Informatique - Exercices d'entraînement 1

Les objectifs de ce document sont de compiler quelques exercices d'entraînement sur les notions "de base" de programmation en Python rencontrées jusqu'à présent, et de vous permettre de tester si vous avez suffisamment compris les exercices faits en TP pour en traiter d'autres !

Ces exercices sont classés par thèmes, et vous trouverez dans ce document des :

D'ici la fin du semestre, d'autres thèmes seront sans doute à rajouter à cette liste ; mais avec un peu de recul, ces thèmes futurs seront principalement des variations autour des notions évoquées ci-dessus, qu'il vous sera donc important de maîtriser.

Alors au travail, et comme il est écrit à l'ouverture de Pyzo : 'Happy coding !'

1 - Boucles for

Exercice 1. — Ecrire un programme qui affiche 50 fois "Je dois ranger mon bureau".

Exercice 2. — Calculer la somme $S = 4 + 5 + \cdots + 2023$.

Exercice 3. — Calculer la somme $S = 4^2 + 5^2 + \cdots + 2023^2$.

Exercice 4. — Calculer la somme $S = 5 + 8 + 11 + \cdots + 401$

Exercice 5. — Demander à l'utilisateur de saisir un entier n, puis afficher la somme :

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

Exercice 6. — **Valeur approchée de $\zeta(2)$.**

1/ Demander à l'utilisateur de saisir un entier n, puis afficher la somme :

$$S_n = 1 + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{n^2}$$

2/ Demander à l'utilisateur de saisir un entier n, puis afficher une valeur approchée de :

$$\frac{\pi^2}{6} - S_n$$

Exercice 7. — **Simulation de lancers de dé.**

Ecrire un programme qui demande à l'utilisateur de saisir un entier n, et qui affiche n entiers choisis aléatoirement entre 1 et 6.

Remarque. On pourra utiliser la fonction `randint` de Python, en l'important à l'aide de l'instruction :

```
from random import randint
```

Ceci fait, l'instruction `randint(a,b)`(avec a et b entiers) renvoie un entier choisi aléatoirement entre a et b au sens large.

Exercice 8. — **Simulation de tirages de lettres.**

Ecrire un programme qui demande à l'utilisateur de saisir un entier n, et qui affiche n lettres minuscules choisies aléatoirement.

Remarque. On pourra introduire la chaîne de caractères :

```
alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

Exercice 9. — Définir la liste suivante :

```
animaux = ['chien', 'mouton', 'chat', 'girafon', 'lapin', 'pingouin', 'presse-purée']
```

Puis écrire un programme qui demande à l'utilisateur de saisir un entier entre 1 et 7, et qui affiche, si l'utilisateur a choisi l'entier 4 :

```
'Le mot girafon possède 7 caractères'
```

Remarque. On pourra envisager un message de réprobation si l'utilisateur ne saisit pas un entier entre 1 et 7.

2 - Constructions de listes avec l'instruction for

Deux exemples pour commencer. L'instruction

```
L = [ k**2 for k in range(2,6) ]
```

crée la liste

```
L = [4, 9, 16, 25]
```

Et l'instruction

```
L = [ 2*elem + 1 for elem in [3,7,10,20] ]
```

crée la liste

```
L = [7, 15, 21, 41]
```

A l'aide de ces exemples, répondre aux questions suivantes.

Exercice 10. — Ecrire une instruction permettant de créer la liste des multiples de 3 compris entre 1 et 100.

Exercice 11. — **Simulation de lancers de dé.** Ecrire une instruction permettant de créer une liste de 20 entiers choisis aléatoirement entre 1 et 6.

Exercice 12. — Ecrire une instruction permettant de créer la liste :

```
[ 0, 1, 0, 1, 0, 1, 0, 1]
```

Exercice 13. — Ecrire une instruction permettant de créer la liste :

