Assignment 3
Traversal and Connectivity

Notes

- Each problem has two cases associated with it: small and large. Each of these test cases accounts for half of the total marks allocated to that specific problem.
- Some of the test cases will be large. You receive full marks if you pass all test cases.
- You can use Java, C++, Python.
- There is a restriction of 15 submission attempts for each question. This limit is in place to prevent excessive use of the automarker. You are encouraged to thoroughly test your program before submitting it to the automarker.
- A small sample input along with its corresponding output has been made available on Canvas. If you have developed additional test cases, you are welcome to share them on Piazza.
- The final submission before the assignment deadline will be the one that is evaluated and marked.
- Please do not share your code with other students.

1 Reverse of a Digraph [20 marks]

Question. For a given set of digraphs, please write a program that prints out the reverse of each digraph.
Input. A Digraph input format (Please refer to "Digraph input format").
Output. Given the input, please report the reversed Digraph and make sure the output adjacency lists are sorted.
Example Input:

```
4
1 3
2 3
0

3
1 2

1
0
```

Example Output:

```
4
2
0
1
0 1
3

0 2
0
0
```

2 Bipartite in Undirected Graphs [30 marks]

Question. Recall that a simple undirected graph $G = (V, E)$ is considered bipartite if its nodes can be divided into two independent sets $\{V_1\}$ and $\{V_2\}$ such that every edge in the graph connects a node in set $\{V_1\}$ and a node in set $\{V_2\}$. Given a collection of simple undirected graphs, your task is to develop a program that can determine whether each graph is bipartite or not.
Each simple undirected graph with n nodes (i.e., $|V| = n$, follows this structure: each node is numbered between 0 to $n - 1$. For each $v \in V$, there exists zero or multiple undirected edges connecting node v to $\{S\} \subset V$. Each graph has the following attributes: no self-loops, no parallel edges, and it may potentially be disconnected.
Input. A simple undirected input format (Please refer to "Simple Undirected Input Format").
Output. Given the input graph, please report a one (1) if and only if it is bipartite, and zero (0) otherwise.
Example Input: Each line is a sequence of at least 3 comma separated integers, where the first integer is m, the second is p and the remaining integers are keys.

```
4
1 2 3
0 2
0 1 3
0 2
4
1 3
0 2
1 3
0 2
0
```

Example Output: For each line of input, the output should have a line with the four values described above, in that order, separated by commas.

```
0
1
```

3 Cycles Detection in a Digraph [30 marks]

Question. For a given set of digraphs, please write a program to determine whether the digraph contains a cycle or not.
Input. A Digraph input format (Please refer to "Digraph input format").
Output. For each input digraph, print out a line with a one (**1**) if the digraph contains a cycle and a zero (**0**) otherwise.
Example Input:

```
4
1 3
2 3
0

3
1 2

1
0
```

Example Output:

```
1
0
```

4 Graph Input Format

4.1 Digraph

A sequence of one or more digraphs is taken from the keyboard (System.in). Each graph is represented by an adjacency list. The first line is an integer n, indicating the order of the graph. This is followed by n white space separated lists of adjacencies for nodes labeled 0 to $n-1$. The example input below shows two digraphs. The input will be terminated by a line consisting of one zero (0), which should not be processed.

- The first has node set $\{0,1,2,3\}$ and arc set $\{(0,1),(0,3),(1,2),(1,3),(2,0)\}$.
- The second has node set $\{0,1,2\}$ and arc set $\{(0,1),(0,2),(2,1)\}$.

```
4
1 3
2 3
0

3
1 2

1
0
```

4.2 Simple Undirected Graph

A sequence of one or more digraphs is taken from the keyboard (System.in). Each graph is represented by an adjacency list. The first line is an integer n, indicating the order of the graph. This is followed by n white space separated lists of adjacencies for nodes labeled 0 to $n-1$. The example input below shows two simple undirected graphs. The input will be terminated by a line consisting of one zero (0), which should not be processed.

- The first has node set $\{0,1,2,3\}$ and edge set $\{(0,1),(0,2),(0,3),(1,0),(1,2),(2,0),(2,1),(2,3),(3,0),(3,2)\}$.
- The second has node set $\{0,1,2,3\}$ and edge set $\{(0,1),(0,3),(1,0),(1,2),(2,1),(2,3),(3,0),(3,2)\}$.

```
4
1 2 3
0 2
0 1 3
0 2
4
1 3
0 2
1 3
0 2
0
```