

## TP PYTHON

**I. Écrire un programme qui permute le contenu de deux variables a et b.**

### **II. Valeur absolue d'un nombre.**

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre entier.
- En sortie : la valeur absolue de ce nombre.

### **III. Signe d'un nombre.**

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre entier.
- En sortie : le signe de ce nombre (positif ; négatif ou nul).

### **IV. Parité d'un nombre.**

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre entier positif.
- En sortie : la parité de ce nombre.

### **V. Jeu de la devinette.**

Le jeu consiste à deviner un nombre entier entre 1 et 100 choisi aléatoirement par la calculatrice en un minimum de coup.

Concevoir un programme avec les contraintes suivantes.

En entrée : un nombre entier entre 1 et 100.

En sortie : le nombre de coups pour deviner le nombre.

Remarque : pour créer un nombre entier aléatoire entre 1 et 100 :

```
from random import randint
nombre = randint(1,100)
```

### **VI. Numéro de sécurité sociale.**

1°) Le numéro de sécurité sociale est constitué de 13 chiffres auquel s'ajoute la clé de contrôle (2 chiffres).

La clé de contrôle est calculée par la formule :  $97 - (\text{numéro de sécurité sociale modulo } 97)$ .

Écrire un programme qui contrôle la validité d'un numéro de sécurité sociale.

On pourra utiliser la fonction `int()` pour convertir le type `str` en type `int`.

Tester votre programme avec 189112610826891 puis 289112610826891.

2°) Reprendre le programme en saisissant le numéro de sécurité sociale à 15 chiffres.

### **VII. Indice de masse corporelle.**

Écrire un script qui calcule l'indice de masse corporelle  $IMC = \text{masse}/\text{taille}^2$  avec la masse en kg et la taille en m d'un adulte et qui en donne l'interprétation.

- $IMC < 18.5$  : maigre ;
- $18,5 \leq IMC < 25$  : corpulence normale ;
- $25 \leq IMC < 30$  : surpoids ;
- $IMC \geq 30$  : obésité).

**VIII.** 1°) Écrire un script qui détermine si un nombre entier positif est premier ou pas. Faire le test avec les nombres suivants et remplir le tableau avec le temps d'exécution.

Nombre	23 456 789	99 999 989	1 000 000 007
Temps d'exécution			

2°) Écrire un script qui décompose un nombre entier positif en un produit de facteurs premiers.

Exemple : 4 290 affichera les facteurs premiers 2 ; 3 ; 5 ; 11 et 13.

**IX.** Concevoir la fonction **pal(ch)** avec les contraintes suivantes :

En entrée : une chaîne de caractère ch.

En sortie : retourne un booléen qui détermine si la chaîne ch est un palindrome, c'est-à-dire une chaîne qui peut se lire indifféremment dans les deux sens.

Vous ne pouvez pas utiliser la fonction **reverse()**.

Par exemple, à partir de la chaîne de caractère ch='ressasser', la fonction retournera True.

Tester le programme avec les chaînes suivantes :

"La mariée ira mal"

"Tu l'as trop écrasée, César, ce Port Salut".

**X.** 1°) Écrire, avec la boucle while, une fonction qui retourne la factorielle d'un nombre entier n positif ou nul. On rappelle que :  $n! = 1 \times 2 \times \dots \times (n-1) \times n$  et  $0! = 1$ .

Comparez avec le résultat de la fonction **factorial()** du module math.

2°) Même programme avec une boucle for.

**XI.** On utilisera dans les exercices suivants la chaîne de caractères ch='azerty0123456789' à titre d'exemple. Vous devrez envisager tous les programmes possibles.

1°) **a)** Créer la fonction **tronq(ch)** avec les paramètres suivants.

En entrée : la chaîne de caractères ch.

En sortie : la chaîne de caractère sans le premier caractère.

**b)** Créer la fonction **ante(ch)** avec les paramètres suivants.

En entrée : la chaîne de caractères ch.

En sortie : la chaîne de caractère jusqu'à l'antépénultième caractère inclus.

**c)** Créer la fonction **portion(ch,i,j)** avec les paramètres suivants.

En entrée : la chaîne de caractères ch et deux nombres entiers positifs tels que  $j > i$ .

En sortie : la chaîne de caractère du caractère d'indice i jusqu'au caractère d'indice j.

2°) **a)** Créer la fonction **inverse(ch)** avec les paramètres suivants.

En entrée : la chaîne de caractères ch.

En sortie : la chaîne de caractère à l'envers.

**b)** Afficher la chaîne ch et la chaîne inversée retournée sous la forme suivante :

a	z	e	r	t	y	0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0	y	T	r	e	z	a

3°) Créer la fonction **subst(ch,old,new)** avec les paramètres suivants.

En entrée : une chaîne de caractères ch, le caractère à remplacer et celui de remplacement.

En sortie : la chaîne de caractères modifiée.

4°) Créer la fonction **indice(ch,car)** avec les paramètres suivants.

