Name: Gaurav singh

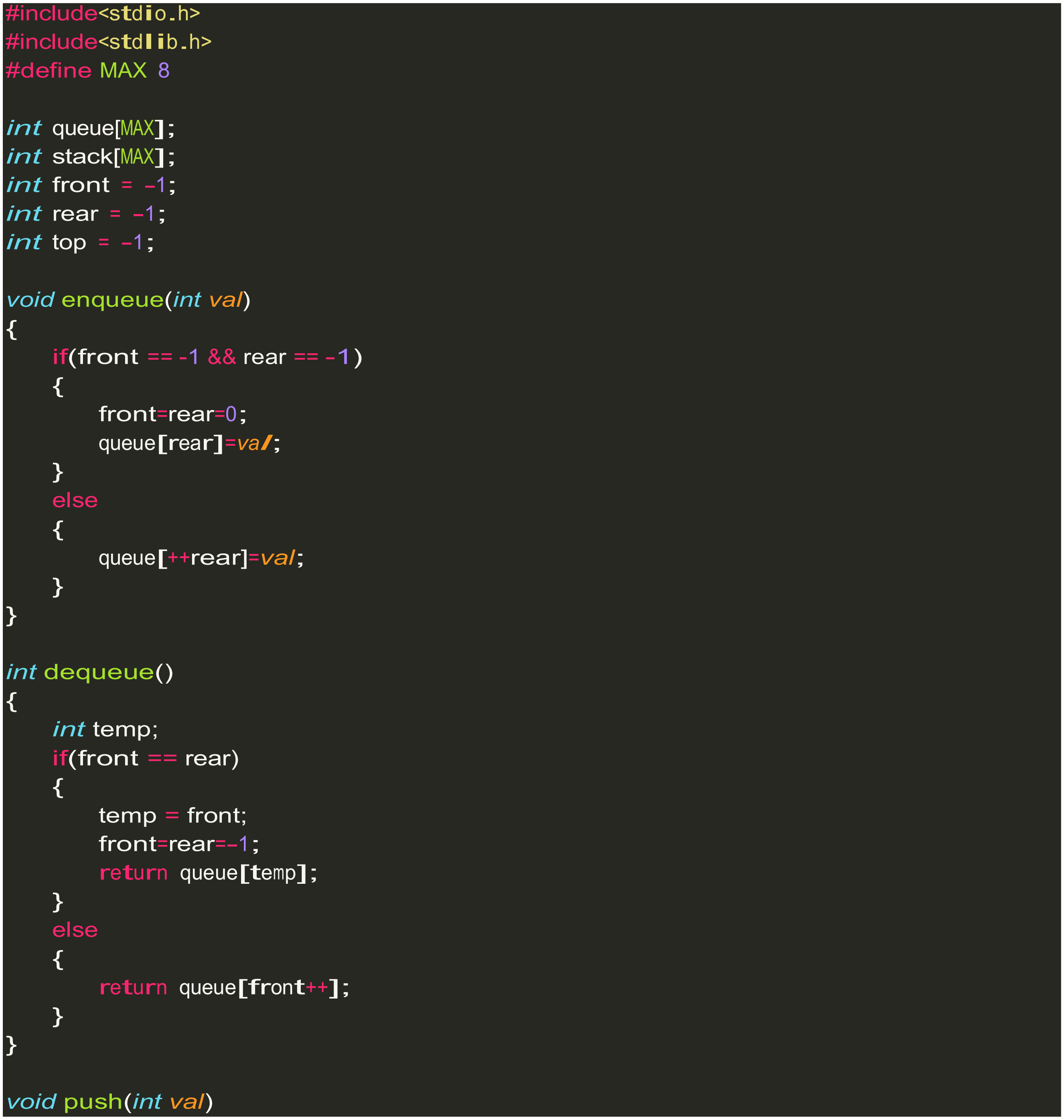
Sec: C4 Roll No: 70

Sub: DSA Lab

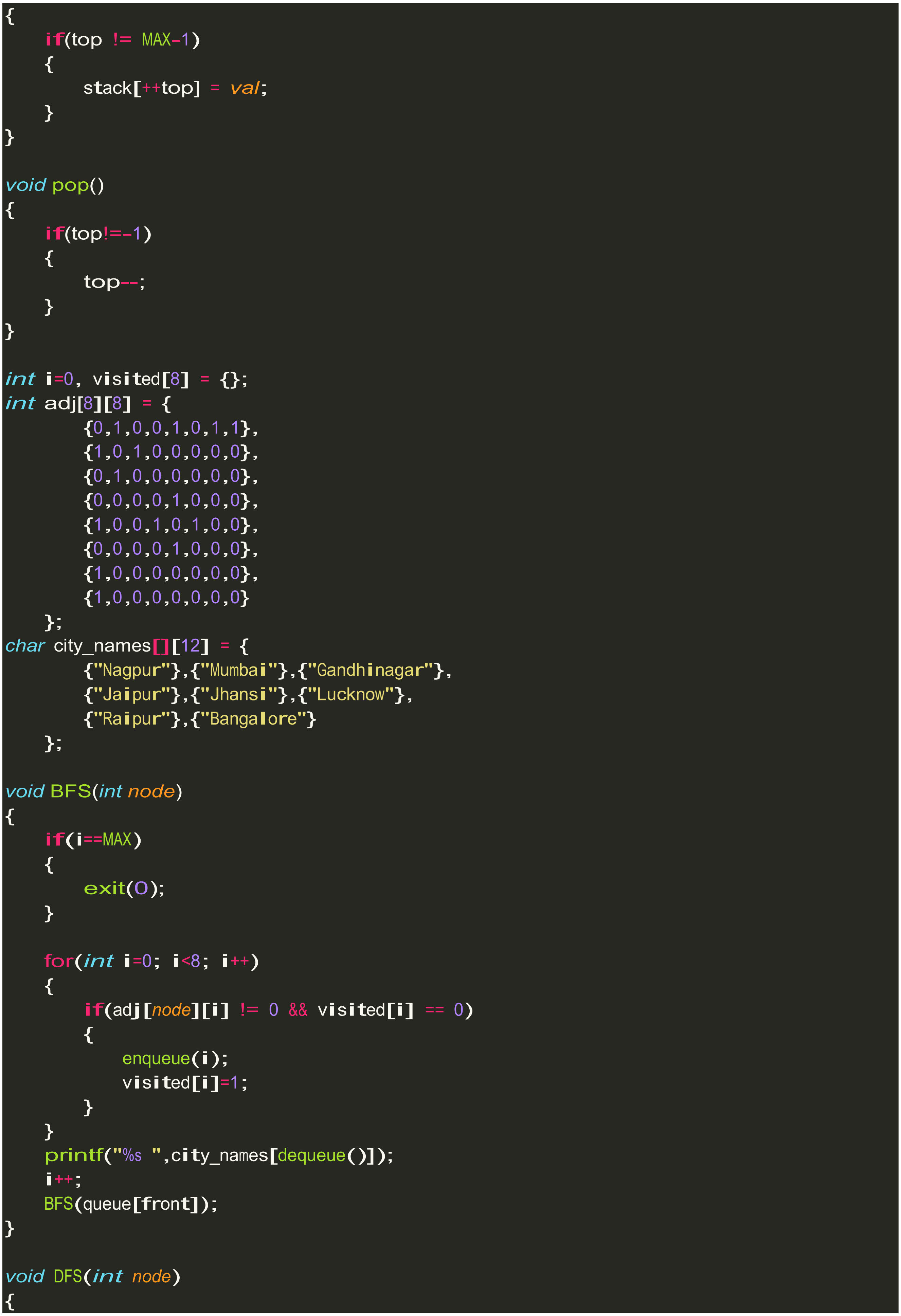
PRACTICAL 07

Consider only blue color marked cities as a node. Represent graph as Adjacency Matrix. Find BFS/T and DFS/T for any 8 cities from the map. It should include Nagpur city necessarily. Start your traversal from Nagpur. Also find the Minimum Spanning tree for the listed cities. Calculate cost of MST.

