

TP numéro 1

Programmation fonctionnelle, ENSIIE

Semestre 4, 2019–20

Exercice 1 : Interaction avec le toplevel

Pour cet exercice, lancer un terminal et entrer la commande `ocaml`. On terminera ses lignes par `;;` pour envoyer la déclaration à l'interpréteur.

1. Définir la constante `pi` de type `float`. On rappelle que $\arctan(1) = \frac{\pi}{4}$.
2. Quels sont les types des opérateurs (`**`) et (`^`) ?
3. Calculer la valeur de 3^{11} .
4. Deviner ce que fait l'opérateur `^` en testant le résultat sur quelques valeurs.

Exercice 2 : Compilation

Pour cet exercice, on va utiliser le compilateur `ocamlc`.

Le contenu d'un fichier `.ml` compilé est exécuté ligne après ligne. On rappelle que la compilation du fichier `.ml` se fait grâce à la commande : `ocamlc foo.ml -o foo`

1. Écrire un programme qui affiche « Bonjour ». Utiliser pour cela la fonction `Printf.printf` dont la syntaxe est similaire au `printf` du langage C.
2. Écrire un programme qui demande un nom et qui affiche un message de bienvenue contenant la valeur saisie par l'utilisateur. On utilisera pour cela la fonction `read_line`.
3. Refaire la question précédente en utilisant cette fois la fonction `Scanf.scanf`.
4. Répondre aux questions suivantes en vous aidant éventuellement de la documentation en ligne de OCaml :
 - a) Quels sont les fichiers créés par `ocamlc` ?
 - b) Que vous inspire la première ligne d'un exécutable créé par `ocamlc` ?
 - c) Quels sont les fichiers créés si on compile avec `ocamlopt` au lieu de `ocamlc` ?
 - d) Quel est la différence entre `ocamlc` et `ocamlopt` ?

Exercice 3 : Premiers pas dans Emacs

Pour cet exercice, on utilisera l'éditeur de texte Emacs et son support du langage OCaml. Une fois un fichier `.ml` ouvert, on peut lancer le mode OCaml en faisant `Ctrl+C` puis `Ctrl+S`. On peut alors faire interpréter la ligne sous le curseur en faisant `Ctrl+C`, `Ctrl+E`. On peut aussi réinterpréter tout le fichier en faisant `Ctrl+C`, `Ctrl+B`.

1. Définir une fonction `cube` qui retourne le cube d'un nombre flottant passé en argument.

2. Définir une fonction `volume` qui retourne le volume d'une sphère étant donné son rayon `r`.
3. Définir une fonction `surface_et_volume` qui retourne la surface et le volume d'une sphère étant donné son rayon `r`.

Exercice 4 : Les tests

1. Écrire une fonction `affiche_pair` telle que `affiche_pair n` affiche « pair » si l'entier `n` est pair, et « impair » sinon.
2. Écrire une fonction `pair` qui renvoie `true` sur la donnée d'un entier pair, et `false` sinon.
3. Écrire une fonction `max_3` qui renvoie le plus grand des 3 éléments passés en argument. Quel est son type ?
4. Écrire une fonction qui, sur la donnée de trois entiers `a`, `b` et `h`, teste s'il existe un triangle rectangle d'hypoténuse `h` et dont les autres cotés ont pour mesures `a` et `b`.
5. Écrire une fonction qui, sur la donnée de trois entiers `a`, `b` et `c`, teste s'il existe un triangle rectangle dont les cotés ont pour mesures `a`, `b` et `c`.

Exercice 5 : Le filtrage

1. Écrire une fonction `a_deux_elts` telle que `a_deux_elts l` renvoie `true` si la liste `l` a exactement deux éléments, et `false` sinon.
2. Écrire une fonction `third` qui, sur la donnée d'un tuple à 3 éléments `t`, renvoie la troisième valeur stockée dans ce tuple. Par exemple, si `t` vaut `(1, "2", '3')`, le résultat de `third t` sera `'3'`.

Exercice 6 : Premières fonctions récursives

1. Écrire une fonction récursive qui, étant données `x` et `n` deux entiers positifs, calcule x^n via :

$$\begin{aligned}
 x^0 &= 1 \\
 x^n &= x \times x^{n-1} && \text{pour tout } n > 0
 \end{aligned}$$

2. Tester cette fonction pour $x = 3$ et $n = 1000000$. Que constatez-vous ?
3. Refaire la question précédente en se basant cette fois sur :

$$\begin{aligned}
 x^0 &= 1 \\
 x^{2n} &= x^n \times x^n && \text{pour tout } n > 0 \\
 x^{2n+1} &= x \times x^n \times x^n && \text{pour tout } n \geq 0
 \end{aligned}$$