

- qa) @ write a program to traverse a graph using BFS method.
⑥ write a program to check whether given graph is connected or not using DFS method

```
#include <stdlib.h>
int graph[20][20], visited[20], n;
void BFS(int start){
    int queue[20], front=0, rear=0;
    visited[start]=1;
    queue[rear++]=start;
    while(front<rear){
        int node=queue[front++];
        printf("%d", node);
        for(int i=0; i<n; i++){
            if(graph[node][i]==1 && !visited[i]){
                visited[i]=1;
                queue[rear++]=i;
            }
        }
    }
}
```

```
int main(){
    int start;
    printf("Enter number of vertices: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix: [n]\n");
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            scanf("%d", &graph[i][j]);
    for(int i=0; i<n; i++)
        visited[i]=0;
    printf("Enter starting vertex: ");
    scanf("%d", &start);
    printf("BFS Traversal: ");
    BFS(start);
    return 0;
}
```

Q1

Enter number of vertices : 4

Enter adjacency matrix :

0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

Enter starting vertex : 2

BFS Traversal : 2 0 3 1

b) #include <stdio.h>

#define MAX 10

int visited[MAX];

int adj[MAX][MAX];

int n;

void DFS(int v){

visited[v] = 1;

printf("vd", v);

for (int i=0; i<n; i++) {

if (adj[v][i]==1 & & visited[i]==0)

DFS(i);

}

int main(){

printf("Enter number of vertices : ");

scanf("%d", &n);

printf("Enter adjacency matrix : ");

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: ");

DFS(0);

} return 0;

olp Enter number of vertices : 4

Önter adjacency matrix:

卷之三

— 1 —

- 0 -

卷之三

DFS traversal starting from vertex 0:

卷之三

LAB-10

Given a file of N employee records with a set K of keys $\{k_1, k_2, \dots, k_K\}$.
the records in file F .

LAB-10

Given a file of N employee records with a set K of keys (6 digits) which uniquely determine the records in file F . Assume that file F is maintained in memory by a hash table (HT) of memory addresses (2 digits). Assume that file F contains L integers with L as the set of memory addresses in last integers.