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

#define max_nodes 7

int queue[max_nodes], visited[max_nodes], adjacency_matrix[max_nodes][max_nodes];
int front = -1, rear = -1;

void enqueue(int node)
{
if (rear == max_nodes - 1)
{
printf("Queue Overflow\n");
exit(1);
}
if (front == -1)
front = 0;
rear++;
queue[rear] = node;
}

int
dequeue()
{
int node;
if (front == -1)
{
printf("Queue Underflow\n");
exit(1);
}
node = queue[front];
if (front == rear)
{
rear = front = -1;
}
else
{
front++;
}

return node;
```

```
}

void bfs(int start_node, int num_nodes)
{
int i, current_node;
enqueue(start_node);
visited[start_node] = 1;

while(front != -1)
{
current_node = dequeue();
printf("%d" , current_node);
for(i = 0; i< num_nodes; i++)
{
if (adjacency_matrix[current_node][i] == 1 && !visited[i])
{
visited[i] = 1;
enqueue(i);
}
}

}
}

int main()
{
int i,j, num_nodes, start_node;
printf("Enter the number of nodes: ");
scanf("%d", &num_nodes);
printf("Enter the adjacency matrix:\n");
for (i = 0; i<num_nodes; i++)
{
for (j = 0; j < num_nodes; j++)
{
scanf("%d", &adjacency_matrix[i][j]);
}
}
printf("Enter the start node: ");
scanf("%d", &start_node);
bfs(start_node, num_nodes);
```

```
return 0;
}
```