```
//Created By Ritwik Chandra Pandey
//On 5th Nov' 2021
//Graph Traversal: BFS implementation


#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 Overflow.\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;

}

void BFS(int v) {
	int w;
	insertQueue(v);
	while(!isEmptyQueue()){
	v = deleteQueue(v);
	printf("%d\n",v);
	visited[v] =1;
	GNODE g = graph[v];
	for(;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;
	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=1;i<=n;i++)
		visited[i]=0;
	printf("Enter Start Vertex for BFS : ");
	scanf("%d", &v);
	printf("BFS of graph : \n");
	BFS(v);
	printf("\n");
}
```