

Rappel structure pour coder une liste chaînée d'entiers :

```
class Maillon
{
    friend class Liste ;
    int Info;
    Maillon * suivant;
    ...
};

class Liste
{
    Maillon * tete;
    public :
        Liste();
        ~Liste();
    ...
};
```

- 1) Écrire le constructeur de liste vide.
- 2) Écrire un constructeur de Maillon vide, un constructeur de Maillon avec une valeur, un constructeur de Maillon avec une valeur et un pointeur suivant.
- 3) Écrire un constructeur créant une liste à partir d'un tableau d'entiers. Ce constructeur aura 2 arguments : le tableau d'entiers et le nombre de valeurs à prendre dedans.
- 4) Écrire la fonction membre `Afficher` sans argument qui affiche tous les éléments d'une liste.
- 5) Écrire le destructeur de la classe Liste.

Proposer 2 versions du destructeur de Liste suivant qu'il existe ou pas un destructeur pour Maillon.

- 6) Écrire la **fonction** `Afficher_Rec` qui affiche tous les éléments d'une liste, de manière récursive.
- 7) Écrire la fonction membre `Afficher_Inverse` qui affiche tous les éléments d'une liste en ordre inverse. (Récursive ?).
- 8) Écrire la fonction membre `Cardinal` sans argument, qui renvoie le nombre d'éléments de la liste.
- 9) Écrire la fonction membre `Chercher` à un argument entier, qui renvoie `true` si cet élément appartient à la liste, `false` sinon.
- 10) En considérant que les informations portées par la liste chaînées sont ordonnées en ordre croissant, écrire la fonction membre `Chercher2` à un argument entier, qui renvoie `true` si cet élément appartient à la liste, `false` sinon.
- 11) Écrire la **fonction** `Nb_Occurence` à un argument entier qui retourne le nombre d'occurrences (nombre d'exemplaires) de l'argument dans la liste.
- 12) Écrire la fonction membre `Supprimer` à un argument entier, qui supprime la première occurrence de cet élément de la liste, s'il existe.
- 13) Écrire la fonction membre `Supprimer_Tout` à un argument entier, qui supprime toutes les occurrences de cet élément de la liste, s'il existe.
- 14) Écrire le constructeur de copie.