RUDIMENTS D'ALGORITHMIE

§1. Structures conditionnelles

Exercice 1 — Écrire les programmes `Max_2(a, b)`, `Max_3(a, b, c)` et `Max_4(a, b, c, d)` qui renvoient le plus grand de leurs arguments.

Exercice 2 — Écrire le programme `Tri(a, b, c)` qui renvoie un triplet, constitué des trois valeurs a, b et c, rangées dans l'ordre croissant.

Exercice 3 — Écrire un programme qui donne le nombre de solutions, dans $\mathbf{R}$, de l'équation $ax^2+bx+c=0$, puis un autre qui donne le nombre de solutions, toujours dans $\mathbf{R}$, de $ax^3+bx^2+cx+d=0$. Pour cette dernière, on cherchera sur internet ou dans un livre la méthode (qui consiste à se ramener à $x^3+px+q=0$ puis à calculer $27q^2+4p^3$).

Exercice 4 — Écrire un programme qui à partir des coordonnées $(x_{\mathrm{A}}, y_{\mathrm{A}})$, $(x_{\mathrm{B}}, y_{\mathrm{B}})$, $(x_{\mathrm{C}}, y_{\mathrm{C}})$ détermine si le triangle ABC est aplati, équilatéral, isocèle, rectangle, isocèle rectangle, ou quelconque.

Exercice 5 — Écrire un programme qui détermine si l'on peut écrire un entier x comme la somme de quatre entiers consécutifs, et renvoie ces quatre entiers lorsque c'est le cas.

§2. Boucles inconditionnelles

Exercice 6 — Écrire un programme qui calcule la somme des entiers inférieurs ou égaux à n qui sont des multiples de 3 ou de 7.

```
>>> Somme(100)
?
```

Exercice 7 — Écrire les programmes qui calculent les sommes suivantes :

a) $0+1+3+6+10+15+21+\cdots+\dfrac{n(n+1)}{2}$,

```
>>> Somme_a(10)
?
```

b) $1-2+3-4+5-6+\ldots$ avec n termes,

```
>>> Somme_b(100)
?
```

c) $u_0+u_1+\cdots+u_{n-1}$ avec $u_{n+1}=u_n^2-1$,

```
>>> Somme_c(1.5, 10)
?
```

d) $1!+2!+3!+4!+\cdots+n!$.

Exercice 8 — Même exercice :

a) $1+x+x^2+x^3+\cdots+x^n$,

b) $x+x^2+x^4+x^8+\cdots+x^{2^n}$,

c) $x-\dfrac{x^2}{2}+\dfrac{x^3}{3}-\dfrac{x^4}{4}+\cdots+(-1)^{n-1}\times\dfrac{x^n}{n}$,

d) $x-\dfrac{x^3}{3}+\dfrac{x^5}{5}-\dfrac{x^7}{7}+\cdots+(-1)^n\times\dfrac{x^{2n+1}}{2n+1}$.

Exercice 9 — Écrire un programme qui calcule la somme de tous les nombres inférieurs (strictement) à 1 000 000 et qui s'écrivent sans le chiffre neuf.

Exercice 10 — Même chose pour les nombres qui s'écrivent sans le chiffre quatre.

Exercice 11 — Écrire un programme qui calcule la somme

$$1+11+111+1111+\ldots$$

jusqu'au nombre qui s'écrit avec n fois le chiffre un.

Exercice 12 — Écrire un programme qui calcule le nombre

$$123456789101112131415161718192021\ldots$$

s'écrivant avec tous les nombres jusqu'à n.

Exercice 13 — Écrire les programmes qui calculent les sommes suivantes :

a) $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\min(i,j)$,

```
>>> Somme_a(100)
?
```

b) $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}|i-j|$,

c) $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{i}j^i$,

```
>>> Somme_c(100)
?
```

d) $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\max(i,j,k)$.

```
>>> Somme_d(100)
?
```

Exercice 14 — Écrire un programme qui compte le nombre de triplets (i, j, k), avec $1 \leqslant i, j, k \leqslant n$, dont les trois composantes sont distinctes. Reprendre la question avec les quadruplets, puis les quintuplets.

```
>>> NombreTriplets(10)
720
>>> NombreQuadruplets(14)
24024
>>> NombreQuintuplets(19)
1395360
```

Exercice 15

a) Écrire un programme qui compte le nombre de points à coordonnées entières dans le cercle de centre $(0, 0)$ et de rayon R.

```
>>> NombreDePoints(7.5)
177
```

b) En déduire un programme qui calcule une valeur approchée de π.

§3. Boucles conditionnelles

Exercice 16

a) Soit B un réel strictement plus grand que 1, et soit $x > 0$. Justifier qu'il existe un unique entier $n \in \mathbf{Z}$ tel que $\mathrm{B}^n \leqslant x < \mathrm{B}^{n+1}$.

b) Dans quel cas n est-il positif ? Négatif ?

c) En déduire un programme `Log(`x`, B)` qui calcule et renvoie la valeur de ce n.

Exercice 17 — Montrer comment, avec le programme `Log(`x`, B)` précédent, obtenir une valeur approchée à 0,001 près de $\log_{10}(3)$.

Exercice 18 — Pour $n \geqslant 2$ on admet que

$$2 + \sum_{k=2}^{n} \frac{1}{k!} < e < 2 + \sum_{k=2}^{n} \frac{1}{k!} + \frac{1}{k \times k!},$$

ce qui donne un encadrement de e d'amplitude $1/(n \times n!)$. Écrire un programme `Encadrement(`ε`)` qui détermine le plus petit n tel que $1/(n \times n!) \leqslant \varepsilon$ et renvoie l'encadrement (a_n, b_n) correspondant ci-dessus.

Exercice 19 — Montrer comment obtenir une valeur approchée de $x = \ln(3)$ avec les exercices précédents.

Exercice 20

a) Écrire un programme qui calcule (sans utiliser les opérations sur les nombres à virgule) la partie entière de $\sqrt{x}$.

b) Écrire un programme qui compte le nombre de manières d'écrire un entier x comme somme de deux carrés, c'est-à-dire le nombre de couples (a, b), tels que $x = a^2 + b^2$. On pourra afficher tous ces couples à l'aide de l'instruction `print`.

c) En déduire un programme qui détermine tous les triangles rectangles dont les côtés ont des longueurs entières inférieures ou égales à un R donné.

Exercice 21

a) Écrire un programme qui teste si un entier x est une puissance (non triviale) de b, c'est-à-dire de la forme $x = b^e$ avec $e \geqslant 2$.

b) En déduire un programme qui teste si x est une puissance, c'est-à-dire de la forme $x = b^e$ avec $b, e \geqslant 2$.

c) Écrire un programme qui compte les puissances inférieures à 10 000.

```
>>> PuissancesInférieures_à_10000()
124
```