

TP 1

Consolidation Programmation

9 septembre 2026

Créez un dossier à votre nom, et à l'intérieur un dossier `infoPSI`. Enregistrez un fichier `TP1.py` dans votre dossier `infoPSI`. Faire **tous les exercices dans ce fichier** (à chaque exercice va correspondre une fonction). Lorsque vous testez une fonction, mettre les appels à la fonction directement **dans votre programme** puis mettre les appels en commentaire avec `#` avant de passer à l'exercice suivant (**ne pas les effacer**). À la fin du TP, vous devez rendre votre fichier `TP1.py` en vous identifiant sur le site <https://cahier-de-prepa.fr/psi-lessouriau>

Important :

- **respectez les noms** qu'on vous demande de donner aux fonctions.
- testez vos fonctions pour les vérifier **avec plusieurs valeurs pertinentes!**
- **laissez en commentaire au moins un appel par fonction**
- **ne pas utiliser de liste dans ce TP**

Exercice 0 : Compte à rebours

Écrivez une fonction `rebours` qui prend en argument un entier n et affiche en ordre décroissant les entiers compris entre n et 0. Par exemple, l'exécution de `rebours(4)` doit donner :

```
4
3
2
1
0
```

Exercice 1 : Mult3pas9

Écrire une fonction `mult3pas9` qui prend en argument un entier n et renvoie `True` si n est multiple de 3 mais pas de 9, et `False` sinon.

Votre fonction ne doit comporter qu'une seule instruction (une seule ligne de code, **pas de if**).

Exercice 2 : Produit

Écrire une fonction `produit` qui lit des entiers entrés par l'utilisateur jusqu'à ce que l'utilisateur entre 0, puis qui affiche le produit des entiers non nuls entrés par l'utilisateur. Ne pas utiliser de liste! Tester votre fonction. Exemple d'exécution :

```
Entrez les nombres à multiplier en terminant par 0 :
4
3
0
Le produit vaut 12.0
```

Exercice 3 : Triangle

Écrire une fonction `triangle` qui prend en argument un entier n , puis qui affiche le triangle d'entiers de taille n . Par exemple, l'exécution de `triangle(4)` doit donner :

```
1
12
123
1234
```

Faire de même une fonction `triangle2` telle que le triangle obtenu pointe vers le bas :

```
1234
123
12
1
```

Exercice 4 : Suite

Écrire une fonction `terme` qui prend en argument un entier n et qui renvoie le terme d'indice n de la suite définie par récurrence par $u_0 = 1$, $u_1 = 2$ et $u_n = 3 * u_{n-1} + u_{n-2}$ pour $n \geq 2$. Par exemple, `terme(4)` vaut 76 et `terme(5)` vaut 251.

Exercice 5 : Premiers

Écrire une fonction `premiers` qui prend en argument un entier n , puis qui affiche les nombres premiers inférieurs ou égaux à n . Par exemple, `premiers(12)` doit afficher 2, 3, 5, 7 et 11.

Exercice 6 : Premiers2

Écrire une fonction `premiers2` qui prend en argument un entier n , puis qui affiche les n premiers nombres premiers. Par exemple, `premiers2(6)` doit afficher 2, 3, 5, 7, 11, 13.

Exercice 7 : Binaire

Écrire une fonction `binaire` qui prend en argument un entier n et qui renvoie l'écriture binaire de cet entier. Par exemple, `binaire(2)` doit renvoyer 10 et `binaire(6)` doit renvoyer 110.