# **CS241 Lab 2, Winter 2020: C function and makefile**

**Objective:**

This lab is designed to help students gain hands-on experience with C functions and the use of the **make** utility from GNU (GNU’s Not Unix) project (write a makefile). The typical layout of a C program is also introduced.

**Description**

The integers produced by the function **rand()** all fall within the range [*0, n*], where *n* is a system-dependent integer value. The sequence of integers generated by this function is supposed to be random. In this lab, you will be able to test to confirm whether this is true or not.

The typical layout of a C program is shown in this lab. A makefile is also written for this lab. Your job for this lab is as follows:

* Make a directory for this lab, for example, cis241/labs/lab02. And go to this directory.
* Use the command below to download the source code from my web page to your current directory.

wget [http://cis.gvsu.edu/~wangx/teaching/cis241/w20/lab02/{lab02.c,lab02.h,main.c,makefile}](http://cis.gvsu.edu/~wangx/teaching/cis241/w20/lab02/%7blab02.c,lab02.h,main.c,makefile%7d)

* You will find that this simple program consists of three files.
  + All of the functions for this program are defined in the file lab02.h with corresponding prototypes. In addition, it also contains preprocessing directives, such as #include <stdio.h>. This **.h** file is commonly known as the **header** **file** or simply **.h file** for a project written in C.

There are lots of header files in the /usr/include/ folder. Please examine some of them, such as stdio.h and ctype.h, to be familiar with the layout of the files.

**Question 01:** What could be the advantage to include the preprocessing directives in a header file?

**Including preprocessing directives in the header file allows the preprocessor to essentially ‘copy’ each of them over to the other .c files in the project. This allows you to use the header file to include common libraries, such as stdio.h, which allows the “printf” function.**

* + The actual implementations of the functions for this program are contained in the file lab02.c. Please note that the first line in this file is

#include “lab02.h”

This is to tell C compiler that all of functions that will be implemented in this file are defined in the lab02.h file.

* + The file main.c contains the **main()** function only. Again, this file starts with the line

#include “lab02.h”

**Note:** Although the function printf(), which is defined in stdio.h, is used in both lab02.c and main.c, you don’t need to include the line #include <stdio.h> in those files.

**Question 02** Why can the function printf() be referenced and used without including the stdio.h file in those files?

**Because we’ve included it in the header file. As explained before, when the project is compiled, each .c file has the header file on top of it….kinda like a head. (appropriate name).**

* + Please examine the source code to understand what it does. Usually, what a function does in included in the comment section of the function.
  + The file makefile is a simple make file that is used to compile, run, and clean up the program.
* Run the make command to build an executable and run it automatically. Check and see if it works as expected.

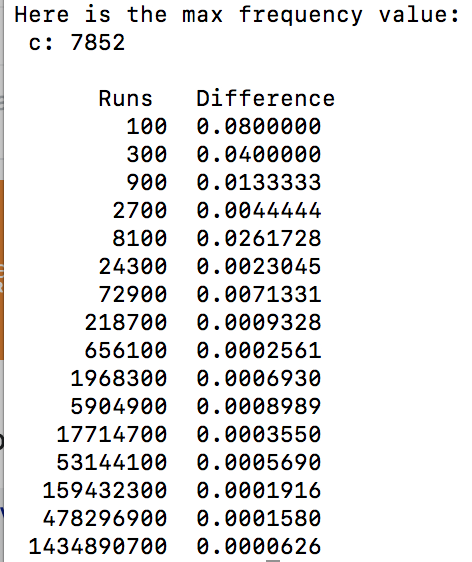
make

make run

* Open **lab02.c** in a text editor, read the code carefully (make sure you fully understand it), and modify the code to implement the function coin\_toss() as follows:

Implement the function to simulate the game of coin tossing by using the function rand(). Given number of runs (the parameter *n*), record the number of heads and number tails. The return value of this function is the absolute value of the difference (abs(heads – tails)).

* Open **main.c** with a text editor and modify it to do a simulation shown by the instructor. Essentially, when you run the program, the following results will be generated by the simulation:



* Run the make command to build an executable and run it automatically. Check and see if it works as expected.

**Deliverables**

To receive full credits for this lab you must complete the following:

1. Turn in a copy of the lab (either hard copy or to Blackboard) with the questions answered.
2. Provide a demo of the working code to your instructor in the EOS lab.

**Grading:**

1. Answers to the questions: 30%
2. Correctly working code: 70%