COMPUTER SCIENCE 110

**INTRODUCTION TO COMPUTER SCIENCE**

**HOMEWORK 2,** Fall 2018 – Python MM

**Assigned**: September 24 **Due**: **by beginning of class** Wednesday, October 10

BD14710_

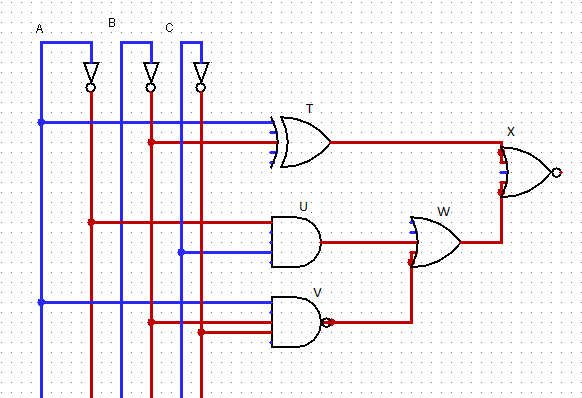
**Part I (60 points) Representing Colors; Boolean logic, expressions, gates, circuits.** **Show your work. You may hand-write the work if neat, otherwise type your work. Take pride in the product of your efforts!). Put your work and answers on another sheet of paper, not this one**.

1. (6 points) Use Microsoft Word (Font Colors | More Colors | Custom) or the Internet to find the RGB, or 6 **hex-digit** values – approximately – for the following colors:

* Lavender
* Maroon
* Purple

The colors will be approximate, but MUST be in the form of 6 **hex-digit values** (two for each of RGB).

1. (12 points; 4 points each)
   1. Write the Boolean expression that describes X in the circuit diagram below.



* 1. If A is TRUE, B is TRUE, and C is FALSE, what are the states of T, U, V, W, and X?
  2. If A if FALSE, B is TRUE, and C is FALSE, what are the states of T, U, V, W, and X?

1. (8 points) Consider the following two Boolean expressions:

**A** AND (**B** OR **C)’**

**A**’ AND (**B**’ OR **C’)**

Prove whether these Boolean expressions are equivalent. Make sure to state your conclusion and reasoning.

1. (9 points) For the following, assume that A = true, B = false, and C = true.  Evaluate each expression below (i.e., state whether it is TRUE or FALSE), **showing your work**:

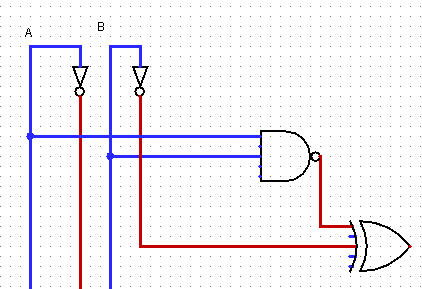
* ( A AND B )’ AND ( B’ AND C’ )
* ( B’ ) OR (B OR C’)
* ( A AND B’ AND C’) OR ( A’ AND B AND C’ ) OR ( A’ AND B AND C’ )

1. (16 Points) Design a circuit using the Sums-of-Products algorithm that implements the following truth table. For full credit, you must show the results from each step of the algorithm, including the final equivalent Boolean expression for this truth table.

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **Output** |
| 0 | 0 | 0 | **1** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 0 | 0 | **1** |
| 1 | 0 | 1 | **1** |
| 1 | 1 | 0 | **0** |
| 1 | 1 | 1 | **0** |

For 4 points extra credit, draw a simpler circuit (one using fewer gates) that does the same thing as the one you drew based directly on the algorithm.

1. (9 points; 3 each)
2. Draw a circuit diagram corresponding to the following Boolean expression: (A’B) + (B + C)’
3. Show the behavior of the following circuit with a truth table



1. Draw the circuit from Question F(2) using only AND, OR, and NOT gates (no XOR or other types of gates).

BD14710_

**Part II (40 points) Python Programming**

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**.

Write a function named **createCollage()** to create a collage of the same image at least four times, then mirror the collage. **Your function must be saved in a file called “hw2.py” in your csis110/hw/hw2 folder AND uploaded through Canvas.** You can use any image you want, either from MediaSources or something you find on the web or a picture of your dog, or whatever. You should create a blank canvas of the proper size (as long as it fits on a normal monitor) using the Python function **makeEmptyPicture**. One of the four images in your collage must be the original, unaltered image. The others must be modified forms of the original image. You can scale, crop, rotate, create a negative, shift or alter colors, make it darker/lighter, or virtually anything else you want. Be creative and have fun with it!!

After making this image, mirror it (you will now have 8 “copies” of the picture). You can do this vertically or horizontally (or some other way), in any direction – just make sure that the original 4 images are still visible after mirroring. The **createCollage()** function you write must make all of this happen – all of the above effects and compositing must occur from the single function **createCollage()**. Of course, I expect you will want to write and use additional functions to make this easier, but someone (like me!) testing your program must be able to simply call **setMediaPath()**, load your program, then execute **createCollage()** to see your collage generated and displayed.

**NOTE**: A perfectly working program as described above will earn 35 points. To earn the remaining 5 points, you must use your own creativity to go above and beyond those minimal requirements. Examples include additional manipulations of the original picture, unique mirroring techniques, original or unique manipulations, etc. Go nuts!

**Tips**:

1. Start with the working program that we give you on Canvas. Get that to work, then change it to use a different picture, add more manipulations, etc.
2. Build on what you’ve learned! There’s actually not much that’s new here, mostly putting together things you’ve already done. 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.
3. To get inspiration, check out some of the collages at <http://home.cc.gatech.edu/gacomputes/46>. Not all of these satisfy the assignment, and some are more ambitious than you have to be, but feel free to be inspired by them! I will also show you some collages from previous semesters for inspiration.

**TURNING IT IN:**

1. Turn in a printout of your code (the hw2.py file) with the rest of your homework.
2. Put a copy of your hw2.py file and your original image (not the collage) in your csis110/hw/hw2 folder.
3. Upload the compressed hw2 directory to Canvas.