NpIiKiymGGEfnOnTuHbt26aV4XD2cfHh6ODRs24F//+hdycnIwYcIEpKeno3Pnzjhw4A
BsbGw068TExGDKlCno3r27ZnCeVatWlSsOo5K+p6cnEhMT4efnpzX/1KlTaNCggTGbJiiemoEBwdDiJJ/K
chkMixevBiLFy8usYyLiwtiY2ONisOo5v3x48dj2rRpOHPmDGQyGW7evImYmBjMnDkTkyZNMiowIiKiylD
drulXJaNq+nPmzIFarUb37t2Rm5uLLl26QKFQYObMmZg6daqpYiQiIjIdPnCnYmQyGebNm4dZs2YhMTE
R2dnZaNGiRbkG/yciIqKqYZLBeeRyOVq0aGGKTREREVUuUzTPS7Gm361bN8hkJd+2cOTIEWM2T0REZH
ps3q+Ytm3bar0uLCxEfHw8LLl++jPDwcGM2TURERCZmVNJfsWKF3vlRUVHIzs42ZtNERESVQ8I1/Up5tO0
IkSOxbt26ytg0ERGRUaR8y16lJP24uDitditUYSIiIjI/Ixq3h84cKDWayEEUlEUlNTce7cOSxYsMCowIiIiMi0jEr6jo6
OWq8tLCzQtGlTLF68GD179jQqMCIiokoh4Wv6FU76KpUKY8aMQevWreHs7GzKmIiIiCqNKa7J1+6avq
WlJXr27Mmn6RERERdUQRnXka9WqFf766y9TxUJERFQ1hJFTDWVU0n/nnXcwc+ZM7N27F6mpqcjMzNS
aiIiIqh1jE34NTvwVvuqa/ePFivPXWW+jTpw8AoH///lrD8QohJoFKpTBMlERERGa1CSX/RokWYOHEijh
49aup4iIiIKpWUO/JJVKKp2Eb9mr8DX90p6uR0RERNVPhe/Tb9KkSZmJ//79+xXdPB
ERUaVg834FLFq0SGdEPiiompPws37FU76w4YNg7u7uyljISIiokpUoaTP6/lERFRjsaZfPsW994mIj994mIiGoaXt
MvJ7Vabeo4iIiIqoaEa/pGDcNLRERENUeFO/IRERHVSBKu6TPpExGRpEj5mj6b94mIiCSCNX0iIpIWNu8TE
RFJA5v3iYiI6KnHmj4REUkLm/eJiIgkQsJJn837REREEsGkT0REkiIz0VQefn5+kMlkOtPkyZMBAMHBwTrLJ
k6caPSxPonN+0REJC1maN4/e/YsVCqV5v5xXly5fRo0cPvPLKK5p548ePx+LFizWv7ezsjAxSF5M+ERFJijlu2X
Nzc9N6vWTJEjRs2BBdu3bVzLOzs4Onp6dxgZWBzftEREQVlJmZqTXl5l5+eXuU5BQQG++uorjB07FjLZPxcK
YmJiULt2bbRq1Qpz585Fbm6uyeNlTZ+IiKTFhM37Pj4+WrMjIyMRFRVV6qo7d+5Eeno6Ro8erZk3fPhw+
Pr6wtvbG5cuXcLs2bORkJCA7du3GxmoNiZ9IiKSHhPdcpeSkgKlUql5rVAoylzniy++QO/eveHt7a2ZN2N2HC
BM2/W7duDS8vL3Tv3h1Xr15Fw4YNTRMsmPSJiIgqTKlUaSX9Uai9ImIiC
rKnGPvr1+/Hu7u7njxxRdLLRcfHw8A8PLyqtiOSsCkT0REmKmEfnUajXWr1+P8PBwWFn9k36vXr2K2Nh
Y9OnTB66urh06RJmzJiBLl26oE2bNkYGqo1Jn4iIqAr88MMPSE5OxtixY7Xmy+Vy/PDDD1i5ciVycnLg4+
ODQYMGYf78+SaPgUmICZ9IiKSFj5wh4iIiJ52rOkTEZGkmLP3vrkx
6RMRkbRIuHmfSZ+IiKRFwkmf1/SJiIgkgjV9IiKSFF7TJyIikgo27xMREdHTjjV9IiKSFJkQkOkZGa+826iJmPSJ
iEha2LxPRERETzvW9ImISFLYe5+IiEgq2LxPRERETzvW9ImISFLYvE9ERCQVEm7eZ9InIiJJkXJNn9f0iYiJII1f
SIikhY27xMREUlHTW2eNxab94mIiCSCNX0iIpIWIR5Nxm6jBmLSJyIiSWHvfSIiInrqsaZPRETSwt77RERE0i
BTP5qM3UZNxOZ9IiIiiWBNn0rVql06Bo1NQaMWWXB1L8C/p7ZE3BE3zfIRbyShS+/bcPPMR2GhBRJ/t8
eXHzZAwq9KM0ZNpN+Wj9zx43dOSElUQG6jRov2uRg37yZ8GuVryhTkyfDZIm8c2+2MwnwZ2gVnYWr0
dTi7FQEAMu9bYskUXyRdsUXWA0s4uhYhKDQDY+amopZDDa3+SY2Em/dZ06dS2diqkJRQC5+801jv8ht
/22HNu43xxsvPYtarAbh9wwbvfP4LlM4FVRwpUdkuxdmj3+i7WLn3T0RvuQpVEfB2WEPk5f7zVbg2qg5
OH3LE/E+v4f3tibh/yxqLx/lplsssgKDQDCza8Be+OHUFM1cm4+JJB6ya7WOGI6KKKO69b+xUE1WbpH/ix
An069cP3t7ekMlk2Llzp7lDIgDnTrniy1UNEHfYTe/yY/s8EH/aBWnXbZF8tRY+e68RajmoUL9JThVHSlS2/
8T+hZ5D78OvaR4atszDWyuTcfuGH9esgUA5GRa4OBmF7wedQNtO2ejcZuHiPggGb+fs8eV83YAAAcn
FfqF30MT/4fwqFuIgOez0S/8Li6fqWXOQ6PyKL5P39ipBqo2ST8nJwf+/v5YvXq1uUOhCrKyVqP3KzeRn
WmJpAR+AVL1l5NpCeBRIgeAPy/ZoajQAgHPZ2vK1GucD/c6BbhyXv85fS/NCj/ud0KboGy9y4mqk2pzTb
93797o3bu3weXz8fz8/ORn//PdbjMzMzKCIsM0KHrXcx+/3cobNS4f0eOeeP9kZkuN3dYRKVSq4G1kXXQ8tl
s+DXLAwDcv20Fa7ka9o4jJ6ki/iDjoiP88CHXtkYMb7KVUWOxmHg/PUQNHR0XB0dN
RMPj68nmYuv/zsjCmD2uOtEQE4f8oFc5f/DkcXXtOn6u3jt+vi7z9sMXfN3xVa//VFN/DxwQRErf8LN/+W
49NFdUwcIVUaYaKpBqqxSX/u3LnIyMjQTCkp/JVtLvkPLZGabIeEejES474cGEzqFQyhA5MNXdYRCX6+O06
OHNIife+TYSbd6Fmvot7EQoLLJCdYalVPv2ONVzci7TmubgXoV7jfASFZmLa0uvYu7E27t2qNo2nRHrV2K
SvUCigVCq1JqoeLGQC1nLeukTVjxCPEv5PBxzx3tZEeNbTbpFq3CYXVtZqXDxlr5mXkqjA7RtyNG9XcufU4j
5dhQU19itVUqq6935UVBRkMpnW1KxZM83yvLw8Ww8TJ48Ga6urrC3t8egQYNw69atSjjyanRNn6onG7sie
Nd7qHntUTcPDZplISvDGpnp1hg24W+cPuqKB3cUUDoXom/YDbh65OPkQXczRk2k38dv18XRHc6IWv8

XbO3Vmuv0tRxUUNgK1FKqERp2H59F1YGDkwq1HFRYPa8umrfLQfN2uQCAnw874MEdazRtmwubWm
r8nWCD//7bGy2fzYanDy9r1QhmeMpey5Yt8cMPP2heW1n9k35nzJiBffv2YevWrXB0dMSUKVMwcOBA
/Pjjj8bFqAeTPpWqccssLN3wi+b1hNlXAQCHdnrg40VNULd+Lua9lAZH50Jkplvj/y47YNaoACRfZe99qn72
bqwNAJg1SHvcibdWJKPn0PsAgIlRN2AhE/j3eD8U5svQPjgLU6Kva8rKbQT2x7ji06g6KCyQwc27AJ16Z2D
olNtVdyBU41hZWcHT01NnfkZGBr744gvExsbihRdeAACsX78ezZs3x+nTp9GxY0fTxmHSrRkhOzsbiYmJm
tdJSUmIj4+Hi4sL6tWrZ8bIpO3Xs87o0zK4xOXvTm9VdcEQGengzfgyy8htBKZE38CU6Bt6l7ftlI2Ve/40cW
RUlUzZe//JO8cUCgUUCoVO+T///BPe3t6wsbFBUFAQoqOjUa9ePZw/fx6FhYUICQnRlG3WrBnq1auHuL
g4kyf9anMB6ty5cwgICEBAQAAAICIiAgEBAVi4cKGZIyMioqeKCXvv+/j4aN1JFh0drbO7wMBAbNiwAQc
OHMCaNWuQlJSE559/HllZWUhLS4NcLoeTk5PWOh4eHkhLSzP5oVebmn5wcDBEDR3hiIiIpCklJUWrI7
m+Wv7jY9C0adMGgYGB8PX1xTfffANbW9sqibNYtanpExERVQVT9t5/8i4yfUn/SU5OTmjSpAkSExPh6e
mJgoICpKena5W5deuW3j4AxmLSJyIiaVEL00wVlJ2djatXr8LLywvt2rWDtbU1Dh8+rFmekJCA5ORkBAUF
meJotVSb5n0iIqIqYYoR9cqx/syZM9GvXz/4+vri5s2biIyMhKWlJcLCwuDo6Ihx48YhIiICLi4uUCqVmDp1K
oKCgkzeiQ9g0iciIqpU169fR1hYGO7duwc3Nzd07twZp0+fhpvbo6eXrlixAhYWFhg0aBDy8/MRGhqKTz7
5pFJiYdInIiJJkcEEt+yVo+yWLVtKXW5jY4PVq1dXyVNmmfSJiEhazDAiX3XBjnxEREQSwZo+ERFJiilH5Ktpm
PSJiEhaqrj3fnXC5n0iIiKJYE2fiIgkRSYEZEZ2xDN2fXNh0iciImlR/28ydhs1EJv3iYiIJII1fSIikhQ27xMREUmFh
HvvM+kTEZG0cEQ+IiIietqxpk9ERJLCEfmIiIikgs37RERE9LRjTZ+IiCRFpn40GbuNmohJn4iIpIXN+0RERPS0
Y02fiIikhYPzEBERSYOUh+Fl8z4REZFEsKZPRETSIuGOfEz6REQkLQKAsbfc1cycz6RPRETSwmv6RERE9NRj
TZ+IiKRFwATX9E0SSZVj0iciImmRcEc+Nu8TERFJBGv6REQkLWoAMhNsowZi0iciIklh730iIiJ66rGmT0RE0
iLhjnxM+kREJC0STvps3iciIpII1vSJiEhaWNMnIiKSCLWJJgNFR0fj2WefhYODA9zd3TFgwAAkJCRolQkODo
ZMJtOaJk6caNxx6sGkT0REklJ8y56xk6GOHz+OyZMn4/Tp0zh06BAKCwvRs2dP5OTkaJUbP348UlNTNd
N7771n6kNn8z4REVFlOnDggNbrDRs2wN3dHefPn0eXLl008+3s7ODp6VmpsbCmT0RE0lJ8Td/YCUBmZ
qbWlJ+fX+buMzIyAAAuLi5a82NiYlC7dm20atUKc+fORW5urskPnTV9IiKSFrUAZEZ2xFM/Wt/Hx0drdmR
kJKKiokpeTa3G9OnT0alTJ7Rq1Uozf/jw4fD19YW3tzcuXbqE2bNnIyEhAdu3bzcuzicw6RMREVVQSkoKlE
ql5rVCoSi1/OTJk3H58mWcOnVKa/6ECRM0/27dujW8vLzQvXt3XL16Fq0bNjRZvEz6REQkLSa8ZU+pVG
ol/dJMmTIFe/fuxYkTJ1C3bt1SywcGBgIAEhMTmfSJiIgqzgRJH4avL4TA1KlTsWPHDhw7dgz169vc534+
HgAgJeXV0UD1ItJn4iIqBJNnjwZsbGx2LVrFxwcHbGx2LVrFxwcHJCWlgYAcHR0hK2tLa5evYrY2Fj06dMHrq6uuHTpEmb
MmIEuXbqgTZs2Jo2FSZ+IiKSlikfkW7NmDYBHA/A/A8bv369Rg9ejTkcjl++OEHrFy5Ejk5OfDx8cGgQYYMwf/
5842LUg0mfiIikRS1Qnub5krdhGFHGDwQfHx8cP37cuHgMxPv0iYiIJII1fSIikhahfjQZu40aiEmfiIikRcJP2
WPSJyIiaania/rVCa/pExERSQRr+kREJC1s3iciIpIIARMkfZNEUuXYvE9ERCQRrOkTEZG0sHmfiIhIItRqAEbe
Z6+umffps3mfiIhIIljTJyIiaWHzPhERkURIOOmzeZ+IiEgiWNMnIiJpkfAwvEz6REQkKUKoIYx8Sp6x65sLkz4
REUmLEMbX1HlNn4iiKoz1vSJiEhahAmu6dfQmj6TPhERSYtaDciMvCZfQ6/ps3mfiIhIIljTJyIiaWHzPhERk
TQItRrCyOb9mnrLHpv3iYiIJII1fSIikhY27xMREUmEWgAyaSZ9Nu8TERFJBGv6REQkLUIAMPY+/ZpZ02fSJ
yIiSRFqAWFk875g0iciIqoBhBrG1/R5yx4RERFVY6zpExGRpLB5n4iISCok3Lz/1CT94l9dRaLAzJEQVZ7MrJr
5RUNUlszsR+d2VdSgi1Bo9Ng8RSg0TTBV7KlJ+llZWQCA4+mbzRwJUeVxbmLuCIgqV1ZWFhwdHStl23K5
HJ6enjiV9p1Jtufp6Qm5XG6SbVUVmaipFyaeoFarcfPmTTg4OEAmk5k7nKdeZmYmfHx8kJKSAqVSae5wi
