Objective:

- To get a grip on problem solving involving repetition structure.

Devise Solution of the following problems using flow charts/Pseudo Code

1. Write a pseudo code that asks the user to enter the amount that he or she has budgeted for a month. A loop should then prompt the user to enter each of his or her expenses for the month and keep a running total. When the loop finishes (decide a sentinel value yourself), the program should display the amount that the user is over or under budget.

2. Write a pseudo code that requests the user to enter two integers. The program should then calculate and report the sum of all the integers between and including the two integers. At this point, assume that the smaller integer is entered first. For example, if the user enters **2** and **9**, the program should report that the sum of all the integers from 2 through 9 is 44.

3. In a sumac sequence, t_1, t_2, …, t_m, each term is an integer greater than or equal 0. Also, each term, starting with the third, is the difference of the preceding two terms (that is, $t_{n+2}=t_n-t_{n+1}$ for $n\geq 1$). The sequence terminates at tm if $t_{m-1}<t_m$.
For example, if we have 120 and 71, then the sumac sequence generated is as follows:

120, 71, 49, 22, 27.

This is a sumac sequence of length 5.

4. Write a program which calculates the sum of following series:

$1^2 + 2^3 + 3^4 + \ldots + N^{N+1}$

You will take value of 'N' from user.

5. Write a program which calculates the sum of first 'N' term of the following series:

$1^{S+1} + 2^{S+1} + 3^{S+2} + 2^{S+3} + 3^{S+5} + 2^{S+8} + 3^{S+13} + \ldots\ldots$

You will take value of 'N' and 'S' from user.

6. Write a program, which prints all the prime numbers in the range of two given numbers m and n.

7. Display the prime factors of a given positive integer.

8. Write a program which computes the following

$$4 * \sum_{k=1}^{10^6} \frac{(-1)^{k+1}}{2k-1} = 4 * (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots)$$

9. Write a pseudo code that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month.
After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

Input Validation:

Do not accept a number less than 1 for the number of years.
Do not accept negative numbers for the monthly rainfall

10. Strong numbers are the numbers whose sum of factorial of digits is equal to the original number. Example: 145 is a strong number because 1! + 4! + 5! = 145
Your task is to write a pseudo which checks whether a given number is strong number or not.

11. A dual prime is 2 prime numbers that are exactly "2" apart. Example: 3, 5 and 11,13, etc. In this problem, you need to display all the dual primes up to a given number 'N'.

12. Given a number, your pseudo code will find and display the longest consecutive repeating digit.
Sample Input: 1999161117
Sample Output: 9

13. A high school has 1000 students and 1000 lockers, one locker for each student. On the first day of school, the fitness coach plays the following game:
She asks the first student to go and open all the lockers. She then asks the second student to go and close all the even-numbered lockers. The third student is asked to check every third locker. If it is open, the student closes it; if it is closed, the student opens it. The fourth student is asked to check every fourth locker. If it is open, the student closes it; if it is closed, the student opens it. The remaining students continue this game. In general, the nth student checks every nth locker. If the locker is open, the student closes it; if it is closed, the student opens it. After all the students have taken their turn, some of the lockers are open and some are closed.
Write a program that prompts the user to enter the number of lockers in a school. After the game is over, the program outputs the number of lockers that are opened. Test run your program for the following inputs: 1000, 5000, 10000. Do you see any pattern developing? Look at the following hint if stuck badly ☺
(*Hint*: Consider locker number 100. This locker is visited by student numbers 1, 2, 4, 5, 10, 20, 25, 50, and 100. These are the positive divisors of 100. Similarly, locker number 30 is visited by student numbers 1, 2, 3, 5, 6, 10, 15, and 30. Notice that if the number of positive divisors of a locker number is odd, then at the end of the game, the locker is opened. If the number of positive divisors of a locker number is even, then at the end of the game, the locker is closed.)