COS 126	General Computer Science	Fall 2016

Programming Exam 1

Instructions. This exam has one question. You have 50 minutes. The exam is *open course materials*, which includes the course textbook, the companion booksite, the course website, your course notes, and code you wrote for the course. Accessing other information or communicating with a non-staff member (such as via email, instant messenger, text message, Facebook, Piazza, phone, or Snapchat) is prohibited.

Submission. Submit your solution electronically, via the link on the *Class Meetings* page. Be sure to click the *Check All Submitted Files* button to verify your submission.

Grading. Your program will be graded for correctness, clarity (including comments), design, and efficiency. You will receive partial credit for a program that correctly implements some of the required functionality. You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed input/output specifications.

Discussing this exam. Discussing or communicating the contents of this exam before solutions have been posted is a violation of the Honor Code.

This exam. You must turn in this exam. Print your name, NetID, precept, and the room in which you are taking the exam in the space below. Also, write and sign the Honor Code pledge. You may fill in this information now.

Name:

NetID:

Precept:

Room:

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

Signature

Problem. Write a program `Election.java` that reads election data and prints the number of electoral votes won by each candidate, along with the minimum number of electoral votes needed to win the election.

Assume that an election has only two candidates and is decided according to the following rules:[1]

- The overall election is composed of several region-wide elections.
- Each region is allocated an integer number of *electoral votes*.
- The candidate with the most votes in a region wins all of that region's electoral votes. In the event of a tie, neither candidate wins any electoral votes for that region.
- A candidate wins the overall election if the candidate wins a *strict majority* (more than 50%) of all allocated electoral votes.

Input specification. The input from standard input consists of a sequence of lines.

- The first line contains two strings (the names of the two candidates).
- Each of the remaining lines contains a string (the name of the region) and three integers (the number of electoral votes allocated to the region, the number of votes for candidate 1, and the number of votes for candidate 2), separated by whitespace.

Here are two examples:

```
% more 2012.txt
Romney Obama
Alabama          9  1255925   795696
Alaska           3   164676   122640
Arizona         11  1233654  1025232
Arkansas         6   647744   394409
California      55  4839958  7854285
   ⋮
West_Virginia    5   417655   238269
Wisconsin       10  1407966  1620985
Wyoming          3   170962    69286
```

two candidates

data for one region

electoral votes

Romney votes

Obama votes

Romney wins West Virginia (5 electoral votes)

```
% more hogwarts.txt
Snape McGonagall
Gryffindor  17 201 623
Hufflepuff  14 400 400
Ravenclaw   11 341 205
Slytherin   15 865   0
```

McGonagall wins 17 electoral votes

Snape wins 26 electoral votes

[1] These rules are similar to those used in U.S. presidential elections, except that there can be more than two candidates, some states are not winner-take-all, and the rules for breaking ties vary by state.

Output specification. The output to standard output consists of three lines. The first two lines are the names of the two candidates, followed by the number of electoral votes won by each candidate; the third line is the minimum number of electoral votes needed to win the election.[2]

```
% java-introcs Election < 2012.txt
Romney 206
Obama 332
270 needed to win
```

```
% java-introcs Election < hogwarts.txt
Snape 26
McGonagall 17
29 needed to win
```

Do *not* print any other output on standard output.

API specification. Your program `Election.java` must be organized as two public functions, with the following API:

`public class Election`	
`public static int majorityOf(int n)`	*smallest integer strictly greater than* $\frac{1}{2}n$ (*n will be a positive integer*)
`public static void main(String[] args)`	*read election data from standard input; print results to standard output*

Restrictions. *Do not use arrays on this exam.*

Test files. The data files `2012.txt` and `hogwarts.txt` are available via the *Class Meetings* page.

Submission. Submit the single file `Election.java` using the link on the *Class Meetings* page.

[2] In the 2012 U.S. presidential election, Governor Romney won 206 electoral votes (including 9 from Alabama and 3 from Alaska) and President Obama won 332 electoral votes (including 55 from California and 10 from Wisconsin). Since a total of 538 electoral votes were allocated, the minimum number of electoral votes needed to win the overall election was 270.

In the Hogwarts election, McGonagall won 17 electoral votes (from Gryffindor) and Snape won 26 electoral votes (11 from Ravenclaw and 15 from Slytherin). Since a total of 57 (17 + 14 + 11 + 15) electoral votes were allocated, the minimum number of electoral votes needed to win the overall election was 29.