```
[ 0, 0, 1, 1, 2, 2, 3, 3]
```

Exercice 14. — **Erreur de calcul ?**

1/ Ecrire une instruction permettant de créer la liste (contenant 10 fois la valeur 0.1) :

```
L = [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
```

2/ Calculer de tête la somme des éléments de L.

3/ A l'aide d'une boucle for, demander à la machine de calculer la somme des éléments de L.

4/ A la lumière du résultat de la question 3, choisir la bonne réponse parmi celles proposées ci-dessous :

- **A.** ☐ Le cours de maths est truffé d'erreurs, je le savais déjà.
- **B.** ☐ Le cours d'info est truffé d'erreurs, je le savais déjà.
- **C.** ☐ Les réponses A et B sont vraies.
- **D.** ☐ La machine ne sait faire que ce pour quoi on l'a programmée.

3 - Instructions conditionnelles if, elif, else

Exercice 15. — Ecrire un programme qui demande à l'utilisateur de saisir 2 valeurs, et qui affiche la plus petite des 2 valeurs.

Exercice 16. — Ecrire un programme qui demande à l'utilisateur deux entiers pairs a et b, et qui affiche la moyenne de ces deux entiers.

Si au moins un de ces entiers est impairs, le programme doit afficher un message d'erreur.

Exercice 17. — Ecrire un programme qui calcule les 20 premiers termes de la table de multiplication par 11, mais n'affiche que ceux qui sont des multiples de 7.

Exercice 18. — Ecrire un programme qui affiche les 20 premiers termes de la table de multiplication par 7 en signalant à l'aide d'un astérisque ceux qui sont multiples de 3.

Exercice 19. — Ecrire un programme qui demande à l'utilisateur de saisir 2 chaînes de caractères, et qui affiche la plus grande des 2 chaînes (celle qui a le plus de caractères).

Si les deux chaînes sont de même longueur, le programme doit afficher 'match nul'.

Exercice 20. — Ecrire un programme qui demande à l'utilisateur un entier, et qui affiche en fonction de la valeur saisie l'un des messages suivants :

— 'Ce nombre est pair'
— 'Ce nombre est impair, mais est multiple de 3'
— 'Ce nombre n'est ni pair ni multiple de 3'

Exercice 21. — On choisit deux nombres distincts a et b dans l'intervalle d'entiers $[\![1, 26]\!]$ tels que leur produit est égal à la somme des 24 autres valeurs restantes. Que peuvent valoir ces nombres ?

4 - Boucles while

Exercice 22. — Ecrire un programme qui affiche les nombres de 2 en 2 jusqu'à 100 avec un for, puis avec un while.

Exercice 23. — *Chanson traditionnelle bretonne*
Ecrire un programme demandant à l'utilisateur de saisir un entier n, et qui affiche, si l'utilisateur a choisi $n = 10$:

