

Spring 2026 COP 3330 Program #7: Use of Java API (StringTokenizer, ArrayList, HashSet, HashMap)

Part A: Preliminary Elections

In many elections, there is a preliminary round, followed by a run-off election. For this assignment we'll simulate the mock preliminary round. Votes will be cast at several precincts and there will be a total number of votes needed to make it to the run-off election. You'll write a program that reads in all of this vote data and outputs a sorted list of the candidates who made it to the run-off election.

The style of this question will be similar to Program #6, where input was read from standard input and output was produced to standard output. Do NOT prompt the user to enter anything and read in the input precisely in the format prescribed.

Input Format (from standard input)

The first line of input will contain two positive integers: n ($n \leq 1000$), and m ($m \leq 10000$), representing the number of precincts in the state and the minimum vote threshold to make it to the run-off election.

The next n lines will contain the voting information, with each line corresponding to the votes made in each precinct.

Each of these n lines will contain 1 or more white space separated Strings of lowercase letters representing each vote received from that precinct. Each of these strings will have in between 1 and 10 letters.

The total number of votes over all precincts won't exceed 1,000,000.

Output Format

Output a sorted list of names of each candidate who has made it to the cut-off election. The list should be outputted with one name per line.

Sample Input

```
5 3
james james james bob aryna bob aryna
aryna sammy aryna sammy sammy james james james james mike
jen jason James lexi jen lexi
joey
sapana aryna lexi
```

Sample Output

```
aryna
james
lexi
sammy
```

Implementation requirements (for part A)

1. You can solve the problem all in a single main method in a class called MakeRunOff.java.
2. You must only read input by lines and use a StringTokenizer object.
3. You must use a HashMap.
4. You must use an ArrayList.
5. You must call Collections.sort.

Part B: Radio Version

The radio version of some songs are "cleaner" than the original version because some words are banned on the radio. For this program, you will make songs safe for the radio!

Your program will read in a list of banned words and their approved substitute words. Then, your program will read in several songs' original lyrics (which might have some banned words). Your program should output each of the song lyrics, substituting each banned word with the approved alternative.

Input Format (from standard input)

The first line of input will contain two positive integers: n ($n \leq 1000$), and s ($s \leq 20$), representing the number of banned words and the number of songs to "make safe for radio", respectively.

The next n lines will each contain b and a , separated by a space, where b represents the banned word and a represents the corresponding approved substitute. Each of these strings will have in between 1 and 10 lowercase letters.

The lyrics to each song follow, with the full lyrics for a single song on a single line.

Output Format

Output each song's radio version on a single line.

Sample Input

```
6 3
tennessee mississippi
you u
shitake stuff
stepped rocked
drugs stuff
aint havent
just when i thought i got him to fall in love with tennessee
looks like were making up for lost time need you to spell it out for me
you stepped inside with a vibe i aint never seen
```

Sample Output

```
just when i thought i got him to fall in love with mississippi
looks like were making up for lost time need u to spell it out for me
u rocked inside with a vibe i havent never seen
```

Implementation requirements (for part B)

1. You must only read input by lines and use a StringTokenizer object.
2. You must use a HashMap.

Deliverables

Please submit two source files:

1. **MakeRunOff.java** which contains your solution to Part A.
2. **Radio.java** which contains your solution to Part B.

Make sure to include a header comment for each file you submit and appropriate internal comments.