Lab Session 3: Control flow

Exercise 1: Star Pyramid

Take the number of rows as input from the user. Write a C program using nested *for* loops that print a pyramid of stars where the number of rows are input from the user. The output should look like the following:

```
Enter the number of rows: 8
        *
       * *
      * * *
     * * * *
    * * * * *
   * * * * * *
  * * * * * * *
 * * * * * * * *
```

Exercise 2: Right-sided tringle of stars

Take the number of rows as input from the user. Write a C program using nested *for* loops that print a right-sided right angle triangle of stars where the number of rows are input from the user. The output should look like the following:

```
Enter the number of rows: 6
     *
    **
   ***
  ****
 *****
******
```

Exercise 3: Find the smallest integer

Write a C program to find the smallest among a series of integers. The program should take the following steps:

1. Ask the user to enter the total number of integers they want to compare.
2. Prompt the user to enter the first integer, which will be considered the initial minimum value.

3. Using a *while* loop, continue to ask the user to input additional integers based on the total number specified earlier.
4. Compare each entered integer with the current minimum value. If the entered integer is smaller than the current minimum value, update the minimum value.
5. After comparing all the integers, display the smallest value found.

Write the C program to accomplish this task and ensure that the program handles the input and comparison of integers correctly.

Exercise 4: Guess the Number Game

It's time to design and develop the C program for an "Guess the Number" game.

In this game, the program will generate a random number between 1 and 100 (inclusive), and the player's task is to guess the correct number.

1. The program should generate a random number and store it.
2. Initialize a variable to keep track of the player's guesses (e.g., guesses = 0).
3. Use a *while(1)* loop to repeatedly ask the player for their guess.
4. Compare the player's guess with the generated number.
5. Provide feedback to the player, such as "Too high" or "Too low," until they guess the correct number.
6. Once the player guesses the correct number, print a congratulatory message along with the number of guesses it took them to win.

a. Start with writing the pseudocode and drawing a flowchart of the algorithm.
b. Discuss it with the student sitting next to you in this session.
c. Now write the C program for this game. Follow good programming practise and make sure the code is well commented and properly intended. Include a summary of the exercise in comments at the start of the code.

Hint:

You can use 'srand(time(NULL))' to seed the random number generator at the beginning of your program to ensure different numbers on each run. Following by using the following for getting a random value between the required range.

rand() % (max_number + 1 - minimum_number) + minimum_number

The output should be similar to the following:

```
Welcome to the Guess the Number game!
I've selected a random number between 1 and 100.
Can you guess what it is?

Enter your guess: 50
Too high! Try again.
Enter your guess: 25
Too low! Try again.
Enter your guess: 37
Too low! Try again.
Enter your guess: 45
Too high! Try again.
Enter your guess: 40
Too low! Try again.
Enter your guess: 42

Congratulations! You guessed the number 42 in 6 tries.
```

Exercise 5: factorial of a positive integer

The factorial function is used frequently in probability problems. The factorial of a positive integer n (written $n!$ and pronounced "n factorial") is equal to the product of the positive integers from 1 to n.

Write a C program that calculates the factorial of a positive integer entered by the user using a *do-while* loop. Implement error handling to ensure that the user inputs a valid positive integer.

Expected output:

```
Factorial Calculator
Enter a positive integer: 5
The factorial of 5 is 120.
```

```
Factorial Calculator
Enter a positive integer: -7
Error: Invalid input. Not a positive integer.
Enter a positive integer: 7
The factorial of 7 is 5040.
```