Owen Goodwin

CPS109 Assignment #3

09/26/18

1:  
def encode2(message, key):  
  alpha="abcdefghijklmnopqrstuvwxyz"  
  rest = ""  
  for letter in alpha:  
    if not(letter in key):  
      rest = rest + letter  
  alpha2 = rest+key  
  secret = ''  
  message = message.lower()  
  for letter in message :  
    if letter.lower() in alpha:  
      i = alpha.find(letter)  
      secret = secret + alpha2[i]  
  return secret

>>> encode2('Alan Turing defined computing', 'turing')  
>>> 'aoaqztxkqhdefkqedcspvtzkqh'

2:  
def encode3(message, key):  
  alpha="abcdefghijklmnopqrstuvwxyz"  
  rest = ""  
  for letter in alpha:  
    if not(letter in key):  
      rest = rest + letter  
  revAlpha = ""  
  for i in range(len(rest)-1, -1,-1):  
    revAlpha = revAlpha+rest[i]  
  alpha2 = key+revAlpha  
  secret = ''  
  message = message.lower()  
  for letter in message :  
    if letter.lower() in alpha:  
      i = alpha.find(letter)  
      secret = secret + alpha2[i]  
  return secret  
  
>>> encode3("Alan Turing defined computing", "turing")  
>>>'tstphfkxpzingxpniroqmfhxpz'  
  
>>> encode3("Ada Lovelace, first programmer'", "earth")  
>>>'etesoghserhzwlkjnloyleqqhl'

3:

#3 generates the correct output.  
4:

#4 generates the correct output.  
5:   
def spaces(stuff):  
  spaced = ""  
  for char in stuff:  
    spaced = spaced + char + " "  
  print spaced  
  
6:  
def spaces2(stuff):  
  spaced = ""  
  for char in stuff:  
    if char == " ":  
      spaced = spaced + char + " "  
    else:  
      spaced = spaced + char  
  print spaced

7:

A: The max value is 255 because the components are stored as 8-bit ints, meaning the highest possible value is 255 (including 0).

B: Since each component uses 8 bits, the memory required to store the color of a pixel would be 24 bits (8\*3)

C: There are 16777216 possible colors in the RGB model (256­­3)

D: This is more than enough colors

9:

The second version is by far the most efficient way of writing this program and what I would most likely do, however, more inexperienced programmers would likely opt for the third version as its step-by-step approach may be easier for them to understand.

10:

def swapRG(image):

for px in getPixels(image):

r = getRed(px)

g = getGreen(px)

b = getBlue(px)

newCol = makeColor(g, r, b)

setColor(px, newCol)