TP 1 d'option informatique - MPSI - Langage OCaml - 02/01/2024

Toutes les fonctions doivent être testées, même lorsque ce n'est pas précisé.

OCaml sur les ordinateurs du lycée. L'IDE installé sur les ordinateurs du lycée s'appelle `WinCaml`, le raccourci se trouve dans le même dossier que `Spyder`. Par défaut, cet IDE lance `Caml Light` et non `OCaml`. Vous devez donc :

- Ouvrir un nouveau fichier et l'enregistrer sur le disque `U:` ou une clé USB.
- Sélectionner `OCaml` dans le menu `CamlTop`.
- Arrêter Caml Light avec l'icône ☒ puis démarrer OCaml avec l'icône ↑.

Ces étapes sont à refaire à chaque fois que vous redémarrez `WinCaml`. Notez les quelque part ! Pour exécuter un code sous WinCaml, utiliser l'icône ↑. ou bien le raccourci clavier `Ctrl + Entrée`.

Exercice 1.

1. Récupérer le fichier annexe disponible sur la page du cours

   `https://informatique-lhp.fr/opt-mpsi.html`

   Prévoir la réponse d'Ocaml lors de l'évaluation de chaque ligne du fichier, puis le vérifier sur machine.
2. Sans utiliser de forçage de type, définir des fonctions ayant pour types :

   `int -> float`, `int*int -> bool`, `int -> int -> bool`,
   `int -> int -> int`, `'a -> 'a -> 'a`.

Exercice 2. Racines d'un polynôme du second degré

Écrire une fonction qui prend en entrée trois réels a, b et c, et renvoie les deux réels racines du polynôme $aX^2 + bX + c$. Dans le cas où le polynôme a une seule racine `r`, votre fonction renverra le couple `(r,r)`. S'il n'y a pas de racine ou une infinité de racines, votre fonction déclenchera une erreur. On testera avec les valeurs du tableau.

a	b	c
1.0	2.0	1.0
1.0	2.0	1.1
1.0	-1.0	-1.0
0.0	0.0	0.0
0.0	0.0	1.0
0.0	1.0	-5.0

Exercice 3. Typage de fonctions

```
let fct1 a b c = a (b c);;   let fct2 a b c = (a b) c;;   let fct3 a b c = a b c;;
```

1. Donner le type de ces trois fonctions, puis vérifier avec OCaml.
2. Trouver des valeurs de `a`, `b` et `c` pouvant être données en argument de `fct1`.
3. Même question pour `fct2` et `fct3`.
4. Trouver une fonction de type `('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c`

Exercice 4. Fonctions avec fonctions en paramètre

1. Écrire une fonction « `fct1: (float -> float) -> float` » qui prend en entrée une fonction f et renvoie $\dfrac{f(5)+f(-5)}{2}$.
2. Écrire une fonction « `fct2: (float -> float) -> float -> float` » qui prend en entrée une fonction f ainsi qu'un flottant x, et renvoie $f(x)^2$.
3. Soit `fct3` la fonction qui prend en entrée une fonction « `f : float -> float` » et renvoie `f`^{2}. Quel est le type de `fct3` ? Écrire `fct3`.

4. Écrire une fonction `somme` qui prend en entrée deux fonctions « `f: int -> int` » et « `g: int -> int` », et renvoie la somme de ces deux fonctions.
5. Écrire une fonction récursive `iteree` qui prend en entrée une fonction « `f: 'a -> 'a` » ainsi qu'un entier $n \geqslant 0$ et renvoie la fonction itérée $f^{(n)} : x \mapsto f \circ \ldots \circ f(x)$. Testez votre fonction.

