**NAME** : Tejesh Santosh Yewale

**ROLL NO. :** A-61

**PRACTICAL NO. D1**

**CODE:**

#include<iostream>

using namespace std;

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 number\_of\_keys;

cout << "\nEnter number of keys : ";

cin >> number\_of\_keys;

int keys[number\_of\_keys];

int freq[number\_of\_keys];

cout << "\n";

for (int i = 0; i < number\_of\_keys; ++i)

{

cout << "Enter key and frequency : ";

cin >> keys[i] >> freq[i];

}

int n = sizeof(keys) / sizeof(keys[0]);

cout << "\nCost of Optimal BST : " << optimalSearchTree(keys, freq, n) << "\n";

return 0;

}

**OUTPUT:**

