CSCI3240 Lab 2: Loops & Functions

Objective:

- To understand the use of iterative structures (for, while, do..while) in problems involving repeated processes.
- To understand the basic use of function in a program

1. Write a program "***lab2Problem1.c***" to print a pattern of numbers from 1 to n as shown below. Each of the numbers is separated by a single space.

```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4
```

Requirement:

- The program should take **a single command-line argument**, n. In the above example, the value of n is 4.

Small hints:

- Since the pattern is symmetric, I generally recommend think of this problem in 2 halves, top half and bottom half. If you can do the top half, reverse the process to get the bottom half

Constraint: $1 <= n <= 9$

Sample Run:

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ gcc lab2Problem1.c -o lab2Problem1
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem1 3
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem1 5
5 5 5 5 5 5 5 5 5
5 4 4 4 4 4 4 4 5
5 4 3 3 3 3 3 4 5
5 4 3 2 2 2 3 4 5
5 4 3 2 1 2 3 4 5
5 4 3 2 2 2 3 4 5
5 4 3 3 3 3 3 4 5
5 4 4 4 4 4 4 4 5
5 5 5 5 5 5 5 5 5
```

2. Write a program "***lab2Problem2.c***" to calculate the permutation and combination for given $\boldsymbol{n}$ and $\boldsymbol{r}$. The values of $\boldsymbol{n}$ and $\boldsymbol{r}$ should be passed to the program as a command-line argument. The permutation and combination are defined for given *n* and *r* as:

$$P(n,r) = \frac{n!}{(n-r)!} \qquad C(n,r) = \frac{P(n,r)}{r!}$$

Output Format:

```
P(5,3) = 60
C(5,3) = 10
```

Requirements:

- The program should take 2 command-line arguments. The first argument should be the value of $\boldsymbol{n}$ and the second argument should be the value of $\boldsymbol{r}$.
- First create a **factorial** function. Then, you can use the factorial function in the functions to compute **permutation** and **combination**.
- The result should be printed in the **main** function.
- The value of n and r must be positive integers.
- If r is greater than n, the program should display: "Error: r should be less than or equal to n". And, then exit.

Constraint:

- $\mathbf{1 \leq n \leq 20}$
- $\mathbf{1 \leq r \leq 20}$

Sample Run:

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ gcc lab2Problem2.c -o lab2Problem2
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem2 5 3
P(5,3) = 60
C(5,3) = 10
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem2 20 5
P(20,5) = 1860480
C(20,5) = 15504
```

Please consider testing your program with the boundary cases. For example:

- $n = 20,\ r = 20$
- $n = 1,\ r = 1$
- $n = 1,\ r = 2$
- $n = 20,\ r = 1$
- Verify your solution using the following link:
  - https://www.calculator.net/permutation-and-combination-calculator.html

Rubrics (Total 100 points):

Problem1 (50 points):

Criteria	Points
Command line argument not used to get the input value	-10
The output grid dimension and numbers in the grid are not correct	-25
The output grid dimension is correct but the numbers in the grid are not correct	-15
The source code is not named correctly (it should be lab2Problem1.c)	-10
No submission or source code is missing	-50

Problem2 (50 points):

Criteria	Points
Command line argument not used to get the input value	-10
The program fails for large values of n and r (but works with smaller values)	-15
The program fails for any value of n and r	-30
The source code is not named correctly (it should be lab2Problem2.c)	-10
No submission or source code is missing	-50

Hints:

- https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm
- https://www.tutorialspoint.com/cprogramming/c_loops.htm
- https://www.tutorialspoint.com/cprogramming/c_functions.htm
- https://www.tutorialspoint.com/cprogramming/c_data_types.htm
- https://www.tutorialspoint.com/c_standard_library/c_function_printf.htm

Steps to Create the Log File:

1. **Open your terminal and start the scripting process by typing:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ script Lab2_Log.txt
```

2. **List all the files in the current directory:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ ls
```

3. **Compile your problem 1:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ gcc lab2Problem1.c -o lab2Problem1
```

4. **Run your problem1:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem1 9
```

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem1 3
```

Note: You can test with multiple other input values here.

5. **Compile problem 2:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ gcc lab2Problem2.c -o lab2Problem2
```

6. **Run your problem 2:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem2 20 19
```

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ ./lab2Problem2 10 3
```

Note: You can test with multiple other input values here.

7. **Exit the scripting process to finish and save the log file:**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ exit
```

8. **Convert the log file from txt to pdf using the txt created from using script**

```
(base) jovyan@jupyter-asainju:~/3240/lab2$ wkhtmltopdf Lab2_Log.txt Lab2_Log.pdf
```

The '**`Lab2_Log.pdf`** ' file will be generated in your current directory. Make sure this file is included in your submission.

Submission Instructions:

Please upload the following files:

- lab2Problem1.c
- lab2Problem2.c
- Lab2_Log.pdf
- AI_Disclaimer.pdf

- **Submission Due:** Check Lab 2 Dropbox