Practice Exercises – Chapter: 07

* Exercise 7.1

Write a program that asks for the number of hours worked by six employees. It stores the values in an array.

Solution 7.1 (a)

```
#include <iostream>
using namespace std;
int main()
    {
      const int NUM_EMPLOYEES = 6;
      int hours[NUM_EMPLOYEES];

      // Get the hours worked by each employee.
      cout << "Enter the hours worked by " << NUM_EMPLOYEES << "
employees: ";
      cin >> hours[0];
      cin >> hours[1];
      cin >> hours[2];
      cin >> hours[3];
      cin >> hours[4];
      cin >> hours[5];

     //Display the values in the array.
     cout << "The hours you entered are:”;
     cout << " " << hours[0];
     cout << " " << hours[1];
     cout << " " << hours[2];
     cout << " " << hours[3];
     cout << " " << hours[4];
     cout << " " << hours[5] << endl;
     return 0;
   }
```

Solution 7.1 (b):

```
#include <iostream>
using namespace std;
int main()
    {
     const int NUM_EMPLOYEES = 6; // Number of employees
     int hours[NUM_EMPLOYEES]; // Each employee's hours
     int count; // Loop counter

     // Input the hours worked.
     for (count = 0; count < NUM_EMPLOYEES; count++)
        {
          cout << “Enter the hours worked by employee ” <<
     (count + 1) << ": ";
          in >> hours[count];
        }
     //Display the contents of the array.
     cout << "The hours you entered are:";
     for (count = 0; count < NUM_EMPLOYEES; count++)
            cout << " " << hours[count];
     cout << endl;
     return 0;
     }
```

* Exercise 7.2: Find the largest element in the array

Write a program that allows declaring an integer array, the number of elements (size of the array) and values for its elements. Then find and display the largest element of the array.

Solution 7.2:

```
#include<iostream>

using namespace std;

int Largest_Element (int[], int);


int main()

{

    int i, a[50], size;

    // Enter the size of the array

    cout<<"Enter array size (Max:50): ";

    cin>>size;

    //check whether size < 1 or not

    if (size<1)

        {
          cout<<"\nThis is out of range.\n";
          exit(0);
        }
    // Enter element’s values of the array
    cout<<"\nEnter value of each element: \n";
    for (i=0; i<size; i++)
        {
            cout<<"\nEnter value of element arr["<<i<<"] : ";
            cin>>a[i];
        }
    // List the element’s values of the array
    cout<<"\nThe values in array are: \n\n";
    for (i=0; i<size;i++)
    {
     cout<<" "<<a[i]<<" ";
    }

    //Call the function to find the maximum value
    int largest_element = Largest_Element (a, size);

    cout<<"\n\nLargest Element’s value in an Array is:
"<<largest_element<<endl;
    return 0;
}

// Function for finding the largest element in array
int Largest_Element (int arr[], int k)
{
    int largest=arr[0];
    int i;
        for (i=0; i<k; i++)
        {
                if (arr[i]>largest)
                {
                    largest=arr[i];
                }
        }
return (largest);
}
```

*** Exercise 7.3: Delete an element in an array**

Write the program to ask to the user to enter the array size and array elements, and then ask the user to enter the position (not index) to delete the element. Display the elements of the array after deletion.

Solution 7.3

```
#include<iostream>
using namespace std;
int * FindandDelete_Element (int[], int, int);
void List_Elements (int[], int);

int main()
{
    int i, a[50], pos, size;
    // Enter the size of the array
    cout<<"Enter array size (Max:50): ";
    cin>>size;
    //check whether size < 1 or not
    if (size<1)
        {
         cout<<"\nThis is out of range.\n";
         exit(0);
        }
    // Enter element’s values of the array
    cout<<"\nEnter value of each element: \n";
    for (i=0; i<size; i++)
        {
            cout<<"\nEnter value of element arr["<<i<<"] : ";
            cin>>a[i];
        }
    // List the element’s values of the array

     cout<<"\nThe values in array are: \n\n";
     List_Elements (a,size);

    // Enter the position (not index) to delete
    cout<<"\n\nEnter position (not index) to delete the element::
";
    cin>>pos;
    if (pos<1 or pos > size)
        {
         cout<<"\nThis is out of range.\n";
         exit(0);
        }

    //Call the function to find and delete element
    int * New_Array = FindandDelete_Element (a, size, pos);

    //List the element’s values of the new array
    cout<<"\nThe values in new array are: \n\n";
    List_Elements (New_Array,size-1);

return 0;
}
// Function for listing the elements of the array
void List_Elements (int arr[],int j)
{
    int k;
    for (k=0; k<j;k++)
    {
     cout<<" "<<arr[k]<<" ";
    }
}

// Function for deleting the element in array
int * FindandDelete_Element (int arr[], int l, int m)
{
    int n;
    if(m>l or m<1)
    {
        cout<<"\nThis is out of range.\n";
        exit(0);
    }
    else
    {
        --m; //calculate the index = pos -1;
        for(n=m;n<=l-1;n++)
```

```
         {
               arr[n]=arr[n+1];
         }
    }
return (arr);
}
```

