COS 135 Individual Assignment 2

Due: Monday 07/02/22 End of the day
This assignment has 2 sections (part #1, and #2) in 2 pages. Please submit a .zip file with image/s and source code/s.

Part #1 (40pts): Upload a photo (if necessary, two images) showing the following workflow in your Ubuntu terminal.

1. Check your current working directory (i.e., folder) using **pwd** command
2. List all the files in your current directory using **ls** command
3. Create a new directory using **mkdir my_folder** command (note: *mkdir* is the make directory command, *my_folder* is the directory name. Notice the space character after the command)
4. Change your current directory to the new directory created using the **cd my_folder** command (notice the space character between *cd* and *my_folder*)
5. Verify the new current directory again using the **pwd** command
6. Create two empty files using the **touch abc.c xyz.c** command (notice the space characters after the touch command and the first filename)
7. Check more information about the new files using **ls -l abc.c xyz.c** command (note: the output provides more information about the file permissions, owner's username, size, and date and time created)
8. Delete one file using **rm abc.c** command
9. Change your current directory to the previous (original) directory using **cd ..** command (notice the space character between cd and ..)
10. Now delete the folder you created along with the remaining file using **rm -rf my_folder** command (command structure: *rm <one space> -rf <one space> my_folder*). Here -rf stands for recursively and forcefully. *rm* command will not work on a folder directly; you need to at least use -r option to remove all the files inside the directory before removing the directory itself. When you use the -f, it will remove the files forcefully - even a file is currently open.

Note: It is not a good practice to use space characters when defining file names and directory names (for instance, instead of *my file*, use *my_file*). Modern operating systems (e.g., Windows, Mac OS, Ubuntu, Fedora) can handle the spaces in files and folder names. However, some Unix and older versions (especially servers and small device operating systems such as FreeRTOS and Embedded Linux) may not support the space character in file or folder names.

Part #2 (60pts): submit your source code/s (you may submit a single program or write separate programs for each**)**

(a) Write a C program to output following table into the console / terminal. Refer to sample codes uploaded in Brightspace for more formatting options. You **must** use format specifiers with field width and alignments.

Name	Age	Country
Peter	23	USA
Manuel	43	Germany
Ken	35	China
Moz	44	Ghana

(b) Use a suitable format specifier to complete the following code (submit the complete code).

```
#include <stdio.h>
int main()
{
    char chr = 67;
    printf("Character having ASCII value 67 is ___.\n", chr);
    return 0;
}
```

(c) Complete the printf() function with suitable format specifiers to output the data in variables (submit the complete code).

```
#include<stdio.h>

int main()
{
    int i_number = 12;
    float f_number = 3.125;
    double d_number = 7.5345345;
    char character = 'T';
    char text[] = "COS 135";

    printf("_____"_____);
}
```