COMPUTER SCIENCE 110

**INTRODUCTION TO COMPUTER SCIENCE**

**HOMEWORK 3 (Python) SOLUTIONS**

# Spring 2018

**Assigned**: Monday, February 19 **Due**: **by beginning of class** Wednesday, March 7



**Part I (50 points) Hardware, and units large and small** (**Show your work; all answers to Parts A and B must be typed. Take pride in the product of your efforts! Do not put any answers on this sheet; you are welcome to print a copy of the table below from Blackboard if you like, or redraw it by hand)**:

**A. Short answer (26 Points)**

1. Chapter 5 (Computing Components, p. 151), 2-16 even numbered questions only. (8 pts.)

|  |  |
| --- | --- |
| 2. | PicoA |
| 4. | MilliD |
| 6. | GigaG |
| 8. | MegaF |
| 10. | Often used to describe size of memory.F or G |
| 12. | Latin for thousandth.D |
| 14. | PetaI |
| 16. | Roughly equivalent to 250I |

1. Your hard drive has 2.2 GB capacity remaining. Can you store a new program that requires 2300 MB? Explain. (3 pts.)

2.2GB = 2.2\*230 = 2,362,232,012 bytes (rounding down to the nearest byte)

2300MB = 2300 \* 220 = 2,411,724,800 bytes

Therefore, the answer is NO, we cannot store the new program since it contains more bytes than we have available.

1. Chapter 5, problem 28 (4 pts.)

A. 512MB machine

512 \* 220

B. 2GB machine

2 \* 230

1. Chapter 5, problem 35 (3 pts.)

The computer component that acts as the state manager is the control unit. It controls the actions of the other components in order to execute instructions in sequence.

1. Chapter 5, problem 40 (3 pts.)

The control unit goes to the address named in the program counter, makes a copy of the contents of that address, puts the copy into the instruction register, and increments the program counter.

1. Chapter 5, problem 44 (3 pts.)

Because RAM is volatile and ROM cannot be changed, there must be places to store data and programs outside of the computer's main memory. Such places are called secondary storage devices. Secondary storage also typically provides more space than primary memory, allowing us to store more information.

1. Chapter 5, problem 48 (2 pts.)

A cylinder is a set of concentric tracks on a hard drive; that is, tracks that line up under one another.

**B. Disk drives (24 Points)**

Consider a hard drive consisting of 4 platters. For each platter, data can be stored on its **top** and **bottom** surfaces. Each surface consists of 150,000 tracks, each divided into 3600 sectors. Each sector can hold 512 bytes of data. The platters rotate at a speed of 10,000 rpm. The track to track arm movement time is 0.0001ms. **Type your answers, and show your work.**

1. What is the maximum amount of data that can be stored on the entire drive, including all platters and sides? Express your answer both in bytes and in gigabytes. Show your work (5 pts.).

4 platters \* (2 surfaces/platter) \* (150000 tracks/surface) \* (3600 sectors/track) \* (512 bytes/sector) = 2,211,840,000,000 bytes. To convert to GB, divide by 230, giving 2059.94 GB (about 2 TB)

1. What is the rotational time of the drive in ms.? (4 pts.)

To convert RPM into ms per revolution, divide RPM into 60000: 60000/10000 = 6 ms

1. What is the best case, worst case, and average time for each component of the access time for this hard drive? (15 pts) all times are in ms

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Worst | Average |
| Seek Time | 0 | 0.0001\*149999 = 14.9999  | 14.9999/3 = 4.999 |
| Latency | 0 | 6 | 6/2 = 3 |
| Transfer | 6/3600 = 0.00167 | 0.00167 | 0.00167 |
| Total | 0.00167 | 21.00157 | 8.00164 |



**Part II (50 points) Python**

Your Python code must follow these guidelines:

* *Always put your name, date, program name, and a general comment describing what your program does at the top of your code. This should be a general statement about how your program behaves.*
* *Variable names should help describe the purpose of a variable.*
* *Function names should help describe the purpose of a function*
* *Every function should be preceded by a comment describing the function’s purpose, input, and output (if any).*

Be aware that if you don’t follow these guidelines, you will lose points, **even if your program runs perfectly**.

Your assignment is to start with a picture of someone’s head and do the following to it:

* Turn the teeth orange.
* Turn the eyes red.
* Turn the hair purple.
* Add some brief, humorous text to the photo. The color of the text can be anything but black and you must use “makeStyle” to choose a font and size.
* Draw a green oval or rectangle around the text.

