

## **Lab Assignment 7**

NAME : Rahul Thapar

ID : 1410110321

Implement Prim's Algorithm.

CODE :

```
/*
    @author : Rahul Thapar
    ID : 1410110321
    Date : 15 Feb, 201

    1. Prim's Algorithm
    2. Kruskal's Algorithm

*/
#include <stdio.h>
#include <stdlib.h>

void main(){

    printf("\n\t Implement :\n");
    printf("\t ++++++\n");
    printf("\n\t 1. Prim's Algorithm\n");
    printf("\t 2. Kruskal's Algorithm\n");

    int choice;
    printf("\n\t CHOICE : ");
    scanf("%d",&choice);

    switch(choice){
        case 1: PrimsFileChose();
                break;

        case 2 : printf("\n\t PENDING\n" );
                break;

        default : printf("Enter the correct option\n");
                break;
    }
}
```

```

PrimsFileChose(){
    printf("\t Take input from : \n");
    printf("\t 1. Graph 1 : Node = 4\n");
    printf("\t 2. Graph 2 : Node = 5 \n");
    /*printf("\t 3. Graph 3 : Node = 6\n");*/
    int prim_file;
    printf("\n \t FILE NUMBER : ");
    scanf("%d",&prim_file);

    printf("\t File Loaded : %d\n",prim_file);

    Prim_Implement(prim_file);
}
Prim_Implement(int file_number){

    FILE *fi
    int d[50][50];
    int a[10];
    int i,j,ch,data,first,zz=0,xx,k,yy,y1=0,y2=0;
    int title=0,titlecc=0;
    int total_distance =0, p=0;
    char c;
    for(i=0;i<10;i++){
        a[i]=100;
    }
    switch(file_number){
        case 1: /*printf("Case 1\n");*/
            file1=fopen("graph1.txt","r");
            Implement_Algorithm(file1);
            break;

            case 2 : //printf("Case 2 \n");
            file1=fopen("graph2.txt","r");
            Implement_Algorithm(file1);
            break;

            default : printf("File doesn't exists\n");
            break;

    }
}
Implement_Algorithm(FILE *file1){

```

```

int d[50][50];
int a[10];
int i,j,ch,data,first,zz=0,xx,k,yy,y1=0,y2=0;
int title=0,titlecc=0;
int total_distance =0, p=0;
char c;

while((c=getc(file1))!=EOF){

    if(c=='\n'){
        if(titlecc==1){
            first++;
        }
        data=0;
        ch=0;
        titlecc=1;
        continue;
    }
    if(c=='\t'){
        data++;
        ch=0;
        continue;
    }

    if(ch==0){
        if(titlecc==1){
            d[first][data]=c '0';
        }
        else{
            title=c '0';
        }
    }

    ch=1;
    {
        d[first][data]=(c '0')+d[first][data]*10;
    }
}
fclose(file1);
printf("\t Number of Nodes = %d\n\n",title);
for(i=0;i<title;i++){
for(j=0;j<title;j++){
printf("\t d[%d][%d]=%d\t",i+1,j+1,d[i][j]);
}
printf("\n");

```

```

}
printf("\t=====\\n");
//main

for(k=1;k<9;k++){
    for(i=0;i<title;i++){
        for(j=0;j<title;j++){

            if(d[i][j]==k){

                a[zz]=i+1;
                a[zz+1]=j+1;
                y1=0;
                y2=0;

                /*for(yy=0;yy<8;yy++){
                    if((i+1)==a[yy]){y1=y1+1;}
                    if((j+1)==a[yy]){y2=y2+1;}
                    printf("a[%d]=%d\\n",yy,a[yy]);
                }
                printf("\\n");*/
                if(y1==2 && y2==2 && a[7]==4){
                    printf("be a ...loop \\n\\n");
                }
                else
                {
                    //printf("aa=%d\\n\\n",aa);

                    zz=zz+2;
                    total_distance=total_distance+k;

                    printf("\\n\\td[%d][%d]=%d\\n",i+1,j+1,d[i][j]);
                    d[j][i]=100;
                    d[i][j]=100;
                    printf("\\ttotal_distance =%d\\n\\n",total_distance);
                    p=p+1;
                    i=title;
                    j=title;
                    if(p==(title 1)){
                        k=15;
                        printf("\\n.....\\n");
                    }
                }
            }
        }
    }
}

```

```

    }
}

}

}

```

## Screenshot :

```

rahthap@rahthap-Inspiron-3521: ~/Desktop/Lab7
rahthap@rahthap-Inspiron-3521:~/Desktop/Lab7$ ./lab7

Implement :
+++++++

1. Prim's Algorithm
2. Kruskal's Algorithm

CHOICE : 1
Take input from :
1. Graph 1 : Node = 4
2. Graph 2 : Node = 5

FILE NUMBER : 1
File Loaded : 1
Number of Nodes = 4

d[1][1]=0      d[1][2]=2      d[1][3]=3      d[1][4]=4
d[2][1]=10     d[2][2]=0      d[2][3]=6      d[2][4]=5
d[3][1]=10     d[3][2]=10     d[3][3]=0      d[3][4]=1
d[4][1]=10     d[4][2]=10     d[4][3]=10     d[4][4]=0
=====

d[3][4]=1
total_distance =1

d[1][2]=2
total_distance =3

d[1][3]=3
total_distance =6

.....
rahthap@rahthap-Inspiron-3521:~/Desktop/Lab7$ ./lab7

Implement :
+++++++

1. Prim's Algorithm
2. Kruskal's Algorithm

CHOICE : 2

PENDING
rahthap@rahthap-Inspiron-3521:~/Desktop/Lab7$

```

```
rahthap@rahthap-inspiron-3521: ~/Desktop/Lab7
Implement :
+++++
1. Prim's Algorithm
2. Kruskal's Algorithm

CHOICE : 1
Take input from :
1. Graph 1 : Node = 4
2. Graph 2 : Node = 5

FILE NUMBER : 2
File Loaded : 2
Number of Nodes = 5

d[1][1]=0      d[1][2]=1      d[1][3]=0      d[1][4]=4      d[1][5]=5
d[2][1]=10     d[2][2]=0      d[2][3]=6      d[2][4]=3      d[2][5]=0
d[3][1]=0      d[3][2]=60     d[3][3]=0      d[3][4]=7      d[3][5]=2
d[4][1]=40     d[4][2]=30     d[4][3]=70     d[4][4]=0      d[4][5]=0
d[5][1]=50     d[5][2]=0      d[5][3]=20     d[5][4]=0      d[5][5]=0
=====

d[1][2]=1
total_distance =1

d[3][5]=2
total_distance =3

d[2][4]=3
total_distance =6

d[1][4]=4
total_distance =10

.....
rahthap@rahthap-Inspiron-3521:~/Desktop/Lab7$
```