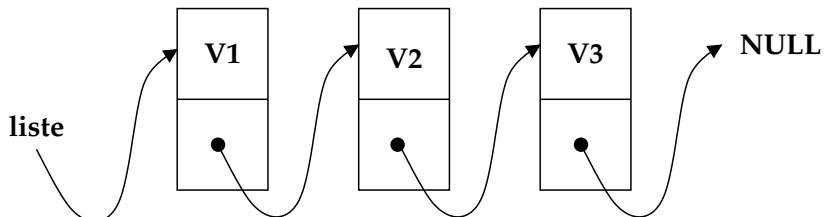


## Programmation en C avancée

CC2: À rendre pour le mercredi 13/01/2021 à 12h

### Liste chaînée et allocation dynamique

D'après Wikipédia, "une liste chaînée désigne en informatique une structure de données représentant une collection ordonnée et de taille arbitraire d'éléments de même type, dont la représentation en mémoire de l'ordinateur est une succession de cellules faites d'un contenu et d'un pointeur vers une autre cellule". Ceci est illustré sur la figure ci-dessous.



L'objectif est d'écrire un programme qui permet de manipuler une liste chaînée d'entiers. Ce programme est à réaliser seul, et à déposer sur Moodle au plus tard le **13/01/2021 à 12h**, sous forme d'un fichier `nom_prenom.c`. Pour chaque fonction, vous devrez :

- expliciter l'intérêt des paramètres,
- et expliquer l'algorithme utilisé.

► 1. Définir une structure qui permet de représenter une cellule de cette liste chaînée. Elle contiendra un entier et un pointeur vers une autre cellule.

Pour déclarer une liste chaînée, il suffit ensuite de déclarer un pointeur qui pointe vers le premier élément de cette liste (ou `NULL` si la liste est vide).

► 2. Écrire trois fonctions d'insertion d'un entier dans une liste :

- en tête, c'est-à-dire, en premier élément,
- en queue, c'est-à-dire, en dernier élément,
- et avant un élément identifié par un pointeur sur une cellule.

**Remarque 1** *L'insertion d'un entier consiste notamment à allouer dynamiquement une cellule pour stocker cet élément.*

► 3. Écrire une fonction qui permet d'afficher le contenu d'une liste.

► 4. Écrire une fonction qui détermine et retourne le nombre d'éléments dans la liste.

► 5. Écrire une fonction qui permet de supprimer un élément identifié par un pointeur sur une cellule.

**Remarque 2** *La suppression d'un élément consiste notamment à désallouer la mémoire utilisée par la cellule.*

► 6. Écrire une fonction qui permet de rechercher un élément dans une liste. Elle renverra un pointeur sur la cellule, si cet élément existe, et un pointeur `NULL`, sinon.

► 7. Écrire une fonction principale qui permet de tester l'ensemble des fonctions ci-dessus.

Maintenant, nous allons utiliser certaines de ces fonctions pour écrire une fonction `trier_liste`, qui permet de trier une liste d'entiers, en utilisant le tri par sélection-permutation vu en cours d'Algorithmique 1, et que vous avez déjà dû implanter un jour.

► 8. Écrire une fonction qui permet de créer aléatoirement une liste de taille donnée.

► 9. Écrire une fonction qui permet de trier le contenu d'une liste par ordre croissant.

► 10. Illustrer l'utilisation de ces deux fonctions dans la fonction principale.