

Exercice 1 :

Le but de cet exercice est d'observer les différences entre les tableaux de caractères (type `char[]`) et les chaînes de caractères (type `String`). On rappelle que si `s` est une variable de type `String` alors on peut utiliser les méthodes suivantes :

- `s.charAt(i)` retourne le caractère à l'indice `i` dans `s` (le premier indice étant 0 comme pour les tableaux)
- `s.length()` retourne le nombre de caractères présents dans `s`
- Si `s2` est aussi une variable de type `String` alors `s + s2` retourne une chaîne de caractères égale à la concaténation de `s` et `s2`. De plus, certains types peuvent être convertis de façon implicite vers un `String`. Par exemple `"test"+"1"`, `"test"+"1"` et `"test"+1` donneront tous la chaîne `"test1"`.
- Si `s2` est aussi une variable de type `String` alors `s.equals(s2)` retourne un `boolean` indiquant si les deux chaînes `s` et `s2` sont identiques.

1. Écrire une fonction qui prend en entrée un tableau de caractères et qui affiche ce tableau.
2. Même question pour une chaîne de caractères.
3. Écrire une fonction qui prend en entrée un tableau de caractères `tab` et deux caractères `oldChar` et `newChar` et qui remplace dans `tab` toutes les occurrences de `oldChar` par `newChar`.
4. Est-il possible de faire la même chose pour un `String`? Dans la négative, proposer une alternative et déterminer combien de `String` seront créés.

Exercice 2 :

Un étudiant, souhaitant écrire une fonction qui compte le nombre de `e` dans une chaîne de caractères, a produit le code suivant :

```
1 public static int nombreDe_e(String s) {
2     resultat = 0;
3     i = 0;
4     char e;
5     while (i < s.length) {
6         if (s[i] == e) resultat + 1;
7         i ++ 1;
8     }
9 }
```

Corriger son code.

Exercice 3 :

1. Écrire un algorithme qui retourne la position de la n -ième occurrence ($n \geq 1$) d'un caractère `c` dans une chaîne de caractères. Par exemple, pour `c='r'` et $n = 3$, la position de la 3-ième occurrence de 'r' dans la chaîne "Crachin breton récurrent" est 15 (on rappelle

que les positions commencent à 0 dans les chaînes). S'il n'existe pas de n -ième occurrence du caractère c , alors l'algorithme doit retourner la valeur -1.

2. Écrire un algorithme qui retourne le nombre d'occurrences de mots deux lettres dans une chaîne de caractères. Dans cette chaîne, les mots sont séparés uniquement par un espace ou par un caractère de ponctuation parmi les suivants : la virgule(','), le point('.'), le point d'exclamation('!'), le point d'interrogation('?'), le deux points(':') et le point-virgule(';'). Par exemple, la phrase "Où es tu ?" comporte trois mots de deux lettres et la phrase "Le si et le la en clé de fa." en comporte huit. Les chaînes commencent toujours par un caractère de l'alphabet (donc pas par un caractère de ponctuation) et se terminent par un point, point d'exclamation ou point d'interrogation.