Devoir en temps limité du Vendredi 28 Février

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)

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

   (a) `3+5=8 ;;`
   (b) `3/8 ;;`
   (c) `[[]] ;;`
   (d) `2+.3. ;;`
   (e) `[1,2;3,0] ;;`
   (f) `"informatique".[2] ;;`
   (g) `[[|2;3|];[|1;2;4|]] ;;`
   (h) `Array.sub ;;`
   (i) `List.hd [[|2;4|]] ;;`
   (j) `[1,2;4,5].(1) ;;`

3. Écrire un code en OCaml qui réalise les opérations suivantes et indiquer la réponse renvoyée par OCaml sous la forme `-: type = valeur` :

   (a) calcul approchée du nombre d'or $\dfrac{1+\sqrt{5}}{2}$,
   (b) affecte au troisième élément du tableau $[|1;4;5;2;8|]$ la valeur 0 ,
   (c) calcul du troisième élément de la liste $[1;4;5;2;8]$,
   (d) créé le tableau $[|2;2;2|]$.

4. Expliquer la façon dont OCaml type les expressions suivantes (typage de `x` et de l'expression complète) :

   (a) `List.length ([2;4]::x) ;;`
   (b) `x@[[11]] ;;`
   (c) `x::[] ;;`

Exercice 2 (Du cours...)

1. **Factorielle :**

   (a) Écrire en OCaml une fonction itérative qui, étant donné n, calcule $n!$.
   (b) Écrire en OCaml une fonction récursive qui, étant donné n, calcule $n!$.

2. **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 $\mathtt{u_n}$.

Exercice 3 (Manipulation récursive de tableau en place)

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

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 :

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

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

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

   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 $[\![i+1, j]\!]$, c'est-à-dire tel que :

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

   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 $[\![i, j]\!]$, c'est-à-dire tel que :

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

   (b) En déduire une fonction

   `maximum a`

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