## Program 9a

Write a program to traverse a graph using BFS method.

```
Code:
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
struct Queue {
       int items[MAX];
       int front, rear;
};
void initQueue(struct Queue* q) {
  q->front = -1;
  q->rear = -1;
}
int isEmpty(struct Queue* q) {
       return q->front == -1;
}
void enqueue(struct Queue* q, int value) {
       if (q->rear == MAX - 1) {
    printf("Queue is full\n");
    return;
       }
```

```
if (q->front == -1) {
    q->front = 0;
  q->rear++;
  q->items[q->rear] = value;
}
int dequeue(struct Queue* q) {
       if (isEmpty(q)) {
    printf("Queue is empty\n");
    return -1;
       }
       int item = q->items[q->front];
  q->front++;
       if (q->front > q->rear) {
    q->front = q->rear = -1;
       return item;
}
void BFS(int graph[MAX][MAX], int n, int startVertex) {
       int visited[MAX] = \{0\};
       struct Queue q;
  initQueue(&q);
```

```
visited[startVertex] = 1;
  enqueue(&q, startVertex);
  printf("BFS Traversal: ");
       while (!isEmpty(&q)) {
    int currentVertex = dequeue(&q);
    printf("%d", currentVertex);
       for (int i = 0; i < n; i++) {
      if (graph[currentVertex][i] == 1 && !visited[i]) {
         visited[i] = 1;
         enqueue(&q, i);
 printf("\n");
int main() {
       int graph[MAX][MAX], n, startVertex;
printf("Enter the number of vertices in the graph: ");
  scanf("%d", &n);
printf("Enter the adjacency matrix of the graph:\n");
       for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
       scanf("%d", &graph[i][j]);
```

```
}

printf("Enter the starting vertex (0 to %d): ", n - 1);

scanf("%d", &startVertex);

BFS(graph, n, startVertex);

return 0;
}

Enter the number of vertices in the graph Enter the adjacency matrix of the graph:
```

```
Enter the number of vertices in the graph: 4
Enter the adjacency matrix of the graph:
0 1 0 0
1 0 1 1
0 1 0 1
0 1 1 0
Enter the starting vertex (0 to 3): 0
BFS Traversal: 0 1 2 3
```

10 10	shr. 67
WAP to traverse a graph using BFS method	a -> rear ++;
Wat to district to graph the say	a - items (a -> rear) = value;
£include (staio. h)	
#include (statisty)	
2 22 23 25 25	int dequere (struct accuera) ?
# define NAX 100	printf (" dueue is emply \n");
Start Sune &	Yelum 1;
Street Subject [MAX];	The state of the s
int hour rear;	
	int ilem . q > items [q -> front];
	9-> fronk++;
void initarene (struct Quesit " ?) (	9, > front +1; if (2-7 front > 9, -> year) f 9, -> front = 9, -> year = -1;
9-> fone 1;	3 1 Sant = 9 - New )
9-7 s(ar=-1;	
	return ileun;
ist strong (struct queue*a)?	
return fact == -(;	Comment of the commen
	Void BFS (int graph CHAY] CHAY], int m, int start Volcx) int visited CHAY] for.
Secret December interpret	Smut Bueut a:
Void comme (sauce Quene + g, int value) {	init Queue (40);
posit ("Queue is but ha");	
returns	wisited [stor+ Verkx] = 1;
	Enqueux (Lag, start Verkx);
11(9→hout ===1) f	privit ("BES Traversal:");
9-hout = 0;	hard are awith 1
	while (! is Emory (fg)) {
	int current levex > dequeux (lg):

	sture 67/
	Quipu <sup>4</sup>
prints ("1-2", coveret Verks);	
for (intia 0; i < n; iss) f	Enter the number of vertices in the graph: 4  Enter the adjacency matrix of the graph:
if Caraph Convent Verky Cities I the 1 VISITED CITY	Ever the adjacency matrix of the graph:
visikd (il ali	0 1 0 0
enquene (Eq. 11);	0 41 0 1
	0 110
3	Ener the starting verte (0 to 5): 0
points ("\n");	D-3 ([MINE] - 0 1
int wain () i	4 70 0 10 1
int graph [MAX] (MAX], M, Start Verikx;	
proof ("Enex thee no of vertices in the graph."); Scanf ("t-d", slan);	
scan ("1.d" , la);	
print ("Enter the adjacency matrix of the graph 12");	
Jor (int i: 0; i <n; 112)1<="" td=""><td></td></n;>	
for (int 120; jen; jen); jen; jen); scanf("red", Legraph ("JC(1));	
3 1409	
print ("Entry the starting vertex (0 to ted ):", "1);	Carlos Ca
Stant ("166", & Stort Verkx);	
855 (graph, n, start Verke);	
youm o:	