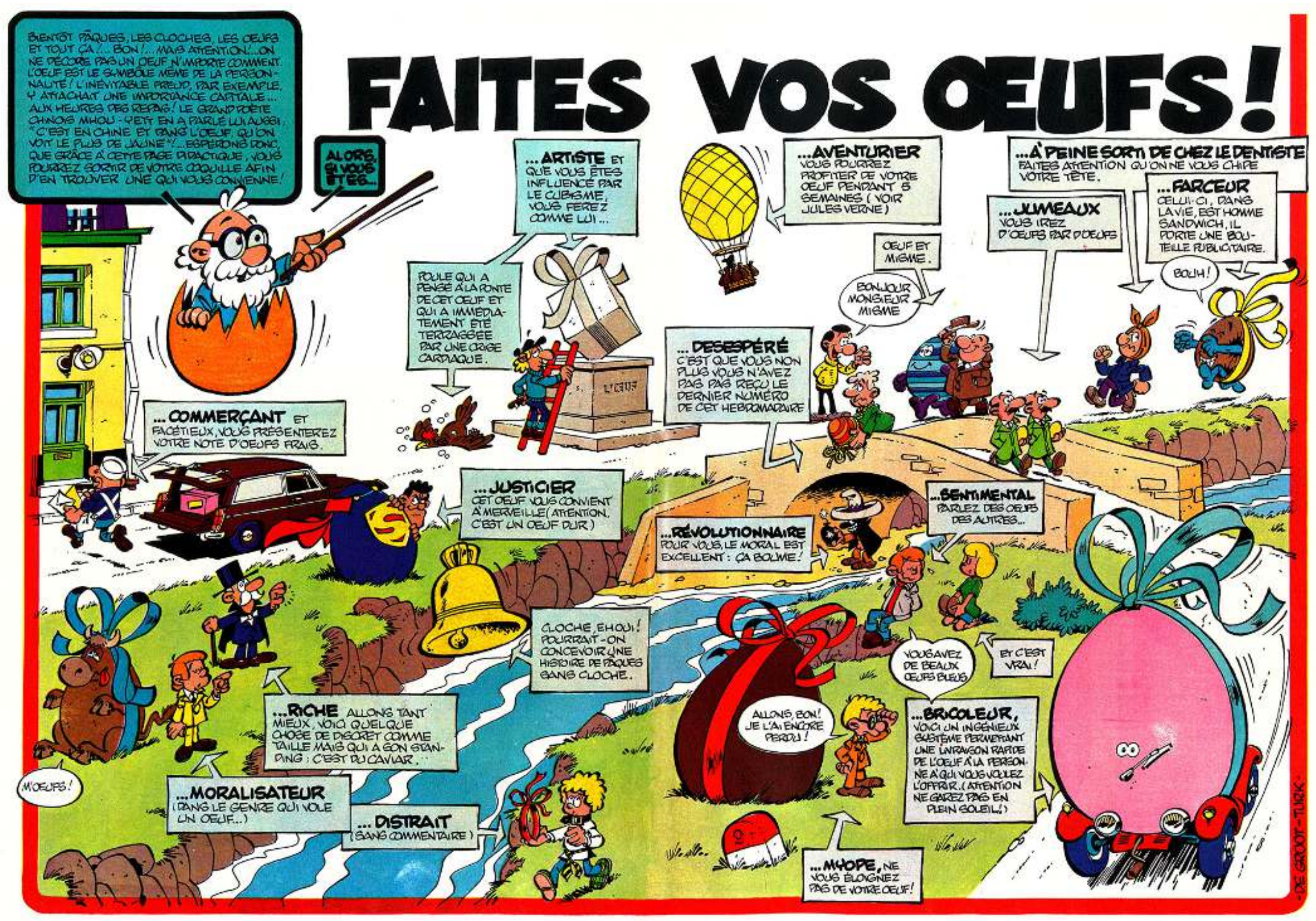
Algorithmics
Midterm Exam #1

Undergraduate 1st year S1#
EPITA

23 avril 2019 - 13:30

Notes (read them !) :

- ☐ You must answer on **the answer sheets provided**. No other sheet will be collected. Keep your rough drafts.
  Do not separate the sheets unless they can be re-stapled before handing them in.
- ☐ The presentation is negatively marked, which means that you are marked out of 20 points and the presentation points (maximum of 2) are taken off this grade.
- ☐ All CAML code not indented will not be marked.
- ☐ Any CAML code must be followed by its evaluation: the CAML response.
- ☐ In the absence of any indication in the document, the only functions that you can use are `failwith` and `invalid_arg` (no other predefined function of CAML).
- ☐ Penciled answers will not be marked.
- ☐ Duration : 2h (May the force...)


