CST 370 – Spring 2022
Homework 10
Due: 04/22/2022 (Friday) (11:55 PM)

How to turn in: Develop **two programs** in **either C++ or Java** on replit.com and then **submit the source programs on Canvas** before the due.

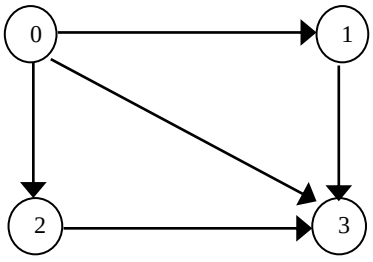
- You **can submit** your programs **multiple times** before the due. However, the **last submission will be used for grading**.
- Due time is 11:55(PM). Since there could be a long delay between your computer and Canvas, you should submit it early.

1. Write a C++ (or Java) program for **hw10_1** that conducts the topological sorting using the Kahn algorithm covered in the class.

Input format: This is a sample input from a user.

```
4
5
0 1
0 2
0 3
1 3
2 3
```

The first line (= 4 in the example) indicates that there are four vertices in the graph. For the homework, you can assume that the first vertex starts from the number 0. The second line (= 5 in the example) represents the number of edges in the graph, and following five lines are the edges. This is the graph with the input data. For this homework, you can **assume that the input graph is always a DAG (= directed acyclic graph)**. Also, you can assume that the **number of vertices** of an input graph is **less than or equal to 25**.



Sample Run 0: Assume that the user typed the following lines

```
4
5
0 1
0 2
0 3
```

```
1 3
2 3
```

This is the correct output. Your program should display the numbers of incoming degrees of each vertex first. For example, the vertex 3 has three incoming degrees which is represented as “In-degree[3]:3”. After the incoming degree information, your program should display the topological order as you learned in the class.

```
In-degree[0]:0
In-degree[1]:1
In-degree[2]:1
In-degree[3]:3
Order:0->1->2->3
```

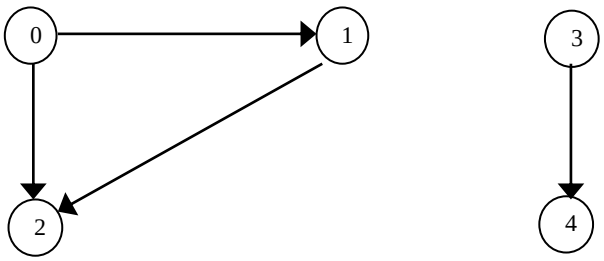
Sample Run 1: Assume that the user typed the following lines

```
5
4
0 1
1 2
0 2
3 4
```

This is the correct output.

```
In-degree[0]:0
In-degree[1]:1
In-degree[2]:2
In-degree[3]:0
In-degree[4]:1
Order:0->3->1->4->2
```

This is the input graph.



Sample Run 2: Assume that the user typed the following lines

```
8
9
0 1
0 4
0 5
```

```
1 5
1 6
6 2
2 3
3 7
6 7
```

This is the correct output.

```
In-degree[0]:0
In-degree[1]:1
In-degree[2]:1
In-degree[3]:1
In-degree[4]:1
In-degree[5]:2
In-degree[6]:1
In-degree[7]:2
Order:0->1->4->5->6->2->3->7
```

[Reference]: https://www.geeksforgeeks.org/topological-sorting-indegree-based-solution/ But this program doesn't work as you learned in the class.

2. Write a C++ program (or Java) for **hw10_2** that displays the alphabetical order of characters for an alien language.

Input format: This is a sample input from a user.

```
3
caa
aaa
aab
```

The first line (= 3 in the example) indicates that there are three words in the following lines which are "caa", "aaa", and "aab". Note that the three words presented in the input are "sorted" order in the alien language. As you can see, the sorted sequence of words is different from our English.
For the program, you should remember that the alphabetical order of characters in the alien language is different from our English. Your program should determine the correct order of characters of the language. For the homework, you can **assume the alien alphabet consists of 26 or fewer characters from a to z**.

Sample Run 0: Assume that the user typed the following lines

```
3
caa
aaa
aab
```

This is the correct output. Since the word "caa" comes before "aaa" and "aab", you know that the character 'c' should come before the character 'a'. Similarly, you know that the character 'a' precedes 'b' based on the sequence "aaa" and "aab".

```
c->a->b
```

Sample Run 1: Assume that the user typed the following lines

```
9
caa
eeb
eef
aaa
aab
deed
deeb
def
daad
```

From the input, you know that 'c' should come before the character 'e'. Similarly, you know the following relationships among the letters: 'c' → 'a', 'c' → d', 'e' → 'a', 'e' → 'd', 'b' → 'f', 'a' → 'd', 'a' → b', 'd' → 'b', 'e' → 'f'. Based on the relationships, this is the correct output. For the homework, you can **assume the input is always valid. In other words, you will have the correct order for the input.**

```
c->e->a->d->b->f
```

Sample Run 2: Assume that the user typed the following lines. From the input, you know the following relationships among the letters: 'a' → 'b', 'a' → d', 'a' → 'c', 'b' → 'd', 'b' → 'c', 'd' → 'c'.

```
5
aab
bba
ddd
cca
ccc
```

This is the correct output.

```
a->b->d->c
```

Hint: **Construct a direct graph** using all characters in the words. Then, use a **topological sorting** to identify the sequence of the characters.