EyO53jVEkIgKysL3t7esLCovD7meXl5KCgwTYuwXC6HjY2NSbZVVZ6amr6FhQXq1q1r7jAkR6lU8guRnm
o8x6tOZdXwH2djY1PjErUp8ZY9IiIiiWDSJyIikggmfaoQhUKBMhIKBQKc4dCVCl4jtPT6KnpyEdERESlY02f
iIhIIpj0iYiIJIJJn4iISCKY9ImIiCSCSZ8qZPXq1fDz84ONjQ0CAwPx888/mzskIpM4ceIE+vXrB29vb8hkMuzcu
dPcIRGZDJM+ldvIvXX3+NiIgIREZG4sKFC/D390doaChu375t7tCIjJaTkwN/f3+sXr3a3KEQmRxv2aNyCwwMR
xLPPPouPP/4YwKPnHvj4+GDDq1KmYM2eOmaMjh2ZTIYdwIAB5g6FyCRY06dyKSgowPnz59GjRw/NPAsLC/SLVyBQzJyP2RW4fewFpIfaDyadHxbnVh4rMpDQx86EKcmmS4R4oLLFPhI5bMQZ1fhu4JtPNgzsvkvzsX
GrIekIy0ShEMVOi4wURmWg++M+jQ/mqcPpq0oh5YV6khZXuuDuxvdv7eVNGXU6rSKi0Y8MXV8i80ExBeRjK+d3l0NsZB61UyIh9zBRmyOpq5s8zlG1DeFHr0TvGlXbR4UPS4YYaJRqYOp0hjfxtRiOKjcmlLzfm7l+E2fOx9MQ7mK7G+tRnyRzGyJ+6oKeYcifEQeQFbPX8t0VEf4ZcMnIZhvJDJBzNyr2s3QoCxY3iY5a5f8PgHbRmyVkdQnzvVxqzK8sXAVRkVPeUkpkOWP0mwRTP+BUEF2Hy7oFVj/qeP29Ls/qMEr6tVULHxv/KGe26znQKcMiG5dXe6IxHvFUZAVUw/6kbTtn5Q6x4t7XZrvKqmi2DNmCwuzaLQtIx/dllMVBqkOIQ0tX3hLS6fr/hbwNnGlzPK+b7afeAETcrL2Xo7mtj7Xhx9Pfta3oVCuIJgwCMBPFt0bQ0YQb8yjHl4k8OhB9XU37GKvjpf4dtSCz/GJJpFd32fsVGAwbsZ2NGuKF0c8OXOb/nmnmv5Dk9cpPwA1Rr1QjMpQ04btw2zBCD7s3tz0GLPMtRm9LTSywsobl8zDpt/5fxDEm32KP3bzzg17aXaEz5PVNmxu2q3X92yDjZcuO4xL3ek1BDeEA/g6nY8jd4n+PwMWpV6mxBv0oqa5d8vmiOwDrEYAQxbymKkvNpRVyxSn9vm/fw+b1W/tMzMNnC5cdU8W0f1sP+l1o8Q0y2bBrXq1yH5tmgaPkJ7mDyFXSHvn9WaCOGYbYiB9Fkk8NxEmYvsm2FMgXaXoD/+m3EXZDEd6XAsBv3wLMxJ+rFdi1NqOTQE5VKbnzsQTvPXVRTbXcBYgDHZzZLmLZfWtc2UFUrywRVmprDnvEOnIWF9wh4QzoJLjh6i3P6i0BIIqbfXJHGXa4ETfEdRzhtqO7pZm+bYxDtkZLOo8OSBp48bdJKeUKgNFrtWRtbu7NOPUJ/4ADoyRiBUg26v8OFbr1UYdtLBjt58nVb3u8Cy/c0t1g+jXGvJ0/pgmKMPFeN5aFJM9E00vvdYS4wqYFYLqVlr0GGp+CO9LnxebjpXuyXoPDPHHEwAJbTgHqtqXQLBELUiUyLfvHFNnnHF6tTdcBNe9WvXq6VXcQ40+sGRJIo7dqXB0Uxst6mtZFg9Z9rF4SNHlYcx+TO4RJ4yMlSP1JsmsGpbm0P2m6g5hh3mb0B8zAHEtqWJqSOyXdXcEtM9sdRlkbyZNz+gGW4ZYLfEOEXREnS5/hJaVPXCTfwxsafzMeR7tsy2yHgUfUWyITJPxpgvzb/tTGXJ1cOR/fLzLTgVKRxr2MgOCPfufZJUvCCsYwjvoMGdiwhyUIp6/RjUGR1R4efX2pSEV9QrgAVOOvf0m2ljfyTy5vJiynW9NyaKp8fC+Mb9J1v2gb6E+yfFVWUGTfavSaZYpwKyMsXS6zZxoXmg1aP3RNhhv2uKGPntC3ssnkx/y1KPUV88JjMlZ0P4bnx4yJvWkQXm9Y5ZWg16ZtqH3o7snhOUpE4sFG9EpKThkStuhQ7YNgGVgA9o2tZuXYXZXmKn+PQ2LVkLqUywzzlZL+rlhFRZ4Ec18hdkA58cYWBxp0MIDHyejrXyM4UD9lpBHEBdLn8QUC5KGK87L9SiNl5lvAFYKRA0KFAbUejfQYl9Z2CVzzrxbiMH3wvW1ljuAn1uPh0NPZ5yI7mkZzjVaeh08NHVb92tAbC2iRE47QxozmTELYmSAnCvBpIjrBc6ZBd7LVf1Rn7UQvUjaE99lGjmBEy/PzQw9LV0vCrmfDSpjKNnFiPZVEyQpxjdLNaDh3SoF1GTgFmbyMjNf7BIaZ0G3xcnVfQfrR5Yqf1d1fKHwfDRP7VKXzzvFdWfy5U3GXe1A7gSEKBtlbvvlPJ/BNlw+Ve1J0aAe8fsZEm2QVIvWV+0IryfZNMLv1LBEmR26iRqqjgmM6/gXYzO8QT5HYtKyHW2dpHLXzNFqOQ+vXYPDy6HNAXP9tp2+M2JxTl3kSSF8YxKIq/EgzBg6wfYZKMYLCBi7UIC9uPwqjq4KaDUY4+qGzQZHgPc9GUDHgAwl4R4kZkR/aFmsAD1Mh1kRVCoR6EWsC1sUkACrdYnJr/kFXcMpAMhRKIYDACUWKQvWqQRjjDI2xoM+Y+hOPo1Xk6Lfl7Y+n34ZyQ2hwKqYpNPYB0BcBSCVuUgrrtPmf1rd32/k4zTFW6Ib2WxRRYdNPaR1m4qbxQI8TjtEPv1GNPzJIn2z0SfvlPvWTzGAvfIapv6HDBcuYsLrvJRDcnoFHvupz4i9BdCfBQ2sKR7OhFNYrhzKX68F4kQ7LwEQRkNEq+RQRtbgvVNjKJUvRFMtf8KPPo4rz6rqdEVKJ8DIOMr0yntbWm2rttDXzKZaWoSFb1lDeaNVJr12xRvt3fMkBXfMZ0mgW92zwWMjSlMwEAiwTzr2mm32ZMGpNOmThjkuXvytzKPdcHD5ViCFSxC2ui3TlJpVJW9tVs/V9SEwvoHkcpw4yJ0FC6UORRpbXlxs+fjOTgQ6ZfZhbXzxzGK0JV4DjVKRpf2m1EZNkc5ZvB/Sa4T9UR1OpZK1R21/Q0BzuwZnCrrWT/T8Bf56u+zPIHuwA4GzLrvXz1iNr5K+6FXj7PS/dORRpFFRm2yBGfPdNA1VSyuZvZCkG3qpjhW9MYNd4kkvB94qMT0IyJnt0hUEO9mN7eMAY0O0lMHTnNAl8rTfRu68BZX5W+19ytrJpFX1bAZ9HIg7zCgUKvJbr7ZN66XPmV/KmwfaC9XBdmWOdXBYbJDe9S+kkOQaX4ZLcDuTsIhDl0Xe9uHM0ZSuZeUYfWH1Bx6FOxzVJpqnLY0o3VSfr3ArnXPHxyRyYdF+Vo1m/ZnCy1nBpoQCuppqDHcTeYROAZNUllOy9kSS/Ek2e4OU/sY+gK5N2i5MR5D3Q8q9nxc7dWcJvvwWhAhnGYWTPyT02wGfvYWtaD0PrXvlW+v3gMLv/xEgXpzv/PEQ/ZEJl+mB64s6zXNzBdeNv6kdlBP7k/SWBNSe1YFZszUA5+4hxBAfeO5vLR7VmI7e+XPcc2X5Nei9Kj0njlszJ5aPN7mnVxeg0sT4Dy3NZAGOnXQOvQ3oEixp4vkx8gH9X8Q8BnxBSv4bcnURPZ9xrWGTj0wzZL7/wAAkEIvrhYOHQjXeT5pqe0PDsTnmG2JXMZzU0Zs9VWzBEfptxFZlHp0KcNbtUMtP1OPNIFlI1mrHZZkUDeEuR5X+V08SE0QxR+0T/sAYzKhpb4T4ezx4wX+uiB+zLsFZXAIRkOP2Oa6kV8bAeAZ05+Qy2+T1h89TyU8jqiJqGe+fTshhZM0+fS2ySZH3y3bYFxMSrB7W/WYUeMb3g3cRM4wiDsRQIQBZdeARxbG6uHHFZJSuIVjlp9wjhYqFCOXQ0qZh/BDXOGf1i3m/Pf41XZ5yVl2tgFvlhmZnbhAb6HAEWi6Sh9+JpS9+wG6U9ZX5tBSfYz3j8dzO5ljkBMlsBOSjOJJn3DHHTPpAZfl1Y8F3T7SewCPf3cMfg0a+/ZfRDY1DJJ8EIRCO7Nx7V5oPNnaZy4jXHdJrEiU/NFP+JbwMKLBZBFTy/yYDkSzUVQhVBF4/aHsQgLr7eMHK4Nv6V47mkRiqXcF1ZXUfsMKMUAD8mkmW1Y8Ay9H7D3Dp6wxbQWlj0hnV5tf9OoBGSqPbOjOSGAdnhCtLZCBdGl6BTJrwSTnd+vNV+L2HSs5tX2P6i5Kj+AEPGhbZt1IV0AIKDDP6z3n9oXP+Wp1KeWYz+9EbbsOvcU4b1bDBaaszdJTq1q1WsT7LRoAFNc/CqSnlZITUeeZ6Chex/p7rqq+ZJ/lOiotf6dVrUn2k8oBXKfuEzVY7Cm2oSYNNvMvP7xYfQ2xDsv+GbvK+N2C13C6KTHeGF+AeSRYs/Zn5KxNk2uZVCJbsqe/Jx6F3zCz/3dmuZUHpuMW5SRrZq4jSPJVbZoczjnOkyTKyjNb3Y9gUvk1p5A4qz2c0ubC0lQoo9tfy/wRy+3nT8wv0Wt/N1w7PUPTDDX0eDGLwcNTFgQqNnU+/3b/JXv8g3hGgN4WGVZ+bIR0a0qKU2h/A8YJcC2+e9mHtb81WfWDLBgqfiJDwD9vbfqyjkDxa1RbhGv9GUG6mUIH4GAmVWRQ7OXwHe5LQX1hUqR7lkSJZdDsPm3ZfFuRyKkk0eImT/uiu4eOPbVBAMXDcBaozfw6KKT0MM7y6zv/O8l1ANvXLI3aVtHNxaRSD56eEbqUCESMoBtVLtRyhs4dj8W5Vxc2W3RtsS5jiEScIRTLqw4IcPM3xW6eUFqM9Ae+vgjwlcypIEdPnyVAdfcT/TbvXNZqDvQBUjgjW+rEk1+TD6ALuTjnjczeTDe2GkQAGzoS6BrdyzZCnnfZ2h6dmc9MPgeMmH/yO44+2+q4gCnzhqHiRzX7FXEwc+jR+cbW08ix9OJxCX+XDB0OMTgkv5qu0EVo3XJN7i4F5tAqFqrTpfmvCjRiNP2S+qWxeaOYzoQi7QSG72q71+B3MZOOQKhtGg+aA0xrqOJwNfWl3BgBb3ljOmp6rS3Mz8frdtjpGdR6Qye7SNb0KUMMjCGeIkaVn8Ngfw2gpQPVJzvm/VqKGiBw+oOqdn3nAjjpxk84LzmLhqsTJsWJ0rWAfpEZuCbvd8HvfeeiFAMXyBubQexCv6XkZVE84wiWPIwJL7qsoPhe2Z9Flpn8cfvBrYGqMQY9H4fpgV7KqTMCVsmR71dNjrmTVoN3CxfS/4WfPhdUtkNwbhSaZBwp+/6DB2Y/vwFZF9CFJ2v5fyuTm6G51KgBFkNDq5BxiBe72ifZLCgO9TbUOOOYT9Tat9AsK9vnBPtJ84+r0JFJiKu6vkCtqH8rNkfF3kUxxGxwCFrZnXbKNOPCm5ALM5zEKcxyRT+5k3SNZo5Gz/KuGvcnvCNrdSxR9Y/bTqnyHgFNZHUgUWbhw5n1wDcnV2mr2tQZ+wcP7ZE0CGuAJcBB5UVDI0ohLlxpFYZq1hFw98mIPcHsJH5ZMLmwz4V5AgCEAfzQeOlB3W8wZV4YWw0ehrfGWwlqIb4pP2d4f2+dU+VqaOsIVqcdNx0O7I2Q0FIJRJfsaYl7KFt1eN/W+GxW6ZyNvXpXNFwQnPpyJ5uXVu2w/gH0BR6xKgvAGXAZLJUlKh+O8N+C5b1d6FkX7f3HHqSvOwHRzJXXJAIQX0nDiG7J0d1svXL0Ow5fCPAYZg4AE7NLvKO/l8QFiljzkiZrVGERwQQxw+1AV9NS4m0NcxvDdEjgqo+z2bQuiCaLUK+EgX8wiMeXFkpcJnRD8nl8MZ8pp+ZMgSrcpCH2Xvj/c7zsX0J3Y6ePW96TJaV3KGPdFvvhWzXpg7Cyq6/e4YaKfQYiAzFCyYBZ2Zm2bQNmu2gHjW2WNhdyLsXiZ/oUpmc/qG5F5lGbYGD4fr09ktcJcG+w/+aTQNMv+x5d1zvZRz4JQUUB2ykYOIh1VSijRd3XVGG9o1Hry3eP7SGUbkePWg7ub1Io5aRpSJjUsGuaVtjENTKTDYNIVE6ZTrW8jTOqvgMUlfCjVyUAS1Qp/d7aWM7y9j7jjh4KiR1wnEpMd8W2T6e4RKBXuW1x4mbi6W/x8Hf1LyYXZgWy6CvjHaQdJy24uWZc46S6p3tjYTwDZjsHjYlffkPN7a1J8v+wp2mBkN/i9Hhuh8a9dyqZN5dOxoDHITnzdIo1m0Lg0HItZpJaYZ53d4wqDw8ULEV72THcEAMfWfO4ne9R79YVXo0Xip5uRhXrW2+tRB3pEeYlr1z2Gdb5jrY0AWPmDlOtfcqSWmAvGzJ9hd1jdMQKr0yCRSKdVdFzn/Zvf7ocOUDs48pIeL0Y0/uIErbdnWSGFQx3jHKqm1Db2uh2qTbh0kfWnhnKU4SPgzJSeaUGtHphmPH0MvF0n/sOnnU4xsfHEdQ/0PMCBv8CAp