Arrays

1. Arrays

The array is another data type for storing values. Unlike an integer variable where it can store only one value at a time, an array can store many values in different locations. An index is used to access the different locations in an array. An array of size 10 has index locations from 0 to 9.

To declare an integer array called n having 10 locations:

```cpp
int n[10];
```

To assign the number 13 into location 2 of the array:

```cpp
n[2] = 13;
```

```cpp
int main () {
    int n[10]; // declare an array called n of size 10
    int m[5] = {8, 5, 4, 3, 9}; // initialize array m with five numbers
    int i;

    n[0] = m[1] + 2; // accessing values in the array
    cout << n[0]; // the number 7 will be printed out

    //input 10 values into the 10 locations of the array n
    for(i=0; i<10; i=i+1){
        cin >> n[i];
    }

    //print the 10 values from the 10 locations of the array
    for(i=0; i<10; i=i+1){
        cout << n[i];
    }

    return 0;
}
```

2. 2-dimensional Arrays

To declare and initialize a 2-dimensional array having 2 rows and 5 columns:

```cpp
int n[2][5] = {{2,4,6,8,10},{1,3,5,7,9}};
```

To access row 1 column 3 of the array:

```cpp
if (n[1][3] > 5) ...
3. **Exercises** (Problems with an asterisk are more difficult)

1. Write a program using a loop to input 10 numbers. After you have entered these 10 numbers, then the program will print out these 10 numbers.

2. Write a program with an array of size 10. Initialize the array with numbers from 1 to 50 of your choice. Now calculate and print out the sum of these numbers.

3. Write a program with an array of size 10. Initialize the array with numbers from 1 to 50 of your choice. Now calculate and print out the average of these numbers.

4. Write a program to generate 10 random numbers from 1 to 50. Store these 10 numbers in an array. Now find and print out the largest of these numbers.

5. Write a program to enter one number and then generate the multiplication table from 1 to 10 for that one number. Put these numbers into an array of size 10. After that, print out the table from the array.

6. Write a program with an array of size 10. Initialize the array with numbers of your choice. Have the user enter a number, and then search for this number in the array. Print out the message “number found” if the number entered is found in the array.

7. * Repeat question 6 but also print out the message “number NOT found” if the number is not found in the array.

8. * Create a 10×10 array and fill it with the appropriate numbers for the multiplication table. You need to use a two-dimensional array.

9. ** Write a program with an array of size 10. Initialize the array with numbers of your choice but not in any order. Now print these numbers in ascending order.

10. ** Write a program to generate 11 random numbers from 1 to 50. Store these 11 numbers in an array. Now calculate and print out the median of these numbers.

11. ** Use the Sieve of Eratosthenes algorithm for finding all the prime numbers from 2 to 500. The algorithm:

   1. Create a list of consecutive integers from two to \( n \): (2, 3, 4, ..., \( n \)).
   2. Initially, let \( p \) equal 2, the first prime number.
   3. Cross out from the list all multiples of \( p \) greater than \( p \).
   4. Find the first number remaining in the list greater than \( p \) (this number is the next prime); let \( p \) equal this number.
   5. Repeat steps 3 and 4 until \( p^2 \) is greater than \( n \).
   6. All of the remaining numbers in the list are prime.

   Hint: For step 1, you will need to use an array to keep track of whether a number has been crossed out or not. The indexes for the array are the integers. Initialize the array with 0’s. A 0 in a location means that that number is not crossed out, and a 1 in that
location means that that number has been crossed out. For step 3, you can cross out a number simply by changing the content of that number index location to a 1.