* Exercise 7.4: Average of element’s values in an array

Write the program to ask to the user to enter the array size and array elements. Then display the elements of the array, calculate the average of element’s values of the array and display the average value.

Solution 7.4:

```
#include<iostream>
using namespace std;
float Average(float[], int);
void List_Elements (float[],int);

int main()
{
    int size, i;
    float a[100], average;
    // Enter the size of the array
    cout<<"Enter array size (Max:50): ";
    cin>>size;
    //check whether size < 1 or not
    if (size<1)
        {
          cout<<"\nThis is out of range.\n";
          exit(0);
        }
    // Enter element’s values of the array
    cout<<"\nEnter value of each element: \n";
    for (i=0; i<size; i++)
        {
            cout<<"\nEnter value of element arr["<<i<<"] : ";
            cin>>a[i];
        }
    // List the element’s values of the array
    cout<<"\nThe values in array are: \n\n";
    List_Elements (a,size);
     //Call the function to calculate the average of element’s values
    float average_values = Average(a, size);
     //Display the average of element’s values
    cout << "\nAverage of "<<size<<" numbers in array is = " << average_values<<"\n";

return 0;
}
// Function for listing the elements of the array
void List_Elements (float arr[],int j)
{
    int k;
    for (k=0; k<j;k++)
    {
     cout<<" "<<arr[k]<<" ";
    }
}
```

```
// Function for calculating the average of element’s values
float Average (float arr[], int j)
{
   int l;
   float average;
   float sum=0.0;
   for (l=0; l<j;l++)
      {
       sum+=arr[l];
      }
   average = sum/j;

return (average);
}
```

*** Exercise 7.5: Find the average of the largest and smallest element in the array**

Write a program that allows declaring an array, the number of elements (size of the array) and values for its elements. Then find calculate and display the average of the largest and the smallest element.

*** Exercise 7.6: Find the average of even and odd numbers in the array**

Write a program that allows declaring an array, the number of elements (size of the array) and values for its elements. Then calculate and display:

- The average of even number (values) of the array.

- The average of odd number (values) of the array.

*** Exercise 7.7: Find duplicate element in an array**

Write a program that allows declaring an array, the number of elements (size of the array) and values for its elements. Then find and display all values that appear 2 or more times in the array. For example, with the array [1 2 3 4 3 4 2 3 5 6], the result will be: 2 3 4

*** Exercise 7.8: Insert new element into an array**

Write a program that allows declaring an array, the number of elements (size of the array) and values for its elements. and then ask the user to enter the index to insert a new element. Display the elements of the array after insertion.

* Exercise 7.9: Delete the duplicate values in an array

Write a program that allows declaring an array, the number of elements (size of the array) and values for its elements. Then find and delete all values that appear from 2nd onwards in the array. Display the elements of the array after deletion. For example, with the array [1 2 3 4 3 4 2 3 5 6], the result will be: [1 2 3 4 5 6];

* Exercise 7.10: Multiply Two arrays

Write a program that allows declaring TWO arrays with the same number of elements (size of the array), then input the values for the two elements. Calculates and replaces the values of elements in the first array by multiplying the value of the element in the first array by the value of the element in the same position in the second array. Display the values of the elements of the first array after replacing. For example, with the first array [1 2 3 4] and the second array [2 3 4 5] the result will be: [2 6 12 20];

* Exercise 7.11 Two-dimension array:

Write a program that allows the user to:

- Declare an integer 2-dimensional array by declaring the size of the array (row and column);

- Enter the elements of the array from the keyboard;

- Display the elements of the array including the index (row and column) and the corresponding value of each element;

- Calculate and display the sum of value of elements in each row;

- Calculate the display sum of value of elements in each column.

