

Le but de ce TP est de se familiariser avec les chaînes de caractères. On considérera que toutes les chaînes de caractères sont uniquement constituées de caractères ASCII (pas de lettres accentuées par exemple). Vous pourriez éventuellement avoir besoin de vous aider d'une table ASCII, facilement trouvable sur Internet.

Exercice 1 : Manipulation de base

1.a] Écrire une fonction `occurrence` qui prend en paramètre une chaîne de caractères `s` et un caractère `c`, et qui renvoie le nombre de fois où le caractère `c` apparaît dans `s`. Par exemple, `occurrence("Bonjour", 'o')` doit renvoyer 2.

1.b] Écrire une fonction `estPalindrome` qui prend en paramètre une chaîne de caractères `s`, et renvoie un booléen indiquant si `s` est un palindrome ou non. Un palindrome est une chaîne de caractères qui se lit de façon identique de gauche à droite ou de droite à gauche, comme par exemple `radar`, `kayak` ou `ressasser`.

1.c] Écrire une fonction `extraireSousChaine` qui prend en paramètre une chaîne de caractères `s`, et deux indices de début `d` et de fin `f`. La fonction doit renvoyer une nouvelle chaîne de caractères, constituée des caractères compris entre les indices `d` (inclus) et `f` (inclus). Par exemple `extraireSousChaine("Bonjour", 2, 5)` doit renvoyer `"njou"`.

1.d] Écrire une fonction `remplacer` qui prend en paramètre une chaîne de caractères `s` et deux caractères `c1` et `c2`. La fonction doit renvoyer une copie de `s`, dans laquelle tous les caractères `c1` ont été remplacés par `c2`. Par exemple, `remplacer("gramme", 'm', 'p')` doit renvoyer `"grappe"`.

1.e] (Challenge) Écrire une fonction `enumererSousChaines` qui prend en paramètre une chaîne de caractères `s`. La fonction doit afficher toutes les sous-chaînes de `s`, par ordre croissant de taille. Par exemple, `enumererSousChaines("abcd")` doit afficher `"a"`, `"b"`, `"c"`, `"d"`, `"ab"`, `"bc"`, `"cd"`, `"abc"`, `"bcd"`, `"abcd"`.

1.f] Afficher la table ASCII. On pourra faire une simple boucle `for` sur le code ASCII du caractère à afficher (de 30 à 127), et afficher des lignes comme cela (avec `'?'` à changer) :

Pour `i = 30`, la table ASCII donne le caractère : `'?'`

...

Pour `i = 127`, la table ASCII donne le caractère : `'?'`

Pourquoi certains caractères ne s'affichent pas comme les autres ?

(challenge) Et si vous commencez à `i = 0`, est-ce que ça fonctionne ? Pourquoi ? Renseignez vous sur la signification des premiers caractères ASCII. Représentent-ils des lettres ou des symboles affichables comme `'?'` ou `'.'` ?

Exercice 2 : Minuscules et majuscules

2.a] Écrire deux fonctions `estMajuscule` et `estMinuscule` qui renvoient un booléen indiquant respectivement si un caractère donné est une lettre majuscule ou minuscule. Les caractères sont représentés par des entiers, en suivant la convention de la table ASCII. On peut donc effectuer des opérations (ajouter, soustraire) ou des comparaisons sur des caractères (ex, `c >= 'A'` ou `c <= 'z'`). Vous pourriez avoir besoin d'une table ASCII pour cet exercice. Écrire ensuite une fonction `estLettre` qui renvoie un booléen indiquant si un caractère donné est une lettre.

2.b] Écrire une fonction `minusculeChar` qui prend en paramètre un caractère `c`. Si `c` est une lettre majuscule, alors la fonction renvoie la même lettre en minuscule. Sinon, la fonction renvoie `c`. Par exemple, `minusculeChar('A')` doit renvoyer `'a'`. `minusculeChar('b')` doit renvoyer `'b'`. `minusculeChar('?')` doit renvoyer `'?'`. De même, écrire une fonction `majusculeChar`.

2.c] Écrire une fonction `minusculeChaine` qui prend en paramètre une chaîne de caractères `s`, et renvoie une copie de `s` dans laquelle toutes les lettres sont transformées en minuscules. Par exemple, `minusculeChaine("l'eXamen ne sera pas TROP dur")` doit renvoyer `"l'examen ne sera pas trop dur"`. De même, écrire une fonction `majusculeChaine`.

2.d] (**Challenge**) Écrire une fonction `majusculeDebut` qui prend en paramètre une chaîne de caractères `s`, et renvoie une copie de `s` dans laquelle la première lettre de chaque mot est transformée en majuscule, et toutes les autres en minuscule. On considère que la chaîne commence toujours par une lettre, et qu'un mot est toujours précédé d'un espace. Par exemple, `majusculeDebut("l'eXamen ne sera pas TROP dur")` doit renvoyer `"L'examen Ne Sera Pas Trop Dur"`.

Exercice 3 : Comptage de mots

3.a] Écrire une fonction `comptageMot` qui prend en paramètre une chaîne de caractères `s`, et qui renvoie le nombre de mots de `s` dont la longueur vaut exactement 2. Les mots sont séparés par un ou plusieurs espaces ou signes de ponctuation. On suppose également que la chaîne `s` ne contient que des lettres, des espaces et des signes de ponctuation. Par exemple, `comptageMot("Le si et le la en cle de fa")` doit renvoyer 8.

3.b] (**Challenge**) Modifiez la fonction précédente pour qu'elle compte les mots de longueur `n`, où `n` est un paramètre de la fonction.

Exercice 4 : Conversion entre chaîne de caractères et entier

Écrire une fonction `conversionEntier` qui prend en paramètre une chaîne de caractères uniquement constituée des caractères de 0 à 9 et renvoie l'entier correspondant. Par exemple, `conversionEntier("123")` doit renvoyer l'entier 123.