TD2 - exo "BOUM"
Correction à l'usage des étudiants

EXERCICE 1 *Écrire un algorithme capable de décompter une durée donnée sous la forme heures, minutes, secondes et d'afficher "BOUM" en fin de décompte.*

CORRECTION :

Il y a plusieurs approches :

1. La première consiste à décompter le nombre de secondes, et à appeler une fonction `convert`, qui convertit le nombre de secondes qu'il reste à écouler en heures, minutes, secondes, et qui affiche ce triplet.
2. La deuxième consiste à utiliser trois boucles imbriquées : la boucle la plus interne compte les secondes, et lorsque le compteur de secondes est à 0, le remet à 59. Cette boucle est elle-même imbriquée dans une boucle qui décompte les minutes, ...
3. La troisième approche consiste à décompter les secondes seulement si on n'a pas atteint 0, auquel cas il faut remettre le compteur de secondes et décompter le compteur de minutes, ... On va utiliser une boucle tant que et des tests à l'intérieur.

Dans tous les cas, on écrit soigneusement l'algo, et on teste sur une valeur pertinente, par exemple $2min10sec$. Comme on ne fait que des impressions, on choisit une action, de paramètres h, m, s entiers qui dénotent respectivement l'heure, la minute et la seconde de début du décompte.

Mise en œuvre :

Approche 1 On va décompter les secondes dans une variable locale entière, `nbSec`. Pour chaque valeur de `nbSec`, l'action fait un appel à la procédure `convertion-impression`, qui réalise la conversion vers le système (heure,min,sec), à l'aide de divisions entières.

```
Action decompte1(h,m,s)
  | D: h,m,s : entiers
  | L: nbSec : entier
  | nbSec ← 3600*s+60*m+h
  | Pour i de nbSec à 0 Faire
  |   | convertion-impression(i)
  | Fpour
FAction
```

Algorithm 1: decompte, v1

Pour `convertion-impression`, on utilise une procédure car on ne fait qu'imprimer.

```
Action convertion-impression(sec)
| D: sec :entier
| L: h,m,s : entiers
| h ← sec/3600
| m ← (sec - 3600*h)/60
| s ← sec - 3600*h - 60*m
| Imprimer(h," :",m," :",s)
FAction
```

Approche 2 Le compteur de secondes doit aller plus vite que le compteur de minutes, donc la boucle de secondes est englobée dans la boucle des minutes. On remarque qu'à la fin de la boucle tant que sur s, s vaut 0 donc on n'a pas besoin de tester cette valeur. Pour l'impression, il faut imprimer le triplet chaque fois que l'heure change, d'où les 3 appels.

```
Action decompte2(h,m,s)
| D: h,m,s : entiers
| Tq (h ≠ 0) faire
| | Tq m≠ 0 faire
| | | Tq s≠ 0 faire
| | | | s ←s-1 Imprimer(h,m,s)
| | | Ftq
| | | s ← 59
| | | m ← m-1
| | | Imprimer(h,m,s)
| | Ftq
| | m← 59
| | h← h-1
| | Imprimer(h,m,s)
| Ftq
| Imprimer("BOUM")
FAction
```

Algorithm 2: decompte, v2

Approche 3 Dans cette solution, une seule boucle tant que extérieure fait décroître l'heure, en distinguant suivant la valeur de la date précédente. On peut donc se contenter d'imprimer

une fois par boucle.

```
Action decompte(h,m,s)
    D: h,m,s : entiers
    Tq (h+m+s ≠ 0) faire
        Si s≠0 alors
            s← s − 1
        Sinon
            s← 59;
            Si m≠0 alors
                m← m-1;
            Sinon
                m←59;
                h← h − 1
            Fsi
        Fsi
        Imprimer(h,m,s)
    Ftq
    Imprimer("BOUM")
FAction
```

Algorithm 3: decompte, v3

■