ECEN 324
Lab #1 – C Programming

This lab assignment is to convert a C++ program into a C program. It is to be completed individually as "Open Book no Solutions." The C program is to produce exactly the same output as the C++ program does and is to be commented as a C program. This means that even though your output looks the same as the output of the C++ program to you on the screen, if a `diff` command shows a difference, the output is not identical.

The C++ program that is to be converted to C (.c file) may be found on the ECEN 324 Linux VM:

```
/home/ecen324/cLab/ecen324_c_lab.cpp
```

You might copy it with a command like the following to your own home directory (or possibly a subdirectory you have made for ECEN 324 under your home directory) to have a file that has a '.c' extension that you would then modify to have only C code in it:

```
cp /home/ecen324/cLab/ecen324_c_lab.cpp  lab1.c
```

There is one section of the code where the C++ code is not given to you. Write the C code to generate the table as given in the C++ file.

- **Your C code should have a "`.c`" extension and be compiled with `gcc`. The command to compile your code might look like:**
  ```
  gcc -o mylab1  lab1.c
  ```

- **You will not be able to use the style checker on the C code but you still need to use good style.**

- **To check the correctness of your program, use "`55`" as the input value the program should ask for and use the "`diff`" command to compare your output with the `c_lab_solution.txt` file in `/home/ecen324/cLab`. Here is an example of a successful correctness check where the diff command sees no differences between the solution file and the output of the compiled program:**

  ```
  [me@jordanvm c_lab]$ gcc -o mylab1 lab1.c
  [me@jordanvm c_lab]$ mylab1 >mylab1.out
  Enter number: 55
  [me@jordanvm c_lab]$ diff /home/ecen324/cLab/c_lab_solution.txt mylab1.out
  [me@jordanvm c_lab]$
  ```

  The lack of output after the `diff` command indicates that the two files were identical.

  The "`>`" in the second command line above, redirects (standard) output generated by `printf()` calls [output generated by `cout` in C++ code] of the `mylab1` executable to the file named `mylab1.out`. The reason the "`Enter number: `" prompt does not get redirected to the `mylab1.out` file, is that the prompt is printed with an `fprintf()` command [a `cerr` in the C++ code] which is sent to standard error.

- **If the `diff` command gives output and says the files being compared differ, but to you the files look identical, there is probably a problem with whitespace. If it is only a problem with whitespace, such as an extra space, or using spaces instead of a tab in the table, running a `diff` command with the `-w` option will show the files as being the same. Using the `-w` option with the `diff` command identifies for sure that it is a whitespace issue if the files are different. To learn about the diff command and its options (like: --color,) you might use the `man` or `info` commands. You might also see:** https://linuxhandbook.com/diff-command/**.**

- **Submit your C code by executing the "submit" command, like:**
  ```
  submit lab1.c
  ```
  The header in the `.cpp` has the proper lab name and format for use with the submit command. You should edit the header with your name and appropriate comments.

- **C programming tutorials that might be helpful:**
  https://computer.howstuffworks.com/c.htm/printable
  https://www.tutorialspoint.com/cprogramming/index.htm

- **A C reference card might also be useful:** https://users.ece.utexas.edu/~adnan/c-refcard.pdf

- **The `man` and `info` commands may be used to access documentation on UNIX/Linux commands, libraries, file formats and other items. There are various *sections* to the "manual" pages and doing a command like:**
  ```
  man printf
  ```
  Might bring up information in section 1 (one) of the man pages where what you really wanted was the information in another section. To get to the documentation on printf that will be most helpful, try:
  ```
  man 3 printf
  ```

- **You will lose points if you print each line of the body of the table with its own `printf()` statement. Do something more elegant than that.**

- **Dynamic memory allocation in C++ is done with `new` and `delete` operators. These are not available in C. A web search with something like "dynamic memory allocation in c" should get you to some good resources. Or, check the appropriate sections in the tutorials given above. Note that in C++, if the new operator fails, the program terminates (this is the default behavior). This is not the case in C for `malloc()`! In C, you have to check that the `malloc()` didn't return a null pointer.**

- **You will not be able to use the style checker on the C code but you still need to use good style.**

- **Properly comment your code. You should not leave the comments in your submitted code that talk about how to do something in C++ and then do it in C. Remove the comments, or change them to be appropriate for your submitted program.**