LAB # 12
Arrays

Objective:

1. To introduce and allow students to work with arrays
2. To introduce the typedef statement
3. To work with and manipulate multidimensional arrays

Theory:

Arrays

Arrays are a consecutive group of variables (memory locations) that all have the same type and share a common name. In many applications, arrays are used to represent vectors and matrices.

An array is declared by writing the type, followed by the array name and size (number of elements in the array) surrounded by square brackets

type array-name [number-elements];

float position[3];

int count[100];

char YourSurname[50];

Array elements and indexing

To refer to a particular element of the array we specify the name of the array and the position number or index of the particular element. Array elements are counted from 0 and not 1. The first elements will always have position and index 0 and the last element will have index number (position) N-1 if the array has N elements

Assigning values to array elements

Elements of an array can be treated like other variables but are identified by writing the name of the array followed by its index in square brackets. So for instance the statements:

marks[1] = 90.0; scaled[1] = (marks[1] - mean)/deviation;

show how the second element of an array called marks is assigned a value and how the second element of the array scaled is assigned the result of a calculation using this element value.

Example:

```
// ScalarProduct.cc
// Calculating the scalar product between vectors input by user

#include <iostream>
using namespace std;

int main()
{
  float vectorA[3], vectorB[3], scalar=0.0;
  int i;

  // Get input vectors from user.
  cout << "Enter elements of first vector: " << endl;
  for(i=0;i<3;i++)
  {
    cin >> vectorA[i];
```

```
  }
  cout << "Enter elements of second vector: " << endl;
  for(i=0;i<3;i++)
  {
    cin >> vectorB[i];
  }

  // Calculate scalar product.
  for(i=0;i<3;i++)
  {
    scalar = scalar + (vectorA[i] * vectorB[i]);
  }

  // Output result.
  cout << "The scalar product is " << scalar << endl;
  return 0;
}
```

Lab Task:

Do task 12.1, 12.2 and 12.3 given in file attached on LMS and attach your source code and output window with this file. And write your observation with lab task.

Post Lab:

Write a program that will input temperatures for consecutive days. The program will store these values into an array and call a function that will return the average of the temperatures. It will also call a function that will return the highest temperature and a function that will return the lowest temperature. The user will input the number of temperatures to be read. There will be no more than 50 temperatures. Use typedef to declare the array type. The average should be displayed to two decimal places.

Sample Run:
Please input the number of temperatures to be read
5
Input temperature 1:
68
Input temperature 2:
75
Input temperature 3:
36
Input temperature 4:
91
Input temperature 5:
84
The average temperature is 70.80
The highest temperature is 91.00
The lowest temperature is 36.00

Learning Outcomes:

Upon successful completion of the lab, students will be able to:

LO1: To work with arrays, user defined data types and multi dimension arrays.