+1X5LUzKuLl2LlwlpvhGhqIc6sZk2Jqz9X6VmLxXsE/a28yXo6mkVHiW4KVeDD8FaUGNuzpiKTFgxqvDdHLmxrjgfcl3AcDVkh97Uir1ClNaZc2Xyg2wMq3cmt5GuKp9d1LZ1yWZ1BNsdKzxXMI97J4L9Q5JBgYBQnd9o7oqVQ1m7oqYC4U+2LiITnW2m1Ezph8UGQLcALxgdP/D0/bfNiBmDSm4Hn5l0ZjbL9PNrDC4yChUx4vH7Rit3gaxeUhqWDC5UzChc0NrJOWYj9KNh4bF+6qDqJSXaAYL7Iq4lQVeA+jDJPBv4L7yJI2sqU6Dz6xv/L6AIoVUq6TgbGF0nVH3bj+SNBVd9HsbeSUMgY0w5mMPIjCbv51oxVJHBL+Ko+GOj7zdrXwTyv0Do8PfQ8lCHmzdptmgWaajBrW4Us36+HMv3jPvTD6uQ7dzj5sP2bCo8pa2PcvHjkp+D9XfUkXjsyqlPhMadeqgSnMZSIU6ivp0wPHuQ8gr8M4Ji9TBWoGe6dHAGkAUBJrrhq/o4/WHvXbkt43+0hzL1vu/bNvehGPEDC8A5X4Bm2x0hLp3vB0F6XWkHFlMzLkgMyvTOAj/i9/c2c6MmxDnXJVF+mAx0ji3P14ImlG0WuRJ+I5hU7DkrNpZxv9rQaKp6+JGqaEPPDQVbPmKsu/aD8cvTn8a9BnPgnNqzbpKafvpmI7nTC3O3P8OALz2jE/LaHoClmp0CiJ1dEvRUPjuAdHOO5s6CrmNpk5LuldGNlMQ4/aR1f5XoG6Xe88x+eJmTYcEL66fMSbWGMiWAK1nqbYCg93eDwBqUfuSQGFEx1AxQw4jS4ojFSK+nvyAhCmgrCgnZ4h2VX7WFUiJxjT7GVsMaBDFa/yvE01LaZVddkUZN2+wjE6UgJEFFB5j9iy9mpr0L3CF3f3Sx/hbnhBqoeOx0YvnWfLM02xGzvJZM8ddLuAjIxcYPvbHBAaUhYqM49jJVLdKBWpBrpSrDcFZPtuW9BKXBoRnMKiEOiKEvPBSDTQRZv22yLWmNOA32qDrGHJGhjdgFbQtzyV3sHuxM6zr6HeqrVLDjtmNlg97Ao/kkDTm5MRVp3Bo1YAQkqp/29BlsQJPNFVkYYC0Hfm4E+pjeqi4IlmcZ4+R2wcMDAi1Pz7H+qCNYIp8xsLDxW4Rwc6RQ0MPntKVQQuNMKXSQaxhV47pGbTU8VMpm8Q0QZPPIFbcwmtIZNeZVcJk7zKvsYRR/QIDZ7sN1hOSGNhOwbnkjw5F4Me+AzeFBxTFMNs9hhfb2vRKZTNdAj/qB7CR4oc/Bc3S9qVaK+RJLGnu4ba3+wiXBpDRKOeXGiBtuwuExUg0BAsQSJJEFLBbUu6+eUFDbUAcgHzGJDsEjIkvbK1Bs+3yQHBuDtrc24v5rBr8F8wmd8nnJ3PHl2T1gLDdsdg4WKSB3T1ZSm6DEqBRQGiOWuNJ7TjA8ARHHJ+0x5Abe8dtGXCwP+rTDqt+cgFg4OrmpsU8N+Kvm9uP9EyQ7K9ttIdQfh6HnGb0QVkIQbUu3m89Km2UqGL+5bixMhz46G/3b/bk/8S3f31SszO9u26hjn0+o1mLPnQqK5N+MaJQJnpp6HlsBZ8nMNXFKajnJG+uqw91gyaVAoIO6gJBRtqVaK/IACQtnLo8C+uw3V8AlxH1mv5l4GcTnAjOMBW0X1hnhj5aD3CBNp8kYKwv/JRUNq6Ow/TGnvlsGGfNKs+13pvr1oUMDSwrGdhyEebsJ/tHFD4Rj4TlXZyU0tX7HVKY9DXZowS/PYGwbcdvfz1h8S5Wh2dKjVTtjpH9c/EnPR1/CmD5aeSJPUVCm9k+7+BbnPp4zz9dz0zIzcnSj5JbRH/hT8WIKjnPVhc0gCqMWMhnd6kyTaJ1Cm2eOxPtR2PzvfK4EYefeHqmn9L81m6Mzrd8QmZ3J9kt0bzrhknd8Wtzlks8Lg0dXMSQS/LyiWvXyTkJ+U3GpTBP0bGnJLWMNu4WFFI0w1HFZuPA3TiQc6N9DPdqaSjqtnF2MRgGeN7ixo0bFjKY7kpR3NMbIK60Ol4FGrvr6jqcS/7oXvdEZOO8+LIUeWpOYfhMR7UugwRq9eOQaM88KahX81F9+5L+dYFSWAyKTSdC99iEUqnBGhDBSkK+89+IO7GDHMxQQq8LdFWW3A8XQzbFxVgxwXq3pUGJJNWkuXkXQZyVNeBQI+NI10zwPxnjUs9X8e4MHOPV0ZjwpdZq6JlYOzzH4SJ6ZLGi1Lt4ykbJbPzpjhjlcrYvy6+HPOrS/dVPhKMcsNknr9uqu0yhGU7E4nuIvXkkLi9IlFDBSXcOFRpo8R7TVuGPtKz1xkVHgOpYKBEk8YOp/vv/+oIbIsHA4fDCbmY8bT4O8J9RC04PEgL/oo/1ptlJE6WeXIw7w9o9unzPuVmvwRKktIGzCP2xPFnrTKdPn6YnlxlXTLVFGkjJnqfRLXfEe4yq7KXjzTFqlxUGfD2tQ5LG5eb69tW1HtkDxGKnCULNo4m9GjWoayqWuwIqBDewFqFEiFOjnjRiOipKU4iB2VvF7GWM3nGyOCSr7DzJRUZjpdwgr6YBfDGFszI+BAspjX+tKSW9uWlYe8QudKkZKmLKHGNSGUbP3tT8PM2zsbbQgsGm7HeGDxsQE5CrJOhxZ3M7/8+n6JjIcRu0EJ06Gc1G7IdU1jAt2oXyfaFUeGAkWnKWnbBHAOmSuKDNpGJu4NTuAM0mxOjcbC+YyTbNTbLnR9jWcrJCfVGqfW/vH+yI73xCB0xObr8wL+DFSV5dUq+zUaR+1AfnPs5GrjlNVt8mxwp8Ly0yC4xZnybFL21LmvqQd0ylwyJSd34t7T/TyR8eKuvwvNjzLoXB6Uf7nm5oUSi3X1uxAnC/9E4W2zXKFjPlVh5L4oZE7EvZFwmuqyCb2DoeJG6UKs7NS6ZGy6BHG4q5+gUJfJSfG07nfsuM7t6AqKtK+J0k0XlvzaHM5bSEXvTLWjyXbIgfXYgOTbmG5dYNxeN/Hf1Xh8jYpoefbjaH5P8B3TkrjgQzf4IZIZPfxA7zz8W0ZWtxqzIbwiDMQ8ctFxRCIzNfk6YJPBfUPhP/Ss2+2yVfx6rP8DUVprgdQhOx+g8HFpOkEbl2ziR73gKaRIPQHRl5B6vmS5oNLHs0ql0Uc1nhyPFG9kbKB0ycMUNHTbClWqtJwxl7A6T2Vnu3hFDDu5bWpd+0mYddXzHXL8/dq0eLXhlLlgnAdYtU6SxIcpEi5V4fwDoInjyLkFtWZeC6vaGyr4EamZb+ZuRLcGwUb3eCCpNGdfDRU+nSqDCMTv5pd8ljFMgwSdaXvcbqT7SGvJ0u3jXJV98LRsF0e6l4Gx6KcQvF2O6mwkTF+XyNK2H0ndEXLbFiaT61jkKa1NJb0UEtdpC2HvlGjuZvF6+vNuHCbVNpN+DTOqPO8yQaYGhi7VnvlMYhzyNBC0B9i5CX7xe6DSdGUo/E/UyM1bPp2Pf3pUl1BGsRjdovtX4mBbXv/AqmlcvvHcd+tGCmOAiDE02tUS0VP5b7IvZrWRiGTZ9YFFivALWJjizU2xK3Ce2BtTGhV1cbCdgHDLXjeHmZ8KvODxRw4XhiLdbJE2B35jIZpxnyOaPmR8HyRGBJqIS/8HE+hGOoKqfX3EvDfpbi9BbSGrlEymWBLZxNEWqTr9m7cUC8tBMaf0fDYAr48YLUeHjz3jEsWxgU2R+KnFK1fCmDf1IHs8yyT9q/R0J9fFhPb4m7HJ1SUGjaTeVW01ar+oqhhsKxvYCGo8WFjfgONGmLgBhOMFc20pHdWgJ8XPF6E2eUNOWQXhlJ/NsUwCgn3Ep2M1n0g8tdGaGmu5ohlaalMO2nJqeN4D6Fp/t/cxa2WUqoc/KpUTL/jNTBQ+DfuBxrdw4Tm5LntZCFWaRC3Jf9wOl7e2jeNMEUJVTZjuF1K1tM1gjXLu9dJbWR9NXm+JHXTh/SNG+N3EQbmUyqJcWGJB7N5bzj/3f1X+72fuhZ1d/NAf1sxpQl0DtrbMGWq1a2eJlyo1oHuo4mW8YvNAqYkL3H0Cf3EPC8JKsR0mBDBFMLnKGQG1ISRv5TNt8XwQX6tEbKgtI/LP/rFfjgtAq1gZGgjGmz7m+I3vQ31N8QbeN5z9oCE2z3GvHt1TUhqvR8VZbKWbC5ehxWHmBvG9kzKiPF1AFLtLbfGZa0OWtO+4Xhnrzfi+B80jLJnSPnAR+I4vPwA6wnJmBv4/6a8P5Qs5+ru8FS7bMSYDdMtQ2gTgjCdHfRAzNhPp9KzWGQ2c3ss0GsK04s2f7tnq4JrRfG5xuSm/K0rBrzH3YMpv0J6Rhv2L7fcRmiXhK/TNmgfMIF1yE5X0G7cXoQl26BtWSqa4wyq73gd3J/ptkBnyVH8y/TfVvgOa3nMKOBXdKk1fMzqGRVaDQ7Z1Nm+OuUEDWwRjgTz8PHxRngzpaQVSjGQMKAs8HbfWtB5Ut+WGD5xPbqxLuETHrpuQWXaFn/JbHbwA0ZEd4+L0rxvuC2e2rQ7XhDz9RAA7oR53iuZhUuKXL0nTzjKYV8vyZ3Xv7ZZb2/gdWxHXYobJi3Q0KyLOJm8Uw3+5VDrcwZgtDXGzUX+nxUhqemMX30kk+P1M0ElNRNfQnKy3sP6/U9A+7l7ePcnQ7sbGf9Yp7L4lv3nq5SFXw3U8GGL3DvfYPSsyWXXm8Sd5WrmL4qKcw/XBHhS6ND1UgGyKZ5ZOdzW1Jxo/4BeGgPHymuo/T71ODXR6q9cbqOhXAmh8zvOZjjDgqIyrq+F8H/4iekyjUZAfgowf0jOaZEt2Tf5BZKpC3ufhd1I9W9ICBQIFbEBY2NKdg25ZFqIi9cVBlJ7rmnxO3u7FrEa8jx/QLMbTe/ZEyq8GaqGhIZa5R3zn7zW2pDw9F2tZrPDWeD06uIHLU9+EXz3QdPMcEcd2rqMqI4ZwSiVFDK1XDVD/BlJfU+qmkvbwhMPUo8/fTGvp4U/NT2PdK9KQg4cCrP43L8eJtNtU0G4Y/0Pl8OZa6BJJRvC0xxOi3SjcpU10k0h2FE8dmzOGN4Nw3J90lA3WPXuvFb2nCl6vVfJjAMfSUzkENFNX+pzTZZzr5kCzRr8kIZuE+XvfNXFyvWF1GrWClJVRP9KiZVhd5Gkc6BGvnG5cZ/M8nDwrt1zdB1qlIDaKhHTm+4F3z3rQLZMHklLiU0qb32MByrI3IK8YAqcOJQVGhTBaodHtEAUg3sDKc3F9L8I9Ue7fgjwaoUKN3oHkXKM41eL0g3TDeEBK5sZDkl6fiRhYGsQ9PUQ/m4GIbfkbKjTDfGNVyC1RffZRYHL5ltHDazAAJrDbnRqNe5XXgP2+rE4hsrBxzkpXbtEbfnPYo36Bjbtyk7AanvWbw7ENx+EPbP7Zd8OJx/bDE4CNPLSNpDY9KqlOt2TV9eoiQW4Sa97yRrCNmxLrR8aN7hyVZdBDoq5hW5x5Aiwov4VRKCXkDTD4C3Nc5wf3zqHrNS7dFY0Qmqeh1YB4OgRBKtcfOkm1dNNMhGMj7a3zmjvj9Wf0cJ/iD6cbiJzLoaQPAU6W4UBbkVyXRtb2pH+JYwrQR24FYO8jvQ1e2xNJOu3a8N5nIUhHqGHwbr/hgF8vOkLk8kxmx30CAKuP/AOoVSHiBhoDEUdHkMxFoIRSJHkUj5yX5iUSIaNf/g1chYejdDfqIvzLyrhtVGr8gRRiiuMeDCeO0CYDyRqS8qKdEjx+qKUeS8vkuw2QnA4VUTBGX8cEdKTyYHBhzQJ3iTkH3bt+WlP5xbwCeNdVzTnMPMrG0VXJ0u6y4Y2Q7PFsYTPMn1w72CjDb2DeBvhyhR26y9CjeNNiU5VyrVwkiDzQ3B8bxjMYc7YlRyZRUKJ3LRXjsdLivZ0V/92/u1mh+VXVpGDXyk9JzTl2qCkpcbGI/QaIOB9QSyEytWC4M5S3iXtjkCz34sGrlQOVdFDPNYRPHfL2rkYtJiaX6j1VVRqhsXpr5eT/zYqnV3/gdcdz1VVUZK+oRg5PPu9TV8kKl4cXL/tzm3zyK5Qv8S1SgrVjm2Jh11oN1kS2YcLdsn2eggG/JOEjHjbL2AldrDrVnBpGO7vZkVtqI0KtKTdCs3Pid1J2xTOVNuPp96QqvjP2Rm9i9r+hL8c48HHFpgrpWaCdzGEmcP4qkgvX7Qy9yg88I1jpzJCF2KVj5bzXLYPjtUr5GuNqA9nPzQ5ONM8mQ5mRHvHD6/LHTbM5+dz5UTx6I1ZhWgL1KqlLXHnX5x1o44f5VYs6qHmF7KxqkvUl+UwkSe8UbutWhkaAN6z1i6aXjrfIUDvmGTmQrmCAxDR3UjNzfbYGs9l8vpyS4wiZZPZOhEeLA2tLuu/i+/bx4kNVz4m2FRzeQHlkMCumRAN7ugi3PVu3WYPFSyhI7ljGwaxjuvUE/7BuE1HRHjpPFAWAhGHZJ4+9vhWVw58ENPwFKubNOj2xUvPFPmOhsAgzQ+ubt4HLbzXaadpNFMahlUjIaxMA1FO9OF2zdeZIrxBoCvL+O2bjgMjQ0sztK0+p9/8bkL0fK1oNHmd1ZNAFYsGvoo1BXNMUKaoElsAdmXE5DV3Uhmb2QEEL3ebOEaVQD0oRZfqSItGGl/5pS+xOvW+IWdVjzPlkNXlJEhz5phAQb2wsG2kGNUcz0Zwzv+vdPdhVC6rdSy+JFZGqwN9GhW1Zs9C6JnqFtdUOQMSYY51oHKHIuXPYU5vB5IXj4T6Rgu6jTzm3wEN28f/1RnW6HVjX4d1VEctBUvf4XW6KnK0cxWj/LnGJsUQ8YcuuxsnXtuXsFWfUuvIeMhyKiLTkPeOI+SgEWDC/wbHc5TEv2vBkcA2t+rBNUiPtZxmeuIrsKLIxTljOzVhyoIJ+vyJDiXuwQomrcEsEO1+tb1lamSsPqYU6bmf35bSvDYL06k5gYP4SS5YYQ9bv6vOYTrIbvkYYNJUgChE0mqDQTp0WTk8yFn4M6TcCZH5ZbIekGBmm/RhCiJqdhQ6Ph8XFuh6Yb0E3WljmNAdEzbz9CypoWlRZLUF3z4bzs/OVQvyZ7gUEa/G/Vkwlx3OmsDCYl7uchxg1+pscvzhJFZ5WU3uSQBz3pxGi2PZfPEpHoY8Qz2HwsMDtZFPyWjSPOLAIzCTizUGRQaOUjW7fGk0l9Ne7dqKD08f1hBfyXjSvwvJs8tDfD5KGPKjkQj6WmjlsYAqSOpc2kM6nkzrz4REswQ3sR5HsUsXaQ8VA2+frc/rxzJGprqLfftO4iGUWYRSoPhJl9gdHEj0ZHdyqwZvRdmlR/TehcDUY5Un4x8/oa2d8tkHPXfJAEC6+vvYlwrD1SEnqzJMWv5PHHkVUl/AnsCBsQ+7xI2XQ5fFHXeLBUI1GW8jUJqtORmlTThaxk8iucsOcVzvySrvl95tQAwpdAL7ZeSuXlm/xrVtB3pGomzAGr7y8aG4a8q3+/ayRQn2zpFH+dFsu9SafN9N3rFhWS8q8AWr3kNg1XeG0kLPkRLaevvQAX9KpVHZMIpqmqcHM6VVVEa97ZXqLOibsYqBhQ/jrv8r0HjE0O3E+wpKW6mNNJLnvT5RoVo5e5Mfq1AbZknvYHr6s8BEOZQm8ZLn7BhAGbtrYw6PQPO6S3hUwHoDnRlMmTSSBvvHwCUBdrdGt8ucGTGe7Bf3tG4V37WWKlgeNTXdJThhqNE5eDDxBRO3FGfr9DdVaF8+eHM9afoaGZ3mLmqnyD9ZFSIbDlbUmk1WZ/zc9c5Q5zXXa/EmuxEpSUAPKwTvbbTTz2nEFNDPpkuUmq19oSd5DfROpsuB2/oEjtYAWMTKv0drCknG3gkqxkj8lmIkJtjvc/a4bzWOpIr8rhUtJFXYr9X/Yh3UBmL1W4JObdNtvoLxfb2lq6MLj5gdaEnwvQUvOyB0YsdUoUhb69j3x6N6yjkb6hhhSvPNtcmpOf7OEhh5eTKBi1tPT7AT4gxfHV05BJbtFOlqe9wyxzpZi/7TbG9uSQL3dMd1rULQAxdPC4w2DEO9LwE1q8nsnUQX0SN6mN3Cyf8c+PLOsOpYGDY/g3mLhTzE4kZb/dg1WkzR/O3eNYvnW4VY1TZLiEtMSyQoqDMs8GCYn3YP9z3DRjkqvBbfKDP4QfqRtvq/ZNzg8I5SLc8vpAiOkVd0xF/h1xMdtVD30pF17kMr3TbxzrYHaBtQlKsoIGD4Tk5Q6WmtEsrdtQJ1LsuuPPFkP+EZTBM07cY+c1G7Y7Mz7SFrBnTmPl5/6Ul9RJoDtE+1AyOYJ8LZZ+eyERhGKA/D0BQxlG/jv88mHDsp4u7ze+6RDcHsYhv7czQ4Qb9wNDC06Iw7njQdGmnROw6VoUaYFdWw7XKnkhMxE0A4r3OUzmspYyT9OAqYF7sjOZt6t3uuxBQQxdbR3GaCg4nJb1Ny7hTd6dkRT8bN54u5MWltZ1Sz9+LDUfutBfCGuRbVZlz3PCpZVaY/iX1Y+pJeSZ+tMUsnfnYYZcvn5SYKeS9YX6B36Fy6bGvG93kkgSe1BnHlMeoI3sZy+e9AFgdSUBiXvuOTxk/vqFQWL0b6uN00m5mhCbffEB3Fqj7JjZ8RQjdlFPSkBjT5ffb8sldXJB3s6CkkPHfCwq76I82FbuGU6b6YOoA+mH6cWekNpxehuVfzB2HW4aPzOH5iaSrgqyNkwLv6rG5/Mfb1e8l6cnXuu+6nwAZ8IRHcYO3LNk97Ulci84MDwRntq/DjnhCvgbAK6WUZE/hbSyiyf9SNI2+jzEbHq1N8rr7GEz8p2wx9Zt5yjBXQUX+BJY/f0eJ0/m5LrxCAtQvqwLfOlXAP1bj4zA4m4Gdav2oDqk8uDV95NzTCCdAaDJVY79QoQo5QiYAHMNTGOQi7ynjqd6M5N6/YQ2U8nndc7jdjaiKwhwnnXZPi8PuaR9pcUz6NS8Jf5rwQSLQB5HtvOFrZPe0UAtzd+FIqUuCJBwXiJ4YnC0ZjIcYrPa5IoW8WCp23FgT8d5e6zGyxRqwvVOnGbLuTkTLRCDcTBOjPuAzWVfyKhIKZFUFdzfZAVcyKkBkb8c+qfcmJAxrc4J3ncn3lvJCsqUsrtbxpN+SqmeZy+uNdeCIbz7n1Mw3Tn0+iIbZb2FMhL2/fIsv4svDxQYfq3qfmQkzYI0Gx0nWNf27ODN8Tu1eiEVnM4AUHhk/VyC/nbeA9tIkrZSRF8N3mL2i9fTBT2aTUpnwLupHD6D/gK68QoWb+ZyUKGWjvlMGw8drx8qOoc+mRZNlBOyAiEkqhK8cEt1SZT1IXiATkS7BEWLdubfpnCeQaqxdUuybzczcW7fcfQnzbfB2o3ZkVrsQ5KKn2CN1PtRqdsimOyhxxtmwJm1y7ddsjYSHd+M/C5ylHGmhy+MWmLg64ooZGoBj0yJ2DG2hgs0Ik6sLN8E9ci+Dqq3VKHP6m4jUivIaScKQwOxDxqz7m3KXhMnb7vKHvMqkGXM9w7hm2nlb7U0LK6aGfzfh30i7mUSTKljKLmvFJVA3YJQEv8L7nd0qEPVHaUqqsD0c2NpKMqQxqXRmXANvRdJYS/sYWyWx9wqoxAqhvRKp4RyMlbHHqgGY8Td93G6YypKt8uJDm61lxrxlFpOQw+XZdQhn5wbmt3CSlbV9jwQSvAEWl/KjXQVl4zEOZVE79YhHwx4PnHqlMZ49YVJ3JW5M5Lb8D2nSMZbK4Rq8pgFR92wrmWrIuUlqDCLIH2CXrmcSDA0QjFE9upuTi7A6gxr5N08CfEllKSp63bbVJuYdpF9EBgDDmBiTPt8KzUcfTcsl5Qs4D6+hYE1cWnIOFIVjgn5IPQ4o7nU4EBsjbtI4bqo87ckEZY00/SKjgvd0hfl32WE3KMvsMyu4ATB+jd5CkWVwCy7GJkWxSvdQEuOzSTJaHCvRV4e7Lee3WGn92U6d3dDp4yWVyVb4hI5UfQ3VBJSXsxTQ6rZZsiKnFfqKCy1tnGWmVqpvyPCuW9mcTgKTbmMKJB2jb6HIlKJwTLxPeF5RRpzw8Zdx5Fyk0LD9oY2u7fE+tO9CgS2sP4tfbXEuxO1w5e7NflypPtM8XIeebQaMT2AnTXpRRaXRxQ26kvPgNmBHNtBfHtGSqgTOWD1zgXvFc2qJtXjRdWAkoCCjw4qiMJYzPl8J5hvJ33wgMx6b2zfMjQP3u0qWivbThcK5CqjzcUO2BeUfQPJiqqHQPCxQ1aUiyy8G8S7bTJHE0FgZfMqrb1r+sDYeuazy/5fbw8S9nWGRfFTKl4RP9H2hEGRvlpeYQ2hmm5HnTjjgmn/SYOt/rr1KFZNtYGAGrXt3dcUm1LLPJnPYbeVbX5IxH2g9kxVDVdAB3LYqFWmrTUiRXFivpg7aVM3GeIBrWFTehWJ4yRg1bhHSbQHSO5Ks8PQKjoGexPRNWEWFyK9wxGk97n3sEdfp5bnmFfKPP/3zD4IaVjZPNmWn3hRUBKWX9K1D8mrUOSTxV+8hj0uE7dA0izYcCR8fgLcZLXCW3tFLbfuL2oMlH1Ks9u/uDzCSAd8OHwNfcVwgSufIz2GPrU4p5eL9X/M8dtSgl2p8+fU1klgpAbBHmyGLMVC5FsbbaPhIyx/t9xnkp+pFkOtqeqEZmbh9SQ7gEQmNC0cdg4ULLlYq/QLpoRdHzaN1WyJGEfuMLpeqePuJB2s+w==

eeuklXLt2TbNNlUqFiIgIODk5wdXVFf/617/w5CMsnmzez8/Px+zZs+Hj4wOFQoFGjRrhiy++wLVr1zQPd3
F2doZMJsPo0aMBPHpkcXR0NOrXrw9bW1v4+/vj22+/1drPd999hyZNmsDW1hbdunXTirMiVCoVxo0bp
9ln06ZN8eGHH+otu2jRIri5uUGpVGLixIkoKCjQLDMkdiKqeqzpk6TY2tri3r17mteHDx+GUqnEoUOHAAC
FhYUIDQ1FUFAQTp48CSsrK7zzzjvo1asXLl26BLlcjuXLl2PDhg1Yt24dmjdvjuXLl2PHjh144YUXStzvqFGjEB
cXh1WrVsHf3x9JSUm4e/cufHx8sG3bNgwaNAgJCQlQKpWwtbUFAERHR+Orr77C2rVr0bhxY5w4cQIjR4
6Em5sbunbtipSUFAwcOBCTJ0/GhAkTcO7cObz11ltGvT9qtRp169bFt9Rp169bF1q1b4erqip9++gkTJkyAl5cXhgwZo
vW+2djY4NixY7h27RrGjBkDV1dXvPvuuwbFTkRmIoieUuHh4eKll14SgihVVqFoUOHhEKhEDNnztQs9/
DwEPn5+Zp1Nm3aJJo2bSrUarvmxXn5+vrC1tRUHDx4UQgjh5eUl3nvvPc3ywsJCUbduduXc2+hBCia9euYtq
0aUIIIRISEgQAcejQlb1xHj16VAAQDx480MzLy8sTdnZ24qefftIqO27cOBEWFiaEEGLu3LmiRYsWWstnz5
6ts60n+fr6ihUrVpS4/EmTJ08WgwYN0rwODw8XLi4uicnRzNvzZo1wt7eXqhUKoNi13fMRFT5WNOnp9
revXthb2+PwsJCqNVqDB8+HFFRURUzrlrVu31rqO/8svvyAxMREODg5a28nLy8O1a28nLy8PVq1eRkZGB1NRUBAYGap
ZZWVmhffv2Ok38+xeLj42FpaVmuGm5iYiJyc3PRo0cPrfkCACAgAAAFy5ckUrDgAICgoyeB8lWb16Ndat
W4fk5GQ8fPgQBQUFaNu2rVYZf39/2NnZae03OzsbKSkpyM7OLjPXInP7Ozsb2Ea+qmN7VmXzcKePnrNCP3o+IrcA9G0H8PqBN4/pBhVS3LdZ6xzZ8FYP5/3snP5Qq6HRxp6KjqAIh0sBQ2FHCSIHaCgXJM1xfTcE0B36ELCv5bvfPnx5lZwu+vF7v0bRbOT4kKVEe8OU3B7HKkjLn0aT3bEqg6owt2V2VZJp0rFtYRsjVjrcE+jUfH+gxYVLXCtGu8KuG9Le8O60qgPDq+Pq5Z2Ltr0X0k73Y2Dd9vXoMvPoFVsDKQ9APW6M0zPrpPsbfjA40qPTAnP/3fzH6YKyBHQnqkCX9cZg8ZXKkQvR8qg7FvYKmh/ZkebBkvvd/x2Bh43vXL/LaQ8zNyF1NNgSmv2Nrf0rzXnHW6kxepddZVXtLsWOYJT07tamy0f6Yh0C4b6TWuG2RmHQbvGbR3QzNbdmOzCd9TXK4NoGyBrxe+/bPk04YmpXPsNP6nBVp5jtnvHmdsC8jvPvKczeLCD0c3WkwrIhFVx4TvvVxAJgw0lG2j4U6D4uFPtAVGJc8lYJAU6a+g8bbtfSHlOCtCl3M9WiIfXDmyVZjiDkJk3v6kYd6Mf6vdmbjvfwB5L9Xh4p9Ya0PbjRVHYbhhlB85jJ18e5sjAPVZjW8rWD7PPLUZk8HUR3m9Qrwz2U3tk3eYs6nqcLKYDvwSbi8vRIdxEWt70Vfs3xjL0y3rjCHcxxXVK3ohd5OWsf3XCQsfEJIGHHH9q8ofAVDLHrXpMpQm8wmSwiaN2
+u6ZRY7PTp02UfZCl+/PFHPPfcc3jjjTc0865evapT7pdffsHDhw81P2e3t7+Pj4wMXFpczYicg82H
uf6DEjRoxA7dq18dJLL+HkyZNISkrCsWPH8Oabb+L69esAgGnTpmHJkiXYuXMn/vjjD7zxxhul3mPv5+eH
8PBwjB07Fjt37tRs85vvgEA+Pr6QiaTYe/evbhz5w6ys7Ph4OCAmTNnYsaMGdi4cGk5cSOuXr2KCxcu4KOPPs
LGjRsBABMnTsSff/6JWbNmISEhAbGxsdiwYYYNBx3njjg3g3Ex8drTQ8ePEDjxo1x7tw5HDx4EP/3f/+HBQsW
4OzZszrrFxQUYNy4cfj999/x3xffITIyElOmTIGFhYVBsROmZi7UwFRZXXm8I195lqempopRo0aJ2rVrC4VC
lRo0aCDGjx8vMjIyhBCPOu5NmzZNKJVK4eTkJCIiIsSoUaNK7MgBAPHz4UM2bMEF5eXkIul4tGjRqJde
vWaZYvXrxYeHp6CplMJsLDw4UQjzofrly5UjRt2lRYW1sLNzc3EoaKo4fP65Zb8+ePaJRo0ZCoVCI559/Xq
xbt86gjnwAdKZNmzaJvLw8Mx0aOHo6CicnJzEpEmTxJw5c4S/v7/v7/O+7Zw4ULh6uoq7O3txfjx40VeXp6
mTFmxsyMfkXnIhCih9xERERE9Vdi8T0REREREJBFM+kREREREJBFM+kREREREJBFM+kRERBLBpE9ERCQRTPpEREQkEUz6REREE
GkT0REJBFM+kRERBLBpE9ERCQRTPpEREQSwaRPREQkEf8P57FQ//8OGr8AAAAASUVORK5CYII=",

```
     "text/plain": [
      "<Figure size 640x480 with 2 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "import numpy as np\n",
   "import matplotlib.pyplot as plt\n",
   "from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay\n",
   "\n",
    "# Assuming you have the y_pred_classes and y_true_classes arrays with predicted and true labels
respectively.\n",
    "\n",
    "# Calculate the confusion matrix\n",
    "cm = confusion_matrix(y_true_classes, y_pred_classes)\n",
    "\n",
    "# Display the confusion matrix using ConfusionMatrixDisplay\n",
    "classes = np.arange(cm.shape[0])  # Assuming your classes are integers from 0 to n_classes-1\n",
    "disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=classes)\n",
    "disp.plot(cmap='viridis', values_format='d')\n",
    "\n",
    "plt.title('DISPLAY CONFUSION MATRIX OF LSTM ARCHITECTURE\\n\\n')\n",
    "plt.xlabel('Predicted Label')\n",
    "plt.ylabel('True Label')\n",
    "plt.show()\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 30,
   "id": "f9aeded0",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAcgAAAIBCAYAAADNri8LAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG9
0bGliIHZlcnNpb24zLjcuMSwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy/bCgiHAAAACXBIWXMAAA9hAAA
PYQGoP6dpAABCX0lEQVR4nO3deZxN9ePH8fd1zdIYM2YwxjKMZMs6X8tY0rSMVIhKDSpblFKpadK
9QVJR8fbOEGkZCEumbiaKkIqLsWWMMxQyDWT+/P/zmfrvuZ7jXlvq+no/HeTzmfs7nc87nnHPnvu9Zr8
MYYwQAANwU+7M7AADA5YiABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwOKSBKTD4fBqWWLZsmXbu3CmHw6
EFBgbK4XBo48aNRdbLz8/8/XO++uu07h4eEKCAhQdHS0evbsqe+//96j/Xggw/qyiuvVGBgoEJC
QtSyZUUuNHTtWJ06ccFtPjzzyiHWeH3zwgWsdFbXcAQEBqlGjhgYPHqyTJ08W2f+7775bDodDodAwYMOO
```

P6WLt2re69915FRUUpICBA4eHhio+P1zvvvKP8/Hz17dtX/v7+2rBhg0fbvLw81a9fX9HR0crKyipyHjk5OR
o7dqxiYmIUEhKiUqVKqU6dOnrggQe0adMmj/rerktJys3N1RtvvKEmTZqoZMmSCg4OVpMmTfTGG28o
NzfXY9rR0dFu67NEiRJq2rSppk+f7lF32bJlZ3wvp6SknHHdFvr444918803q3Tp0goMDFSNGjX01FNP6bff
fvOoe6b3+eLFi884n+joaLVr1+6s/VmwYIHi4uIUERGhoKAgXXnllbr77rtd07/uuuu8+l9+4YUXXPN1OByKj
4+3zm/ixImuNrb/naIsWrRIDodDFSpUUEFBQZHL7O32lKTMzEwNHTpUDRo0UHBwsK644grVrVtXAwY
M0L59+1z1evTooeDg4CL7dvr/8emfaaf3q6hh6tSprukVNfTt29dj/suWLdMdd9yhyMhI+fv7KyIiQu3bt9f
cuXXMl+b4NL/bnkq/L90dTp061vndWrFihW265RRUrVlRgYKAqV66s9u3ba8aMGWec3umK+1T7HL377r
tur6dPn67PPPvvMo7x27doeH3LeCggI0KRJkzzU6n19OYPPXu2HA6HIiMjlZycrGHDhnnUOXHihO644w4
tXrxY1157rQYOHKjw8HDt3LlT77//vqUqVKkqSFCxfqrrvuUkBAgLp166a6a6desqJydHK1as0N
NPP62ffvpJb7/99jkt8+nLnZGRofnz5+uf//kt//yntm/nfrU/VmZmZWrBggaKjozVz5kyNHDlSDofDo96kSZP
Ut29flStXTvfdd5+qV6V6+uo0ePKjU1Vffff7/279+vkSNHav78+erbt6+WL1/uNp3XX39d69WiRIki
+3/nnXfqk08+UZcuXdSnTx/l5uZq06ZN+vjjj9WiRQvVqlXLVdeXdZmVlaW2bdvvqiy++ULt27dSjRw8VK1Z
MixcvVv/+/TV37lxr3xo2bKgnn3xSkrR//35NmjRJ3Bt3V3Z2tvr06ePR/8cee0xNmjTxKG/evHmRy1zoqae
e0quuvvqoGDRpowIABCg8P15o1azRu3DilpKQoNTVVNWvWdGtT1Pu8QYMGDZ53f2YwePVpPP/204uLilJ
SUpKCgIG3btk1Llix+/d2tfnuu+/0xhtvaODAgapdu7arvH79+q6/AwMDtXNApdu/+uvXn4+/q6/AwMDtXTpUqWlp
SkyMtJtnsnJyQoMDDzjFzqb5ORkRUdHa+fOnfr888+LDGBvt+cvv/yi+Ph47d69W3fddZZceeOAB+fv768cf
f9TkyZM1b948bdmyxac+FmXMmDE6duyY6/WiRYs0c+ZMvf766ypoyrvEWLFq6/W7durW7dunlMq
0aNGm6vhwwZohdffFHVq1fXgw8+qCpVvqui3337TokWLdOeddyo5OfmctqGvfP1c8nb5vDF79mwlJCSo
YcOG6t+/v8LCwrRjxw59+eWXmjhxorp27er9xMyMyoF+/fqaoWe/YscNIMqNGjbKOHzJkiJFkDh486Crr3r2
7KVGixHn369prrzV33HGHeeKJ0zVqlXP2PfXX3/dY1xeWqU2bNnjzHGmF9++cUEBwebWrVqmX
379nnU37p1qxkzZozrtSTTr18/63Ycz55tJJmlS5e6ymzLXVBQYJo1a1a2YcDodJS0vzzmM6UKVOMn5+f+fzzz
40ks2zZMo86K1euNE6n01xzzTUmMzPTY/x3331n7GA==+fd3Kq1SpYtq2betWlp
++27oshb799lsjybz00kse4/Ly8syhQ4dcr31dlw888ICRZN58802PuuPGjTOSTN++fd3Kq1SpYtq2betWlp
6eboKDg03t2rXdypcuXWokmdmzZ59xGYsyY8YMI8kkJCSYvLw8t3GrVq0yQUFBpl69eiY3N9dVfj7vc9uy
/VFubq4JCQkxrVu3to4/cOCAtdz2vjx9vjfeeKMJCQlx2z7GGLNnzx5TrFgxc+eddxpJ5rvvvvvNqWY4dO2ZKl
Chh3njjDRMTE2N69OhR5Ly92Z65ubmmQYMGJigoyCxfvtxjOhkZGWbgwIGu12fbDqf/H5/tM23UqFFG
ktmxY4dX0ytK4bbo1KmTycnJ8Ri/ePFis2DBgiLbFbUNL/bnkrfLZ/POO+94vHeuvvpqU6dOHZOdne1Rv6j
3cVE4B/n/du/ereXLl6tz587q3LmzduzYoa+//tqtzt69e/Xvf/9brVu31uOPP+4xDafTqeesq19/jKK6/o2L
FjmjxSssqXL+9R/6qrrlL//v0v6HI4HA5dc801Msbol19+8RifnJys1q1b6/rrr1ft2rWt3+aGDh0qh8Oh5ORkl
SxZ0mN848aN1aNHD0mnDtfeeuteevbZZ5Weni5JevTRR+Xn56exY8eesa/bt2+XJLVs2dJjnnPpSpV
2vfVmXe/fu1eTJk3XDDTdYw3169dP119/vSZNmqS9e/eesY9ly5Z5ZVrVq1XH29UIYOHaqwsDC9/fbbHkc
5mjZtqgEDBmj9+vX64IMPPLuh8i3Lo0CFlZmZat4UkRURnPO0AwMDdccdd3gc3po5c6c6bCwsLUpk0bn6Y
3b948nThxQnfddZc6d6+6suXPner0Hatuec+bM0bp16zRo0CBdc801Hm1CQkL00ksv+dTHP8Pzzz+v8PB
wTZkyRX5+fh7j27Rp49Vh9ovhbJ9LF9L27dvdvPEkT+fv7e4zz9X182Qbk8ePHdejQIY/h+PHjRbax1c/MzP
RqfjNnzlSJEiXUrl07NW3aVNWqVfMIj08++UR36Ij08++UR5eXm67777vJrmggULdOWVV7odKrkUdu7cKUktk9ksv+dTHP8Pzzz+v8PB
+3b5+WLl2qLl26SJK6dOmiDz74QDk5Oa46x48fV2pqqq699lpVrlzZq/mNHz9eOTk5euKJJzR//nx99NFH
GjlypMfhtNNVqVJF0qnQzsvLO2NdX9blJ598ovz8/fOshm0LdunVTXl7eWc/d5eXlae/evR7rstDRo0et7ztz
hl+R27p1qzZv3qwOHTooJCSkyP5Jp85Rnu70eWVkZJxxxGbwRERGhK664QgsWLNDvv/9+3tM7Xc/deuXfXt
t9+6BdOMGTPUqVMn64f5mSQnJ+v6669XZGSkOnfurKNHj2rBggvetbVtz48++kiSvP6/LmTb7n+8NuJC
OnnypHVehf+7W7du1aZNm2rLQVGfS9Lzl88XVapUUWpq6lm//Hrjsg3DqFGjzH
WzsrKs9e+++26v5pecnKmwOHTroiiuukCQlJCTo/fffd/vgXLrxwDzPneqp6vwjfT9u3b9
eqrr2rOnDmqW7eux/mrmTNKiAgQB06dJAkde7cWYPH9aiRYtcbdZt26bc3Fyf+l2lShUhNHTpUM2bM
UK9evdSyZUs98MADZ23XrFkzFkzcXFaeLEiapUqZ6K6du2q8ePHa/fu3W71fJ2Fha/fu3W71fF2XP//8s6Qzn5crHHf6tXUfYU
mu9blhwwb16tVLaWlp6tSpk3U6vXr1sr7vDhw4cEh4cF79i46OVkhIIEf/bO/zwu15PooVK6ann35aq1evuXK
lXXrrbdq+PDhWrNmzXlPW5JuuOEGRUZGaubMmZJQrfe1a9f6f6f6dj5IUnp6upYsWaLOnTtTLkipXrqzmzZtbj
4RI3m3PjRs3KjQ0VFFRUV73o6jPm7Jly/q0qxjdsJuMSSdffl8MWDAAO3Zs0fVqlXT
DTfcoMGDB2vFihX+s1JpfkIp1z8cADD+iuu+7yKJ8+fbrHxT3SqcM4tm+RfzzpXZQff/xR69ev14gRI1xlX
bp00fDhw/Xpp5+qbdu2kuTaXfVX6p5+qbdu2kuTaG/XmG5ovdc9H4T/qH11zzTWaNm2ax8U3ycnJatu2ratP1atjZ2ScnJtu2ratP1atXV6NGjZScn

KyOHTueV78ff/xxTZ8+XRs2bNC///1v64U/p3M4HPr00081evRovffee5o5c6Zmzpypfv366e6779a///1vl
SpVyuc+HT169Kz1C8edfoThP//5j8f67NmzZ5FfzAYPHqxWrVp5lIeHh59X/wrHn94/2/u8qL1bXw0dOlS1
atXS+PHj9emnn+qTTz7RoEGDFBMTo+TkZLcLOHzldDp19913a+bMmXruueeUnJysqKgotWrVyqdDbikp