```
C'est dans 10 ans je m'en irai j'entends le loup le renard chanter
C'est dans 9 ans je m'en irai j'entends le loup le renard chanter
...
C'est dans 1 ans je m'en irai j'entends le loup le renard chanter
```

Ce programme devra être écrit avec l'instruction while ; et dans un premier temps, on pourra négliger la faute d'orthographe de la dernière phrase.

Exercice 24. — Ecrire un programme qui demande à l'utilisateur un nombre flottant, et qui calcule le plus grand entier dont le carré est plus petit que ce nombre.

Exercice 25. — Pour tout entier naturel non nul n, on pose :

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

Il est connu que S_n tend vers $+\infty$ lorsque n tend vers $+\infty$.

Ecrire un programme demandant à l'utilisateur de choisir un nombre X, et qui affiche la première valeur de n telle que $S_n > X$.

Exercice 26. — Pour tout entier naturel non nul n, on pose :

$$S_n = 1 + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{n^2}$$

Il est connu que S_n tend (en croissant) vers $\frac{\pi^2}{6}$ lorsque n tend vers $+\infty$.

Ecrire un programme demandant à l'utilisateur de choisir un nombre epsilon, et qui affiche la première valeur de n telle que $\frac{\pi^2}{6} - S_n <$ epsilon.

Exercice 27. — Ecrire un programme demande à l'utilisateur de saisir un entier n, et qui affiche le plus petit diviseur $d \geqslant 2$ de n.

Exercice 28. — Ecrire un programme demandant à l'utilisateur de saisir un entier N, et affichant une liste de N points situés dans le quart de disque "supérieur droit" de centre 0 et de rayon 1.

Exercice 29. — Ecrire un programme demandant à l'utilisateur de saisir un entier N$\leqslant$ 20, et affichant une liste de N entiers dont la somme des chiffres est égale à 6, ces entiers étant choisis aléatoirement entre 1 et 1000. Le programme doit renvoyer un message d'erreurs si N$>$ 20.

5 - Chaînes de caractères

Exercice 30. — Créer un programme demandant à l'utilisateur de saisir un mot, et qui renvoie l'initiale de ce mot.

Exercice 31. — Créer un programme demandant à l'utilisateur de saisir un mot de 4 lettres ou plus, et qui renvoie les quatre premières lettres de ce mot. Un message d'erreur cordial doit être affiché si l'utilisateur a saisi un mot de 3 lettres ou moins.

Exercice 32. — Créer un programme demandant à l'utilisateur de saisir une chaîne de caractères, et qui affiche le nombre de 'e' contenus dans cette chaîne de caractères.

Exercice 33. — Créer un programme demandant à l'utilisateur de saisir une chaîne de caractères, et qui remplace toutes les voyelles de cette chaîne par des 'o'.

Ainsi, si l'utilisateur a saisi

```
'Vive la MPSI'
```

le programme doit afficher

```
'Vovo lo MPSO'
```

Exercice 34. — **Jeu de cartes 1.** Créer une liste contenant l'ensemble des valeurs des cartes :

```
Lval = [ '1', '2',..., '10', 'V', 'D', 'R' ]
```

Exercice 35. — **Jeu de cartes 2.** Créer une liste contenant l'ensemble des couleurs des cartes :

```
Lcoul = [ 'Pique', 'Coeur', 'Carreau', 'Trefle' ]
```

Exercice 36. — **Jeu de cartes 3.** Créer une liste correspondant aux 52 cartes d'un jeu :

```
Ljeu = [ '1 de Pique',..., 'R de Pique', '1 de Coeur',..., 'R de Trefle' ]
```

Exercice 37. — **Jeu de cartes 4.** Créer un programme choisissant au hasard 5 cartes (2 à 2 distinctes) d'un jeu de 52 cartes.

Exercice 38. — **Jeu de cartes 5.** Créer un programme choisissant au hasard 5 cartes (2 à 2 distinctes) d'un jeu de 52 cartes, et déterminant si ces cartes sont de la même couleur.

Exercice 39. — Créer un programme demandant à l'utilisateur de saisir une chaîne de caractères, et qui affiche la chaîne de caractères obtenue en supprimant une lettre sur 2. Par exemple, si l'utilisateur a saisi 'informatique', le programme doit afficher 'ifraiu'.

Exercice 40. — Reprendre la liste

```
animaux = ['chien', 'mouton', 'chat', 'girafon', 'lapin', 'pingouin', 'presse-purée']
```

de l'exercice 9, et créer la liste

```
compar = ['a-t-il plus de pattes', 'est-il plus grand', 'vole-t-il plus vite',
          'mange-t-il plus de carottes', 'est-il aussi affectueux']
```

A l'aide de ces deux listes, créer un programme qui génère aléatoirement 3 questions comme :

```
Le mouton a-t-il plus de pattes que le pingouin?

Le girafon vole-t-il plus vite que le chien?

