info421 : Programmation fonctionnelle
TD 1 : premières expressions en Caml

Responsables : Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
email : Pierre.Hyvernat@univ-savoie.fr)
http://lama.univ-savoie.fr/~hyvernat/

Exercice 1 : expressions et types simples

Question 1. Tartempion lance l'interprète Caml et tape les lignes suivantes.

```
# 401 / 10 ;;
# 10 + 1.5 ;;
#   "Bonjour, "
  ^ "je m'appelle Bob."  ;;
# "Je m'appelle" ^ (2*"to") ;;
# let pi = 3.14159 ;;
# let x = 3.0 *. pi /. 2.0 ;;
# if (x < 2*pi) then x
                else (x-2*pi) ;;
# let n = 42 in
  ( if (n > String.length("anticonstitutionnellement"))
      then 3+String.length("Bob")
      else 0 )
  + int_of_float (2. *. pi)  ;;
# if ( if n<0 then true else (pi > 22.0 /. 7.0) )
    then "Toto"
    else "Tata" ;;
```

Quelles sont les expressions valides, et quelles sont leurs types et leurs valeurs ?

Lesquelles sont des *définitions* et lesquelles sont des *valeurs* ?

Exercice 2 : Définitions

Rappels : nous avons vu plusieurs utilisation du mot clé `let` en cours :

- une définition simple : `let` *var* `=` *expr*
- une définition multiple : `let` var_1 `=` $expr_1$ `and` var_2 `=` $expr_2$
- une définition récursive : `let rec` *var* `=` *expr*
- une définition locale : `let` *var* `=` *expr* `in` *expr*$'$

Une définition locale peut elle aussi être simple, multiple ou récursive... Chaque définition modifie l'environnement en rajoutant des liaisons entre des noms de variables et des valeurs...

Question 1. Quelles sont les différences entre les expressions Caml suivantes, et que se passe-t'il quand on les donne à l'interprète Caml :

```
# 2 + 1 + 3 ; ;
# let n = 2 + 1 + 3 ; ;
# let rec n = 2 + 1 + 3 ; ;
```

Question 2. Décrivez l'environnement après chaque ligne :

```
# let a = 1 ;;
# let b = 2 ;;
# let c = 3 ;;
# let b = a+2 ;;
# let c = a+b+c ;;
```

```
# let a = 0 and d = 1
  in  a + b + d ;;
# let e = d + 1 ;;
# let l = String.length ;;
# let un_nombre = let e = 2.718 in
                  let l = 42 in
                  e +. (float_of_int l) ;;
# l ("Salut" ^ " !") ;;
# #quit ;;
```

Question 3. En voulant définir deux valeurs, Tartempion écrit la chose suivante dans l'interprète Caml :

```
# let a=42 and b=a+1 ; ;
```

Que se passe-t'il, et pourquoi ?

Comment Tartempion aurait-il du s'y prendre ?

Exercice 3 : un peu de récursion

Question 1. Tartempion essaie de définir la suite

$$u_0 = 0 \qquad u_n = n + u_{n-1}$$

Il commence par écrire

```
# let rec u n =
    if (n < 1)
      then 0
      else u (n-1) ;;
```

mais après quelques tests, il se rend compte qu'il a oublié la partie "$n+\ldots$". Il fait un deuxième essai :

```
# let u n =
    if n<1
      then 0
      else n + u (n-1) ;;
```

Que se passe-t'il ?

Question 2. Écrivez une fonction `somme` à un argument n pour calculer $1 + 2 + 3 + \ldots + n$.

Question 3. Ecrivez une fonction récursive `fact` pour calculer : $n! = 1 \times 2 \times \ldots \times n$.

Question 4. Écrivez une fonction `somme2` qui calcule la somme $1 + 2^2 + 3^2 + 4^2 + \ldots + n^2$.

Question 5. Écrivez des fonctions pour calculer :

- $\sum_{i=1}^{n} i^3$,
- $\sum_{i=1}^{n} (3i^2 + 2i)/3$,
- $\sum_{i=1}^{n} (-1)^i \times |7i^3 - 4i^2 + 12|$,
- $\sum_{i=1}^{n} i^i$.

Question 6. Essayez de trouver une manière plus "élégante" de programmer les fonctions de la questions précédente en utilisant une fonction d'*ordre supérieur*.

Donnez le type de la fonction correspondante.