

1 Fonctions

1. Étant données deux variables `chaine1` et `chaine2` de type `str` et une variable `x` de type entier¹, donner des expressions qui représentent :
 - a. le nombre de caractères contenus dans `chaine1`
 - b. le nombre de caractères contenus dans la chaîne la plus longue des deux
 - c. la valeur absolue de `x`
 - d. un caractère pris au hasard dans la chaîne `chaine1`, en supposant que `chaine1` est non vide
2. Soit la spécification suivante : *Écrire une fonction `somme` qui prend en paramètre 2 entiers `x` et `y` et renvoie leur somme. Par exemple :*

```
>>> somme(2, 3)
5
```

Parmi les fonctions suivantes, indiquer celles qui satisfont ou non la spécification, justifier.

<code>def somme1(x, y):</code>	<code>def somme3():</code>	<code>def somme6(x, y)</code>
<code>x + y</code>	<code>return 2 + 3</code>	<code>return y + x</code>
<code>def somme2(x, y):</code>	<code>def somme4(x, y):</code>	<code>def somme7(x, y):</code>
<code>x = 2</code>	<code>return x + y</code>	<code>return x + y</code>
<code>y = 3</code>	<code>DEF somme5(x, y):</code>	<code>return x + x</code>
<code>return x + y</code>	<code>return x + y</code>	

2 Préconditions

3. Qu'est-ce que la précondition d'une fonction ?
4. À votre avis, quelles sont les préconditions des fonctions et opérateurs Python suivants :
 - a. `sqrt` du module `math` qui renvoie la racine carrée de son paramètre
 - b. `abs` qui renvoie la valeur absolue de son paramètre
 - c. `choice` du module `random` qui renvoie un caractère tiré au hasard dans les caractères de son paramètre (on considérera uniquement le cas où le paramètre est de type `str`)
 - d. `%` (reste de la division euclidienne)
 - e. `randint` du module `random`. `randint(a, b)` renvoie un entier tiré au hasard dans l'intervalle `[a, b]`, `a` et `b` inclus.
 - f. `len` qui renvoie la longueur de la chaîne passée en paramètre (on considérera uniquement le cas où le paramètre est de type `str`)
5. Donner les préconditions des fonctions dont les spécifications sont les suivantes :
 - a. renvoie le périmètre du quadrilatère dont les longueurs des côtés (des entiers), sont passées en paramètre
 - b. renvoie la moyenne des 2 notes passées en paramètre

3 Appels de fonctions

Dans cet exercice la docstring complète n'est pas donnée car non pertinente pour la compétence travaillée.

6. Décrivez le résultat des expressions suivantes ainsi que l'état de la mémoire durant leur évaluation :

<code>def f(x:int, y:int)->int:</code>	<code>def g(x:int, y:int)->int:</code>	<code>DELTA = 1</code>
<code>return x + y + 1</code>	<code>res = x</code>	<code>def h(x:int, y:int)->int:</code>
	<code>res = res + y</code>	<code>res = x + y</code>
	<code>return res + 1</code>	<code>return res + DELTA</code>

- a. `f(2, 3+4)`
- b. `g(2, 3+4)`
- c. `h(DELTA+1, 3+4)`

¹abus de langage : la bonne formulation est "variable associée à des valeurs de type `str`".

7. Que pensez-vous du code suivant ?

```
x = 3
def f(x : int, y : int) -> int:
    return x + y + 1
def g(z : int) -> int:
    x = z**2
    return x + 12
```

4 Tests et documentation

8. Donner le verdict associé aux tests suivants :

```
def f(x : int , y : int) -> int:
    """Renvoie le reste de la division entière de x^2 par y

    Précondition : y non nul
    Exemple(s) :
    $$$ f(6, 5)
    1
    $$$ f(5, 0)
    1
    $$$ f(7, 5)
    1
    """
    return (x**2) % y
```

9. Quels tests vous semblent pertinents pour les fonctions et opérateurs Python suivants :

- `abs` qui renvoie la valeur absolue de son paramètre
 - opérateur `%`
 - `len` qui renvoie la longueur de la chaîne passée en paramètre (on considérera uniquement le cas où le paramètre est de type `str`)
10. Écrire la documentation complète de la fonction `double_abs` qui prend en paramètre un entier `x` et qui renvoie le double de la valeur absolue de `x`.

5 Écriture de fonctions

Le but de l'exercice est de traiter un problème complexe en le décomposant et en le codant avec des fonctions. Le problème est : on dispose d'un entier positif à 4 chiffres (compris entre 1000 et 9999 inclus), par ex 1234. On veut calculer une chaîne de caractères de taille 4 qui masque les chiffres des centaines et des dizaines de cet entier. Par exemple le masquage de 1234 produira la chaîne de caractères `'1**4'`².

- Questions à se poser avant d'attaquer le problème, en s'appuyant sur l'exemple de l'énoncé :
 - quelle est la donnée principale ? son type ? quels opérateurs connus s'appliquent aux données de ce type ?
 - quel est le type du résultat à calculer pour résoudre le problème ?
 - quelles opérations a-t-on besoin d'appliquer à la donnée principale pour résoudre le problème ? Dispose-t-on d'opérateurs prédéfinis qui réalisent ces opérations de manière directe ?
 - de même : quelles opérations sont nécessaires pour construire le résultat ? Dispose-t-on d'opérateurs prédéfinis qui réalisent ces opérations de manière directe ?
- Suggérer une décomposition de ce problème en sous-problème pouvant être codés par des fonctions.
- Écrire une fonction `masque` qui prend en paramètre un entier comme décrit dans l'énoncé et renvoie une chaîne de caractères correspondant à son masquage, ainsi que les fonctions intermédiaires prévues précédemment. Par exemple `masque(1234)` vaut `'1**4'`.

²On trouve cette fonctionnalité, en plus élaborée, pour masquer partiellement les numéros de carte bancaire ou de téléphone.