FAITES VOS ŒUFS!
BIENTÔT PÂQUES, LES CLOCHES, LES OEUFS ET TOUT ÇA!... BON!... MAIS ATTENTION!...ON NE DÉCORE PAS UN OEUF N'IMPORTE COMMENT. L'OEUF EST LE SYMBOLE MÊME DE LA PERSONNALITÉ! L'INÉVITABLE FREUD, PAR EXEMPLE, Y ATTACHAIT UNE IMPORTANCE CAPITALE... AUX HEURES DES REPAS! LE GRAND POÈTE CHINOIS MHOU-YETT EN A PARLÉ LUI AUSSI: "C'EST EN CHINE ET DANS L'OEUF QU'ON VOIT LE PLUS DE JAUNE"!... ESPÉRONS DONC QUE GRÂCE À CETTE PAGE DIDACTIQUE, VOUS POURREZ SORTIR DE VOTRE COQUILLE AFIN D'EN TROUVER UNE QUI VOUS CONVIENNE!
ALORS, SI VOUS ÊTES...
...ARTISTE ET QUE VOUS ÊTES INFLUENCÉ PAR LE CUBISME, VOUS FEREZ COMME LUI...
POULE QUI A PENSÉ À LA PONTE DE CET OEUF ET QUI A IMMÉDIATEMENT ÉTÉ TERRASSÉE PAR UNE CRISE CARDIAQUE.
L'OEUF
...AVENTURIER VOUS POURREZ PROFITER DE VOTRE OEUF PENDANT 5 SEMAINES (VOIR JULES VERNE)
...À PEINE SORTI DE CHEZ LE DENTISTE FAITES ATTENTION QU'ON NE VOUS CHIPE VOTRE TÊTE.
...JUMEAUX VOUS IREZ D'OEUFS PAR D'OEUFS
...FARCEUR CELUI-CI, DANS LA VIE, EST HOMME SANDWICH, IL PORTE UNE BOUTEILLE PUBLICITAIRE.
BOUH!
OEUF ET MISME.
BONJOUR MONSIEUR MISME
...DESESPÉRÉ C'EST QUE VOUS NON PLUS VOUS N'AVEZ PAS PAS REÇU LE DERNIER NUMÉRO DE CET HEBDOMADAIRE
...COMMERÇANT ET FACÉTIEUX, VOUS PRÉSENTEREZ VOTRE NOTE D'OEUFS FRAIS.
...JUSTICIER CET OEUF VOUS CONVIENT À MERVEILLE (ATTENTION, C'EST UN OEUF DUR)
...RÉVOLUTIONNAIRE POUR VOUS, LE MORAL EST EXCELLENT: ÇA BOUME!
...SENTIMENTAL PARLEZ DES OEUFS DES AUTRES...
CLOCHE, EH OUI! POURRAIT-ON CONCEVOIR UNE HISTOIRE DE PÂQUES SANS CLOCHE.
VOUS AVEZ DE BEAUX OEUFS BLEUS
ET C'EST VRAI!
...RICHE ALLONS TANT MIEUX, VOICI QUELQUE CHOSE DE DISCRET COMME TAILLE MAIS QUI A SON STANDING: C'EST DU CAVIAR.
M'OEUFS!
...MORALISATEUR (DANS LE GENRE QUI VOLE UN OEUF...)
...DISTRAIT (SANS COMMENTAIRE)
ALLONS, BON! JE L'AI ENCORE PERDU!
...BRICOLEUR, VOICI UN INGÉNIEUX SYSTÈME PERMETTANT UNE LIVRAISON RAPIDE DE L'OEUF À LA PERSONNE À QUI VOUS VOULEZ L'OFFRIR. (ATTENTION NE GAREZ PAS EN PLEIN SOLEIL!)
...MYOPE, NE VOUS ÉLOIGNEZ PAS DE VOTRE OEUF!
-DE GROOT-TURK-

Exercise 1 (Abstract Types: Vector (errors and extension) – *6 points*)

Let the algebraic abstract data type *Vector* studied in the course defined as follows.

