PRIM'S MINIMUM SPANNING TREE

Exp. No.:
AIM:

ALGORITHM:

PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>

#define INF 9999999

int **createGraph(int n)
{
   int **G, i, j;
   G = (int **)malloc(n * sizeof(int));
   for (i = 0; i < n; i++)
   {
      G[i] = (int *)malloc(n * sizeof(int));
   }
   for (i = 0; i < n; i++)
   {
      printf("Enter Weight of %d Vertex (0 if Not Connected):", i);
      for (j = 0; j < n; j++)
      {
         scanf("%d", &G[i][j]);
      }
   }
   return G;
}

void disp(int n, int **G)
{
   int i, j;
   for (i = 0; i < n; i++)
   {
      for (j = 0; j < n; j++)
      {
         printf("%d ", G[i][j]);
      }
      printf("\n");
```

```
    }
}

void Prims(int **G, int V, int r)
{
    int i, j, x, y;
    int *selected;
    int no_edge;
    unsigned long int min;
    selected = (int *)calloc(V, sizeof(int));
    memset(selected, 0, sizeof(selected));
    no_edge = 0;
    selected[r] = 1;

    printf("Edge : Weight\n");
    while (no_edge < V - 1)
    {
        min = INF;
        x = 0;
        y = 0;
        for (i = 0; i < V; i++)
        {
            if (selected[i])
            {
                for (j = 0; j < V; j++)
                {
                    if (!selected[j] && G[i][j])
                    {
                        if (min > G[i][j])
                        {
                            min = G[i][j];
                            x = i;
                            y = j;
                        }
                    }
                }
            }
        }
        printf("%d - %d : %d\n", x, y, G[x][y]);
```

```
        selected[y] = 1;
        no_edge++;
    }
}

int main()
{
    int V, **G, root;
    printf("Enter Number of Vertices:");
    scanf("%d", &V);
    G = createGraph(V);
    printf("Enter Root Vertex:");
    scanf("%d", &root);
    Prims(G, V, root);
    getch();
    clrscr();
    return 0;
}
```

OUTPUT:

```
Enter Number of Vertices:7
Enter Weight of 0 Vertex (0 if Not Connected):0 28 0 0 0 10 0
Enter Weight of 1 Vertex (0 if Not Connected):28 0 16 0 0 0 14
Enter Weight of 2 Vertex (0 if Not Connected):0 16 0 12 0 0 0
Enter Weight of 3 Vertex (0 if Not Connected):0 0 12 0 22 0 18
Enter Weight of 4 Vertex (0 if Not Connected):0 0 0 22 0 25 24
Enter Weight of 5 Vertex (0 if Not Connected):10 0 0 0 25 0 0
Enter Weight of 6 Vertex (0 if Not Connected):0 14 0 18 24 0 0
Enter Root Vertex:2
Edge : Weight
2 - 3 : 12
2 - 1 : 16
1 - 6 : 14
3 - 4 : 22
4 - 5 : 25
5 - 0 : 10
_
```

RESULT: