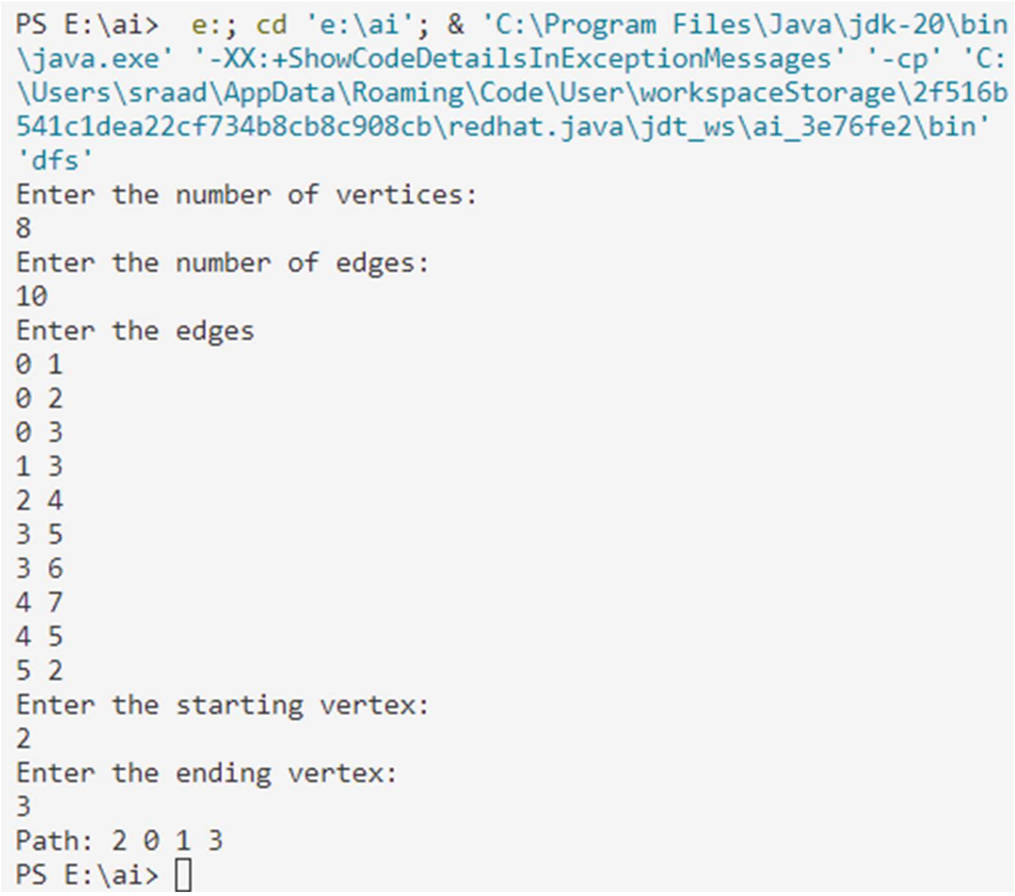
DFS

-21Z201

```
import java.util.*;
class dfs {
    private LinkedList<Integer> l[];
    private boolean v[];
    private ArrayList<Integer> path;
    void newgraph(int n) {
        l = new LinkedList[n];
        v = new boolean[n];
        path = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            l[i] = new LinkedList<Integer>();
            v[i] = false;
        }
    }
    void edge(int i, int j) {
        l[i].add(j);
        l[j].add(i);
    }
    void DFS(int x, int end) {
        v[x] = true;
        path.add(x);
        if (x == end) {
            printPath();
            return;
        }
        Iterator<Integer> it = l[x].listIterator();
        while (it.hasNext()) {
            int y = it.next();
            if (!v[y]) {
                DFS(y, end);
            }
        }
        path.remove(path.size() - 1);
    }
    void printPath() {
        System.out.print("Path: ");
        for (int node : path) {
            System.out.print(node + " ");
        } System.out.println();
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        dfs ob = new dfs();
        System.out.println("Enter the number of vertices:");
        int n = sc.nextInt();
        ob.newgraph(n);
        System.out.println("Enter the number of edges:");
        int e = sc.nextInt();
        System.out.println("Enter the edges");
        for (int i = 0; i < e; i++) {
            int x = sc.nextInt();
            int y = sc.nextInt();
            ob.edge(x, y);
        }
        System.out.println("Enter the starting vertex:");
        int start = sc.nextInt();
        System.out.println("Enter the ending vertex:");
        int end = sc.nextInt();
        sc.close();
        ob.DFS(start, end);
    }
}
```

OUTPUT:



```
PS E:\ai>  e:; cd 'e:\ai'; & 'C:\Program Files\Java\jdk-20\bin
\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\sraad\AppData\Roaming\Code\User\workspaceStorage\2f516b
541c1dea22cf734b8cb8c908cb\redhat.java\jdt_ws\ai_3e76fe2\bin'
'dfs'
Enter the number of vertices:
8
Enter the number of edges:
10
Enter the edges
0 1
0 2
0 3
1 3
2 4
3 5
3 6
4 7
4 5
5 2
Enter the starting vertex:
2
Enter the ending vertex:
3
Path: 2 0 1 3
PS E:\ai> 
```