Liste d’exercices

## Types et données

Facile:

1. Écrire une fonction sum qui à partir de deux entiers renvoi leur somme.

print(sum(8,4)) → 12

1. Écrire une fonction oppose qui à partir d’un nombre renvoi son opposé.

print(oppose(8)) → -8

Moyen:

1. Écrire une fonction waitInteger qui renvoie vrai si l'entrée de l’utilisateur est un entier, faux sinon.

print(waitInteger()) → true

1. Écrire une fonction isBissextile qui calcule si une année est bisextile ou non.

Une année est bissextile si l'année est divisible par 4 et non divisible par 100 ;

si l'année est divisible par 400 (« divisible » signifie que la division donne un nombre entier, sans reste).

print(isBissextile(2000)) → true

Difficile:

1. Écrire une fonction isValidCardNumber qui calcul si un numéro de carte bancaire est valide ou non à l’aide de l’algorithme de luhn.

print(isValidCardNumber(8532)) → true

L'algorithme de Luhn commence par la fin du numéro. Prendre le dernier chiffre et se déplacer vers la gauche. Doubler la valeur de tous les chiffres de rang pair. Si le double d'un chiffre est supérieur ou égal à 10, le remplacer par la somme de ses chiffres. Réaliser la somme

de tous les chiffres obtenus. La clé de contrôle c (donc le dernier chiffre) est égal à

c =(10−(s mod 10) mod 10)

Exemple:

Calculer X pour le nombre 853X, en posant d'abord X=0.

Prendre le 3, le doubler pour obtenir 6.

Prendre le 5, non doublé.

Enfin, doubler le 8 (8\*2=16 et 1+6=7) pour obtenir 7.

La somme est 6+5+7 = 18. Comme 18 modulo 10 = 8, calculer 10 - 8 = 2

Donc 8532 est valide mais pas 8531, ni 8533 ni 3, etc.

1. Faire un jeu qui attend que l’utilisateur retrouve le nombre calculé aléatoirement par l’algo, renvoi c’est plus si le nombre est trop grand, c’est moins si le nombre est trop petit, et c’est gagné une fois que l’utilisateur à trouver.

Compté également le nombre d'essais et le temps passé à trouver la solution.

## Listes

Facile:

1. Ecrire une fonction first qui renvoi le premier élément de la liste

print(first([7,5,9,2])) → 7

1. Ecrire une fonction last qui renvoi le premier élément de la liste

print(last([7,5,9,2])) → 2

1. Ecrire une fonction initListe qui initialise une liste avec les entiers de 0 à n

print(initListe(4)) → [0, 1, 2, 3, 4]

Moyen:

1. Écrire une fonction contient qui vérifie si une fonction contient ou non l’élément demandé.

print(contain([0,2,4,8],2) → vrai

print(contain([0,2,4,8],3) → faux

1. Écrire une fonction moyenne qui renvoie la moyenne des éléments d’une liste.

print(moyenne([2,4,6,8])) → 5

1. Écrire une fonction occurs qui compte le nombre d'occurrence (d’apparition) d’un élément dans une liste.

print(occurs([2,5,4,2,6,7,8,5,7],7) → 2

print(occurs([2,5,4,2,6,7,8,5,7],4) → 1

print(occurs([2,5,4,2,6,7,8,5,7],1) → 0

1. Écrire une fonction sort qui tri les éléments d’une liste.

print(sort([2,4,3]) → 2 , 3 , 4

1. Ecrire une fonction contains qui renvoi vrai si tous les éléments de la liste 2 sont contenues dans la première, faux sinon;

print(contains([2,5,7,9,6],[2,6]) → vrai

Difficile:

1. Trier une liste en utilisant un algo de tri rapide ou fusion
2. Écrire une fonction qui fait la somme de deux matrices (listes de listes).

print(sum([[1,2],[4,8]],[[5,6],[7,2]]) → [[6,8],[11,10]]

1. Écrire une fonction qui initialise une table de multiplication de taille n:

print(tableMult(4)):

[[1,2,3,4],

[2,4,6,8],

[3,6,9,12],

[4,8,12,16]]

## Projet

Facile:

1. Faire un pendu (texte)
2. Faire un Morpion (tableau):
3. Faire un Memory avec des nombres à retrouver dans un tableau

Moyen:

1. Démineur (tableau)

Turtle

1. Marche aléatoire avec turtle

https://docs.python.org/fr/3/library/turtle.html

<https://www.youtube.com/watch?v=wTHTleFmFzU>

1. Faire une marche en ligne droite à l’infini
2. Faire une fonction tourner aléatoire
3. Afficher la position au bout de 10, 100, 1000, 2000, 3000 pas
4. Ajouter plusieurs curseurs

import turtle, random

tortue=turtle.Turtle()

while True:

    tortue.forward(10)

    choix=random.randint(0,3)

    if (choix==0):

        tortue.left(90)

    if (choix==2):

        tortue.right(90)

Difficile:

1. Faire un mémory avec des images
2. Démineur avec IHM
3. Casse brique
4. Jeu de la vie

<https://www.youtube.com/watch?v=S-W0NX97DB0>

# D. Turtle

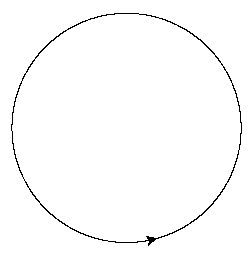
**A) Itératif**

Facile:

1. Dessiner un carré



1. Dessiner un cercle

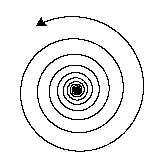


Moyen

3. Dessiner un escargot carré



4. Dessiner un escargot rond

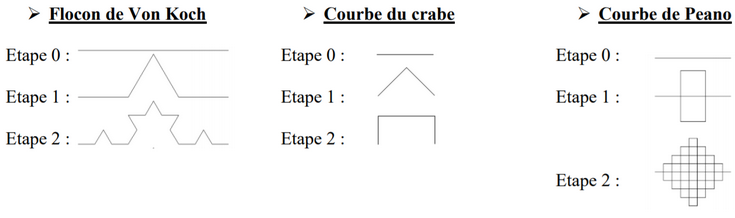


Difficile:

5. Faire ricocher tortue dans un rectangle

**b) Fractale (récursif) donc difficile**

1. **2. et 3.**



**4. Tapis ou triangle de Sierpiński (triangle ou carré)**

