```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "#Assignment 3\n"
      ],
      "metadata": {
        "id": "Y-SWXJz6KrvO"
      }
    },
```

```
{
   "cell_type": "markdown",
   "source": [
      "#Importing Packages and loading the dataset"
   ],
   "metadata": {
      "id": "f1MgPTnCM_ik"
   }
},
{
   "cell_type": "code",
   "source": [
      "from tensorflow.keras.models import Sequential\n",
      "from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense\n",
      "from tensorflow.keras.preprocessing.image import ImageDataGenerator as idm\n",
      "from keras.preprocessing import image\n",
      "import numpy as np\n",
      "from google.colab import drive\n",
      "drive.mount('/content/drive')\n",
      "import warnings\n",
      "#Supressing warnings\n",
      "warnings.filterwarnings('ignore')"
   ],
   "metadata": {
      "id": "RVrxKNNRND2l",
```

```
            "colab": {
                "base_uri": "https://localhost:8080/"
            },
            "outputId": "c28e88b2-59b1-4d7a-feef-ec060a4b3d71"
        },
        "execution_count": 42,
        "outputs": [
            {
                "output_type": "stream",
                "name": "stdout",
                "text": [
                    "Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount(\"/content/drive\", force_remount=True).\n"
                ]
            }
        ]
    },
    {
        "cell_type": "markdown",
        "source": [
            "# 2.Image Augmentation"
        ],
        "metadata": {
            "id": "NvKmqbBpNr60"
        }
    },
```

```
{
   "cell_type": "code",
   "source": [
      "# Creating augmentation on training variable\n",
      "train_flowers=idm(rescale=1./255,zoom_range=0.2,horizontal_flip=True)\n",
      "\n",
      "# Passing training data to train variable\n",
      "Xtrain = train_flowers.flow_from_directory('/content/drive/MyDrive/Assignment Dataset/flowers',target_size=(76,76),class_mode='categorical',batch_size=100)"
   ],
   "metadata": {
      "id": "Hibiw2_QNvDA",
      "colab": {
         "base_uri": "https://localhost:8080/"
      },
      "outputId": "05d4c9dd-a86c-4658-a977-ddce37a1c99a"
   },
   "execution_count": 36,
   "outputs": [
      {
         "output_type": "stream",
         "name": "stdout",
         "text": [
            "Found 4317 images belonging to 5 classes.\n"
         ]
      }
```

```
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "# Creating augmentation on testing variable\n",
     "test_flowers=idm(rescale=1./255)\n",
     "\n",
     "# Passing testing data to test variable\n",
     "Xtest = test_flowers.flow_from_directory('/content/drive/MyDrive/Assignment Dataset/flowers',target_size=(76,76),class_mode='categorical',batch_size=100)"
    ],
    "metadata": {
     "id": "EFaav0igOU4O",
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "outputId": "8f482e60-3b9c-4595-a932-48ce2abf4cb2"
    },
    "execution_count": 38,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "Found 4317 images belonging to 5 classes.\n"
```

```
          ]
        }
    ]
},
{
    "cell_type": "markdown",
    "source": [
        "#3 & 4.Creating Model and Adding layers"
    ],
    "metadata": {
        "id": "4UFGMaJgSsTq"
    }
},
{
    "cell_type": "code",
    "source": [
        "Flower_model = Sequential()\n",
        "Flower_model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3)))\n",
        "Flower_model.add(MaxPooling2D(pool_size=(2,2)))\n",
        "Flower_model.add(Flatten())\n",
        "Flower_model.add(Dense(300,activation='relu'))\n",
        "Flower_model.add(Dense(150,activation='relu'))\n",
        "Flower_model.add(Dense(5,activation='softmax'))"
    ],
    "metadata": {
```

```
            "id": "TtD2CeNrSt2Y"
        },
        "execution_count": 106,
        "outputs": []
    },
    {
        "cell_type": "markdown",
        "source": [
            "#5.Compiling the model"
        ],
        "metadata": {
            "id": "YR7oyHYlidYr"
        }
    },
    {
        "cell_type": "code",
        "source": [

"Flower_model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])"
        ],
        "metadata": {
            "id": "f3PKgIc4T3Gg"
        },
        "execution_count": 107,
        "outputs": []
    },
```

```
{
   "cell_type": "markdown",
   "source": [
      "#6. Fit the Model"
   ],
   "metadata": {
      "id": "sdMctRVCUURC"
   }
},
{
   "cell_type": "code",
   "source": [
      "Flower_model.fit_generator(Xtrain,steps_per_epoch= len (Xtrain),epochs=
10,validation_data=Xtest,validation_steps= len (Xtest))"
   ],
   "metadata": {
      "colab": {
         "base_uri": "https://localhost:8080/"
      },
      "id": "B4GRVOUJUajl",
      "outputId": "e4dabfa8-d173-4be6-b327-b96c50e23ecb"
   },
   "execution_count": 105,
   "outputs": [
      {
         "output_type": "stream",
```

```
          "name": "stdout",
          "text": [
            "Epoch 1/10\n",
            "44/44 [==============================] - 50s 1s/step - loss: 1.5659 - accuracy: 0.3692 - val_loss: 1.2139 - val_accuracy: 0.5022\n",
            "Epoch 2/10\n",
            "44/44 [==============================] - 48s 1s/step - loss: 1.1107 - accuracy: 0.5420 - val_loss: 1.0876 - val_accuracy: 0.5847\n",
            "Epoch 3/10\n",
            "44/44 [==============================] - 48s 1s/step - loss: 1.0149 - accuracy: 0.6006 - val_loss: 1.0763 - val_accuracy: 0.5810\n",
            "Epoch 4/10\n",
            "44/44 [==============================] - 49s 1s/step - loss: 0.9310 - accuracy: 0.6456 - val_loss: 0.8610 - val_accuracy: 0.6736\n",
            "Epoch 5/10\n",
            "44/44 [==============================] - 48s 1s/step - loss: 0.8593 - accuracy: 0.6715 - val_loss: 0.8155 - val_accuracy: 0.6933\n",
            "Epoch 6/10\n",
            "44/44 [==============================] - 47s 1s/step - loss: 0.8197 - accuracy: 0.6910 - val_loss: 0.7720 - val_accuracy: 0.7132\n",
            "Epoch 7/10\n",
            "44/44 [==============================] - 48s 1s/step - loss: 0.7673 - accuracy: 0.7051 - val_loss: 0.7954 - val_accuracy: 0.7044\n",
            "Epoch 8/10\n",
            "44/44 [==============================] - 47s 1s/step - loss: 0.7366 - accuracy: 0.7267 - val_loss: 0.7716 - val_accuracy: 0.7213\n",
            "Epoch 9/10\n",
            "44/44 [==============================] - 48s 1s/step - loss: 0.6921 - accuracy: 0.7343 - val_loss: 0.6432 - val_accuracy: 0.7501\n",
```

```
                "Epoch 10/10\n",
                "44/44 [==============================] - 47s 1s/step - loss: 0.6527 - accuracy: 0.7575 - val_loss: 0.6604 - val_accuracy: 0.7554\n"
            ]
        },
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "<keras.callbacks.History at 0x7f9608202090>"
                ]
            },
            "metadata": {},
            "execution_count": 105
        }
    ]
},
{
    "cell_type": "markdown",
    "source": [
        "#7. Save the model"
    ],
    "metadata": {
        "id": "HUrKXOqSWoyI"
    }
},
```

```
{
   "cell_type": "code",
   "source": [
      "Flower_model.save('Flower.h5')"
   ],
   "metadata": {
      "id": "7Ge-hXhNXKnD"
   },
   "execution_count": 40,
   "outputs": []
},
{
   "cell_type": "markdown",
   "source": [
      "#8. Test the model"
   ],
   "metadata": {
      "id": "h1kh3jlnXOsx"
   }
},
{
   "cell_type": "code",
   "source": [
      "from tensorflow.keras.preprocessing import image\n",
      "test1=image.load_img('/content/drive/MyDrive/Assignment
Dataset/flowers/dandelion/138166590_47c6cb9dd0.jpg',target_size=(76,76))\n",
```

```
      "test1"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 93
      },
      "id": "cLSmt_R_XSnP",
      "outputId": "1b41ee16-2e6d-4096-aa56-d99baa6080c9"
    },
    "execution_count": 108,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "<PIL.Image.Image image mode=RGB size=76x76 at 0x7F96080A9890>"
          ],
          "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAEwAAABMCAIAAABI9cZ8AAAKMWlDQ1BJQ0MgUHJvZmlsZQAAeJydln
dUU9kWh8+9N71QkhCKlNBraFICSA29SJEuKjEJEErAkAAiNkRUcERRkaYIMijggKNDkbEiioUBUbHrBBlE1HF
wFBuWSWStGd+8ee/Nm98f935rn73P3Wfvfda6AJD8gwXCTFgJgAyhWBTh58WIjYtnYAcBDPAAA2wA4HCz
s0IW+EYCmQJ82IxsmRP4F726DiD5+yrTP4zBAP+flLlZIjEAUJiM5/L42VwZF8k4PVecJbdPyZi2NE3OMErOIlm
CMlaTc/IsW3z2mWUPOfMyhDwZy3PO4mXw5Nwn4405Er6MkWAZF+cI+LkyviZjg3RJhkDGb+SxGXxONgA
oktwu5nNTZGwtY5IoMoIt43kA4EjJX/DSL1jMzxPLD8XOzFouEiSniBkmXFOGjZMTi+HPz03ni8XMMA43jSPi
MdiZGVkc4XIAZs/8WRR5bRmyIjvYODk4MG0tbb4o1H9d/JuS93aWXoR/7hlEH/jD9ld+mQ0AsKZltdn6h21p
FQBd6wFQu/2HzWAvAIqyvnUOfXEeunxeUsTiLGcrq9zcXEsBn2spL+jv+p8Of0NffM9Svt3v5WF485M4knQx
Q143bmZ6pkTEyM7icPkM5p+H+B8H/nUeFhH8JL6IL5RFRMumTCBMlrVbyBOIBZlChkD4n5r4D8P+pNm5lo
na+BHQllgCpSEaQH4eACgqESAJe2Qr0O99C8ZHA/nNi9GZmJ37z4L+fVe4TP7IFiR/jmNHRDK4ElHO7Jr8Wgl
0IABFQAPqQBvoAxPABLbAEbgAD+ADAkEoiARxYDHgghSQAUQgFxSAtaAYlIKtYCeoBnWgETSDNnAYdIFj4D
```

