

TP 2 - Premières fonctions. Produits cartésiens

Programmation fonctionnelle en Caml
L3 INFO - Semestre 6

1. Quelques fonctions sur les nombres

Écrire les fonctions suivantes :

1. *troncature* de type *float* -> *int*, qui à un réel x , associe l'entier obtenu en débarassant x de ses décimales.

```
# troncature 42.45 ; ;
- : int = 42
# troncature (-42.45) ; ;
- : int = -42
```
2. *decimales* de type *float* -> *float*, qui à un réel x associe sa partie décimale (Attention aux nombres négatifs !).
(On pourra utiliser la fonction `abs_float`.)

```
#decimales(-42.45) ; ;
- : float = 0.45
```
3. *partie_entiere* de type *float* -> *int*, qui au réel x , associe le plus grand entier inférieur à x . (La partie entière de -3.2 est -4 , celle de -3.00 est -3 et celle de 5.3 est 5 .)
La partie entière d'un réel x est égal à sa troncature si $x - \text{troncature}(x)$ est positif ou nul et à sa troncature moins 1 sinon.

```
#partie_entiere(-5.61) ; ;
- : int = -6
```
4. *plus_proche_entier* de type *float* -> *int*, qui au réel x associe l'entier le plus proche.
On pourra noter qu'il s'agit de la partie entière de $x + 0.5$.

```
#plus_proche_entier(-5.52) ; ;
- : int = -6
```
5. *arrondi* de type *float* -> *float*, qui à x associe le réel à deux décimales le plus proche du réel x .
Il faut utiliser la fonction précédente, bien entendu.

```
#arrondi(52.6543) ; ;
- : float = 52.65
```

2. Fonction de conversion francs-euros

Écrire une fonction *francs_en_euros* de type *float* -> *float*, permettant de convertir en euros une somme exprimée en francs français. On rappelle que le taux de conversion est de 1 euros pour 6.55957 francs.

Améliorer la présentation en arrondissant le résultat au cent inférieur (ce qui revient à arrondir à deux décimales).

3. Quelle heure est-il ?

On veut mettre une heure donnée sous forme hh.mm sous une forme plus conviviale.

Écrire une fonction `quelle_heure_est_il` qui réalise ce type de transformation. Par exemple :

```
#quelle_heure_est_il(14.45) ; ;
- : string = "Il est 14 heure 45"
#quelle_heure_est_il(12.0) ; ;
- : string = "Il est midi pile."
#quelle_heure_est_il(0.34) ; ;
- : string = "Il est minuit 34"
```

Commencez par écrire deux fonctions heures et minutes telles que par exemple :

```
#heures 23.42 ; ;
- : int = 23
#minutes 23.42 ; ;
- : int = 42
```

4. Fonctions sur un produit cartésien

Écrire une fonction `reel` qui à un réel x et à deux entiers a et b associe le réel dont la troncature est le maximum de a et b (calculé obligatoirement dans une définition locale) et qui a même partie décimale que x .

On pourra utiliser la fonction `abs_float`

```
#reel(4,5,-4.123) ; ;
- : float = 5.123
#reel(7,5,4.123) ; ;
- : float = 7.123
```

5. Écrire les fonctions entières suivantes :

1. `chiffre(n)` qui à un entier n associe son dernier chiffre
2. `echange(n,p)` qui à deux entiers n et p associe l'entier obtenu en remplaçant le dernier chiffre de n par celui de p .

6. Fonction booléennes

Écrire les fonctions booléennes qui à une triplet d'entier (a,b,c) associent respectivement vrai si et seulement si :

1. a, b et c ont même valeur
2. a et b sont égaux mais différents de c
3. la valeur de b est strictement comprise entre a et c

4. parmi a, b et c deux valeurs au moins sont identiques
5. parmi a, b et c deux valeurs exactement sont identiques
6. parmi a, b et c deux valeurs au plus sont identiques

7. Résolution de l'équation du second degré

On rappelle que pour résoudre l'équation du second degré (E), on calcule le discriminant $\Delta = b^2 - 4ac$ et qu'alors

- Si $\Delta > 0$, (E) admet deux racines réelles distinctes
- Si $\Delta = 0$, (E) admet une racine réelle double
- Si $\Delta < 0$, (E) n'admet pas de racines réelles

Écrire une fonction `nb_sol(a, b, c)` qui calcule le nombre de solutions de l'équation $ax^2 + bx + c = 0$. On utilisera obligatoirement une définition locale pour calculer Δ .

8. Évaluation d'un appel de fonction

Pour A, B et C, on suppose que l'environnement initial E_0 est vide. Pour chaque requête formulée au cours de la session suivante, donnez la réplique du système Caml, et décrire précisément l'évolution de l'environnement.

A.

```
let x = 3 ; ;
let y = 4 ; ;
let f = fun x -> 3*x + y*2 ; ;
let x = 2 in y = x+1 ; ;
f 4 ; ;
f x ; ;
```

B.

```
let f = fun a -> let b = 2 in a * b ; ;
let b = 3 and x = 2 ; ;
f(x) ; ;
f(b) ; ;
```

C.

```
let a = 2 ; ;
let b = 12 in 2*b + 4 ; ;
let f = fun x -> 2*x + a ; ;
f(a) ; ;
```