/\* Assignment No.

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

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 probability :";

for(i=1;i<=n;i++)

{

cin>>p[i];

}

cout<<"\nEnter unsuccessful probability :";

for(i=0;i<=n;i++)

{

cin>>q[i];

}

OBST();

print\_tab();

return 0;

}

void OBST()

{

for(i=0;i<n;i++)

{

r[i][i] = 0;

c[i][i] = 0;

w[i][i] = q[i];

w[i][i+1]=q[i]+p[i+1]+q[i+1]; //w[i][i];

c[i][i+1]=w[i][i+1];

r[i][i+1]=i+1;

}

c[n][n]=0.0;

r[n][n]=0.0;

w[n][n]=q[n];

for(j=2;j<=n;j++)

{

for(i=0;i<=n-j;i++)

{

w[i][i+j]=w[i][i+j-1]+p[i+j]+q[i+j];

c[i][i+j]=999;

for(k=i+1;k<=i+j;k++)

{

if(c[i][i+j]>c[i][k-1]+c[k][i+j])

{

c[i][i+j]=c[i][k-1]+c[k][i+j];

r[i][i+j]=k;

}

}

c[i][i+j]=c[i][i+j]+w[i][i+j];

}

}

}

void print\_tab()

{

cout<<endl<<"----OBST TABLE----"<<endl;

k=0;

while(k<=n)

{

for(i=0,j=i+k;i<n,j<=n;i++,j++)

cout<<"w"<<i<<j<<"="<<w[i][j]<<"\t";

cout<<endl;

for(i=0,j=i+k;i<n,j<=n;i++,j++)

cout<<"c"<<i<<j<<"="<<c[i][j]<<"\t";

cout<<endl;

for(i=0,j=i+k;i<n,j<=n;i++,j++)

cout<<"r"<<i<<j<<"="<<r[i][j]<<"\t";

cout<<endl<<endl;

k++;

}

}