

TD2 - Algorithmique et C de base - Boucle WHILE

Objectifs

Conception et Implémentation C de programmes simples, utilisation des boucles «tant que». Premières fonctions.

1 Nano programmes C

EXERCICE 1 *Écrire un programme qui énumère les entiers de 1 à 100 avec une boucle while.*

EXERCICE 2 *Écrire un programme qui demande des entiers à l'utilisateur et s'arrête lorsque celui-ci a écrit 0.*

EXERCICE 3 *Écrire un programme qui demande un entier N à l'utilisateur et imprime successivement $N, N - 1 \dots 0$.*

2 Analyse/Conception

On ne traduira pas les algorithmes en langage C.

EXERCICE 4 *Écrire un algorithme capable de décompter une durée donnée sous la forme heures, minutes, secondes et d'afficher "BOUM" en fin de décompte.*

EXERCICE 5 *Concevoir un algorithme qui :*

- Choisit au hasard un nombre x compris entre 0 et 64.
- Demande à l'utilisateur un nombre y , dit "gagné" si $x = y$, et dans le cas contraire donne une indication "plus petit" ou "plus grand".
- Lorsque le nombre a été deviné, imprime le nombre de tentatives.

EXERCICE 6 *Écrire un algorithme permettant de calculer le reste de la division euclidienne de a par b , sans utiliser `div` ni `mod`.*

EXERCICE 7 (Syracuse) *Soit la suite définie par $u_0 = N$ (entier positif) et $u_{n+1} = \frac{u_n}{2}$ si u_n est pair et $u_{n+1} = 3u_n + 1$ sinon. La conjecture de Syracuse dit que cette suite passe par 1. Écrire un algorithme qui vérifie cette conjecture pour tous les N de \min à \max .*

EXERCICE 8 *Concevoir un algorithme qui calcule et affiche le plus petit entier n tel que*

$$\sum_{i=1}^n \left(\sum_{j=1}^i (i+j) \right) \geq 1000.$$

EXERCICE 9 *Soit u_n définie par $u_0 = 1515$ et $u_{n+1} = 3u_n + 42$ pour $n \in \mathbb{N}^*$. Écrire un algorithme qui calcule le premier n strictement supérieur à 1 million.*

3 Fonctions C

On écrira directement le code C :

EXERCICE 10 Soit la suite u suivante : $u_n = \begin{cases} A & \text{si } n = 0, \\ 2 + 3 * u_{n-1} & \text{sinon.} \end{cases}$

Écrire un programme complet qui imprime les 10 premiers termes de la suite pour $A = 3$.

EXERCICE 11 Écrire une fonction `int sommation(int n)` qui étant donné un entier $n > 0$ renvoie $\sum_{i=1}^n i$ (la somme des n premiers entiers).