TD 4 d'Introduction à la programmation

Exercices sur les types

Cinq minutes plus tard

On peut exprimer un instant dans une journée, à la minute près, par deux entiers, le premier compris entre 0 et 23 et le second compris entre 0 et 59. Par exemple *« Cinq heures et trente minutes »* est représenté par les entiers 5 et 30. Concevoir une fonction qui, à partir d'un instant t exprimé par deux entiers, rend les deux entiers qui représenteront l'heure qu'il sera 5 minutes après t. Par exemple $(5, 30)$ devra renvoyer $(5, 35)$, mais $(3, 56)$ devra renvoyer $(4, 1)$ et enfin $(23, 59)$ devra renvoyer $(0, 4)$.

À la seconde près

À présent, nous voulons manipuler des heures à la seconde près, et étendre les opérations possibles. Pour cela, il sera souvent intéressant de convertir un instant, représenté par trois entiers $(h : m : s)$ (représentant respectivement les heures, les minutes et les secondes) en le nombre de secondes depuis le début de la journée (un entier).

1. Construire le type `horaire`, autrement dit :
   - créer le type `horaire` ;
   - écrire une fonction qui à partir de trois entiers h, m et s vérifie si h est compris entre 0 et 23 et si m et s sont compris entre 0 et 59, qui lance un message d'erreur si ce n'est pas vrai et qui renvoie un objet de type `horaire` dont les heures, les minutes et les secondes sont respectivement h, m et s ;
   - écrire trois fonctions `Heures`, `Minutes` et `Secondes` qui, à partir d'un instant t de type `horaire` (de la forme h, m, s)) renvoient chacune un entier correspondant respectivement à h, m et à s.
2. Concevoir et écrire en *Caml* une fonction qui à partir d'un objet de type `horaire` et représentant un instant t, renvoie le nombre de secondes écoulés entre le début de la journée et l'instant t.
3. Concevoir et écrire en *Caml* une fonction qui fait l'opération inverse.
4. Concevoir et écrire en *Caml* une fonction qui à partir de deux instants t_1 et t_2 de type `horaire` indique si t_1 est antérieur ou non à t_2.
5. Concevoir et écrire en *Caml* une fonction qui à partir de deux instants t_1 et t_2 de type `horaire` renvoie un autre objet de type `horaire`, indiquant la durée qui sépare les deux instants.

Les complexes

On peut représenter un nombre complexe par deux réels, le premier étant sa partie réelle, et le deuxième sa partie imaginaire.

1. Construire un type `complexe`.
2. Concevoir et écrire en *Caml* une fonction qui prend en entrée deux nombres complexes c_1 et c_2 qui renvoie le nombre complexe égal à la somme $c_1 + c_2$.
3. Concevoir et écrire en *Caml* une fonction qui prend en entrée deux nombres complexes c_1 et c_2 qui renvoie le nombre complexe égal au produit $c_1 c_2$.
4. Concevoir et écrire en *Caml* une fonction qui prend en entrée un complexe c et qui renvoie sa norme.
5. Concevoir et écrire en *Caml* une fonction qui prend en entrée un complexe c et qui renvoie son argument (il s'agit de l'angle θ tel que $c = |c|.e^{i\theta}$).
6. Concevoir et écrire en *Caml* une fonction qui prend en entrée un complexe c et qui renvoie ce complexe sous la forme d'une paire norme-argument.

Les `flottants`

On peut représenter un nombre décimal d par deux entiers appelés *mantisse* et *exposant* de telle façon que :

$$d = mantisse * 10^{exposant}$$

(Dans cet exercice on se limitera à la base 10).

Par exemple 3,5 peut être représenté par 35 et -1 puisque $3,5 = 35 * 10^{-1}$.

1. Construire un type `flottant` capable de représenter un nombre décimal par deux entiers.
2. Concevoir et écrire en *Caml* une fonction qui étant donnés deux `flottants` d_1 et d_2 renvoie un autre `flottant` représentant le produit $d_1 d_2$.
3. En supposant que vous disposez d'une fonction `puissance` qui, à partir d'un entier x et d'un entier n renvoie l'entier x^n, concevoir et écrire en *Caml* une fonction qui à partir de deux `flottants` d_1 et d_2 renvoie un autre `flottant` représentant la somme $d_1 + d_2$.