



COMPUTER SCIENCE 12B (FALL, 2020) ADVANCED PROGRAMMING TECHNIQUES IN JAVA

PROGRAMMING ASSIGNMENT 4

Program Description:

This assignment will test your understanding of the use of the `Scanner` objects, `Strings`, file processing, `Arrays`, and everything covered so far.

You should limit yourself to the Java features covered in class so far (lecture 8).

Modularity in your code is very important, YOU MUST USE STATIC METHODS.

Problem 1:

Students are often asked to write term papers containing a certain number of words. Counting words in a long paper is a tedious task, but the computer can help. Write a program that counts the number of words, lines, and total characters (not including whitespace) in a paper, assuming that consecutive words are separated either by spaces or end-of-line characters. Remember to prompt the user for a file to read (you can create your own `.txt` file).

Problem 2:

Write a program that reads a file containing data about the changing popularity of various baby names over time and display the data about a particular name. Each line of the file stores a name followed by integers representing the name's popularity in each decade: 1900, 1910, 1920, and so on. The ranking range from 1 (most popular) to 1000 (least popular), or 0 for a name that was less popular than the 1000th name. The following lines are a sample of the file format:

```
Sally 0 0 0 0 0 0 0 0 0 0 886
Sam 58 69 99 131 168 236 278 380 467 408 466
```

Your program should prompt the user for a name and search the file for that name:

This program allows you to search through the data from the Social Security Administration to see how popular a particular name has been since 1900.

Name? **Sam**

If the name is found, the program should display data about the name on the screen:

```
Statistics on name "Sam"
1900: 58
1910: 69
```

NOTE: For this problem, use the provided file `names.txt`

Problem 3:

For this problem you need to use the provided files `matrixA.txt`, `matrixB.txt`, `matrixC.txt`, and `matrixD.txt`. Each file contains the dimensions of the matrix on the first line, with the matrix below.

Write a program that prompts for two of these file names, reads in the matrix from each file and prints the result of their sum and product into a file named `results.txt`. Your program should handle the case where the matrices are not of compatible dimensions for one of the operations by printing an error message on the console window.

How to add two matrices?

Matrix addition is defined element-wise. That is, each element in the resulting matrix is the sum of the elements in the corresponding position in the two matrices being added. Thus, both matrices must be of the same dimension.

For example:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

How to multiply two matrices?

Matrix multiplication is not an element-wise operation. The value at each position (i, j) in the resulting matrix is determined by summing the product of each pair of elements from the i-th row of the first matrix and the j-th column of the second matrix. The example below will hopefully clarify this operation. Also note that the number of columns of the first matrix must equal the number of rows of the second matrix, but the other dimensions do not need to be equal.

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \times \begin{bmatrix} g & h \\ k & l \\ m & n \end{bmatrix} = \begin{bmatrix} a \cdot g + b \cdot k + c \cdot m & a \cdot h + b \cdot l + c \cdot n \\ d \cdot g + e \cdot k + f \cdot m & d \cdot h + e \cdot l + f \cdot n \end{bmatrix}$$

Grading:

You will be graded on

- **External Correctness:** The output of your program should match exactly what is expected. Programs that do not compile will not receive points for external correctness.
- **Internal Correctness:** Your source code should follow the stylistic guidelines linked in LATTE. Also, remember to include the comment header at the beginning of your program.

Submission:

Create a folder containing your Java source code (programs). Compress (zip) the folder and upload it to Latte by the day it is due. For late policy check the syllabus.