UNIVERSITY OF DUBLIN

TRINITY COLLEGE

Faculty of Engineering and Systems Sciences

Department of Computer Science

B.A.(Mod.) Computer Science
Junior Freshman Examination

Trinity Term 2000

1BA2 - Introduction to Programming

Thursday 8th June

Luce Hall

14.00-17.00

Dr. Vinny Cahill

Attempt 5 questions only.

1) A sequence of numbers $X_0, X_1, X_2, X_3, \ldots$ is generated by the following formula:

$$X_{n+1} = X_n + 3 * (X_{n-1} - X_{n-2})$$

where $n \geq 2$ and
$X_0 = 0$
$X_1 = 1$
$X_2 = 3$

Write a Java application that calculates the largest number in this sequence that is less than some number entered by the user (e.g., if the number entered by the user were 4, then the application should produce 3).

Modify your application so that it produces the largest number in the sequence that is less than LONG.MAX_VALUE and yet does not produce a run-time error.

2) Design a class called `Rational` whose instances represent rational numbers, i.e., numbers of the form `X/Y` where `X` (the numerator) and `Y` (the denominator) are integers. The numerator and denominator must be represented explicitly in the class.

Your class should provide at least the following methods:

- an appropriate constructor;
- a method to return the numerator of the current rational number;
- a method to return the denominator of the current rational number;
- a method to return the value of the current rational number as a real number;
- a method to add the current rational number to another rational number to produce a new rational number;
- a method to multiply another rational number by the current rational number to produce a new rational number;
- a `static` method that takes a rational number as a parameter, reduces it to its lowest terms, and returns the resulting rational number. For example, if the given rational number has numerator 6 and denominator 8, the rational number returned should have numerator 3 and denominator 4.

Hint: find the largest number that is a divisor of both the numerator and denominator and divide both by this number.

3) Design a Java class whose instances represent sets of integer numbers and that provides the following methods:

- an appropriate constructor;
- a method to add a number to the current set (if it is not already a member);
- a method to remove a number from the set (if it is a member);
- a method to compute the union of the set with another set (i.e., the set of numbers that are members of either set);
- a method to compute the intersection of the set with another set (i.e., the set of numbers that are members of both sets);
- a method to compute the difference of the set with another set (i.e., the set of numbers that are members of the current set but not also members of the other set).
-

4) Design a Java application that reads a chemical formula describing some compound and prints out the molecular weight of the compound. Some examples of chemical formulae are as follows:

H20
C6H1206

where a number following the symbol for an element denotes the number of instances of that element in the compound. The molecular weight of a compound is the sum of the atomic weights of each of the atoms making up the compound.

Your program need only work for the first 10 elements of the periodic table whose atomic weights are as follows:

Element	Symbol	Atomic weight
Hydrogen	H	1.008
Helium	He	4.003
Lithium	Li	6.94
Beryllium	Be	9.013
Boron	B	10.82
Carbon	C	12.01
Nitrogen	N	14.008
Oxygen	O	16.0
Fluorine	F	19.0
Neon	Ne	20.183

Comment on how easily your application can be extended to work for the whole periodic table.

N.B. Your application need not handle bracketed sub-formulae within a formula.

5) Design a Java class `Matrix` whose instances represent integer matrices. As well as appropriate constructors, your class should provide the following two methods:

- a method to return the index of the largest element in a specified row of the matrix;
- a method to determine if the matrix has a "saddle point". A saddle point is an element that is both the largest in its row and the smallest in its column.

6) Design a Java class whose instances represent books including any attributes and methods that you think appropriate.

Making use of your book class, design a further class whose instances represent bookshelves and which includes methods to:

- retrieve a named book from the shelf (if it is present);
- return a book to the shelf (if there is space); and
- report whether or not a named book is on the shelf.

7) Java is a strongly-typed language in that every variable has a type and only values that conform to the corresponding type can be assigned to a given variable. Describe the facilities that Java provides to define types and the rules that govern which objects can be assigned to a variable of a given (reference) type.

8) Write brief notes on **all** of the following:

a) recursion;
b) garbage collection;
c) the facilities that Java provides for building graphical user interfaces.