Assignment-6

Name: Kalyane Satyam Sanjay

Roll No.: 201IT269

1. LCS Problem Statement: Given two sequences, find the length of longest sub-sequence present in both of them. A sub-sequence is a sequence that appears in the same relative order, but not necessarily contiguous. (4 (Tables)+ 4 (Output Value)+ 4 (Output))

Code:

```
#include <bits/stdc++.h>

using namespace std;

void input(string &s1, string &s2)
{
    cout << "Enter the first sequence : ";
    cin >> s1;
    cout << "Enter another sequence : ";
    cin >> s2;
}



int main()
{
    string s1;
    string s2;
    input(s1,s2);
    int n = s1.size();
    int m = s2.size();
    vector<vector<int>>mem(n,vector<int>(m,-1));
    cout << "string 1 : " << s1 << endl;
    cout << "string 2 : " << s2 << endl;


    vector<vector<int>>t(n+1,vector<int>(m+1,0));

    for(int i = 1; i <= n ; i++)
    {
        for(int j = 1; j <= m ; j++)
        {
            if(s1[i-1] == s2[j-1])
            {
```

```
                t[i][j] = 1+t[i-1][j-1];
            }
            else
            {
                t[i][j] = max(t[i-1][j],t[i][j-1]);
            }
        }
    }
    cout << "Maximum length of common sequence : " << t[n][m] << endl;

    for(int i = 0; i <= n ; i++)
    {
        for(int j = 0; j <= m ; j++)
        {
            cout << t[i][j] << "\t";
        }
        cout << endl;
    }

    int x = n;
    int y = m;
    string ans;
    while(n > 0 && m > 0)
    {
        if(t[n][m] == t[n-1][m])
        {
            n--;
        }
        else if(t[n][m] == t[n][m-1])
        {
            m--;
        }
        else
        {
            ans.push_back(s1[n-1]);
            n--;
            m--;
        }
    }
    reverse(ans.begin(), ans.end());
    cout << "Longest common subsequence is : " << ans << endl;
    return 0;
}
```

Output:

```
PS F:\IT-300\Assignment-6> cd "f:\IT-300\Assignment-6\" ; if ($?) { g++ LCS.cpp -o LCS } ; if ($?) { .\LCS }
Enter the first sequence : ABACDEABA
Enter another sequence : BCADCEEABE
string 1 : ABACDEABA
string 2 : BCADCEEABE
Maximum length of common sequence : 6
0       0       0       0       0       0       0       0       0       0       0
0       0       0       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       2       2
0       1       1       2       2       2       2       2       2       2       2
0       1       2       2       2       3       3       3       3       3       3
0       1       2       2       3       3       3       3       3       3       3
0       1       2       2       3       3       4       4       4       4       4
0       1       2       3       3       3       4       4       5       5       5
0       1       2       3       3       3       4       4       5       6       6
0       1       2       3       3       3       4       4       5       6       6
Longest common subsequence is : BACEAB
PS F:\IT-300\Assignment-6>
```


2. The Longest Increasing Sub-sequence (LIS) problem is to find the length of the longest sub-sequence of a given sequence such that all elements of the sub-sequence are sorted in increasing order. For example, the length of LIS for {10, 22, 9, 33, 21, 50, 41, 60, 80} is 6 and LIS is {10, 22, 33, 50, 60, 80}. Print the length and the LIS sequence. (4(Output value) + 4(Output))

Code:

```cpp
#include <bits/stdc++.h>

using namespace std;


void input(vector<int>&arr)
{
    int n;
    cout << "Enter the length of sequence : ";
    cin >> n;
    while( n <= 0)
    {
        cout << "Enter valid positive value : ";
        cin >> n;
    }
    for(int i = 0; i < n ; i++)
    {
        int element;
        cout << "Enter element " << i+1 << " : ";
        cin >> element;
```

```
        arr.push_back(element);
    }
}
int main()
{
    vector<int>arr;
    input(arr);
    int n = arr.size();
    vector<int> increasing(n,0);
    vector<int> parent(n,-1);
    int max_value = INT_MIN;
    int max_index = -1;
    // for(int i = 0; i < n ; i++)
    // {
    //     parent.emplace_back(i);
    // }

    for(int i = 0; i < n ; i++)
    {
        increasing[i] = 1;
        for(int j = 0; j < i ; j++)
        {
            if(arr[j] < arr[i])
            {
                if(increasing[j]+1 > increasing[i])
                {
                    parent[i] = j;
                    increasing[i] = 1+increasing[j];
                }
            }

        }
        if(increasing[i] > max_value)
        {
            max_value = increasing[i];
            max_index = i;
        }
    }
    cout << endl;
    cout << "Length of LIS is : " << max_value << endl;
    // cout << "LIS for n length : " << endl;
    // for(int i = 0; i < n ; i++)
    // {
    //     cout << increasing[i] << '\t';
    // }
    // cout << endl;

    // cout << "Parent(previous element) index for each element" << endl;
```

```
    // for(int i = 0; i < n ; i++)
    // {
    //     cout << parent[i] << '\t';
    // }
    // cout << endl;
    vector<int> ans;
    ans.emplace_back(arr[max_index]);
    while(parent[max_index] != -1)
    {
        ans.emplace_back(arr[parent[max_index]]);
        max_index = parent[max_index];
    }
    int ans_len = ans.size();
    reverse(ans.begin(),ans.end());
    cout << "LIS : "<< endl;
    for(int i = 0; i < ans_len ; i++)
    {
        cout << ans[i] << "\t";

    }
    cout << endl;
}
```

Output:

```
PS F:\IT-300\Assignment-6> cd "f:\IT-300\Assignment-6\" ; if ($?) { g++ LIS.cpp -o LIS } ; if ($?) { .\LIS }
Enter the length of sequence : 9
Enter element 1 : 10
Enter element 2 : 22
Enter element 3 : 9
Enter element 4 : 33
Enter element 5 : 21
Enter element 6 : 50
Enter element 7 : 41
Enter element 8 : 60
Enter element 9 : 80

Length of LIS is : 6
LIS :
10      22      33      50      60      80
PS F:\IT-300\Assignment-6>
```

Thank You