Assignment No. 8

```
************************************************************************************************************

/*Problem Statement:-

        Given sequence k = k1 <k2 < ... < kn of n sorted keys, with a search probability pi for each
        key ki . Build the Binary search tree that has the least search cost given the access
probability
        for each key.
************************************************************************************************************/



#include<stdio.h>
#include<iostream>
#define max 20
#define inf 999

using namespace std;

int i,j,k,n,min,r[max][max];
float p[max],q[max],w[max][max],c[max][max];

void OBST();
void print(int,int);
void print_tab();
int main()
{

        cout<<"\nEnter no. of nodes :";
        cin>>n;
        cout<<"\nEnter successful search probability :";
        for(i=1;i<=n;i++)
        {
                cin>>p[i];
        }
        cout<<"\nEnter unsuccessful search probability :";
        for(i=0;i<=n;i++)
        {
                cin>>q[i];
        }
        OBST();

        print_tab();

        return 0;
}
void OBST()
{
        int x;
        for(i=0;i<=n;i++) //Calculate values for first row
        {
                //for row j-i=0
                r[i][i] = 0;
        c[i][i] = 0;
        w[i][i] = q[i];
    }
    for(i=0;i<n;i++) //calculate values for second row
    {
        //for row j-i=1
        j=i+1;
        w[i][j]=w[i][i]+p[j]+q[j]; //w[i][i];
        c[i][j]=w[i][j];
        r[i][j]=j;
    }

//calculate values for third row onwards
    for(j=2;j<=n;j++)
    {
        for(i=0;i<=n-j;i++)
        {
            x=i+j;
            w[i][x]=w[i][x-1]+p[x]+q[x];
            c[i][x]=inf;
            for(k=i+1;k<=x;k++)
            {
                if(c[i][x]>c[i][k-1]+c[k][x])
                {
                    c[i][x]=c[i][k-1]+c[k][x];
                    r[i][x]=k;
                }
            }
            c[i][x]=c[i][x]+w[i][x];
        }
    }
}
void print_tab()
{
    cout<<endl<<"----OBST TABLE----"<<endl;
    k=0;
    while(k<=n)    //print rows from 0 to n
    {
        for(i=0,j=i+k;i<n,j<=n;i++,j++)                     //print weight
            cout<<"w"<<i<<j<<"="<<w[i][j]<<"\t";
        cout<<endl;
        for(i=0,j=i+k;i<n,j<=n;i++,j++)                     //print cost
            cout<<"c"<<i<<j<<"="<<c[i][j]<<"\t";
        cout<<endl;
        for(i=0,j=i+k;i<n,j<=n;i++,j++)                     //print root
            cout<<"r"<<i<<j<<"="<<r[i][j]<<"\t";
        cout<<endl<<endl;
        k++;
    }
}
```

```
/****************************************** OUTPUT *******************************************


Enter no. of nodes :4

Enter successful probability :3 3 1 1

Enter unsuccessful probability :2 3 1 1 1

----OBST TABLE----

w00=2  w11=3  w22=1  w33=1  w44=1
c00=0  c11=0  c22=0  c33=0  c44=0
r00=0  r11=0  r22=0  r33=0  r44=0

w01=8  w12=7  w23=3  w34=3
c01=8  c12=7  c23=3  c34=3
r01=1  r12=2  r23=3  r34=4

w02=12        w13=9  w24=5
c02=19c13=12c24=8
r02=1  r13=2  r24=3

w03=14        w14=11
c03=25c14=19
r03=2  r14=2

w04=16
c04=32
r04=2


*/
```