VISHWAKARMA INSTITUTE OF TECHNOLOGY

NAME	Arpit Sudhir Vidhale
ROLL NO.	60
DIVISION	CS-D
BATCH	B3
PRN NO.	12111229

DS LAB ASSIGNMENT 7

Question:
Write a Program to accept a graph from user and represent it with Adjacency Matrix and perform BFS and DFS traversals on it.

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct queue
{
    int size;
    int f;
    int r;
    int *arr;
};

int isEmpty(struct queue *q)
{
    if (q->r == q->f)
    {
        return 1;
    }
    return 0;
}

int isFull(struct queue *q)
{
```

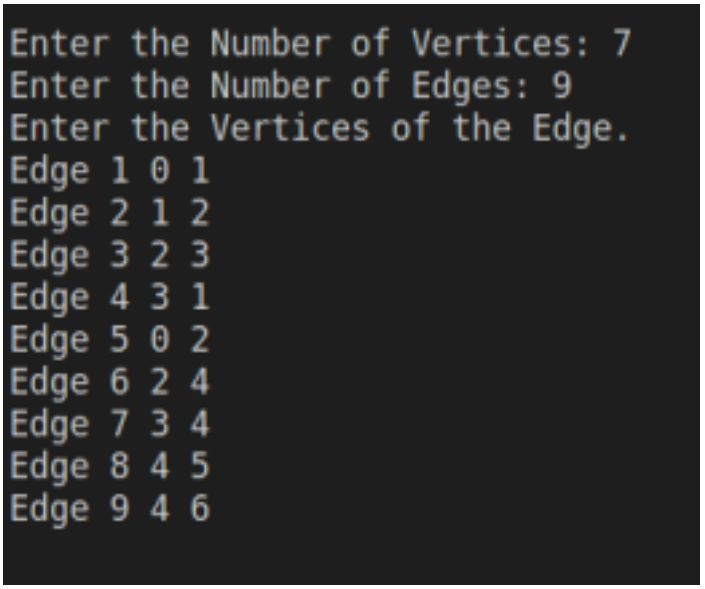
```
    if (q->r == q->size - 1)
    {
        return 1;
    }
    return 0;
}

void enqueue(struct queue *q, int val)
{
    if (isFull(q))
    {
        printf("This Queue is full\n");
    }
    else
    {
        q->r++;
        q->arr[q->r] = val;
    }
}

int dequeue(struct queue
*q) {
    int a = -1;
    if (isEmpty(q))
    {
        printf("This Queue is
    empty\n"); }
    else
    {
        q->f++;
        a = q->arr[q->f];
    }
    return a;
}

void DFS(int s)
{
    printf("%d ", s);
    visited[s] = 1;
    for (int j = 0; j < nv; j++)
    {
        if (a[s][j] == 1 && !visited[j])
```

```
        {
            DFS(j);
        }
    }
}
void BFS(int s)
{
    struct queue q;
    q.size = 400;
    q.f = q.r = 0;
    q.arr = (int *)malloc(q.size *
    sizeof(int)); printf("%d ", s);
    visited[s] = 1;
    enqueue(&q, s);
    while (!isEmpty(&q))
    {
        int node = dequeue(&q);
        for (int j = 0; j < nv; j++)
        {
            if (a[node][j] == 1 && visited[j] ==
            0) {
                printf("%d ", j);
                visited[j] = 1;
                enqueue(&q, j);
            }
        }
    }
}
int main()
{
     int ch;
    printf("\nEnter the Number of Vertices:
    "); scanf("%d", &nv);

    for (int i = 0; i < nv; i++)
    {
        for (int j = 0; j < nv; j++)
        {
            a[i][j] = 0;
            a[j][i] = 0;
        }
        visited[i] = 0;
```

```
}
printf("Enter the Number of Edges:
"); scanf("%d", &ne);
printf("Enter the Vertices of the
Edge.\n"); for (int k = 0; k < ne; k++)
{
    printf("Edge %d ", k + 1);
    scanf("%d %d", &v1, &v2);
    a[v1][v2] = 1;
    a[v2][v1] = 1;
}

do
{
    printf("\n\nEnter your
    Choice\n1.BFS\n2.DFS\n3.Exit:\n"); scanf("%d", &ch);
    switch (ch)
    {
    case 1:
        for (int i = 0; i < nv; i++)
        {
            visited[i] = 0;
        }
        printf("\nEnter the starting poition: ");
        scanf("%d", &s);
        printf("\n");
        BFS(s);

    case 2:
        for (int i = 0; i < nv; i++)
        {
            visited[i] = 0;
        }
        printf("\nEnter the starting poition: ");
        scanf("%d", &s);
        printf("\n");
        DFS(s);
        break;

    case 3:
        printf("Exit!!");
        break;
```

```
            default:
                printf("Invalid Choice!!");
                break;
            }
        } while (1);
        return 0;
    }
```

Output:



```
Enter your Choice
1.BFS
2.DFS
3.Exit:
1

Enter the starting poition: 3

3 1 2 4 0 5 6

Enter your Choice
1.BFS
2.DFS
3.Exit:
2

Enter the starting poition: 4

4 2 0 1 3 5 6

Enter your Choice
1.BFS
2.DFS
3.Exit:
3
Exit!!
```