KSpWrJjuvPNOV1mXLl305JNP6vDhwx7rwpvtmZmZ6fP7vajPG+nUBScXWocOHaynCwoD8VJ93njLl88l
6ezL54tevXqpYsWKeu2117R06Vl1XbpU//znP3XllVfq3Xff9emI3mUbkNWrV7dembZixQpQprfafTWeSVbG
fz3nvvqUSJErryiu1bds2Saf+AaKjo12hIsl1OKzwA+5MfKnri9PfXH/8R927d69eeeUVpaenu/aEC23cuFE/
/PCDunXr5lpG6dQl32+99ZYYyMzZMVEhJyzv12Op2KiYnR9u3bVadOHa/bBQQEaNCQRo0aJD279+vL774
QmPHjtX7778vPz8/33nNX7CBaxpSCf+gX1933nT
v8Lxp58zOZ/3uTe6dOmiLl26KDMzU6tWrdLUqVM1Y8YMtW/fXhs2bPDptqnTde3aVW+88YbWrVunG
TNmqHPnzl59kfqj9957T02bNtVvv/3muhUmZi17tseRC2+2ZdHIiM/nxS72djhdpUqVzji/V54
61z/VwqdLbl81WbNm3Upk0bHT9+XKtXr9asWbM0YcIEtWvXXTps2bfL6XORlG5CXijFGM2fOVFZWlq6+
+mqP8enp6Tp27JiCg4NdtxysX79eDRs9gkUJCAgo8jaXwnOvp39Anf6P2qZNG9Wq
VUsPPvig69yKdOqDRZKeeOIJPfEHHEx7TnzNnjnr27KmrrrpKxYsX1/r1673u94VSvnx5de7cWXfeeafq1Kmj
999/X1OnTvV5XRbu6r1673u94VSvnx5de7cWXfeeafq1Kmj
999/X1OnTvV5XRbu6fz4449Fbqcff/xRkjy2eZkZVzrs3BdtmvXTmPHjlViYuI5SLlnR/SvKrl27lJmZaX1PXgo
hISFq3bq1WrduLT8/P0bNk2rVq1SXFzcOU8zNjZW1apV0+OPP64ap6PJCg4NdtxysX79eDRs9gkUJCAgo8jaXwnOvp39Anf6P2qZNG9Wq
ZI+A9GZ71qpVSz/88IP27Nnj02HWy8kfP5sutIv5uXxSxBQUFqVWrVmrVqpXKlCmjoUOH6pNPPlH37t29a
n/ZnoO8VL744gvt3btXL774ombPnu02vP322zpZv66VK1d6Vb9
KlSravHmzdVxheeGFLUUpX768nnjCS1YsEDffPONambMmKHrr7/eYxlnz56t+vXru87hBAUF6YYbb
tCXX36pPXv2eNXvC83Pz0/169d3nT+SfFuXdVdii+0PTp01W8eHHdfPPNZ5xW27ZtFRcXp+HDh5/xgQ
e+qFGjhmrUqKEPP/ywyG/8hTez/1lXHf5R48aNJZ26j/B8denSRVVPt2rXP+iXzdMnJyfLz81NKSorHe
7h///5avny5x/nr09m2Z/v27SXJ6//ry1GNGjVUs2ZZNzZ8/3+0eywvhYn0uXWrn9D4+p5tPztpOldB/k/fffb0
qUKGFOnDhhHV+9enVz8803u1737vDXSDJvvPGGR938932Q/HwzevRo132Q27ZtMyVKDBXX3219Z7Ebdu2
ud0b9vjjjxun02m+//57t3qHDx82VapUMQ0bnNnQrL2q5Dx06ZIKCgkyHDh2MMcyXX77csDLTp0+3LuNL
L71kihUrZn799VdjjDFFffWVcTqdJi4uzhw9etSj/vfft1+mTp3qUe7rtiyZYZYvZtWuXR/nhw4dNQoVTFhY
mOseQV/XZ4e/vY0kM378el+6/rXv4wk8+CDD7qVF3Wv4KJFizzufT3F3f+89I8l07drV4z7I77//////3pQo
UcLUrVvX7X62i3kfZFWlvn666+t42699VYjyaxZs8ZjnDfbf3Qf5xvjt37jRDhgwxcixYtcpXZ7mWzueqq8wN
N9xgHbd3717jcDjMyJEji5x3odO3Z05OjqlXr54pUaKE2wNtvv936NJfs7GzNmTNHrVu3LvLcym233
6XCrk7fLZ2N47S5SsdZ96KGHjCQzd+5cr6f6/tznEmpeXV+Q3wNtvv936NJfs7GzNmTNHrVu3LvLcym233
aaxY8cqPT1dEREzXVV7V9+3Y99thjmjt3rtq1a6ewsDDt3r1bs2fP1qZNm1xX2FWrVk0zZsxQQkKCate
u7fb0l6+//lqzZ8923U8oSc8++6e6xmz56ta6+9Vg8++KBq1aqlffv2aerUqdq/f7/eeecdr9ZF6dKl1bNnT40fP
14bN25UcnKynE6n61yqbRkHDRkAJQUJSYmqWLFnrrrbf08MMPq1atWm5P0lm2bJk++ugJ61OGfLVu
3Tp17dpVt9xyi1q1aqxw8HD9+uuvmjmZtmvbt26cxY8a47hH0dV2+/rr2rRpkx5++++GEtXrzYtaf46aefav78
+YqLi9Orr77qVT9vueUW1a1bV6+99pr69evndnvWC8uXLrffg1a9f/4xPIrnnnnv03Xffezfaysfr55591zz33KC
wsTGvWrNGUKVNUunRpffDBBz7fAnEm27Zts263mJgYxcbGqkWLWFmrWrJluvvlmRUVF6ciRI/rwww+1f
PlydezYUTExMefdhypVqqrgeYeaLVatWadu2bUU+8qxixYr6/xz/+oeTk5LM+QtG2PefOnav4+Hhde+21uvv
uu9WyZUv5+fnpp59+0Owow4WZcsW6dbuXLlXBcFJSQkaP369Xpp+nXPF0pIl9rD3mfOnX9/rz/oa/ZWvlxXngaDqyDQS
xevFipqak+P2pNunifS3+86Mub5fNWhw4dVlVqVbVv317TVlaWlixZOgULFqhJkyauIwRMFgaHZy6auIwZeOafYPk8X
Yw9SUpFDUd/M5syZYySZyZMnF9nXZcuWGUlm7NixrrK8vDwzadIk06pVKxkMaGmr8/PxMl5pVTM+ePc
0PP/zgMY0tW7aYPn36mOjoaOPv729KlixpWrZsad58801z8uRJt7p7p79+41vXv3NhUrVjTFixcc3eHHpl27d
uabb77xmO6Zvslu377dOJ1OO7VrV1O6dGnTqlWrIpfRGGOqVq1qYmJi3MpWr15tunbtaipUqGD8/PxM
WFiYufHGG820adNMfn6+T/2xxdNMfn6+T/2+xODgbBk5cqSJi4sZ5cuXN8WLFzdhYWHmhhtuMB988IG1jS/rMjs727z+z+
+uumUaNGpkSJEiYoKMj84x//MGPGjjLE+aeRMe1lTp041klxPECrcgyxqGDJkiFfr4MMPPzStW2c2YWFhJi
AgwFx11VXmySefdHt/FzrfPcii+nr//feb3NxcM3HiRNOxY0Y0dTpUoVExAQYICgkxMTIwZNWqU9akkxvi+
B2njzR7ko48+aiSZ7du3F1nnMXmJLMunXrzjrv07enMaf2iAYPHmzq1atngoKCgoKClbt65JSkoy+/fv
d9X7M/Ygixri4uIh8kqemeppoOHTqYIlgIU7x4cVO2bFnTvn17M3/+Tvn17M3/+fOvQz7YNjbnwn0vdu3c/5+X7I9t7Z+b
MmaZz586mWrVq5oorrjCBgYHm6quvNoMGDbI+GexMP/fQQAA8Af/8xfpAABgQ0ACAGBBQAIAYEF
AAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQ
AIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUAC

AGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgB
gQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYE
FAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBB
QAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBBQAIAYEFAAgBgQUACAGBR3NcGX375pUaNGqX
Vq1dr//79mjdvnjp27HjGNsuWLVNiYqJ++uknRUVF6bnnnlOPHj28nmdBQYH27dunkiVLyuFw+NplAMD
fgDFGR48eVYUKFVSs2MXfv/M5ILOystSgQQP16tVLd9xxx1nr79ixQ23btlXfvn2VnJys1NRU9e7dW+XLl1
ebNm28mue+ffsUFRXla1cBAH9De/bsUaVKlS76fRtGHPOjR2Os+5BDhgwQAsXLtSGDRtcZZ07d9aRI0e
0ePFir+aTkZGhUqVKac+ePQoJCTnX7gIA/sIyMzMVFRWlI0eOKDQ09KLPz+c9SF+tXLlS8fHxbmVt2rTR44
8/XmSb7OxsZWdnu14fPXpUkhQSEkJAAsD/uEt1qu2iH8RNS0tTuXLl3MrKlSunzMxMnThxwtpmxmxIgRCg
0NdQ0cXgUAXGqX5VWsSUlJysjIcA179uz5s7sEAPgfc9EPsUZGRurAgQNuZQcOHFBlSIiuuOIKa5uAgAAF
BARc7K4BAFCkix6QzZs316JFi9zKPvvsMzVv3vxizxr42zPGKC8vT/n5+X92V4A0qnjx4pfFLX0+B+Sx
8e0bds21+sdO3Zo7dq1Cg8PV+XKlZWUlKRff/1V06dPlyT17dtX48aN0zPPPKNevXrp888/1/vv6+FCxde
uKUA/gfl5ORo//79On78+CgoKUvny5eXv7/+n9sPngPz+++pPz+91/fXXu14nJiZKkrp3767JkyfrzvUUK
sRjuEtu76akPKAgKnuWsvzLJzLWHMDhnlNqYhwOTeOgEzlYPMWyHSoJtmnNHvMdwaOzbEZPYxycuqVpaGA
3Tu0ap+MWw0+S5kD8gyW+kZoVSXQK7tWbVKWrvVkmnO6G+dVpv8N1VcmBcnD6Bz60zNCSjhzgpayTfcfmtu
C9xTLeuqV0BdZe7N7Qj99hKJg2Tpj+hBHBzV7KrgDu0DUTdHhD3+SivMpUBnRxOyHmnAeOH+ZD6wrDb
w0Ad5D1eMrsUSrhuWqeiIaKrvHHN2n7Qq7lRXzS0M3M3gzdQtV7Es6OvFKpjOpfp2ilHt4KXtvfKTcpnJc
y6a1fvVD8+mx4jKv5RfUu8qEeVFaFMpBDbCfnYa/bOSGKUxdtxDzy2yMN3LodX6QPqIxNOs4fJ3Lh2EBK
7n07R1TodxFHtnDVWSU/MazeZbtxuehuQ73rbvmfH4c7lNWP7VRVXMCzsTZbUjEpXXdPrThrFtnhTFbTzuFV
OwcXeyt7xPqH5JvOVxYjNC9dtROH0Oc7Lx1SY7PPQu+qI4GtM5mUQ8pVXhsvF8d0wgBS5aWOzc3uBTzqV
1xSMPfdvsrEqmq4pQPxeKEg9rVF8TNLqzbB0hS+9dW9KaS6S5WE1h5tD62ddzZ7GvbVFOmx5JJrVW2pxMrg9
eDTuN5w0rmzl3a1YhOhpcBvWPA55nONG5SbEjwC2W2iTyzYgYDhHWWT3hi9YIE+77zfZhPSLG+DmwpVr
1W02sN85/vy5Q2F37BdK05e7VeRdjdFXW5/3FTWdyBHc63r84J5wD2zXnffiSB2SGJIT3/u+c9zdrt9c9J7+X
2BZ/0B+aQxP6Y8T9d2Y5v6Ka0dTRV1FU3Q+P8Jbl6p+yHW+ohTkL+5ht15J5Lx1QzZrCqU71uN6P4S0OzWd5
VPIu4w6zxUHiSbBVtU+mDN5fUb1U3dTU03XtJHhSHnWeHyVj+NHjQ6V3wFB0cSb1Qi4bCM5iYb1j6S8Tkq1VCEJT
AuuZ3tN5Cmp4DLvMy4YwNJr6ggpQW6aZvgU5fnVt4wZX6QcZBvjv0LxvFBmGRbxh5ba57nRgdTm8l9F0J
qb9d6mD2Gp3GdFSwt+ezfyFF9K5m4MpnDJbTqq/a4Lv+RlsUtqg8DaQN4T+Q7rU03yb1RkGq2rwh8hx0c7J
xzsrJ3P0rnDw4UZlAePnJPHCk31c6Cn/xDBn3LNM59+V7Nc9uyK8mVF7UXRrTTzJk8qQpLw5n1lV5jN8C8dR
qFvbJl4PUSsCW4pqfBXm1Wv3Rt2Ne4HFt7oyiTHeGVNFBA8u/PJyS2Cd9L1+q5iu0Un9uqqRgGB0kbbtpw3l
4aKI6JYNFSf9+pNAIhqR5mMJn5v1pP3o6Y9Po2x7UtDTTEqAGj67BC6GPb2PWvx8WdyfZsY6Bc1Rl71Gyp
sATzCx6ZI6IqQyIfqGSqU3XxqJ3qbZJ2KYZmRNMmXPNP7f4Ukj9JyN/6TnXJpyHbEIBlHx5TKmt4B0zBHeSql
4M1JNzpxQ7UQ8u6M/NBMMJ3VGnOvu3fddKJqWlJXbuKGh0u7ByBmGeDwFI3Jg/LUOMDqT6Vp+0PFvTyh8m
Tm4oqD+4L9RgLn7OmbV90kBm4MhzN+3zJYdRj/q2WQdP9GxDxYrOXnZZvkZ/GQgk4S3OEqbn8Fxl6XG1Qe3F
