NAME:- MODI PARTANYA

ADA
LAB TEST.

USN:- 1BM19CS088

Q 5

```
# include<stdio.h>

int u, v, n, i, j, k=1;
int visited[10], weight[10][10], min, minweight=0;

int main(){
  printf("\n Enter the number of nodes: ");
  scanf("%d", &n);
  printf("\n Enter the adjacency matrix: \n");
   printf("\n Enter the 9999 for infinity \n");
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++){
            scanf("%d", &weight[i][j]);
            if(weight[i][j]==0)
              { weight[i][j]=999;
              }
        }
    }
    visited[1]=1;
    printf("\n");
    while(k<n){
        for(i=1, min=9999; i<=n; i++)
        {
            for(j=1; j<=n; j++)
              { if(weight[i][j]<min)
                  { if(visited[i]!=0)
                    { min=weight[i][j];
                      u=i;
                      v=j;
                    }
```

```
if (visited[u]==0 || visited[v]==0)
    {   printf("\n (%d %d) weight: %d", u, v, min);
        k++;
        min weight += min;
        visited[v] = 1;
    }
    weight[u][v] = weight[v][u] = 9999;
}
printf("\n Total minimum weight = %d", minweight);
return 0;
}
```

Modified :-

```
#include<stdio.h>
int specified_node = 2;
int minikey(int key[], int mstSet[], int n)
{   int min = 100, min_index;
    int v;
    for (v=0; v<n; v++)
    {   if (mstSet[v]==0 && key[v] < min)
        {   min = key[v], min_index = v;
        } return min_index;
    }
}
int
```

```
int printMST(int parent[10], int graph[10][10], int n)
{
    int i;
    printf("Edge \t Weight\n");
    for(i=1;i<n;i++)
    {
        if(parent[i]>10);
        else
        printf("%d - %d \t %d \n", parent[i], i, graph[i][parent[i]]);
    }

void primMST(int graph[10][10], int n)
{
    int parent[n];
    int key[n];
    int mstSet[n];
    int i, count, v, u;

    for(i=0;i<n;i++)
        key[i]=100, mstSet[i]=0;

    key[0]=0;
    parent[0]=-1;
    mstSet[specified_node]=1;
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
            {
                printf("%d\t", graph[i][j]);
            }
            printf("\n");}
```

```
for (count = 0; count < n-1; count++)
{
    u = miniKey(key, mstSet, n);
    mstSet[u] = 1;
    for (v=0; v<n; v++)
    {
        if (graph[u][v] != 0 && graph[u][v] != 999 &&
            mstSet[v] == 0 && graph[u][v] < key[v])
        {
            parent[v] = u;
            key[v] = graph[u][v];
        }
    }
    printMST(parent, graph, n);
}

int main()
{
    int graph[10][10];
    int i, j, n;
    printf("Enter numbers of nodes \n");
    scanf("%d", &n);
    printf("Enter adjacency matrix \n");
    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
            scanf("%d", &graph[i][j]);
    }
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
        {
            if (i == specified_node || j == specified_node)
                if (i == j); else
                    graph[i][j] = 999;
        }
    }
    printMST(graph, n);
    return 0;
}
```