TP 4 - Structure de données : chaînes de caractères

1 Premières manipulations

Dans cette partie, les manipulations sont à effectuer dans la console.

Question 1. Créer une chaîne `s1` valant `"Science"` et une chaîne `s2` valant `"sans conscience n'est que ruine de l'âme."`.

Question 2. Afficher les valeurs de `s1` et `s2`.

Question 3. Accéder aux valeurs d'indice 0, 5 et 41 dans la chaîne `s2`.

Question 4. Créer la chaîne `s` valant `"Science sans conscience n'est que ruine de l'âme."` en concaténant les chaînes `s1` et `s2`.

Question 5. Evaluer le code `s[0]="A"`, conclure.

2 Algorithmes fondamentaux

Question 6. Ecrire le code de la fonction `present` prenant en paramètre une chaîne de caractères `s` et un caractère `c`. Elle renvoie un booléen indiquant si `s` contient `c`. Ainsi l'appel `present("conscience",'o')` renvoie `True`.

Question 7. Ecrire le code de la fonction `compter` prenant en paramètre une chaîne de caractères `s` et un caractère `c`, et qui renvoie le nombre d'occurrences de `c` dans `s`. Par exemple, dans la chaîne `"conscience"`, le nombre d'occurrences de `'c'` est 3, le nombre d'occurrences de `'o'` est 1 et le nombre d'occurrences de `'a'` est 0.

On appelle **facteur** d'une chaîne une sous-chaîne contenue dans cette chaîne. Par exemple, `"conscience"` est un facteur de `s="Science sans conscience n'est que ruine de l'âme."`.

Question 8. Ecrire le code de la fonction `verifFacteur` prenant en paramètre une chaîne `s`, une chaîne `f`, supposée de taille inférieure à `s`, et un entier `i`. Elle renvoie un booléen indiquant si le facteur `f` se trouve à l'indice `i` de la chaîne `s`. Avec la chaîne `s` donnée ci-dessus, le facteur `"conscience"`, la fonction doit retourner `True` si l'indice est 13 et `False` sinon.

On pourra utiliser `assert` pour assurer que l'indice `i` soit un indice valide de `s`.

Question 9. En déduire le code de la fonction `estFacteur` prenant en paramètre deux chaînes de caractères `s` et `f`. Elle retourne un booléen indiquant si la chaîne `f` est un facteur de `s`. Par exemple, l'appel `estFacteur("bonjour","bon")` retourne `True`.

Question 10. Ecrire le code d'une fonction `miroir` prenant en paramètre une chaîne de caractères et retournant son inverse. Par exemple, l'inverse de `"jour"` est `"ruoj"`.

3 Autres exercices

Exercice 1 (**Pangramme**)**.** Un pangramme est une chaîne de caractères contenant tous les caractères de l'alphabet. On suppose que les chaînes ne comporte aucun signe diacritique (signe accompagnant une lettre : cédille, accent ...) ni aucune majuscule.

Dans le module `string`, on dispose de la chaîne de caractères `ascii_lowercase` contenant tout l'alphabet latin minuscule (sans accent). Pour pouvoir utiliser cette variable, il faut l'importer avec la commande suivante : `from string import ascii_lowercase`.

Question 11. Ecrire une fonction `estPangramme` prenant en paramètre une chaîne de caractères et retournant un booléen indiquant si la chaîne est un pangramme ou pas. Par exemple, avec la chaîne `"portez ce vieux whisky au juge blond qui fume"` on doit obtenir `True`.

Question 12. Ecrire une fonction `manquePourPangramme` prenant en paramètre une chaîne de caractères et retournant la liste des caractères manquants pour que ce soit un pangramme.

