### Design and Analysis of Algorithms Assignment - 6

###### Name: Dhanraj Kore

###### Div: TY B

###### Roll No: 60

###### Batch : B-3

**Longest Increasing Sub-sequence**

**Approach 1: Using Recursion**

**CODE :**

#include<stdio.h>

#include<stdlib.h>

int \_lis( int arr[], int n, int \*max\_ref)

{

    if (n == 1)

        return 1;

    int res, max\_ending\_here = 1;

    for (int i = 1; i < n; i++)

    {

        res = \_lis(arr, i, max\_ref);

        if (arr[i-1] < arr[n-1] && res + 1 > max\_ending\_here)

            max\_ending\_here = res + 1;

    }

    if (\*max\_ref < max\_ending\_here)

       \*max\_ref = max\_ending\_here;

    return max\_ending\_here;

}

int lis(int arr[], int n)

{

    int max = 1;

    \_lis( arr, n, &max );

    return max;

}

int main()

{

    int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };

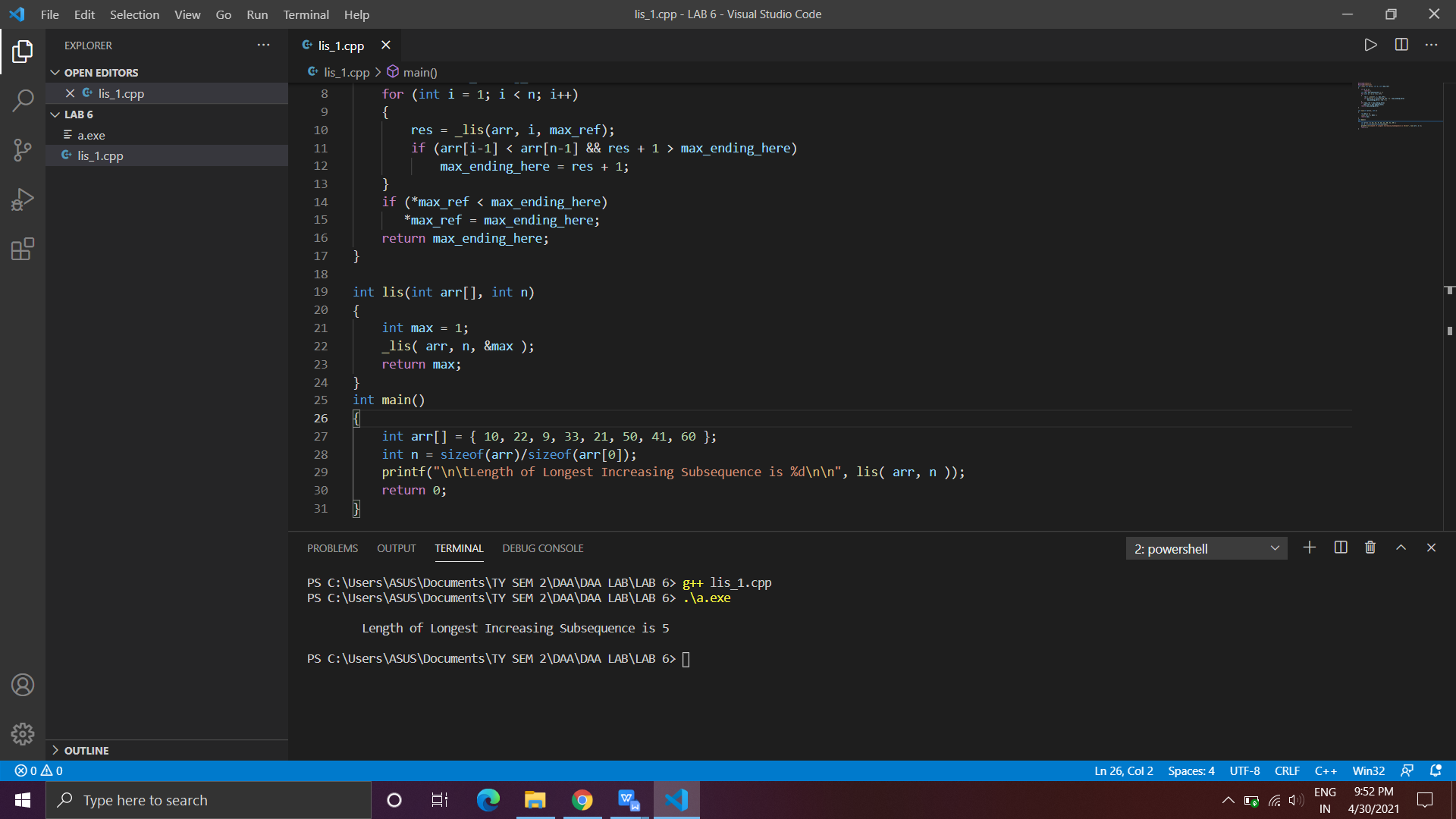
    int n = sizeof(arr)/sizeof(arr[0]);

    printf("\n\tLength of Longest Increasing Subsequence is %d\n\n", lis( arr, n ));

    return 0;

}

**O/P :**



**Time Complexity:** Exponential

**Space Complexity:** O(1)

**Approach 2: Using Dynamic Programming**

**CODE :**

#include<bits/stdc++.h>

using namespace std;

void reverse(int\* c,int cnt)

{

    int i=0,j=cnt-1;

    while(i<j)

