

Travaux dirigés d'algorithmique

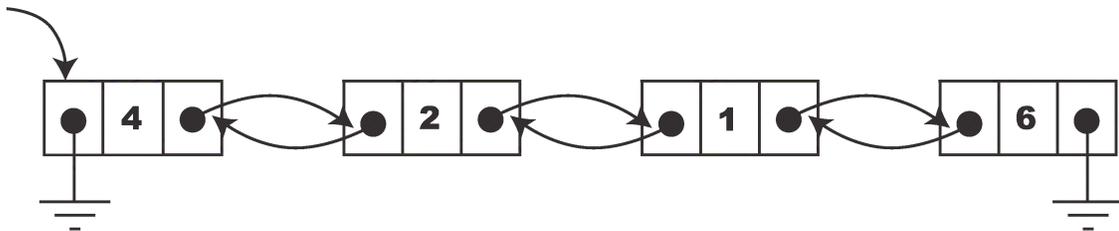
Cours d'Informatique

— IMAC Première Année —

Fonctions et pointeurs

Dans cette séance, nous nous intéressons à l'implémentation des listes chaînées.

Liste doublement chaînée : Un maillon d'une liste chaînée, appelé **cellule**, contient un des éléments de la liste ainsi qu'un lien vers la cellule suivante et un autre vers la cellule précédente. Se donner une liste chaînée, c'est alors se donner un lien vers la première cellule de la liste. Par exemple, la liste d'entier (4,2,1,6) est représentée par la liste chaînée d'entiers suivante :



Nous allons gérer un répertoire de contacts représenté par une liste doublement chaînée triée par ordre alphabétique. Ce répertoire sera rempli par l'utilisateur via la ligne de commande et sera sauvegardé dans un fichier.

► Exercice 1. Création de type

Pour créer ce **Repertoire**, nous avons d'abord besoin de définir un **Contact**.

Un **contact** est représenté par :

- un **nom** (chaîne de caractères),
- un **prenom** (chaîne de caractères),
- une **date** de naissance (un tableau de 3 entiers)
- et une adresse **email** (chaîne de caractères).

Un **contact** est aussi un maillon du **Repertoire**.

Un **Repertoire** n'est autre que le début de la liste.

Créez les types **Contact** et **Repertoire**.

► **Exercice 2. Gestion dynamique des contacts**

- Écrire une fonction `AlloueContact` qui alloue de la mémoire pour une nouvelle cellule et retourne l'adresse de la cellule créée.
- Écrire une fonction `CreeContact` prenant les informations d'un contact en paramètre et renvoyant un pointeur sur la nouvelle cellule créée. Pensez à mettre les champs suivant et précédent de cette cellule à `NULL`.
- Écrire une fonction `LibereContact` qui libère l'espace mémoire d'un **Contact**. Ainsi qu'une fonction `LibereRepertoire` qui libère l'espace mémoire d'un **Repertoire**.
- Dans votre programme principal, créer un répertoire de 3 contacts en créant successivement les cellules et en les reliant. Puis libérez le, vérifiez que **valgrind** ne vous sorte pas d'erreur.

► **Exercice 3. Affichage**

- Écrire une fonction `AfficheContact` qui affiche un contact sur la sortie standard :
Nom Prénom - [JJ/MM/AAAA] - email
- Écrire une fonction `AfficheRepertoire` qui affiche la liste des contacts sur la sortie standard (un contact par ligne). Pensez à utiliser la fonction précédente.
- Modifier vos fonctions pour qu'elles écrivent dans un fichier.

► **Exercice 4. Insertion d'un contact**

- Écrire une fonction qui insère un **Contact** dans un **Repertoire**
Attention, l'insertion doit respecter l'ordre alphabétique du nom puis du prénom. (utilisez les fonctions de "string.h" et le manuel).
/!\ Pensez à tous les cas possible, faites des dessins !!! /!\
- Réaliser une fonction demandant à l'utilisateur de rentrer les informations d'un **Contact** et qui construit le **Repertoire** en conséquence. La saisie s'arrête lorsque l'utilisateur rentre un "!".