

C - Les boucles `for` - Exercices 🍌

Voici 10 exercices progressifs en C pour apprendre à maîtriser les boucles **for**, adaptés à un cours débutant mais exigeant. Chaque exercice inclut un objectif clair et une consigne précise.

5TTR

6TTR

 Découverte

1) Affichage simple de nombres

Objectif : comprendre la structure de base d'une boucle `for`. **Consigne** : afficher les nombres de 1 à 10 sur une seule ligne, séparés par des espaces.

2) Compter à rebours

Objectif : travailler la décrémentation dans un `for`. **Consigne** : afficher les nombres de 10 à 1.

3) Afficher les nombres pairs

Objectif : utiliser une condition dans une boucle `for`. **Consigne** : afficher tous les nombres pairs entre 1 et 50.

4) Somme des N premiers nombres

Objectif : accumuler une valeur dans une variable. **Consigne** : demander un nombre `n` à l'utilisateur et calculer la somme des nombres de 1 à `n`.

5) Compter les multiples

Objectif : analyser une séquence. **Consigne** : afficher tous les multiples de 7 entre 1 et 200, et afficher ensuite combien il y en avait.

6) Table de multiplication

Objectif : produire une sortie structurée. **Consigne** : demander un entier et afficher sa table de multiplication (1 → 10).

7) Calcul de factorielle

Objectif : utiliser une variable multiplicative. **Consigne** : demander un nombre entre 1 et 12 et calculer `n!` (factorielle de n).

8) Identifier le minimum et le maximum

Objectif : parcourir plusieurs valeurs entrées par l'utilisateur. **Consigne** : demander 10 nombres et déterminer le minimum et le maximum.

9) Triangle d'étoiles (motif ASCII)

Objectif : utiliser une boucle `for` imbriquée. **Consigne** : afficher un triangle comme ceci (pour une hauteur `n`) :

```
*  
**  
***  
****
```

10) Nombres premiers dans un intervalle

Objectif : implanter une logique plus complexe avec boucles imbriquées. **Consigne** : afficher tous les **nombres premiers entre 1 et 200**.

Utiliser une boucle `for` externe pour parcourir les nombres et une boucle interne pour tester la divisibilité.