# Made By Lakshya and Pranath On 14/4/19  
  
import time  
import xlrd  
from PIL import Image  
import numpy  
from random import shuffle  
from itertools import combinations  
  
# Global  
path = "C:\\Users\\Lakshya Sharma\\Desktop\\Projects\\Math Project\\version 2\\"  
data = [] # Stores Student Details from excel sheet as a 2D Array of strings  
shape\_pixels = [] # Stores (R,G,B,A) Pixel Value in a 2D Array  
stud = [] # Stores Encoded colour values for each Student in a 2D Array  
boundary = 50 # Boundary Margin in Pixels  
rowpixels = 2100 # Number of pixels in one column  
colpixels = 2100 # Number of pixels in one row  
pixels = numpy.array([[None] \* colpixels] \* rowpixels) # Stores (R,G,B,A) Pixel Value in a 2D Array  
rowsheet = 0 # Number of Student Tuples  
colsheet = 0 # Number of Student Attributes  
objects = 0 # Number of identified colourable objects in the image  
visited = numpy.array([]) # Stores identified objects of an image as a 2D array  
key = [i for i in range(256)] # Stores Possible colour values (0 to 255)  
shuffle(key) # This shuffles the array with value range 0-255  
# Possible characters in the data:  
character\_array = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',  
 'u', 'v', 'w', 'x', 'y', 'z', 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', 'A', 'S', 'D', 'F',  
 'G', 'H', 'J', 'K', 'L', 'Z', 'X', 'C', 'V', 'B', 'N', 'M', '1', '2', '3', '4', '5', '6', '7', '8',  
 '9', '0', '.', '@']  
  
  
# UDF  
  
  
def Read():  
 *"""Reads Values from excel sheet and stores it in a 2D array"""* print("\nReading Data from Excel Sheet\n")  
 workbook = xlrd.open\_workbook(path + "Stud\_Details.xlsx")  
 sheet = workbook.sheet\_by\_index(0)  
 global rowsheet  
 global colsheet  
 rowsheet = sheet.nrows - 1  
 colsheet = sheet.ncols  
  
 for i in range(1, sheet.nrows):  
 temp = []  
 for j in range(0, sheet.ncols):  
 cell = sheet.cell\_value(i, j)  
 if type(cell) == float:  
 cell = int(cell)  
 temp.append(str(cell))  
 data.append(temp)  
 print("Data has been read from the Excel Sheet\n")  
  
  
def ImageGeneration():  
 *"""Generates an image on the basis of user's choice of shape"""* shape = ""  
 print("Enter Choice of Shape : ")  
 print("1. Circle\n2. Triangle\n3. Square\n4. Rectangle \n5. Pentagon\n6. Hexagon")  
 ch = int(input("\nYour Choice : "))  
 if ch == 1:  
 shape = "Circle"  
 elif ch == 2:  
 shape = "Triangle"  
 elif ch == 3:  
 shape = "Square"  
 elif ch == 4:  
 shape = "Rectangle"  
 elif ch == 5:  
 shape = "Pentagon"  
 elif ch == 6:  
 shape = "Hexagon"  
  
 rows, cols = ShapeProcessing(shape)  
  
 print("\nImage is being created\n")  
 # Making Boundary  
 black = (0, 0, 0)  
 for i in range(rowpixels):  
 for j in range(colpixels):  
 if i < boundary or i > rowpixels - boundary - 1 or j < boundary or j > colpixels - boundary - 1:  
 pixels[i][j] = black  
  
 # Putting Multiple Copies of Image in array  
 for i in range(boundary, colpixels - boundary):  
 for j in range(boundary, rowpixels - boundary):  
 pixels[i][j] = shape\_pixels[(i - boundary) % cols][(j - boundary) % rows]  
  
