

Python (Boucles)

Kais Klai

`kais.klai@lipn.univ-paris13.fr`

Dans ce qui suit, et pour chaque exercice, vous devez d'abord écrire l'algorithme avant de le traduire en Python. Aussi, assurez vous que les saisies son sécurisées. Par exemple, si vous êtes sensés récupérer un nombre positif au clavier, utilisez une boucle pour re-demander à l'utilisateur de redonner le nombre tant que la valeur donnée n'est pas positive.

Remarque: Sauvegardez tous vos programmes dans un répertoire spécifique de façon à pouvoir facilement les retrouver en cas de besoin dans le futur.

Echauffement

1. Ecrire un programme qui demande à l'utilisateur un message et un nombre, et qui affiche le message ce nombre de fois.
2. Ecrire un programme qui calcule la moyenne de N nombres saisis au clavier.
3. Ecrire un programme qui calcule le minimum de N nombres saisis au clavier. Le calcul se fait à la volée c-a-d.,i durant la saisie des valeurs.
4. Ecrire un programme qui affiche tous les nombres qui sont divisibles par 7 mais qui ne sont pas multiples de 5, et qui sont entre 2000 et 3200 (inclus). Ces nombres doivent être affichés sur une même ligne, et séparés par des virgules.
5. Ecrire un programme qui calcule une multiplication par additions successives.
6. Ecrire un programme qui calcule une division par soustractions successives.
7. Ecrire un programme qui calcule x^n par multiplications successives.
8. Ecrire un programme qui calcule $\sum_{i=1}^{i=n} x^i$. n et x sont donnés par l'utilisateur.
9. **Nombres premiers**
 - Ecrire un programme qui détermine si un nombre donné par l'utilisateur est *premier*. Un nombre entier est dit premier s'il a exactement deux diviseurs distincts : 1 et lui même.
 - Ecrire un programme qui affiche tous les nombres premiers inférieurs à 1000
10. **Nombres parfaits**
 - Ecrire un programme qui détermine si un nombre donné par l'utilisateur est *parfait*. Un nombre entier est dit parfait s'il est égal à la somme de ses diviseurs (excepté lui même). Par exemple, 9 n'est pas un nombre parfait car 9 n'est pas égal à $1 + 3$, mais 6 est un nombre parfait puisque $6 = 1 + 2 + 3$.
 - Ecrire un programme qui affiche tous les nombres parfaits inférieurs à 1000

11. **pgcd**

Ecrire un programme qui calcule le *plus grand commun diviseur* (pgcd) de deux nombres entiers par soustraction successives :

$$gcd(a, b) = \begin{cases} pgcd(a - b, a) & \text{si } a > b \\ pgcd(a, b - a) & \text{si } b > a \\ a & \text{si } a = b \end{cases}$$

Note: On suppose que a et b sont deux nombres entiers positifs.

12. **Factorielle d'un nombre**

En mathématique, la factorielle d'un entier positif n , noté $n!$, est le produit de tous les entiers positifs $\leq n$. Par exemple, $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$. Notez que, par convention, $0!$ est 1.

Ecrire un programme qui calcule la factorielle d'un nombre entier positif donné par l'utilisateur.

Séries mathématiques

- Ecrire un programme qui calcule la valeur des séries suivantes à un rang p donné :

$$\text{Srie } U_n \begin{cases} U_0 = 4.5 \\ U_{n+1} = 2.U_n - 3 \end{cases}$$

$$\text{Sries } W_n \text{ et } X_n \begin{cases} W_0 = 2 \\ X_0 = -1 \\ W_{n+1} = W_n + X_n \\ X_{n+1} = X_n - W_n \end{cases}$$

Le nombre mystère

- Nous voulons que l'utilisateur devine un nombre entre 1 et 10000 choisi au hasard par l'ordinateur. L'utilisateur procède par des essais successifs : pour chaque valeur introduite, l'ordinateur affiche un message précisant si la valeur introduite est supérieure ou inférieure au nombre mystère. Le jeu continue jusqu'à ce que l'utilisateur introduise la bonne valeur.

Note: Pour obtenir un nombre aléatoire, vous pouvez utiliser la fonction `randint(start, end)` (bibliothèque *random*). Cette fonction retourne une valeur entre *start* et *end*.

- Ecrire un programme qui permet à l'utilisateur de deviner le *nombre mystère*.
- Changer le programme précédent pour qu'il s'arrête dans deux cas : (1) L'utilisateur devine le nombre mystère, (2) L'utilisateur effectue 10 essais sans trouver le nombre mystère.

Décimal à l'infini

- On cherche à utiliser la méthode de division à la main apprise il y a bien longtemps pour calculer le résultat de la division de deux entiers, sachant que ce résultat peut être un nombre à virgule. La méthode permet de calculer un nombre infini de décimales.

Rappel : on divise un nombre entier p appelé dividende, par un nombre entier q appelé diviseur. On peut réaliser la division entière de p par q : $p = b.q + r$. b est la partie entière de p/q , r est le reste et donc est compris entre 0 et $q - 1$. $p/q = (bq + r)/q = b + (r/q)$. r/q est compris entre 0 et 1 (plus petit que 1), c'est donc la partie décimale de p/q , que nous nommerons d . La méthode utilisée consiste simplement à extraire un par un les chiffres décimaux de d .

En multipliant d par 10, on "fait passer" le premier chiffre après la virgule devant cette virgule. Or, faire $10.d$ c'est faire $10r/q$; donc il suffit de multiplier le reste de la division précédente (r) par 10 et de diviser à nouveau par q (division entière). Le quotient de cette division entière est le chiffre décimal recherché, le reste de cette division est de nouveau traité pour extraire le chiffre suivant ...

Les nombres affichés sont des nombres rationnels (de l'ensemble \mathbb{Q}). Un nombre rationnel a-t-il forcément un nombre fini de décimales ? On fixe donc une condition d'arrêt possible au calcul : 1000 décimales ont été calculées. Quelle peut être l'autre condition d'arrêt du calcul?

Ecrivez un programme qui saisit deux entiers et affiche le résultat de la division en utilisant la méthode décrite ci-dessus. Exemple de résultat obtenu par la division de 803 Par 619. Ceci vous renseigne sur le type de présentation à donner.

```
803/619=1,29725363489499192245557350565428109854604200323101777059773828
756058158319870759289176090468497576736672051696284329563812600969305331
1793214862681744749596122778675282714054927302100161550888529886914378029
07915993537964458804523 424878836833602584814216478190630048465266558966 ...
```

1 Triangle

1. Ecrire un programme qui affiche un triangle avec les caractère * sur un nombre donné de lignes. Par exemple, le triangle suivant est affiché sur 5 lignes:

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

2. Maintenant, sous cette forme ...

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

```
**  
*
```

3. Maintenant, sous cette forme ...

```
      *  
    * - *  
  * - * - *  
* - * - * - *  
* - * - * - * - *
```

4. Et maintenant, sous cette forme ...

```
* - * - * - * - *  
* - * - * - *  
* - * - *  
* - *  
*
```

Boucles et caractères

- Ecrire un programme qui affiche les lettres de l'alphabet en majuscule et leurs codes de la table ASCII. Faites de même pour les lettres minuscules.
- Ecrire un programme qui compte et qui affiche le nombre de voyelles et le nombre de consonnes dans une chaine de caractères donnée par l'utilisateur.
- Ecrire un programme qui compte et qui affiche le nombre de lettres minuscules et le nombre de lettres majuscules dans une chaine de caractères donnée par l'utilisateur.