TP n°2 : parfaire sa maîtrise des itérations...

1 Quelques manipulations sur les nombres

Exercice 1 (Suite de Fibonacci) On note $(F_n)_{n\in\mathbb{N}}$ la suite des nombres de Fibonacci définie par :

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

Écrire une fonction `fibo(n)` qui prend en argument un entier naturel n et renvoie le nombre de Fibonacci F_n. Par exemple, `fibo(6)` renverra 8.

Exercice 2 Écrire un script qui calcule le plus entier n dont la somme des diviseurs dépasse 100.

Exercice 3 On pose pour $n \in \mathbb{N}^*$, $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$. Écrire un script qui calcule le plus petit entier n tel que $H_n \geqslant 10$.

Exercice 4 (Une somme double) Calculer avec Python les sommes

$$\sum_{i=1}^{50}\sum_{j=1}^{20}(i+j)i \quad \text{et} \quad \sum_{i=1}^{50}\sum_{j=1}^{i}(i+j)i.$$

Combien d'additions sont effectuées ?

Exercice 5 (Test de primalité)

1. Écrire une fonction booléenne `est_premier(n)` qui prend en argument un entier naturel n et renvoie True si n est premier et False sinon.
2. Estimer le nombre de divisions nécessaires dans le pire des cas.
3. Tester avec les nombres de Mersenne[1] suivants : $n = 2^p - 1$ avec $p = 18, 19, 31$. Si le temps de calcul pour $p = 31$ est trop long, améliorer votre algorithme en diminuant le nombre de divisions nécessaires. 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.
4. Écrire une fonction `premier_suivant(n)` qui prend en argument un entier naturel n et qui renvoie le plus petit nombre premier $p \geqslant n$.

2 Les chaînes de caractère sont aussi des objets itérables...

Exercice 6 Écrire une fonction `poids(mot)` qui prend en argument un mot (une chaîne de caractère) et renvoie la somme des codes ascii de ses lettres. Par exemple si `mot` est la chaîne `'buk'`, `poids(mot)` renverra $98 + 117 + 107 = 322$. On rappelle que la fonction `ord` renvoie le code ascii d'un caractère. Par exemple `ord('b')` renvoie 98

1. Les nombres de Mersenne fournissent des exemples de très grands nombres premiers. Attention, ils ne sont pas tous premiers, si $M_p = 2^p - 1$ est premier, alors p est premier. La réciproque est fausse. Le test de Lucas est un algorithme extrêmement efficace pour tester la primalité des nombres de Mersenne.

Exercice 7 (Verlan) Écrire une fonction `verlan(mot)` qui prend en argument un mot et renvoie le mot écrit à l'envers. Par exemple si `mot` est la chaîne `'volley'`, `verlan(mot)` renverra `'yellov'`. Attention, pour cette question on interdit toute utilisation de «slicing» (pour ceux qui connaissent). On rappelle que l'opérateur de concaténation est l'opérateur `+`.

Exercice 8 (Dessiner des rectangles) (tiré du site France IOI)

1. Écrire une procédure qui prend en paramètre un caractère et deux entiers, et qui affiche un rectangle rempli du caractère fourni, dont le nombre de lignes et de colonnes sont les entiers fournis. Appeler ensuite cette fonction pour afficher un rectangle de 2 lignes de 10 caractères 'X', puis un rectangle de 3 lignes de 5 caractères 'O'. On devra alors obtenir

   ```
   XXXXXXXXXX
   XXXXXXXXXX
   OOOOO
   OOOOO
   OOOOO
   ```

2. Même question mais cette fois-ci avec un triangle du type ci-dessous.

   ```
      *
     ***
    *****
   *******
   ```

On pourra utiliser dans les `print` comme deuxième argument `end=''` qui évite les sauts de ligne. Par exemple, la suite d'instructions

```
print('toto', end ='')
print('riri')
print('mumu')
```

affiche

```
totoriri
mumu
```

3 Pour les plus rapides...quelques défis «eulériens»

Exercice 9 (Project Euler, Problem 1) If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

Exercice 10 (Project Euler, Problem 16) $2^{15} = 32768$ and the sum of its digits is $3+2+7+6+8 = 26$.

What is the sum of the digits of the number 2^{1000} ?