Q4By6By2AE3AFSMA6egCnwCsxAEISFyBAVUod0IEPIHLKFWJAb5AMFQxFQHJQIJUNCSAIVQOugUqgcqob
qoWboW+godBq6AA1Dt6BRaBL6FXoHIzAJpsFasBFsBbNgTzgljoQXwcnwMjgfLoK3wJVwA3wQ7oRPw5fgE
VgKP4GnEYAQETqiizARFsJGQpF4JAkRIauQEqQCaUDakB6kH7mKSJGnyFsUBkVFMVBMlAvKHxWF4qKWoV
ahNqOqUQdQnag+1FXUKGoK9RFNRmuizdHO6AB0LDoZnYsuRlegm9Ad6LPoEfQ4+hUGg6FjjDGOGH9MH
CYVswKzGbMb0445hRnGjGGmsVisOtYc64oNxXKwYmwxtgp7EHsSewU7jn2DI+J0cLY4X1w8TogrxFXgWnA
ncFdwE7gZvBLeEO+MD8Xz8MvxZfhGfA9+CD+OnyEoE4wJroRIQiphLaGS0EY4S7hLeEEkEvWITsRwooC4hlh
JPEQ8TxwlviVRSGYkNimBJCFtIe0nnSLdIr0gk8lGZA9yPFlM3kJuJp8h3ye/UaAqWCoEKPAUVivUKHQqXFF4p
ohXNFT0VFysmK9YoXhEcUjxqRJeyUiJrcRRWqVUo3RU6YbStDJV2UY5VDlDebNyi/IF5UcULMWI4kPhUYoo
+yhnKGNUhKpPZVO51HXURupZ6jgNQzOmBdBSaaW0b2iDtCkVioqdSrRKnkqNynEVKR2hG9ED6On0Mvph
+nX6O1UtVU9Vvuom1TbVK6qv1eaoeajx1UrU2tVG1N6pM9R91NPUt6l3qd/TQGmYaYRr5Grs0Tir8XQObY
7LHO6ckjmH59zWhDXNNCM0V2ju0xzQnNbS1vLTytKq0jqj9VSbru2hnaq9Q/uE9qQOVcdNR6CzQ+ekzmO
GCsOTkc6oZPQxpnQ1df11Jbr1uoO6M3rGelF6hXrtevf0Cfos/ST9Hfq9+lMGOgYhBgUGrQa3DfGGLMMUw
12G/YavjYyNYow2GHUZPTJWMw4wzjduNb5rQjZxN1lm0mByzRRjyjJNM91tetkMNrM3SzGrMRsyh80dzA
Xmu82HLdAWThZCiwaLG0wS05OZw2xljlrSLYMtCy27LJ9ZGVjFW22z6rf6aG1vnW7daH3HhmITaFNo02Pzq
62ZLde2xvbaXPJc37mr53bPfW5nbse322N3055qH2K/wb7X/oODo4PIoc1h0tHAMdGx1vEGi8YKY21mnXd
CO3k5rXY65vTW2cFZ7HzY+RcXpkuaS4vLo3nG8/jzGueNueq5clzrXaVuDLdEt71uUnddd457g/sDD30PnkeT
x4SnqWeq50HPZ17WXiKvDq/XbGf2SvYpb8Tbz7vEe9CH4hPlU+1z31fPN9m31XfKz95vhd8pf7R/kP82/xsB
WgHcgOaAqUDHwJWBfUGkoAVB1UEPgs2CRcE9IXBIYMj2kLvzDecL53eFgtCA0O2h98KMw5aFfR+OCQ8Lr
wl/GGETURDRv4C6YMmClgWvIr0iyyLvRJlESaJ6oxWjE6Kbo1/HeMeUx0hjrWJXxl6K04gTxHXHY+Oj45vipxf
6LNy5cDzBPqE44foi40V5iy4s1licvvj4EsUlnCVHEtGJMYktie85oZwGzvTSgKW1S6e4bO4u7hOeB28Hb5Lvyi/
nTyS5JpUnPUp2Td6ePJninlKR8lTAFlQLnqf6p9alvk4LTduf9ik9Jr09A5eRmHFUSBGmCfsytTPzMoezzLOKs6T
LnJftXDYlChI1ZUPZi7K7xTTZz9SAxESyXjKa45ZTk/MmNzr3SJ5ynjBvYLnZ8k3LJ/J9879egVrBXdFboFuwtmB0
pefK+lXQqqWrelfrry5aPb7Gb82BtYS1aWt/KLQuLC98uS5mXU+RVtGaorH1futbixWKRcU3NrhsqNuI2ijYOL
hp7qaqTR9LeCUXS61LK0rfb+ZuvviVzVeVX33akrRlsMyhbM9WzFbh1uvb3LcdKFcuzy8f2x6yvXMHY0fJjpc7l
+y8UGFXUbeLsEuyS1oZXNldZVC1tep9dUr1SI1XTXutZu2m2te7ebuv7PHY01anVVda926vYO/Ner/6zgajho
p9mH05+x42Rjf2f836urlJo6m06cN+4X7pgYgDfc2Ozc0tmi1lrXCrpHXyYMLBy994f9Pdxmyrb6e3lx4ChySH
Hn+b+O31w0GHe4+wjrR9Z/hdbQe1o6QT6lzeOdWV0iXtjusePhp4tLfHpafje8vv9x/TPVZzXOV42QnCiaITn0
7mn5w+lXXq6enk02O9S3rvnIk9c60vvG/wbNDZ8+d8z53p9+w/ed71/LELzheOXmRd7LrkcKlzwH6g4wf7Hz
oGHQY7hxyHui87Xe4Znjd84or7ldNXva+euxZw7dLI/JHh61HXb95IuCG9ybv56Fb6ree3c27P3FlzF3235J7Sv
Yr7mvcbfjT9sV3qID0+6j068GDBgztj3LEnP2X/9H686CH5YcWEzkTzI9tHxyZ9Jy8/Xvh4/EnWk5mnxT8r/1z7z
OTZd794/DIwFTs1/lz0/NOvm1+ov9j/0u5l73TY9P1XGa9mXpe8UX9z4C3rbf+7mHcTM7nvse8rP5h+6PkY9
PHup4xPn34D94Tz+6TMXDkAADF0SURBVHicNbr5u6dZVeW59z7nvON3uvONG3NExpDzBGSCQGqRkIhM
AgoqgkIr2lrYQvlU2dVapS0qWjSWVrUU4EQVJQ6gVSlgMyRJZipJzlNkZMxxI+74nb/v+55z79t79g13rf1jPsz5rL
XzBm52dd3GXrcuKrGxjezaI48lsu795boPTeVSj9QxxpPkCdfY6yHMejEa70ghlKZDgrK8wAZmCAYRUNYdiHrt
7uV2Q99Kv0PexipIr7jkAh9eW0jS/ut5/9IlRnBFE0CDGGlaFIICYzNu8DVlbsZGhgh1RzWIdBI8cVBNJUrDztMd
I6NjMyHRkxhxbFdiW4jy20GyFaGss1cxYlbGeqk1zsJVYsFOOSYyYc2pk4GqPbK2xCUbQDDn2rO2oc061kSFA
baRmWSCIiojsBAXUkhpGh2UKWQrtRI1gIng1AauuamJAnUmjkUc4NRmhUKwEG4SgRpQVjDF+EsmBQ5hZ
tdHMQJHQGg0AEBAJMVMEgbbLmiACI68OwQhIi5aM80RUq6+5LmM2pijAlVrqLI6lH+upm2FjMWA54vE4
REbwjjOgTMAXhFZsi4zF0EAl2HiFBtADE4gIGFBCxQgJWod5DxccuhZOsrTgpktQMRMbP5FqXsVCK0JqFFMN
gA1jnKpxBjJFRAAVNpMESHiaKU6ArGIGNmJkAdV2ZtBqzjwzGBQTRqNiMsoSRMexipahidAd0SxQVWusxBa
tru5u7SB4NFTPEvI1Q1AYI1gyxEzOkZO0xNIZSHU2ip4tcVTEphFrgYA0oICSQSBIU1hInBbRJHmK0/GCJhmR6
oYTZGmahiTuOLG5YwjOUYgKlYJT6xlzTTIQhIJRHOYzrZSSHGvDSIqIJBpRFw2GRAtvpiF6Y0JCgGBBXaUekQE

7aCYEFKWeATNSnrUxN3NkgDCosq+moYm7lVYmiWxTA4E7TgqnpXGphzxwnMY4ExCEiMEToygCiIhTcGC6
ELKmk+YFQZLIEtFSirQAK4mqAaw5NDFjk0goc2pl7AqDTpXAJAaj5I56FpRiBoBoFMBZnsuQFMgiInqL7DQ1B
omUQURAyTooEoMWsWEbcSpciRk1ygwgRGlmkqIzzbGObNBekrGvZiPP2QxKZ8rIpdPY1T2tRMrUWx0GGz
yTR7KojC5QdxngClLLmsSIJzBFYRRT5KQk001S5/Zlq8utfK9ZcI68m9RexzFaTXpALUgLnF/LkJiVWx3spJDtlVb
bjlIFAECReUuZklE0IgacAYOqPvoYAMiitRU7YZU4ZhGyvgrAhsYaIqmqoFCMSaqUGssG2Yd0FqNCiBqAk6CK
DnLoAAYq0iTEGMM5cCMAADGsKL7B838IO991Mo233ST3/77+xx+dUDspnHWNemqdC8V/e+t0ISuW8q
W3vIRHJugEIltNMUMoW7Z0QbSxahICJiUDxGZsYnQYfEzQLEduG2JEtZhm2BGOooGUIjQsVqQxMEthhjJN
MdQaCLxTI4IKENAp0ni66diiikmVMysJ+porhkItCU7q0DbUcqnAdBYaHsnsye59fwaSS4Ip2izbDdG5uZX2hf8
Bf/s7yxeuysoaNDWAn1g7u/8XNz/37vHWum3lxfKe3vvfmcV0LSbp1EdDIGmSY1LOaytTW0DHQZJjVVCFsYj
UmyEDFoCSWjEaUlOkWDiknsnJERpGIAaJSl7mDPWizSpCBi/QFUxVmVVJGYRm46qazlS1mKIDUsWpQx2bq
1txYywJo89A02xGuR+jzgI2o8M3mp99k7v7xc1Hfrr2BtLS7jRpvrez2B6sHYTje1ufe0/sh3TqwyhfcBoKGzXPg
8v9rLl2ITvY7mjmck5Tto0xJJlHaJFOM7IJOOAYkSNkwaaRvDKwWIRiGg0CGXGBJTKyzASDNVE1DUCWxiC7U
ZqIEMBbIwIogI5MCbS+s36pHg41VCQ2ogYuAogIBJJKqYWuLBvH3TimXZ1u02v/Lz+fzb//XenP3A3v/qkFaQ
wGlw2vDGbuoe94g25P0ex4vWnRVxPOvXYXO5eblvPw6V9M+jv88Z+plw92W9QOplbjW0rkIEugKUyKmIg
BViEcqFTeB4W6hjFxhTRNjKqyASjRpzBAJVFsAKINLcjAQBBToZlCash7PyIBAJdA0UKq+lJVmIgjolFs0BmbO0R
UVY/gcmMKaEHrmgUco2zc2/77P0ug0x0OJooUXXnff2H2U2rZMu6srabru8EY11rW97zEDzdMkPTyo5OPf
UvG/Sm5V+zbv7j3YG7arQc+Xnzn/903CoRepDHOmbTmGGWHWIJlFhNxaCE0rCwdNlkDZQ3eQUJoDSUOH
KKJGlQ0BgIwPhACe5nGqKTI6BIih66A3AFNdjTsSjVQVEC0PmrtBUhV1VhqLLeweOCXJ3cfDe95kSa5v/oNG7
0c3dPds09SuLh0I3QOLWdltrpabg3roMDqE9ddXNAnn6ktepPKs6f9/sWt4B/tzLd2t8//5ht3u0vZxfOX3v7K5
jO/nkaepl44wYDoIjQqNYGLkEUABiKKkSvUWiLWwoqo4JDaDYjSGKFGsLlhkqjQKDkiw1pZpUCJSK7CoDSZSL
2jk2EERj/iWEOToCZKKcGc2LaNpYX53IteUxw/iB/4QFy4Zi0RLDtxvpeEsP/Y0QU23VCksd257ZB5xRFAic3ms
LT4Gz+WzXauujJ/1bHwW6+vdOPxOKkUekcO2S9/4dwLT972Gz/zgnq8MwnQZ20QnKCrlT2ZGtAasoaIyGJ
WOmWBBkjsLjMApgYmGVYBqIKWogMOAE1EmKkxhqIrG/KRZ6nOrEYiwhrqhqSyw75MGhwjJBxtipqhyaEj
qTcAHW3l/shy3LcWLCSAV6GgrYvsWi0NwVy85MaVq+pZA5JbwzC/jKnBt7045Ck4qY8dcJBLsnbrl769u7h3Z
bYzu+EoQXfGk+27P1h4Jm1IplwbnSJWLGUkAWFlZaFEE5IEUdjMZpKIQUFl8KSkhMEqYoHIlmQKEDBIzL2aG
q3DHhoidF4IAIPXysCskaaCzsQ4da2OcR32DcRep2v0zvcPeeHInmVzYgH+n489FcvcULa1g1/5wrOIdutpLxA
59ltaLF9RyOZaoKlKZ865cm+nrS+9PuyM7M7Tf3vDgbh78bHnZlf3H75Onfnbf1hHxa1PJuMAYFwCkBogViH
MUIuMyEBiYcliBFWv6EErGjDXKKaChpmrKEbJqEQQtRWLZb3keMYiicZSU2cmBohESSkAt8gY1YriVEUc5M6
Gb+79V7ddXYx1t2zZ888dOLbSW6F3vjWzo4WYdH/gbXjHYdO/tOtWUuC59afspcvj0RRcUYelw+cuwcI8ab
ObEZw4tLqiulDCpYE223rt0oHzjz9abfJH/qH39Efyyh7JrPU2klFGKwZrB1oQkFJmAHSYSipIYACgDrGsFEVBlT
1IVMuoxrLA0AsAjC0UjUZHi8YYAySEDRABKmFmkXNOEb2B1EbmqK2ILnnDy5PXHx3cWG3h6tLwucsGkrm
17l9/6szkiefjLM8XsADqndiDvcH+o1FT2LhilCL1NyGDZjLz68FZF83QLgCceFFV47ACXNh36MTt1/3qzi03sekt
z93z/NafSe6UMyBQcZKKBWGb25iw6xEaEIUoAQCQaahKYNtkgwEw5AqjEIGFg9JMbSAimyr7lLuoLQ8+AsU
Z1JEzJ62udVasam5sp40nV+z1/3K2MQmvv1tvPMRQ5JDjnv2MXLz5zWU9wOmFEeVzksjwmVN6cRoW3f4
VM7fqdTPoaFpEa61NO/L8iGlcP3eKLz38nVe8+FC9bpJh81MfeeTu/f13HBj0L+4MPxP++31ZmUJqKLOsCpF
9WhJmOp8YYxWUghIAAAE20vEWWCpgR2gdYktyAGJAUXEmSchmarsQM1WLl3OBmkgaNQ2lmZtHsRmUi
YlO1rruR68z/8dtG5/8Bnz1m3LXDfTBX7mweET7V/GBvziri0t7Xn7UeYh5N+8YW9POBvB6GA84aRvtIiTlwi2
RJ6oix/YlUuvyDO79QgLT88dvxSefePiP/9R+7BfNgXmeW4D1Pv67z9ZJ27Qsew8JQEi0DdBWMVZaxtRRQsP
kjBjUqCNSDeoTaxSko5gyJciGlJAMtAwkwklmM6UZCU4ie7AgxI34BuO8bZdxaNgE85/fznd8z9Ijj/YPntf1Gd7
7NJ84bENnbeWmmupNrRUGlyxB8/XnMedsIdl6QuevVZeynmfZINisYwJkRDy6Kmw+BkMPP/RmhqHc/EP0
/GngL7P2Y1VR1wiMpWH9T/+q/crjB5be9IwDJgRKwQZwhAiQ7chUSETIohgEhgZBmoguLaTpWGyn1EyEVV
1BjAotdIRpVF+DegJRSwrgsdqNcd41qSKDsvzMZ+z/zZuvfuXyzdfj7jg++p2di1uMplc9/9TiYQf1mbh2Mlnw/N

