1) Write a C++ Program to Multiply Two Matrix Using Multi-Dimensional Arrays. This program takes two matrices of order r1*c1 and r2*c2 respectively. Then, the program multiplies these two matrices (if possible) and displays it on the screen.

2) Write a function to find the norm of a matrix. The norm is defined as the square root of the sum of squares of all elements in the matrix

3) We are given two sorted arrays (all elements are in ascending order. We need to merge these two arrays such that the initial numbers (after complete sorting) are in the first array and the remaining numbers are in the second array.

4) Given an unsorted array of size n. Array elements are in the range from 1 to n. One number from set {1, 2, …, n} is missing and one number occurs twice in the array. Find these two numbers.

5) An array my_Array**[]** consisting 'a' s, 'b' s and 'c's. The task is to write a function that arranges the array such that all 'a's are placed first, then all 'b's and then all 'c's in last.

6) Write a Menu Driven C++ program that creates one-dimensional array arr[] and initialize it with user. The program should do following Tasks using Menu, the menu operations are implemented using functions:

   a) Write a function C**ount(),** that counts the occurrences of x (a number) in arr[].
   b) Write a function **Partition(),** that take the first element of the array x and put x in a position such that all smaller elements (smaller than x) are before x, and put all greater elements (greater than x) after x.
   c) Write a function **next_XOR(),**the count of elements which are equal to the XOR of the next two elements.
   d) Write a function Duplicates(),which calculated the frequencies of all the elements and display them.
   e) Write a function **Circular(),**which replace every element of the array by the sum of next two consecutive elements in a **circular** manner i.e. arr[0] = arr[1] + arr[2], arr[1] = arr[2] + arr[3], … arr[n – 1] = arr[0] + arr[1].
   f) Write a function **Search(),** takes an array and element to search in the array and returns the index of element if the element is found. And return the negative number if not found.
   g) Write a function **shift_Circular (),** which shifts an array circularly left by two positions. Thus, if p[0] = 15, p[1]= 30, p[2] = 28, p[3]= 19 and p[4] = 61 then after the shift p[0] = 28, p[1] = 19, p[2] = 61, p[3] = 15 and p[4] = 30.

7) Write a Menu Driven C++ program that creates a two-dimensional array/Matrix of size **n X m where n represent the students and m represent the subjects** and initialize it with user. The program should do following Tasks using Menu, the menu operations are implemented using functions:

   a) **Total_Marks**: Calculates total/sum of the values in the specified row (student).

   b) **Avg_Subject**: Calculates Average of the values in the specified column (subject).

   c) **Stud_Highest**: Finds highest value in the specified row of the array and return that subject name.

   d) **Stud_Lowest:** Finds lowest value in the specified row of the array and return that subject name.

8) Write a Menu Driven C++ program that creates a character array/string by taking input from user and perform following tasks by displaying menu to user, the menu operations are implemented using functions:

   a) Calculate length of string.

   b) Count number of words in string.

   c) Check a string is palindrome or not.

   d) Find a word within the array. If found display its starting position.

   e) Convert a string in lowercase.

   f) Convert a string in uppercase.

**Note**: Make all code separately and then merge them all in a menu. Use switch statement for menu.

9) Implement the Tic-Tac-Toe game in C++ using functions and 2D array of 3X3 size. Rules of the Game

   a) The game is to be played between computer and user.
   b) One of the players chooses ‘O’ and the other ‘X’ to mark their respective cells.
   c) The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character (‘O’ or ‘X’).
   d) If no one wins, then the game is said to be draw.

10) Solving system of linear equations with two variables using Cramer’s rule. In mathematics the coefficients of linear equations are represented in the form of matrices. A matrix is a two-dimensional array in programming. The steps of Cramer’s rule to solve the system of equations are:

**Step 1**: Determining the coefficient matrix from linear equations e.g.

a1x+b1y=d1

a2x+b2y=d2

Coefficient matrix: D = $\begin{array}{|c|c|} \hline a1 & b1 \\ \hline a2 & b2 \\ \hline \end{array}$

**Step 2**: Determining the constant column C = $\begin{array}{|c|} \hline d1 \\ \hline d2 \\ \hline \end{array}$

**Step 3**: Finding determinant |D| = (a1*b2) - (b1*a2) and checking if |D| is not equal to zero then do the following:

1. Determining X-matrix: DX = $\begin{array}{|c|c|} \hline d1 & b1 \\ \hline d2 & b2 \\ \hline \end{array}$

   The coefficients of the x–column are replaced by the constant column

2. Determining Y-matrix: Dy =

| | |
|---|---|
| a1 | d1 |
| a2 | d2 |

   The coefficients of the Y−column are replaced by the constant column

3. To solve for x: x=|DX|/|D|

4. To solve for y: y=|Dy|/|D|

   **Note** that if |D|=0 then the matrix is singular and solving the system of equation is not possible.

You're required to automate all these steps by writing a program in C++ using two dimensional arrays where required. Ask user to enter elements for coefficient matrix D and constant column C. Once the D and C are populated you should perform all steps above and display the values for x and y to the user. Note that in the case of a singular matrix, display a message for the system of equations can't be solved.

Rubrics

- The elements of coefficient matrix and constant column must be taken dynamically
- Use of single structure of nested loop to find both x and y matrix
- Use of single structure of nested loop to find determinant of both x and y matrix

11) Write a program to ask the user for the number of arrays he wants to maintain and then populate each array by taking elements from the user. Merge all these arrays in another array and finally remove duplicate elements from the merged array.

12) Consider an integer array, the number of elements in which is determined by the user. The elements are also taken as input from the user. Write a program to find those pairs of elements that have the maximum and minimum difference among all element pairs.