Interrogation d'informatique

PCC-ASINSA-SCAN 1^{st} year - Avril 2016

Total duration : *1h30*

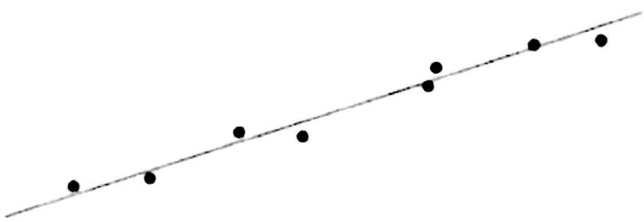
Allowed materials : *None.*

- The graduation can still change.
- A program **poorly indented, poorly commented** or with **poor variable names** will be **penalized**.
- The assignment is on 5 pages.
- All the questions are independant. If it is necessary to have solved A to solve B, then you can do as if you had solved A (and clearly indicate your assumption) to solve B.
- You can define syntactic aliases as long as you define them before.

Thought of the day:
« Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. » *Martin Golding*

1 Linear interpolation (11pts)

During an experiment, a set of n points (in the 2D space) (x_i, y_i) has been measured. We want to find the equation of line, $y = ax + b$, that fits the best those n points. The following figure illustrates this problem : the n points represent the measurements while the approximate line is in red.



To do so, we will use the *least squares* method : **we search** (a, b) **such that** $y_i \simeq ax_i + b$, $\forall i \in \{1..n\}$.

We can reformulate the problem as an over-constrained system of linear equations, having the following matrix representation :

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_i & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{S} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}}_{V}$$

The solution of a such system is given by : $S = \begin{bmatrix} a \\ b \end{bmatrix} = (U^{\intercal}U)^{-1}U^{\intercal}V$

In the following, we will represent the matrices U and $U^{\intercal}$ by 2D arrays of real numbers (n lines and 2 columns for U) and vectors S (2 elements) and V (n elements) by 1D arrays of real numbers.

(Q1.1) Matrix Transposition (1 pt)

Write the method `transposeMatrix` that takes a matrix A as parameter and returns its transpose matrix $B = A^{\intercal}$.

Reminder : if $A = (a_{i,j})$ is a matrix of dimensions (M, N), then $B = A^{\intercal}$ is a matrix of dimensions (N, M) such that $b_{j,i} = a_{i,j}$

(Q1.2) Displaying a matrix (1pt)

We want to be able to display matrices used throughout this problem. Write the method `displayMatrix` that takes a matrix A as parameter and displays the matrix under the following format :

1.2	2.4	3.5	4.0	1.1
0.3	4.6	32.6	1.4	2.3
4.2	1.9	0.3	1.9	1.3
6.7	76.4	3.4	2.0	2.3
10.9	9.4	21.3	4.4	2.7

(Q1.3) Product of a matrix by a vector (1.5 pt)

Write the method `matrixVectorProduct` that takes a matrix A and a vector B as parameters and returns the vector representing the result of the product $C = AB$. In this method as in the following one, we consider that the dimensions of the vector and the matrix are compatible to allow the product.

Reminder : if $A = (a_{i,j})$ is a matrix of dimensions (M, N) and $B = (b_j)$ is a vector (column) of dimension N, then $C = AB = (c_i)$ is a vector of dimension M such that $c_i = \sum_{j=1}^{N} a_{i,j}b_j$

(Q1.4) Product of two matrices (2 pt)

Write the method `matrixProduct` that takes a matrix A and a matrix B as parameters and returns the result of their product $C = AB$.

Reminder : if $A = (a_{i,j})$ is a matrix of dimensions (M, N) and $b = (b_{j,k})$ is a matrix of dimensions (N, P) then $C = AB = (c_{i,k})$ is a matrix of dimensions (M, P) such that :
$c_{i,k} = \sum_{j=1}^{N} a_{i,j}b_{j,k}$

(Q1.5) System resolution (2.5 pts)

Write the method **solveSystem** that takes the vectors $X = (x_i)$ and $Y = (y_i)$ as parameters and returns the vector $S = \begin{bmatrix} a \\ b \end{bmatrix}$ by calling the methods written in (Q1.1), (Q1.3) and (Q1.4). We assume that the method **invertMatrix** is already written. This method takes a square matrix A as parameter and returns its inverse A^{-1}.