h5u7rkWzx3IVBZ6VhNAI1CmLgxMyqRgZR1LlsqkGwFe4vnPx9gXcALZPnq9yxGu7B83b7zf2kQd5577Km0M
HUgixKclpGmyE5pEyE2AghoEFld1FkQCFBpjBnmKTZWZ6hZlpD3SBgRpjFQSlxZX0VUJTECTpPGjWdshf7o9f
Cn79R//TK+7qj+1h/2sczme/ZfvPaWZadf/cPH7LZU/WiK0vbP4SNn81xDs2OfGbbcmE1p2GJhKO9CLwJZKh1
M+dGvwp6bmzRlGprqGzNZzICp6mTgq2e/c5WApsMzDP1vfvnMD/xOO4HoMi0ELNqKJLfgETRQmAEoKCF
Zwx61BkdJW7RFWBOzpXZqlLlRkoRUtQjgKg0TJq+KQHvmCBGmUWhsPOgrby9JTCH2kfv0rjV//uo6mEPAm
7ffUbrUyVxiIeHcQeVhRXVtEVuE+w0dIVOyVFqNKezOIDEwiXKsp6t48+2oqXPCYDh7mUMOk74kLptdghM3
HVE/aC4O3viB8+/7q+IPfm70D79AJksks6oqCKZlFKKPihGtdSKCBiqjVkB9nOZoMywjsKgNgMq5qoiiUEpmw
CQRMDGpQcpaVHSgqniyHWCGvffNztYcYgSHZ3fhvk83z559BKbd3lrxfW+bv3y/zyGa54cwVtjbJmKrAusMU
6ELMwLMlwJ2A9sQ22AGW7ST1WcsnPPmMFcEeCZqiK3F5NIDg/IakLIV18/Isnv3NfzABw/cc8fx939F7ERm
PkaNwkBOwUIWtAkUIRhAQYZgdhkmXq1AyTpGaGqdKoIzqaHSQ4k4YDZDbSphELVg51tKqdnt4y6zm8oSm
d+/F7tri0skS27jNmePn1zYfuYpito92T32ri6N+jIyO5e5c80sfW7KnQIPt2S0AbsKOdgG4jxlG04nDdjcbzXJKxSf
QHPZ0jjsRtttK7RDq5VCZy9tnx703f5bXrb/utn280/CePkLH335wgvuSyK1vVUrmrGLpq+KlSqSApA1nmPGxi
hioWCRjQUBqBkr0BQaBM/CDTZjNUpIagogW7TzjJKEqQHnIaTRGduxZkbtzz6dPvpsRDSda69RhYvfGkpkxk
QDz/fy85/mcMThbsU8wJmDoNxEcCbbRNGGrwFYr6tK5WGnHQd7QlJQ+yVqZugrnb+uF66eG52ZrNx6Qu
Mpmuld767f9fsXbHbfrOFZgInwLFXuGkGIajQKBQXBSOprrSqtq9hRbFzMAqTjGAAhA1FjBAKjzJCCAQBDYA
ugHMw8lt0CHNJ0aPMc0wwfPxV/4gWTy39z++/8nLnw7StJy8/f+PI8hyc+OzVDGW+CucEd30t2BNBmc5bJA
ZaQMEHQUDGlqbEELexcp1J6Po3xWUDj5L4IZfA76fTRqSHJj8/DSJ/62qX29Q8+81T47H+5x66IzExohDxnDe
3xAsQ0U6gQxahn8CoB40wxsbXhPCWmiGQF1Aj5hvuMXOm0ghgYGMios0Aj5U7AJZekCK8+qD+wZr7+8e
N/9varb75nhRrfuu34wZcd+cIfX/zSL3yzs9fdsIdPPxfzLkY7gizyMxXtARLF0hmFGCSiuBbprAlXJLxUcBeSJrU9g
TXTjBo+YAYDoE5s3ZSO1qHZHly9+uwffAP+7j8Y3n0P0eezDn34V+HCJ8r9q7ZhnkVFgAEqsHIQIESv0AAH9B
wTh5mAJao5qiFV9bVaj3UD5BUBlRAcRlZ80V3zdgoj47fHs1ccgh97rfv+dx+C4eD5ezcO3b2/PnUpAOVHe2
m7p7XX7XVaPnDhE+fn1zBdY5oC7wOIgEPArrUFwIPq72B3DngD4HaYPUedKLDkQOKsb9J21BNgZzA4C5NB
uu/O8tzDu7SKBw5+T4zCrqYASgOppH9q40UfblrzEi/DpTNYXwKyCIUKKBhEp/kCLeyRdkaS0KziYSQeYxiBidF
7AzPDTYQWFItYGLCJ2gn51kQZ4FSEvpMLXzxbW79dpkfTIj20Wix2t7/5XKgGWc7dPA3ZxbUfa5vNUfMtNPc
4uwPM3uXZrK4tt8O+cZlZWBRYsbOvxs4NAFMKnqmCQvmxf8RDHdvJguF2a228+Tne95OvMtUWhG2LbKt
4aWuyOod/9I3NO5bANwgD7BsSBRRWBYhqHKmCRnQOETAocKO1YBJUG+VI3BhtQGNAgwTYTFlSwhMvt
C1NEm8v6WhuxQz37P2+xQu/9vrMFc13H8Ef+N+uk53ps/ef7Rm4eNrsOST7X2ZhGGB+OUw3zT+6eD0kG4
FHaJYyOVXRD2bxarALJXxrFDOwCrqU6XYtY7C3pU2E1EP/atNOE575j52Bo/sNROxPo2/ggcezP/oQPP4FPP
mSNyy/8lPSMdaxGZn1swrrQimBA3CqCJpDdw3bC5AJxIBjVBcEh7QzplhF9QCGUAATBCdoCDtHYKWTpA1A
HuVEcrxTv3TFgBGjpIHe+JZw+IY2D3siUU9tscTRGVh+CbAx0ii2yPQjEFFi2AfcBNwPAAUWFWTYPC7J1OAqy
BzDowBLICtoz2jdS7ITb475mfj0g+b4HZR24+AcTnW3miRuZeG2R1aOpnnHmzax8nCXJqcpbDM4RUuQAqJ
KF1b3UJ6zKmwz7BE7LLjaUt9H36CJqpY4CiCCKCJSmOF2FerIkWDetV425/JcFxKaT6SN4Q9/F5u6Z7oLLk3c8
bZ5wd7l16zgFvh7rTXqCEBAQfhqpDHQCOFZQA3SA+jbtJthm2Nf4AnHt+YxBctJ6EJy3MP63+DFh+5fp/D0g
w//7Rdn58+4Hi0v5r1yyz9+3Z0vBmvBFpA48o3oVFEBI6AAsACAMeidBgNqTObMDCOyKoIxmCWadhOTqF
GEIEYQo1ofIUx0uCuf+Ldwx9FtFrxwRS9sQpjaTg+SVnzi/svJxkXdwN6eJAz6R4+5IJCVQGqgm0AW8TkCRc5T
fKaBd4j5KtJ1OWw1sjeGCO440IT5jMBLTTzL+mhOixV0cHdd3vmm9N//n6/+/peci4tHLz1/dM/qJ2xqIak++v
Odu967k0ceMsqMohdQpX9GykhIlCVCIibQKLAYKLrWRhWEMXKrZTCLPuCUhQYkDMBKXOt7f9Q251bf+sq
D1xw+ULPb2rEU8NKz8a3vvf3H70lkE6E2x9+V7f/+8uBL0bdzWiNtNbAe+f6azjm6hqjP1tYyZGwroOOlAC2h
XOiQ4wlWxlKL3dcRrkqyH+Dx8rvf4uU75fWvOfo3nx9suRe2Rz974M4P/tSHdp87c3E2DDgbERkEkgqxJowA
ACIiXhQUDAMqKjSAEyKr5CgKamqBMmqlUpaSplqWqF4wqipYULj/tGqa3P/Q+sa5oIRqkul2/LUPth77yoOu
zHSqT9TmZNfJwONSltBIapDzCRwOOCfwiHANsGJxPcKCgwcsvCyYDWWncBrcGssUhD2tuLgU5NvW3lk//lX
9y0/QDQc6588++dof+fCNL3w54rkHvv1LB4sPPXovnnjPFq7x4jWxQ4YjzlgIEQDQgKBaMOxUE1YFAVIfY6qF

