Computer Programming Midterm

Checkpoints

1. You should do the midterm in your own. You are not allowed to share code with others and/or copy code from other resources. If you are caught, as in the syllabus, you will get a failing grade.
2. Internet is not allowed except eTL
3. Grading will be done in Linux environment using java 1.8.
4. Program failed to compile/run will result 0.
5. Do not loop your program to repeat unless you are told so.
6. Do not change input/output format unless you are told so.
7. Do not color console.
8. Write your name and student number at top of program as a comment
9. Make sure you take attendance before leave. People who marked absent will not be graded.
10. DO NOT include Korean (and any other language, except English) comments
    - Korean comments will cause compilation errors in Linux environment, which will result in 0 for your grade

Submission

1. Submit your midterm on eTL.
2. Zip your file (or tar) as '<Student ID>-midterm.zip' ex.) 2017-12345-midterm.zip
3. No late submission is allowed

1.) Given a set of points in plane (2-D), we want to output all the pairs of points that are the closest and the farthest. The input points are given in a file, and the output should be written to another file. The names of the input and output files should be passed to your program as arguments to the main method.
e.g.) java Midterm1 input.txt output.txt
Write a Midterm1.java which satisfies following:

- Output all the pairs of points which are the closest, and also all the pairs which are the farthest (when there are two or more farthest or closest pairs, all of them should be output).
  - Distance, d, between two points (x1, y1) and (x2, y2) can be computed using the equation $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- Input format :

  [Number of points]
  [x1] [y1]
  [x2] [y2]
  ..
  e.g.)
  4
  0 0
  1 0
  1 4
  1 5
- Output format:

  The farthest pair['s' in case of multiple pairs] [is/are]:
  ([L1_x], [L1_y]), ([L1'_x], [L1'_y])
  ...
  The closest pair['s'] [is/are]:
  ([s1_x], [s1_y]), ([s1'_x], [s1'_y])
  …
  e.g.)
  The farthest pair is:
  (0, 0), (1, 5)
  The closest pairs are:
  (0, 0), (1, 0)
  (1, 4), (1, 5)

2.) Write a Midterm2.java which satisfies following:

- Given two strings as parameters, report their “longest common substring”.
  - “Substring of a string S is a string S' that occurs "in" S” - Wikipedia
    - ex.)substrings of the string "apple" would be "apple", "appl", "pple", "app", "ppl", "ple", "ap", "pp", "pl", "le", "a", "p", "l", "e", ""
  - Given two strings s1 and s2, their longest common substring(s) is(are) the longest string(s) that is(are) a substring(s) of both the strings.
- If there are multiple longest common substrings, print only one occurrence of the longest common substring that occurs leftmost in the first string (parameter).
  - ex.) java Midterm2 aaabbababaa aababaabbaa
    aabba

3.) Write a Midterm3.java which satisfies following:

- Read an input file and output the number of words, average length of a word, and how many words are above the average length, into an output file.
- File contains only characters from ‘a’-’z’ and ‘A’-’Z’, space, comma, and period. Any sequence of characters (without space, comma or period) is considered as a word.
  - ex.) I like banana, but apple is good too. I hate fish.

    Words: I, like, banana, but, apple, is, good, too, I, hate, fish

- Round the average length to two decimal places.
- Input filename and output filename will be given through parameter.
- Report number of words, average length then number of words above the average length

  - ex.) java Midterm3 input1.txt output1.txt
    [output1.txt]
    Number of words = 11
    Average length of a word = 3.36
    Number of words above the average length = 6

- Follow exactly same format as example. Do not include any extra characters on output.

Java API
java.lang.Math
public static double sqrt(double a)
Returns the correctly rounded positive square root of a double value. Special cases:
If the argument is NaN or less than zero, then the result is NaN.
If the argument is positive infinity, then the result is positive infinity.
If the argument is positive zero or negative zero, then the result is the same as the argument.
Otherwise, the result is the double value closest to the true mathematical square root of the argument value.

Parameters:
a - a value.

Returns:
the positive square root of a. If the argument is NaN or less than zero, the result is NaN.

java.lang.String

public String substring(int beginIndex, int endIndex)

Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.
Examples:
"hamburger".substring(4, 8) returns "urge"
"smiles".substring(1, 5) returns "mile"

Parameters:
beginIndex - the beginning index, inclusive.

endIndex - the ending index, exclusive.

Returns:
the specified substring.

public char[] toCharArray()

Converts this string to a new character array.
Returns:
a newly allocated character array whose length is the length of this string and whose contents are initialized to contain the character sequence represented by this string.

public String replace(char oldChar, char newChar)

Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.

If the character oldChar does not occur in the character sequence represented by this String object, then a reference to this String object is returned. Otherwise, a new String object is created that represents a character sequence identical to the character sequence represented by this String object, except that every occurrence of oldChar is replaced by an occurrence of newChar.

Parameters:

oldChar - the old character.

newChar - the new character.

Returns:

a string derived from this string by replacing every occurrence of oldChar with newChar.