Structures de contrôle

1 Années particulières

Les jeux olympiques modernes sont organisés tous les quatre ans, depuis la première édition en 1896. Depuis 1924 ont également lieu des jeux olympiques d'hiver, tous les quatre ans jusqu'en 1992, puis à nouveau tous les quatre ans depuis 1994. Les guerres mondiales 1914-1918 et 1939-1945 ont vu l'annulation de ces événements.

On suppose que le nom `y` désigne un entier naturel correspondant à une année.

1. Proposer un programme indiquant si durant l'année désignée par `y` ont été organisés des jeux olympiques d'été (pour les années situées dans le futur, on pourra supposer que les jeux continuent d'être organisés avec une périodicité de quatre ans).

2. Faire de même pour les jeux olympiques d'hiver.

Pour s'assurer que les solstices et équinoxes ne changent pas de date dans l'année, il n'est pas possible d'avoir des années dont la durée est toujours la même. Aussi a-t-on introduit le principe d'années *bissextiles*, qui comptent une journée de plus en février.

Avant 1582, ces années étaient celles divisibles par 4. Après 1582, pour plus de précision, les années divisibles par 100 ne sont plus bissextiles, excepté si elles sont divisibles par 400 (2000 était bissextile, mais 1900 ne l'était pas).

3. Écrire un programme affichant si l'année désignée par `y` est bissextile ou non.

4. Si ce n'était pas encore le cas, réduire le programme précédent à un seul et unique test en utilisant des relations logiques.

2 Rendu de monnaie

On suppose que le nom `m` désigne une somme d'argent, exprimée en euros, sous la forme d'un entier (pas de centimes).

Écrire un programme indiquant comment obtenir cette somme avec des billets de 5, 10, 20, 50, 100 et 200 euros et des pièces de 1 et 2 euros, avec le moins de pièces et de billets possibles. Par exemple, si `m` vaut 543, le programme affichera :

```
2 billets de 200 euros
1 billet de 100 euros
2 billets de 20 euros
1 pièce de 2 euros
1 pièce de 1 euro
```

On pourra, dans un premier temps, ne pas se poser la question des pluriels et écrire « billet(s) » au lieu de « billet » ou « billets », puis améliorer le programme ensuite.

3 Calcul de $\sqrt{a}$

Une première méthode pour calculer la racine carrée d'un entier a (sans utiliser la fonction racine carrée, bien évidemment), consiste à chercher la limite de la suite suivante :

$$u_0 = 1 \qquad \text{et} \qquad u_{n+1} = \frac{1}{2}\left(u_n + \frac{a}{u_n}\right)$$

1. Écrire un programme qui calcule et affiche les vingts premiers termes de la suite. On pourra prendre par exemple $a = 2$ puis $a = 3$ et comparer les résultats obtenus avec les valeurs fournies par la fonction `sqrt`.

Une autre méthode permet d'obtenir des approximations rationnelles de plus en plus précises d'une racine carrée, grâce à deux suites couplées :

$$\begin{cases} u_0 = 1 \\ d_0 = 1 \end{cases} \qquad \text{et} \qquad \begin{cases} u_{n+1} = (a-1)\times u_n + a \times d_n \\ d_{n+1} = u_n + (a-1)\times d_n \end{cases}$$

2. Écrire un programme qui calcule et affiche u_k, d_k et u_k/d_k pour $1 \leq k \leq 20$, et vérifier que la suite de rationnels u_k/d_k tend bien vers $\sqrt{a}$.

4 Calcul du nombre π

Il existe énormément de méthodes permettant de calculer le nombre π, si souvent utile en mathématiques. De très nombreux mathématiciens ont proposé une ou plusieurs techniques pour ce faire. On présente ici quelques suites convergeant vers le nombre π.

Pour chacune de ces suites (selon votre rapidité), écrire un programme qui calcule et affiche le n-ième terme de la suite, et déterminer combien de chiffres sont corrects pour $n = 1$, $n = 3$, $n = 10$, $n = 30$ et $n = 100$. Quelles sont les méthodes les plus efficaces ? On pourra remarquer que certaines méthodes ne permettent pas d'obtenir autant de décimales qu'on le souhaite, même avec des n très grands, nous y reviendrons ultérieurement.

