Course Name:	**Structured Programming Methodology**	**Semester:**	**I**
Date of Performance:	**30 /9/2025**	**DIV/ Batch No:**	**A1**
Student Name:	**Arav Arun**	**Roll No:**	**16010125013**

Experiment No: 4

Title: Write a program in C++ to demonstrate the use of Array

Aim and Objective of the Experiment:

Write a C++ program to demonstrate the concept and usage of arrays.

a. To understand what arrays are and why they are used.
b. To learn how to declare, initialize, and access array elements.
c. To perform operations like input, display, and computation using arrays.

COs to be achieved:

CO: Apply the concepts of arrays and strings. (CO3).

Theory:

An array is a collection of elements of the same data type stored in contiguous memory locations. Arrays help in storing multiple values under a single name, which can be accessed using indices (starting from 0).

Types of Arrays:

1. **One-dimensional Array:** Linear list of elements (e.g., list of marks).
2. **Two-dimensional Array:** Tabular form of data (e.g., matrix).
3. **Multi-dimensional Array:** Arrays with more than two dimensions.

Syntax for Array Declaration:

```
data_type array_name[size];              // One-dimensional Array
data_type array_name[size][size];        // Two-dimensional Array
```

Example:

```
int arr[5];          // Declares an integer 1D-array with 5 elements.
Int arr[3][3];       // Declares an integer 2D-array with 9 elements.
```

Algorithm (for 1-D Array Demonstration):

4. Start the program.
5. Declare an array of required size.
6. Read `n` elements from the user and store them in the array.
7. Display the array elements.
8. Perform a sample operation (e.g., find sum of elements).
9. Display the result.

10. End the program.

Algorithm (for 2-D Array Demonstration):

1. Start the program.
2. Declare a 2D array of appropriate size.
3. Read the number of rows and columns from the user.
4. Input elements into the 2D array using nested loops.
   Outer loop → for rows
   Inner loop → for columns
5. Display the array elements in matrix form using nested loops.
6. Perform a sample operation (e.g., find sum of elements).
7. Display the result.
8. End the program

Problem Statements:

1. C++ program to count total number of elements divisible by a specific number in an array
   Input:
   Enter array elements:
   10
   15
   20
   25
   30
   Number: 10
   Output:
   Total elements divisible by 10 is 3

2. C++ Program to find the Largest Element in a Matrix
   Input: {12, 25, 6},
   {8, 15, 20},
   {3, 50, 10}
   Output: 25

Code :

Program 1 :

```
#include <iostream>
using namespace std;

int main()
```

```
{
   int n=0, divCount=0,num=0,i=0;
   cout << "Enter number of array elements to input  : " << endl;
   cin>>n;
   int arr[n];
   cout<<"Enter "<<n<<" array elements : "<<endl;
   for(int i = 0;i<n;i++)
   {
      cin>>arr[i];
   }
   cout<<"Enter number to check divisibility of array elements from : "<<endl;
   cin>>num;

   for(i = 0;i<n;i++)
   {
      if(arr[i]%num==0)
         divCount++;
   }
   cout<<"Total number of array elements divisible by "<<num<<" : "<<divCount<<endl;
   return 0;
}
```

Program 2 :

```
#include <iostream>
using namespace std;

int main() {
   int rows, cols;
   cout << "Enter number of rows: ";
   cin >> rows;
   cout << "Enter number of columns: ";
   cin >> cols;

   int matrix[100][100];
   cout << "Enter the elements of the matrix : "<<endl;

   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         cin >> matrix[i][j];
      }
   }

   int largest = matrix[0][0];
```

```
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] > largest) {
                largest = matrix[i][j];
            }
        }
    }

    cout << "Largest element in the matrix is : "<< largest << endl;
    return 0;
}
```

Output:

Output for program 1 :

```
Enter number of array elements to input  :
5
Enter 5 array elements :
10
15
20
25
30
Enter number to check divisibility of array elements from :
10
Total number of array elements divisible by 10 : 3
```

Output for program 2:

```
Enter number of rows: 3
Enter number of columns: 3
Enter the elements of the matrix :
12
25
6
8
15
20
3
50
10
Largest element in the matrix is : 50

Process returned 0 (0x0)   execution time : 42.081 s
Press any key to continue.
```

Post Lab Subjective/Objective type Questions:

1. C++ program to find Smallest and Largest elements from One Dimensional Array Elements

Sol :

```
#include <iostream>
using namespace std;

int main() {
   int n;
   cout << "Enter number of elements in array : ";
   cin >> n;

   int arr[n];

   cout << "Enter array elements  : " << endl;
   for (int i = 0; i < n; i++) {
      cin >> arr[i];
   }

   int smallest = arr[0];
   int largest = arr[0];

   for (int i = 1; i < n; i++) {
      if (arr[i] < smallest)
         smallest = arr[i];
      if (arr[i] > largest)
         largest = arr[i];
   }

   cout << "Smallest element of array : " << smallest << endl;
   cout << "Largest element of array : " << largest << endl;
   return 0;
}
```

Output :

```
Enter number of elements in array: 6
Enter array elements :
5
7
9
12
169
450
Smallest element of array : 5
Largest element of array : 450

Process returned 0 (0x0)   execution time : 14.111 s
Press any key to continue.
```

2. Write a C++ program that Sum of Rows and Columns of a Matrix

Sol:

```
#include <iostream>
using namespace std;

int main() {
   int rows, cols;
   cout << "Enter number of rows: ";
   cin >> rows;
   cout << "Enter number of columns: ";
   cin >> cols;

   int matrix[100][100];
   cout << "Enter elements of the matrix:\n";
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         cin >> matrix[i][j];
      }
   }

   cout << "\nSum of each row:\n";
   for (int i = 0; i < rows; i++) {
      int rowSum = 0;
      for (int j = 0; j < cols; j++) {
         rowSum += matrix[i][j];
      }
      cout << "Row " << i + 1 << " = " << rowSum << endl;
   }

   cout << "\nSum of each column:\n";
   for (int j = 0; j < cols; j++) {
      int colSum = 0;
      for (int i = 0; i < rows; i++) {
         colSum += matrix[i][j];
      }
      cout << "Column " << j + 1 << " = " << colSum << endl;
   }
   return 0;
}
```

Output:

```
(base) ➜  SPM_4 ./main
Enter number of rows: 3
Enter number of columns: 3
Enter elements of the matrix:
1
2
3
4
5
6
7
8
9

Sum of each row:
Row 1 = 6
Row 2 = 15
Row 3 = 24

Sum of each column:
Column 1 = 12
Column 2 = 15
Column 3 = 18
(base) ➜  SPM_4
```

Conclusion:

Through this experiment, I got a clear idea of how arrays work in C++. I learned how to declare and use them, take input, display values, and perform simple operations. Working with both 1-D and 2-D arrays also showed me how useful they are for solving real problems like finding sums, largest elements, or numbers divisible by a given value. Overall, it gave me practical hands-on experience with arrays and made the concept much easier to understand.

Signature of faculty in-charge with Date: