#include <iostream>

using namespace std;

#define INT\_MAX 100

//function to get sum of array elements

int sum(int freq[] , int i , int j ){

int sum = 0 ;

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

sum += freq[k];

}

return sum;

}

// recursive function to calculate cost of optimal binary search tree

int optCost ( int freq[] , int i , int j){

if ( j < i )

return 0;

if( j == i)

return freq[i];

//get sum of frequencies

int fsum = sum(freq,i , j);

int min = INT\_MAX;

//one by one consider all elements as root and recursively find cost of the bst

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

int cost = optCost(freq, i , r-1) + optCost(freq , r+1 , j);

if (cost < min)

min = cost;

}

// return minimum value

return min + fsum;

}

int optimalSearchTree (int key[], int freq[] , int n ){

return optCost(freq,0,n-1);

}

int main(){

int num;

cout << "Enter the number of key :: ";

cin >> num ;

int\* keys = new int[num];

int\* freq = new int[num];

cout << "Enter the keys in ascending order :: "<< endl;

for (int i=0 ; i<num ; i++){

int k,f;

cout << "key :: ";

cin>> k;

keys[i] = k;

cout << "Frequencies :: ";

cin>>f;

freq[i] = f;

}

cout << " Cost of optimal BST is " << optimalSearchTree(keys,freq,num);

delete[] keys;

delete[] freq;

return 0;

}