Assignment overview

For this assignment you will complete a Java program that takes as input a list of (one-word) names and calculates the "best" names according to a couple of curious formulas. Some of the code is provided for you. You need to fill in some parts.

Submission and marking

Due date: As shown on Learning Hub. Your last submission will be the one that counts. Late assignments will not be graded.

Submit:

- Your Java source code
- File name is not important, but as a matter of style, at least rename it from "starter_code.java" (the file I'm giving you)

Marking: This lab is worth 15 marks.

Provided for you

1. The file starter_code.java containing some starter code:

- `read_input_file()` – a function that will read the input file for you and create/return an array for you to process.
- Code stubs for two functions that you will need to write.
- Main program that needs to be completed.

2. Several sample files that you can use for testing (see **Sample output**).

Sample input

An input file consists of some number of one-word names in all capital letters. The names are all on one line, separated by commas (no space). For example, here is the contents of one sample input file:

```
MARY,PATRICIA,LINDA,BARBARA,ELIZABETH,JENNIFER,MARIA,ROGER,JONATHAN,
RALPH,NICHOLAS,BENJAMIN,BRUCE,HARRY
```

Note: This section is informational only. You do not need to write code to read this input file. There is a function in the given Java code that will read the above-formatted input file and return an array of Strings (the names).

Part 1 (3 marks)

Complete the function `alpha_score()`. The function's one argument is a name (String). The return value (int) is the *alpha_score* for the name. The alpha_score for a name is calculated by associating a value with each letter: A=1, B=2, C=3, etc., and then adding the values together.

For example, the name JUSTIN has an alpha_score of 10+21+19+20+9+14 = 93.

The function *must not* perform any input or output except the arguments and return value.

Do not change the signature of the function (the number and type of the arguments and return value).

You do not need to verify/validate the input to this function (or the one in Part 2). You may assume that the name argument contains nothing but uppercase letters. However, if it offends your sensibilities as a programmer to ignore this risk, then (a) good for you and (b) go ahead and have your function return a value of -1 if the name contains any invalid character(s).

Part 2 (3 marks)

Complete the function `zeta_score()`. This function is similar to `alpha_score()` except it returns the *product* of the letter values.

The zeta_score for JUSTIN is 10*21*19*20*9*14 = 10054800.

NOTE: this function returns a long value. Your sample data files do not contain any names whose zeta_score will exceed a regular int, but the data files used in marking just might!

Part 3 (5 marks)

Write code in main() that finds the name with the highest alpha_score, and the name with the highest zeta_score, in the input file. These might not be the same name. Your program should display the results as follows (for whatever input file is being used at the time, of course):

```
Input file is 5_names.txt
5 names processed
Highest alpha_score is ZOEY with 71
Highest zeta_score is MINNIE with 1031940
```

You may include other interesting or helpful information in the output if you wish. For example, while you are writing/testing your code it may be useful to display the scores for every name in the list. However, when you submit your solution for marking, please *do not* display the entire list of names. (Some of the data files are very large.)

Part 4 (4 marks)

You must use good programming style throughout your code. The following is the *bare minimum of required style*:

- Header comments with at least the following information:
  - Your name, ID, and set
  - A short description of the program (a sentence is enough)
- A descriptive comment for any significant block of code such as a function or an important loop
- Comments for any other code that is “unusual” or otherwise interesting
- Meaningful variable names for important variables

You may use Javadoc-style comments or not, as you wish, as long as they cover the basics above.

Tips

A clever trick for converting a character to an integer value in Java: Do a little arithmetic on the character. For example, the following statement:

```
System.out.println('b'+1);
```

... will print the value 99, which is the integer ASCII value of 'c'.

Sample output

For information purposes, below are the correct answers (best names) for all of the included sample data files. NOTE: Your program does not need to produce all of this information in one execution (although it's fine if it does). The only requirement is to read and display the output for one input file.

```
Input file is 5_names.txt
5 names processed
Best alpha_score is ZOEY with 71
Best zeta_score is MINNIE with 1031940

Input file is 14_names.txt
14 names processed
Best alpha_score is ELIZABETH with 88
Best zeta_score is JENNIFER with 47628000

Input file is 37_names.txt
37 names processed
Best alpha_score is AUGUSTUS with 129
Best zeta_score is AUGUSTUS with 468050940

Input file is 300_names.txt
333 names processed
Best alpha_score is LIZZETTE with 123
Best zeta_score is DEMETRIUS with 1680588000

Input file is 1000_names.txt
1036 names processed
Best alpha_score is CHRISTOPHER with 139
Best zeta_score is CHRISTOPHER with 255301632000

Input file is 5000_names.txt
5163 names processed
Best alpha_score is SYLVESTER with 145
Best zeta_score is CHRISTOPHER with 255301632000
```