exercices de T.C.d'I.

BASES DE NUMÉRATION

MPSI 1/MPSI 3

Table des matières

§1. Entiers naturels

Exercice 1

1) Donner la valeur (écrite en base dix) de $\overline{101001101}^{2}$ et de $\overline{1A7EF2}^{(16)}$.

2) Écrire le nombre 21 en base 2.

1) ?

Exercice 2

1) Rappeler les critères de divisibilité par 2, par 5 et par 10.

2) Quels sont les nombres d pour lequelles la divisibilité de x, par B, se résume à l'observation du chiffre des unités ?

3) Expliquer le critère de divisibilité par 4 en base 10. Puis en base 16.

1) ?

Exercice 3

1) Rappeler les critères de divisibilité par 3 et par 9.

2) Quels sont les nombres d pour lequelles la divisibilité de x, par B, se résume à l'observation de la somme des chiffres de x ?

1) ?

Exercice 4 — Soit B un entier naturel au moins égal à 2.

1) Écrire un programme `PoidsFaible(`x`, B = 10)` qui renvoie le chiffre des unités de x en base B. Puis un programme `Chiffre(`x`, `k`, B = 10)` qui renvoie le chiffre de rang k ($k = 0$ correspondant aux unités).

2) Écrire le programme `NombreDeChiffres(`x`, B = 10)`, puis le programme `SommeDesChiffres(`x`, B)`.

3) Écrire le programme `PoidsFort(`x`, B = 10)` qui renvoie le chiffre de poids fort de x en base B.

1) ?

Exercice 5 — On fixe dans cet exercice une base entière $B \geqslant 2$ et un entier $r \geqslant 1$. On considère la fonction φ définie sur $\{0\,;1\,;\ldots\,;B^{2r}-1\}$ par

$$\varphi(x) = \overline{y_{3r-1}y_{3r-2}\cdots y_{r+1}y_r}^{(B)},$$

où $\overline{y_{4r-1}\cdots y_1y_0}^{(B)}$ est l'écriture en base B de x^2.

1) Que peut-on dire du nombre de chiffres de x, s'il est dans le domaine de définition de φ ?

2) Et que peut-on dire du nombre de chiffres de son carré ?

3) Programmer la fonction φ en Python, et la tester sur différentes valeurs.

4) En déduire un programme `FabriquerGénérateur(r, init = None, B = 10)` qui renvoie une fonction `G()` produisant des nombres pseudo-aléatoires, en utilisant la formule de récurrence $x_{n+1} = \varphi(x_n)$. Le paramètre `init` représente la valeur de départ x_0 (qu'on réduira au besoin modulo B^{2r}). Si elle n'est pas indiquée, on choisira x_0 en fonction de l'heure (en expliquant le bricolage effectué).

```
from time import time as Maintenant
```

1) ?

Exercice 6 — La réponse ultime à la grande question sur la vie, l'univers et tout le reste est le nombre 42. Soit $\mathrm{B} \geqslant 5$ une base de numération ; on dit que l'écriture $\overline{x_{r-1} \dots x_1 x_0}^{(\mathrm{B})}$ « contient » 42 lorsqu'il existe un indice i tel que $x_i = 2$ et $x_{i+1} = 4$.

1) Pourquoi prend-t-on B au moins égal à 5 dans cet exercice ?

2) Certains prétendent que $42 = 9 \times 6$. Existe-t-il une base B dans laquelle ceci est vrai ?

3) Écrire un programme `Apparitions(x, B = 10)` qui calcule le nombre d'apparition de 42 dans l'écriture en base B de x.

4) Déterminer le plus petit nombre x tel que 42 apparaît (au moins) 42 fois dans l'écriture décimale de x^x.

5) Le plus petit entier naturel dans l'écriture décimale duquel apparaît au moins 42 fois 42 est

$$u_1 = \frac{100^{42} - 1}{99} \times 42.$$

Appelons u_2, u_3, etc. les suivants. Justifier l'expression de u_1, puis écrire un programme qui calcule u_n.

1) ?

