Aim:

Write a program to implement Breadth First Search of a graph.

Source Code:

GraphsBFS.c

```c
#include<stdio.h>
#include<stdlib.h>
#define MAX 99
struct node
{
    struct node *next;
    int vertex;
};
typedef struct node * GNODE;
GNODE graph[20];
int visited[20];
int queue[MAX],front=-1,rear=-1;
int n;
void insertQueue(int vertex)
{
    if (rear==MAX-1)
    printf("queue over flow.\n");
    else
    {
        if(front==-1)
        {
            front=0;
        }
        rear++;
        queue[rear]=vertex;
    }
}
int isemptyQueue()
{
    if(front==-1||front>rear)
    return 1;
    else
    return 0;
}
int deleteQueue()
{
    int deleteItem;
    if(front==-1||front>rear)
    {
        printf("Queue underflow.\n");
        exit(0);
    }
    deleteItem=queue[front];
    front++;
    return deleteItem;
}
```

```c
void BFS(int v)
{
    int w;
    insertQueue(v);
    while(! isemptyQueue())
    {
        v=deleteQueue();
        printf("%d\n",v);
        visited[v]=1;
        GNODE g;
        for(g=graph[v];g!=NULL;g=g->next)
        {
            w=g->vertex;
            if(visited[w]==0)
            {
                insertQueue(w);
                visited[w]=1;
            }
        }
    }
}
void main()
{
    int N,E,s,d,i,j,v,w;
    GNODE p,q;
    printf("Enter the number of vertices : ");
    scanf("%d",&N);
    printf("Enter the number of edges : ");
    scanf("%d",&E);
    for(i=1;i<=E;i++)
    {
        printf("Enter source : ");
        scanf("%d",&s);
        printf("Enter destination : ");
        scanf("%d",&d);
        q=(GNODE)malloc(sizeof(struct node));
        q->vertex=d;
        q->next=NULL;
        if(graph[s]==NULL)
        {
            graph[s]=q;
        }
        else
        {
            p=graph[s];
            while(p->next!=NULL)
            p=p->next;
            p->next=q;
        }
    }
    for(i=0;i<=n;i++)
    visited[i]=0;
    printf("Enter Start Vertex for BFS : ");
    scanf("%d",&v);
    printf("BFS of graph : \n");
    BFS(v);
```

```
    printf("\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of vertices : 5
Enter the number of edges : 5
Enter source : 1
Enter destination : 2
Enter source : 1
Enter destination : 4
Enter source : 4
Enter destination : 2
Enter source : 2
Enter destination : 3
Enter source : 4
Enter destination : 5
Enter Start Vertex for BFS : 1
BFS of graph :
1
2
4
3
5

Test Case - 2
User Output
Enter the number of vertices : 4
Enter the number of edges : 3
Enter source : 1
Enter destination : 2
Enter source : 2
Enter destination : 3
Enter source : 3
Enter destination : 4
Enter Start Vertex for BFS : 2
BFS of graph :
2
3
4