FC3bXcWX0/KFYo/LHEwQjJ3KJNOo6ZEFkSrbHP7/RBwLbK9cYoLrnQiczE7wHvw+Cx+2hU7W+M3Lrnck3U
nUS5tdy3/xVYv9pYXl6L4QgW+dJ1DkdB0zKPXVpSeC7IGpbnpXRW6EmMDe/zZ8QDW4QLk0efj+8DL/Y
fTUhhDd4k+/5PwY6nmzVp1mq9XV2fqs0DX+ymgqwBkZRZDoYfmDq+RJl3mXYxGczYvvm5YhL0TzXTLjvb
2Y0m4Tkqa+m6UVbtk+AwnVtsnE7+xKp3jcdrODlQ+e+aYxnpuAWXhrD9cqJHqXKjjBP8YN/fRwfGGbVwEJz
pVmVHrdh3JpH1FG9eMmSJjOqbjg5RWJDJhY0DGbWRmyNgOZFsaTNcSqe2Rd1WGaZ8u3nJ1hJTBW7qdRW
UYA3VNi5rYxD2zJ3/bK5MTOqe3BQAQAyIw3kCh3WVEi7RvGXfNz6jJ7XxRMkNfB0VEu4e8bYXkAv3Jy3TWR
bhDxWfMxpMY9GHvFoRPyXG91W8qb9lk5Mf2cGwVe4O8+SEmH0tGSRj5mfYqIVOuEeNH4HG/Jj/Yq7VXyl
QuYPyE3mXY2Dq8p6sYoD7vmjmWJzJqJHaGkSDTi7Jb1fGMnR3DwjUH7XiaPPaN5+7mN1yqkZvdYMOd8V4o
qPw6FjiLqgzk+dNlZmWtENcRuGCQp9nwDpA5TvAgK6ZKYC5e1nsR+5xY0ncyzLRzmEczuKn61oHBxvfvBS
K7Q0xb+hYTb4WLKmHhj0bGoTNKQ9yAxkDQ8cZp7EGqz3D1cX0QVXGLC2vV5J3TKKEy9OwPN3nZMzRWs2L
Fim/ZkPb0gm9hqFJRwqL7k2ZUhqJb1e66F71s2bWJa3P+Cf9i7dPk7WNyFcHV0XDRu8MgdkOqTG8RtFB1m0Sw
v2rU+5nDJtQu1mTPqBgYuD0tO8bZ+tgfoXf9LZEqQXQH2jv47D+3F4X4sL6eSvVGsnLHuN+z9k9HqVv2XVnZ
K2W2KJlcBTqTlPLGjl4B7FcQ3VLl6cPj2h1UQnAR9GfXbmG6U5aXklW2j3ZYUBuVc8+sz0ZyY5y+0y0ZoJPx
Aw37eSyGtpFe2yYWgmRF3pOnUqDDKZ4f2vNGu6M8LlrMqRfjs8xt1U9h8N0tR1lDrVf0zIVCu9dWH1LWQuM7
G+bhQ9GCTbQSbFsqR3N6rzcHX2eMKcs3nzOuU/rj6nXP7vKJ5bE3eoDsAYo1T0MG2vH6AKhXEodXM4Uq0w
J4bBXT1dqdAcuTHwmM3xlJgwB7xjr7c5kZW1FrYf4pfQkKNrKmP/uzpiJqDWZ9HHM4hvbYmQ6F0lS6ZvuSb
Lkx5bNqZR8Nk2A2GMkEbV7QoI/VdP9nS9R7MWnUCZUsQ0zq3FAKGkI5tLShN8a2mBY4k7F2T9IbbmMVsMGv
oRfHcL3D8yHc1ABr5zjkVHDTlqAyR1wdXDfZoI7Q6Nc5jc2fGGWE1NAu7SOQTG6eYKjXl+XPYh6v6aYq0Wxf

cd9Xre5uTu1desaSd9r/fpUVapUVp06RUtaq/Llpe7dr9XLL0/Tyy/f5WqTmXlMs2fP0pw5L1unm5lZQ7N
nz9acOXOKXO7LCecgAVxUPXu210svTVFa2iGlpR3S8OHvqHfvDta6P/64VVlZJ5STk6u5cz/XlCkf6bnnek
mSjhw5qkWLvtLx4yeVn5+v1NRvNWHCXN155w2u9j/8sFnZ2Tk6ceKkJk6cp2XL1ujxx7u4xufk5OrkyWw
VFBjl5eXr5Mls1y0nzZvX13ff/ayvvlonY4x27dqvOXM+V0xMDUnSF1+s1sqVPyonJ1c5ObmaOnWBli79X
q1bx0qSbr/9ep08maMJE+YoPz9fq1Zt0Pz5X+q22651W8bJkz9Sjx7t5HQ63cpTUv6jbdv2qKCgQEeOHFX
//q+qRIkr9I9/1FJUVDnt2vWR1q5Ndg2SNGvWcPXp01GS9MADt2vq1I+1efNO5ebmaejQiapePPUo1alQ
+67otXNZjx46roKBAX3+9Tm+8kaI2bU796lKjRrW0b99BffjhMhUUFOjgwcN6991Fiomp6bYMM2f+R6V
Lh+qmm5pZt+/MmTNVunRpj/O6l61Lsp96ni7cbvWlOPzEwHDxhxMnqpiff/7EnDjxnTHmj8OlmP/p8zzz
kJOz0jz8cCdTqlRJU6pUSfPII3eb3NyVxpjvzIMP3mEefPAOV91Bg3qZ8PBQExQUaJo3r2dWrJjkGpee/h/T
tGkdU7JkCVOyZAlTr95VZvLk593m1aNHOxMaGmxKlLjCtG4dazZsSHEbHxf3DyPJbRgypI9r/KRJz5lataN
cHCQqVChrHn44U7mxIkVxpjvzMKFY0y9eleZ4OAgU6pUSdO0aR3zwQcvu01/1aqppnHj2iYoKNDUqFH
ZTJ8+1G38Tz/NMg6Hw2zfPs9jPY0Y0c9ER1cwQUGBpmzZMNO27TXmhx/eK3K9SvIYP3LkI6Z8+TKmVK
mS5qabmpktW+Z4tW537vzIXHNNQxMaGmxKlixhatasYoYNe8jk569y1Zk/f7SJialpQkJKmIiIcHPPPTebg
wc/c5t/kyZXm8GDexfZ5yZNmriujD2Ty+UQ6//Yw8o5xIq/h5Mnq2jHjgmqWrWM/vAsZ+Ay19irWoUP4
+dh5QAAXIYISAAALAhIAAAsCEgAACwISOAv6dRFmJf/JXaA7y6Xa0cJSOAvyM/vN0k5+v9fUgL+Vgp/Iuy
Pz6H9M/yPPUkH+HtwOrNUqtRHSk/vIqmUgoIkB3cx4bJ38oxjjTE6fvy40tPTVapUKY+HKVxqBCTwFxUZ
+Y4kKT39Nkn+4j5fXP52eFWrVKlSioyMvMh9OTsCEviLcjiMypefooiIFOXmlhEBicvfprPW8PPz+9P3HAsR
kMBfnNN5XE7n7j+7G4AX/lqPfeIiHQAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsC
EgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsC
EgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsC
EgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACzOKSDfeustRUdHK
zAwULGxsfr222/PWH/MmDGqWbOmrrjiCkVFRemJJ57QyZMnz6nDAABcCj4H5KxZs5SYmKghQ4ZozZo
1atCggdq0aaP09HRr/RkzZujZZ5/VkCFDtHHjRk2ePFmzZs3SwIEDz7vzAABcLD4H5GuvvaY+ffqoZ8+euvr
qqzVhwgQFBQVpypQp1vpff/21rZsqa5duyo6OIo33XSTunpcta9TgAA/kw+BWROTo5Wr16t+Pj4/06
gWDHFx8dr5cqV1jYtWrXYH4yy+/aNGiRbr11luLXivlQ+dOiQ8vPzZVa5cO
bfycuXKadOmTdY2XbtTNNdfIGKO8vDz17dv3jIdYR4wYoaFDh/rSNQAqiLfhXrsmXLNHz4cI0f
P15r1qzR3LlztXDhQv3zn/8ssk/3wIT7C6vWTzn/8ssk1SUpIyMjJcw549ey52NwEAcOPTHmSZMmXkdDp14MABt/IDBw4oMjLS
2ub555/Xfffdp969e6e0uS6tWrp6ysLD3wwAMaNGiQihXzvPkfAJzXP+ahRI6Wmp
rrKCgoKlJqaubNm1vbHD9+3+3CMEnU6nJMkY4+t/AQC43PgdkQkKCDh48q/AQC4JHzag5SkxMRES1be/eXY0bN1Tpk01ZswYWVl
qWfPnpKkbt26qWLFihoxYoQkqX379nrttdfUqFEjxRvr/bkcxxeC8IdEp0/wB5vTBGQ/IiS8wyaVUEvfvgQ17wK6Mqcjpt8a9P4R6Be4Lb/QmfZ2bLJhJNeMRnxJXLWnmHJZdawqRxJA8PTtQmDdLAH/Z14/3Cc0APdvk3JtPs+sJIrO78hkI7+MttKcGYjj+sFR0RVrS7x+ZxKaR0PUACoiGPs4F/5Ofvm2E1RUh8TFsDszS1VR/4wfdZ6HtTqbXWN/Gu9XKyzk0OwM
8BxzszMTIWGhiojI0MhISHnMSXBesTAMBX5xc3Fy4LvMOzWAEAsCAgAQCwICABALAgIAEAsCAgAQC
wICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQ
CwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgA
QCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAg
AQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCA
gAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsCAgAQCwICABALAgIAEAsC
AgAQCwICABALA4p4B86623FB0drcDAQMXGxurbb789Y/0jR46oX79+Kl++vAICAlSjRg0tWrTonDoMAHDBx+wxl3
MClUNzXBrNmzVJiYqmTJig2NhYjRkzRm3atmtNHmzZsVERHhUT/nJ0etW7dWREAEPvjgA1WsWFG7du1SWFjY
qVKlLkT/AQC4KBzGGONLy93E3lYRFVG7du1hqxm4QgAwOXA50OsiYmJmzhxzW48KhRo7Rp0yb6/6EyZM0KhRo7Rp0yb6+fm
dUyczMzMVGhqqjIwMhYSEKsMVGhqqjIwMhYSEnNM0TnGcR1sAwNXKK48XLgs8I5Ph1hzcnK0evVqxfH/3cCxYopPj5e
utLb56KOP1Lx5c/qXr10/lypVT3bp1NXz4cOc5sn+zsbGVmZroFDh5Sfn69y5y5cq5V05Y5cq5lZcrV05
paWnWNr/88os++OAD5efna9GiRXr++ef16quv6sUXXyzyf0aMGKHQ0FDXEBUVdZ3v22j08k+fDc8SIEQoNDXUNUVFRvnQTAIDzdtGvYi0oKFBERIQmTpwokJiYHTeuGDKj5dBu9uBpk8SFaHg9OnEUbGiCClekFB
ERITefvttNWrUSAkJCRo0aJAmTJhQ5DwGDBiggwcPuoa0tDQfewsAwPnx6SrW0qVLy+l06sCBA27lBw4cUGRkpLVN+fLl5e/vL6fT6SqrXbu21q9fr9zcXBUUFHiorBZM/HwICDiVQqVVJ1BCRSHAn+fLY6vmBzF
GR1jbly5eXn5+fnE6nq6x27dpKS0tTbm6utbX6+8Pe2+cKggbYH4QRMp4TnpQmNxxEm0zkvBz0AlXEMABj0EtOiOqIxrwaRSSEKh1eDHv5QjFcz1pR1E5cOLCmwbu3UBWAcMHbJFhSqkfbAkkKv0xyT2ssOGYC
QoNTUVDVv3tzapmXLtq2bZsKCgpcZVu2bFH58uWt4QgAwOXA50OsiYmJmzhxzqZNm6aNGzfqoYceUl
ZWlnr27Ck3wlqpUH5TLiHV5CgDpiZJy/w0aF4mDEJ6Yn9hI0ewEkkzBR1q4d8D+FY24Y4kq0/SRTmT4aC1ihkVpZdnH1mwhd6Y+0kLMYUPuXwmAabKAQW/TRuDxLlDBQGbjvWwvVKKAY93fBZnMlR5AuaLGMKOAqbcGxo0L3dwG2/X6eq0AgTD2OxOzhnNF2Jc7AFN1ClZZ
BgpaWlqWHDhlq8eLHrwp3du3erWLH/HQwUDWEpNPBXGpd+6WkKWIREgF3Mu5aRAyQXzYY/B3aJyJgx7gNoTNfwNbJTe2wAwjRBTPM+fzPUEhn4Ng+cCZm+vW+FcC5MlWgQcuzmmqe+EXRUJ7LNgQ5YMyUmZB1YbYD/K/
ERITefvttNWrUSAkJCRo0aJAmTJhQZJukpCRlZGS4hj179lzsbgIA4Mani3TKlCkjp9OpAwcOuJUfOHBAkZ
GR1jbly5eXn5+fnE6nq6x27dpKS0tTbm6utbX6+fnE6nq6x27dpKS0tTTk6O/P39PdoEBAQoICDAl64BAHBB+bQH6e/vr0aNGio1NVVVqlSA
QoNTUVDVv3tzapmXLltq2bZsKCgpcZVu2bFH58uWt4QgAwOXA50OsiYmJmzhxoqZNm6aNGzfqoYceUl
ZWlnr27Ck3wlqpUH5TLiHV5CgDpiZJy/w0aF4mDEJ6Yn9hI0ewEkkzBR1q4d8D+FY24Y4kq0/SRTmT4aC1ihkVpZdnH1mwhd6Y+0kLMYUPuXwmAabKAQW/TRuDxLlDBQGbjvWwvVKKAY93fBZnMlR5AuaLGMKOAqbcGxo0L3dwG2/X6eq0AgTD2OxOzhnNF2Jc7AFN1ClZZ

vg/yz8B9kADwd/A3vg8SAID/FQQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQ
AABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkA
AAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJA
AAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBC
QAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQk
AAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJ
AAAFgQkAAAW5xSQb731lqKjoxUYGKjY2Fh9++23XrVLSUmRw+FQx44dz2W2AABcMj4H5KxZs5SYmKg
hQ4ZozZo1atCggdq0aaP09PQzttu5c6eeeuoptWrV6pw7CwDApeJzQL722mvq06ePevbsqauvvloTJkxQ
UFCQpkyZUmSb/Px83XPPPRo6dKiuvPLK8+owAACXgk8BmZOTo9WrVys+Pv6/EyhWTPHx8Vq5cmWR7
V588UFRETo/vvv92o+2dnZyszMdBsAALiUfArIQ4cOKT8/X+XKlXMrL1eunNLS0qxtVqxYocmTJ2vixIez
2fEiBEKDQ11DVFRUb50EwCA83ZRr2I9evSo7rvvvPk2cOFFlypTxul1SUplyMjJcw549ey5iLwEA8FTcl8plyp
SR0+nUgQMH3MoPHDigyMhIj/rbt2/Xzp071071b59e1dZQUHBqRkXL67NmzerWrVqHu0CAgIUEBDgS9cA
ALigfNqD9Pf3V6NGjZSamuoqKygoUGpqqpo3b+5Rv1atWlq/fr3Wrl3rGm677TZdf/31Wrt2LYdOAQCXL
Z/2ICUpMTFR3bt3V+PGjdW0aVONGTNGWVlZ6tmzpySpW7duqlixokaMGKHAwEDVrVvXrX2pUqUkya
McAlDLic8BmZCQoIMHD2rw4MFKS0tTw4YNtXjxYteFO7t371axYjygBwDw1+Ywxpg/uxNnk5mZqdDQ
UGVkZCgkJOQ8puS4YH0CAPjq/OLmwmWBd9jVAwDAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAw
IKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAM
CCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABAD
AgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQ
AwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAE
AMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwIKA
BADAgoAEAMCCgAQAwIKABADAgoAEAMCCgAQAwOKcAvKtt95SdHS0AgMDFRsbq1+//bbIun379lVM
pkxS0tLSkKpqamrplEIEYjAgkpbwAQkASzlUpWdIrkJmpuMC6ZlTiKllM2dlS8ec59uY5OeOGFfA0UUZ
MpHBwzp6IYTl8mWxHuoZz2tLU1q6nzlZqvpMBbEh0WlovUHo6QH2PG7T9eIDRRxJxBCBGFkk
vZwe9JwLKzBzhN5jz2t4wQX55fQkKSGu4jGDaSGZyAjtv9IBLpRvf2znVfNd1hAoCAj3n3c9y2PBJFuN
3WsfE9Wu6Vz9fRYoRXDUSj7pG6ibFX7hVRvDt9TyO2svBz6hEFXvMxeD1nHLkmIe6GXmfGnh05rDXbrb
jCXktUaVzCj2hpmswHnbuk3ucwK58Od7O7f/0D3bWZknObnmGoMfaXJm5zwQvnc9/l6wKvbs33/+
f/T+LUlk4ISiJ+iIxhDq+Pwk3G5ztiQlyXM0UWUE8Hx+R67dv/XCANH/l5/Wm8WjZ9tO5LE+IKHOgb
f2UR8Tgw1jyA6q3TkpWZnp4ub6p/RW/OTeS3nFozxRM4/9/Gg5w/Vp0lCzdKjJt3G4N7rjY4d/19ye
3Pq/5EEH/U5w6zKfd2C9Bj2Fp4BjxzZXvuVXBjBjN5UDVNLy0l0mH8IK0yY/qT1cPPs7pbc5n+vBS9u6
NHP9JMVqVqTC0Pb9p9tX49bQDo1+W7tHnKWZGm9TOxB2s7Y3TkNnR0o5Ww4XKSzcw9qIpVXLwzKrY
tqb6C4MfRm2TlINqSOWlswQs8I1/iUxoRFldQ5uXpMVRSxYBTnkJqyIgYDBCJhjEmcqIAqSv/8pA+Lk
2wWmvFrVSp79KgBjAfzWflDpcmuzY3lcOcdw4rI3y/7AlnbM9zdB6oKtQbLG9PT3iM3aGfPkF5Fo0Ea6p
6Ql0oQ8GLrvfg/r0qOE7E1JS2LN02/Aw/P9R4MbI2bd0w6KoLKBcU9eD9Pqe06eU1+Ug+0vH+BfLBHN
ZZLwlxQ6iWnBgIZtbIKB0WAAFT87tnTvRL6Bqf7Ey3j9cAIz71f3N8Q5ZfPZ0ZkNvTtN2Ea+t0fsgWVhI
rJG9fVeQMV0EF5QAezm0+LNxh6tHlh8iLLYAAPlYQc3Y6s3IRDzyqLjFSqtKK/Bs7zk5vGqoNQeMnX
tH20Y0lnbwI0WnHZ5pmvGIN2B5tnBL7p9mlBoMX6s4w3kKoo/YnhWoqaWaQuG0Dqw5hbdD8aJ7lm
s1Yz62Rv6eUZh4RhYUHQ2Fx9zEAqSpJJQ8gdF5CjNIVEGN7wDzCCswCwCeZOFwDIcHCZq4mDDM8
bcN/F3MiCtF6mEQ7N2KY7oUgm6VbS5r9z1F2Tr8xtfBZTZA4/sbqJQs4HVhKqpxqAjCrKZDqxV3QF
c5mdT5PTDeAGp2YQq7OX9YD2NbONd9P8ZjIcXoRIgT5n1kQTRWNTMBVGJOhEh0KXxoUdQUgjFfj8v1
k1r7pO8pAwXQ+YjYwKJvOUCJZoYKAx7dHkgQVe0AAAAAAAAASUVORK5CYII=
vg/yz8B9kADwd/A3vg8SAID/FQQkAAAWBCQAABYEJAAAFgQkAAAWBCQAABYEJAAAFgQkAAAWBCQ

ICGDBniceoBwMXj832QAAD8L+BZrAAAWBCQAABYEJAAAFgQkAAAWBCQuGz06NFDHTt2LHL8unXrd
NtttykiIkKBgYGKjo5WQkKC0tPT9cILL8jhcJxxKJyHw+FQ3759Pabfr18/ORwO9ejRw6v+1qpVSwEBAdbH
Jl533XWu+QYGBqpGjRoaMWKExzMkf/jhB911110qV66cAgMDVb16dfXp00dbtmyRJO3cuVMOh0Nr16
61zuPxxx/3eF3Y5kzD1KlTtWzZsiLH/3GZMjMzNWjQINdP1kVGRio+Pl5z587Vjh07vJrX1KlTi3zYh8PhcD2
b+fS+h4eHKy4uTsuXL3drU9T2rlWrlhdbDvAOAYm/hIMHD+rGG29UeHi4Pv30U23cuFHvvPOOKlSooKys
LD311FPav3+/a6hUqZJefPFFt7JCUVFRSklJ0YkTJ1xlJ0+e1IwZM1S5cmWv+rNixQdOHFCnTp10rRp06x
1+vTpo/3792vz5s1KSkrS4MGD3X7B5uOPP1azZs2UnZ2t5ORkZ2t5ORkbdy4Ue+9955CQ0P1/PPPn+OaOrV8f1z
uJ598UnXq1HErS0hIcNXfvHmz27j9+/crIiJCknTkyBG1aNFC06dPV1JSktasWaMvv/xSCQkJeuaZZxQSEuL
TvLy1ZMkS7d+/X19++aUqVKigdu3aeTyR6/T57N+/XytWrDjn9Qac7rJ8Fitwuq+++koZGRmaNGmSihc/9
batWrWqrr/+eled4OBg199Op1MlS5a0PtLwH//4h7Zv3665c+fqnnnvukSTNnTtXlStXVtWqVb3qz+TJk9W
1a1fFxcWpf//+GjBggEedoKAg1/x79uypcePG6k48ePq2fPnrr11ls1b948V5uqVasqNjZWR4
4c8aofNk6n0225g4ODVbx48Slf7xgREVHk3t3AgQO1c+dObdmyxe35lzVq1FCXLl0UGBjo2h7ezMtbpUu
XVmRkpClJjIzVw4EClpKRo1apVuu2221x1LsR8gDNhDxJ/CZGRkcrLy9O8efMuyE/d9OrVS++8847r9ZQpU
1xPgzqbo0ePavbs2br33nvVunVrZWRkeBwC/CNjjJYvX65NmzbJ399fkvTpp5/q0KFDeuaZZ6xtLodnDxcU
FCglJUX33HOP9eHQWF4MZ04cULTp0+XJNe6Ay4VAhJ/Cc2aNdPAgQPVtWtXtXlSlTRrfccotGjRrlcdjNW/
fee69WrFihXbt2adeuXfrqq6907733etU2UxFv1atXV506deR0OtW5c2dNnjzZo9748eMVEBAgh8Oh2Nbr
SooKNBjjz0mSdq6daskeX3OrEWLFgoODnYbzhTKvqhUqZsmSPdq0aPVqFEj3XxjjjW511q9f77FObO0ewgXPFIVb8Zbz00ktKTEzU559/rlWrVmnChAkaPny4vvzyS9WrV8+naZ
UtW1Zt27bV1KlTZYxR27ZtVaZMGa/aTpkyxS1M7733XV8r9VVmXxenNN99UyZIlXeX33HOPBg0apMOHD2vIkC
Fq0aKFWrRoIcn3H3ydNWuWatewu7VZWeHj4fC1fvtyt335+fpIu3Y/S2syaNUu1atXShg0b0b9Mwzz2jq1Km
ufhWqWbOmPvroI7cyfukEFxIBib+U0qVL66677tJdd92nl4cOHKyYymRqNHjy7yQpkz6dWrlx555BFJ0ltvvve
VVm59//lnffPONvv32W7fzjvn5+Up5GfPn1cZaGhobrqqqskSe+//76uuoqNWvWTPHx8apRo4YkadO
mTV79sk1UVJRrWoWuuOIKr/p/p8NlWrVrUe0i1btqxKlSpV5I+h+yIkJERZWVkqKChwe1Zz4bnW03/CKCo
qStWrV1f16tWVl5en22+/XRs2bHB7/Fq2/v/XRs2bHB7Fq2/v7/HOgEuJA6x4i/L399f1apVU1ZW1jm1v/nmm5WTk6Pc3Fy
1adPGqzaTJ0/Wtddeq3Xr1mnt2rWuITExUZMnT/6sNNN6lXr15KTU3Vjh07tHjxYj388MNKSEhQxYoVi2zrcDj02GOPaeTIkTp+/LirPC8vz6Pf53pOG
rAhIHFZWbZsmWJiYtyGoUOH6uqrr1ZQUJCefPJNWzYUM2aNdP777+vSZMm6b777jvn+YWEhHh93uq
jjz7Sb7/9ptv91jXO3atVW7du0z7kWGh4erW7dueuGFF1RQUKAOHTro66+/lp+fn7p27apatWqpS5cu
ysjI0LBhw855mXxVs2ZNlS9f3m1f3f3m1YvXq1q8//ffPON7r33Xg0bNkwxMTFq1aqVZs6cqVGjRvn06+6zZs1SXF
ycHnzwQdWpU0ePPfaYOnTooEmTJp21bffu3ZWbm6tx48a5yn766ePflepUsX3FQAUgZ+7AgDAgj1IAA
AsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALAhIAAAsCEgAACwISAAALP4PK
2++XP1Ss2QAAAAASUVORK5CYII=",
     "text/plain": [
      "<Figure size 500x500 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }

```
  ],
  "source": [
   "def graph():\n",
   "    import matplotlib.pyplot as plt\n",
   "    data=[AC]\n",
   "    alg=\"LSTM ARCHITECTURE\"\n",
   "    plt.figure(figsize=(5,5))\n",
   "    b=plt.bar(alg,data,color=(\"YELLOW\"))\n",
   "    plt.title(\"THE ACCURACY SCORE OF LSTM ARCHITECTURE IS\\n\\n\\n\")\n",
   "    plt.legend(b,data,fontsize=9)\n",
   "graph()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "4171c8be",
  "metadata": {},
  "outputs": [],
  "source": []
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "689e3dcd",
  "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": "3.11.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```