

TP - Premières boucles `while`

Objectif(s)

- ★ Comprendre le fonctionnement des boucles `while`
- ★ savoir utiliser les boucles
- ★ savoir choisir quel type de boucle utiliser

Remarque:

Les premiers exercices reprennent le TD. La fin du sujet propose de nouvelles questions.

Exercice 1 – Boucles `while` simples

Question 1

Ré-écrivez les fonctions `affiche_nombres` et `affiche_nombres_decroissants` en utilisant un `while` plutôt qu'un `for`.

Pour rappel, `affiche_nombres` prend en argument une valeur (`int`) n et affiche la valeur de n suivi de `:`, suivi tous les entiers de 1 à n , séparés par des espaces, tandis que `affiche_nombres_decroissants` affiche ces mêmes nombres dans l'ordre décroissant.

Faites des tests assez étoffés, notamment pour vous assurer de l'arrêt systématique de vos fonctions.

Question 2

Écrivez une fonction `affiche_carres` qui prend en argument une valeur (`int`) n et affiche tous les carrés parfaits inférieurs ou égaux à n . Par exemple, pour $n = 30$, elle devra afficher 1 4 9 16 25.

Question 3

Écrivez une fonction `affiche_carres_entre` qui prend en argument deux valeurs (`int`) $n < m$ et affiche tous les carrés parfaits compris entre n et m . Si $n \geq m$, on n'affichera rien. Par exemple, pour $n = 7$ et $m = 40$, elle devra afficher 9 16 25 36.

Question 4

Écrivez une fonction `saisie_valeur` qui demande des saisies de valeurs à l'utilisateur et retourne la première valeur saisie entre 0 et 3. Tant que l'utilisateur saisie des valeurs en dehors de l'intervalle $[0, 3]$, la fonction demande une nouvelle valeur.

Renforcement

Exercice 2 – Nombres premiers

Question 1

Écrivez une fonction `est_premier` qui renvoie 1 si un entier n passé en paramètre est premier, 0 sinon :

- avec une boucle `for` interrompue.
- avec une boucle `while`.

Question 2

Écrivez une fonction `premier_plus_petit_que` qui prend en entrée un paramètre n et affiche tous les nombres premier inférieurs à n .

Question 3

Modifiez votre fonction pour qu'en plus de l'affichage, elle retourne le nombre de nombres premiers trouvés.

Question 4

Modifiez votre fonction pour que les nombres affichés soient séparés par des virgules, et que l’affichage se termine par un point.

Exercice 3 – Manipulation des nombres

Dans cet exercice, on va manipuler des nombres entiers dans leur écriture décimale, à savoir en base 10, comme vous les écrivez d’habitude.

Question 1

Écrivez une fonction `premier_chiffre` qui prend en paramètre un nombre entier n , et qui retourne le premier chiffre de l’écriture décimale de n . Par exemple, pour 4729, la réponse est 4.

Question 2

Écrivez une fonction `nombre_miroir` qui retourne le nombre s’écrivant avec les mêmes chiffres dans l’ordre inverse. Par exemple, sur l’entrée 18374, la fonction retournera 47381.

Question 3

Écrivez une fonction `nb_chiffre_donne` qui prend en paramètre un nombre n et un chiffre c , et qui retourne le nombre de chiffres c utilisés dans l’écriture décimale de n .

Question 4

Écrivez une fonction `somme_des_chiffres` qui retourne la somme des chiffres d’un nombre entier positif ou nul passé en paramètre.

Question 5

Écrivez une fonction qui prend en paramètre deux nombres, n et s , ainsi qu’un chiffre c , et qui retourne combien de fois le chiffre c est utilisé dans l’écriture de l’ensemble des nombres de 1 à n dont la somme des chiffres vaut s .

Question 6

Utilisez la fonction précédente pour révéler le chiffre le plus fréquent dans les nombres de 1 à 1000 ayant pour somme des chiffres 9.

(Si vous avez fini tous les renforcements des TP précédents, allez donc vous intéresser à la section défi...)