

Instructions. You will have 50 minutes to create and submit one program and a readme file. Download the project zip file, which includes all the required files, from the **Exams** page.

Resources. You may use your book, your notes, your code from programming assignments and precepts,, the code on the COS 126 course website, the booksite, and you may read Piazza. No form of communication is permitted (e.g., talking, texting, etc.) during the exam, except with course staff.

Grading. Your program will be graded on correctness, design, efficiency, and comments. You will receive partial credit for a program that correctly implements some of the required functionality.

Submissions. Submit your work using the link on the **Exams** page.

Discussing this exam. Due to travel for extracurriculars and sports, some of your peers will take this exam next week. Do not discuss exam contents with anyone who has not taken the exam.

This paper. In addition to submitting your code electronically, you *must* return this paper. Fill in the information below, then transcribe and sign the Honor Code pledge. You may do so now.

NAME: _____

NETID: _____

PRECEPT: _____

EXAM ROOM: _____

"I pledge my honor that I will not violate the Honor Code during this examination."

SIGNATURE: _____

Overview. Write a program, `Links.java`, that prints a sequence of numbers, starting with 1, ending with n , where the order is determined by a set of links between the numbers.

Input. The input consists of an integer, n , and a series of one-directional links, $a \rightarrow b$, where a and b are integers in the range 1 to n , inclusive.

Output. First, print 1. Then, print the number that 1 links to. Let's call that x (i.e., $1 \rightarrow x$). Then, print the number y that x links to (i.e., $x \rightarrow y$). Then, print the integer that y links to, and so on. Continue until you reach n or until there is no link to follow. If you reach n , print `done`. Otherwise, print `stuck`.

Assumption. Assume that there is no number that has more than one outgoing link. For example, assume that the links, $2 \rightarrow 3$ and $2 \rightarrow 4$ could not be in the same input file.

Exceptions. Your program must print `loop` if following a link would cause a number to repeat.

Examples. Here are a few examples. Your output must match this exactly.

links1.txt

```
4
1 -> 2
2 -> 3
3 -> 4
```

```
% java-introcs Links < links1.txt
1, 2, 3, 4, done
```

links2.txt

```
226
126 -> 226
1 -> 126
226 -> 217
```

```
% java-introcs Links < links2.txt
1, 126, 226, done
```

links3.txt

```
10
1 -> 5
5 -> 2
```

```
% java-introcs Links < links3.txt
1, 5, 2, stuck
```

links4.txt

```
4
1 -> 2
2 -> 1
```

```
% java-introcs Links < links4.txt
1, 2, loop
```

Answer the questions in the `readme` about how you would create a program that prints the shortest sequence from 1 to n if you couldn't assume that each number had at most one outgoing link.