Task 1: Bigrams [50 points]

You are given a plain text file which contains a single line of text (no new line characters).

a) Count the number of appearances of letter bigrams in the text.
b) For a given sequence of letters $x_0x_1x_2 \dots x_n$ and based on bigram frequencies calculated from a) find the most probable continuation consisting of three letters $y_0y_1y_2$.

Input

Input is given through command line arguments in the following order

`inputTextFile`	Path to the input text file
`sequencesTextFile`	Path to text file containing input character sequences (one per line) for which you need to find the continuation
`outputBigramsFile`	Path to output text file which should contain list of bigrams and their count (specified in a))
`outputSequencesFile`	Path to output file that should contain most probable result corresponding to sequences specified in b) (one per line)

Notes:

- Input file will contain only printable ASCII characters (including spaces).
- Input file will not be empty.
- The last character in the file will appear at least twice.
- Input character sequence will not contain characters that do not appear in the input file.

Output

a) Output file with bigrams should contain list of bigrams that appeared in text, and number of their appearances (one pair per line, doesn't have to be sorted), i.e.:

$c_{00}c_{01}\ count_0$

$c_{10}c_{11}\ count_1$

$c_{20}c_{21}\ count_2$

…

$c_{m0}c_{m1}\ count_m$

b) Output file with result sequence should contain the most probable resulting sequence(s), i.e.

$x_0x_1x_2 \dots x_ny_0y_1y_2$

Notes:

- Valid values for c_{ij} and y_k are ASCII characters that appear in text file (including spaces).
- Valid values for $count_i$ are positive integer numbers.
- Output folder(s) should be created if it doesn't exist.

Example

Input file:

```
+yxy +xy xy xxxxxxxy
```

Input sequence file:

```
+y
```

a) Output file with bigram counts should contain (only) these lines:

```
 + 1
 x 2
+x 1
+y 1
xx 6
xy 4
y  3
yx 1
```

b) Output file with expected sequence(s):

```
+y xx
```

Calling

In case you're producing Windows .exe, your program will be called like:

```
Bigrams.exe inputTextFile sequencesTextFile outputBigramsFile
outputSequenceFile
```

In case of Python script:

```
Python bigrams.py inputTextFile sequencesTextFile outputBigramsFile
outputSequenceFile
```

In case you write an Octave function, you should have a `Bigrams.m` file containing

```
function Bigrams (inputTextFile, sequencesTextFile, outputBigramsFile,
outputSequenceFile)
```

which does the processing and saves the results into the output folder as described.

In case of Java:

```
java Bigrams inputTextFile sequencesTextFile outputBigramsFile
outputSequenceFile
```

or

```
java -jar Bigrams.jar inputTextFile sequencesTextFile
outputBigramsFile outputSequenceFile
```

Document update history

5/20/2016 7:04 PM: Document created