GkUNyAmaqsRCxRntI6A16hkUkXqIU/yJd8rr7gJfQ9vRhS379p9Y/ORfD97z28Fe1S99oXzxTatJrGMaTKjP/u2
u5nD0xyw8HqFGqZRMEVa8e5zCjnffm8C9Pq6lptVgA7CKoDYMg33BDeHJU1R6uw6yt8M7I3cSBldgsmlH6
3HuYJIlfu5WW65ZPwic88JxCHNEI+yfBhqIqioBGtSotEzFGuQpRCEZslvRhRakkepAm9ux1yFrtKp0Z5fqSwpB
SRGxJDSSrcCRowA5wnRpT72852Z4/Z7nhi1+18+ngEeGzz599bF44ocO9795vhCxRxd0Pmg9dc8qrJGei/C9p
vkUm/3gnjORlHpIKxB7iiNnSCIGGBN1AEmVNXi4esEevMbK4bq+SqO+5geObA0vH7vRgxUU0lzKY7CUwXoF
/hRgBBVAIiBVVljC1gK2UhwbjF7netzuogMaejSbyqVkFgcDHe6iXFVV0C5YbFTmTOX44c9cj/Xgm/et427feJ4
ZfO0PlvWp2Ze/9PRDTxb/7kOtZjp1AEx52h4DGniAwzJgqngDmKfIitpLAnvFXqPxIsQKklE2e5qLu4MkmDQp
ZFwbnw3JHnj/vtnvws1HadzDxYvzl9apJd32wckFLvWSOYjgJI+xH1UqgRrAEhhREBQgRRGUShpnfB0BACswJ
aLVhGjgYhfMbMwzj+wFFJWg5ZCUAa38xMuRkh00naIguyDbU9kZ4E///OhjH9X//dPx3390CrlJ82RrYe+V3e
76n2u8IHEO405mncaI8G0x+8gvIlyH4o0Fa/aCzIfstjCbIkaC/T5u+KRfgLPmkQ+bNYQLF+PGg+nSsae2HGpLYi
i60Czu9/1by3n1CVMUbhAEAAD/2ZaRAACj1oBhxhBsXpGKDRUZMXEWEoQR6mAE3pNGUFUUDSqIJdk1cc
7NMtq32HQtDBvbpbm1hcnbX6ldxi7Ugx2YN9nLb6mGAbsvXw1hQ56R9HAOD9XTWzS7ZOQw0ybQeaMn
RYdAZHUqsojiYxKNkMCu0pGWXvWSeL22Z5tB3IHtM+QQ5m9chKTwjbGuPPXkkyfvuFZZwuWt5bdtjdcRroI
aNZZERD2AA9tG7aFLsKmlVEjn0BZgM+nUduSkboArHM9UK5QdJkEpwCYHbK/lldPVPZMiQfXaC/LIH/vnpq
vHW2cvbcKDD0FR0cL1DSxj66Zbtj9/auHlvzje+L3ZJRmdsnurYBbQfAfgeoBCcSPRVgiSJN0QEpcWApcV5hNa
kf7XfPEipE1wj7GczJ962Ns2H791ua4DjM+zZpfOVt3FXphWtnR0eG28vau1ACgBsgoJKigajKjEEANaophIGoG
DOISR46kqB6qQwYAEJCBFxSnigVdAueDCGHvz0QC+/7V863UrhxbFdpY+88mnU0O9jA8cX1o7yooJ2BUsS5
ChcevP/lb/xAstd6Ied5AG+3dFXJ3ZnoNJgBnqXRQfFV7RdIBckwXRUpsrQAtgj99CG4+ooJYrg62r/3QW//iv3F
/8nothv0mZpVaJ66evHPtxCFcQZooGwBCwYIKQKM4Rk1pLxjJYBKc9Qk6VAKdEYUuAVRl5hDpWEAQQvOF
HkpX5rrKQm/zGu+desJYIQ1NNv/rlndWOe+B8eO8Pd/TAXhQS3nvpC1/cl+bbC9XsYbd4MCSHkDKN52xWR
z9y5piaSZQlJ07s1wnuiTBSeDyDNfYl2rWAA8U5C2cRVvjKGckEOjdeq7o7nez83K+7T/+36o7XwNf+4pY4mbj
p6c4b7eQJVlWyqKTqEHLAQssMNScVSDJJCMUJEMSSeCKd6K5cZWUAABgKNKgMiEoL4uY77blV+/e/tHD9
HFd1MBS+8rXdrcpeacJ737MHDr1cJhH6O9/91pf+4jOOXn1s1ocDP7U03cZkXmM3u/e3BeZbdqiEdjYE2Am
2L82JEFPVxkhuhs8GO/b1t11gM3hSpgt6x0/D0i127taFurrIUdPW2gf+LdWp+cbnnOk80Xrh86v3JLHOVBUA
VFUtUgGmQzpnspQy1CLVVollF+d7ZiGheUSXkGFVFQgKM4VIIAoAKkA7VtmFD9wzqKIUCH//5Su/+dGrCad
ZHf1ucfkMS3jayFJcbO79pvn4gzR58LHT33bS3+0dJy677q/ruz+k0PiQRmzqYq+TAsISpC6xz5AuOjg4hTaEFy
TFiTD4LnQXbNlJHvxE63V34RObu3mbv/vgJk4uWi79MysAATUy68zEpprC/xIWQC1ql9IzDG32hUoqRWnyA
pZS4lLQsbEyTYRSYyLqVDWKChAROLRu6usr1eCKL1ujL//T7L4NWnNyvqrXOvSqN99q5s/j1fqDv/7A/iPyvg9
YvTj/xEP9u3+22Hxqd/NpuvYIhjtg637dv6jpdcDXJJSJ/W4W2rXueO5l1FGaJN3rBFyIpS5/H1cX0ryaBQ+f/esE
SNt3Vlv3AseTpbu6K8nTX8EbXoOUKAdANgARADQHyNXkmhSIGVCjoJAXllPeJzjWUIKZoswr1YFmLJUKKqo
AICgKIFAf5eKs/+++vPK+v1l489tO3NGxx1ctRnjlG14ES8f+5D9sfu5/bO05pJ/5BqHGp767qVBDGbMbDp38A
RuGg+pKGjZN1aG4H8wo4lQZGoQcEezVmjhq6kXi4B8Sa9MwBXtCd9dTd2DZ9EpzDKuRaV0HcwdOd2xaW
Dx0OPvO/3QmU4vAPgKAMYZysi1Iu9gtxSWQWOtQlWKHaNo27TxrjNGAW8AbNgQVw6SsBhBR0RKUSHk
0OqBhE//7r8x9+hNnrNG9a/HVd+XZ0vE/+ZU/WyqhUbnxpD21Lr/wHqTQbbbpkQdesPvQBWu8SjK60mgia
cm442DFxOdFndokgoDMpXGmmhEdos5+/49/18gecJt1PZWzj26GDOVsLhdZ1ULG9sA2yegtv14dP5KnpYkl
GGMAgFM1He1msJhDNzE5a7AKDkpyTWYQsVauog8CpoLUk0sMswCBJKBt0ExMJjTzYZTolc2hibaw9rob6Y
6bj+02FVs5dEMSDS/to5/83eCz5MPv1ycf2w0Gl9Nv7XvpfhjDE490D97UvchWD4NiAFmhVGUKep/GeRp+P
Y6fQSzScEWg547tSdM+Be6s3VGiRecOw0JaXUj0SvReKJHWodkXvugW3zQureQEYoRyolLzAuZTSAptCpwU
mIhaMlPiTk0F4wwMMYaRaGOkEomKSESkpUKBrgREJK4hNUo51huX3vfN6qW/3Dx+8eK1N61Wk+49dyzv
u9G87pe8n2SphDve7F5+t9kZmTkR6i/EFZjn0b1/P54FNbuKqR2erpqN1mDT6fVcD6T3w7K1DfxU5Rw1S7n
M+cun9l0/OQwR9t14ZHTpMbJ76vwYT83pR1rA4LpqHHRRytS4BHKHWkoyZxbakKyaVmpcgotKRBKFU4+D
DIYWpJEp67TB6Zg52uAFGpFCKNFOSe05pTYSImQJZYTXvWPQquHbH+pc002e+dzVdPpXpy71v+fHPc1DRT

Fm7sGvXjvqzf/+pzQ/skCra2e/lVWmufcpqaNGl+Helc6BPp6cNCZQkTz4BbOlbu/NYhNoNuX+Tw0XbuikS/o
Hv+NcMsUg5drNgtBNR9minryxUVVWYYiaQ3AcDTaZlBnaJEpXJWWbYhFgUmJjlTBwhLwCDiBBksrVNUWP
wStGAGOMAZcYB+wSW5ZA1iS5NTZxdgEe/OW1fXNSIa3ePkeQ3/VvwlwLaAhbGJ+611fbW23a+uK91zcRH
vibzx5aCot70x9+a6vThucenv7mj21hnWX7rzcWWJvuCn/9T0O6sBSuPehccuIQBNPr7d2nGP7495mKkwb6
hAl2OqfPXq82KIN60QxqRafojPa6RHMwt0B722XiciCNDhKiVAypmSrtDnx/169PebIb41RZEAA4IywEHZpEJ
QEB0CgEkUtxroREZNAKHvNm5B95ZHzgzedHOw1HnGUGMaknK+aA/sv3Q5j0P3P/7M7XvsWvdD/y277W
ydo1NOynv/IZD6nxOxu+Bc3l9MTN9MP/5sXqJuGBy7AaFgFlNDE6PXrn6+YO65f//MsQBqcffARDtn9FN86B
qiJS0phCoFFNEbtWyy52OkjatDPKk4QstCNVwOy1YvYjnQyAhjSsQRoFUTZYqEqPGLSaCUdsPKsCDZHHxnQk
K9rtjbPbiZ+95EPVez8RB1MbPA2FoBJNoo6am+68OmHrdnfvXOL1r311eG505GY0OfEYz15qQIldsv3VUZFkx
aIWSy5Mj7uqGc07sS49nn3pf043zvUjbL7hNUc6KgArR24/EssOIBu8CcEKCCY4Q22rVeAkB1eaTpn6dpozRp
doWoCGJLGaWgo6VGpGNJkATtF4NMakilRgAUpgIGDlsZqBsKEuEgCzNL28ee9nzGt/fbqWpD66WAkiupnG
QNMLZjrG+X04SezZ7072dGK6PB3syNv+9RurSkcVX3vc+WdTI/XctT7pCQzi5nOx/+x3sn1JtlPplButX3hjvfyC
721Of0ezleN3Xuf2PQWYys5FIuot5J0jRC1iG9jgOAVG1A7uaxXdVlJ0i3yxJykaK5zZHMXEECv1u9HvRr+jMBE
JIgbQsjj1VkQEAoRtYTWqSsYYjnUTfZHOZ9QpXDaswjSGkAAAhAgS+K3H47Uv7BNjyfE/P2R1f/nIxc7mhEjXX
WEa1ctBNq5U4PN03w3J/Bpw0d2r1NVGqoW7jTvyMpprJQUQXiyOHdLmbK8zUw+NmbPdpNlApd2t+w+L
MWiwLeibaDJqoWsvJ0Wvs6+7WJh8n53LEpslNnHGOPARxGPwpGOVBkAMgABpSloiAigqGHUmKHkhFZzF
2M7aYqeYak2UuiwX0wEVURNB1Xy2b8Yn4WykyyQ//Y6b87xrdrZcEjXPt3aiGD25yiVbKDJYfxK4p4Mq3/OD
xq0hpbDIAB132SYpfupPzkjlDCbYzptwS/fIt+Ns5pINruZdotCEvEZEFdAV5F43L8m10rKEJF2Yw/lW2y5ZizVp
Qti2FgS1FvRIgIiYqGHWoEAMqMYYp8q+5lqRmhg02AbHEWyDBK2iVZZVWUCOEoVVCDAERo9OsbF0JN+B
dvHoZbjzRdebbXnZK5ddoh/4CPzgz0RwtD0yO+MrvI+ladrztyp4DcD9q7Bv9YGvmdsOFVI+g5ORQrxy6spiG
1ZuP09psXXpQUjKrJsMjKiqtWQ7SZbkaIxXW9Wks8bNGsmDNW00EQCYABrFCCKiCCKxxogRlMkIqWqAgBk
QEUeltLDjWdCqabiOBIlR47I14xJjiAwYFGBhyDIi1k++j3Gh/aY3Pn3XyWxi98FC1/WuHU7Tr57a0/cmYPQma
tM/fZaefv7e4fRLQEopmbljmpZ3fP+KT6YnjtmY5zuPn923f+7KFNQqzl38yiNglx+O09iL1oApHEo788msssyzy
aSuxv3ptDI485kJpEYQHBAa0vi/6h/EhAmAOMoOKRGkBpwTl4AFJJoFp2bioIYKAhuljKy1UrQQBckiKliCTAV
qybpzUK2/6Pbklje8tLs2FV0WlesOh5VsnCWKcbLQSr/0eT58/XWT7SoTNi0Qo/f86CdFbN6yU4bnzl6b9LY6r
QbY/NXvAhCky/qO9yQCFKNElZgLJpAkWYmtVjC7vt4e74z7s3o6mAmDuoWkjKlAErEtiAAAiqqoqKIMwFRE
FQJKyRjKiMgRNZbYaDrCRNgMZpGFPDO7JLHaRFVVBIPkEhgA7snpo7+3ffveZnbpn3R83o++giqnLkhBvc1Tp
n6syVf0vu1j6fLR3QDTZkDREepDD6fGWOlmL3zBbU8+EbQVkv2vE3Zved2dvko9KhkmRaggGGkBLibpAhmx
fhAq3p1eubCzvrm9PehzLUOS2rpFdEhStBBXLNA/Z1VkAi8Sg9SWLEMSuTI6cxKIrRdoea3G4Hq0XftODJVpw
AAplKtmuo2I0We8k8LrXs7fuHc7y9JbX3VD0alq05msn/3z//qPI4TRZKN2Sev2Zbi4mRy4VgbTV72056tR1GA
MxFYzm7ncdGU6uH4tugSw/Xkd3zKdXgZoEI0AIyPlsJQYU0KRuKpIcnTBxn493d0wKrq0OEtdmrrEEkwIbBtw
BnZJh4gyYbUIiLkjUNXIkhmfmnljpsyqhigFYPDOzGoUzM5Mp8NqzDVbdnssAkQwNM/gPPz5r91eIMxIJtXpL
3/z6aR109KJfXuX5Q3XU8lWuIYJVGNXj07N+s/Zci+hgqbC0NTw0LNfw7gtww0sM0Owtii7V88mDsU4bBgA
1Krt0jTXuTSBXpIRIBh0IQJWCqMBD4NR9mC8qsYEO4guEZfz/LKYVUAEQjQENWkAw8wGsKEoIspCHdCkMK
rceJjOfHoJzlyuQ8VNMhsmsT0PKDgtIWP8/N89FDL4+Q+EA2vLzz5uDH3Uj6av/sHbjr6qSEs7Fo5Td+rS4Mpu
fe7cBYAYOLXoZQgm4Ot+HiBbs6vHqfH7F2CrgBvftXv3Oy6GOioZADBz1OnB6jyVmbFJ0gVnU9tO8rK0eaJG
YTjjmepMELygl2ZknFiXkM7h0h5oLxOQAimJRlBAw0E8qCUIHigvyWaaWBOFbQOVStiG0IymtcYWdTo0l+m
cMblxky147b+4YfVwosEfvcE98vQcVLtxtn3T/roeVyVko9mu1ORHzSwy+F2bpiYiBiAinUC9vRO9B9VT3zliW
DY24MHzTpRUFTLqdTArbd6JvV6SORiiJ9AsZkAwJ9pYcdFKw8WEmNhWPIqy0xcIaA25nNrzaDumseLIkECM
rEpFoAionihJJEsZIXaVKsXoDQj3xVWzWEThUsxBEJQm0Y6zH/vMk74JgaW/Ud18zUnXOqnR1RrFQQ21Ma4
2Ukd54Q3XfPPzl2NVNaiKICKW4ItPnyeJF8/W65e38tK0chCJ6IRyzTtQruryii6WPZaMp8S1rT17HxcxG8+bjA
w36mPYnI2HyGNUDRQ3sb+FZkSGMe3p/LIsFOQtGwFrDRplxSYIMdgEkQobjUrNWkOlDhTXqmR7LqTifMq

