APS 105 — Computer Fundamentals
Lab 4: Loops and Functions
Winter 2018

The goal of this laboratory is to practice the material on for loops and functions. You are to write two separate C programs. The first part is to do a "drawing" of a triangle of various sizes. The other program is a number conversion using functions.

Instructions:

- Similar to previous labs, your solution will be marked by your TA during your scheduled lab period for programming style and your understanding, **and** should also be submitted electronically by the end of your scheduled lab period.

Preparation

Read through this entire document carefully, and do the work to create the programs that are described below. You are encouraged to:

- Read the class bulletin board on Piazza, to see if others had similar problems. If you do not see anything helpful there, ask a question. Do not ask for, or ever give a full or partial solution to the lab assignment.
- Ask for assistance from your lab/tutorial TAs.
- Attend and ask questions during the plenary sessions.

Notes:

- In the sample output examples that follow:
  - The text that would be entered by the program user is displayed using a **bold** font.
  - The text <**enter**> stands for the user pressing the enter key on the keyboard. On some keyboards, the enter key may be labeled return.
- When you execute your programs, the user's text will not be displayed in bold and <enter> is not going to show.
- Throughout this lab, there is a single space after the colon (:) in each output line.

Part 1 – Drawing a Triangle

In a file called **Lab4Part1.c**, write a complete C program that uses the ^ character to draw a triangle of a given number of rows. The program first prompts the user to enter the number of rows in the triangle. Your program may assume that the input is a valid integer from 1 to 20 (inclusive).

Here are some sample outputs from the execution of the program. The output of your program should match the sample output.
Sample output 1:

```
Enter the number of rows in the triangle: 1
^
```

Sample output 2:

```
Enter the number of rows in the triangle: 2
 ^
^^^
```

Sample output 3:

```
Enter the number of rows in the triangle: 3
  ^
 ^ ^
^^^^^
```

Sample output 4:

```
Enter the number of rows in the triangle: 10
         ^
        ^ ^
       ^   ^
      ^     ^
     ^       ^
    ^         ^
   ^           ^
  ^             ^
 ^               ^
^^^^^^^^^^^^^^^^^^^
```

Hint*:* You may find it helpful to draw the required output on a piece of graph paper before writing your program.

Part 2 – Number conversion

In a file called **Lab4Part2.c**, write a complete C program consisting of a main program and two functions that converts a binary number to decimal or a decimal number to binary.

main() part of the program:

1. Prompts the user by entering either **B** or **D** for choosing between two options of:
   - Conversion of Binary number to Decimal number, or
   - Conversion of Decimal number to Binary number respectively.
2. Prompts the user to input either a binary number or a decimal number depending on the choice made earlier.
3. Calls the proper function for conversion.
4. Display the results of the conversion that is either the binary or the decimal number.

Functions:

convertDecimalToBinary

- Receives a decimal number and returns the binary equivalent of that number.

convertBinaryToDecimal

- Receives a binary number and returns the decimal equivalent of that number.

Notes:

- All numbers are positive integers.

Below are several sample outputs from an execution of the program, which are separated by lines such as:

___Run 1____

In the sample output examples that follow, the user input appears after the colon character (':') and is followed by the **enter** key. Note that there is a single space after the colon (:) in each output line.

_____Run 1_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **B**
Enter your number: 101
101 in binary = 5 in decimal

_____Run 2_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **B**
Enter your number: 1111
1111 in binary = 15 in decimal

_____Run 3_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary**: B**
Enter your number: 10000000
10000000 in binary = 128 in decimal

_____Run 4_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **D**
Enter your number: 65
65 in decimal = 1000001 in binary

_____Run 5_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **D**
Enter your number: 258
258 in decimal = 100000010 in binary

_____Run 6_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **b**
Enter your number: Invalid input; Goodbye

_____Run 7_____
Enter B for conversion of Binary to Decimal, OR
Enter D for conversion of Decimal to Binary: **d**
Enter your number: Invalid input; Goodbye

Grading by TA and submitting your program for Auto-Marking

There are a total of 10 marks available in this lab, marked in two different ways:

1. **By your TA, for 4 marks out of 10.** Once you are ready, show your program to your TA so that we can mark your program for style, and to ask you a few questions to test your understanding of what is happening. Programs with good style have been described in class, but briefly, they are:

   a. Clear comments that describe what is happening in the program.

   b. Good choices for variable names that indicate their purpose. Please adopt the following naming convention illustrated above — if have you a variable that is described by multiple words, user lower case for the first letter of the first word, and Upper case for all subsequent words, e.g. **blendMode**.

   c. Properly indented code that has appropriate spacing between lines for readability.

   The TA will also ask you some questions to be sure that you understand the underlying concepts being exercised in this lab.

2. **By an auto-marking program, for 6 marks out of 10.** You should run the following command:

   /share/copy/aps105s/lab4/exercise

   Within the directory that contains your solution programs.

   This program will look for the files **Lab4Part1.c** and **Lab4Part2.c** in your directory, compile them, and run them on some test cases. If there is anything wrong, the **exercise** program will report this to you, so read its output carefully, and fix the errors that it reports.

   A key part of what you are to learn in this lab is the use of this program - most software programs give this kind of report. Read through the output of the **exercise** program and see if it is happy with everything, or if it is reporting an error. If there is an error, it will say so, and then it is up to you to fix what is wrong and try again. You'll need to do this for every subsequent lab.

   Once you have determined that your program is as correct as you can make it, then you must submit your program for auto-marking. This must be done by the end of your lab period as that is the due time. To do so, go into the directory containing your **Lab4Part1.c** and **Lab4Part2.c** file (and make sure this is the right place!) and type the following command:

   /share/copy/aps105s/lab4/submit

   This command will re-run the exercise program to check that everything looks OK. If it finds a problem, it will ask you if you are sure that you want to submit. Note that you may submit your work as many times as you want prior to the deadline; only the most recent submission is marked.

   Important Note: You must submit your lab by the end of your assigned lab period. Late submissions will not be accepted, and you will receive a grade of zero.

You can also check to see if what you think you have submitted is actually there, for peace of mind, using the following command:

/share/copy/aps105s/lab4/viewsubmitted

After the Final Deadline — Obtaining Automark

After all lab sections have finished, a short time later, you will be able to run the automarker to determine the automarked fraction of your grade on the code you have submitted. To do so run the following command:

/share/copy/aps105s/lab4/marker

This command will compile and run your code, and test it with all of the test cases used to determine the automark grade. You will be able to see those test cases' output and what went right or wrong.

Good Luck!