

# Programmation fonctionnelle (HAI102I)

Licence Informatique  
Département Informatique  
Faculté des Sciences de Montpellier  
Université de Montpellier



---

## Examen du 10 janvier 2022

Le seul document autorisé est une feuille A4 recto-verso. L'examen dure 2h. Le barème est donné à titre indicatif. Le sujet comporte 2 pages et il y a 3 exercices.

Dans ce qui suit, vous avez le droit d'utiliser les fonctions des questions précédentes même si vous n'êtes pas parvenu à les écrire. Dès qu'une fonction prendra plusieurs arguments (au moins deux), vous écrirez la fonction sous sa forme curryfiée.

### Exercice 1 (7 pts)

On considère les nombres de Fermat, qui sont des nombres qui peuvent s'écrire sous la forme  $2^{2^n} + 1$ , avec  $n$  un entier naturel. On notera  $F_n$  le nombre de Fermat de rang  $n$  égal à  $2^{2^n} + 1$ . Les premiers nombres de Fermat sont  $F_0 = 3$ ,  $F_1 = 5$ ,  $F_2 = 17$ ,  $F_3 = 257$ , etc.

1. Écrire la fonction `pow`, qui étant donnés deux entiers  $x$  et  $y$ , calcule  $x^y$ .  
Pour ce faire, vous utiliserez la fonction `x ** y`, qui réalise bien  $x^y$  mais où `x` et `y` doivent être des flottants. Il vous faudra également utiliser les fonctions de conversion `float_of_int` (de `int` vers `float`) et `int_of_float` (de `float` vers `int`).
2. Écrire la fonction `fermat_nbr`, qui étant donné un entier  $n$ , rend  $F_n$ , c'est-à-dire le nombre de Fermat de rang  $n$ .
3. Écrire la fonction `is_fermat`, qui étant donné un entier  $p$ , teste si  $p$  est un nombre de Fermat, c'est-à-dire s'il existe un entier naturel  $n$  tel que  $p = F_n$ .  
Pour ce faire, vous écrirez, au préalable, la fonction `is_fermat_rec`, qui étant donnés deux entiers  $p$  et  $n$ , teste s'il existe un nombre de Fermat égal à  $p$  de rang supérieur ou égal à  $n$ . Par exemple, `(is_fermat_rec 15 2)` s'évalue en `false` car  $F_2 = 17 > 15$ .
4. Écrire la fonction `fermat_list`, qui étant donné un entier  $n$ , rend la liste des  $n$  premiers nombres de Fermat (l'ordre de cette liste est libre et peut être croissant ou décroissant sur les nombres de Fermat).

### Exercice 2 (7 pts)

On considère le type somme `my_string`, représentant les chaînes de caractères et défini de la façon suivante :

```
type my_string =  
  | Empty  
  | Cons of char * my_string
```

- Écrire la fonction `to_my_string`, qui étant donnée une chaîne de caractères `s` de type `string` (type prédéfini d'OCaml), rend la valeur de type `my_string` correspondant à `s`.  
Exemples d'évaluation :
  - `(to_my_string "")` s'évalue en `Empty`.
  - `(to_my_string "L1")` s'évalue en `Cons ('L', Cons ('1', Empty))`.
 Pour ce faire, vous aurez besoin des 3 fonctions suivantes sur le type `string` :
  - `(String.length s)` rend la taille de la chaîne de caractères `s`.
  - `(String.get s i)` rend le caractère à l'indice `i` dans la chaîne de caractères `s`.
  - `(String.sub s pos len)` rend la sous-chaîne de la chaîne de caractères `s` qui commence à la position `pos` et a la longueur `len`.
- Écrire la fonction `from_my_string`, qui étant donnée une chaîne de caractères `s` de type `my_string`, rend une chaîne de caractères de type `string` correspondant à `s`.  
Exemples d'évaluation :
  - `(from_my_string Empty)` s'évalue en `"`.
  - `(from_my_string (Cons ('L', Cons ('1', Empty))))` s'évalue en `"L1"`.
 Pour ce faire, vous aurez besoin de la fonction `Char.escaped`, qui étant donné un caractère `c`, rend la chaîne de caractères (de type `string`) contenant `c`. Par exemple, `(Char.escaped 'L')` s'évalue en `"L"`.
- Écrire la fonction `my_length`, qui étant donnée une chaîne de caractères `s` de type `my_string`, rend la longueur de `s` (le nombre de caractères).  
Exemples d'évaluation :
  - `(my_length Empty)` s'évalue en `0`.
  - `(my_length (Cons ('L', Cons ('1', Empty))))` s'évalue en `2`.
- Écrire la fonction `my_get`, qui étant donnés une chaîne de caractères `s` de type `my_string` et un entier `i`, rend le caractère d'indice `i` dans `s` (le premier caractère de `s` est d'indice 0 si `s` n'est pas `Empty`). S'il n'y a pas de caractère d'indice `i` dans `s`, vous devrez lever l'exception `Invalid_argument` avec le message de votre choix.  
Exemples d'évaluation :
  - `(my_get (Cons ('L', Cons ('1', Empty))) 1)` s'évalue en `'1'`.
  - `(my_get (Cons ('L', Cons ('1', Empty))) 2)` lève une exception.

### Exercice 3 (7 pts)

- Écrire la fonction `map_sum`, qui étant données une fonction `f` et une liste `[a1; ...; an]`, réalise le calcul  $(f\ a1) + \dots + (f\ an)$  en utilisant uniquement les accesseurs sur les listes `List.hd` et `List.tl`.  
Quelle est la signature de cette fonction ?
- Réécrire la fonction `map_sum` mais en utilisant le filtrage sur les listes (utilisation de `match` ou de `function`).
- Réécrire la fonction `map_sum` mais en utilisant uniquement l'itérateur `List.fold_left`. On rappelle que `(List.fold_left f init [a1; ...; an])` est équivalent à l'expression `(f (... (f (f init a1) a2) ...)) an`.