 # Saving array as Image  
 im = Image.open(path + "IP.png")  
 for i in range(colpixels):  
 for j in range(rowpixels):  
 im.putpixel((i, j), pixels[j][i])  
 im.save(path + "IP.png")  
  
 print("Image Created\n")  
  
  
def ShapeProcessing(shape):  
 *"""Opens an image and identifies all the pixels to get a 2d array of pixels of an image"""* im = Image.open(path + "Sample\\" + shape + ".png")  
 x = y = 0  
 check = False  
 temp = []  
 temp\_x = 0  
 while True:  
 try:  
 temp.append(im.getpixel((x, y)))  
 check = False  
 x += 1  
 except IndexError:  
 if check:  
 rows = y  
 cols = temp\_x  
 break  
 temp\_x = x  
 y += 1  
 x = 0  
 check = True  
  
 for i in range(rows):  
 temp1 = []  
 for j in range(cols):  
 index = i \* rows + j  
 temp1.append(temp[index])  
 shape\_pixels.append(temp1)  
  
 return rows, cols  
  
  
def Object\_Identification():  
 *"""Identifies Objects in the image"""* print("Object Identification is starting\n")  
 global visited  
 visited = numpy.array([[0] \* colpixels] \* rowpixels)  
 check = True  
 i = 0  
 j = 0  
 black = [(0, 0, 0), (0, 0, 0, 255)]  
 obj = 1  
 used = []  
 temp = 0  
  
 def Check():  
 for i in range(rowpixels):  
 for j in range(colpixels):  
 if visited[i][j] == 0:  
 return True  
 return False  
  
 while check:  
  
 # Case of just going through blacks  
 if pixels[i][j] in black:  
 visited[i][j] = -1  
 if j + 1 == colpixels:  
 if i + 1 == rowpixels:  
 check = Check()  
 continue  
 else:  
 j = 0  
 i = i + 1  
 continue  
 else:  
 j = j + 1  
 continue  
  
 # Case of Object Identification  
 if pixels[i][j] not in black and visited[i][j] == 0:  
 for k in range(j, colpixels):  
 if pixels[i][k] not in black and pixels[i - 1][k] not in black:  
 temp = k + 1  
 obj = visited[i - 1][k]  
 break  
 if pixels[i][k] in black:  
 if pixels[i][j - 1] not in black and visited[i][j - 1] != 0:  
 temp = k  
 obj = visited[i][j - 1]  
 break  
 else:  
 temp = k  
 obj = 1  
 while obj in used:  
 obj += 1  
 used.append(obj)  
 break  
 for k in range(j, temp):  
 visited[i][k] = obj  
 j = temp  
 # Finding No. of Objects after Completion  
 objects = 1  
 while objects in used:  
 objects += 1  
 objects -= 1  
 print(objects, "Objects in the image have been identified\n")  
  
  
def TestImage():  
 *"""Just a test case"""* white = (255, 255, 255)  
 black = (0, 0, 0)  
 im = Image.open(path + "Square.png")  
 for i in range(10):  
 for j in range(10):  
 if i in range(0, 1) or j in range(0, 1) or i in range(9, 10) or j in range(9, 10):  
 im.putpixel((i, j), black)  
 else:  
 im.putpixel((i, j), white)  
 im.save(path + "Square.png")  
  
  
def Encode():  
 *"""Encodes details of each student into an array of colours"""* print("Student Details are being Encoded\n")  
 global stud  
 combination = []  
 for comb in combinations(key, 3):  
 combination.append(comb)  
 shuffle(combination)  
 for i in range(rowsheet):  
 temp = []  
 for j in range(colsheet):  
 for k in range(len(data[i][j])):  
 a = ord(data[i][j][k])  
 temp.append(combination[a])  
 temp += [combination[a]]\*2  
 shuffle(temp)  
 stud.append(temp)  
 print("Student Data has been Encoded\n")  
  
  
def Colour():  
 *"""Colours the picture according to the student"""* print("Images are being created for each student\n")  
 for i in range(rowsheet):  
 im = Image.open(path + "IP.png")  
 for x in range(rowpixels):  
 for y in range(colpixels):  
 if visited[y][x] != -1:  
 im.putpixel((x, y), stud[i][(visited[y, x]) % len(stud[i])])  
 im.save(path + "OP" + str(i + 1) + ".png")  
 print("Image has been created for student",i+1,"\n")  
 print("All the Images have been saved\n")  
  
  
# Main  
  
start = time.time()  
  
Read()  
  
ImageGeneration()  
  
Object\_Identification()  
  
Encode()  
  
Colour()  
  
end = time.time()  
  
print("Total Time Taken : ", int(end - start), "seconds\n")