En entrée : la chaîne de caractères ch et un caractère car.

En sortie : le caractère et son indice dans la chaîne de caractères (-1 si le caractère ne se trouve pas dans la chaîne de caractères).

**a)** Utiliser la fonction **find()** dans un premier temps.

**b)** Ne pas utiliser la fonction **find()** dans un second temps.

**XII.** Reprendre tous les exercices précédents en remplaçant "chaîne de caractère" par "liste". On prendra la liste li = ['a', 'z', 'e', 'r', 't', 'y', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'] pour les tests.

**XIII.** Définir une fonction **trouve(ch,li)** avec les paramètres suivants :

En entrée : une chaîne de caractères ch et une liste de caractères li.

En sortie : un booléen qui confirme ou infirme qu'on puisse écrire le mot de la chaîne de caractères avec la liste de caractères. Exemple : 'aimer' et ['a', 'v', 'm', 'i', 'r', 'r', 'e'].

**a)** Utiliser la fonction **count()** dans un premier temps.

**b)** Ne pas utiliser la fonction **count()** dans un second temps.

**XIV. 7°)** Écrire une fonction qui, à partir d'une liste de 0 et de 1 uniquement, indique si les 0 et les 1 sont alternés.

**XV.** Soit la liste de nombres [8, 3, 12, 5, 45, 25, 5, 52, 1]. Triez les nombres de cette liste par ordre croissant, sans utiliser la fonction **sort()**. Les fonctions **min()**, **append()** et **remove()** vous seront utiles.

**XVI.** Générez aléatoirement une séquence de nucléotides d'ADN de 20 bases en utilisant une liste avec la méthode **append()** et/ou une chaîne de caractères. La fonction **choice()** est interdite.

**XVII.** Transformez la séquence nucléotidique TCTGTTAACCATCCACTTCG en sa séquence complémentaire inverse. Afficher la séquence initiale, complémentaire et complémentaire inverse. Ne pas utiliser la fonction **reverse()**.

**XVIII.** Écrire un programme qui demande à l'utilisateur d'entrer des notes d'élèves comprises par exemple, entre 0 et 20. La boucle se termine lorsque l'utilisateur appuie sur Entrée sans note. Avec les notes ainsi entrées, construire progressivement la liste des notes.

Après chaque entrée d'une nouvelle note, afficher la liste de notes, le nombre de notes entrées, la note la plus élevée, la note la plus basse, la moyenne de toutes les notes et la note médiane. Créer une fonction pour chaque calcul. Vous ne pouvez pas utiliser les fonctions **min** et **max** et **sort**. La fonction **sort()** ne sera utilisée que dans la fonction de calcul de la médiane.

**XIX.** Construire les listes suivantes en compréhension (une seule ligne d'instructions) :

- [1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 76, 81, 86, 91, 96].
- [58, 55, 52, 49, 46, 43, 40, 37, 34, 31, 28, 25, 22, 19] (utiliser **range** avec un pas négatif).
- La liste des entiers pairs compris entre 0 et 100 (utiliser la commande modulo %).
- La liste des diviseurs de 20876 (utiliser la commande modulo %).
- La liste ['\*2\*', '\*22\*', '\*222\*', '\*2222\*', '\*22222\*', '\*222222\*'] (utiliser **range** et concaténation de chaînes de caractères : 'a+b' donne 'ab' et 'a'\*5 donne 'aaaaa').
- La liste des couples d'entiers entre 0 et 10 dont la somme est multiple de 5 (double boucle **for** et **test if**).

## XX. Suite de Fibonacci.

1°) Concevoir un programme avec les contraintes suivantes.

- En entrée : le nombre **n** de termes de Fibonacci.
- En sortie : la liste des **n** termes de la suite.

2°) Concevoir un programme avec les contraintes suivantes.

- En entrée : la liste précédente des **n** premiers termes de Fibonacci.
- En sortie : la liste des quotients de deux termes consécutifs.

Afficher la valeur du quotient  $\frac{1+\sqrt{5}}{2}$ .

## XXI. Calcul de sommes et de produits.

1°) Calculer les sommes suivantes.

a) $1+2+3+\dots+100$	b) $3+6+9+\dots+201$	c) $-1-4-7-\dots-31$
----------------------	----------------------	----------------------

2°) Calculer les produits suivants.

a) $2 \times 4 \times 6 \times 8 \times \dots \times 40$	b) $24 \times 12 \times 6 \times 3 \times \dots \times 0,1875$
--	--

3°) Calculer les sommes suivantes. On affichera la valeur exacte et une valeur approchée de chaque somme.

a) $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{10}$ .	b) $1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \dots + \frac{1}{51}$ .
c) $1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{12} + \dots + \frac{1}{768}$	d) $1 + \frac{1}{2} + \frac{1}{2 \times 3} + \frac{1}{2 \times 3 \times 4} + \dots + \frac{1}{2 \times 3 \times \dots \times 10}$