Name:- Aditya Patil
Div- SE A Roll No-42

Experiment No. 6

Code-

```
#include<stdio.h>
int DistMat[10][10],visitedCity[10],n,cost=0, InitialVertex = 0;
int main()
{
        getDistMat();
        printf("\n\nOptimal Path is:--> :\n");
        mincost(InitialVertex);
        printf("\n\nMinimum cost is %d\n",cost);
        return 0;
}
void getDistMat()
{
        int i,j;
        printf("Enter Number of Cities :-->");
        scanf("%d",&n);
        printf("\nEnter the Distance Matrix--> \n");
        for(i=0;i<n;i++)
        {
        printf("\n Enter Distance of Row:%d\n",i+1);
        for (j=0;j<n;j++)
        {
        scanf("%d",&DistMat[i][j]);
        }
        visitedCity[i]=0;
        }

        printf("\n\nCost Matrix :-->");
        for(i=0;i<n;i++)
        {
        printf("\n");
        for(j=0;j<n;j++)
        {
                printf("\t%d",DistMat[i][j]);
        }
        }
        }
```

```
void mincost(int city)
{
            int i,ncity;
            visitedCity[city]=1;
            printf("%d --> ",city+1);
            ncity = least(city);
            if(ncity==999)
            {
            ncity=0;
            printf("%d",ncity+1);
            cost +=DistMat[city][ncity];
            return;
            }
            mincost(ncity);
}
 int least(int c)
{
            int i,nc=999;
            int min = 999,kmin;
            for(i=0;i<n;i++)
            {
            if((DistMat[c][i]!=0)&&(visitedCity[i]==0))
            if(DistMat[c][i]+DistMat[i][c]<min)
            {
            min=DistMat[i][0] + DistMat[c][i];
            kmin=DistMat[c][i];
            nc=i;
            }
            }
            if(min!=999)
            cost+=kmin;
            return nc;
}
```

Output

```
 Enter Number of Cities :-->2
 Enter the Distance Matrix-->

  Enter Distance of Row:1
 12
 1
```

```
Enter Distance of Row:2
8
55
Cost Matrix :-->
    12 1
   8 55

Optimal Path is:--> :
1 --> 2 --> 1

Minimum cost is 9
```