Programming Paradigms 2024: Java Coursework

The following files are all provided, they must be downloaded from moodle:
Input.txt // an example file to be read in and analysed
Output.txt // a sample output, given the Input.txt as the input

The provided files, namely 'Input.txt' and 'Output.txt,' are available for download on Moodle. The Java code you write should accomplish the following tasks:

1. Read the content from 'Input.txt' and identify key words, including 'Iteration,' 'Phase,' and special indicators starting with '-' or '--,' indicating the start and end of certain algorithms.
2. Capture the counts of 'Iteration' and 'Phase,' along with the names of algorithms. Also, record the values of 'Elapsed Time (ms)', 'Evaluations', and ' Improvements '.
3. Output a comma-separated summary file with a header, including columns for 'Iteration,' 'Phase', 'Algorithm', 'Elapsed Time (ms)','Evaluations', and ' Improvements '.
4. Record the sum of the Elapsed Time consumed by all algorithms at the end, using '-1' to denote 'Iteration' and 'Phase', and 'All' to indicate all algorithms.

For instance, given the content of 'Input.txt' provided:

```
Iteration 1 Starts              Time 14: 16:56
Phase 1: Main Search Starts     Time 14: 16:56
- Ejection Algorithm Starts     Time 14: 16:56
obj.     Dist.     Duty(h)     Vehicles     Unassigned     ElapsedTime
*     19164420061.0000     61.0000     3.6000     1     17     14:16:56
- Ejection Algorithm Ends       Time 14: 16:56     Elapsed Time (ms): 46     Evaluations: 3183     Improvements: 0
-- N0_Insert Starts            Time 14: 16:56
obj.     Dist.     Duty(h)     Vehicles     Unassigned     ElapsedTime
-- N0_Insert Ends              Time 14: 16:56     Elapsed Time (ms): 1     Evaluations: 68
- Ejection Algorithm Starts     Time 14: 16:56
obj.     Dist.     Duty(h)     Vehicles     Unassigned     ElapsedTime
- Ejection Algorithm Ends       Time 14: 16:56     Elapsed Time (ms): 40     Evaluations: 3081     Improvements: 0
```

The expected 'Output.txt' should look like this (other examples are available on Moodle):

```
Iteration,Phase,Algorithm,Elapsed Time (ms),Evaluations,Improvements
1,1,Ejection Algorithm,86,6264,0
1,1,N0_Insert,1,68,0
-1,-1,All,87,6332,0
```

Important points to note:

1. The 'Elapsed Time (ms)' for 'Ejection Search' is an accumulated number, which is the sum of 46 and 40, similar to the 'Evaluations'(i.e. 3183+3081) and 'Improvements'.
2. The order of the algorithm list in the output should follow the FIFO (First In, First Out) order. Your code should scan the document from the top to bottom.
3. Ensure your code accommodates a different 'Input.txt' file, with similar content and structure, placed in the same directory as your files.
4. 80% marks will be awarded for correct functionality (i.e. correct output), 10% will be given to Time complexity analysis, 10% will be given to java code judging result. 0% will be given to incorrect output.

Definitions

Standard input

System.in, means that the stream from which input to the program is taken. Typically this is the keyboard, but it can be specified that input is to come from a serial port or a disk file.

Standard output

System.out, means that the stream to which output from the program is sent. Typically this is a display, but it can be redirected to a serial port or a file.

Submission

You must submit a single Java source code file containing all your code for this coursework. This file must be called PGP.java and must not require any other files outside of the standard Java packages which are always available. The file must compile and execute without warnings or errors using the command.

Compile: javac -encoding UTF-8 -sourcepath . PGP.java

Execute: java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m PGP < Input.txt > Output.txt

Technical Notes

This part contains important technical information and it is important that you read and understand all the information below.

You program **MAY** have multiple classes if you wish, but **only in one java file**. And only the class with your main method **SHOULD** be marked as public.

Your program **MUST** read its input from standard input.

Your program **SHOULD** send its output to standard output (by executing above command, it will produce Output.txt in the same directory as PGP.java and Input.txt, so no FileWriter is required).

If your program exits with a non-zero exit code, it will be judged as a run-error.

Program submitted will be run inside a sandbox. The sandbox will allocate 2GB of memory for your program. Your entire program, including its runtime environment, must execute within this memory limit. For Java, the runtime environment includes the interpreter (JVM).

We suggest that you do not use package statements (that is, we suggest that your solution reside in the “default package”).

Please use JDK versions latter than 7.

Possible results

A submission can have the following results:

CORRECT The submission passed all tests: you solved this problem!

0% will be given to errors listed below:

COMPILER-ERROR There was an error when compiling your program. Note that when compilation takes more than 30 seconds, it is aborted and this counts as a compilation error.

TIMELIMIT Your program took longer than the maximum allowed time for this problem, 5 seconds. Therefore it has been aborted. This might indicate that your program hangs in a loop or that your solution is not efficient enough.

RUN-ERROR There was an error during the execution of your program. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g., by indexing arrays out of bounds), trying to use more memory than the limit, reading or writing to files, etc. Also check that your program exits with exit code 0!

NO-OUTPUT Your program did not generate any output. Check that you write to standard out.

OUTPUT-LIMIT Your program generated more output than the allowed limit. The solution is considered incorrect.

WRONG-ANSWER The output of your program was incorrect. This can happen simply because your solution is not correct, but remember that your output must comply exactly with the specifications of the judges. See testing below for more details. The judges may have prepared multiple test files for each problem.

Deadline is 6pm Monday the 8th of April

Any Late Submission will receive 0 marks

Plagiarism is not allowed