

## prims.c

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 100
#define TEMP 0
#define PERM 1
#define infinity 9999
#define NIL -1
struct edge
{
    int u;
    int v;
};
int n;
int adj[MAX][MAX];
int predecessor[MAX];
int status[MAX];
int length[MAX];
void create_graph();
void maketree(int r, struct edge tree[MAX]);
int min_temp();
void main()
{
    int wt_tree=0;
    int i;
    int root;
    struct edge tree[MAX];
    create_graph();
    printf("Enter the root vertex: ");
    scanf("%d",&root);
    maketree(root,tree);
    printf("Edges to be included in the minimum spanning tree are:\n");
    for(i=1;i<n;i++)
    {
        printf("%c->%c ",(char)(tree[i].u+97),(char)(tree[i].v+97));
        wt_tree=wt_tree+adj[tree[i].u][tree[i].v];
    }
    printf("\nWeight of the minimum spanning tree is: %d\n",wt_tree);
}

void maketree(int r,struct edge tree[MAX])
{
    int current,i;
    int count=0;
    for(i=0;i<n;i++)
    {
        predecessor[i]=NIL;
        length[i]=infinity;
        status[i]=TEMP;
    }
    length[r]=0;
    while(1)
    {
        current=min_temp();
        if(current==NIL)
        {
            if(count==n-1)
```

```

        return;
    else
    {
        printf("Spanning tree not possible");
        exit(1);
    }
}
status[current]=PERM;
if(current!=r)
{
    count++;
    tree[count].u=predecessor[current];
    tree[count].v=current;
}
for(i=0;i<n;i++)
{
    if(adj[current][i]>0 && status[i]==TEMP)
    {
        if(adj[current][i]<length[i])
        {
            length[i]=adj[current][i];
            predecessor[i]=current;
        }
    }
}
}
}

int min_temp()
{
    int i;
    int min=infinity;
    int k=-1;
    for(i=0;i<n;i++)
    {
        if(status[i]==TEMP && length[i]<min)
        {
            min=length[i];
            k=i;
        }
    }
    return k;
}

void create_graph()
{
    int i,max_edges,origin,destin,wt;
    printf("Enter number of vertices: ");
    scanf("%d",&n);
    max_edges=n*(n-1)/2;
    for(i=1;i<=max_edges;i++)
    {
        printf("Enter edge %d(0 0 to quit): ",i);
        scanf("%d %d",&origin,&destin);
        if((origin==0) && (destin==0))
            break;
        printf("Enter weight for this edge: ");
        scanf("%d",&wt);
        if(origin>=n || destin>=n || origin<0 || destin<0)
        {

```

```
        printf("Invalid edge!\n");
        i--;
    }
    else
        adj[origin][destin]=wt;
}
}
```