

Part I

Webpage:

http://www.dcs.shef.ac.uk/~yg/campus_only/com6470/

Any work must be your own work. Except your own classes, you are allowed to use only

- the System class, the String class, and the Math class from the java.lang package;
- classes provided by the sheffield package;

unless otherwise noted.

1 [10%]

A DNA sequence is made from four types of the nucleic acids (A, C, T and G). Write a program `CountAcids.java` that takes a short sequence of DNA from a command line, and displays the number of each nucleic acid in the sequence. If the sequence has any symbol other than the four valid nucleic acids, your program should exit with a polite warning message. If the sequence is longer than 10 nucleic acids but otherwise valid, your program should exit with a warning, indicating the number of acids in the sequence. A typical outcome of this program may be

```
> java CountAcids ACTTGATG
A: 2, C: 1, T: 3, G: 2

> java CountAcids ACTTAGACTTAACTTGG
This sequence (18 acids) is too long.

> java CountAcids GeorgeWBush
Please enter a valid DNA sequence.
```

2 [10%]

A transposition cipher first chooses a key to number the columns, then write a plaintext in rows, and finally reads the ciphertext by columns, starting with the column whose key letter is the lowest alphabetically. Shown below is one example whereby a string 'jamesbond' is used as the key:

key	j a m e s b o n d	
	(5)(1)(6)(4)(9)(2)(8)(7)(3)	plaintext :
	s e n d o n e	send one million euros to my swiss
	m i l l i o n e	bank account
	u r o s t o m	
	y s w i s s b	ciphertext :
	a n k a c c o u	EIR NTOOTSCG EMBUADLSW
	n t (y e g h i a)	YSMUYANNLOSK E OINNOSCH
	random characters	I IAE

Write a program `TranspositionCipher.java` that reads a key from the keyboard and a plaintext from a file `plain.txt`, then displays the ciphertext. The `plain.txt` may contain multiple lines with lower case letters and spaces. A sample of `plain.txt` is provided on the com6470 webpage, however your code should be able to handle other input files. The ciphertext should be displayed on the screen using upper case letters and spaces. A key is a sequence of lower case letters only, and any letter cannot appear more than once. If there is more than one space left in the final row, they should be filled in with randomly generated lower case letters before encryption.

For further detail of a transposition cipher, you may find an extensive list by searching the web with, e.g., *Google*.

3 [10%]

Choose either {3.1 and 3.2} or {3.1 and 3.3}.

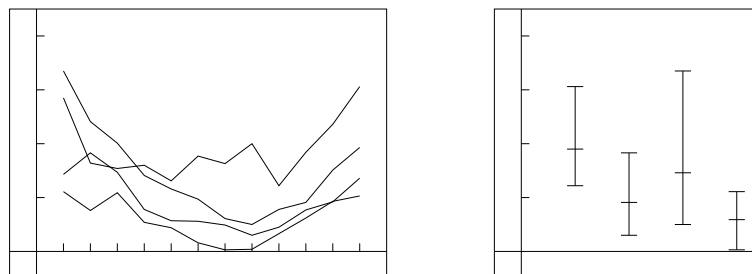
3.1

Write a program `PrimeNumbers.java` that reads a positive integer n from the keyboard, then displays a complete list of prime numbers between 1 and n in the ascent order. It should also show how many prime numbers exist between 1 and n .

3.2

Files `city1.txt`, `city2.txt`, `city3.txt`, and `city4.txt`, provided on the com6470 webpage, contain monthly rainfalls (from January to December) of four cities. Write a program `RainFalls.java` that reads monthly data from files, and draws a figure that shows rainfalls of these cities. The program should not produce more than one graphics window.

Two, not very creative, examples are shown below. The graph on the left simply plots monthly rainfalls of four cities. The graph on the right shows the maximum, the average, and the minimum rainfalls for each city.



3.3

Two sinusoidal voltages v_1 and v_2 are fed to the x and y plates of an oscilloscope:

$$v_1 = 5 \sin(10t)$$

$$v_2 = 2 \sin\left(5t + \frac{\pi}{6}\right)$$

Write a program `Lissajou.java` that draws the resulting Lissajou figure.

Try a web search if you don't know (or have forgotten) about the Lissajou figure.

4 [10%]

In this task, you may use the `Hashtable` class from the `java.util` package, if needed.

Write a program `VocabularyList.java` that reads plain text from a specified input file, counts the number of occurrences for each of different words, and outputs a vocabulary list into a file `vocabulary.txt`.

An input file contains an English text of multiple lines and arbitrary length, consisting of words separated by spaces and/or new lines. For simplicity, all words are spelled using lower case letters only. Digits are replaced with spelled words (e.g., 'year nineteen ninety' instead of 'year 1990', or 'two point five million pounds' instead of '£2.5M'). All of punctuation are removed except *apostrophe* (`'`). You may use the file `news.txt` from the lab exercises for testing your work, however your code should be able to handle other input files.

The vocabulary list is in the two column format — words in the first column and their frequencies in the second. It is first sorted in the frequency order, i.e., words that occurred more frequently appear at the higher rank. Words with the same frequency should be ordered alphabetically.