Optimal binary search tree

**#include <stdio.h>**

**#include <limits.h>**

**Int sum(int freq[], int i, int j);**

**Int optCost(int freq[], int i, int j)**

**{**

**If (j < i)**

**Return 0;**

**If (j == i)**

**Return freq[i];**

**Int fsum = sum(freq, i, j);**

**Int min = INT\_MAX;**

**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 min + fsum;**

**}**

**Int optimalSearchTree(int keys[], int freq[], int n)**

**{**

**Return optCost(freq, 0, n-1);**

**}**

**Int sum(int freq[], int i, int j)**

**{**

**Int s = 0;**

**For (int k = i; k <=j; k++)**

**S += freq[k];**

**Return s;**

**}**

**Int main()**

**{**

**Int keys[] = {10, 12, 20};**

**Int freq[] = {34, 8, 50};**

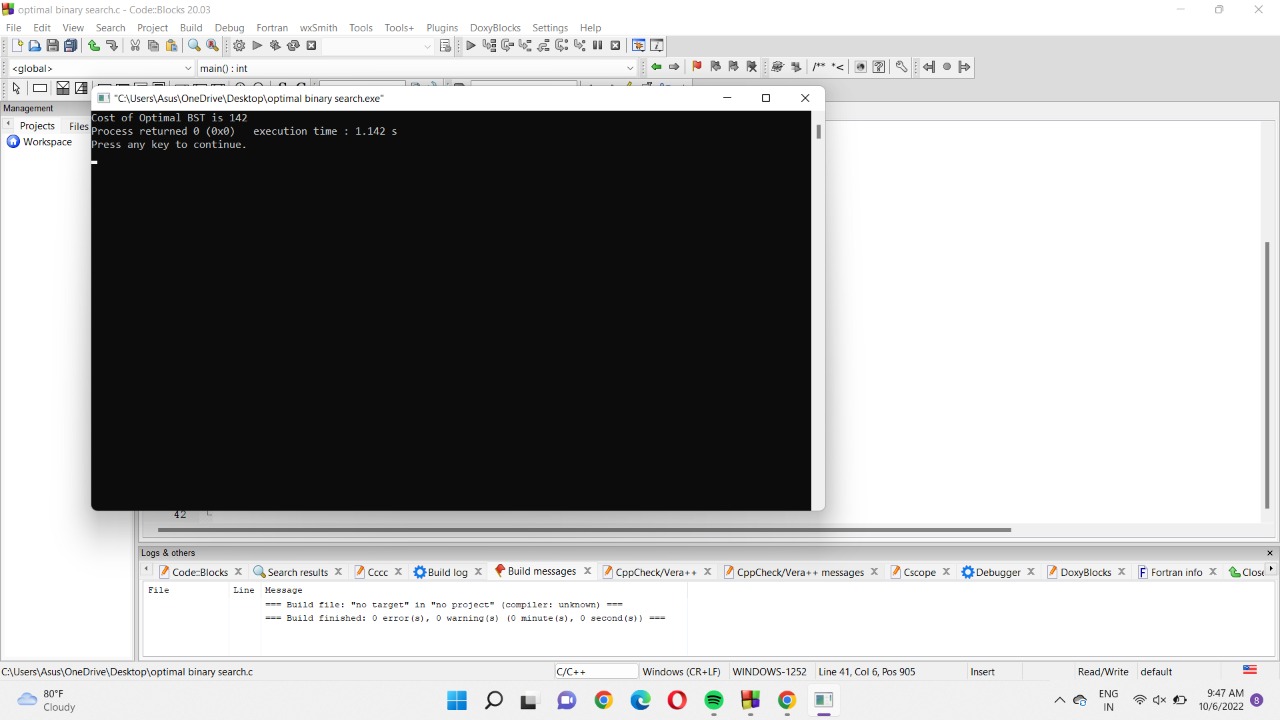
**Int n = sizeof(keys)/sizeof(keys[0]);**

**Printf(“Cost of Optimal BST is %d “,**

**optimalSearchTree(keys, freq, n));**

**return 0;**

**}**

****