dBqaEaaWv/o96YkmrgOw1d3OD9S1DOoHOnG0t7lnoPbsBhsBDrCen13emDOBnmCl5QoRI5kfeqf2nSmf7
L7ynmqU8R4aJJTUmkbSgxZK6vXK1KPvqd5u6rPMwqU0GwcQ88iBRVuCaxGlr6lNPgxgbIuxL08T5HLsZUku
QgTwmFVYizMgIRlBZiJ0VZMeRKzRR6xA54PU37VmY6mqn3UWUlCXjfhpP7BUsoM61qvDtH56i66DwX/75
w61W3kFKepqyOXDjck7ZR/8kHDhYpHnhSpAWsirEWDuGiGUr3TKaEIDlJMU0l9LiXCnzrXKx1Zqm2aQBmdJ
02J/MmnqGYVgIorNQRm1qwSlvz7hiJaLEkGUbd2m2TXaibUkWe3auq2ABPaGiCUCKAEQ1Rh9k4t2pqVRsy
ohHelTZUe7nLk52EQwRZQpJDc6YntAt15Q3/MLFtRXKwLTa7uMfT7MkzmK9NFfYXBs4WNWTB5qjKytL2DL
Lc2au1MQSAYH35cJ3e9dc4sBFaTSHdq7LpWl1sFhxc53uUrmcJJnWwpvDjZ1me72qd8Y706k0bAVChuJp2
Mi4oqFHDUqosWHjYXYFrk6oqkMquNaztMwQNNRQB2ZATYCsN1y58ZRxF9MJedQbrl/1dbJ/rk27NlTNyEv
DGAWspbww3NYDmamwbl3/nbOnJ/XYc2opr7oHpsPh8Oid/+SZyO+euXr5V39rIy0wawkYIiIEUCMyHzjDH
CEzEAqCIpYFrFCu3a6k1qpNKlgfx3NXdaOvm1sqk2ldEzogwIajqQEnohPQqCGwsCiLTrneJRmbLYmQaXuNZ
B6A/3+qRC80rMNqvgATEG92vFBOrd6+K+sXQyseLNI3nfy++WB2VX2PQKOm2JZmwkSeFg6Wd/14Pxfaf+D
oytyBttl7/auTE7e0rQAH+Kv/Cr/6U1kyB73SGIzEqoGwjUmGuYPaMVizkha9bqdYBrvUmucYfbAis2Y2rkC3o
dnVna2wPpV6KoyUENUR+oYmlUqkwMgRkFAAELHZ5I1Nroc4U5xLs9Wjmu0BUys0LI1aUNI8XVvLdzebZsZ
FgqD11Jkq42tv39s+vI/OkZ1wx0iaGVAGa2fT4Kc45lmrkTSlK+fP9WHYxs7xhTgKwZaYsn2I9r/9P12N0OLOKK
vMCFkMihFKCQwTQJdsVuTqqJMtrEi7SXLV3UmVDNQGbwFYZ9gouhpqI7nRgMpK2Rh9NAFVgxowkRRUlQ
hQZwMQo8sdrS3vWVtZ6MlWsrt9WawS6VS3+pdPHj5KhiEhC3rh8nbudffqVvf4gfPnHjpz1mQCrZTIAll0Fp
Nopl5NLYVao7I92Ez7flqHqTSFOsfQRreY0khaiWs6YDxADZqVkrAgsyOcqTGJyw21e+3VXi9tmy64OF3d2h5
M6kprAVEUlQnXM9YafVAJgl7rSpsZmBooGBEhFiQAUACACGGAl65yhunBVve249cdO3TIpqAp2lnAeAXxOl
eWJgZqrD+1dfk6n52ZVoPh+d78ElCNGSZtcdYYJQ1QkdQBUgHLrN2MfZhZmFQTSKI1mFGZiMt8Nq92kIYJh
Bw1qquNZqlGAhskddg4dC17YKnToQ4ErgS8394aT7Z3Q+MNKKooRERAsqheZokh0VCzNBidQgOIiIBorGiEf
z7CNDg4j51OVRzY0y57J27Itvq7p58fEE3w8MHlzcmwN0eYSGnTvcVicch9z10HLl49ZzOft03qdI5cp5WJJRK
uK7Zeej4RUmZOIkevWyOZehFVYxoKWM5cCc4GaanOCijbaFgrkkgQxmpc7Fibp2XP7e20F1q2rIO/Mh2Md
prZBCGwAUT950VLFFGiShMhKrElIfSIhjSqRtUQ0RChGkFl0D5eOquTcQOCnM7TYrqyCv8fxa6Md/WOTUsA
AAAASUVORK5CYII=\n"

            },

            "metadata": {},

            "execution_count": 108

        }

    ]

},

{

    "cell_type": "code",

    "source": [

        "x=image.img_to_array(test1)\n",

```
    "x=np.expand_dims(x,axis=0)\n",
    "predicted=np.argmax(Flower_model.predict(x))\n",
    "Prediction_category=['daisy','dandelion','rose','sunflower','tulip']\n",
    "Prediction_category[predicted]"
],
"metadata": {
    "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 36
    },
    "id": "eJgk-zXiY_MP",
    "outputId": "5c3eb4e6-1478-4b7d-efed-bfa00357e9ea"
},
"execution_count": 88,
"outputs": [
    {
        "output_type": "execute_result",
        "data": {
            "text/plain": [
                "'dandelion'"
            ],
            "application/vnd.google.colaboratory.intrinsic+json": {
                "type": "string"
            }
        },
```

```
            "metadata": {},
            "execution_count": 88
          }
        ]
      },
      {
        "cell_type": "code",
        "source": [
          "test2=image.load_img('/content/drive/MyDrive/Assignment Dataset/flowers/sunflower/14121915990_4b76718077_m.jpg',target_size=(76,76))\n",
          "test2"
        ],
        "metadata": {
          "colab": {
            "base_uri": "https://localhost:8080/",
            "height": 93
          },
          "id": "NlEBeE4plthD",
          "outputId": "9ce72baf-ee2f-4987-8333-30e0b6bb8e76"
        },
        "execution_count": 103,
        "outputs": [
          {
            "output_type": "execute_result",
            "data": {
              "text/plain": [
```

"<PIL.Image.Image image mode=RGB size=76x76 at 0x7F96081F8550>"

],

"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAEwAAABMCAIAAABI9cZ8AAA0DUlEQVR4nGW7Z5RdV5UuOudaa6eT
Q506lUulLFnBsiRLcpJxwNjGYBu7TbQxwZfGBpP6PuIl2DT0A25D00DDI5nQGAecDRjnHGQFK+dS5aqTw857
rfl+nJLs+94aZ9TYtccZo/bcM3zf/OYsvOmuXYBIRAAMEQEjAIWgEUhEJAnIWESKMQaoGAESIHLoHEZSSsYYA
DCJAIyYRKSQhA4BI1MxIaHCWIIHPIzNma1lrmibKa/ZbGcMXYZdATshKUXCM8DiYMjAZ4wREWMsVIRIAIqI
AwABiyjSkDECCQQARNR5is4FASCAIgIABpxAIhIAMAJGAESEiAwJQTFAJAYUIjEkhhwYksGZhiCQcWScc0TqfL
gCDRiTJAhJAGiEnBHoBjiRFmda5FHl2uRjI0YUUiwXDNrxmSR3v73wu6ilA9UMpUKD37b26qsTL6KvGZ4GQl
OoiFEEqDHkgIwYY9CxliMBI4WKAyEiACAiICIiB9SQdQ5njIESCBxQQMcozhARATiQQMaBc+CIGoACiDjnnZcB
UiEBACilGCiOxEApBGKoEIAzBoQgmSSd+SFmh7Xq9dlv/3L5Vi8u38vO/tma4TqXcSX6Yna764Efbvon4S4wt
Op6NqWmj+5ubIxz0FHqJBOgUgw0FQGqTnwRSQBARhwZABAC78QOY4goABFRMZQIiMgZ44CaYAyRM0C
lMzKYAOIMGAPgovPExBAYcc4ZE6EW6FxIiAvug4a6amwWb4iQX564vw3CVIjY+JfcDVtrv/W53u+nroJfB2E8
DuaEzx6qfesL+488NH0dxvWfTz7+3dzSM5JPX5X4ktrTahw8+NllNxW5+9HCB/wg3cIhJ0jeNPzxz7W3uFGfG
2ioZU6aQRw4BxSEHBAJOn5lBByQETDGBDINmc64DqARCY4EAIiKUDGlmGJ48gAoAIVIjEEnzTig5Yc5ZcfAvV
T9WAs1TrF384/kUqVkePA6657AoCWGFqcj8Xw771kf7L3wuL/hNDp0uf6Tzw/fdGbXgz26k9DhkeqvPz5y0
YnK8IXwuaOVMypjRnMcSnXtiwNn75iS252fvD/5u9sXLSgdefr7+vNkTEuOMeEjEQdiiAyRM8YQGRMde4AR
InEGgiOQ5Aw4AwRFCMBQ4sksZUiMEAlvuXe3ZICISICKOOekVNLTVEyN62KYu5c1L1IFjU984NXcFaN17dbk
+RT3Zto/T6vP/Lc3/sn82rg1OW1vqpmrRpx7jgY9TwXPtGJlzrzIT21IHrhwydUPHb+n392xsvC0v/u1dI7Gy4FJ
ckf3c+fo77RnG8ksCj3WbEZ1PC0/vF9ELu/Tvfr3/uv4RQEoBggnq0anNCqliIix+Zs4X32AiBQyCcTozZrEGQAAf
vq+vUSksFMs58OaBS5hYo14/KL+7T848Zn32mt7EmaJ7DHzro3GR2Tdj6VCu2a0uJHzuMy3vabKxeVoOd6u
yud6/2anMp9M3PHC7Nbt5mAok1sS92w48e34wqH6rEzSoTiIJqx72PjoVnHnS/La08a+VMOWFdHmrUA6H
/Plb6aOhaqCLIaICuhNM+bDsIMFOF973mKSApyvrm/5JiLizfftETDvYg4IAJoQSsbWapN724Prsr/YGn2zEfwp
anw4dCLDkvX025PeY/E21wO9qrvpVHzioA9a1DMMB48mNc1btOHqfOzPmbhVGYdyEHIvehD+Nd9nrtp7l
82mZ98oJ7R0JG3VjGXPMwaD0W8v8j7xmsmWwtIi/KH82l5NEzGl1YQ0BSeQCB1nKqUYY0qpzqN3EurUzY
5JCggYMkA6eRhDIsJb7983j3kd8FMEAC2reUkMT997TqFHHuZkjLFYId2abkauDHUMQ0LOlS2ZryGJVttNdo
FdA7cIA5GeXRfMTMLj/e4Xk9bs1JKuMw7HLGPnw/79rw+6e/TeLntmKuTpvN1sLSrktMT41gv6fvbKgcLVhz
e731yp/uRZMuC6hkFqCcS9nl+V7psMshKgA5nzj8pYJ2gBADk75cmOV+etBejYCQAC2Ulfqw6MIiKmK309Y6
c7GNXbViHrtlGF5bpdZZEHnOlVP0jlrLKfTsmW47ka6SwklmLJsl/OmMPji5vuvutGrXuXPd8T/PuJf4w/O3VGS
nt1YdZ9XlUst1AKYqPHG/lC3pvje4+md0+UeoaSF+xfT0ZzXx3iupEpAk9qh7eHg8uzh/1lGZiVyBQDmgdF7Piz
40xkANAJYJCR6mBex0BEZEoBZ/jZh+Y9qbGYH7U0FYCwJSbRNFeWtp0VfliHmmAR2jRTtmzHDTwYd2DX5J
KZ8YZgaQNUxa4vzMOW9XYu5yXjYLLY6AmH54H1QyFIBZsuW169s2Qndz3ZeuylYcNSjQY/0miHUuVjwhC
mL2QBtDNW1Yey9WTayHb5uQGIZUEH8JrgGtnH8IWypXM/Io1LKYGzDs3qmEFvZinAKQ4EgPMlhiGCeNPF
VGdkZUx7qTXbzXYos8GsZMW6acHrvzgxqbEWtH039LS7t2XHai4z/EzIDK2FnAcQvVIxjj6WXVwsbVoexWJO
sccUaW9339MtPTp/4iKPg2W0lmzpefCNuuMk7Yg4GK7B7CioqeDspb3KEAdORF1WfcFwmO9msYTWrvntl
mkNeIVcM91wQhWWecqAiHOuEDjnp2xjJ6kPEcHJa8T5ACVGRCQ6TgcAkrHIbA2pynm1W4Nay877mZDV
pHRYz2DvzJSX1FvhndtygRIsaqV1PBKS05IesiFDGRBme0wzuXKqtvu0WDyebAsZW9k+//R8/EAIc2VIOpjpm
7niUv23v0bHUTEjOxs43bFEvd38+wtHewc1y7GYZVQ8aM36WjUQJJimwgoEgXZF8VP7vH/5e7SRowQAItX
BlE75YScTlWieAxERAgECAmiIBCCAzddZ3dcSKn82vsNjVd4l61MwVtUtiJQzIziE0v/Tzsxc0NVqTgPTxpUXT1l
WSnLGbMc0I5ho2BOTNctIbV7p8hCMjLcoBl6M5RB0C7oHtPFtwXmrgg88Ga5bn/S0NpP+tCtilmVzEfe0wk