Le pingouin est-il aussi affectueux que le presse-purée?
```

6 - Fonctions

Exercice 41. — Ecrire le code Python d'une fonction `fonc1(x)` qui prend comme paramètre un flottant x, et qui retourne $x^2 + x + 1$.

Exercice 42. — Ecrire le code Python d'une fonction NB4(L) qui reçoit comme paramètre une liste d'entiers L, et qui renvoie le nombre d'apparitions de l'entier 4 dans la liste L.

Par exemple l'instruction NB4([1,2,4,5,7,4,12]) doit produire le résultat : 2

Exercice 43. — Ecrire le code Python d'une fonction NB(L,n) qui reçoit comme paramètres une liste d'entiers L et un entier n, et qui renvoie le nombre d'apparitions de l'entier n dans la liste L.

Par exemple l'instruction NB([1,2,4,2,7,4,2],2) doit produire le résultat : 3.

Exercice 44. — Ecrire le code Python d'une fonction Pairs(L) qui reçoit comme paramètre une liste d'entiers L, et qui renvoie le nombre d'entiers pairs de L, ainsi que la liste des positions de ces entiers.

Par exemple l'instruction Pairs([1,2,4,5,7,11,12]) doit produire le résultat : 3,[1,2,6]

Exercice 45. — **A epsilon près...**

Ecrire une fonction `Vapproch(x,y,epsilon)` de paramètres trois flottants x, y et `epsilon`, qui retourne True si $|y - x| \leqslant$ `epsilon` et False sinon.

Exercice 46. — **Minimum et maximum**

Ecrire une fonction `MinMax(L)` de paramètre une liste non-vide, qui retourne le plus petit et le plus grand élément de cette liste.

Exercice 47. — Soit (u_n) la suite réelle définie par $u_0 = 2$ et :

$$\forall n \in \mathbb{N},\ u_{n+1} = \frac{2u_n + 1}{3}$$

1/ Ecrire une fonction F(N) en Python qui reçoit comme paramètre un entier naturel N, et qui retourne la valeur de u_N.

2/ Ecrire une instruction permettant d'afficher les 20 premiers termes de la suite (u_n).

Exercice 48. — Soit (u_n) la suite réelle définie par $u_0 = 1$, $u_1 = 3$ et :

$$\forall n \in \mathbb{N},\ u_{n+2} = u_{n+1} - u_n + 1$$

1/ Ecrire une fonction F(N) en Python qui reçoit comme paramètre un entier naturel N, et qui retourne la valeur de u_N.

2/ Ecrire une instruction permettant d'afficher les 20 premiers termes de la suite (u_n).

Exercice 49. — **Deux suites liées**.

Soient (u_n) et (v_n) les deux suites réelles respectivement définies par $u_0 = 2$ et $v_0 = 10$, et :

$$\forall n \in \mathbb{N},\ u_{n+1} = \frac{2u_n + v_n}{3} \quad \text{et} \quad v_{n+1} = \frac{u_n + 3v_n}{4}$$

1/ Ecrire une fonction F(N) en Python qui reçoit comme paramètre un entier naturel N, et qui retourne les valeurs de u_N et de v_N.

2/ Ecrire les deux lignes de code permettant d'afficher les 20 premiers termes (valeurs approchées) des suites (u_n) et (v_n).

Exercice 50. — **Formule de Héron - aire d'un triangle.**

1/ Ecrire une fonction `Dper(a,b,c)`de paramètres les côtés d'un triangle a, b, c, et qui retourne le demi-périmètre de ce triangle.

2/ Le mathématicien grec Héron d'Alexandrie a établi la formule suivante qui donne l'aire A d'un triangle de côtés a, b, c et de demi-périmètre p :

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

A l'aide de cette formule et de la question précédente, ecrire une fonction `AireT(a,b,c)`de paramètres les côtés d'un triangle a, b, c, et qui retourne l'aire de ce triangle.

Exercice 51. — Ecrire une fonction qui reçoit une chaîne de caractères en paramètre, et qui retourne le nombre de voyelles (minuscules, sans accents) de cette chaîne.

Exercice 52. — Ecrire une fonction qui reçoit une chaîne de caractères en paramètre, et qui retourne True si le mot ne contient que des voyelles (minuscules, sans accents) et False sinon.

Exercice 53. — Ecrire une fonction qui reçoit deux chaînes de caractères en paramètre, et qui retourne True si les mots contiennent autant de voyelles (minuscules, sans accents) et False sinon.

Exercice 54. — Afrikaans. En Afrikaans*, les lettres autorisées sont celles de notre alphabet à l'exception des lettres C, Q, X et Z.

Ecrire une fonction `Afrikaans(chaine)` de paramètre une chaîne de caractères, qui renvoie False si elle contient l'une des lettres C, Q, X ou Z (minuscule ou majuscule), et True sinon.

Exercice 55. — Codage basique.

1/ Construire une fonction `codage(TXT)` qui reçoit comme paramètre le texte TXT, et qui renvoie la chaîne de caractères constituée des lettres de rang pair dans TXT, et la chaîne de caractères constituée des lettres de rang impair dans TXT.

2/ Construire la fonction `decodage(TXT1, TXT2)` qui permet de faire l'opération inverse de celle de la question 1.

Exercice 56. — Ecriture décimale.

Construire une fonction `Deci(n)` qui reçoit comme paramètre un entier n, et qui retourne la liste de ses chiffres dans son écriture décimale.

Par exemple `Deci(453)` doit renvoyer `[4,5,3]`.

Exercice 57. — Ecriture binaire.

Construire une fonction `Bin(n)` qui reçoit comme paramètre un entier n, et qui retourne son écriture binaire.

Par exemple `Bin(13)` doit renvoyer `1101`.

Exercice 58. — Binaire $\longrightarrow$ Décimale.

Construire une fonction `BinDeci(n)` qui reçoit comme paramètre un entier n en binaire, et qui retourne l'écriture décimale de n.

Par exemple `BinDeci(1101)` doit renvoyer `13`.

Exercice 59. — Nombres premiers.

En arithmétique, un nombre premier est un entier $p \geqslant 2$ qui n'admet que 2 diviseurs dans $\mathbb{N}$: 1 et lui-même.

On peut montrer aisément que p est premier SSI il n'est divisible par aucun entier d tel que : $2 \leqslant d \leqslant \sqrt{p}$.

1/ Ecrire une fonction `Tprem(n)`de paramètre un entier n, qui retourne True si n est premier, et False sinon.

2/ Ecrire une fonction `Listeprem(n)`de paramètre un entier n, qui retourne la liste des nombres premiers compris entre 2 et n au sens large.

---

*. Qui est l'une des 12 (oui, douze !) langues officielles de l'Afrique du Sud.

Exercice 60. — Nombres de Fermat.

Les nombres de Fermat sont, en arithmétique, des nombres qui s'écrivent $F_n = 2^{2^n} + 1$ pour n entier naturel. Fermat conjectura qu'ils étaient tous premiers (en fait, F_1, F_2, F_3 et F_4 le sont ; mais F_5 ne l'est pas).

1/ Ecrire une fonction `Fermat(n)`de paramètre un entier n, qui retourne le nombre F_n.

2/ Déterminer le plus petit diviseur $d > 1$ de F_5.

Exercice 61. — Nombres de Mersenne.