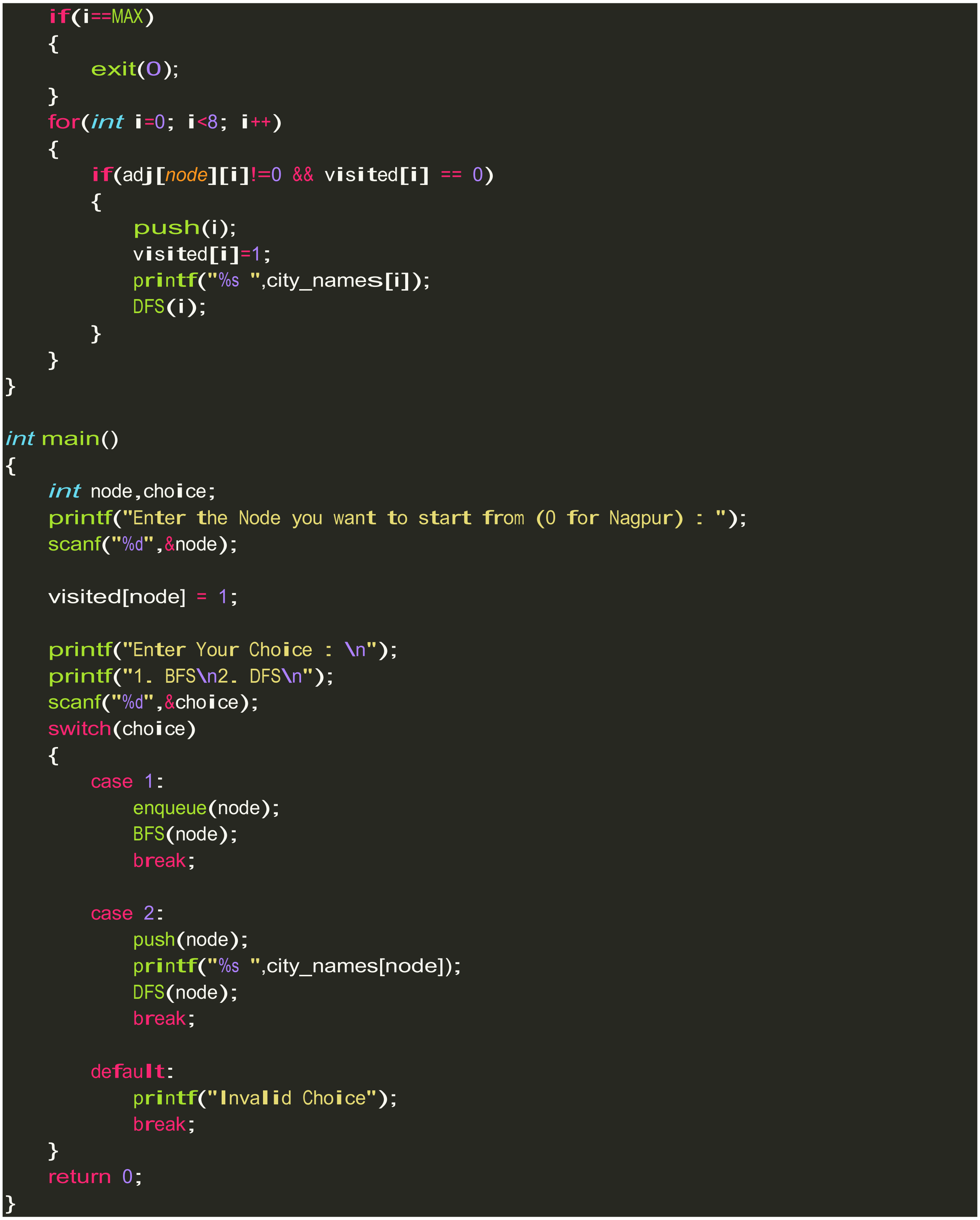
CODE:



```
#include<stdio.h>
#include<stdlib.h>
#define MAX 8

int queue[MAX];
int stack[MAX];
int front = -1;
int rear = -1;
int top = -1;

void enqueue(int val)
{
    if(front == -1 && rear == -1)
    {
        front=rear=0;
        queue[rear]=val;
    }
    else
    {
        queue[++rear]=val;
    }
}

int dequeue()
{
    int temp;
    if(front == rear)
    {
        temp = front;
        front=rear=-1;
        return queue[temp];
    }
    else
    {
        return queue[front++];
    }
}

void push(int val)
```