Exercice 7 — Pour chaque entier $n \geqslant 1$ on note x_n le n-ième nombre sans chiffre 9 (dans son écriture décimale). Ainsi $x_1 = 1$, $x_2 = 2$, $\dots$, $x_8 = 8$, $x_9 = 10$, etc.

1) Qui sont x_{20} et x_{100} ?

2) Calculer x_{9^r} pour tout entier r.

3) En déduire un programme `SansNeuf(n)` qui calcule et renvoie la valeur de x_n.

4) En déduire un programme `SommeDesInversesSansNeuf(ε)` qui calcule une valeur approchée à ε près de la somme

$$\lim_{\mathrm{N}\to\infty} \sum_{n=1}^{\mathrm{N}} \frac{1}{x_n}.$$

1) ?

§2. Paquets

Exercice 8

1) On donne un entier naturel x représentant un certain nombre de secondes. Écrire le programme `VersHMS(`x`)` qui renvoie un triplet (h, m, s) indiquant la valeur de x en une combinaison d'heures, de minutes (obligatoirement moins de soixante) et de secondes (moins de soixante aussi).
2) Écrire le programme réciproque `DepuisHMS(`h`, `m`, `s`)`.
3) Soit `θ` un réel exprimant une mesure d'angle en degrés. Écrire les programmes de conversions `VersDMS(θ)` et `DepuisDMS(`t`, `d`, `m`, `s`)` qui passent de `θ` à la représentation en tours, degrés, minutes et secondes d'angles.

1) ?

Exercice 9 — Soit x un rationnel positif. On cherche à décomposer x en une somme de fractions de numérateur 1, et ayant toutes des dénominateurs distincts :

$$x = \frac{1}{q_0} + \frac{1}{q_1} + \cdots + \frac{1}{q_{r-1}},$$

avec $0 < q_0 < q_1 < \cdots < q_{r-1}$.

1) Si $x = p/q$, quelle est la plus grande fraction $1/q_i$ inférieure ou égale à x, et vérifiant $q_i > q_{\min}$?
2) En déduire un programme `FractionsÉgyptiennes(`p`, `q`)` qui construit et renvoie une liste `[`q_0`, `q_1`, ..., `q_{r-1}`]` de dénominateurs convenables.
3) Si l'on utilise une stratégie gloutonne, c'est-à-dire si l'on prend à chaque étape une fraction $1/q_i$ aussi grande que possible, obtient-on toujours une décomposition ayant un nombre de termes minimal ?
4) Prouver, toujours avec la stratégie gloutonne, que le procédé finit toujours, c'est-à-dire que l'algorithme de la deuxième question trouve une dcomposition convenable pour toutes les fractions p/q.

1) ?

Exercice 10

1) Écrire un programme `Romains(`x`)` qui étant donné un entier naturel x entre 1 et 3899 construit une chaîne de caractères constituant une représentation valide de x en chiffres romains.

```
>>> Romains(3899)
"MMMDCCCLXXXIX"
```

2) Écrire le programme réciproque `Arabes(`s`)`.

1) ?

Exercice 11 — On considère un ensemble $\mathscr{S}$ d'entiers strictement positifs et contenant 1 : ses éléments sont les *pièces* d'un système monétaire. Si x est un entier naturel, on peut alors l'écrire d'au moins une manière

$$x = \sum_{i=0}^{r-1} x_i,$$

avec $x_i \in \mathscr{S}$ pour chaque indice i (il suffit de prendre $r = x$ et $x_i = 1$ pour tout indice i). L'ensemble $\mathscr{S}$ sera représenté par une liste, rangée dans l'ordre croissant.

1) Écrire un programme `Glouton(`x`, `$\mathscr{S}$`)` qui renvoie la liste `[`x_0`, `x_1`, ..., `x_{r-1}`]` obtenue en prenant, à chaque étape, la plus grosse pièce possible.

```
>>> Glouton(27, [1, 2, 5, 10, 50])
[10, 10, 5, 2]
```

2) Montrer sur un exemple que le nombre de pièces utilisées, lorsqu'on utilise cette méthode, n'est pas toujours minimal.

3) (*Difficile.*) Écrire un programme `DécompositionMinimale(`x`, `$\mathscr{S}$`)` qui calcule et renvoie une décomposition utilisant un nombre minimal de pièces.

