ADA LAB PROGRAM

PROGRAM:-

a) Print all the nodes reachable from a given starting node in a digraph using BSF method.

b) check whether a given graph is connected or not using DFS method.

Modification:-

- given an undirected graph ,, print all components line by line.

```
#include<stdio.h>
# include < conio.h>

void insertq (int q[], int node, int *f, int *r)
   if ((*f == -1) && (*r == -1))
      {   (*f)++, (*r)++, q[*f] = node ;

      }
      else {
            (*r)++, q[*r] = node;
            }
         }
   int deleteq (int q[], int *f, int *r)
   { int temp;
     temp = q[*f];
      if (*f == *r) *f = *r = -1;
     else (*f)++;
     // return temp;
    }
void bfs (int n, int adj [][10], int visited [])
{ int q[20], f = -1, r = -1, v, i ;
   insertq (q, src, &f, &r);
   while ((f<=r) && (f != -1)
   { v= deleteq (q, &f, &r);
      if (visited [v] != 1)
        {
           visited [v] =1;
                printf("%d", v);
        }
```

```
for ( i=1 ;  i<=n ; i++)
        if ((adj [v][i] ==1) && (visited [i] != 1))
          insert q (q, i, &f, &r);
        }
    }
 Void DFS (int n, int cost [10][10], int u, int S[ ])
 { int v ;
   S[u]=1;
   for (v=0; v<n; v++)
      { if ((cost [u][v] ==1) && (S[v]==0))
              DFS (n, cost , v, s);
          }
      }
int main (){
    int n, v, s, adj [10][10], src , visidet [10], choice;
     int cost [10][10];
     int s [10], con, flag;
 for (;;){
  printf ("1. Print the Reachable nodes \n
              2. Check the connectivity of the graph \n
              3. exit\n");
      printf (" Enter the choice !");
          scanf ("%d", & choice);
    switch (choice) {
        case 1: printf ("enter number of vertices \n");
        scanf ("%d", &n);
        printf ("enter adjecency matrix \n");
          for ( i=1 ; i<=n ; i++).
         {
               visited [i] = 0;
for ( j=1 ; j <=n ; j++ )
    scanf ( "%d", &adj[i][j]);

}
printf (" enter starting vertex \n");
scanf (" %d" , &src ):
printf (" The nodes reachable are \n"):
   bfs (n, adj , src, visited);
      break;
Case2: printf ( "Enter number of nodes \n");
       scanf ("%d" , &n);
       printf (" Enter the adjacency matrix \n");
   for ( i=0; i < n ; i ++)
    { for (j =0; j < n ; j++)
         scanf ("%d", &cost [i][j]);

}
  con=0;
  for (j =0 ; j <n; j++)
   { for (i =0; i<n ; i++)
        s[i]=0;
      DFS (n , cost , j , s);
       flag =0;
      for (i=0 ; i<n; i++)
        { if (s[i] = =0)
             flag =1;
        } if ( flag ==0)
             con =1;
        }
```

```
if (con==1)
        printf ("Graph is connected\n");
    else
          printf ("Graph is not connected\n");
                  break;
                  default : exit(0);
              }
          }
      }
```

Modified Program

```
# include <stdio.h>
  void dfs(int);
    int a [10] [10], vis[10], n;
    void main ()
    { int  i, j, comp=1
    printf ("enter number of vertices \n");
      scanf ("%d", &n);
      printf ("enter adjacency matrix\n");
      for (i=1; i<=n; i++)
        { for (j=1; j<=n; j++)
          { scanf ("%d", & a[i][j]);
          }
        }
      for (i=1 ; i<=n ; i++)
          vis [i] = 0;
        for (i= 1; i<=n; i++){
          if (vis[i]==0) {
              printf ("component %d\n", comp);
                comp++;
                dfs (i);
                printf("\n");
              }
            }
          }
```

```
void dfs (int v)
  { int i ;
    vis [v] = 1;
     printf ("%d\t", v);
      for (i=1 ; i<=n ; i++)
       {  if (a[v][i] == 1&& vis[i]==0)
              dfs [[i];
          }
     }
```