Experiment 24: Prim’s Algorithm

Aim:
To write a C program to construct a Minimum Spanning Tree using Prim’s Algorithm.

Algorithm:

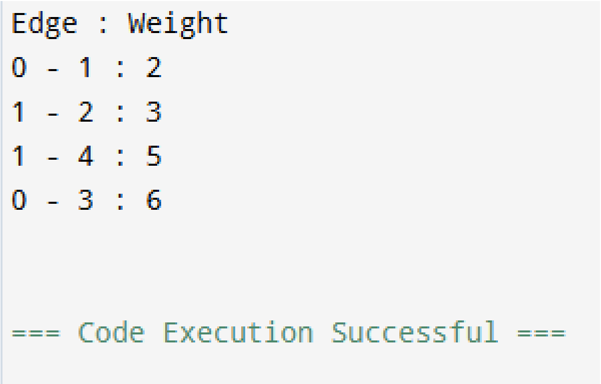
1. Start the program.
2. Select an arbitrary vertex as the starting vertex.
3. Repeat until all vertices are included:
    - Pick the minimum weight edge that connects a visited vertex to an unvisited vertex.
    - Add that edge to MST.
4. Print MST edges and cost.
5. Stop.

Code:

```
#include <stdio.h>

#define INF 9999

#define V 5

void prim(int graph[V][V]) {

   int selected[V] = {0}, edges = 0;

   selected[0] = 1;

   printf("Edge : Weight\n");

   while (edges < V - 1) {

      int min = INF, x = 0, y = 0;

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

         if (selected[i]) {

            for (int j = 0; j < V; j++) {

               if (!selected[j] && graph[i][j]) {

                  if (graph[i][j] < min) {

                     min = graph[i][j];

                     x = i; y = j;

                  }

               }
```

```
                }
            }
        }
        printf("%d - %d : %d\n", x, y, graph[x][y]);
        selected[y] = 1;
        edges++;
    }
}
int main() {
    int graph[V][V] = {
        {0,2,0,6,0},
        {2,0,3,8,5},
        {0,3,0,0,7},
        {6,8,0,0,9},
        {0,5,7,9,0}
    };
    prim(graph);
    return 0;
}
```

Sample Output:



```
Edge : Weight
0 - 1 : 2
1 - 2 : 3
1 - 4 : 5
0 - 3 : 6


=== Code Execution Successful ===
```

Result:
The program successfully finds the MST using Prim’s Algorithm.