Lycée Louis-Le-Grand, Paris 2015/2016
MPSI 4 – Informatique commune
G. Dewaele, A. Troesch

TP n° 2 : Boucles

La boucle `while` a été introduite dans le TP précédent. La boucle `for` est une itération, possible sur les objets de tout itérable : `for i in iterable` donne successivement à `i` toutes les valeurs que contient `iterable`. Pour le moment, nous ne l'utiliserons qu'avec `range` (voir `help` ou poly). Ainsi, `for i in range(a,b)` fait répéter les instructions du bloc qui suit pour chaque valeur de i dans $[\![a, b-1]\!]$, dans l'ordre croissant.

Certains exercices de ce TP sont des reprises d'exercices du TP n° 1, que tout le monde n'avait pas eu le temps de terminer. Évidemment, vous pouvez vous dispenser de les refaire si vous les avez déjà traités.

Exercice 1 – Structures conditionnelles simples
Il est fréquent de devoir différencier l'action à effectuer suivant les cas. On utilise pour cette situation la structure conditionnelle

```
if bool:
    instructions
else:
    instructions
```

Le booléen `bool` est le plus souvent obtenu sous forme d'un test (`==`, `!=`, `>`, `>=`, `<`, `<=`, `is`, `in`*).*
Si la discussion porte sur plus de deux termes, on peut ajouter des tests intermédiaires grâce à `elif` *(abréviation de else if).*

Écrire dans un programme une fonction prenant en paramètre une année, renvoyant un booléen, égal à `True` si et seulement si l'année est bissextile. On demandera une année à l'utilisateur, et on lui affichera en réponse un texte disant si l'année est bissextile ou non.
On rappelle que depuis octobre 1582, une année n est bissextile si et seulement si n est divisible par 4, sauf si n est divisible par 100, mais pas par 400. On rappelle également qu'avant 1582, les années bissextiles étaient exactement les années multiples de 4.

Exercice 2 – Structures itératives conditionnelles
Les structures itératives (boucles) permettent de répéter un bloc d'instructions un grand nombre de fois. Nous n'étudions pour le moment que les boucles dont l'arrêt est conditionné par une condition, ou plutôt dont l'arrêt est conditionné par la non réalisation d'une certaine condition. Il s'agit de la boucle :

```
while bool:
    instructions
```

1. Soit pour tout $n \in \mathbb{N}^*$, $S_n = \sum_{k=1}^{n} \frac{1}{n}$. Écrire un programme déterminant la plus petite valeur de n pour laquelle $S_n > A$, A étant un réel entré par l'utilisateur. N'essayez pas votre programme avec des valeurs de A supérieures à 20.
2. On rappelle que si a et b sont deux entiers strictement positifs et r le reste de la division euclidienne de a par b, alors, si $r \neq 0$, le pgcd de a et b est égal au pgcd de b et r. En répétant cette opération jusqu'à obtenir un reste nul, on peut donc calculer le pgcd (c'est l'algorithme d'Euclide).
   Écrire un programme demandant à l'utilisateur deux entiers a et b, puis calculer et afficher leur pgcd. Qu'en est-il du ppcm ?

Exercice 3 – On définit la suite de Syracuse par $u_0 \in \mathbb{N}^*$, et

$$u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

On veut vérifier la propriété suivante : il existe un rang N tel que $u_N = 1$ (et à partir de ce rang, la suite boucle sur la séquence 4, 2, 1, 4, 2, 1, etc.). Écrire un programme demandant à l'utilisateur une valeur initiale u_0, calculant les

différents termes de la suite tant qu'ils ne sont pas égaux à 1, et affichant pour terminer la première valeur de N pour laquelle $u_N = 1$, ainsi que la plus grande valeur obtenue pour u_n.

Exercice 4 – Calculs de suites récurrentes, et de sommes
Les questions sont indépendantes.