Do not use a picture from the Media Sources folder. Use a picture of yourself, a friend, or someone famous. You can find the picture on the Internet or it can be one of your own.

**Tips**:

1. Start with the template we provide for you on Blackboard.
2. A perfectly working program as described above will earn 45 points. To earn the remaining 5 points, you must use your own creativity to go above and beyond those minimal requirements. Examples include particularly challenging hair/eye/teeth configurations or colorings, changes to other parts of the picture, whatever else you can think of that would make it more interesting, etc. Go nuts!
3. Build on what you’ve learned! If you don’t have your book beside you and your code from the labs in front of you when you’re doing this, then you are going to waste a lot of time.
4. Use the following strategy to design your program:
	1. Write 3 (or more) separate functions to do the teeth/eyes/hair color changing. These should all be similar to the bigBenColors() function from Lab 4! Use the “explore” function on your own in the command area (not inside your program) to find the proper ranges and color distances; experiment!
	2. Your main function should be called “graffiti( )” and should do the following:
		1. Get the original file from the proper folder, using getMediaPath as in HW 1 and HW2.
		2. ***Call*** your three (or more) color manipulation functions to transform the picture
		3. Add the text (you do not need to write a separate function for this, it’s only a couple lines of code): use “makeStyle” and “addTextWithStyle”.
		4. Draw the oval/rectangle (you do not need to write a separate function for this, it’s only a line of code): use “addOval” or “addRect”
		5. Display the final product !!!

**TURNING IT IN:**

1. Turn in a printout of your code (the hw3.py file) with the rest of your homework.
2. Put a copy of your hw3.py file and a copy of the original picture in your csis110/hw/hw3 folder. THIS IS REQUIRED IN ORDER FOR ME TO GRADE IT AND FOR YOU TO EARN FULL CREDIT.

# CSIS110, HW3, Python portion (graffiti function), Spring 2018. Turns

# William Shatner's face into

# something even scarier: orange teeth, red eyes, and purple hair.

# Also adds a text message surrounded by an oval.

# Turns the teeth orange

def teeth(facePic):

 orange = makeColor (255, 70, 0)

 for facex in range (181, 262):

 for facey in range (309, 321):

 p = getPixel (facePic, facex, facey)

 color = getColor (p)

 if (distance (color, white) < 190.0):

 setColor (p, orange)

# Turns the whites of the eyes red

def eyes(facePic):

 # approximate color of eye whites in this picture (not pure white)

 eyeWhites = makeColor (191, 148, 142)

 #left eye (doing eyes separately to avoid changing other parts of face)

 for facex in range (170, 197):

 for facey in range (210, 217):

 p = getPixel (facePic, facex, facey)

 color = getColor (p)

 if (distance (color, eyeWhites) < 90.0):

 setColor (p, red)

 #right eye

 for facex in range (268, 290):

 for facey in range (215, 220):

 p = getPixel (facePic, facex, facey)

 color = getColor (p)

 if (distance (color, eyeWhites) < 110.0):

 setColor (p, red)

# Turn hair purple

def hair(facePic):

 # approximate colors of two sections of his hair

 hairColor1 = makeColor (199, 192, 176)

 hairColor2 = makeColor (119, 101, 90)

 # able to do it with one rectangle by comparing each pixel to both hair

 # colors and not changing any pixel with lots more red than blue & green

 # (since that's probably skin)

 purple = makeColor (51, 0, 154)

 for facey in range (35, 225):

 for facex in range (48, 322):

 p = getPixel (facePic, facex, facey)

 color = getColor (p)

 r = getRed (p)

 g = getGreen (p)

 b = getBlue (p)

 if (distance (color, hairColor1) < 90.0 or distance (color,hairColor2) < 90 or distance (color, white) < 50) \

 and not((r > g+20) and (r > b+20)):

 setColor (p, purple)

 # Next line just slows it down so we can watch

 repaint(facePic)

# Adds a red, 24-point message to the picture on his shirt

def printCaption(facePic):

 myStyle = makeStyle(mono, italic, 24)

 addTextWithStyle(facePic, 206, 536, "Use Priceline!", myStyle, red)

# draws a green oval around the text message

def drawStuff (facePic):

 addOval(facePic, 196, 510, 200, 36, green)

# main function to carry out all of above, as described in top comment,

# plus display the picture

def graffiti():

 shatPic = makePicture (getMediaPath ("shatner.jpg"))

 teeth (shatPic)

 eyes (shatPic)

 hair (shatPic)

 printCaption (shatPic)

 drawStuff (shatPic)