Les nombres de Mersenne sont, en arithmétique, des nombres qui s'écrivent $M_n = 2^n - 1$ pour n entier naturel. Mersenne conjectura que si p est premier, alors M_p est premier.

1/ Ecrire une fonction `Mersenne(n)`de paramètre un entier n, qui retourne le nombre M_n.

2/ Construire la liste des nombres premiers compris entre 2 et 100.

3/ A l'aide des questions précédentes, déterminer le plus petit nombre premier p_0 tel que M_{p_0} n'est pas premier ; puis déterminer le plus petit diviseur $d > 1$ de M_{p_0}.

Ce qui prouve que la conjecture de Mersenne était fausse.

Exercice 62. — Nombres d'Armstrong.

On souhaite déterminer les entiers naturels <1000 qui sont égaux à la somme des cubes de leurs chiffres. De tels entiers seront appelés des nombres d'Armstrong.

Par exemple, l'entier 0 est un nombre d'Armstrong car $0^3 = 0$; mais l'entier 59 n'en est pas un car $5^3 + 9^3 = 854 \neq 59$.

1/ Ecrire une fonction `Somcubchif(n)` qui reçoit comme paramètre un entier naturel $n < 1000$ et renvoie la somme des cubes de ses chiffres.

Par exemple, `Somcubchif(256)` devra renvoyer : $2^3 + 5^3 + 6^3 = 349$.

2/ Ecrire une instruction qui fournit la liste des nombres d'Armstrong inférieurs à 999.

Exercice 63. — Nombres amicaux.

On définit $d(n)$ comme la somme des diviseurs propres de n (diviseurs de n strictement inférieur à n).

a et b sont des **nombres amicaux** si $d(a) = b$ et $d(b) = a$ avec $a \neq b$.

Exemple — Les diviseurs propres de 220 sont 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110.
Les diviseurs propres de 284 sont 1, 2, 4, 71, 142.
220 et 284 sont donc des nombres amicaux.

1/ Ecrire une fonction Divpro(n) qui retourne la liste des diviseurs propres d'un entier $n \geqslant 2$.

2/ Ecrire une fonction Ami(a,b) qui teste si deux entiers a et b sont amicaux.

Exercice 64. — Nombres de Carmichael.

Un entier naturel C supérieur ou égal à 4 est un **entier de Carmichael** il vérifie les deux assertions suivantes :

$$1)\ C \text{ n'est pas premier} \qquad 2)\ \forall n \in [\![0, C-1]\!],\ \left(n^C - n\right) \text{ est multiple de } C$$

1/ Ecrire une fonction `Tcarmi(N)` qui reçoit comme paramètre un entier N, et qui renvoie True si N est un entier de Carmichael, False sinon.

2/ Ecrire une fonction `Cherchecarmi(a,b)` qui reçoit comme paramètres deux entiers a et b, et qui renvoie la liste des entiers de Carmichael compris au sens large entre a et b.

A l'aide de cette fonction, vérifier qu'il existe un unique entier de Carmichael inférieur à 1000.

Exercice 65. — Racines N-ièmes de l'unité. Soit N un entier naturel supérieur ou égal à 2.

1/ On note $z_0, \ldots, z_{N-1}$ les racines N-ièmes de l'unité, et $M_0, \ldots, M_{N-1}$ les points images respectifs de ces nombres complexes. Rappeler, pour tout $k \in [\![0, N-1]\!]$, l'abscisse et l'ordonnée du point M_k.

2/ Ecrire une fonction `Racines(N)` qui reçoit comme paramètre un entier N supérieur ou égal à 2, et qui retourne la liste des abscisses LX et la liste des ordonnées LY des points $M_0, \ldots, M_{N-1}$ définis à la question précédente.

Dans peu de temps, nous verrons en TP comment créer des graphiques en Python, ce qui nous permettra de générer de magnifiques illustrations à l'aide des questions précédentes.

Exercice 66. — A la Bolzano-Weierstrass... Déterminer 11 entiers naturels n_0, n_1, ..., n_{10} tels que :

— $n_0 < n_1 < \cdots < n_{10}$;

— pour tout $k \in [\![0, 10]\!]$, $\cos(n_k)$ est une valeur approchée de -1 à 10^{-k} près.

Exercice 67. — (Centrale-Supélec TSI 2019)

La suite de Fibonacci est la suite (F_n) définie en posant :

$$F_0 = 0\,; \qquad F_1 = 1\,; \qquad \forall n \in \mathbb{N}, \quad F_{n+2} = F_{n+1} + F_n$$

1/ Ecrire une fonction Python `fibonacci(p)` qui prend en paramètre un entier naturel p, et renvoie la liste des $p+1$ premiers termes de la suite de Fibonacci.

Par exemple `fibonacci(5)` doit renvoyer `[0, 1, 1, 2, 3, 5]`.

2/ Ecrire une fonction Python `recherche(n)` qui prend en paramètre un entier naturel n, et renvoie le plus grand entier naturel p tel que $F_p \leqslant n$. Par exemple, `recherche(7)` doit renvoyer 5.

Exercice 68. — Pour tout entier naturel n, on pose :

$$S_n = \sum_{k=0}^{n} \frac{(-1)^k}{2k+1} \qquad \left(S_n = 1 - \frac{1}{3} + \frac{1}{5} - \cdots + \frac{(-1)^n}{2n+1}\right)$$

Ecrire une fonction $F(N)$ qui reçoit comme paramètre un entier N, et qui retourne la valeur de S_N.

Puis Ecrire un programme qui demande à l'utilisateur de saisir un entier N, et qui affiche la liste $L = [S_0, S_1, \ldots, S_N]$.

Exercice 69. — **(CCPINP-MP 2021)**

On rappelle qu'un nombre premier est un entier naturel $n \geqslant 2$ qui admet exactement deux diviseurs (1 et lui-même) dans $\mathbb{N}$.

1/ Ecrire une fonction booléenne `estPremier(n)` qui prend en argument un entier naturel non nul n et qui renvoie le booléen `True` si n est premier et le booléen `False` sinon.

On pourra utiliser le critère suivant : un entier $n \geqslant 2$ qui n'est divisible par aucun entier $d \geqslant 2$ tel que $d^2 \leqslant n$ est premier.

2/ En déduire une fonction `LPrem(n)` qui prend en argument un entier naturel non nul n, et renvoie la liste des nombres premiers inférieurs ou égaux à n.

3/ Pour p un nombre premier et n un entier naturel, on appelle valuation p-adique de n, et on note $v_p(n)$ le plus grand entier k tel que p^k divise n.[†]

Ecrire une fonction `Valpadic(n,p)` qui prend en argument un entier naturel n non nul et un nombre premier p, et qui renvoie la valuation p-adique de n.

Exercice 70. — **Scrabble gallois**. Créer un programme demandant à l'utilisateur de saisir un mot en majuscules, et qui retourne le nombre de points au Scrabble de ce mot. L'affichage produit doit être analogue à celui indiqué ci-contre.

Dans le cas où vous ne le connaîtriez pas, le Scrabble est un jeu dont le principe est de placer des mots sur un plateau de jeu. Chaque lettre est affectée d'un certain nombre de points, explicitement :

- **1 point** pour chaque A, E, I, O, U, L, N, R, S, T ;
- **2 points** pour chaque D, M, G ;
- **3 points** pour chaque B, C, P ;
- **4 points** pour chaque F, H, V ;
- **8 points** pour chaque J, Q ;
- **10 points** pour chaque K, W, X, Y, Z.

