

## Devoir en temps limité du Vendredi 3 Mars

Dans l'ensemble du devoir, toutes les fonctions seront :

- écrites en langage OCaml,
- précédées d'une explication des variables utilisées,
- précédées d'une explication de l'algorithme.

### Exercice 1 (La syntaxe en OCaml)

- Quels sont les éléments caractéristiques des types `array` et `list` en OCaml, différences et analogies ?
- Donner la réponse renvoyée par OCaml pour les codes suivants sous la forme `-: type = valeur` :
 

(a) <code>3+5=8</code> ; ;	(b) <code>3/8</code> ; ;
(c) <code>[]</code> ; ;	(d) <code>2+.3</code> ; ;
(e) <code>[1,2;3,0]</code> ; ;	(f) <code>"informatique".[2]</code> ; ;
(g) <code>[[ 2;3 ]; [ 1;2;4 ]]</code> ; ;	(h) <code>Array.sub</code> ; ;
(i) <code>List.hd [ 2;4 ]</code> ; ;	(j) <code>[1,2;4,5].(1)</code> ; ;
- Écrire un code en OCaml qui réalise les opérations suivantes et indiquer la réponse renvoyée par OCaml sous la forme `-: type = valeur` :
  - calcul approchée du nombre d'or  $\frac{1 + \sqrt{5}}{2}$ ,
  - affecte au troisième élément du tableau `[|1; 4; 5; 2; 8|]` la valeur 0,
  - calcul du troisième élément de la liste `[1; 4; 5; 2; 8]`,
  - créé le tableau `[|2; 2; 2|]`.
- Expliquer la façon dont OCaml type les expressions suivantes (typage de `x` et de l'expression complète) :
  - `List.length ([2;4]::x)` ; ;
  - `x@[|11|]` ; ;
  - `x::[]` ; ;

### Exercice 2 (Du cours...)

- Factorielle** :
  - Écrire en OCaml une fonction itérative qui, étant donné  $n$ , calcule  $n!$ .
  - Écrire en OCaml une fonction récursive qui, étant donné  $n$ , calcule  $n!$ .
  - Écrire en OCaml une fonction récursive terminale qui, étant donné  $n$ , calcule  $n!$ .
- Suite récurrente** :
 

Considérons la suite réelle  $(u_n)$  définie par :

$$\begin{cases} u_0 = 2, \\ \forall n \in \mathbb{N}, u_{n+1} = \ln(1 + u_n) \end{cases}$$

- (a) Écrire en OCaml une fonction itérative qui renvoie, pour tout entier naturel  $n$ , la valeur de  $u_n$ .
- (b) Écrire en OCaml une fonction récursive qui renvoie, pour tout entier naturel  $n$ , la valeur de  $u_n$ .
- (c) Écrire en OCaml une fonction récursive terminale qui renvoie pour tout entier naturel  $n$ , la valeur de  $u_n$ .

---

### Exercice 3 (Manipulation récursive de tableau en place)

On considère un tableau  $a$  de longueur  $n$  contenant des valeurs d'un ensemble totalement ordonné  $(E, \leq)$ .

Nous nous proposons de rechercher un indice de la valeur maximale parmi les valeurs contenues dans le tableau  $a$ , autrement un entier naturel  $p$  compris entre 0 et  $n - 1$  tel que :

$$a[p] = \max(\{a[0], a[1], \dots, a[n - 1]\}).$$

1. Écrire en OCaml un algorithme itératif qui répond à cette question.
2. Nous nous proposons maintenant de développer une approche récursive.
  - (a) Soit  $i$  et  $j$  deux entiers naturels compris entre 0 et  $n - 1$  avec  $i \leq j$ .

Nous nous intéressons dans cette question à la recherche d'un indice de la valeur maximale parmi les valeurs  $a[k]$ ,  $k$  appartenant à  $\llbracket i, j \rrbracket$ , autrement dit un entier naturel  $p$  compris entre  $i$  et  $j$  tel que :

$$a[p] = \max(\{a[i], a[i + 1], \dots, a[j - 1], a[j]\}).$$

Quel est cet indice dans le cas où  $i = j$  ?

Dans le cas où  $i < j$ , expliquer comment déterminer l'indice cherché à partir d'un indice  $q$  de la valeur maximale parmi les valeurs  $a[k]$ ,  $k$  appartenant à  $\llbracket i + 1, j \rrbracket$ , c'est-à-dire tel que :

$$a[q] = \max(\{a[i + 1], a[i + 2], \dots, a[j - 1], a[j]\}).$$

En déduire une fonction récursive

```
maximum_aux a i j
```

qui renvoie un indice  $p$  de la valeur maximale parmi les valeurs  $a[k]$ ,  $k$  appartenant à  $\llbracket i, j \rrbracket$ , c'est-à-dire tel que :

$$a[p] = \max(\{a[i], a[i + 1], \dots, a[j - 1], a[j]\}).$$

- (b) En déduire une fonction

```
maximum a
```

qui renvoie un indice de la valeur maximale parmi les valeurs contenues dans tout le tableau  $a$ .