
N. S. I. Term. - DS01 (jeudi 23 septembre 2021)

**Spécification et signature d'une fonction
Pré-conditions et tests****TIERS TEMPS : sauf exercices 5, 7 et 9.****Exercice 1.**

On veut écrire une fonction `maxL` qui renvoie la valeur du minimum d'une liste `L` d'entiers ou de flottants donnée en argument.

1. Décrire en une phrase ou deux le principe de l'algorithme à mettre en œuvre
2. Proposer une implémentation de la fonction `maxL`, incluant la vérification d'une pré-condition pertinente.

Exercice 2.

Écrire une fonction, `occ`, qui renvoie le nombre d'occurrences d'une valeur `v` dans une liste `L`.

On utilisera un COMPTEUR.

Exercice 3.

Écrire une fonction, `occurrences`, qui prend en argument une liste `L` et une valeur `v`, et renvoie la liste des positions (positives) auxquelles on trouve la valeur `v` dans la liste `L`.

On utilisera une liste `Lpos`, servant d'accumulateur, initialisée à une liste vide, dans laquelle on stockera les positions auxquelles on trouve la valeur `v` dans `L`.

Exercice 4.

On veut écrire une fonction `consecutifs`, qui renvoie un booléen (`True` ou `False`), selon qu'une liste `L` passée en argument contient au moins deux valeurs consécutives égales ou non.

1. Proposer un jeu de deux tests pour cette fonction.
2. Proposer une implémentation de la fonction `consecutifs`.

Exercice 5.

Écrire une fonction, `sommeL`, qui renvoie la somme des valeurs d'une liste `L`.

On utilisera un ACCUMULATEUR.

Exercice 6.

Écrire une fonction, `resultatBAC`, qui prend en argument un flottant, `moybac`, et renvoie un couple de chaînes de caractères, choisies parmi les chaînes "Recalé", "Reçu", "AB", "B", "TB", "" (chaîne vide), selon que le flottant en entrée représentant la moyenne des notes du baccalauréat à l'issue du second tour, permet d'être reçu ou recalé, et d'obtenir une mention ou non. Le premier élément du couple renvoyé indiquera si le candidat est reçu ou non, le second élément du couple renvoyé donnera la mention obtenue (une chaîne vide si le candidat est recalé ou n'obtient pas de mention).

La fonction devra comporter une précondition vérifiant si la valeur en entrée est comprise entre 0 et 20.

Exercice 7.

- Écrire une fonction `examenliste`, qui prend en argument une liste d'entiers relatifs `L`, et renvoie sous la forme d'un tuple (un triplet ici) :
 - la liste des valeurs de `L` qui sont strictement positives (sans en changer l'ordre) ;
 - la liste des valeurs de `L` qui sont strictement négatives (sans en changer l'ordre) ;
 - un booléen indiquant si la liste contient des valeurs nulles ou non.
- Proposer un jeu de tests pour la fonction `examenliste`.

Exercice 8.

On rappelle la définition d'une année bissextile (https://fr.wikipedia.org/wiki/Ann%C3%A9e_bissextile) :

Une année est bissextile (comportant 366 jours) seulement dans les deux cas suivants :

- si l'année est divisible par 4 et non divisible par 100 ;
- si l'année est divisible par 400.

Sinon, l'année n'est pas bissextile : elle a la durée habituelle de 365 jours.

Ainsi, 2021 n'est pas bissextile. L'an [2008](#) était bissextile suivant la première règle (divisible par 4 et non divisible par 100). L'an [1900](#) n'était pas bissextile car divisible par 4 et par 100 (première règle non respectée) et non divisible par 400 (seconde règle non respectée non plus). L'an [2000](#) était bissextile car divisible par 400.

- Définir une fonction, `est_bissextile`, à un paramètre, prenant un entier positif en argument, représentant une année, et renvoyant un booléen, selon que l'entier en entrée correspond à une année bissextile ou non.
- Donner la signature de cette fonction.
- Proposer un jeu de trois ou quatre tests - selon ce qui vous semble nécessaire, utilisant `assert`.

Exercice 9.

On considère la fonction `mystere` suivante :

```
def mystere(a):
    x = 0
    if a % 2 == 0:
        x = x + 1
    if a % 3 == 0:
        x = x + 1
    return x
```

- Que renvoient les appels `mystere(12)`, `mystere(13)` et `mystere(9)` ?
- Quelle est la signature de la fonction `mystere` ?
- Proposer une spécification pour la fonction `mystere`.

Exercice 10. [Mastermind](#)

On souhaite écrire une fonction, `testecombinaison`, qui prend en argument deux tuples, `t1` et `t2`, à cinq éléments, de type `str`, dont les valeurs sont dans l'ensemble {"J", "B", "R", "V", "C", "N"} (représentant les six couleurs de l'édition originale du jeu : jaune, bleu, rouge, vert, blanc, noir), et qui renvoie sous la forme d'un tuple :

- le nombre d'éléments de `t1` qui sont dans `t2` à la même place ;
- le nombre d'éléments de `t1` - hormis ceux déjà comptés comme étant à la bonne place, qui sont dans `t2`, mais à une place différente.

- Exemples d'appels à la fonction :

```
>>> testecombinaison(('J', 'B', 'V', 'R', 'J'), ('J', 'R', 'R', 'J', 'J'))
(2, 1)
>>> testecombinaison(('J', 'B', 'V', 'R', 'J'), ('J', 'B', 'V', 'B', 'J'))
(4, 0)
>>> testecombinaison(('J', 'B', 'V', 'R', 'J'), ('B', 'J', 'R', 'J', 'V'))
(0, 5)
```

- Donner deux autres exemples d'appels à la fonction, renvoyant, respectivement, (2, 3) et (0, 2).
- Proposer une implémentation de la fonction `testecombinaison(t1, t2)`.