TP2: boucles

Le document de référence doit être apporté en TP. Les exercices 1 à 8, incontournables, doivent être parfaitement traités en séance de TP. Les exercices suivants peuvent être traités en fin de séance ou à la maison, pour s'entraîner. Tous ces exercices sont susceptibles d'être posés en DS.

Créer un dossier "TP2" dans votre répertoire personnel. Pour l'exercice 1, dans Pyzo, enregistrer un fichier nommé "exo1.py" dans le dossier "TP2". Pour l'exercice 2, dans Pyzo, enregistrer un fichier nommé "exo2.py" dans le dossier "TP2"..... Pour chaque exercice, le fichier python sera régulièrement sauvegardé et éxécuté intégralement avec la commande "Run file as script" du menu Run (Ctrl+Shift+E). Les instructions d'affichage seront saisies sur le fichier python. Au fur et à mesure de l'avancée de l'exercice, les instructions devenant inutiles et gênantes pourront être désactivées en utilisant le caractère #.

Exercice 1. *Recopier et faire éxécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.*

$i = 0$	$i = 2 * i$
$print(i)$	$print(i)$
$i = 5$	$i = i + 1$
$print(i)$	$print(i)$

Exercice 2. *Recopier et faire éxécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.*

$(x, y) = (2, 3)$	$(x, y) = (y, x + y)$
$print(x)$	$print(x, y)$
$print(y)$	$(x, y) = (y, x)$
$print(x, y)$	$print(x, y)$

Exercice 3.

1. *Faire afficher les carrés des entiers compris entre* 1 *et* 100.
2. *Faire afficher les carrés des entiers pairs compris entre* 1 *et* 100 *(en utilisant* $range(.., .., ..)$*).*
3. *Faire afficher les carrés des entiers impairs compris entre* 1 *et* 100 *(en utilisant* $range(.., .., ..)$*).*

Exercice 4. *Dans cet exercice, on donnera deux réponses à chaque question.*

1. *Faire afficher les triplets d'entiers* $(a, b, a \times b)$ *où* a *et* b *sont compris entre* 1 *et* 9.
2. *Faire afficher les triplets d'entiers* $(a, b, a \times b)$ *où* a *et* b *vérifient* $1 \leq a \leq b \leq 9$.

Exercice 5. *Déterminer le plus petit entier* n *tel que* $3^n > 123456789$. *Faire afficher cet entier. On devra utiliser une boucle "Tant que".*

Idée générale pour un algorithme de calcul de somme

Soit $(x_i)_{i \in I}$ une famille finie de nombres. On souhaite calculer la somme des x_i avec $i \in I$. Pour cela, on va considérer les termes les uns après les autres et utiliser une variable s telle que, à chaque étape de l'algorithme, s est la somme des termes considérés jusqu'à cette étape.

Variante
"initialisation avec un premier terme x_{i_0}"
Affecter à s la valeur x_{i_0}
Pour i variant dans $I \setminus \{i_0\}$:
    Réaffecter à s la valeur de $s + x_i$.

Variante
"initialisation sans premier terme"
Affecter à s la valeur 0
Pour i variant dans I :
    Réaffecter à s la valeur de $s + x_i$.

Postcondition : s est la somme des x_i avec $i \in I$.

Exercice 6.

1. *Calculer* $\sum_{i=1}^{10} i^i$ *(somme des* i^i *pour* i *variant de* 1 *à* 10*) et faire afficher le résultat. (On utilisera la variante "Initialisation avec un premier terme".)*
2. *Calculer* $\sum_{1 \leq i \leq j \leq 10} i^j$ *(somme des* i^j *avec* i,j *entiers tels que* $1 \leq i \leq j \leq 10$*) et faire afficher le résultat. (On utilisera la variante "Initialisation sans premier terme".)*

En python, $float("inf")$ désigne le flottant $+\infty$.

Idée générale pour un algorithme de minimisation

Soit $(x_i)_{i \in I}$ une famille finie de nombres. On cherche à déterminer m le minimum des termes et un indice j tel que x_j est minimum (i.e. $x_j = m$). Pour cela, on va considérer les termes les uns après les autres et utiliser deux variables m et j telles que, à chaque étape de l'algorithme, m est le minimum des termes considérés jusqu'à cette étape et j est un indice tel que $x_j = m$.

Variante
"initialisation avec un premier terme x_{i_0}"
Affecter à m la valeur x_{i_0}
Affecter à j la valeur i_0
Pour i variant dans $I \setminus \{i_0\}$:
    Si $x_i < m$:
        Réaffecter à m la valeur x_i.
        Réaffecter à j la valeur i.

Variante
"initialisation sans premier terme"
Affecter à m la valeur $+\infty$
Affecter à j la valeur "vide"
Pour i variant dans I :
    Si $x_i < m$:
        Réaffecter à m la valeur x_i.
        Réaffecter à j la valeur i.

Postcondition : m est le minimum des termes et j est un indice tel que x_j est minimum.

Exercice 7.

1. *Exécuter l'instruction : from math import cos*
2. *Déterminer un élément* $j \in [[0, 99]]$ *tel que* $cos(j)$ *est minimum.*
   (On utilisera la variante "initialisation avec un premier terme".) Faire afficher j*.*
3. *Déterminer un couple* (k, l) *d'entiers avec* $0 \leq k \leq l \leq 99$ *tel que* $cos(cos(k)l)$ *est minimum.*
   (On utilisera la variante "initialisation sans premier terme".) Faire afficher (k, l)*.*

Idée générale pour un algorithme de dénombrement

Soit E un ensemble fini dont certains éléments vérifient une certaine propriété et pas les autres. On cherche à compter le nombre d'éléments de E vérifiant la propriété. Pour cela, on va considérer les éléments de E les uns après les autres et utiliser un compteur, i.e. une variable c telle que, à chaque étape, c est le nombre d'éléments considérés jusqu'à cette étape vérifiant la propriété.

On affecte à c la valeur 0
Pour x variant dans E :
    Si x vérifie la propriété :
        On réaffecte à c la valeur de $c + 1$.
Postcondition : c est le nombre d'éléments de E vérifiant la propriété.

Exercice 8. *On appelle diviseur strict d'un entier naturel* n *tout diviseur de* n *appartenant à* $[[2, n-1]]$*. Ecrire une fonction nommée "nbdivi" ayant pour argument* n *(entier) et retournant le nombre de diviseurs stricts de* n*. Faire afficher nbdivi(12).*

Exercice 9. *On considère la suite* $(u_n)_{n\in\mathbb{N}}$ *définie par* $u_0 = 2.0$ *et pour tout* $n \in \mathbb{N}$, $u_{n+1} = \frac{1}{2}(u_n + \frac{2}{u_n})$.

1. *Faire calculer* u_{10} *puis faire afficher le résultat.*
   Pour cela, utiliser une variable x *que l'on initialisera à la valeur de* u_0 *puis, pour* k *variant de* 0 *à* 9, *faire passer la valeur de* x *de* u_k *à* u_{k+1}.
2. *On admet que la suite* $(u_n)_{n\in\mathbb{N}}$ *est décroissante et converge vers* $\sqrt{2}$.
   De plus $1.414 < \sqrt{2} < 1.415$.
   Déterminer le plus petit entier n *tel que* $u_n < 1.415$ *puis faire afficher* n *et* u_n.
   Pour cela on utilisera deux variables x *et* n *et une boucle "Tant que* $x >= 1.415$*" ayant pour invariant "*x *a pour valeur* u_n*"*

Exercice 10. *On appelle triplet pythagoricien tout triplet d'entiers naturels non nuls* (a, b, c) *tel que* $a^2 + b^2 = c^2$. *Faire afficher les triplets d'entiers pythagoriciens* (a, b, c) *tels que* $a \leq b \leq c \leq 100$.

Exercice 11.

1. *Ecrire une fonction nommée valu ayant pour arguments deux entiers* a *et* b *(avec* $a \geq 2$ *et* $b \geq 1$*) et retournant le plus grand entier naturel* n *tel que* a^n *divise* b.
2. *Faire afficher* $valu(2, 60)$, $valu(3, 60)$, $valu(2, 180)$, $valu(3, 180)$, $valu(2, 270)$, $valu(3, 270)$.

Exercice 12.

1. *Ecrire une fonction "pgcd" ayant pour arguments deux entiers* a *et* b *tels que* $1 \leq a \leq b$ *et renvoyant le plus grand commun diviseur de* a *et* b. *On utilisera une boucle "Tant que".*
   Faire afficher $pgcd(126, 230)$.
2. *Ecrire une fonction "ppcm" ayant pour argument deux entiers* a *et* b *tels que* $1 \leq a \leq b$ *et renvoyant le plus petit commun multiple (supérieur à* 1*) de* a *et* b.
   On utilisera une boucle "Tant que".
   Faire afficher $ppcm(126, 230)$.

Exercice 13. *Ecrire une fonction nommée "facto" prenant pour argument* n *(entier) et retournant* $n!$. *Faire afficher facto(5).*

Exercice 14.

