

★ 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 int g(int n, int a, int b) {
2   if (n == 0) {
3     return a;
4   } else {
5     return g(n-1, b, a+b);
6   }
7 }
8 int f(n) {
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 *Chaîne* muni des opérations suivantes :

$$\begin{cases}
 \text{chvide} : \emptyset \mapsto \text{Chaîne} \\
 \text{premier} : \text{Chaîne} \mapsto \text{Caractère} \quad (\text{défini ssi la chaîne n'est pas vide}) \\
 \text{reste} : \text{Chaîne} \mapsto \text{Chaîne} \quad (\text{défini ssi la chaîne n'est pas vide}) \\
 \text{adj} : \text{Chaîne} \times \text{Caractère} \mapsto \text{Chaîne}
 \end{cases}$$

Écrire les fonctions suivantes. Vous préciserez les préconditions nécessaires.

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

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

- ▷ Question 15: *croissante* :  $\begin{cases} \text{Chaîne} \mapsto \text{booléen} \\ \text{retourne si la chaîne est croissante (dans l'ordre lexicographique)} \end{cases}$
- ▷ Question 16: *nnaturels* :  $\begin{cases} \mathbb{N} \mapsto \text{Chaîne} \\ \text{retourne une chaîne formée des } n \text{ premiers entiers naturels} \end{cases}$   
 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: *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: *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: *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: *difference* :  $\begin{cases} \text{Chaîne} \times \text{Chaîne} \mapsto \text{Chaîne} \\ \text{retourne toutes les lettres de } ch1 \text{ ne faisant pas partie de } ch2 \end{cases}$