

## 5 Petits problèmes

### Exercice 5.1 (Crible d'Ératosthène)

Le *crible d'Ératosthène* est un algorithme simple qui permet de trouver tous les nombres premiers inférieurs à un certain entier naturel donné  $N$ . L'algorithme procède par éliminations :

- Au départ, tout nombre à partir de 2 est supposé premier.
- Ensuite, pour chaque entier qui est premier, ses multiples ne sont pas premiers et sont donc éliminés.

Écrire la fonction `eratosthenes` qui appliquée à un entier  $n$  strictement supérieur à 1 donne la liste de tous les entiers premiers jusqu'à  $n$ .

*Exemple d'application :*

```
# eratosthenes 30 ;;  
- : int list = [2; 3; 5; 7; 11; 13; 17; 19; 23; 29]
```

### Exercice 5.2 (Liste de couples)

1. Écrire la fonction `combine` ayant les spécifications suivantes :

```
val combine : 'a list -> 'b list -> ('a * 'b) list
```

Transform a pair of lists into a list of pairs : `combine [a1;...;an] [b1;...;bn]` is `[(a1,b1);...;(an,bn)]`. Raise `Invalid_argument` if the two lists have different lengths. Not tail-recursive.

*Exemple :*

```
# combine [1; 2; 3] ['a'; 'b'; 'c'];;  
- : (int * char) list = [(1, 'a'); (2, 'b'); (3, 'c')]
```

2. Écrire la fonction `split` ayant les spécifications suivantes :

```
val split : ('a * 'b) list -> 'a list * 'b list
```

Transform a list of pairs into a pair of lists : `split [(a1,b1); ...; (an,bn)]` is `([a1; ...; an], [b1; ...; bn])`. Not tail-recursive.

*Exemple :*

```
# split [(1,'a'); (2,'b'); (3,'c')];;  
- : int list * char list = ([1; 2; 3], ['a'; 'b'; 'c'])
```

### Exercice 5.3 (Bonus : Suite)

Soit la suite suivante :

<i>ligne 0 :</i>	1	contient 1 "1"
<i>ligne 1 :</i>	11	contient 2 "1"
<i>ligne 2 :</i>	21	contient 1 "2" suivi de 1 "1"
<i>ligne 3 :</i>	1211	contient 1 "1" suivi de 1 "2" suivi de 2 "1"
<i>ligne 4 :</i>	111221	...
<i>ligne 5 :</i>	312211	
<i>ligne 6 :</i>	13112221	
...		

Écrire une fonction `sequence` qui retourne la  $n^{\text{ème}}$  ligne de cette suite, sous forme d'une liste d'entiers.

*Exemple d'application :*

```
# sequence 5 ;;  
- : int list = [3; 1; 2; 2; 1; 1]  
# sequence 0 ;;  
- : int list = [1]
```