
ITC MPSI - TD 2 - 8 SEPTEMBRE 2025

Exercice 1 (Sur feuille)

On suppose que des variables u et v existent. Donner un bout de programme qui échange le contenu des variables u et v .

Exercice 2 (Sur feuille)

Voici un bout de code Python :

```
p = 1
for k in range(1, 11) :
    p = p * (k+1)/k
```

Écrire le contenu de p sous forme d'une expression mathématique. La calculer.

Exercice 3 (Sur feuille)

Donner les valeurs prises par les variables u et s après les instructions suivantes :

```
u = 3
s = 3
for k in range(1, 100):
    u = u+k
    s = s+u
```

Exercice 4 (Sur machine)

Soit a un réel. On définit la suite $(u_n)_n$ par $u_0 = a > 0$ et $u_{n+1} = \sqrt{(n+1)u_n}$ pour tout $n \geq 0$.

Écrire un bout de code Python qui demande à l'utilisateur d'entrer une valeur de a , et une valeur de n , et affiche le terme u_n .

Exercice 5 (Sur machine)

On pose $u_0 = 2$ et pour tout $n \in \mathbf{N}$, $u_{n+1} = \sqrt{u_n}$ et on s'intéresse à la somme $\sum_{k=0}^{100} u_k$.

Écrire un bout de code Python qui calcule cette somme.

Exercice 6 (Sur machine)

Écrire un script qui demande à l'utilisateur de saisir un entier n , puis qui calcule et affiche le résultat de la somme double suivante :

$$\sum_{1 \leq i, j \leq n} \frac{1}{i + j}$$

Exercice 7 (Facultatif)

On suppose définies 4 constantes : EQU, ISO, SCA et ERR. On s'intéresse au problème algorithmique suivant :

Entrée : 3 entiers

Sortie :

- EQU si les trois entiers peuvent correspondre aux longueurs des côtés d'un triangle équilatéral.
- ISO si les trois entiers peuvent correspondre aux longueurs des côtés d'un triangle isocèle non-équilatéral.
- SCA si les trois entiers peuvent correspondre aux longueurs des côtés d'un triangle quelconque (non-isocèle).
- ERR si les arguments ne peuvent pas correspondre aux longueurs des côtés d'un triangle.

J'ai sur ma machine environ une dizaine de fonctions qui prétendent résoudre ce problème : l'une de ces fonctions est juste, les autres sont fausses, de différentes façons.

Votre défi, si vous décidez de l'accepter, est d'écrire un jeu de tests tel que :

- L'implémentation correcte passe tous vos tests.
- Chacune des implémentations incorrectes échoue au moins l'un de vos tests.

Votre jeu de tests devra prendre la forme d'une suite d'instructions de la forme

```
assert ( triangle (<arguments> ) == <resultat > )
```

où $\langle \text{arguments} \rangle$ correspond aux arguments que vous passez à la fonction et $\langle \text{resultat} \rangle$ est le résultat attendu.