10. dijkstra

```
#include <stdio.h>
#include <conio.h>
 #include INFINITY 9999
 #define  MAX 10

 void  dijkstra( int G[MAX][MAX], int n,
           int startnode );

 int main() {

   int G[MAX][MAX], i, j, n, u;
    printf("Enter no of vertices");
    scanf("%d", &n);
   printf("Enter the adjacency matrix\n");
    for(i=0; i<n; i++)
       for(j=0; j<n; j++)
           scanf("%d", &G[i][j]);
   printf("Enter the starting node");
    scanf("%d", &u);
      dijkstra(G, n, u);
     return 0;
   }

   void  dijkstra( int  G[MAX][MAX],
        int n,  int startnode){
   {
      int cost[MAX][MAX], ~~dij~~distance[MAX],
       pred[MAX];
```

```
int visited[MAX], count ,mindistane ;, nextnode,
   i, j ;
 for (i=0; i<n; i++)
     for (j =0 ; j<n ; j++)
          if (G[i][j] = =0)
       Cost[i][j] = INFINITY;
       else
       cost[i][j] = G[i][j];

   for (i= 0 ; i<n; i++)
       {
       distane [i] = Cost [ Starnode][i] ;
       pred[i] = startnode;
         visited [i] = 0;
         }
       distane [startnode]= 0;
       visited [startnode]= 1;
       count = 1;
       while (count <n-1) {
          mindistane = INFINITY;
          for (i=0 ; i<n; i++) {
       if (distane [i] < mindistane &&!
           visited [i]) {
          mindistane = distance[i];
             nextnode = i;

          }
          visited [nextnode] = 1;
          for (i=0 ; i<n; i++)
               if (!visited [i] ){
          if (mindistane [nextnode] [i] <
               distane [i] )

             {
```

```
distance [i] = mindistance + cost [nextnode]
    [i];
    pred [i] = nextnode;
    }
    count ++;
    for (i=0; i<n; i++)
        if (i != startnode) {
        printf (" \n Distance of node
    %d = %d", i, distance [i] );
        printf (" \n path= %d", i);
           j = i;
           do {
              j = pred [i];
              ~~printf(~~
              printf (" <- %d", j);
           } while (j != startnode);
        }}
```

Output:-

```
Enter the no. of Vertices :: 4
Enter the adjacency matrix ::
 0  1  1  1
 1  0  1  0
 1  1  0  1
 1  0  1  0

Enter the starting node :: 1
Distance of 0 = 1
path = 0 <- 1
Distance of 2 = 1
path = 2 <- 1
Distance of 3 = 2
```

12-1-23

```
path = 3 <- 0 <- 1.
```