Institut National Universitaire Champollion

TP 1 - Prise en main

Programmation fonctionnelle en Caml
L3 INFO - Semestre 6

Objectifs : Prise en main de l'environnement de programmation Caml Light.
Introduction à la programmation fonctionnelle, expressions élémentaires, types de base, priorités et conversions.

1. Expressions élémentaires et types de base

Éxécuter et commenter les requêtes suivantes :
Vous pouvez taper ces commandes dans l'éditeur, les recopier par copier/coller dans la fenêtre de terminal de Caml puis recopier dans l'éditeur le résultat de l'exécution.

Entiers et réels

```
1.2 + 1 ;;
1.2 + 2.3 ;;
-2E-1 +. 2. ;;
(sqrt(4.)+. 2.)/. 3.5 ;;
-2 * 3 ;;
2.1 +. 4.9 ;;
```

```
sqrt 9. ;;
10/3 ;;
10 mod 3 ;;
2+ 3*5 ;;
-(5+1)*(-2+5)+2*3 ;;
```

Booléens

```
1 = 2 ;;
4 < 5 ;;
4.2 < 4.7 ;;
(1=1)=(2<1) ;;
true = 1 ;;
true or false ;;
true or false = (1=1) & (4<5) ;;
```

```
(1+2 = 2+1) & 4>5 ;;
1+2 = 2+1 & 4>5 ;;
1+2 = 2+1 or 4>5 ;;
1+2 = 2+1 > 4 > 5 ;;
1+2 = 2+1 > (4>5) ;;
false < true ;;
4 + 1 < 6 & ('a'<'h' or "debut" = "fin") ;;
```

Exercice : Construire et évaluer des expressions booléennes permettant de prouver que
- les comparaisons sont prioritaires sur not
- not est prioritaire sur & et or
- & est prioritaire sur or

Chaînes de caractères

```
"salut" ;;
"salut"^"à tous" ;;
"salut"^" à tous" ;;
"salut" < "bonjour" ;;
"salut" < "Salut" ;;
```

```
"A" < "a" ;;
'A'< 'a' ;;
'a';;
int_of_char('a') ;;
'a' < 'b' ;;
```

```
'a'< "bonjour" ;;
"a" < "bonjour" ;;
'a'^"près" ;;
```

```
"a"^"près" ;;
"12" > "2" ;;
```

2. Conversion de types

Caml offre plusieurs fonctions permettant de convertir certaines valeurs d'un type dans un autre. La fonction `int_of_float` convertit par exemple un réel en entier.
Testez les fonctions de conversions avec les exemples suivants :

```
int_of_float ;;
int_of_float(4.0) ;;
int_of_float(4.25) ;;
```

```
int_of_float(-4.25) ;;
int_of_float(4.25e-34) ;;
```

```
string_of_int(-235) ;;
string_of_int(55e2) ;;
```

```
string_of_float(55e2);;
string_of_float(-55e-2) ;;
```

```
int_of_string "345" ;;
int_of_string "34.5" ;;
float_of_string "34.5" ;;
```

```
float_of_string
"99999999999999999999999.9" ;;
```

3. Définitions globales et locales

Donner la valeur de chaque expression, et préciser l'évolution de l'environnement.

```
let x = 2  ; ;
let y = x + 3  ; ;
let x = y + 5  ; ;
let z = y*2 in x+z+y*y  ; ;
```

```
let x = 3 in x*x+2*x*y + 4*y  ; ;
let x = 1 in x = 2*x  ; ;
let x = 0 in x = 2*x  ; ;
```

Définitions locales emboîtées

```
let x = 5 in
     let prod = x*x in
          prod + prod*prod  ; ;

let resultat = let x = 5 in
     let prod = x*x in
          prod + prod * prod  ; ;
```

```

let val = let x = 3 ad y = 4 in
     let x = x+y and y = x-y in
          x*x + y*y  ; ;

let y = 2 in val*val+2*val*y  ; ;
```

Expressions conditionnelles.

```
if (1=1)
     then "salut" else "au revoir"  ; ;

let x = 3 in
     if (x<0) then x else x*x  ; ;

if (5>0)
     then 1 else "erreur"  ; ;
```

```

let x=3 and y=3 in
     let y = y*x in
          if y mod 2 = 0 then "pair"
          else "impair"  ; ;
```