

Premières boucles

1 Boucles for simples

1.1 Affichages de nombres

1. Écrivez une fonction `affiche_nombres` qui prend en argument une valeur (*int*) n et affiche tous les entiers de 1 à n , séparés par des espaces.
2. Écrivez une fonction `affiche_nombres_pairs` qui prend en argument une valeur (*int*) n et affiche tous les entiers positifs pairs plus petits ou égaux à n , séparés par des espaces.
3. Écrivez une fonction `affiche_nombres_decroissants` qui prend en argument une valeur (*int*) n et affiche tous les entiers positifs de n à 1, séparés par des espaces.

1.2 Compteurs

4. Écrivez la fonction `factorielle` qui calcule et retourne la valeur de $n!$ pour un entier n .
5. Écrivez une fonction `somme_impairs` qui calcule la somme des entiers impairs inférieurs à un paramètre n .

2 Boucles while

6. Ré-écrivez les fonctions `affiche_nombres` et `affiche_nombres_decroissants` en utilisant un *while* plutôt qu'un *for*.
7. Écrivez une fonction `affiche_carres` qui prend en argument une valeur (*int*) n et affiche tous les carrés parfaits inférieurs ou égaux à n .
8. 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.
9. 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.

3 Renforcement

10. 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*.
11. Écrivez une fonction `premier_plus_petit_que` qui prend en entrée un paramètre n et affiche tous les nombres premiers inférieurs à n .
12. Modifiez votre fonction pour qu'en plus de l'affichage, elle retourne le nombre de nombres premiers trouvés.
13. Modifiez votre fonction pour que les nombres affichés soient séparés par des virgules, et que l'affichage se termine par un point.

(Vous pouvez aussi reprendre les fonctions du TD précédent que vous n'auriez pas finies.)