    {

        swap(c[i],c[j]);

        i++;

        j--;

    }

}

void solve(int \*a,int n)

{

    int \*b = new int[n];

    for(int i=0;i<n;i++)

        b[i] = 1;

    for(int i=1;i<n;i++)

    {

        for(int j=i-1;j>=0;j--)

        {

            if(a[j]>=a[i])

                continue;

            int temp = b[j]+1;

            if(temp>b[i])

                b[i] = temp;

        }

    }

    //cout<<endl;

    int max = 0;

    for(int i=0;i<n;i++)

        if(b[i]>max)

            max = b[i];

    cout<<max<<endl;

    int temp = max,cnt=0,k=0;

    int \*c = new int[n];

    for(int i=n-1;i>=0;i--)

        if(temp==b[i])

        {

            cnt++;

            c[k++] = a[i];

            temp--;

        }

    reverse(c,cnt);

    cout<<"\tLongest Increasing Subsequence : ";

    for(int i=0;i<cnt;i++)

        cout<<c[i]<<" ";

    cout<<endl;

}

int main()

{

    int n;

    cout<<"Enter the total no of elements : ";

    cin>>n;

    int \*a = new int[n];

    cout<<"Enter the elements : "<<endl;

    for(int i=0;i<n;i++)

        cin>>a[i];

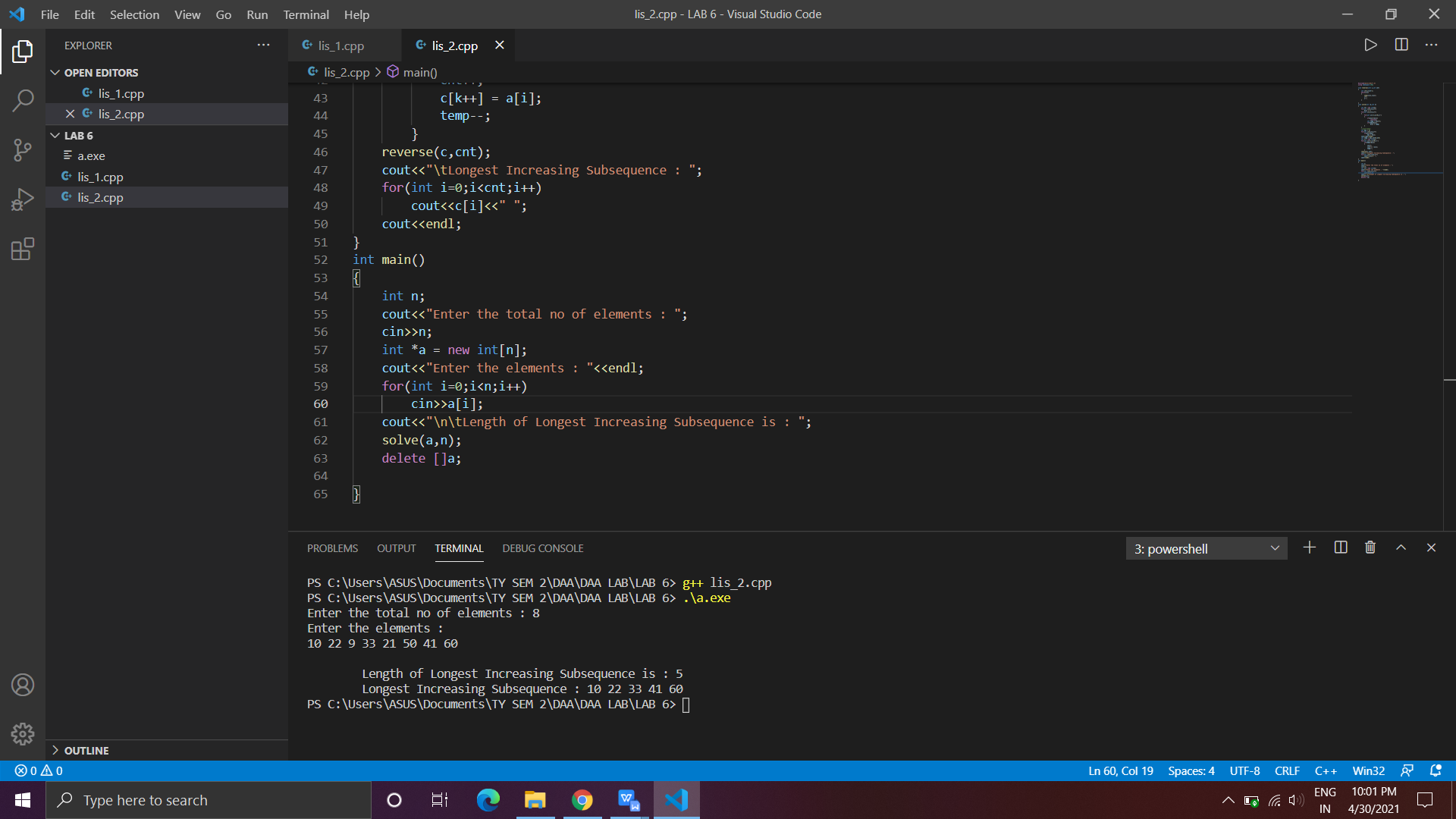
    cout<<"\n\tLength of Longest Increasing Subsequence is : ";

    solve(a,n);

    delete []a;

}

**O/P :**



**Time Complexity:** O(n^2)

**Space Complexity:** O(n)