Exercice 5. Fonctions récursives

1. Sur feuille, prévoir le type de `add` et `f` ainsi que les résultats des appels à ces fonctions. Vérifier votre réponse avec OCaml.

```
let rec add n m =
  if n = 0 then m else add (n-1) (m+1);;
```

```
add 0 10;;
add 5 10;;
add 5 (-1);;
add (-4) (-1);;
```

```
let rec f x = f x;;
f 0;;
```

2. Sans définir de fonction intermédiaire, écrire une fonction récursive de complexité $\mathcal{O}(n)$ qui prend en entrée deux entiers (p, n), et renvoie $S_n = \sum_{k=1}^{n} p^k$.
   Indication : exprimer S_n en fonction de S_{n-1}, d'une addition et d'une multiplication (mais pas de mise à la puissance).
3. Le symétrique d'un entier positif $n \geqslant 0$ est obtenu en écrivant les chiffres de n à l'envers. Par exemple, le symétrique de 123 est 321. Sans utiliser de conversion de type, écrire une fonction récursive qui renvoie le symétrique de l'entier donné en paramètre. Votre fonction devra être de complexité linéaire en le nombre de chiffres.
4. Écrire une fonction récursive qui prend en entrée deux entiers $p \geqslant 0$ et n, et renvoie $\sum_{k=1}^{n} k^p$.

Exercice 6. Notes de colles

L'un des colleurs de MPSI, monsieur Koufandrelli, est bien embêté : il a perdu les notes de sa dernière séance de colles lors de laquelle il avait interrogé 6 élèves. Les notes sont des entiers naturels inférieurs ou égaux à 10. Heureusement, il se souvient de $n \in \mathbb{N}$ la somme des 6 notes. Par exemple, avec $n = 37$, une répartition possible des notes est la suivante :

$$7/10 \qquad 6/10 \qquad 2/10 \qquad 10/10 \qquad 5/10 \qquad 7/10$$

1. Écrire une fonction qui prend en entrée l'entier $n \in \mathbb{N}$ et renvoie u_n le nombre de répartitions possibles pour les notes. Par exemple :

$$u_0 = 1, \qquad u_1 = 6, \qquad u_2 = 21, \qquad u_{37} = 61242, \qquad u_{60} = 1, \qquad u_{61} = 0.$$

Exercice 7. Ensemble défini récursivement

Soit $A \subset \mathbb{N}$ le plus petit ensemble tel que :

- ⋆ A contient 0 et 1
- ⋆ Si $a \in A$ alors $2a \in A$ et $5a + 6 \in A$

Écrire une fonction « `contient: int -> bool` » qui renvoie `true` lorsque l'entier donné en paramètre appartient à A et `false` sinon. Par exemple :

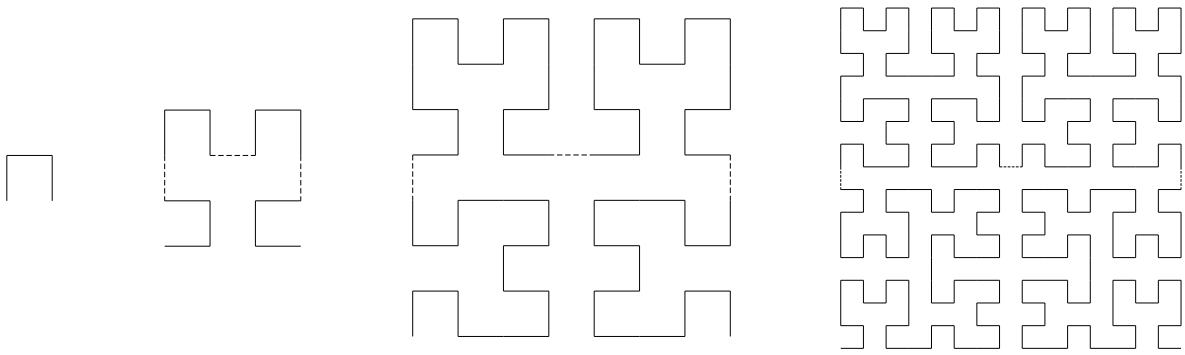
n	0	1	-1	126	216	472	1926	2345
contient n	true	true	false	true	false	true	true	false

Exercice 8. Décomposition en facteurs premiers

Écrire une fonction récursive « `print_decomp: int -> unit` » qui affiche la décomposition d'un entier $n \in \mathbb{N}$ en facteurs premiers. Par exemple :

n	`-1`	`0`	`1`	`2`	`4`	`2520`	`790614`
Affichage	Erreur	`0`	`1`	`2`	`2^2`	`2^3 3^2 5 7`	`2 3^3 11^4`

