Introduction

This week's lab is another example of a split lab to give you more experience with planning a project before you sit down to write code. We will be combining several concepts this week with arrays!

Lab Objectives

By successfully completing today's lab, you will be able to:

- Conduct a brainstorming session with one or two individuals in the class.
- Create a planning document outlining your solution to the code.
- Implement a programming project that uses arrays.

Prior to Lab

- You should be familiar with input and output, conditionals such as if, if/else, and switch statements, for and while loops, and an introduction to arrays.
- This lab corresponds with zyBooks Chapter 5, 2.11, and 2.12.

Deadline and Collaboration Policy

- Part 2 of this assignment is due by 11:59 PM on Tuesday (10/8/2019) via Canvas.
- Part 3 of this assignment is due by 11:59 PM on Friday (10/11/2019) via Canvas.
  - More instructions on what and how to submit are included at the end of this document.
- You should write your solutions to Part 2 and Part 3 of this lab by yourself. In this lab, you will talk at a high level with others in your lab about a solution. However, you will create all code by yourself and **you should not talk about specific code to anyone but a course instructor or lab teaching assistant.**
- Your zyBooks chapters and lecture slides are available resources you can use to assist you with this lab.

Lab Instructions

The Problem:

As you learned in zyBooks and in lecture, ultimately everything a computer does ends up as a calculation involving binary digits: 0 or 1. In today's lab, you will build a program that takes in a decimal (or base-10) number and calculates and displays the binary (base-2) equivalent.

Some examples of binary conversion:

$10_{10} == 1010_2$
$42_{10} == 101010_2$
$22_{10} == 10110_2$

This week's lab again has substantial planning on the first day, and you coding your solution on the second day. Hopefully, you're getting pretty good at thinking about the requirements of a program and how to explain your solution on paper by now.

Lab 07 Structure	
Tuesday	Thursday
30 minutes: work with 1-2 other individuals on Part 1 20 minutes: work individually on Part 2 Submit results of Part 2 to Canvas by 11:59PM.	50 minutes: work individually on Part 3 Submit to Canvas by 11:59PM on Friday.

Part 1: Planning with a Group (First 30 minutes)
***** Do not use a computer or calculator in this part *****

You probably know your group members better by now, so share your favorite songs. Maybe discuss the differences between Coldplay and Radiohead, any my personal favorite question: Will Vampire Weekend ever release a 4th album? (You can probably guess my music tastes pretty well from this. Feel free to come up with your own, cooler questions.)

Working with the people in your row, brainstorm ways that you can write a program that converts between base-10 and base-2 (an example run of the program is included below). Here are some specifications for how your program needs to operate:

- The user is prompted to enter a positive integer value in base-10
  - If the number is valid, your program calculates and displays the binary number.
  - If the number is not a positive integer value, the program re-prompts the user to enter a positive integer value in base-10.
- The above steps complete until the user enters the EOF character (entered via control-z).
  - Talk with your group members to figure out how to accomplish this.
- Your program should use a while loop to accomplish the above steps.
- For calculating and printing the binary values, you should use a while loop and a for-loop respectively.
- You should be storing your calculations for the binary number in an integer array.
- Your output should match the output given below.

Specific items that you should consider with your group during your planning session:

- Verify that you each of you knows how to manually convert between decimal and binary.
- Draw out a flow chart that shows the steps that need to happen
- Decide on any sentinel values for loops (e.g., when do loops finish).

Sample Output

```
./a.out

Starting the CPSC 1011 Decimal to Binary Converter!

Please enter a positive whole number (or EOF to quit): 42

        42 (base-10) is equivalent to 101010 (base-2)!

Please enter a positive whole number (or EOF to quit): -2

        Sorry, that was not a positive whole number.

Please enter a positive whole number (or EOF to quit): 1.1

        Sorry, that was not a positive whole number.

Please enter a positive whole number (or EOF to quit): 11.0

        11 (base-10) is equivalent to 1011 (base-2)!

Please enter a positive whole number (or EOF to quit): <EOF>

        Thank you for using the CPSC 1011 Decimal to Binary Generator. Goodbye!
```

Part 2: Planning on Your Own (20 minutes of Tuesday Lab)
***** It's okay to use a computer and/or calculator for this portion *****

Note: This should not be when you're coding your solution. Make sure that you follow the steps below, using the guidelines and sample code above, to write and then submit your document before you begin.

Taking the information that was created during your brainstorm, write your own solution *in psuedocode*. How you choose to structure the planning document is of your choice, but you need to do the following:

- Review the explanation of *pseudocode* in the Safari playlist for this lab (https://learning.oreilly.com/playlists/1480ac0b-84d1-4ef6-97f4-f6936af15daf) or using the pseudocode.pdf document in Canvas.
- *Pseudocode* is NOT code. It is structured English that where you convey the key parts of your algorithm.
- Thoroughly consider all the issues you need to fully create a program as specified above (such as how to make the formatting look correct, how many variables do you need, what type, etc.).
- Include the following mandatory header:
    - Your First (or preferred name) and Last Name
    - Your Lab and Lecture Section
    - Lab 07 – Part 2
    - Today's Date
    - Collaboration Statement: In this statement you indicate who you worked with, and which lecture sections they are enrolled in.

Submit this part of the assignment to Canvas by 11:59PM (as specified in the submission Guidelines). Late submissions will not be accepted. This part is to help you think before you code.

Part 3: Implementation (50 minutes on Thursday).

Lab Exercise

Using your planning document from Part 2, you will create a C program called binaryconverter.c that creates the converter as described above.

Make sure you include the standard lab comment block introduced in Lab 3.

Bonus Exercise

Create a copy of your binaryconverter.c file called decimalconverter.c that takes in a binary number and does a conversion to decimal.

Submission Guidelines

- Part 1: No submission, but don't skip it. Thinking aloud and collaborating at a high level is incredibly important in computer science. It'll help you ask better questions at office hours, write cleaner code, do great group work, and even do better in interviews.
- Part 2: Submit your writeup to the Canvas assignment associated for this lab by 11:59PM on Tuesday (10/8/2019). You should check within Canvas to make sure your submission was uploaded correctly and in its entirety.
- Submit your binaryconverter.c program to Canvas assignment associated for this lab by 11:59PM on Friday (10/11/2019). You should verify within Canvas to make sure your submission was uploaded correctly and in its entirety.
    - If you opt to complete decimalconverter.c, submit that to the associated Canvas assignment by 11:59PM on Friday (10/11/2019).

Grading Rubric

- If you are not present and attending lab during Part 1 and Part 2 of this lab on Tuesday, you will not receive credit for the planning portion of this lab.
- If your document for Part 2 is not submitted successfully to Canvas by the due date, you will not receive credit for the planning portion of this lab.
- If you do not check in your code with a TA for Part 3 of this lab on Thursday, you will not receive credit for the coding portion of this lab!
- If your code for Part 3 is not submitted successfully to Canvas by the due date, you will not receive credit for the coding portion of this lab.
- Your assignment will be graded out of 100 points. The approximate grading distribution is:

- Brainstorming document completeness/accurate
  - Logic covers all requirements specified 10 points
  - Document is simple and easy to follow 10 points
- Program binaryconverter.c
  - Program operates correctly 20 points
  - Output is formatted correctly 20 points
- Program compiles without errors or warnings 10 points
- Proper code formatting and commenting (See Code Style Guide on Canvas) 30 points
- Optional bonus exercise decimalconverter.c up to + 10 bonus points
  - Requires binaryconverter.c is submitted