

Assignment7-Primz.cpp

```
1 // Prim's Algorithm
2
3 #include <cstring>
4 #include <iostream>
5 using namespace std;
6
7 #define INF 9999999
8
9 // number of vertices in grapj
10 #define V 5
11
12 // create a 2d array of size 5x5
13 //for adjacency matrix to represent graph
14
15 int G[V][V] = {
16     {0, 9, 75, 0, 0},
17     {9, 0, 95, 19, 42},
18     {75, 95, 0, 51, 66},
19     {0, 19, 51, 0, 31},
20     {0, 42, 66, 31, 0}};
21
22 int main() {
23     string buildings[] = {"First Year","Main Building","Auditorium","Science
College","Ground"};
24     int no_edge; // number of edge
25
26     // create a array to track selected vertex
27     // selected will become true otherwise false
28     int selected[V];
29
30     // set selected false initially
31     memset(selected, false, sizeof(selected));
32
33     // set number of edge to 0
34     no_edge = 0;
35
36     // the number of egde in minimum spanning tree will be
37     // always less than (V -1), where V is number of vertices in
38     //graph
39
40     // choose 0th vertex and make it true
41     selected[0] = true;
42
43     int x; // row number
44     int y; // col number
45
46     // print for edge and weight
47     cout << "Building 1 - Building 2"
48         << " : "
49         << "Weight";
```

```
50     cout << endl;
51     while (no_edge < V - 1) {
52         //For every vertex in the set S, find the all adjacent vertices
53         // , calculate the distance from the vertex selected at step 1.
54         // if the vertex is already in the set S, discard it otherwise
55         //choose another vertex nearest to selected vertex at step 1.
56
57         int min = INF;
58         x = 0;
59         y = 0;
60
61         for (int i = 0; i < V; i++) {
62             if (selected[i]) {
63                 for (int j = 0; j < V; j++) {
64                     if (!selected[j] && G[i][j]) { // not in selected and there is an edge
65                         if (min > G[i][j]) {
66                             min = G[i][j];
67                             x = i;
68                             y = j;
69                         }
90                     }
91                 }
92             }
93         }
94         cout << buildings[x]<< " - " << buildings[y]<< " : " << G[x][y];
95         cout << endl;
96         selected[y] = true;
97         no_edge++;
98     }
99
100     return 0;
101 }
102
103 /*
104 Output
105 Building 1 - Building 2 : Weight
106 First Year - Main Building : 9
107 Main Building - Science College : 19
108 Science College - Ground : 31
109 Science College - Auditorium : 51
110
111 */
112
```