CL-1002
Programming Fundamentals
Lab # 15

Objectives:

- Practice and understanding on basic C++ programs
- User defined Functions
- Pass by Value
- Pass by reference
- Arrays
- Passing array as argument to function
- File Handling

Note: Carefully read the following instructions (*Each instruction contains a weightage***)**

1. First think about statement problems and then write your program.
2. Write Program in C/C++ compiler/IDE and save source file **for each program.**
3. Do not copy from any source otherwise you will be penalized with negative marks.
4. Complete your lab **within given Time Slot**.
5. Add your source code in this word document + Make one ZIP file of your all source codes.
6. Please submit your **Both files** with this naming convention ROLLNO_SECTION_LABNO.
7. Submit your lab on Google Classroom.

Problem 1:

Write a C++ program to generate 100 random numbers between 0 and 1000, sort them in ascending order, and write them to a file named "sortedNumbers.txt".

- Use the C++ rand() function to generate 100 random numbers. Sort the numbers in ascending order using a sorting algorithm of your choice.
- Use file handling to write the sorted numbers to a file named "sortedNumbers.txt".
- Display a message to the user once the sorted numbers have been written to the file.

Problem 2:

Write a C++ program that allows the user to input an array of 10 integers, an old value, and a new value. The program should then search for all occurrences of the old value in the array and replace them with the new value. If the old value does not exist in the array or the array is empty, the program should output an appropriate message.

- Use an array of 10 integers to store the user input.
- Prompt the user to input an old value and a new value.
- Search for all occurrences of the old value in the array and replace them with the new value.
- Output an appropriate message if the old value does not exist in the array or the array is empty.

Assumptions:

- The program should be able to handle multiple occurrences of the old value in the array.
- The program does not need to preserve the order of the array after replacing the old value with the new value.

Problem 3:

Write a C++ program that will create two same size arrays of Size 10, then you need to fill the both arrays by taking values from user. You need to pass these two arrays to function

int SimilarCheck(int[ ], int[ ]) that will compare the both elements of arrays with each other's and should return the count of elements which are common in both arrays.

Problem 4:

Write a C++ program to determine whether a given integer is an Armstrong number or a perfect number. To accomplish this, write two functions Armstrong_num() and perfect_num(), both of which take an integer argument and return a boolean value. The Armstrong_num() function determines whether the given number is an Armstrong number, while the perfect_num() function determines whether the given number is a perfect number.

To start, the program should prompt the user to enter a number. Then, the program should call both Armstrong_num() and perfect_num() functions to determine whether the entered number is an Armstrong number or a perfect number. Finally, the program should output the result to the user.

Note: An Armstrong number is a number where the sum of each digit raised to the power of the number of digits is equal to the original number. A perfect number is a number where the sum of its factors (excluding itself) is equal to the original number.

Problem 5:

Write a C++ program that reads integer values from a file named "numbers.txt" and stores them in an integer array. The program should then display the sum of all the numbers in the array. The program should have a function named readFile that reads the values from the file and stores them in the array. The readFile function should take the array as a parameter and use pass by reference to modify the array elements. The program should also have a function named sumArray that takes the array as a parameter and calculates the sum of all the elements in the array using pass by value. Finally, the program should display the sum of all the numbers in the array.

The program should include the following:

- A function named readFile that reads values from "numbers.txt" and stores them in an array using pass by reference.
- A function named sumArray that takes the array as a parameter and calculates the sum of all the elements in the array using pass by value.
- The program should use an array to store the values read from the file.
- The program should use file handling to read the values from the "numbers.txt" file.
- The program should output the sum of all the numbers in the array.