TP 2bis : séquence, variables, conditionnelles (suite)

DEUG IF121 : Informatique Fondamentale

27–31 octobre 2003

Exercice 1 : *Second degré*

Une équation du second degré a la forme générale suivante : $ax^2 + bx + c = 0$, où a, b et c sont des nombres réels et $a \neq 0$. On rappelle que $\Delta = b^2 - 4ac$ est appelé discriminant de l'équation, et que l'équation a les solutions réelles

$$\frac{-b-\sqrt{\Delta}}{2a} \quad \text{et} \quad \frac{-b+\sqrt{\Delta}}{2a}$$

Écrire un programme qui lit les coefficients a, b et c d'une équation du second degré, et affiche les solutions, si elles existent. S'il n'y a pas de solution, le programme doit en informer l'utilisateur.

Lorsqu'il y a des solutions x_1 et x_2, vérifier si l'on a bien $ax_i^2 + bx_i + c = 0$. Si ce n'est pas le cas, pourquoi ?

Note : pour calculer une racine carrée, on dispose de la méthode `Math.sqrt`. Par exemple, `Math.sqrt(9)` vaut `3.0`. La méthode `Math.sqrt` opère sur un flottant de type `double` et renvoie un flottant de type `double`.

Exercice 2 : *Notation hexadécimale*

On rappelle que la notation hexadécimale d'un nombre entier est son écriture en base 16, à l'aide des chiffres de `0` à `9` et de `A` à `F`.

Écrire un programme qui lit un nombre de deux chiffres en base 16 et affiche le nombre décimal correspondant. Par exemple :

```
> java HexaVersDecimal
Entrez un nombre à deux chiffres en base 16 : B4
En base 10, ceci fait 180
```

Pour obtenir les chiffres tapés par l'utilisateur, on pourra soit utiliser la méthode `Deug.readChar` pour lire l'entrée caractère par caractère, soit utiliser `Deug.readLine` puis `Deug.charAt` (rappelons que `Deug.charAt(s, n)` renvoie le `n`^ème caractère de la chaîne `s`, en commençant à 0).

Exercice 3 : *Chiffres romains*

Écrire un programme qui affiche un nombre entier positif en chiffres romains (*cf.* amphi 2).

On rappelle la valeur des chiffres romains :

chiffre romain	I	V	X	L	C	D	M
valeur décimale	1	5	10	50	100	500	1000

On rappelle aussi les règles d'écriture des nombres en chiffres romains :

- Normalement, on écrit les symboles du plus grand au plus petit (il peut y avoir plusieurs M, plusieurs C, plusieurs X, plusieurs I), et leur valeur s'ajoute. Par exemple, 172 s'écrit CLXXII.
- Si on se trouve avec quatre symboles identiques à la suite, on utilise à la place la notation soustractive : un symbole placé avant un symbole plus grand se retranche à lui. Par exemple, 199 s'écrit CXCIX.

Exercice 4 : *Affichage d'une durée*

(a) Écrire un programme qui lit un nombre N de secondes et affiche le nombre de jours, d'heures, de minutes et de secondes que N représente. Par exemple, $N = 262839$ correspond à « 3 jours, 1 heures, 0 minutes et 39 secondes ».

(b) Modifier le programme précédent pour qu'il omette les unités auxquelles correspond le nombre 0, et qu'il accorde au singulier les unités auxquelles correspond le nombre 1. Par exemple, pour $N = 262\,839$, on doit obtenir « 3 jours, 1 heure et 39 secondes ».

Exercice 5 : *Comptage de secondes*

(a) Écrire un programme qui compte le nombre de secondes entre le 1$^{\text{er}}$ janvier à minuit et maintenant, c'est-à-dire le nombre de secondes déjà écoulées dans l'année. Par exemple, le 2 janvier à 17h40, 150 000 secondes se sont écoulées. Le programme lira la date et l'heure actuelle sous forme de 5 nombres (mois, jour, heure, minute et seconde).

(b) En ne tenant pas compte des années bissextiles, augmenter le programme précédent pour qu'il affiche le nombre de secondes écoulées depuis le 1$^{\text{er}}$ janvier 1970.

(c) Sachant que depuis 1970, les années bissextiles sont exactement les années multiples de 4, modifier le programme précédent pour qu'il affiche vraiment le nombre de secondes écoulées depuis le 1$^{\text{er}}$ janvier 1970.

Ce nombre est la manière dont les systèmes Unix (par exemple, les machines de la salle de TP) représentent la date et l'heure.

(d) À combien de secondes correspond le 9 septembre 2001 à 01h46m40s ?

(e) Écrire un programme qui effectue la conversion inverse : du nombre de secondes à l'écriture habituelle en année, mois, jour, heure, minute, seconde. On pourra passer par des étapes analogues aux *(a)*, *(b)*, *(c)*. À quelle date correspond la deux milliardième seconde ?