

Fiche de Révision

⌚ Date de création	@10 décembre 2025 07:56
Ⓜ Matière	Structure des données / Langage C

■ 1. Bases de la programmation en C

- **Programme** : Suite d'instructions données à l'ordinateur pour exécuter des actions.
- **Langage machine** : L'ordinateur ne comprend que le binaire (0 et 1).
- **Langages de haut niveau** : C, C++, Java, Visual Basic... traduits en binaire par un compilateur.
- **Code source** : Instructions écrites par le programmeur dans un langage de haut niveau.
- **Compilateur** : Traduit le code source en binaire → produit un **exécutable (.exe)**.
- **Préprocesseur** : Traite les directives (#include, #define) avant compilation.
- **Fonction main()** : Point d'entrée du programme C.
- **Instruction** : Commande exécutée par l'ordinateur (ex. `printf("Hello World");`).

■ 2. Outils du programmeur

- **Éditeur de texte** : Écrire le code source (Bloc-notes, vi, IDE).
- **Compilateur** : Transformer le code source en binaire.
- **Débogueur** : Identifier et corriger les erreurs.
- **IDE** : Environnement complet (éditeur + compilateur + débogueur). Exemples : Code::Blocks, Visual C++, Xcode.

■ 3. Types de programmes

- **Programmes en console** : Interface texte (CMD sous Windows). Plus simples, utilisés pour l'apprentissage.

- **Programmes avec fenêtres** : Interface graphique (ex. Paint). Plus interactifs, mais plus complexes à programmer.
-

4. Introduction aux algorithmes

- **Algorithme** : Suite finie d'opérations pour résoudre un problème.
 - **Élaboration** : Décrire les données, les méthodes et prouver le bon fonctionnement.
 - **Complexité** : Temps de calcul et espace mémoire nécessaires.
-

5. Types et structures de données

- **Type de données** : Définit les valeurs possibles et les opérations autorisées (ex. int, float, char).
 - **Typage** : Association d'un objet à un ensemble de valeurs et d'opérations.
 - **Structure de données** : Manière d'organiser les données pour les utiliser efficacement.
 - **Structures du langage** : types primitifs (tableaux).
 - **Structures abstraites** : piles, files, listes.
-

6. Structures de données abstraites

- **Définition** : Spécification mathématique d'un ensemble de données et des opérations possibles.
 - **Structure dynamique** : Taille variable selon les besoins (ex. listes chaînées).
-

7. Classification des structures de données

- **Structures linéaires** : Organisation séquentielle (tableaux, listes, piles, files).
 - **Structures non linéaires** : Organisation hiérarchique ou en réseau (arbres, graphes).
-

8. Tableaux

- **Définition** : Ensemble d'éléments de même type, indexés et stockés en mémoire contiguë.
 - **Déclaration** : `int tab[10];` (tableau de 10 entiers).
 - **Accès** : `tab[i]` → élément à l'indice i.
 - **Dimension** : 1D (simple), 2D (matrice), etc.
-

9. Structures en C (Enregistrements)

◆ Définition

- Une **structure** regroupe plusieurs variables de types différents sous un même nom.
- Permet de modéliser des entités complexes (ex. étudiant, employé).
- Chaque élément est appelé un **champ**.

◆ Déclaration

```
Type identifiant_du_type = structure
    champ1 : type1
    champ2 : type2
    ...
Fin identifiant_du_type
```

```
struct Etudiant {
    char nom[20];
    char prenom[20];
    char sexe;
    float moyenne;
    long CNE;
};
```

◆ Création d'une variable

VAR identificateur : NomEnregistrement

```
struct Etudiant Et1;
```

◆ Accès aux champs

nom_enregistrement . nom_champ

```
Et1.nom = "Hanine";  
Et1.prenom = "Azedine";  
Et1.sex = 'M';  
Et1.moyenne = 13.25;  
Et1.CNE = 9997070890;
```

✓ Synthèse générale

- **Langage C** : Bas niveau, puissant, proche de la machine.
- **Programme C** : Code source → compilateur → exécutable.
- **Algorithme** : Suite d'opérations avec complexité temps/espace.
- **Structures de données** : Organisation efficace (tableaux, listes, piles, files, arbres, graphes).
- **Structures en C** : Regroupent plusieurs types dans une seule entité, utiles pour modéliser des objets complexes.

👉 Avec cette fiche, tu as un **support complet** pour réviser la programmation en C et les structures de données.