

Write a program to traverse a graph using BFS method.

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

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

```
#define MAX 20
```

```
int queue[MAX];
```

```
int front=0;
```

```
int rear=-1;
```

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

```
void enqueue(int val){
```

```
    if(rear==MAX-1){
```

```
        return;
```

```
    }
```

```
    rear++;
```

```
    queue[rear]=val;
```

```
}
```

```
int deque(){
```

```
    if(front==-1 || front>rear){
```

```
        return -1;
```

```
    }
```

```
    int c=queue[front];
```

```
    front++;
```

```
    return c;
```

```
}
```

```
void graph(int G[MAX][MAX],int st,int v){
```

```
    visited[st]=1;
```

```
    enqueue(st);
```

```
    int k;
```

```
    while(front<=rear){
```

```
        k=deque();
```

```
        printf("%d\n",k);
```

```
        for(int i=0;i<v;i++){
```

```
            if( G[k][i]==1 && visited[i]!=1){
```

```
                enqueue(i);
```

```
                visited[i]=1;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main(){
```

```
    int v;
```

```
    int G[MAX][MAX];
```

```
    printf("enter no of vertices\n");
```

```
    scanf("%d",&v);
```

```
    printf(" enter adjacency matrix\n");
```

```
    for(int i=0;i<v;i++){
```

```
        for(int j=0;j<v;j++){
```

```

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

    }

}

int st;

printf("enter starting vertex\n");

scanf("%d",&st);

graph(G,st,v);

return 0;

}

```

Output:

```

enter no of vertices
4
enter adjacency matrix
0 1 1 0
1 0 1 1
1 1 0 0
0 1 0 0
enter starting vertex
0
0
1
2
3

```