Lab Hand-out 4.5

Loop Exercise

For loop exercises

1. **Sum of natural numbers:** Write a program that asks the user for a positive integer `n`, and then uses a `for` loop to calculate and display the sum of all natural numbers from 1 to `n`.
2. **Multiplication table:** Create a program that prompts the user for a number and displays its multiplication table up to 10. For example, if the user enters 5, the output should show `5 x 1 = 5`, `5 x 2 = 10`, and so on.
3. **Factorial calculation:** Write a program that takes a non-negative integer from the user and calculates its factorial using a `for` loop. The factorial of `n` is the product of all integers from 1 to `n`.
4. **Count even numbers:** Write a program that takes two integers from the user, a starting number and an ending number. Use a `for` loop to count how many even numbers are in that range (inclusive) and display the count.
5. **Nested loop pattern:** Write a program that uses nested `for` loops to print a right-angled triangle pattern of asterisks. The number of rows should be determined by user input.

   Input: 4

   Output:

   ```
   *
   **
   ***
   ****
   ```

6. **Display characters of a string:** Write a program that asks the user for a word or a sentence. Use a `for` loop to print each character of the string on a new line.
7. **Sum of odd numbers:** Write a program that takes an integer `n` and finds the sum of the first `n` odd natural numbers. Use a `for` loop for this.

While loop exercises

8. **Sum of positive integers:** Write a program that repeatedly asks the user to enter a positive integer and calculates the sum of all entered numbers. The loop should terminate when the user enters a negative number.
9. **Count digits:** Create a program that takes an integer from the user and uses a `while` loop to count the number of digits in that integer.
10. **Sum of digits:** Create a program that prompts the user for an integer and uses a `while` loop to calculate the sum of its digits. For example, if the input is `123`, the output should be `6`.
11. **Find GCD:** Write a program to find the Greatest Common Divisor (GCD) of two numbers entered by the user. You can use a `while` loop.
12. **Count a specific character:** Write a program that asks for a sentence and a single character. Use a `while` loop to count how many times that character appears in the sentence.
13. **Divisible by 9:** Write a program to find the numbers and their sum between 100 and 200 that are divisible by 9. Use a `while` loop for this.

Do-while loop exercises

14. **Simple menu:** Create a program that presents a menu of options to the user (e.g., "1. Continue", "2. Exit"). Use a `do-while` loop to ensure the menu is displayed and the program runs at least once before checking the user's input to continue or exit.
15. **Input validation:** Prompt the user to enter a number between 1 and 10. Use a `do-while` loop to continue asking for input until a valid number is provided.
16. **Countdown timer:** Write a simple program that asks the user for a number to start a countdown. Use a do-while loop to print every number from the one the user entered down to 1, followed by a final message like "Blast Off!".
17. **Process positive integers:** Write a program that asks the user to enter positive integers to find the count, maximum, minimum, and average. The process terminates when the user enters -1, and the negative number is not included in the calculations.
18. **Rewriting a `while` loop:** Take an existing `while` loop that validates user input for a positive number and rewrite it using a `do-while` loop.

Mixed loop exercises

19. **Power of a number:** Write a program that calculates x^y, where `x` and `y` are positive integers provided by the user. Use a loop to multiply `x` by itself `y` times.
20. **Pyramid of numbers:** Write a program using nested loops to print a pyramid pattern of numbers. For example, for 4 rows:

```
   1
  2 3
 4 5 6
7 8 9 10
```