1. *Un entier* n *supérieur à* 2 *est premier ssi il n'admet pas de diviseur* $k \in [[2, n-1]]$. *Ecrire une fonction nommée "prem" ayant en argument* n *(un entier supérieur à* 2*) et renvoyant True si* n *est premier et False sinon.*
2. *Faire afficher les nombre premiers inférieurs à* 1000.
3. *Faire afficher le plus petit nombre premier* n *supérieur à* 1000.
4. *On appelle couple de nombres premiers jumeaux tous couple de nombres premiers de la forme* $(n, n + 2)$. *Ecrire une fonction "jum" ayant pour argument un entier* n *supérieur à* 2 *et renvoyant true si* $(n, n + 2)$ *est un couple de nombres premiers jumeaux et False sinon.*
5. *Faire afficher les couples de nombres premiers jumeaux inférieurs à* 1000.
6. *Faire afficher le plus petit couple de nombres premiers jumeaux supérieurs à* 1000.
7. *Un entier* $n \geq 2$ *est appelé nombre chanceux d'Euler ssi pour tout* $k \in [[0, n-2]]$, k^2+k+n *est premier. Ecrire une fonction nommée "chanceux" ayant pour argument* n *(un entier supérieur à 2) et renvoyant* $True$ *si* n *est un nombre chanceux d'Euler et* $False$ *sinon.*
8. *Faire afficher les nombres chanceux d'Euler inférieurs à* 100.

Exercice 15.

1. *Soit* $n \geq 2$. *On appelle diviseur strict de* n *tout diviseur de* n *compris entre* 1 *et* $n - 1$. *Ecrire une fonction nommée "nbdivi" ayant pour argument un entier* $n \geq 2$ *et retournant le nombre de diviseurs stricts de* n. *Faire afficher* $nbdivi(12)$.
2. *Ecrire une fonction nommée "sodivi" ayant pour argument un entier* $n \geq 2$ *et retournant la somme des diviseurs stricts de* n. *Faire afficher* $sodivi(12)$.

3. *Soit $n \geq 2$. On dit que n est dit parfait ssi n est égal à la somme de ses diviseurs stricts. Ecrire une fonction nommée "parfait" ayant pour argument un entier $n \geq 2$ et retournant True si n est parfait et False sinon.*
4. *Faire afficher les nombre parfaits inférieurs à 500.*
5. *Soit $a \geq 2$ et $b \geq 2$. On dit que a et b sont amicaux ssi la somme des diviseurs stricts de a est égale à b et la somme des diviseurs stricts de b est égale à a. Ecrire une fonction "ami" ayant pour arguments deux entiers $a \geq 2$ et $b \geq 2$ et renvoyant True si a et b sont amicaux et False sinon.*
6. *Faire afficher les couples (a, b) d'entiers amicaux tels que $a \leq b \leq 300$.*

Exercice 16.

1. *Ecrire une fonction nommée entrac ayant pour argument un entier naturel n et retournant le plus grand entier naturel p tel que $p^2 \leq n$. On utilisera une boucle "Tant que". Faire afficher $entrac(200)$.*
2. *Ecrire une fonction nommée entracbis ayant pour argument un entier naturel n et retournant le plus grand entier naturel p tel que $\sum_{k=1}^{p}(2k-1) \leq n$. On utilisera une boucle "Tant que". Faire afficher $entracbis(200)$.*

Remarque : Pour tout $p \in \mathbb{N}$, $\sum_{k=1}^{p}(2k-1) = p^2$ donc les deux fonctions aboutissent au même résultat. Ceci dit, la seconde fonction est moins couteuse en calculs.

Exercice 17.

1. *Ecrire une fonction nommée f ayant pour argument un entier n et renvoyant $\frac{n}{2}$ si n est pair et $3n+1$ si n est impair.*
2. *Faire afficher $f(14)$ et $f(9)$.*
3. *A l'aide d'une boucle, faire calculer $\underbrace{f(f(...f(f(}_{100 fois}1000))...))$*

   (l'entier obtenu en appliquant à 1000 la fonction f 100 fois de suite).
   Faire afficher le résultat.

Exercice 18. *En utilisant une boucle "Tant que", déterminer le plus petit entier $n \in \mathbb{N}^*$ tel que le reste de la division euclidienne de 2^n par 999 soit égal à 1. Faire afficher cet entier.*

Exercice 19. *On considère la suite de Fibonacci définie par récurrence double par $u_0 = 0$, $u_1 = 1$ et pour tout $n \in \mathbb{N}$, $u_{n+2} = u_{n+1} + u_n$. Faire calculer u_{100} puis faire afficher le résultat.*

Pour cela, utiliser un couple de variables (x, y) que l'on initialisera à la valeur de (u_0, u_1) puis, pour k variant de 0 à 98, faire passer la valeur de (x, y) de (u_k, u_{k+1}) à (u_{k+1}, u_{k+2}).