S.No: 15	Exp. Name: ***Find the no.of nodes that can be visited in a directed graph***	Date:2024-06-02

ID: 92132321021

Aim:

Write a program in C for the following operations on Graph(G) of Cities

1. Create a Graph of N cities using the Adjacency Matrix.
2. Print all the nodes reachable from a given starting node in a directed graph using DFS or BFS

Source Code:

dfsBfs.c

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

//write your code here..
#define MAX 100

int a[MAX][MAX], visited[MAX], n;

void bfs(int start) {
    int queue[MAX], front = -1, rear = -1;
    int i, current;

    queue[++rear] = start;
    visited[start] = 1;

    while (front != rear) {
        current = queue[++front];
        for (i = 1; i <= n; i++) {
            if (a[current][i] == 1 && !visited[i]) {
                queue[++rear] = i;
                visited[i] = 1;
                printf("%d ", i);
            }
        }
    }
}

void dfs(int start) {
    int stack[MAX], top = -1;
    int i, current;

    stack[++top] = start;
    visited[start] = 1;

    while (top != -1) {
        current = stack[top];
        for (i = 1; i <= n; i++) {
            if (a[current][i] == 1 && !visited[i]) {
                stack[++top] = i;
                visited[i] = 1;
                printf("%d ", i);
                break;
            }
```

```
        }
        if (i > n) {
            top--;
        }
    }
}
int main()
{   int ch, start, i,j;
    printf("no of cities: ");
    scanf("%d",&n);
    printf("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++)
        visited[i]=0;
    printf("starting city: ");
    scanf("%d",&start);
    printf("1. BFS");
    printf("\n2. DFS");
    printf("\n3: Exit");
    printf("\nChoice: ");
    scanf("%d", &ch);
    switch(ch)
    {   case 1: printf("Cities reachable from %d:\n",start);
            bfs(start);
                break;
        case 2: printf("Cities reachable from %d:\n",start);
            dfs(start);
                break;
        case 3: exit(0);
        default: printf("Invalid choice\n");
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
no of cities: 3
adjacency matrix: 0 1 1
0 0 1
0 0 0
starting city: 1
1. BFS 1
2. DFS 1
3: Exit 1
Choice: 1
Cities reachable from 1:
2 3

Test Case - 2
User Output
no of cities: 4
adjacency matrix: 0 1 0 1
0 0 1 0
0 0 0 1
0 0 0 0
starting city: 1
1. BFS 2
2. DFS 2
3: Exit 2
Choice: 2
Cities reachable from 1:
2 3 4

ID: 92132321021