(Q1.6) Computing the error (1.5 pts)

Write the method **computeError** that takes as parameters the vectors $X = (x_i)$, $Y = (y_i)$ and the vector $S = \begin{bmatrix} a \\ b \end{bmatrix}$ and returns the approximation error $E = \sum_{i=1}^{n} (y_i - ax_i - b)^2$

(Q1.7) Solving a problem (1.5 pts)

During an experiment, the $n = 6$ following values have been measured :
$X = \{x_1, x_2, x_3, x_4, x_5, x_6\} = \{0.8, 2.2, 2.8, 4.2, 4.8, 6.2\}$
and
$Y = \{y_1, y_2, y_3, y_4, y_5, y_6\} = \{11.9, 21.7, 31.8, 42.0, 51.9, 61.9\}$

Write the **main** method that computes the solution $S = \begin{bmatrix} a \\ b \end{bmatrix}$ as well as the corresponding approximation error E. We will display on the screen the values of a, b and E.

For your information : the solution to this problem is $S = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 9.57 \\ 3.34 \end{bmatrix}$

2 The return of the Binary (9.5 pts)

We want to go back to the Practical Work "Binary Arithmetic" but this time using Object Oriented Programming. The objective is to manipulate positive integers under their binary form using a set of integers whose values are 0 or 1.

Warning : Throughout this exercise 2, it is mandatory to specify the name of the files in which your answers should be placed. The names of the file must be given **with** their extension.
Reminder : the extension of the file named `Hello.txt` is `txt`, i.e. the string of characters after the dot.
To simplify the problem, we will consider that the positive integers are coded on 2 bits. We will use a class called `Binary` having two **public** attributes of type `int` named `b0` and `b1` and corresponding respectively to the least significant bit and the most significant bit. The corresponding integer is

$$2^1 \times b_1 + 2^0 \times b_0$$

(Q2.1) Creation of the class (3 pts)

Write the body of the class `Binary` including :
- The attributes (0.5 pt) ;
- The constructor taking two integers corresponding to b_0 and b_1 as parameter (1 pt) ;
- The default constructor that takes no parameter (0.5 pt) ;
- The `toString` method that returns a textual representation of the binary number. For instance the 01_2 should show as : `01` (1 pt).

(Q2.2) Testing of the class (1 pt)

Write the code that can be used to test this first version of `Binary` by :
- creating two Binary u and v initialized to 01 and 11 respectively, $u : 01_2$ $v : 11_2$;
- displaying a textual representation of u and v.

Remember that you need to write the full code that allows the compilation and execution of your program (reminder : specify in which file your code should be placed).

(Q2.3) Some useful methods (2 pts)

Create the following methods of the `Binary` class and give the code of the `main` method allowing to test each of them by calling them for u :
- `inDecimal` : returns the corresponding integer in base 10 ;
- `isEven` : returns `true` if and only if the number is even.

(Q2.4) Directly from an integer (2 pt)

Write the code allowing to the user of the `Binary` class to create *in a single instruction* an instance of `Binary` from an integer of type `int` belonging to $\{0, 1, 2, 3\}$.
We will write this method as well as the potential modification that are required in the `Binary` class. **Warning**, those modifications must not change the way your code was executing until now.

(Q2.5) Summing ? (1.5 pts)

We assume that u and v are two instances of `Binary` that have been declared and initialized. We want to compute the result of the sum of :

- the value of u and v ;
- the (initial) value of u and 5.

To do so, we want to use the following code in the `main` method :

```
u.addInteger(v);
System.out.println(v);
u.addInteger(5);
System.out.println(u);
```

Is this possible ? If yes, suggest the modification of the code allowing to do so. If not, explain why.