

Write a program to traverse a graph using BFS method.

```
#include <stdio.h>
```

```
#define MAX 10
```

```
int queue[MAX], front = -1, rear = -1;
```

```
int visited[MAX];
```

```
void enqueue(int v)
```

```
{
```

```
    if (front == -1)
```

```
        front = 0;
```

```
    queue[++rear] = v;
```

```
}
```

```
int dequeue()
```

```
{
```

```
    return queue[front++];
```

```
}
```

```
void bfs(int graph[MAX][MAX], int n, int start)
```

```
{
```

```
    int i, v;
```

```
    for (i = 0; i < n; i++)
```

```
        visited[i] = 0;
```

```
        enqueue(start);
```

```
        visited[start] = 1;
```

```
printf("BFS Traversal: ");

while (front <= rear)
{
    v = dequeue();
    printf("%d ", v);

    for (i = 0; i < n; i++)
    {
        if (graph[v][i] == 1 && visited[i] == 0)
        {
            enqueue(i);
            visited[i] = 1;
        }
    }
}

int main()
{
    int graph[MAX][MAX], n, i, j, start;

    printf("Enter number of vertices: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix row by row:\n");
    for (i = 0; i < n; i++)
    {
```

```

    printf("Row %d: ", i);

    for (j = 0; j < n; j++)

        scanf("%d", &graph[i][j]);

}

printf("Enter starting vertex: ");

scanf("%d", &start);

bfs(graph, n, start);

return 0;
}

```

OUTPUT:

```

60 }           scanf("%d", &graph[i][j]);
61
62
63
64

printf("Enter starting vertex: ");
scanf("%d", &start);

bfs(graph, n, start);

return 0;
}

PS C:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa"
PS C:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa" ; if ($?) { gcc BFS.c -o BFS } ; if ($?) { .\BFS }
PS C:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa> Enter number of vertices: 4
Enter adjacency matrix row by row:
Row 0: 0 1 1 0
Row 1: 1 0 0 1
Row 2: 1 0 0 1
Row 3: 0 1 1 0
Enter starting vertex: 0
BFS Traversal: 0 1 2 3
PS C:\Users\WISCHAL\OneDrive\Documents\Desktop\dsa>

```