

Travaux Pratiques : Listes Chaînées en Langage C

DUT : IDSD

2024-2025

Objectifs pédagogiques

À l'issue de ce TP, l'étudiant devra être capable de :

- Comprendre le fonctionnement d'une liste simplement chaînée
- Implémenter les opérations de base : création, insertion, suppression, affichage
- Utiliser la mémoire dynamique via les pointeurs
- Comparer les listes chaînées aux tableaux
- Respecter des conventions de nommage normalisées

Règles de nommage obligatoires

Les conventions suivantes sont obligatoires pour garantir l'uniformité des rendus :

- Structure de nœud : `Node`
- Champs de la structure :
 - `int data` : donnée contenue dans le nœud
 - `Node* next` : pointeur vers le nœud suivant
- Pointeur principal de la liste : `head`
- Prototypes imposés :
 - `Node* create_node(int value);`
 - `void insert_at_head(Node** head, int value);`
 - `void insert_at_tail(Node** head, int value);`
 - `void delete_value(Node** head, int target);`
 - `void print_list(Node* head);`
 - `void free_list(Node** head);`

Travail demandé

Étape 1 : Déclaration de la structure

Déclarez une structure `Node` représentant un nœud avec les deux champs obligatoires. Ajoutez un commentaire expliquant clairement le rôle de chaque champ.

```
1 // Structure représentant un noeud de la liste chainee
2 typedef struct Node {
3     int data;           // Donnee stockee dans le noeud
4     struct Node* next;  // Pointeur vers le noeud suivant
5 } Node;
```

Étape 2 : Fonctions à implémenter

Implémentez les fonctions suivantes :

- a) **create_node** : Alloue dynamiquement un nœud et initialise ses champs.
- b) **insert_at_head** : Insère un nœud au début de la liste.
- c) **insert_at_tail** : Insère un nœud à la fin de la liste.
- d) **delete_value** : Supprime le premier nœud contenant une valeur donnée.
- e) **print_list** : Affiche les valeurs de tous les nœuds de la liste.
- f) **free_list** : Libère toute la mémoire allouée dynamiquement.

Étape 3 : Fonction `main()` et tests

Écrivez une fonction `main()` permettant de :

- Ajouter plusieurs éléments en tête et en fin de liste
- Afficher la liste à chaque étape
- Supprimer un élément (présent ou absent)
- Libérer la mémoire allouée

Étape 4 : Rapport d'analyse

Rédigez un court rapport au format PDF répondant aux questions suivantes :

1. **Complexité** : Quelles sont les complexités temporelles des opérations réalisées ?
2. **Comparaison** : Quels sont les avantages et inconvénients des listes chaînées par rapport aux tableaux ?
3. **Mémoire** : Pourquoi faut-il toujours libérer la mémoire des nœuds ? Que risque-t-on si on ne le fait pas ?
4. **Extension possible (facultatif)** : Proposez une fonctionnalité supplémentaire (inversion, insertion triée, etc.).