

Feuille d'exercices 7 : les sous-programmes

Ecrire les algorithmes et les programmes pour résoudre les problèmes suivant, en utilisant la notion de sous-programme (ainsi que les instructions input et print).

Exercice 1 :

version 1 : calculer et afficher les 10 premiers termes de la table de multiplication de 4

version 2 : calculer et afficher les 10 premiers termes de la table de multiplication d'un entier donné par l'utilisateur

version 3 : calculer et afficher les termes i à j de la table de multiplication d'un entier donné par l'utilisateur, i et j étant donnés par l'utilisateur.

Exercice 2 : écrire

- un sous-programme qui calcule le minimum de 2 nombres
- un sous-programme qui calcule le maximum de 2 nombres
- un programme qui calcule et affiche le minimum et le maximum de 3 nombres

Exercice 3 : reprendre l'exercice « calculatrice » (feuille 1)

Exercice 4 : reprendre les deux exercices de la feuille 5 :

coder une chaîne de caractères

décoder une chaîne qui a été codée suivant le programme précédent

- généraliser l'exercice avec deux sous-programmes
- vérifier avec votre programme principal que la chaîne décodée est égale à la chaîne initiale.

Exercice 5 : écrire un sous-programme qui demande à l'utilisateur de saisir un réel compris entre 0 et 20, qui ré-itére sa demande tant que le réel n'est pas correct et qui renvoie le réel (lorsqu'il est correct). Généraliser avec un réel compris entre 2 bornes.

Exercice 6 : Une machine outil perd 10% de sa valeur chaque année. Ecrire un programme qui permet de calculer sa valeur cinq ans après son achat et qui affiche le nombre d'années nécessaires pour que la valeur soit divisée par deux.

Généraliser le programme pour que l'utilisateur puisse faire plusieurs simulations en faisant varier les valeurs 10%, cinq années, divisée par deux.

Exercice 7 : Une balle chute de 400 pixels ; à chaque rebond, la hauteur diminue de 10% ; afficher la hauteur du rebond tant qu'il est supérieur à 5 pixels ; afficher le nombre de rebonds.

Généraliser.

Exercice 8 : Conjecture de Syracuse

On part d'un entier n auquel on fait subir une transformation :

- si n est pair, on le divise par deux ;
- si n est impair, on le multiplie par 3, et ajoute 1.
- Puis, on recommence sur le résultat. Par exemple, en partant de $n=10$, on obtient :

10 5 16 8 4 2 1 4 2 1 etc.

Conjecture : quel que soit l'entier n , on finit par retomber sur 1.

a) écrire un programme qui permet de tester la conjecture pour un entier donné ; calculer en même temps sa durée de vol, c'est-à-dire le nombre d'entiers rencontrés avant de trouver 1.

b) généraliser ce programme pour tester tous les entiers dans un intervalle d'entiers. Dans cet intervalle, repérer l'entier qui a la plus grande durée de vol.