

Exercice 1 — Fonctions et conditionnelles (niveau facile)

Objectif : écrire des fonctions simples et les utiliser dans `main`.

Énoncé

1. Écrire une fonction `int estPair(int n)` qui retourne 1 si `n` est pair, 0 sinon.
2. Écrire une fonction `int max2(int a, int b)` qui retourne le maximum entre `a` et `b`.
3. Dans `main`, lire deux entiers, afficher si le premier est pair ou impair, puis afficher le maximum des deux.

Exercice 2 — Boucles et tableaux (niveau moyen)

Objectif : manipuler tableaux et boucles, faire une fonction qui calcule des résultats agrégés.

Énoncé

1. Écrire une fonction `int sommeTableau(int tab[], int n)` qui renvoie la somme des `n` éléments.
2. Écrire une fonction `double moyenneTableau(int tab[], int n)` qui renvoie la moyenne (double).
3. Dans `main`, lire un entier `n` ($1 \leq n \leq 100$), puis `n` entiers ; afficher la somme et la moyenne avec 2 décimales.
4. Bonus : écrire `int indiceMax(int tab[], int n)` qui renvoie l'indice du premier élément maximum.

Exercice 3 — Chaînes de caractères et fonctions (niveau intermédiaire)

Objectif : travailler avec `char[]`, fonctions de la librairie `<string.h>` et écrire algorithmes simples sur chaînes.

Énoncé

1. Écrire une fonction `void inverser(char s[])` qui inverse une chaîne (in-place).
2. Écrire `int estPalindrome(char s[])` qui renvoie 1 si la chaîne, sans tenir compte des espaces et de la casse, est un palindrome, 0 sinon.
3. Dans `main`, lire une ligne (utiliser `fgets`) et afficher la chaîne inversée et si elle est palindrome.

Palindrome : oui

Conseils : normaliser la casse avec `tolower`, ignorer les espaces/ponctuation si demandé.

Exercice 4 — Tableaux de chaînes + recherches (niveau avancé)

Objectif : manipuler tableaux de `char*`, effectuer des recherches et trier.

Énoncé

1. Lire `n` (≤ 50) puis `n` mots (chaînes sans espaces, max 50 caractères). Stocker dans un tableau `char tab[n][51]`.
2. Écrire une fonction `int rechercher(char tab[][51], int n, char cle[])` qui renvoie l'indice du mot (ou -1).