

## Colle 01

Cette première colle consiste à écrire du code Python, et à reconnaître du pseudo-code.

### Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) et renvoie la somme des valeurs de la liste ;
- Utilisez cette fonction pour en écrire une autre qui calcule la moyenne arithmétique de la liste `li` donnée en argument (i.e.,  $\sum_{j=0}^{n-1} li[j]/n$ ).

### Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour  $n$  la hauteur donnée en argument de la fonction.

```
*  
**  
***  
****  
*****
```

### QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut définir  $f(a, b) = (7 * a + b) \bmod 10$  en Python, quel(s) code(s) est/sont correct(s) ?

- `def f(a, b): return (7*a + b) % 10`
- `def f(a, b): print( (7*a + b) % 10 )`
- `def f(a, b): return (7a + b) mod 10`
- `def f(a, b): return (7*a + b) // 10`

### Pseudo-code à corriger

Le pseudo-code suivant est-il correct pour vérifier que toutes les valeurs du tableau `tab` donné en argument sont paires ? Corrigez-le s'il faut.

```
Fonction tousPairs(tab) =  
  N = longueur(tab)  
  Pour chaque i = 0 à N-1 (inclus) faire  
    Si tab[i] mod 2 != 0 Alors Renvoyer Faux  
  Fin Pour  
  Renvoyer Vrai
```

## Colle 01

Cette première colle consiste à écrire du code Python, et à reconnaître du pseudo-code.

### Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) et renvoie le produit des valeurs de la liste ;
- Utilisez cette fonction pour en écrire une autre qui calcule la moyenne géométrique de la liste `li` donnée en argument (i.e.,  $(\prod_{j=0}^{n-1} li[j])^{1/n}$ ).

### Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour  $n$  la hauteur donnée en argument de la fonction.

```
*****
****
***
**
*
```

### QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut définir  $f(a, b) = (a + 11 * b) \bmod 93$  en Python, quel(s) code(s) est/sont correct(s) ?

- `def f(a, b): return (a + 11*b) % 93`
- `def f(a, b): print( (a + 11*b) % 93 )`
- `def f(a, b): return (a + 11b) mod 93`
- `def f(a, b): return (a + 11*b) // 93`

### Pseudo-code à corriger

Le pseudo-code suivant est-il correct pour vérifier que toutes les valeurs du tableau `tab` donné en argument sont impaires ? Corrigez-le s'il faut.

```
Fonction tousImpairs(tab) =
  N = longueur(tab)
  Pour chaque i = 2 à N+1 (inclus) faire
    Si (tab[i] mod 2) != 0 Alors Renvoyer Faux
  Fin Pour
  Renvoyer Vrai
```

## Colle 01

Cette première colle consiste à écrire du code Python, et à reconnaître du pseudo-code.

### Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) qui ne contient que des flottants positifs strictement, et renvoie la somme de leurs inverses ;
- Utilisez cette fonction pour en écrire une autre qui calcule la moyenne harmonique de la liste `li` donnée en argument (i.e.,  $n / \sum_{j=0}^{n-1} \frac{1}{li[j]}$ ).

### Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour  $n$  la hauteur donnée en argument de la fonction.

```
aaaaaa
a  a
a  a
a  a
a  a
aaaaaa
```

### QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut définir  $f(a, b) = (a + 11 * b) \bmod 93$  en Python, quel(s) code(s) est/sont correct(s) ?

- `def f(a, b): return (a + 11*b) % 93`
- `def f(a, b): print( (a + 11*b) % 93 )`
- `def f(a, b): return (a + 11b) mod 93`
- `def f(a, b): return (a + 11*b) // 93`

### Pseudo-code à corriger

Le pseudo-code suivant est-il correct pour vérifier que toutes les valeurs du tableau `tab` donné en argument sont des multiples de 17 ? Corrigez-le s'il faut.

```
Fonction tousMultiples17(tab) =
  N = longueur(tab)
  Pour chaque i = 0 à N (inclus) faire
    Si (tab[i] mod 17) != 0 Alors Renvoyer Faux
  Fin Pour
  Renvoyer Vrai
```