```
Saisissez un mot en majuscules
MPSI
Le mot  MPSI  rapporte  7  points au Scrabble
```

Question bonus : modifier le programme précédent pour qu'il s'adapte à la version galloise du Scrabble. L'intérêt étant qu'en Gallois, on dispose notamment des caractères L (qui vaut 1 point) et LL (qui en vaut 5), D (1 point) et DD (1 point aussi)  ...[‡]

†. Par exemple : $v_2(12) = 2$, $v_3(12) = 1$, $v_5(12) = v_7(12) = 0$.

‡. Pour plus de détails, voir ici : https ://fr.wikipedia.org/wiki/Lettres_du_Scrabble‡Gallois (points dans la version galloise du Scrabble).

Exercice 71. — Résumé de texte en 200 mots +/- 10%...

On se propose dans cet exo de coder quelques fonctions "simples" sur les chaînes de caractères.

1/ Proposer une fonction `test(résumé, caractère)` prenant en argument une chaîne de caractères appelée `résumé` et un caractère nommé `caractère`, et retournant True si ce caractère est présent dans le résumé.

2/ On souhaite créer une fonction `positioncaractère` prenant en argument une chaîne de caractère appelée `résumé` et un caractère, et renvoyant la liste des positions où apparaît le caractère dans le résumé.

Par exemple : `positioncaractère('Quel beau texte', 'u')` doit retourner `[1,8]`.

Compléter le code ci-dessous :

