ASSIGNMENT-LAB 01

Course Code: CSE - 2322 **Course Title:** Data Structures (& Lab)
Course Teacher: Mohammed Shamsul Alam

Name: Ezaz Ahmed ID: C223009 Section: 3AM

Problem 01: Write a program to find the largest number and it's location from a given list of integers.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
    int num[] = {12, 45, 67, 23, 90, 31, 88};
    int n = sizeof(num) / sizeof(num[0]);
    int lar = num[0];
    int loc = 0;
    for (int i = 1; i < n; i++)
    {
        if (num[i] > lar)
        {
            lar = num[i];
            loc = i;
        }
    }
    cout << "The largest number is: " <<lar<<endl;
    cout << "The location of the largest number is: " <<
loc+1<<endl; // In Data Structure the index starts from 1 not 0.
    return 0;
}
```

Problem 02: Write a program to calculate the roots of the quadratic equation ax2 + bx + c = 0 where a, b and c are known.

Answer:

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    double a = 4.0;
    double b = -6.0;
    double c = -3.0;
    double D = b * b - 4 * a * c;

    if (D > 0)
```

```
    {
        double root1 = (-b + sqrt(D)) / (2 * a);
        double root2 = (-b - sqrt(D)) / (2 * a);

        cout << "Root 1: " << root1 << endl;
        cout << "Root 2: " << root2 << endl;
    }
    else if (D == 0)
    {
        double root = -b / (2 * a);
        cout << "Root: " << root << endl;
    }
    else
    {
        double rP = -b / (2 * a);
        double iP = sqrt(-D) / (2 * a);

        cout << "Root 1: " << rP << " + " << iP << "i" << endl;
        cout << "Root 2: " << rP << " - " << iP << "i" << endl;
    }

    return 0;
}
```

Problem 03: Write a program to create an array of n elements to read the marks of n students and then count how many students passed [pass marks ≥ 40] in the examination.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cout << "Enter the number of students: ";
    cin >> n;
    double m[n];
    int c = 0;
    cout << "Enter the marks: ";
    for (int i = 0; i < n; i++)
    {
        cin >> m[i];
        if (m[i] >= 40)
        {
            c++;
        }
    }
    cout << "Number of students passed: " << c << endl;
```

```
    return 0;
}
```

Problem 04: Write a program to create an array of n elements and then insert an element to the list.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter the number of elements in the list: ";
    cin >> n;
    int arr[10000];
    cout << "Enter elements: " << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int pos, ele;
    cout <<"Enter the position to insert the element: ";
    cin >> pos;
    if (pos < 1 || pos > n + 1)
    {
        cout << "Invalid position." << endl;
    }
    else
    {
        cout << "Enter the element to insert: ";
        cin >> ele;
        for (int i = n - 1; i >= pos - 1; i--)
        {
            arr[i + 1] = arr[i];
        }
        arr[pos - 1] = ele;
        n++;
        cout << "Updated list after insertion: " << endl;
        for (int i = 0; i < n; i++)
        {
            cout << arr[i] << " ";
        }
    }
    return 0;
}
```

Problem 05: Write a program to create an array of n elements and then delete an element from the list.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter the number of elements in the list: ";
    cin >> n;
    int arr[10000];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    int pos;
    cout << "Enter the position of the element to delete: ";
    cin >> pos;
    if (pos < 1 || pos > n)
    {
        cout << "Invalid position." << endl;
    }
    else
    {
        for (int i = pos - 1; i < n - 1; i++)
        {
            arr[i] = arr[i + 1];
        }
        n--;
        cout << "After deletion: " << endl;
        for (int i = 0; i < n; i++)
        {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Problem 06:

Write a program to sort n numbers using Bubble Sort algorithm.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
```

```
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    int b[n];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++)
    {
        cin >> b[i];
    }
    bool bubb = true;
    for (int i = 0; i < n - 1; i++)
    {
        bubb = false;
        for (int j = 0; j < n - i - 1; j++)
        {
            if (b[j] > b[j + 1])
            {
                swap(b[j], b[j + 1]);
                bubb = true;
            }
        }
        if (!bubb)
        {
            break;
        }
    }
    cout << "Sorted array: ";
    for (int i = 0; i < n; i++)
    {
        cout << b[i] << " ";
    }
    return 0;
}
```

Problem 07: Write a program to search an element from a list of n numbers using Linear Search algorithm.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter the number of elements in the list: ";
    cin >> n;

    int arr[10000];
    cout << "Enter elements: ";
```

```
    for (int i = 1; i <= n; i++)
    {
        cin >> arr[i];
    }

    int k;
    cout << "Enter the element to search: ";
    cin >> k;
    int index = -1;

    for (int i = 1; i <= n; i++)
    {
        if (arr[i] == k)
        {
            index = i;
            break;
        }
    }

    if (index != -1)
    {
        cout << "Element found at index: " << index << endl;
    }
    else
    {
        cout << "Element not found in the list." << endl;
    }

    return 0;
}
```

Problem 08: Write a program to search an element from a list of n numbers using Binary Search algorithm.

Answer:

```
#include <iostream>
using namespace std;
int main()
{
   int n;
   cout << "Enter the number of elements in the sorted list: ";
   cin >> n;

   int arr[10000];
   cout << "Enter elements: ";
   for (int i = 0; i < n; i++)
```

```
{
   cin >> arr[i];
}

int k;
cout << "Enter the element to search for: ";
cin >> k;

int low = 0;
int high = n - 1;
int index = -1;

while (low <= high)
{
   int mid = (high + low) / 2;
   if (arr[mid] == k)
   {
      index = mid + 1; // Increment index by 1 to make it start from 1
      break;
   }
   else if (arr[mid] < k)
   {
      low = mid + 1;
   }
   else
   {
      high = mid - 1;
   }
}

if (index != -1)
{
   cout << "Element found at index: " << index << endl;
}
else
{
   cout << "Element not found in the list." << endl;
```

```
    }
    return 0;
}
```

Problem 09: Write a program to determine whether a number n is prime or not where 1< n < 215 by using sieve method.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    const int limit = 215;
    vector<bool> isPrime(limit, true);
    for (int p = 2; p * p < limit; p++)
    {
        if (isPrime[p])
        {
            for (int i = p * p; i < limit; i += p)
            {
                isPrime[i] = false;
            }
        }
    }
    int n;
    cout <<"Enter the number to check:";
    cin >> n;
    if (n <= 1 || n >= limit)
    {
        cout << "Invalid input. Number must be between 2 and 214." << endl;
        return 1;
    }
    if (isPrime[n])
    {
        cout << n << " is a prime number." << endl;
    }
    else
    {
        cout << n << " is not a prime number." << endl;
```

```
    }
    return 0;
}
```

Problem 10: Write a program to write 100 randomly generated integer to a file called RAND.DAT. And then read the contents of the file and display them on the screen.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    ofstream output("RAND.DAT",ios::app);
    if (!output)
    {
        cerr << "Error opening the file." << endl;
        return 1;
    }
    for (int i = 0; i < 100; i++)
    {
        int random = rand() %501 +1000;
        output << random << endl;
    }
    output.close();
    ifstream input("RAND.DAT");
    if (!input)
    {
        cerr << "Error opening the file." << endl;
        return 1;
    }
    int num;
    while (input >> num)
    {
        cout << num << endl;
    }
    input.close();
    return 0;
}
```