### **Building Java Programs, Chapter 6 Lab Handout**

### **Token-Based File Processing**

1. Write a static method named evenNumbers that accepts a string of text as a parameter. Assume that the text is a series of integers, and process this text and report various statistics about the integers. Report the total number of numbers, the sum of the numbers, the total count of even numbers and the percent of even numbers. For example, if the string is the following:

"5 7 2 8 9 10 12 98 7 14 20 22"

Your method should produce the following output.

Total numbers = 12

Sum of numbers = 214

Total evens = 8

Percent evens = 66.66666666666667

If a non-integer element is encountered, it should be ignored & not cause the program to crash. For example: "5 7 2 8 9 10 oops 12 98 7 14 20 22" will product output identical to the original example above.

\*\*\*\*\*\* The questions ask to pass a string to my methods, but I constructed a scanner in main that used the string I would have passed to the method and passed that instead, sorry! \*\*\*\*\*



2. Write a static method named negativeSum that accepts a string of text as a parameter. Assume that the text is a series of integers, and determine whether or not the cumulative sum starting from the first number is ever negative. The method should produce a message indicating whether or not a negative sum is possible and it should return true if a negative sum can be reached and false if it can't be reached. For example, if the string contains the following text,

"38 4 19 -27 -15 -3 4 19 38"

your method will consider the sum of just one number (38), the sum of the first two numbers (38 + 4), the sum of the first three numbers (38 + 4 + 19), and so on up to the sum of all of the numbers. None of these sums is negative, so the method would produce the following message:

no negative sum

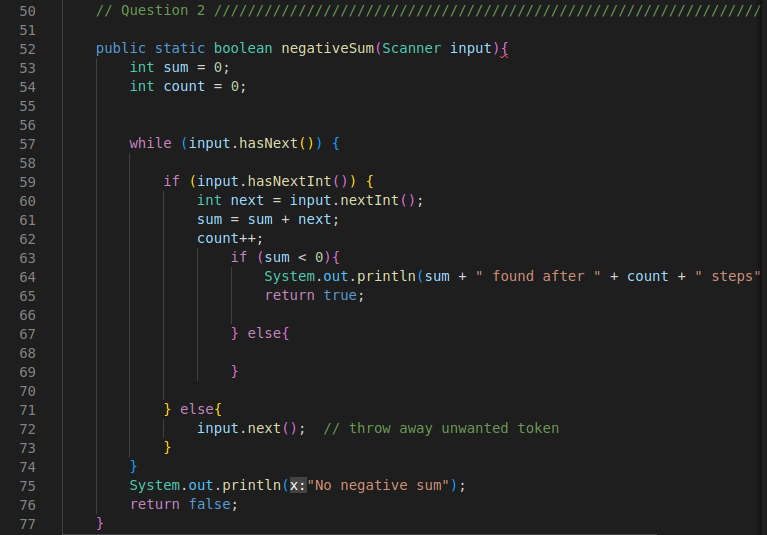
and would return false. If the text is the following,

"14 7 -10 9 -18 -10 17 42 98"

the method finds that a negative sum is reached after adding 6 numbers together  
(14 + 7 + -10 + 9 + -18 + -10) and that the sum is -8. It should report the following:

-8 after 6 steps

and should return true, indicating that a negative sum can be reached. The method should return the first negative sum it encounters. **Also:** As in question #1, if non-integer element is encountered, it should be ignored. It should not cause the program to crash.



3. Write a static method named boyGirl that takes a string of text as a parameter. Assume that the text represents a series of names followed by integers, and that the names alternate between boys' names and girls names. Your method should compute the sum of the boys' integers and the sum of the girls' integers separately and print them. The line could end with either a boy or girl; you may not assume that it contains an even number of names. For example, if the String contains the following text,

"JP 3 Helene 7 Jordan 14 Iva 13 Sergey 4 Marianne 9 Kenneth 6"

then your method should produce the following output:

4 boys, 3 girls

Boys sum = 26

Girls sum = 29



### **Line-based Processing**

4. Write a static method named textCount that takes a Scanner representing a file as a parameter and that reports various statistics about the file. In particular, your method should report the number of lines in the file, the total number of characters (not counting any new-line \n characters) in the file, and the length and text of the longest line. You may assume that the input file has at least one line of input.

For example, if the file contains the following text:

Twas brillig and the slithy toves

did gyre and gimble in the wabe.

All mimsey were the borogroves,

and the mome raths outgrabe.

"Beware the Jabberwock, my son,

the jaws that bite, the claws that catch,

Beware the JubJub bird and shun

the frumious bandersnatch."

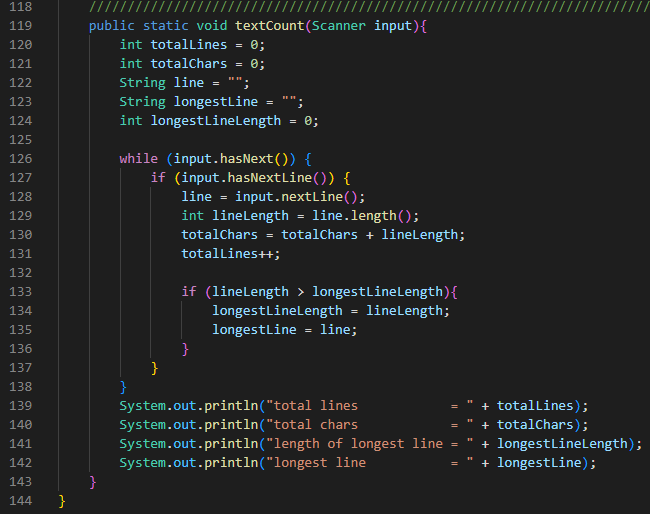
The program would find that there are 9 lines (blank lines count), a total of 254 characters in the file (33 on the first line, 32 on the next line, 31 on the next line, 28 on the next, 0 on the next, etc) and that the longest line is the one that begins with "the jaws that bite". The method would produce the following output:

Total lines = 9

Total chars = 254

Length of longest line = 41

Longest line = the jaws that bite, the claws that catch,



### **Line-and-Token Processing**

5. Write a static method named fixSpacing that accepts a Scanner representing a file as a parameter and writes that file's text to the console, with multiple spaces or tabs reduced to single spaces between words that appear on the same line.

For example, if the input file contains the following text,

four score and

seven years ago our

fathers brought forth

on this continent

a new

nation

then your method should produce the following output:

four score and

seven years ago our

fathers brought forth

on this continent

a new

nation

Notice that some lines might be blank. Each word is to appear on the same line in the output on which it appears in the input file.

