/\*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 <iostream>

#define MAX 999999;

using namespace std;

class obst

{

public:

int mincost(int[],int[],int);

int obstcost(int[],int,int);

int sum(int prob[],int i,int j)

{

int s=0;

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

{

s+=prob[k];

}

return s;

}

};

int obst::obstcost(int prob[],int i,int j)

{

if(j<i)

{

return 0;

}

else if(i==j)

{

return prob[i];

}

else

{

int psum = sum(prob,i,j);

int min = MAX;

for(int r=i;r<=j;r++)

{

int cost = obstcost(prob,i,r-1) + obstcost(prob,r+1,j);

//cout<<"\ncost = "<<cost;

if(cost<min)

{

min = cost;

}

}

//cout<<"\ncost = "<<min + psum;

return (min + psum);

}

}

int obst::mincost(int keys[],int prob[],int n)

{

return obstcost(prob,0,n-1);

}

int main() {

obst o;

int n,keys[30],prob[30];

cout<<"\nEnter the No of Keys: ";

cin>>n;

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

{

cout<<"\n\nEnter the Key: ";

cin>>keys[i];

cout<<"\nEnter the Probability: ";

cin>>prob[i];

}

int c = o.mincost(keys,prob,n);

cout<<"\n\nLeast Search Cost is: "<<c;

return 0;

}