Pour comparaison, on rappelle la valeur de π arrondie à seize décimales :

$$\pi \simeq 3.1415926535897932$$

Méthode de Leibnitz (XVII^e) **:**

$$u_n = 4 \times \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}$$

Méthode de Leibnitz (bis) :

$$u_n = 8 \times \sum_{k=0}^{n} \frac{1}{(4k+1)(4k+3)}$$

Méthode d'Euler (XVIII^e) **:**

$$u_n = \sqrt{6 \times \sum_{k=1}^{n} \frac{1}{k^2}}$$

D'après les sommes de Riemann (XIX^e) **:**

$$u_n = 4 \times \sum_{k=1}^{n} \frac{n}{n^2+k^2}$$

Méthode de Machin (XVIII^e) **:**

$$u_n = 4 \times \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}\left(4\left(\frac{1}{5}\right)^{2k+1} - \left(\frac{1}{239}\right)^{2k+1}\right)$$

Méthode d'Archimède :

La méthode proposée par Archimède consiste à calculer les périmètres, notés p_n, des polygones à 6×2^n côtés inscrits dans un cercle de diamètre unité. Comme les polygones se rapprochent de plus en plus d'un cercle, leur périmètre tend vers π.

Pythagore a pu montrer (ce que vous pouvez retrouver à titre d'exercice de géométrie) que ces périmètres pouvaient être calculés itérativement par les suites couplées suivantes (les a_n correspondant à la distance entre un côté du polygone et le centre du cercle) :

$$a_0 = \sqrt{\frac{1}{4} - \frac{1}{16}} \quad , \quad a_{n+1} = \sqrt{\frac{a_n}{4} + \frac{1}{8}} \quad \text{et} \quad p_0 = 3 \quad , \quad p_{n+1} = \frac{p_n}{2a_{n+1}}$$

Á l'époque, Pythagore avait proposé une estimation de π en effectuant le calcul à la main jusque p_4 (soit un polygone à 96 côtés).

Méthode de Woon (XX^e) **:**

$$a_0 = 1 \quad , \quad a_n = \sqrt{1 + \left(\sum_{k=0}^{n-1} a_k\right)^2} \quad \text{et} \quad u_n = \frac{2^{n+1}}{a_n}$$

Méthode des fractions continues (XVII^e) **:**

$$u_1 = 2 + \frac{2}{1+1} \quad , \quad u_2 = 2 + \frac{2}{1+\frac{1}{\frac{1}{2}+1}} \quad , \quad u_3 = 2 + \frac{2}{1+\frac{1}{\frac{1}{2}+\frac{1}{\frac{1}{3}+1}}} \quad , \quad u_4 = 2 + \frac{2}{1+\frac{1}{\frac{1}{2}+\frac{1}{\frac{1}{3}+\frac{1}{\frac{1}{4}+1}}}}$$

Méthode de Brown (XX^e) **:**

Pour calculer u_n, on considère la suite $a_0 = n$ et où a_k est le plus petit entier supérieur ou égal à a_{k-1} divisible par $n-k$. On pose $u_n = \frac{n^2}{a_{n-2}}$.

Par exemple, pour $n = 10$, on a $a_0 = 10$, $a_1 = 18$ (divisible par $10-1=9$), $a_2 = 24$, $a_3 = 28$, $a_4 = 30$, $a_5 = 30$, $a_6 = 32$, $a_7 = 33$, $a_8 = 34$ et donc $u_{10} = \frac{10^2}{34} \simeq 2.941176$.

Méthode de Buffon (XVIII^e) **:**

Lorsque l'on laisse tomber une aiguille de longueur 1 sur un parquet dont les lattes sont infinies dans une direction et de largeur 1 dans l'autre, la probabilité que l'aiguille repose sur deux lattes (c'est-à-dire chevauche une rainure) est égale à $2/\pi$.

La fonction `random` du module `random` retourne un réel aléatoirement choisi dans l'intervale $[0,1]$.

Pour obtenir une estimation u_n de π, on réalise n expériences consistant à tirer aléatoirement la position de l'aiguille une fois à terre et on compte le nombre p de fois où l'aiguille repose sur deux lattes distinctes. On a alors $u_n = 2n/p$.

Bien évidemment, ce n'est pas ici une vraie suite puisque la valeur de u_n sera différente à chaque exécution du programme, mais plus n sera grand, plus les valeurs u_n devraient être proches de π.

5 Paires amicales

Une *paire amicale* est un couple de deux entiers `n` et `p` tels que la somme des diviseurs positifs de `n` est égale à `p` et la somme des diviseurs positifs de `p` est égale à `n`.

1. Proposer un programme indiquant si un nombre désigné par `n` fait partie d'une paire amicale.

2. Déterminer toutes les paires amicales d'entiers inférieurs à 10000.