

ECS1.1 - TP N° 3 : Boucles For et While - Structures If-Then

Attention : penser à changer le répertoire courant avant de sauvegarder vos fichiers.

Écrire des programmes qui généralisent les objectifs suivants :

Exercice 1 Algorithme de Syracuse

On choisit un entier naturel quelconque. S'il est pair, on le divise par 2, sinon on le multiplie par 3 et on ajoute 1. En répétant ce processus, on constate qu'au bout d'un certain nombre d'itérations, on aboutit à 1 (actuellement personne n'a su le démontrer).

Exemple : 7 ; 22 ; 11 ; 34 ; 17 ; 52 ; 26 ; 13. ...

- Terminer « à la main » la liste précédente.

- Écrire une fonction `syracuse(n)` qui affiche le nombre suivant le nombre n dans la liste de syracuse.

- Compléter ce programme de façon à lui faire effectuer la boucle complète jusqu'à arriver à 1.

Exercice 2 Fonction partie entière : `partieentiere(n)`

Déterminer la partie entière du réel n .

Améliorer le programme, en permettant d'entrer un réel négatif.

Exercice 3 Le nombre caché : `nombrecache()`

Version 1 :

À l'aide de la fonction `rand()` qui donne un nombre réel aléatoire entre 0 et 1, et en utilisant votre fonction `partieentiere`, faire choisir à l'ordinateur un nombre entier entre 1 et 1000.

Dans la même fonction, écrire une série d'instructions demandant à l'utilisateur d'entrer plusieurs fois de suite un nombre entier, en affichant à chaque fois si le nombre entier est trop grand ou trop petit, et ce jusqu'à le nombre soit deviné.

Ajouter un compteur d'essais et une mention correspondante lorsque le nombre est trouvé.

Version 2 :

L'utilisateur choisit dans sa tête un nombre entier entre 1 et 1000.

Écrire une suite d'instructions qui permettent à l'ordinateur de déterminer par une méthode de dichotomie, et en questionnant l'utilisateur, un intervalle $[a, b]$ tel que : $n \in [a, b]$ et $b - a < 1$.

Afficher alors le nombre deviné par l'ordinateur.

Exercice 4 Fonction `divisibilite7(n)`.

Affiche « oui » ou « non » selon que l'entier n est divisible par 7 ou non.