For example: Assume that, you declare a 2-dimensional array A which consists of 2 rows and 3 columns. And you enter 6 values for the elements of the array as follows:

2

4

6

8

10

12

So, result of displaying the elements of the array including the index (row and column) and the corresponding value of each element is as following:

A[0][0] = 2 A[0][1] = 4   A[0][2] = 6

A[1][0] = 8 A[1][1] = 10 A[1][2] = 12

And, result of calculating of sum of element's values in each row and column as follows:

Calculate and display the sum of value of elements in each row:

Sum of value of elements in row 0: 12

Sum of value of elements in row 1: 30

Calculate and display the sum of value of elements in each column:

Sum of value of elements in column 0: 10

Sum of value of elements in column 1: 14

Sum of value of elements in column 2: 18

Solution 7.11

```
#include<iostream>
using namespace std;
void Display_Array (int[10][10], int, int);
void Row_Sum  (int[10][10], int, int);
void Col_Sum  (int[10][10], int, int);

int main()
{
    int row, col, i, j, arr[10][10];
    cout<<"Enter the Row Size for Array: ";
    cin>>row;
    cout<<"Enter the Column Size for Array: ";
    cin>>col;
    //Check whether size (row as well as column) < 1 or not
    if (row<1 or col<1)
        {
            cout<<"\nThis is out of range.\n";
            exit(0);
        }
    cout<<"Enter "<<i*j<<" Array Elements: \n";
    for(i=0; i<row; i++)
    {
        for(j=0; j<col; j++)
            cin>>arr[i][j];
    }
    cout<<"\nArray Elements with its Index:\n";
    Display_Array (arr, i, j);

    cout<<"\nCalculate and display the sum of value of elements in
each row:\n";
    Row_Sum (arr, i, j);

    cout<<"\nCalculate and display the sum of value of elements in
each column:\n";
    Col_Sum (arr, i, j);

    return 0;
}

void Display_Array (int arr[10][10], int row, int col)
{
    int m,n;
    for(m=0; m<row; m++)
    {
        for(n=0; n<col; n++)
            cout<<"arr["<<m<<"]["<<n<<"] = "<<arr[m][n]<<"  ";
        cout<<endl;
    }
    cout<<endl;
}

void Row_Sum (int arr[10][10], int row, int col)
{
    int m,n;
    for(m=0; m<row; m++)
    {
        int sumRow=0;
        for(n=0; n<col; n++)
            {
                sumRow += arr[m][n];
            }
            cout << "\nSum of value of elements in row  "<<m<<":
" << sumRow<<"\n";
        cout<<endl;
    }
    cout<<endl;
}

void Col_Sum (int arr[10][10], int row, int col)
{
    int m,n;
    for(n=0; n<col; n++)
    {
        int sumCol=0;
        for(m=0; m<row; m++)
            {
                sumCol += arr[m][n];
            }
            cout << "\nSum of value of elements in col  "<<n<<":
" << sumCol<<"\n";
        cout<<endl;
    }
    cout<<endl;
}
```

* Exercise 7.12 Two-dimension array:

Write a program that allows the user to:

- Declare an integer 2-dimensional array by declaring the size of the array (row and col);

- Enter the elements of the array from the keyboard;

- Calculate and display the sum of value of elements in row number *n*; *n* is entered from keyboard and *0<=n<row*;

- Calculate the display sum of value of elements in column number *m*; *m* is entered from keyboard and *0<=m<col*.

- Find and display the largest and smallest element in row number *n*; *n* is entered from keyboard and *0<=n<row*;

- Find and display the largest and smallest element in column number *m*; *m* is entered from keyboard and *0<=m<col*;

* Exercise 7.13: Exercise to do in the online system (online.vku.udn.vn)

Write a program to:

- Enter the positive integer n from the keyboard (n>0).
- Declare a 1-dimensional array of integer type (up to 50 elements). Enter values for the n elements of the array.
- Display the elements of the array.
- Find and display the largest number and its occurrences in the array.

Data (Input)

- The first line contains a positive integer n;
- The second line contains n integers, each number separated by space;

Result (Output)

- The first line contains n integers, each number separated by space;
- The second line contains the largest number;
- The third line contains the number of occurrences of the largest number;

Example

Input

```
6
5 8 3 11 9 11
```

Output

```
5 8 3 11 9 11
11
2
```

* Exercise 7.14: Exercise to do in the online system (online.vku.udn.vn)

Write a program to:

- Enter the positive integer n from the keyboard (n>0).
- Declare a 1-dimensional array of integer type (up to 50 elements). Enter values for the n elements of the array.
- Display the elements of the array.
- Displays the total number of prime numbers of the array.

Data

- The first line contains a positive integer n;

- The next n lines, each line contains 1 integer (values of array’s elements).

Result

- The first n lines, each line contain an integer (values of array’s elements);
- The (n+1) line contains the total number of prime numbers of the array.

Example

Input

```
6
5
8
3
11
9
11
```

Output

```
5
8
3
11
9
11
4
```