1) ?

Exercice 12

1) Écrire le programme `Ératosthène(M)` qui renvoie la liste, rangée dans l'ordre croissant, de tous les nombres premiers inférieurs ou égaux à M.

2) Soit x un nombre pair au moins égal à quatre. Écrire un programme qui construit la liste de tous les couples (p, q) de nombres premiers impairs, avec $p \geqslant q$, tels que $x = p + q$.

3) Écrire un programme efficace `GoldbachFort(M)` qui construit et renvoie une liste `S`, de longueur M + 1, telle que pour tout $x \geqslant 4$, `S[`x`]` est le nombre de décompositions $x = p + q$ avec p et q premiers impairs et $p \geqslant q$. La conjecture de Goldbach (1742) affirme que `S[`x`]` n'est jamais nul.

4) Soit x un nombre impair au moins égal à neuf. Écrire un programme qui construit la liste de tous les triplets (p, q, r) de nombres premiers impairs, avec $p \geqslant q \geqslant r$, tels que $x = p + q + r$.

5) Écrire un programme efficace `GoldbachFaible(M)` qui construit et renvoie une liste `T`, de longueur M + 1, telle que pour tout $x \geqslant 9$, `T[`x`]` est le nombre de décompositions $x = p + q + r$ avec p, q et r premiers impairs et $p \geqslant q \geqslant r$. Le fait que `T[`$x$`]` n'est jamais nul est une conséquence de la conjecture de Goldbach (puisqu'alors $x - 3$ est un nombre pair au moins égal à six). Harald Helfgott en a publié une démonstration en 2013.

1) ?

§3. Nombres à virgule flottante

Exercice 13 — Quels sont les nombres réels qui, en base B, ont une écriture « qui se termine » ?

?

Exercice 14 — Soit B un entier au moins égal à deux.

1) Prouver que le nombre $\xi = \overline{0{,}x_{-1}x_{-2}x_{-3}\ldots}^{(\mathrm{B})}$ avec $x_{-1} = x_{-2} = x_{-3} = \cdots = \mathrm{B} - 1$ est égal à 1.

2) Plus généralement, montrer que les fractions B-adiques, c'est-à-dire les nombres de la forme x/B^e avec $x, e \in \mathbf{Z}$, ont exactement deux écritures en base B.

3) Réciproquement, prouver que l'écriture en base B des réels qui ne sont pas des fractions B-adiques est unique.

4) Pour être sûr qu'on a bien compris : quelle est l'autre écriture de 120 en base dix ?

1) ?

Exercice 15 — Dans cet exercice B est une base entière au moins égale à deux.

1) Soit a un entier entre 0 et $\mathrm{B} - 1$. Quel nombre s'écrit $\overline{0{,}aaaa\ldots}^{(\mathrm{B})}$?

2) Prouver qu'un réel est un nombre rationnel si et seulement si son écriture en base B est périodique à partir d'un certain rang.

1) ?

Exercice 16

1) Quelle est l'écriture scientifique, en base deux, de 1/10 ?

2) Notons x son arrondi à 53 chiffres significatifs. Est-ce que x est égal à, est plus petit que, ou bien est plus grand que 1/10 ?

3) Reprendre les mêmes questions avec 2/10 (dont on notera y l'arrondi) et 3/10 (dont on notera z l'arrondi).

4) Est-ce que $x + x$ est égal à y ? Est-ce que $x + x + x$ est égal à z ?

1) ?

§4. Bases non entières

Exercice 17 — Soit B un réel strictement plus grand que 1.

1) Soit x un réel strictement positif. Montrer qu'il existe un entier $r \in \mathbf{Z}$ et des entiers $x_i \in \{0\,;1\,;\ldots\,;\lceil \mathrm{B}\rceil - 1\}$ tels que

$$x = \lim_{\mathrm{N}\to\infty} \sum_{k=-\mathrm{N}}^{r-1} x_k \mathrm{B}^k.$$

2) Écrire le programme `Développement(`x`, B, N = 0)` qui construit et renvoie la liste `[`$x_{-\mathrm{N}}$`, ..., `x_0`, `x_1`, ..., `x_{r-1}`]`.

1) ?