CSE 1321L: Programming and Problem Solving I Lab

Assignment 2 – 100 points
Conditional and Looping Statements

What students will learn:

1) Problem solving
2) Structure programs to include conditional logic
3) Write code that includes if/else statements
4) Using loops

Overview: For this assignment, you're going to practice making decisions in your code. It's because of those decisions that your program can behave differently depending on the values in your variables; these decisions are called **conditional statements**. In practical terms, this means you're going to expand on the concepts from assignment 1 and 2, but also include things like IF, ELSE IF, ELSE and /MATCH statements. Again, start early, practice, and ask a lot of questions.
Follow the same conventions for class names and file names for your source code. Make sure to follow the FYE Submission Guidelines. Finally, we don't mean to lecture, but we want to remind you not to cheat. This is the core of what a lot of you will be doing for a living, so master it now.

Assignment 2A:

Image Color Depth: In Assignment 1, we learned that each pixel of a computer image is represented with three values, RGB. If we add transparency, we would have RGBA ("A" for alpha channel). Traditionally one byte is used for each color, meaning that each pixel takes up 4 bytes of space. These are known as 8 bits per color channel encoding, or 8 BPC for short. More recently, encodings with larger ranges of color have become popular with artists and photographers. Programs like Photoshop now support 16 BPC and 32 BPC encoding.

For this assignment, you will try to determine the encoding of an RGBA image based on its width, height, and file size. After the user enters this information, you will first check if the information is valid. If not, you will notify the user and skip to the end of the program.
(*Hint: You will have to check multiple potential issues with your IF statements*)

If the detail is valid, you will then calculate the BPC encoding. Based on the value, you will use a Switch statement to print out if the image is 8, 16, or 32 BPC. If it is outside the range, inform the user that your computer does not know how to read this encoding.

You must save the code in a file called "Assignment2A.py".

Sample Output #1:

```
[Image Encoding Checker]
What is the image width? 10
What is the image height? -20
What is the file size (in bytes)? -200

The information is invalid - please re-enter it.
```

Sample Output #2:

```
[Image Encoding Checker]
What is the image width? 10
What is the image height? 20
What is the file size (in bytes)? 202

The information is invalid - please re-enter it.
```

Sample Output #3:

```
[Image Encoding Checker]
What is the image width? 50
What is the image height? 50
What is the file size (in bytes)? 10000

The RGBA image is encoded with 8 bits per channel.
```

Sample Output #4:

```
[Image Encoding Checker]
What is the image width? 100
What is the image height? 100
What is the file size (in bytes)? 80000

The RGBA image is encoded with 16 bits per channel.
```

Assignment 2B: *Diamonds in the sky.*

In our earliest labs, we asked you to print a diamond pattern to the screen using predefined print statements. Now that we know how to use loops, we can make more dynamic and customizable patterns.

For this assignment, we will prompt the user to enter a single character and a maximum width for the diamond. If they enter a number less than 3, we'll prompt them to choose a correct width. If they enter an even number (greater than 3), we will add 1 to it and let the user know the final diamond size. Then we will generate and print out the diamond using the user's inputted character and ' ' symbols.

Hints: Each line of the "diamond" is made up of two parts – the character in the center and the spaces to the left of it. The amount of "left space" decreases as we go towards the middle diamond, then increases afterwards as we go towards the bottom. Could we use multiple loops (or even nested loops) to model this behavior? Also, note that the number of characters increases (and later decreases) by two on each line.

Call the file name Assignment2B.py.

User input is indicated in **bold**

Sample Output #1:

```
Enter a character to use: +
Enter the diamond's width: 5
  +
 +++
+++++
 +++
  +
```

Sample Output #2:

```
Enter a character to use: *
Enter the diamond's width: 8
To make a diamond, we'll use 9 as the width instead.
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Sample Output #3:

```
Enter a character to use: ~
Enter the diamond's width: -2
Please enter a width of at least 3.
Enter the diamond's width: 3
  ~
 ~~~
  ~
```

Submission:

1. You will submit 2 separate files containing source code – one for each of the assignments above. Make sure the files are named correctly and to include the header with your name and course section listed.
2. Upload both files (simultaneously) to the assignment submission folder in Gradescope.
3. We'll continue to work with you on this assignment if something messes up, so long as you submit by the due date. However, start early because we work during the weekday.