9) b) write a program to check whether given graph is connected or not using DFS method :

```
# include <stdio.h>
# define max 10

int visited [MAX];
int adj [MAX] [MAX];
int n;

void DFS (int v) {
    visited [v] = 1;
    printf (" %d ", v);

    for (int i = 0 ; i<n ; i++) {
      if (adj[v][i] == 1 && !visited [i]) {
            DFS(i);
        }
      }
   }

int main () {

printf ("enter number of vertices:");
scanf ("%d", &n);

printf ("enter adjacency martix :\n");
for (int i = 0; i<n ; i++)
{
   for (int j = 0 ; j<n ; j++)
   {
```

```
scanf ("%d", & adj [i] [j]);
  }
}
for (int i=0; i<n; i++)
    visited [i] =0;

printf ("DFS Traversal starting from vertex
        0:\n");
DFS (0);
return 0;
}.
```

output:

```
enter number of vertices: 4
enter adjacency matrix:
0 1 1 0
0 0 1 0
0 0 0 1
0 0 0 0

DFS Traversal starting from vertex 0:
0 1 2 3.
```