Défis logiques

# But du laboratoire

* Utiliser les collections
* Utiliser les conditions
* Utiliser les boucles
* Utiliser TOUT!

# Des chats avec des chapeaux[[1]](#footnote-1)

Nicolas est devenu un fou des chats, il pense que ça le rend charmant. Il a 100 chats ! Une journée où son isolation de COVID le rend un peu cinglé, il décide de faire un grand rond avec ses chats et de leur mettre des chapeaux. Cependant, il a une méthode bien particulière. À chaque fois qu’il passe devant un chat sans chapeau il lui en met un, mais s’il a déjà un chapeau, il lui enlève. Nicolas va faire 100 tours, voici comment il procède.

Le premier tour, il arrête à chaque chat, ils auront donc tous un chapeau.

Le deuxième tour, il arrête seulement à chaque 2 chats.

Le troisième tour, il arrête à chaque 3 chats, et ainsi de suite.

Le centième et dernier tour, il va arrêter seulement au 100e et dernier chat.

Finalement, votre application doit afficher quels chats ont encore un chapeau sur la tête. De plus, dites à Nicolas qu’il a besoin de nouveaux passe-temps.

Réponse :

Le 1e chat a un chapeau.

Le 4e chat a un chapeau.

Le 9e chat a un chapeau.

Le 16e chat a un chapeau.

Le 25e chat a un chapeau.

Le 36e chat a un chapeau.

Le 49e chat a un chapeau.

Le 64e chat a un chapeau.

Le 81e chat a un chapeau.

Le 100e chat a un chapeau.

# 

# Une petite marche en attendant[[2]](#footnote-2)

Nicolas habite la ville de Cartésia, là où toutes les routes forment une grille parfaite, un peu comme Manhattan en fait. Il arrive 10 minutes en avance à son rendez-vous et n’a pas envie d’attendre sans bouger. Heureusement, il s’est programmé une petite application qui lui génère des parcours de marche sur son téléphone. Nicolas veut marcher 10 minutes pour être **de retour à l’heure et l’endroit exact du rendez-vous**.

L’application de marche lui envoie donc ces parcours :

ls\_parcours = [

['n'],

['n','s','n','s','n','s','n','s','n','s'],

['n','s'],

['e','w','e','w','n','s','n','s','e','w'],

['n','s','n','s','n','s','n','s','n','s','n','s'],

['n','s','e','w','n','s','e','w','n','s'],

['n','s','e','w','n','s','e','w','n','s','e','w','n','s','e','w'],

['n','s','n','s','n','s','n','s','n','n'],

['e','e','e','w','n','s','n','s','e','w'],

['s','e','w','n','n','s','e','w','n','s']

]

Votre application doit afficher lesquels de ces parcours répondent à son besoin.

**Important :**

* Chaque parcours (liste) contient uniquement les caractères [n, s, e, w].
* Un parcours n’est jamais vide.
* Il faut marcher 1 minute à chaque indication par le parcours.

Réponse :

Le parcours 1 est erroné.

Le parcours 2 est bon!

Le parcours 3 est erroné.

Le parcours 4 est bon!

Le parcours 5 est erroné.

Le parcours 6 est bon!

Le parcours 7 est erroné.

Le parcours 8 est erroné.

Le parcours 9 est erroné.

Le parcours 10 est bon!

1. Exercice tiré du livre Python Basics - [Real Python p.278](https://github.com/realpython/python-basics-exercises/blob/master/ch09-lists-tuples-and-dictionaries/9b-challenge-cats-with-hats.py) [↑](#footnote-ref-1)
2. Exercice trouvé sur Codewars : [Take a Ten Minute Walk](https://www.codewars.com/kata/54da539698b8a2ad76000228) [↑](#footnote-ref-2)