```
def positioncaractère(résumé, carac):
        ListPos =[]
        for i in range(len(résumé)):
        # A compléter

        return ListPos
```

3/ En s'inspirant des fonctions précédentes, proposer une fonction `nombremots` prenant en argument une chaîne de caractères appelée `résumé` et renvoyant le nombre de mots de cette chaîne de caractères.

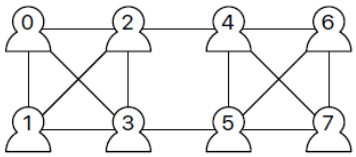
Exercice 72. — Réseaux sociaux (extrait de X-ENS MP/PC 2016)

Structure de données. Nous supposerons que les individus sont numérotés de 0 à $(n-1)$, où n est le nombre total d'individus. Nous représenterons chaque lien d'amitié entre deux individus i et j par une liste contenant leurs deux numéros dans un ordre quelconque, c.-à-d. par la liste `[i,j]` ou par la liste `[j,i]` indifféremment.

Un réseau social R entre n individus sera représenté par une liste `reseau` à deux éléments où :

- `reseau[0]` = n contient le nombre d'individus appartenant au réseau
- `reseau[1]` = la liste non-ordonnée (et potentiellement vide) des liens d'amitié déclarés entre les individus

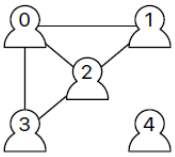
La figure 1 ci-dessous donne l'exemple d'un réseau social et d'une représentation possible sous la forme de liste. Chaque lien d'amitié entre deux personnes est représenté par un trait entre elles.



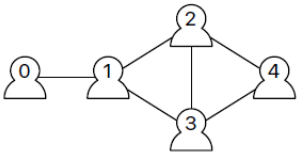