Exercice 9. Courbe de Hilbert



H_1 H_2 H_3 H_4

La courbe de Hilbert H_n est définie par récurrence sur $n \in \mathbb{N}^*$ (voir les figures ci-dessus) :

- ⋆ H_1 est composée d'un segment horizontal et de deux segments verticaux.
- ⋆ Pour tout $n \geqslant 1$, on construit H_{n+1} à partir de quatre copies de H_n :
  - Les deux premières copies sont placées en haut à gauche et en haut à droite de la figure.
  - Les deux autres copies subissent une rotation, puis sont placées en bas à gauche et en bas à droite de la figure.
  - Les quatre copies de H_n sont reliées à l'aide de trois segments (représentés en pointillés).

1. Écrire une fonction qui prend en entrée l'entier n et affiche H_n dans la console. Les traits verticaux seront remplacés par le caractère `'|'` et les traits horizontaux par `'_'`. Dans cette question, vous n'avez pas le droit d'utiliser de liste ou de tableau. Par exemple, avec $n = 3$, on obtient :

```
 _   _   _   _
| |_| | | | |_| |
|_   _| |_   _|
 _| |_ _ _| |_
|  _ _   _ _  |
|_|  _| |_  |_|
 _  |_   _|  _
| |_ _| |_ _| |
```

Exercices à rendre au plus tard le 15/02/2024 à 20h

Les exercices doivent m'être envoyés à l'adresse `rendu.opt.mpsi1@gmx.fr` ou `rendu.opt.mpsi2@gmx.fr` en fonction de votre classe. Attention, ce n'est pas la même adresse mail que pour l'informatique commune.

Exercice 10. Fonctions élémentaires

1. Écrire une fonction « `signe_int: int -> string` » qui prend en entrée un entier n et renvoie la chaîne de caractères `"positif"` lorsque n est strictement positif, `"negatif"` lorsque n est strictement négatif et déclenche une erreur lorsque n vaut 0.
2. De même, écrire une fonction « `signe_float: float -> string` ». En particulier, testez l'évaluation de « `signe_float 0.5` » et de « `signe_float 0.` ».
3. À l'aide d'un `if then else`, écrire une fonction « `abs_float : float -> float` » qui calcule la valeur absolue d'un flottant.

Exercice 11. Palindromes

Un ***palindrome*** est une chaîne de caractères dont la suite des caractères est la même lorsqu'on la lit de gauche à droite et quand on la lit de droite à gauche. Par exemple, `"kayak"` et `"radar"` sont des palindromes.

1. Sans définir de fonction intermédiaire, écrire une fonction récursive « `est_palindrome: string -> bool` » qui indique si la chaîne de caractères donnée en argument est un palindrome. On utilisera les fonctions `String.length` et `String.sub` (voir l'aide en ligne d'OCaml).

Exercice 12. Épelons les nombres

Écrire une fonction récursive « `epeler: int -> string` » qui prend en entrée un entier n et construit la chaîne de caractères contenant les chiffres de n écrits en toutes lettres séparés par des tirets. Par exemple, « `epeler 483` » doit s'évaluer en `"quatre-huit-trois"`. En OCaml, la concaténation de deux chaînes de caractères `s1`, `s2` s'écrit « `s1 ^ s2` » Dans le cas où $n < 0$, votre fonction déclenchera une erreur.

Exercice 13. Séquences régulières (exercice facultatif)

Soit $n \in \mathbb{N}^*$ un entier et $u = (x_1, \dots, x_n) \in \mathbb{N}^n$ un n-uplet. On dit que u est une ***séquence régulière de longueur*** n si les trois conditions suivantes sont respectées :

$$x_1 = 1, \qquad x_1 \leqslant x_2 \leqslant \ldots \leqslant x_n, \qquad \forall k \in [\![2; n]\!] : x_k \leqslant x_1 + x_2 + \ldots + x_{k-1}.$$

1. À la main, déterminer le nombre de séquences régulières de longueurs $n = 1, 2, 3, 4$. Justifier.
2. Écrire une fonction « `seq_reg: int -> int` » qui renvoie le nombre de séquences régulières de longueur n où n est donné en entrée.