

Exercices supplémentaires

La majorité des exercices proposés sont issus du site <https://pydefis.callicode.fr> qui recense tous les défis des différentes éditions du concours [c0d1ng UP](#).

Vous pouvez vous y créer un compte, pour valider les défis et progresser au Hall of Fame... mais ce n'est absolument pas obligé.

1. Désamorçage d'un explosif (I)

 Lien Capytale : **0ee2-74101**

Le découpage de nombres selon un certain nombre de chiffres est quelque chose de courant en programmation.

Pour cela, l'astuce réside en l'utilisation de la division euclidienne, par la bonne puissance de 10.

Par exemple, si on veut récupérer le chiffre des unités d'un nombre, il suffit de prendre le reste de la division euclidienne (opérateur `%` en Python) du nombre par 10. Les autres chiffres seront donnés par le quotient (opérateur `//` en Python).

Script Python

```
>>> 3748 % 10
8
>>> 3748 // 10
374
```

Pour les deux derniers chiffres, on effectuera une division euclidienne par 100, pour les trois derniers par 1000, etc.

Correction


Script Python

```
1  entree = 797114
2  U = entree // 1000
3  N = entree % 1000
4  for k in range(N):
5      U = (U*13) % 1000
6  print(U)
```

2. SW IV : Il a mis son mot de passe sur un post-it !

 Lien Capytale : [4f1a-74191](#)

Encore la division euclidienne...

 **Correction**

 Script Python


```

1  x = 1694
2  y = 1546
3  for k in range(50):
4      z = x # il est impératif de mettre x en tampon pour faire le 2e calcul avec la bonne
5  valeur de x
6      x = (z + 2*y) % 2018
7      y = (-3*z + y) % 2018
8
9  declinaison = (x - 900) / 10
10 ascension_droite = (y / 150) * 2
11
12 print(declinaison, ascension_droite)
```

3. Toc Boum

 Lien Capytale : [69bf-74184](#)

Utiliser la «brute-force» !

 **Correction**

Version basique

Où l'on affiche tous les couples possibles, et on trouve le bon en les passant en revue «à la main».

Version évoluée

On va mémoriser le meilleur couple, c'est-à-dire celui dont la différence est la plus petite. J'appelle `best_a` et `best_b` ces deux valeurs, que j'initialise à des valeurs volontairement éloignées.

Puis à chaque couple trouvé (comme dans la version basique), je compare la différence des deux nombres à celle des meilleurs. Si elle est plus petite, je remplace par les nouvelles valeurs de `a` et de `b`

Pour faire la différence entre deux valeurs, peu importe le signe: on utilisera la valeur absolue, `abs` en Python.

4. Le message pour Queulorior

 Lien Capytale : **2549-94446**

En utilisant Processing.

Ou bien le module `turtle`.

5. Cerbère


 Lien Capytale : **0d75-95569**

Brute-forcez: tester toutes les longueurs entières possibles...


6. SW I: à l'assaut de Gunray

 Lien Capytale : **a5b7-101013**

Utilisez des accumulateurs.

 **Correction**



 Script Python

```
1  epaisseur_totale = 0
2  volume = 0
3  temps = 0
4  moitie = 0
5  while epaisseur_totale < 70:
6      epaisseur = 3 - 0.005*volume
7      volume += 8*epaisseur
8      epaisseur_totale += epaisseur
9      temps += 1
10     if epaisseur_totale > 35 and moitie == 0:
11         moitie = temps
12     print(moitie, temps)
```