Exercice 2 (**Cybercafé**)**.** Un gérant de cybercafé souhaite savoir combien de clients ont pu entrer dans son magasin sans se voir allouer d'ordinateur parce qu'ils étaient tous occupés. On connaît la capacité en ordinateurs du cybercafé et on dispose des données des entrées et sorties des clients sous forme d'une chaîne de caractères. Par exemple, dans la chaîne `"ABCDDCEFFEBGAG"`, la première occurrence d'une lettre indique l'arrivée d'un client et la seconde occurrence sa sortie. Ainsi, si le cybercafé dispose de 3 ordinateurs, les clients D et F n'ont pas pu avoir d'ordinateurs. Pour la chaîne `"ABCDDCBEFFEGAG"` et un nombre d'ordinateur de 2, les clients C, D et F ne peuvent pas avoir d'ordinateurs.

Question 13. Ecrire une fonction `nbClientsSansOrdis` prenant en paramètre un entier `n` donnant le nombre d'ordinateurs disponibles dans le cybercafé et une chaîne de caractères `s` représentant les entrées et sorties des clients. La fonction renvoie le nombre de clients n'ayant pas eu d'ordinateur dès leur entrée. Par exemple, `nbClientsSansOrdis("ABCDDCEFFEBGAG",3)` renvoie 2 et `nbClientsSansOrdis("ABCDDCBEFFEGAG",2)` renvoie 3.

Question 14. Modifier la fonction précédente pour qu'elle renvoie un couple dont le premier terme est un entier représentant le nombre de clients n'ayant pas eu d'ordinateurs à leur arrivée, et le deuxième une liste de caractères de ceux qui ont dû attendre. Dans le premier exemple, la fonction renvoie `(2,["D","F"])` et dans le second, la fonction renvoie `(3,["C","D","F"])`.

Exercice 3 (**Mot de passe**)**.** Joseph Marchand a installé sur l'ordinateur de son amie Alice un *keylogger* (un logiciel espion qui capture les entrées du clavier) pour récupérer son mot de passe. Malheureusement, le *keylogger* a enregistré toutes les frappes sur le clavier, sans distinction de s'il s'agissait d'un mot de passe ou non. Joseph se retrouve donc avec une suite de caractères composée de lettres minuscules, majuscules, de nombres et de caractères spéciaux. Il sait juste que le mot de passe d'Alice répond aux exigences de sécurité suivantes :

- contenir au moins une minuscule (a-z) ;
- contenir au moins une majuscule (A-Z) ;
- contenir au moins un chiffre (0-9) ;
- contenir au moins un caractère spécial parmi `!"#$% &'()*+,-./:;<=>@[\]^_{|}~`
- et la taille du mot de passe est de k caractères.

Question 15. Ecrire le code de la fonction `motDePasseSecure` prenant en paramètre une chaîne de caractères représentant un mot de passe et renvoyant un booléen indiquant si la chaîne respecte tous quatre premiers critères donnés ci-dessus.

Question 16. Ecrire le code de la fonction `extractionMotsDePasse` prenant en paramètre une chaîne de caractères `ch` et un entier `k`. Elle renvoie la liste de tous les mots de passe possible de taille k présents dans la chaîne `ch`.

Exercice 4 (**Plus petit préfixe**)**.** On suppose disposer d'une liste de chaînes de caractères toutes distinctes. On souhaite mettre au point une fonction permettant de trouver le plus court préfixe de chaque mot qui ne soit préfixe d'aucun autre mot de la liste.

Question 17. Ecrire le code de la fonction `plusCourtPrefixe` prenant en paramètre une lise de chaînes de caractères distinctes et renvoyant une nouvelle liste contenant les plus courts préfixes de chacun des mots qui ne soient préfixes d'aucun autre mot. Par exemple, pour la liste `["CHAR", "AND", "BOOL", "CASE", "CATCH", "BONFIRE"]`, la fonction renvoie `["CH", "A", "BOO", "CAS", "CAT", "BON"]`.

Indication : ce problème est difficile, il faut commencer par bien analyser le problème et décomposer la solution en plusieurs petites fonctions.