Lab 10: Topological Sort

In this lab, we will implement an algorithm for topological sorting. When a graph structure (i.e. a set of nodes and edges) is given, your program prints a list of nodes as a result of topological sort. As we have discussed in class, topological sorting needs queue ADT in order to save the nodes that do not have any in-degree during the sorting process.

1. Input and Output

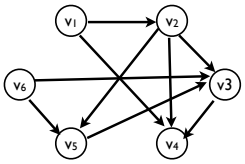
Read a set of vertices in the first line and a set of edges in the second line from the given input file. Each line is described below. You may assume that the node is represented by an integer.

- Vertices are given in the first line. Each vertex is separated by a space.
- Edges are given in the second line. Each edge is represented by a pair of vertices. For example, "1-3" represents an edge from the vertex 1 to 3.

An exemplary input file is given below; the corresponding graph is provided on the right.

Input.txt

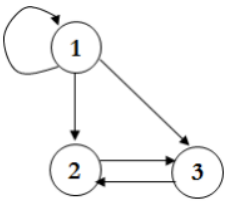
```
1 2 3 4 5 6
1-2 1-4 2-5 2-4- 2-3 3-4 5-3 6-3 6-5
```



Expected output (should be printed in standard output):
1 6 2 5 3 4

2. Data Structure for Topological Sorting

You can use an adjacency matrix to store your graph information as we have discussed in class. An example is shown below.



	1	2	3
1	1	1	1
2	0	0	1
3	0	1	0

3. Program Description

- name : p12.c
- input : an input file name is given as a command line argument. See the example in "1. input"
- output : the corresponding result in the standard output

Submit to the course website (https://portal.hanyang.ac.kr) your source code and a written report. Your report should include the description of your own implementation.