Instructions. This exam has one question with five parts. You have 50 minutes.

Resources. This exam is open notes, open book, and open to the web—but you may only access the course website, booksite, codePost, and Piazza. You may not use Google or any other search engine. You may not communicate with anyone during this exam (e.g., talking, email, etc.).

Submit. When you are done, submit your program using the submit link on the Meetings page.

Grading. Your program will be graded on correctness, design, efficiency, and clarity including comments. You will receive partial credit for a program that correctly implements some of the required functionality. You will lose a large number of points if your program does not compile.

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

This paper. Fill in the information below, then transcribe and sign the Honor Code pledge.

NAME: _____

NETID: _____

PRECEPT: _____

EXAM ROOM: _____

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

_____

_____

SIGNATURE: _____

Overview. Remember what it was like to program in TOY? If you wanted to insert a new instruction into your existing code, you had to renumber the lines below it and make sure all your jumps still went to the right instructions. What a pain! Instead, let's write a Java program to do it for us automatically.

Input & Output. Create a Java program that takes a TOY program as input and produces a TOY program as output. Let's simplify TOY for the purposes of this exam; we'll assume every line in the input is in the format, "XX: XXXX". No pseudocode, no comments, no empty lines.

Requirements. Your program must have exactly one instance variable: a symbol table whose keys are memory locations and whose values are instructions. Implement the following methods:

```
// adds each line to the ST, where a line is a String in the format: "XX: XXXX"
public ToyStory(String[] lines)

// return a String containing the program in the form: "XX: XXXX\nXX: XXXX\n"
public String toString()

// returns true if there is an instruction defined at memory location 'mem'
public boolean isDefined(String mem)

// returns true if instruction at 'mem' has op code C or D and ends in 'addr'
public boolean containsJumpTo(String mem, String addr)

// finds all instructions that jump to 'addr' and changes them to jump to 'addr'+1
public void incrementAllJumpsTo(String addr)

// EXTRA CREDIT -- MORE DETAILS IN THE TEMPLATE
public void renumberAfter(String min)
```

Strings. As a reminder, to get the first two characters of a String s, call s.substring(0, 2). To check if two Strings, a and b, have the same content, use a.equals(b), not a == b.

Examples. Examples of how to use each method above are provided in the main() method of the template file. Please read the template file in its entirety before starting this exam!

Special cases. Assume that the client of your code will always provide properly formatted TOY programs as input and will not call any of your methods with any input that could cause strange behavior (e.g., incrementing past "FF", loading/storing memory addresses, etc.).