```
/// Author: Prathamesh Patil - Exp: 2 BFS and
DFS
// BFS:
#include<iostream>
#include<queue>
#define N 6
using namespace std;

int visited[N]{0};
queue<int> q;
int graph[N][N];


void bfs(int v){
    visited[v] = 1;
    cout<<v<<" ";
    for(int i=0;i<N;i++){
        if(graph[v][i] == 1 && !visited[i]){
            q.push(i);
            visited[i]=1;
        }
    }
    if(!q.empty()){
        int num = q.front();
        q.pop();
        bfs(num);
    }
}

int main(){
    int v;

    cout<<"Enter the graph in matrix of
["<<N<<"X"<<N<<"]: "<<endl;
    for(int i=0;i<N;i++){
        for(int j=0;j<N;j++){
            cin>>graph[i][j];
        }
    }

    cout<<"Enter the starting point: ";
    cin>>v;

    cout<<"The traversal order is: "<<endl;
    bfs(v);

    return 0;
}
```

```
// DFS
// Author: Prathamesh Patil
#include <iostream>
#include <stack>
#define n 6
using namespace std;

int visited[n]{0};

void DFS(int graph[n][n], int start){
    visited[start] = 1;
    for(int i=0;i<n;i++){
        if(graph[start][i] == 1 && !visited[i]){
            cout<<"The edge is: ["<<start<<",
"<<i<<"]"<<endl;
            DFS(graph, i);
        }
    }
}

int main(){
    int v;
    int graph[n][n];
    cout<<"Enter the matrix
("<<n<<"X"<<n<<"):"<<endl;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            cin>>graph[i][j];
        }
    }
    cout<<"Enter the vertex to start from: ";
    cin>>v;
    cout<<"DFS of Graph: "<<endl;
    DFS(graph, v);
    return 0;
}
```

OUTPUT:

```
/tmp/dmIX7BM/yt.o
Enter the graph in matrix of [6X6]:
1 0 0 1 0 1
1 0 0 0 0 1
10 0 0 1
 1 0 0 1 0 1
 1 0 0 1 0 1
0 1 0 1 0 1 0
1 0 1 1 0 1 0 1
0 1 0 1 1 0 1 0
Enter the starting point: The traversal
0 3 5 1 2 4
```