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.

IV. Parité d'un nombre.

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre.
- 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 :

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 - (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 = masse/taille² avec la masse en kg et la taille en m d'un adulte et qui en donne l'interprétation.

- IMC < 18.5 : maigreur ;
- 18,5<= IMC < 25 : corpulence normale ;
- 25 <= IMC < 30 : surpoids ;
- IMC>= 30 : obésité).

VIII. 1°) Écrire un script qui détermine si un nombre entier positif est premier ou pas. 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.

X. 1°) Écrire, avec la boucle while, une fonction qui retourne la factorielle d'un nombre entier n positif ou nul. On rappelle que : $n! = 1\times2\times...\times(n-1)\times n$ 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 modifée.
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.

5°) Créer la fonction **occ**(ch) avec les paramètres suivants.
En entrée : une chaîne de caractères ch. Par exemple ch='12223454224650798613460573 1' ou li=[1,2,2,2,3,4,5,4,2,2,2,4,6,5,0,7,9,8,6,1,3,4,6,0,5,7,3,1].
En sortie : une liste li comptabilisant le nombre de caractère correspondant à un chiffre.
Par exemple, la liste li=[0, 1, 1, 0, 3, 0, 1, 2, 1,0] signifie qu'il y un 1, un 2, trois 4, 1 six, 2 sept et un 8.
a) Utiliser la fonction **count**() dans un premier temps.
b) Ne pas utiliser la fonction **count**() 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. Écrire une fonction similaire à la fonction **shuffle**(), puis à la fonction **choice**().
XIV. 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.
XV. 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.
XVI. 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.
XVII. 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.

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

XIX. Soit la liste de nombres [5, 1, 1, 2, 5, 6, 3, 4, 4, 4, 2]. Enlevez les doublons de cette liste, triez-là et affichez-là.

XX. Générez aléatoirement une séquence nucléotides d'ADN de 50 bases contenant 10% de A, 50% de G, 30% de T et 10% de C.

XXI. 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.

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

- [1, 6, 11, 16, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 76, 81, 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)

XXIII. Écrire un programme en python qui permet de déterminer si une année (dont le millésime est introduit par l'utilisateur) est bissextile ou non.

Une année A est bissextile si A est divisible par 4. Elle ne l'est cependant pas si A est un multiple de 100, à moins que A ne soit multiple de 400.

XXIV. 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 $\frac{1+\sqrt{5}}{2}$.

XXV. Calcul de sommes et de produits.

1°) Calculer les sommes suivantes.

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

2°) Calculer les produits suivants.

a) $2\times4\times6\times8\times....\times40$	b) $24\times12\times6\times3\times....\times0{,}1875$

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

a) $1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{10}$.	b) $1+\frac{1}{3}+\frac{1}{5}+\frac{1}{7}+...+\frac{1}{51}$.
c) $1+\frac{1}{3}+\frac{1}{6}+\frac{1}{12}+...+\frac{1}{768}$	d) $1+\frac{1}{2}+\frac{1}{2\times3}+\frac{1}{2\times3\times4}+...+\frac{1}{2\times3\times...\times10}$