

Sécurité et Technologies informatiques**3^{ème} Année Cycle Ingénieur****TD5 : Liste chaînée et pointeurs doubles****VERSION 1.0****A. ABDALLAH****TABLE DES MATIERES**

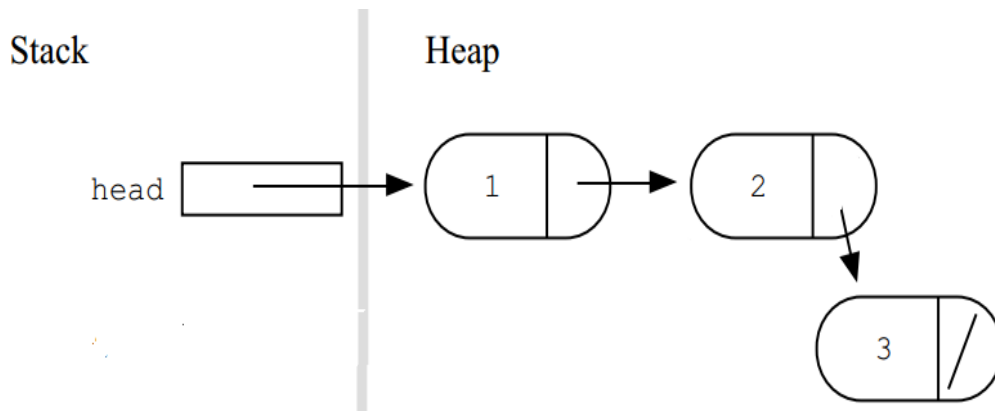
- Liste chaînée
- Pointeur double
- Compilation automatique avec l'outil *make*, un gestionnaire de **dépendances entre fichiers**.

2016/2017

1 Objectifs

Vous trouverez sur la plateforme **Célène** un code de gestion d'une liste d'entiers. Je vous laisse le temps nécessaire pour analyser le code fourni.

La fonction **build_Liste_123()** construit en mémoire la liste initiale suivante :



La structure est définie ainsi :

```
typedef liste {  
    int item ;  
    struct liste * next ;  
}
```

Pour implanter une fonction, commencer par dessiner un diagramme (voir figure) qui montre la mémoire pile (stack) et la mémoire dynamique (heap) avant et après l'appel à la fonction !

Exercice 1

Créer un fichier de compilation automatisée *Makefile* pour ce projet. Compilez-le et Testez.

Toutes les fonctions suivantes doivent être définies dans le module *liste.c*.

Exercice 2

Ecrire une fonction, qui prend un pointeur double comme paramètre d'appel, permettant de renverser la liste « passée en argument » ;

void inverser_Liste(Liste **headRef); // ne pas modifier le prototype

Tester la fonction.

Exercice 3

Ecrire une fonction qui permet d'insérer un nœud à la nième position. Cette fonction renvoie un void. Votre fonction doit appeler la fonction *insérer_Tete_Liste_2* déjà développée. N'oubliez pas de gérer le cas particulier où on vous demande d'insérer un nœud à la position 0 (tête de liste).

void inserer_Nieme(Liste **headRef, int position, int item); // ne pas modifier le prototype

Tester la fonction.

Exercice 4

Ecrire une fonction qui sérialise le contenu de la liste dans un fichier binaire. Utiliser la fonction prédéfinie *fwrite*. La fonction renvoie 0 si l'écriture s'est bien déroulée, -1 sinon. Donner le nom *liste.bin* à votre fichier.

*int write_Liste (Liste *liste, FILE *file) ;*

La fonction **write_Liste** doit écrire, en premier, dans le fichier le nombre de nœuds de la liste passée en argument (taille de la liste). Penser à écrire une fonction, *tailleListe*, séparée qui retourne la taille de la liste.

Tester la fonction.

