

★ **Exercice 1:** Code mystère.

- ▷ **Question 1:** Calculez les valeurs renvoyées par la fonction  $f$  pour  $n$  variant entre 1 et 5.
- ▷ **Question 2:** Quelle est la fonction mathématique vue en cours que  $f()$  calcule ?
- ▷ **Question 3:** Quelle est la complexité algorithmique du calcul ?

```

1 def g(n:Int, a:Int, b:Int):Int = {
2   if (n == 0) {
3     return a;
4   } else {
5     return g(n-1,b,a+b);
6   }
7 }
8 def f(n:Int):Int = {
9   return g(n,0,1);
10 }

```

Notez que tout le travail est fait par la fonction  $g$ , et la fonction  $f$  ne sert qu'à donner une valeur initiale aux arguments  $a$  et  $b$ , qui servent d'accumulateur. Il s'agit là d'une technique assez classique en récursivité.

★ **Exercice 2:** Soit le type **Chaine** muni des opérations suivantes :

$$\left\{ \begin{array}{l} \text{chvide} : \emptyset \mapsto \text{Chaine} \\ \text{premier} : \text{Chaine} \mapsto \text{Caractère} \quad (\text{défini ssi la chaîne n'est pas vide}) \\ \text{reste} : \text{Chaine} \mapsto \text{Chaine} \quad (\text{défini ssi la chaîne n'est pas vide}) \\ \text{adj} : \text{Chaine} \times \text{Caractère} \mapsto \text{Chaine} \end{array} \right.$$

Écrire les fonctions suivantes. Vous préciserez les préconditions nécessaires.

- ▷ **Question 1:**  $\text{longueur} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \mathbb{N} \\ \text{retourne le nombre de lettres composant la chaîne} \end{array} \right.$
- ▷ **Question 2:**  $\text{est\_membre} : \left\{ \begin{array}{l} \text{Chaine} \times \text{caractère} \mapsto \text{booléen} \\ \text{retourne VRAI ssi le caractère fait partie de la chaîne} \end{array} \right.$
- ▷ **Question 3:**  $\text{occurrence} : \left\{ \begin{array}{l} \text{Chaine} \times \text{caractère} \mapsto \mathbb{N} \\ \text{retourne le nombre d'occurrences du caractère dans la chaîne} \end{array} \right.$
- ▷ **Question 4:**  $\text{tous\_différents} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{booléen} \\ \text{retourne VRAI ssi tous les membres de la chaîne sont différents} \end{array} \right.$
- ▷ **Question 5:**  $\text{supprime} : \left\{ \begin{array}{l} \text{Chaine} \times \text{caractère} \mapsto \text{Chaine} \\ \text{retourne la chaîne privée de la première occurrence du caractère.} \end{array} \right.$   
Si le caractère ne fait pas partie de la chaîne, celle-ci est inchangée.
- ▷ **Question 6:**  $\text{deuxieme} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{caractère} \\ \text{retourne le deuxième caractère de la chaîne} \end{array} \right.$
- ▷ **Question 7:**  $\text{dernier} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{caractère} \\ \text{retourne le dernier caractère de la chaîne} \end{array} \right.$
- ▷ **Question 8:**  $\text{saufdernier} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{Chaine} \\ \text{retourne la chaîne privée de son dernier caractère} \end{array} \right.$
- ▷ **Question 9:**  $\text{nieme} : \left\{ \begin{array}{l} \text{Chaine} \times \mathbb{N} \mapsto \text{caractère} \\ \text{retourne le nieme caractère de la chaîne} \end{array} \right.$
- ▷ **Question 10:**  $\text{npremiers} : \left\{ \begin{array}{l} \text{Chaine} \times \mathbb{N} \mapsto \text{Chaine} \\ \text{retourne les } n \text{ premiers caractères de la chaîne} \end{array} \right.$
- ▷ **Question 11:**  $\text{nderniers} : \left\{ \begin{array}{l} \text{Chaine} \times \mathbb{N} \mapsto \text{Chaine} \\ \text{retourne les } n \text{ derniers caractères de la chaîne} \end{array} \right.$
- ▷ **Question 12:**  $\text{retourne} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{Chaine} \\ \text{retourne la chaîne lue en sens inverse} \end{array} \right.$
- ▷ **Question 13:**  $\text{concat} : \left\{ \begin{array}{l} \text{Chaine} \times \text{Chaine} \mapsto \text{Chaine} \\ \text{retourne les deux chaînes concaténées} \end{array} \right.$
- ▷ **Question 14:**  $\text{min\_ch} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{caractère} \\ \text{retourne le caractère le plus petit de la chaîne} \end{array} \right.$

On considère l'ordre lexicographique, et on suppose l'existence d'une fonction  $\min(a,b)$ .

- ▷ **Question 15:**  $\text{croissante} : \left\{ \begin{array}{l} \text{Chaine} \mapsto \text{booléen} \\ \text{retourne si la chaîne est croissante (dans l'ordre lexicographique)} \end{array} \right.$

- ▷ **Question 16:**  $n\text{naturels} : \begin{cases} \mathbb{N} \mapsto \text{Chaîne} \\ \text{retourne une chaîne formée des } n \text{ premiers entiers naturels} \end{cases}$   
 Dans un premier temps, on construira  $\{n, n-1, n-2, \dots, 3, 2, 1\}$  avant de construire  $\{1, 2, 3, \dots, n\}$ .
- ▷ **Question 17:**  $\text{palindrome} : \begin{cases} \text{Chaîne} \mapsto \text{booléen} \\ \text{retourne VRAI si la chaîne est un palindrome} \end{cases}$   
 Un palindrome se lit indifféremment de droite à gauche ou de gauche à droite. Exemple : « Esope reste et se repose ». On peut ignorer les espaces.
- ▷ **Question 18:**  $\text{anagramme} : \begin{cases} \text{Chaîne} \times \text{Chaîne} \mapsto \text{booléen} \\ \text{retourne VRAI si les chaînes sont des anagrammes l'une de l'autre} \end{cases}$   
 Une anagramme d'un mot est un autre mot obtenu en permutant les lettres. Exemples : «chien» et «niche»; «baignade» et «badinage»; «Séduction», «éconduits» et «on discute».
- ▷ **Question 19:**  $\text{union} : \begin{cases} \text{Chaîne} \times \text{Chaîne} \mapsto \text{Chaîne} \\ \text{retourne une chaîne formée de toutes les lettres de } ch1 \text{ et } ch2, \text{ sans doublons} \end{cases}$   
 On peut supposer dans un premier temps que  $ch1$  et  $ch2$  ne contiennent pas de doublons.
- ▷ **Question 20:**  $\text{difference} : \begin{cases} \text{Chaîne} \times \text{Chaîne} \mapsto \text{Chaîne} \\ \text{retourne toutes les lettres de } ch1 \text{ ne faisant pas partie de } ch2 \end{cases}$