AjwcFtBV4znh2x9fTSnT33c7bTj7SStw4JTW6HIJgkDlxxFEoyxpVSSpxMUQUdSoMIyAgRGaBiqBQxjtj52LmW
qfv/zV/8iXnwcfGHYgF6hrRULmGmwcxAti8s1Z1aaywXz9YiRlwPmdYOzbla1A41V2rlti+1TKAVmlMBkyBtnl0
AkHRFqMcSH3vtjWDju1h+wKq9mLr+UrMmK6aVdALPCaOEoXteYtWCViUM9JAVM07CMdsT+1a+snSr+lzK
jJkUkvCBuGDIODAkINkBScYYB9QZ1xkXnHc4kNAYR8YYA4YnORyHThLnAuGhHupRVlf74JzutV/uMu1GtVW
dNg7sgH3brLG53HCfODgb+C4rtf3RSmW6OVtxXea0u2Ke5WNDekF9rtjPJOo9veHsUYgm1GydTfkPrl4Hx59
TMwes1LmtW75RKlCBSduQIYmYEanVZ+tLPbFwQGgjKswKAL5iGBatBVmd3sqvYICGRMZAADLGmEANQU
PgDAzOLIEaJ8YU5/PsVgOhMQ4AkqjTSwmdoVQEDANdL3jh1dpGp9rfnT8y/UK6FICIQ5b40JCR69JfGff+vkN
1JcvMymu8rbwgme9lLGrX2hAaMgaR7V9/eaq4tBU0eT2AqSlYduFv99FD70jcPflG3naqZ15UdR3QEBYVSu
ON/lSh2jXcPifXzGZmjcXXYP3OiEPv+ssW6A+4CZgehcGl1i9mRiNWZ4CC5nEQGArGpZSMMyClEDlwrkgpxRk
Q4xIUACChJilCBQD4rSeOEpFSEHHBVPM9+Ig29yU/EDpETDfbs4QyyGStWtU5diLx+x2psmMxLi0pg0a13OC
pWExLGOOVuViYO3f95A2be9ZeIXlv5HOPO1fe8eLUeYlnKgfTzeJnDg588talxfJcrH4US0e8HcfkpLb5PHq5Gj
M1ECrpLCioVliIxxuvLdvO+aAMAyWCJrUtmu8iOLJASQAQgIQUkRLIQAEwlERKKeTIEEESECOGqEgCIaJAJER
AJAWYguSfzWsKi64+o/GD4MB/82Q7WYS0eZqIHQrqsHw1resWZSyB7FJKKVnwhNtj9c8cYdo+/GDf3KKiaa
btn+2+ONpZuXT5xPY9s2ctqzEFuzbt8MISZ3J82uovmPZEPdFlLvKdnlSvamppx4McaE3x+toTK6cuhUp5zdH
Nzy49ZrJSxERccS5EBxgVkcZ5B+mlQgKUhArlPDwyBMUIiRgH1fE5ISEA4L8+dYRzjqCk1uJev8adEf8VRwLHxK
SxSElxYfDpZPORjK6rrqDAEnf+1XgSI6UluKulE9z1bVcwLWK39EwzJs3+8//ztfbYQfGVr1weOv+xEup/a91xOL
622jLeJ78cTd3tB3o8oWw7dJs6mqHrki4MiK2Zc15bGIMHVkwMu8dGjp+vivmd+W25SDUSkYj0DtYrCQCgE
BRQFCpiyAEVqjfpG3AiAkadKtP5PhEJYBoTXJeBUlnQHUnqsLlB+ORgTSNzk/lAzHhkUZo3BoJf3ddztKZLEpwF
hi8CEmMVVFLFg8zq5IG9yYuNdGyqJHbthq9+8Yz/+q/HzaU3dS35gBQpUQviIjPXuKuoAGN+fRaySbNGXjILC
UzrWmNCLDDYlIFzptJKend92WFdTffVH57LXxEEioEiIgUQIY+iSCAjpRSCQskIGUcZAesAPwIggoIwksTwVM+
FP39yV5hMqwAUMiKKlCQiphoiEGemduXtDy1drn3/D8XdtnvVxvaSQTNv2QdejMZ9/vRhOPecZY8/V82ni5
6fiKnmwKpFM8eOOg0emmyi1Fx37jfNngtJgXAPzL7xp0zvGfri699+wArNFHKnXtYziWgu80mt/iMntub1vgf
eVlrwQr7EWV2hoUkKucuIKUowdFWHl3e8xpAQQFGn3iJnKpJECAAhqQ7RQeQRqY7kRUQitOIUKER+shntZ
HkXWrG9wZJa/PRkSSw8zb+aTs+tgvaJq+TUnee/HYK4OuNV6i0edJMzl7cXDJzpv/ef129eND4+blx17rZG14+
fe6O2/d6vxvNf05Ji9dnf7brkf4Y2Oqx97+rm1tFtWXlBlIxkqI3iGSvcGDcujlAdymxXvGqqeIABCmaQZqMPzN
Ui3mEsIVMISFJx6IhXQESRJAlA1KHvoDo9FxEgRkpyzpVSjEijjrc5WRTTSeiRT55ZnOXTRimETH166NzC6Xo29
5PJ9v7Si/tP8DZdGMS2ZDkP57rWNVf0bbUPu4vTOeuJZ/2RRfHvPLhwx98/NbHz0d7+VGrpBWdd84yeWU+
+pqMuIrX5wIKdfUWXay+PbGtzNZKakmAf7P2oqWjWMCzUFPNRYSQpkMikwSKQTCmQhIohCkKNcRIMAB
SQYkhEoAhQMaQOB0BERGKIjLFISkUk1MmuVCgl0fSMKrZ6IDZhY/7a6upd5v9zNDZyb+0VR/Vu0O8dsRpT
KLe9cU3f3O1BQ3qZ2VQGhILf3bFxbmyn0Wvc8r5t6xbenF59HN7YdaTnl025G92XE8lif9Tsjzdiuo+p25LlH5Y
Xb1dttaPnKcvtHz/jXamAW8oLdE1JIkAJhMgBFZw8b/YZJw8x7OiHRESEDBABAanTSQNDUqSIOl0Y/vTlE52
WJZz9hzfzZGLJBy+Kbgmw+kj40DvZudxl2ULsH9s33LdzfPSwdf8X9pRHwezlyZRgvml7broQfPbOvvPPu/xty+
9JebVHHk6efVFrdAbChd9XYWAq9LOG3ggwaQgVY8JMR7Aj6h0tnwnxcRVZyBkoEsgYV5I4KlIcpZQ68ojm2+
J5EQDmFQ0pJedcSehQVqVUR6EUyCTIDpaQQqWUUtT5Dv7HK8eUUpLC49u+YZRez3A7kZVnZxqiy0+HyQ
D9OOcq7TrB2seOhmfFDlf9cMTQuR7INgQE+WFtfGyp2dPOhm66MFerwkRJ//MLG93lBdHoY0I2qIrtSnpwY
OXKq6XqaUK2LTQmfE3mFNiIyBUwxiRIQgaKFBBjTEXz7PRN1yEAMSnn7883kzRv+fxPIsaYVEpKeTICUCklMF
IcLcTs8tP+u9H8g3v40+uGz5f1h3PAZiteOqGVXEfOQRTuzqjNk15iSNZCRrrOQpflZBaEgyM1k0+FoU7cnJz2PI
OeeHnqk8NHPvaRBjDP8OGhYy9u17pngXHGiXtaZHAlOGtFIDr1IwJJIIhCQMYISSrkHS1qPhoREehNw+Y1gU
7yndI7EDrqjlIKkc23j6QAAT/xrWLcXO35u6zsldbir9kvfvfsK265aJE2+fBSLUl+BCmNNxvS8w3J/LnkWT3lF01L
OBVlRGp4rdaOQp1Eq85Dw2/MgRGH09/FJvbzAMJv/uG6y2685LT4q7no6ROtsx9sf1OmZuNRnsG87iSJCBl
XoHDeGEDsCOeMgYrmMZ1OerSjKb6p6LzF1W+di8y/CwQiUoqICH/w5HQLpw25YHrXHf2r3qt2vTOeSJyo
mZf0PH72CjxwKEoIM/Q86fOW4kkKbD09qBq1hKgcilaug0qba0pKBXMlLlAmMkLwjBCVIKA5EReu3TsEh40

