Lab Session 4:

Part A:

Exercise 1: Loops

Write a program that takes a integer amount as input from the user and count the number of digits in it.

E.g. if the user enters 5463, the number of digits are 4.

Hint: How about using '/' operator!

Exercise 2: Functions

Write a C program that has functions to calculate area of a square, a circle and a rectangle. Take user for which area is needed to be calculated, parameter (length or side of square, radius of circle, etc), and print the resulting area. Use switch to handle the three area conditions.

Exercise 3: Fibonacci sequence

The Fibonacci series is a sequence of numbers where each number is the sum of the two preceding ones, usually starting with 0 and 1. The sequence begins 0, 1, 1, 2, 3, 5, 8, 13, 21, and so on.

Write a C program that takes the number of terms in the Fibonacci series as input from the user and output the series upto the entered term.

(a) First use standard function to program the series.
(b) User recursion function to program the series.

The output should look like the following:

```
Enter the number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13
```

Exercise 4: Password checker

Write a C program that takes a username and password as input and checks whether the password is valid based on the following criteria:

Password must contain at least one uppercase letter, one lowercase letter, one digit, and one special character, with a minimum length requirement of 8 characters.

Your program should display whether the entered password is valid or not.

Hint: Check strlen() function and the ctype.h header functions (HERE)

Use of pointers is not allowed for this exercise!

Part B

Mini Project

Note: You are required to show the output of this exercise to one of the TA at the start of the next lab session (13th Nov). This is not graded but we must log that you have completed this exercise.

Write a C program that reads an integer amount (between 1 – 1000000) from the user and converts it into its textual representation in words.

For example,

- If the user enters "11, the program should output “eleven”.
- If the user enters "199’, the program should output “one hundred ninety nine”.
- If the user enters "1234", the program should output "one thousand two hundred thirty four".
- If the user enters "53245, the program should output “fifty three thousand two hundred forty five”.
- If the user enters " 1005010, the program should output “one million five thousand ten”.

Have a look at this number to word calculater to see what the output should be of your C program: https://www.calculatorsoup.com/calculators/conversions/numberstowords.php

Requirements:

- Use control flow statements (if-else, switch, loops) where necessary.
- Implement the conversion logic in one or more functions (as necessary).
- You can use arrays and strings to store and manipulate the textual representation of numbers (if needed)
- **Use of pointers is not allowed for this exercise!**

Hints:

- Break down the integer into groups of three digits (ones, thousands, millions, etc.) and convert each group to words separately.
- Use arrays or switch statements to map digit values to their textual representations.
- Handle special cases for numbers less than 20 (e.g., "eleven," "twelve") and multiples of 10 (e.g., "twenty," "thirty").
- Combine the textual representations of the individual groups to form the complete text.