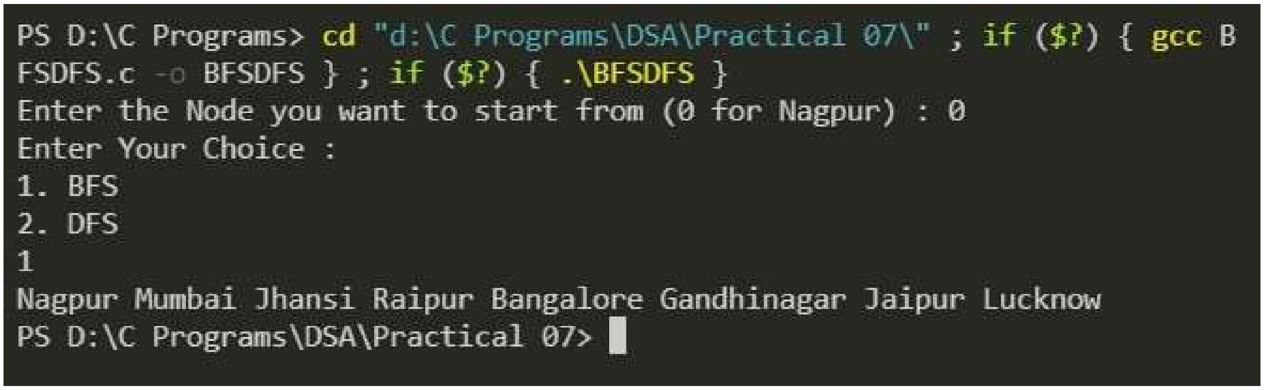
```
{
    if(top != MAX-1)
    {
        stack[++top] = val;
    }
}

void pop()
{
    if(top!=-1)
    {
        top--;
    }
}

int i=0, visited[8] = {};
int adj[8][8] = {
        {0,1,0,0,1,0,1,1},
        {1,0,1,0,0,0,0,0},
        {0,1,0,0,0,0,0,0},
        {0,0,0,0,1,0,0,0},
        {1,0,0,1,0,1,0,0},
        {0,0,0,0,1,0,0,0},
        {1,0,0,0,0,0,0,0},
        {1,0,0,0,0,0,0,0}
    };
char city_names[][12] = {
        {"Nagpur"},{"Mumbai"},{"Gandhinagar"},
        {"Jaipur"},{"Jhansi"},{"Lucknow"},
        {"Raipur"},{"Bangalore"}
    };

void BFS(int node)
{
    if(i==MAX)
    {
        exit(0);
    }

    for(int i=0; i<8; i++)
    {
        if(adj[node][i] != 0 && visited[i] == 0)
        {
            enqueue(i);
            visited[i]=1;
        }
    }
    printf("%s ",city_names[dequeue()]);
    i++;
    BFS(queue[front]);
}

void DFS(int node)
{
```



```
    if(i==MAX)
    {
        exit(0);
    }
    for(int i=0; i<8; i++)
    {
        if(adj[node][i]!=0 && visited[i] == 0)
        {
            push(i);
            visited[i]=1;
            printf("%s ",city_names[i]);
            DFS(i);
        }
    }
}

int main()
{
    int node,choice;
    printf("Enter the Node you want to start from (0 for Nagpur) : ");
    scanf("%d",&node);

    visited[node] = 1;

    printf("Enter Your Choice : \n");
    printf("1. BFS\n2. DFS\n");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            enqueue(node);
            BFS(node);
            break;

        case 2:
            push(node);
            printf("%s ",city_names[node]);
            DFS(node);
            break;

        default:
            printf("Invalid Choice");
            break;
    }
    return 0;
}
```

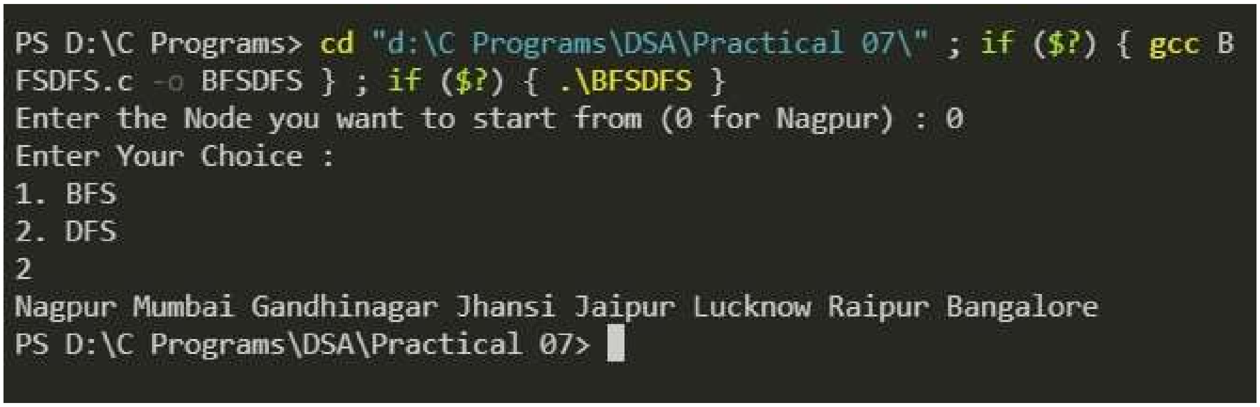
OUTPUT:

#1 BFS



```
PS D:\C Programs> cd "d:\C Programs\DSA\Practical 07\" ; if ($?) { gcc B
FSDFS.c -o BFSDFS } ; if ($?) { .\BFSDFS }
Enter the Node you want to start from (0 for Nagpur) : 0
Enter Your Choice :
1. BFS
2. DFS
1
Nagpur Mumbai Jhansi Raipur Bangalore Gandhinagar Jaipur Lucknow
PS D:\C Programs\DSA\Practical 07>
```

#2 DFS



```
PS D:\C Programs> cd "d:\C Programs\DSA\Practical 07\" ; if ($?) { gcc B
FSDFS.c -o BFSDFS } ; if ($?) { .\BFSDFS }
Enter the Node you want to start from (0 for Nagpur) : 0
Enter Your Choice :
1. BFS
2. DFS
2
Nagpur Mumbai Gandhinagar Jhansi Jaipur Lucknow Raipur Bangalore
PS D:\C Programs\DSA\Practical 07>
```