n9ohijetapCNSR+BmjAVKEqFgvPP6iVAqxTkHVB0nKclOKq5ARJ20nD9q/j4T8+D3ZmVSxBhXQB0Vj4hEk7kk
M8wI7alGovjr07esOOu0A5F8BmrpxomGpUEqrh2Y9bJSo1y8PdNKXXi+vfMBPUxnMpWpEvQU5eyO/kl/Ek
KlpaAdirzVmmqQFYPuYjg38q2XeHyUFYMoHWORilSkKUBBRBQBogCiKFKAyIlFQIwDEQFypQgUAahTAXnK
XR0sYOykACcVwbwkpwiIKAQCJRkASaU63PXKj44YJpMAmtJ8vZKPnE9d4Y0MGDNjOrgEGPpVt14i34Z8UTt
qZ165z7FllxMapdKsHrdXD8SQ0T99POvsHHNzRtP2IbHF7fqnS7Z8NpHsAqPhaiGE8PLUHc/ObHVEWyeLiIBY
p050hlX8TbAAIuqojEj/x81TccvmkxHnA5wgBIWKkIBOvhFJJ4dTwAAAb/7ahcX8wLGZ15WX8dQEWFEXDn3
3Kvvw6DFotkBpkU3tpjWlDOdoc6ySPD4alSNLQlDzgxwAxvhgDEOs62cNfv66JfvufHLZcuDduWy/F5fe2KT13
A7lX7C3bFngox2bjvsZ6BAUoreWz46piKzzYEQSOmz7pI2nSijSfDYq6ITovEkMYF4iJ+jcUScnCfiRL51Wclt5y2
xM1jlwYC2K58lu+swK7Ol/WpJdm61ZWnpupn7X3zKK6bWG5jAPmHZ8ptLdlXc8Py5UgqlMXJ61RTv3bWm
ojXUPGVMTfrYfjKyYtqOHHuiZ6f5gz5b3RV6/1JQdm005fR6vCohzjBS3hdQAmM0xSYRe3Ek5FDAh8a3GnBp
7dPxJJ4ciApAQOqHQSUJ4C3mYV18XLvq8qtP3bv7hteec/qVrFg/mlGoqGWTjASvmB44bCxfd8OCJvktefik3
1dZOVPxppzHd8imMRnqyTGEoleKmJ7nr6G9sbxx7amz2qHn0JT+qgwIzncUVK+CWW4PbLvv+QKqBMaM
WD+JOIdRbmkyQ7rS5K2WBR7yBKaX7EemNRMhtHRVXQJKUJHWqfnR+lUTqLfZEQPLkUOD/A62dN0II+LtH
7iNv2879s1efcXzzqlfsSuCoUHeJAqQA9CrNQP7hHWLH32ZPNLqEqXlB6AUeC8PuYu/0TKmtpDDMpGHkWJ
prU4MLw8tXhss2wOBqZYJ5YMx7ZW/sYMvp3vhyTE8cjfrXWY+/6C9OqjyxOmHPQOzA+dVP/FK8eIN+86Pi
B2Xi/5o68+uzz+hocyOj1BFb6zdDipQCAD6fhygZoMJTQNpBmo4vO46NYH4WhEgKSJQOv3jFpt+usut8JjzSB
MholkmynWiU224bojrYrVq/hB3SQs6aXlRqB2nL5Lp+ZGrKMJOB51MkbeXEjVmHdW3ujRYvVJVDADVWlR
6SudxytuSSjwbtGewO+Ow23KxB3RAl3yzEmsGN7fNG7ZxmTibVH9cObTmrectYcJ2GWpBQi6Z2TBmna5lp
ovS8Xg5cgoL5MQF08B4B3wKbSADyTdDsdCfIfnf/T37f3v9U974HFpZfwqeOHkzNHexRrL1ocT6VFLMuVFx1
hK12FIZhEMpACFG2/cCIuRhrhZIxYTBIxQxmLBRUPq/X270Hsgv4sku13uUwsNjvKkIYa11pXnoZvzmHmukcY
awvdfiPnz90+mLt0SZndTn8Tu1/7xP/uWDvp7J5+G3jGwN47DsL1p+T+5hfrKtQP1VdJRAhSJhXyhXC/y8+Wa
RAslO9yHzniV9/+MW2PhQYMyk/yYFveXJxvBcqrgFNxJyXDmC/Cx84E+58ZuBXfywFuu574BGXpJCYr5wEJJJ
xMCkaHmoOFrVZK37zmjpBrKWc+ydgvPSuAfXIbf+Rf9/H08tX03uWHhk+beirv4+68sUrh3YkeTxlepMqflqP
qxq8ewHeXNl5sXrhbfqts6zrEff+aTIBLUUeY4JOCTkAESkOJ0d0+JYeRRFjrGNYh/GwDq3/l38c9QLVZ42e//zF
pR6B1chtgJ0Bc8H11dFvbimMnHa2OfUGfuiHelFG48etlgiEYqECYExiM64yCRMXDrf/6wsY9bDXX3W1SKKy
qhD0xuVsAF/+TS4zbFFoJAajqM0MUxOWSFSCVRvVWcnphYas+4psmRmAe55Pr13qDjN1aPgLFtccN/Y0/yf
0WWSFyJkkYMROUYKOYerkWJWITpEEBQSKJCnFTtLFnz/y52W7fvys3p9e9RnynIP55THXakSjC+wj15s3Wj
nbbuv3P5g8dyTcfyKgUPzwKQ19NwiZqaGM+HCvLBbMod7oxg+52f4gjBuyzv74x/6+niNOCMOWCDT1jb9e
XFy4YqCYX2KMZxL19OxdCWCzllqQMtPpMPDXdRvbn6596fT8tzm7rDX5JBdebfFH/mvneqtYzGe6WaCBm
Qi4iSiICBWxDmzgya0d7GAsKhV1QjRUJ+eTjAMAfvavhyee+kw6lUmd+W2ORJGNMmFr4uOlxSLwWlu+Env
i24l+yDOYrMLEfs1lVMxFLS+VSzntNivmg0QXONOgFY2u1X68D7gPQPDtn7+zZ+26nn23uYaVTbpPqA/mu9
8jGnz/gfdbl41u2t+f8f3DL+m7j2SqdmCp/Of/Dee2H4nFIX/6h/7B/mdWj3H10hv+WaaUjs4QUcJboE8RsflsZ
KROLnZ0uk2KSHUYVQd1iAj/1xOzVe94Juw7/o/zfQF9a25OZN+zkX5YOPGzl7wbt2bv610+2zdsHT3KZrcFH
MO1lxmR4bdfsbSk6yqmlJkvOLUmE4LZc1FyAU8V4irlxyP/xp9sCVPLFhfOXpn8HoOlezNfjHljk8//81Xrmkqjv
ip841eLj85Ou2EiloLQK/70ttSaBc8DF1Ut8uzcuHfBI9pPBKBizTd9ooiIOis7CMABGSoClJ0aK5UEQuCnatV8q/
2FRw8qAELQyJt69ZthbbfdaH3n8pk/7bk413rqnWdHPcuZPa7VJuXMlNp0KcQGEISUDvMahsndxqzZNQCO
F0zuY74vpIv5LcRmpRkX/9o8QOATKCQzV9s52FfwZ79SLPaumLl/9Lj75OPpJ/clKhABWUz5piG+f82Um0hnj
WqiG3SMxU16NP7LAz1nGy3ZEW2kUgIQACQC0PwcVgPWoW8KgUhKBZ29iI4m0ClUQhISAikimcpt/sHRey
7y9OjOp1J7mtt+8s/EK3BiN3z9j4Of3HpscJWKFcHxwZ6E7IBKLnCbhywt5ToehASzJZXORjN1qLxkFIQaY4ugF
0hxZAwwmsyvWpE9uGni0L5Xnh4jaE0nHz+hV7HthYm01orHYbrqDKyQXf22jSCSG+7w7mkSMlnTPZjvIlWn
dSFEVIo440oRAPqkEFGRIkVIDHCe0c5vgHQuFSEpBGI2khCiTNgVFA/JZFdKb8+Yx2fjt72a6u2dBCEK3SJwQP
O0oAzBTLG2J3d0nxs4UJkBJLCK5vPbgZgpPbVn1Hx++G5GPqqAqQik0xcc3vP3L/5wZ2rdcjDTZmGoJVmoQI

YQRdzWIjIM040yTISTe6w3dm87/9Cai8SfPB5DkNAZjyMCQ+wAIGehIgkoAQmEVNDZ+wCYR8hT9KBzRKT4
IO1dKMeoNKdxO0j27SsfX9Bz5nff+ditv2GMJ3VhUKpRa1RHxyFV0aSXSnT7TXv23TfAlz8Y0xLO+A5+2gUqR
d7/fmDgs9dO9BssiNyoTsXqtrHZ8YrVf9bGZcGO70xvfOrd5U2zZS7IYyZMj7b1QgaBB2FxXJ24+I6ZFx8esj0VJ
VyOprIdUd7HYh9W6CGqToePiAQkO9oUERBTQEgKALCzCQkSOxtJwABAyvl2VGQs64jTVxOnvfbcuWY8ocj
OFQp7X97znkPBslTXq9vncisSbjWHq6r/fs/gDReMf/se798/lch2hdW4fvcLeFMGfE1e8D9S528IrLj85eNDq4
bpxX1leffWXCKBCb23O71/Fl/aHqyZvrR3xQQAzrYhhtBbxIYjLG5n4sEVn7ztku09f7lozzErtvrgh/LO9tdO21P
XzEhrxb24qzmn6DYRMSaklDnGqyoCEMAIOhqHAjq5cNbZZEHECEiRwk//9ZDw5d6HP0QwB55CzRTVTMS8
KFNhPmBjoKtXJpIjZ6T+9uTsae7e1/Y1IR3mNq33nj5ihe3glx+N9oyrnz9pApjtSAqGSoamML2ICS7Xn7n2Cze
+7fvPFXpWX7Z59Lzduw8fqa751BVvuCWYO5Z48XihifrikTMyy4q9zvYFfc8+lNyRsLx+SdtYn6JAAmeMvXXnC
k72lxrJGKdaBIAaQQgACBr8n2IPAHQGePj5xydf/v07jWhWMqkk5mhoZqLUNaKDBj2FHuXFZo7VVBJGupKk
2FTN2/PaoQh0JZoxDDSu++B9eIv49fO9CsoxhIhpJ5t4/p8/vm1u7KVXk/9XLs5dU2hTe19//Ie58pOfvUq0Rl
WYDg8eG/TdWl+XpvV4sWbkQRjvgxPF34XBRKXqNftuVLqPxEnxtyognesQoi5AItkmjJiQ1JFD5uW8U6Z26C
5e/un3B1PPOPEIGsrKZZ3DbNNFG4YWDb7yyj/8qGRFiWY7J2WoqXJ6aFk2nkfRqk35/3j4eSMX12RL6lDIrvK
9saYTMsWBQ8w0OhTsHRefLy+4bfqZuw4/8ts5s/GuLz1hlJp96kfL2r9L6kxrwGFHgtuVNFuN7MZkNkB14Qh
8Z8roffTQhp7UQHPJB5Q1oCjqTDE6CuopLyFiACpGMsOgCiIgBqBAwqkttI6dvKPHf+Vnt/bluwQqrmuHdh99
8fmn8v0JYSKjgDHVatelLxJaT7Nay/b2aXosmyrG00opNTflP/63Z3RdZ0wBKj8MNU3r/IG4ZcmI4qnkhnPOjA
y+Y2el5TzT7a3Qi94ty48Z6SCf9hp11rR51GCJrO8zmD1uivM+p2SMy99YXTcxro1F+Wp8Y6jSiP5b+WrHTo6J
LlGZE/1GWEkzZYfQiBQgn2+fGXaYA3QmIGuXLLJbjm3bjz70aHm2DHo4U55koEJXCz1029yIBSsW6om4VW
mOSnBtv9s9kjtt9YIly7P57st27Tywb/cRTdOIMAxDYKgLzQ0DUMgrdvfCa0XfulL7h7WdB7a8e/Oq1k/LNfM
dZ+ILLySstPTBZQAz8cV5MZ7esqkeOZyTZl7dihq6IwpJNQXcoFCebKdOLQoyxmKyWQPDdMY8EZuzA4OxDK
o6MQkgOJCah5DOBXOb6rlXn/vL7+86cnjWadZcJ6LQC1WomdLMUL7fyxf00txUAJEbOX0963bv39+15pptjz
43XalZFp2+fkFhoMsUfHCk7yv/9rnPf+2ma993jeN4UQi2ru765ZdEb37k0psTQ2seeOj+PaXiPUeKv/6zq1sJy
d0YS9cSMJV7d72qqDo13Wy2Q6fuq9np0ly7GjYaPLCVboQQRUAMydVbnsU9Fl8dG62E5uDM45GMI5KruJ
ewPPRMLjUOSPDmmlZnSfbqG1cePnTs4x/+xV8euvV4zcwagQlxx2kvWbzw+IlRZrB0PCYDoLiKWVCdq/nNV
Fdh8LQP/iiRG37+l7eNb/vDVR99z9SxN4ZGTp+YmeztLWqGXiu3ZqfryXQqaDsvvD6+rNgzZXZv/cg3tKjJ46lZ
wxrc9jPq2bDtoTsyxuw7Lr5cg7FIM1OJohO6jUajUqvGYjHBtPvve3DgrFvzZ1xJqDSQZxz/0fHBz05r/lVjl/T04U
Ox7825Z0Ui87Uw/xN1zGVhXAUBC220gJgE4oDzWPLlx6bscHbHz9+lx3Ho7E/uePDH/gxTRmAaUiEI0gvZtO
M4dWxk9HQkW3YLrrzyMi2Wak2gsWz9XPc7dv/b1vzKhW653tfXx00FUmWz2VbT8V0hQcZ1yFz+zTk7XRS
ejemUL7WEbLK6priSnGOi4ZfNg38S8sRIrisUPJIyiqLZ0tyTzz0jiFUi/YrPPw6MGPBPwFkLe/DQiXMc2inUkd8
mKleL9xVaT26fTWxbfMCwZ+um3uOiy6HNhAJigIwQAIQbueLQNtVwQ4MfefIHEEqMC4xC30NN04LQdhw
DmWCO1nQd4OjYwaqeYrk2MVo+0Rzfkyjc/f6PXAq2drQ0RiCsWBI15rpuMpmslKcAIPC4+5evG+lc4oJbLD3
rWU7b8yVpEQIDCFTL0ExYc0M5ijWPPrHzof8EmnOjnK6XoyjHIcgBs3WVgWRDNmLt2RkbUvLenEiMmed+
NCpATZU527vgYOiZq/HZM8Xdd9Ev4tTwMR8ScsAIFCLiVe/bUKu2G0FTT0YsZKGNUiEgkYIwDJOJhMZYqVo
1zJgVyxmmnkin49xbvWaxnsorCft2HahUp9acsfHo4SMbN58ei2uOLZutlm27szP1MCBEppSUgvWaPLf2Oh
o5JwzDsNPUy/ndRZMJN/J0Sy+3/YJmvHbfTXz6oMux2m4yu/sjn77qeOqLBh348MR5yS4WhWjF3ZcbH07Y
vxd1Hi3bvJk/e6DBUjb/Q9cjdmx5GPoWC13QEDvzXMT33/ze8t7t62756ZO/+4Qcrw5dcr3VbA+ccd3Ov12/
YcO7WBA0mUGthhfM7T1QzqRz3cWubI6WjSxuep7vhaalc859Lxw9Plmam0smYxs2nhFIxRHqjfKxY3MqM
oh7UmnxmBF6/sDyjdbyD6NozfkpJA+VFMJVkABgGgsZY0FEEYFC5R56euKxb1x75arHHtnxrY+M1FXviLivfpx
FSXPSfke99sBgnDJxFWiGBr4XQX7pItnVVW0u2zn9jnG1csYwHcjGqQUAePGli/N9F+49uL3enH737fdxzofM
wuVD73n/TWVNBroWP/PizXMNXGgGbcY8LzxyaCIK+eJF2Q2b1zb9gCR4LZtARqG68867zzv3fNdrb9lyFmP
oeQ5yGB+fPDFWBRXkutPZbLpLiMbg5ZQ+PSI9hJamkLuslRHgh5pCpukSPMEYUxrKZjlkV5h379q5/aIr3zb3
wK1x4AmdIg+m/aIuprm1uqe5GzKxRB+WpmzJuG2e/lDvXww3nYH9bSNtYxrRBQC8/Qcffvn1g6s2Lnz63ic9

