

■ Devoir Surveillé — Programmation en C

Durée : 2 heures — Aucun document autorisé

Partie 1 — Questions de cours (40 minutes)

1. Expliquer la différence entre **variable locale**, **globale** et **statique**.
2. Quelle est la différence entre un **pointeur** et une **référence** en C ?
3. Que se passe-t-il si on oublie le caractère `\'0` dans une chaîne ?
4. Donner les avantages et les risques liés à l'utilisation des **pointeurs**.
5. Quelle est la différence entre **malloc**, **calloc**, et **realloc** ?
6. Citer trois erreurs courantes liées à la manipulation mémoire.
7. Quelle est la différence entre un **fichier texte** et un **fichier binaire** ?
8. Pourquoi est-il important de toujours **vérifier le retour** d'un appel à `fopen()` ?
9. Donner le rôle du **Makefile** dans un projet C.
10. Quelle est la différence entre `==` et `=` en C ?

Partie 2 — Exercices d'application (1 heure)

Exercice 1 — Tableaux et pointeurs

Écrire une fonction **inverser()** qui prend en paramètre un tableau d'entiers et sa taille, et qui inverse l'ordre des éléments du tableau *en place* (sans utiliser de tableau auxiliaire).

Exercice 2 — Gestion mémoire

Écrire une fonction **allouer_et_initialiser()** qui alloue dynamiquement un tableau de N entiers et initialise chaque élément à la valeur de son indice. Afficher le contenu du tableau dans le main, puis libérer la mémoire.

Exercice 3 — Structures et passage par adresse

Définir une structure **Etudiant** contenant un nom, une note et un identifiant. Écrire une fonction **augmenter_note()** qui augmente la note de l'étudiant de 1 point si elle est inférieure à 20, en utilisant le passage par pointeur.

Exercice 4 — Fichiers

Écrire un programme qui crée un fichier *notes.txt* et y écrit la liste de 5 étudiants avec leurs notes. Puis lire ce fichier et afficher les étudiants ayant obtenu une note ≥ 10 .

Partie 3 — Mini-projet (20 minutes)

Objectif : Combiner les notions vues pour construire un petit programme complet.

Contexte :

On souhaite écrire un programme de gestion de stock d'articles. Chaque article possède un nom, un prix et une quantité. Le programme doit permettre :

- de saisir N articles au clavier ;
- d'afficher la liste des articles ;
- de calculer la valeur totale du stock (prix \times quantité) ;

- d'enregistrer les données dans un fichier *stock.txt*.

Indications :

Utiliser une structure **Article** et des fonctions dédiées pour la saisie, l'affichage et le calcul de la valeur totale. Penser à gérer les erreurs d'ouverture de fichier.

■ *Bonus (facultatif)* : afficher l'article le plus cher et celui avec la plus grande quantité.