TYPES

vector

USES

integer, element, boolean

OPERATIONS

vect	:	integer × integer → vector
modify	:	vector × integer × element → vector
nth	:	vector × integer → element
isinit	:	vector × integer → boolean
lowerlimit	:	vector → integer
upperlimit	:	vector → integer

PRECONDITIONS

nth(v,i) is defined if-and-only-if lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) & isinit(v,i)=true

AXIOMS

lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) $\Rightarrow$ nth(modify(v,i,e),i) = e
lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) & lowerlimit(v) $\leqslant$ j $\leqslant$ upperlimit(v) & i$\neq$j
$\Rightarrow$ nth(modify(v,i,e),j) = nth(v,j)

lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) $\Rightarrow$ isinit(modify(v,i,e),i)=true
lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) & lowerlimit(v) $\leqslant$ j $\leqslant$ upperlimit(v)
$\Rightarrow$ isinit(modify(v,i,e),j)=isinit(v, j)

lowerlimit(vect(i,j))=i
lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) $\Rightarrow$ lowerlimit(modify(v,i,e))=lowerlimit(v)

upperlimit(vect(i,j))=j
lowerlimit(v) $\leqslant$ i $\leqslant$ upperlimit(v) $\Rightarrow$ upperlimit(modify(v,i,e))=upperlimit(v)

WITH

vector	v
integer	i, j, k
element	e

1. This definition is incorrect. Indeed, this set of axioms has two problems. For each of these problems, precise its nature, give a description and give a solution to fix it.

2. Now that the problems are solved, we have the algebraic type *Vector*. We suggest an extension to this type by defining a new operation: *reinitialize*. This will allow us to set a given position of the vector to its initial state (*i.e. uninitialized)*. Its profile is the following:

   OPERATIONS

   reinitialize: vector × integer → vector

   (a) Precise the possible domain of definition of this operation (the preconditions).

   (b) Give the axioms allowing a complete definition of this operation.

Exercise 2 (Insertion Sort – *7 points*)

1. Write the function `insert` that adds an element in its place in a list sorted according to a given comparison function.

   Examples of results:

```
♯ insert 12 [1;5;9;13;15;28] (function x -> function y -> x <= y) ;;
 - : int list = [1; 5; 9; 12; 13; 15; 28]
♯ insert 12 [28;15;13;9;5;1] (function a -> function b -> a >= b) ;;
 - : int list = [28; 15; 13; 12; 9; 5; 1]
♯ insert 12 [] (function a -> function b -> a >= b) ;;
 - : int list = [12]
```

2. Use the function `insert` to write a function that sorts a list in order according to a given comparison function.

   Examples of results:

```
♯ insertion_sort (function x -> function y -> x >= y) [12;5;47;1;23;0;48;35;3;14;9;11;8;7;65] ;;
 - : int list = [65; 48; 47; 35; 23; 14; 12; 11; 9; 8; 7; 5; 3; 1; 0]

♯ let longer s1 s2 = String.length s1 > String.length s2 ;;
val longer : string -> string -> bool = <fun>
♯ insertion_sort longer ["Caml"; "C#"; "Python"; "C"; "Javascript"];;
 - : string list = ["Javascript"; "Python"; "Caml"; "C#"; "C"]
```

Exercise 3 (Association – 5 points)

Write down the function `assoc k list` where `k` is a natural and `list` a list of pairs (*key*, *value*) sorted in increasing order with respect to keys. We assume keys are always naturals. The function returns the value corresponding to the key `k`. It raises an exception if `k` is not a valid key or if it does not correspond to any pair.

Examples of applications:

```
♯ assoc 5 [(1, "one"); (2, "two"); (3, "three"); (5, "five"); (8, "eight")];;
 - : string = "five"

♯ assoc 4 [(1, "one"); (2, "two"); (3, "three"); (5, "five"); (8, "eight")];;
 Exception: Failure "not found".

♯ assoc (-1) [(1, "one"); (2, "two"); (3, "three"); (5, "five"); (8, "eight")];;
 Exception: Invalid_argument "k not a natural".
```

Exercise 4 (Mystery – 2 points)

The `mystery` function is defined as

```
♯ let mystery = function
        [] -> failwith "..."
      | e::f::l -> (let rec aux_mystery m1 m2 = function
                        [] -> m2
                      | e::l -> if e < m1 then aux_mystery e m1 l
                                else if e < m2 then aux_mystery m1 e l
                                else aux_mystery m1 m2 l
                    in if e < f then aux_mystery e f l else aux_mystery f e l);;
```

1. Give the results of the successive evaluations of the phrases on the answer sheets.
2. What is the return value of `mystery`?