1. Soit la suite définie par $u_0 = 0$, et pour tout $n \in \mathbb{N}$, $u_{n+1} = \sqrt{3u_n + 4}$.
   (a) Écrire un programme demandant à l'utilisateur un entier n et affichant tous les termes de la suite jusqu'à u_n. Que peut-on conjecturer quant à la convergence de cette suite ?
   (b) Écrire une fonction retournant le plus petit entier n pour lequel $u_n > 4 - \varepsilon$, où $\varepsilon > 0$. Que trouve-t-on pour $\varepsilon = 10^{-8}$?
2. Calculer $\displaystyle\sum_{n=0}^{1000} u_n$ où $u_0 = 1$ et $\forall n \geqslant 0$, $u_{n+1} = \dfrac{1}{u_n + 1}$.
3. Écrire un programme affichant les n premiers termes de la suite définie par $u_0 = 0$, $u_1 = 2$, pour tout $k \in \mathbb{N}$, $u_{k+2} = \sin u_k + 2\cos u_{k+1}$ (la valeur de n étant demandée à l'utilisateur). La suite $(u_n)_{n\in\mathbb{N}}$ semble-t-elle convergente ?
4. Soit $f(x, y) = x\cos y + y\cos x$. On définit une suite u_n par $u_0 = 0$, $u_1 = 1$, $u_{n+2} = f(u_{n+1}, u_n)$ si n est pair, et $u_{n+2} = f(u_{n+1}, u_{n-1})$ si n est impair. Afficher les premières valeurs de (u_n), jusqu'à l'indice N, l'entier N étant entré par l'utilisateur.

Exercice 5 – L'entier n étant donné par l'utilisateur, afficher les n premières lignes du triangle de Pascal. On veillera à aligner (suivant leurs unités) les valeurs situées sur une même colonne.

Exercice 6 – On appelle nombre parfait un nombre n dont la somme des diviseurs propres (donc différents de n) est égal à n. On appelle couple de nombres amicaux (m, n) un couple constitué de deux entiers strictement positifs distincts tels que la somme des diviseurs propres de chacun est égal à l'autre.

1. Écrire une fonction calculant la somme des diviseurs propres d'un entier n
2. Écrire une fonction recherchant tous les nombres parfaits inférieurs à un entier N donné, et calculant la somme des inverses des diviseurs de ces nombres. Quel commentaire faites-vous ?
3. Écrire une fonction recherchant tous les couples de nombres amicaux (m, n), tels que $m \leqslant N$, pour un entier N donné.

Exercice 7 – Écrire un calendrier perpétuel, valable à partir de l'avénement du calendrier grégorien le vendredi 15 octobre 1582 (lendemain du jeudi 4 octobre 1582). Le programme devra, étant donné un jour entré sous la forme AAAAMMJJ, donner le jour de la semaine correspondant.

NB : La date donnée correspond à la date de passage au calendrier grégorien pour les premiers pays ayant fait cette transition (en particulier l'Italie et la France). La plupart des pays ont fait la transition beaucoup plus tard, certains au début du XX$^{\text{e}}$ siècle (par exemple la Russie lors de la révolution de 1917). Certaines entités continuent encore aujourd'hui d'utiliser le calendrier julien, comme par exemple l'église orthodoxe russe.

Exercice 8 – Écrire, pour $(b, c) \in [\![2, 16]\!]^2$ un convertisseur de l'écriture d'un entier positif n de la base b à la base c. On fera entrer b, n (en base b) et c par l'utilisateur. On utilisera les lettres 'a', 'b', 'c', 'd', 'e' et 'f' pour désigner les chiffres suivant le chiffre 9 en bases 11 à 16.

Exercice 9 –

1. Écrire une fonction prenant en argument une base $b \in [\![2, 36]\!]$, deux chaînes de caractères représentant des entiers x et y en base b (les chiffres suivant 9 sont les lettres de l'alphabet dans l'ordre alphabétique), et réalisant l'addition en base b de ces deux entiers (sans faire de conversion de base). Les seules opérations autorisées sont `+ - * % //`.
2. De même pour le produit de deux nombres en base b.

Exercice 10 –

1. Écrire une fonction calculant le codage en complément à 2 sur 16 bits d'un entier de $[\![-32768, 32767]\!]$.
2. Réciproquement, écrire une fonction convertissant un code en complément à 2 sur 16 bit en une valeur numérique $n \in [\![-32768, 32767]\!]$.