LgmLhlNxrFzkOXa1pKvgU+9Kd6P736ObSs2JdDo7N9P2fZ8xZqCZKxi2HUoGsYS+YeOyQk+G62xytHL44JEVK
5fFYjFFFEZBpisOGHnNwLLiEQaNuSjQPTjjx6gaymla8hBpw1h7pUs7VMp/gFg+UNwQvq+CZHzh4Mxdi5ufy
Nak0wif35fauZeCeFd5qqVb6l8urPq9zMLkrF605yYW9vnd3bLaBhPgoYHnSepNno84ERGedV7/jMMTsSgIQ
9eRP/1632e+1UoL6/L1hwNlvXhsgc69UiR4gAMj/XOTpXatHZFkyAmZDJAJhoyErme6Mrl8MpnkwAPTjC1Zv
FLXxcTEGHFIWDFE0jTuB26opKUbxFmlGajTb4+zsOnaIDSOhKS4LYKs9P7+xQo+UyxeuzpfWlZAwi0/+sHnvv
N/J7/3sfSBOccjpkgXZvWKlamrL5mbPZ6ejTdMDkkdWoGlGcHowO2H8tekZOggBkwAAG685NL3/a9fUGD/
+vbr6zOHXvu1X0hFbhQas3DZbQNK1ydmWoKCNZsua7Unpscmmk5gaiZFHfrEFZBpmn4YCl0rFLsSidjwgtz
w0qLyqF1voCZA4/lUxvFcpVQYhvGEUa3UCb1splBLvL1ZWOYFYZKZyCSQQJXVYqNhOOLuvltp2L3+3TGYeW
99U3jESW0yzj/PkhkBKJIW+UG7L5fZcO1K3PCrJbjLi2p7tQ8BHI4H3TPa0WSYlkwmAy8w4kQkFg+kq5GkSL/
6y3dPv/ZEI86h9pkXnsXb/6G6YvkwaEVcOSF7+eXHhnq6dUZnnrO5Vase2n88CEMG0jITMoKkmdCTuuu6X
OeTk41sMpVM45KFg4qxsbm5mm1bwhBCKRUFrkwmYpaVQ0bp8iNgWolkt4N6UjdVa9o348qJR1EtlozlBv
oLozdvzf59werg1THNfrkQGg1SgkQUeL60Yw46a5N9K3csO+oEQwvklsxHnMrCvw38IhENKSE01RLIQgojbo
hmu+Q6EVKg5Qbfe/7AxLNX/vyN6MnD3XarMtQba7ttZFrMZAnTajpuO/DLz7yMLMzn82GgfM9SCJoFUoT
ADE3TCGHJiqG5UiUW65lrOVzXNM3yW34rsjVkAclEzATggYLA95PJBJT+PJv6Kuea64dPfvWDnmowNLZe9+
myfSINpWN7x2/8UiOqZ1JFt7x3kll52wt00BXXVEwFIvWyk82yZFifK/uwBFSq69AHmudP+ovuzT7tW4lueVT
X+1yOotWcCgOGemDXZqp88j+eioGNCTNszSVmalPcSPhePZfOWEKfnJsRgus6DxWbKc1m0prUXVLK1IsGt4
LA50zXGXOarXTOOj5WWbW0R8pAF4IlQAgBigjQ8+1ms53L5Vo11zUdk6dGZn/TzFyUgOZ1H+/pX7AxUIW
wufvo6ON2Pb1801mH/vra9FyDDG28ojEGnHMvUCyuFU0cOvvK1dO/aqKkXGF2tD43EWoREAM9S2tSDxx
QVzW1vBU4qGkiRpLihGHSef3+7zzyNczERcDLZT+dyUkZ2o5cNjLoed5cqYKC+TIizhUFyFjLJgRkjHu+O9eYM7
RcEESWUamUt8eshNMWlfKiZCye7Upn8gkv8P0ojBlmK/DysZTn+kxoJ45Na1oTR4+NLB0rOdoLr4QcHlo+/P
bQHdv9avdUo/T+zXcW1xq1lr0wnliQ9PTAQzPpBgFiM5unrYkfmQHs86BHOfkcRwNymfSBni/vEv/cRWXFW
j7k46VjfncSP3zDunK0ujp3r8cTSzatP/z6c1DLexEvZsyWY49NOiuXZ0ePVzzyERlJ4EzTNCIiux0wzpUC09SlCk1
T78oVAEOvLQNfNpvtm259/8T4XLPezKYsiTAw1MsQhGB2O5BM6ZawhO77frVZhpB87oU13wrIgXb34nO4
RZZX+I9//95XP1w5Z8R0KqYt660qe2ib3tUV+K3huHF8zTAIFjO4dyBUov9qlrs4t+MbdiJYnKptOb/v3tJPduKC
QrM+ne3GS96zBCCmwqmudRf3rPnO/f+2UmtmWFLvSliNZuW0VcOvvn5YAg72Fw8fn+PIEMkwCJDnU4Vq
tSw0qtXDWCIuw7C3rxj4NS90dL48ovGhrvzK8zaaku88cihvJP0gWL5iIQq5e+fhXLaLCxxcUGzabQUyaHmRCl
975fnh4WXb926/6cbPl2cPxMfv1PLw+zty//OGmqlRT1cGnLYvsdYMNQm2BwsWs3rZqDV9x1SDCg/7rHbR
0bYUaJYUJCMGxMKB2tyJRI75rYaZW7Hg4zuSa75GsTaqrE+SofS8mmYak6XAC4VUxuhEjQEqpRiSIB6G0XRl
NiIwYlZvbw9jZFi647QtwxSAFTW1buuHVv/zXbteq5WapY0Lh5igeDz1xq7D9bpactpwcTjXPzL4xs6DMT2Zj
WfNmNEzkF+08PSXntuZMtK6CHbvOHTXtvxvHu391AeqCVNoKqGUo5lRKhNaOvQOYNKC6UmVG3atTKarZ
53HiAqyf/fHeDSth7HO//ewKLJtV2OCxc1kba4SC+oudz2l3vburxrpPEeUvq/pscOjY5H0At8JpSQCJDAMzYlC
oRm6pkkg23ORg65hEHqapiFyK5FZmtVee+DOV//+pyUf/09r8LJpr2okdNe3dZ2XZ2ZqFWdyciZQ3rLVi0Cjie
mZsfHKxERj8fJeLwoqs3j7N3908GCpKeH6C3xuWMkCW7paeFHgEWMmZ5gES+8e0HM9bGZSpIbqWNmx
5+1Hze6LDPHS8uh1VzBGQISCQj0RD0gyRxkpNRvKqAjRwT9+76X7vt2uT1/3Lz+TIjEx08QIODMZ10/uSFMQ
BYg8CmTgR0QyCkDTIZOPCZ0JoSfiFMfYHOI7/v2RFW+/Rgvsx376jblZCF1vcLgrlU/7gfTabtJMNCqtWrnleUE
2l1q2YmjhyEC55lx/wxU33n6XiZYf2hAYFlk/v19GAb6ypy40sxypuVA2Cxc7yte7g+NjCjCKqopcOGvXom2b7z
m+dtve3gu4wdsQ5+B2eW5T13U0cOma3gXFVG7F2btf+DvHpMP97nT/UF/x6O5XxusucQAlOWOMsVTM6
GwhgFRRFBEhM7gkRM64hqvWLx49UDMNR2ps8xU/axXzidLEP358g0p288hhGKnIc13//Td+YGqirpTKZuK
2E5QqVRlIK2YUerpId9sSCvGCPP1zex754ZEH/hSkQQ+8z1zsmMmgOqoP9XIzHX3zbvrGdWR1cz4X7C/+Ib7
3g/76rx/WtiBxSqeaUT4eZV1scWyv9CbnEoNkpnD9+l7N5HoyPzldDsMwYbFFI33tuu0HYAdt13VJZWw30DS
2eCiXTsdRgqHpkaTZSm26NuZ4SlhJZAyaY+s3rOldkdjxfG3FZx6Kk/rr59bq+bTnAlfATWb4msNrDc/84HWXSi
SfOAZRqdzMF7pTccF1vn373lQ+15URcPZ3lK2kUIee+NH0M3/6H1urf3xxgLuVj57l3ndCG59btGGTefXITosb

o+CXFo4XojvasWs3OB/zs+Gz/i+Z6lZ65OozxbBWN5YEfsiCSAVBFIU+YxAzxdKlixu2a/vtS6++tdFiq5atKGSinl
SQAMdXrGJHttR3HDq2ff/+2dKMCTxv6cu7M2ctG1l81vqmziK3O5lNmZROh8FFX3t265efGFh1GZOQDPWR
t7/3Hd/Zu2pk1YP3/G3vrj0YeFJ5Q4OFeq189MTY+MTUitVLBgu6pqUTXo1bPihvydu+8u7bn/rL3lw5ZF0D/J
Xp4qFj8YrdeunV1gG8xCkuCKuw1r8ym0oOj51f9Y4eLPU6IjfcfWKR8TRHR6mM7XsBRSySEEWR9DzksOnMt
a7XOmPj2SD1h+75FpA9erw2W2Uy1rVg45p4V9ILvf0Hd/iyzYQKKWiHshn4o3OlHUeOTW6fSNV5rA2gWkl
Bv791Y8uEprSGr/tc/8U3TEy0+Jk3Hv/zv46P7tGNVK3CXnvpkO+EpPxCLhbTDOXT9NgcSJMnNJx5SYYWKK6
xkjIzK25+7pxrPj1ZH3y+ZJLlmZwiJ3pxL7w+c8FY6tOl6sxAttCyVbOe9rSvnFe6xHr1o3OHnirYMCULgRR+xP9f
vaJDFKVfm0kAAAAASUVORK5CYII=\n"

            },

            "metadata": {},

            "execution_count": 103

        }

    ]

},

{

    "cell_type": "code",

    "source": [

        "y = image.img_to_array(test2)\n",

        "y = np.expand_dims(y,axis=0)\n",

        "predicted = np.argmax(Flower_model.predict(y))\n",

        "Prediction_category = ['daisy','dandelion','rose','sunflower','tulip']\n",

        "Prediction_category[predicted]"

    ],

    "metadata": {

        "colab": {

            "base_uri": "https://localhost:8080/",

            "height": 36

        },

```
            "id": "NKSdrnRjmGyE",
            "outputId": "58a9434b-9a6b-44e9-b090-bd6cd599978d"
          },
          "execution_count": 115,
          "outputs": [
            {
              "output_type": "execute_result",
              "data": {
                "text/plain": [
                  "'sunflower'"
                ],
                "application/vnd.google.colaboratory.intrinsic+json": {
                  "type": "string"
                }
              },
              "metadata": {},
              "execution_count": 115
            }
          ]
        }
      ]
}
```