```
reseau = [ 8,
           [ [0,1], [1,3], [3,2], [2,0], [0,3], [2,1], [4,5],
             [5,7], [7,6], [6,4], [7,4], [6,5], [2,4], [5,3] ]
         ]
```

Figure 1. Un réseau à 8 individus ayant 14 liens d'amitié déclarés et une de ses représentations possibles en mémoire sous forme d'une liste [Nombre d'individus, liste des liens d'amitié].

Question 1. Donner une représentation sous forme de listes pour chacun des deux réseaux sociaux ci-dessous :



Réseau A



Réseau B

Question 2. Ecrire une fonction `creerReseauVide(n)` qui crée, initialise et renvoie la représentation sous forme de liste du réseau à n individus n'ayant aucun lien d'amitié déclaré entre eux.

Question 3. Ecrire une fonction `estUnLienEntre(paire,i,j)` où `paire` est une liste à deux éléments, et i et j sont deux entiers, et qui renvoie `True` si les deux éléments contenus dans `paire` sont i et j dans un ordre quelconque ; et renvoie `False` sinon.

Question 4. Ecrire une fonction `sontAmis(reseau,i,j)` qui renvoie `True` s'il existe un lien d'amitié entre les individus i et j dans le réseau `reseau` ; et renvoie `False` sinon.

Question 5. Ecrire une procédure `declareAmis(reseau,i,j)` qui modifie le réseau `reseau` pour y ajouter le lien d'amitié entre les individus i et j *si ce lien n'y figure pas déjà*.

Question 6. Ecrire une fonction `listeDesAmisDe(reseau,i)` qui renvoie la liste des amis de i dans le réseau `reseau`.