Exercice 5

Ecrire une fonction qui lit le contenu d'un fichier binaire contenant une liste d'entiers. Utiliser la fonction prédéfinie *fread*. La fonction renvoie 0 si la lecture s'est bien déroulée, -1 sinon.

*int read_Liste (Liste *liste, File *file) ;*

La fonction **read_Liste** doit lire, en premier, dans le fichier, le nombre de nœuds de la liste (taille de la liste). Ensuite, itérer sur les nœuds de la liste.

Tester la fonction.

Exercice 6

Ecrire une fonction qui déplace un nœud d'une liste source à une liste destination. En effet, la fonction doit déplacer le nœud frontal de la liste source et l'ajouter, comme nœud frontal, à la liste de destination.

*void deplaceNoeud(Liste** destRef, Liste** sourceRef)*

Voici une fonction de test, dans main, qui teste *deplaceNoeud()* :

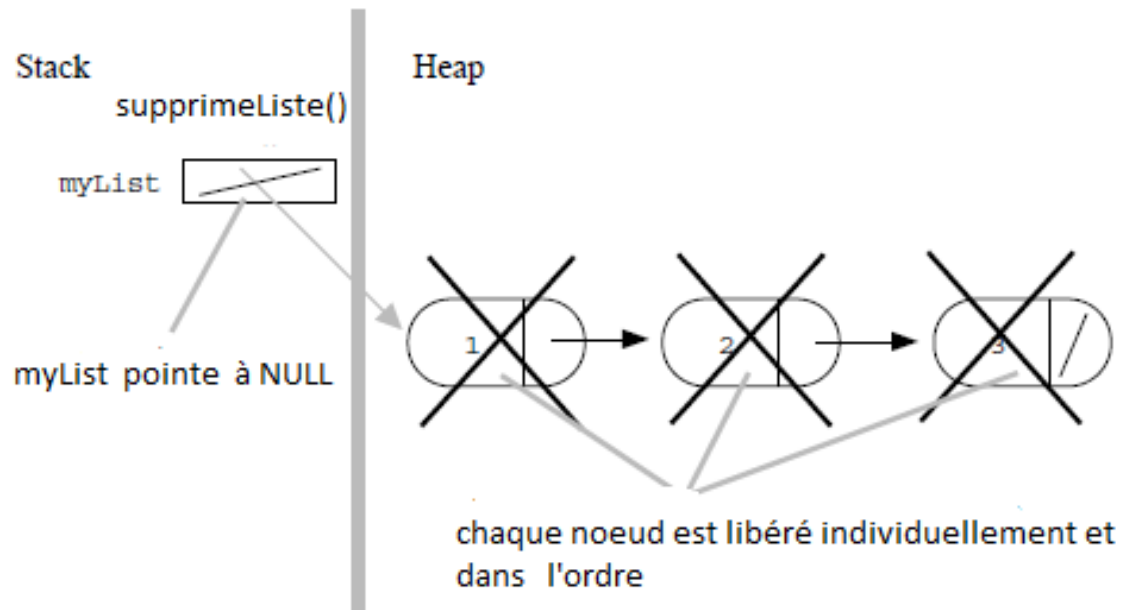
```
void deplacerNoeudTest() {  
  
    Liste* a = build_Liste_123() ;// la liste { 1, 2, 3}  
    Liste * b = build_Liste_123() ; // la liste { 1, 2, 3}  
  
    deplaceNoeud(&a, &b); // b est la liste source, a est la destination  
  
    // a == {1, 1, 2, 3}  
    // b == {2, 3}  
}
```

Récrire, la fonction **inverser_Liste()** en utilisant, cette fois, la fonction *deplaceNoeud()*.

Exercice 7

Ecrire une liste fonction `SupprimeListe ()` qui prend une liste, désalloue toute sa mémoire et initialise son pointeur de tête à NULL (la liste vide).

*`void SupprimeListe (Liste** headRef)`*



Exercice 8

Découvrir quelle action réalise la fonction mystérieuse *mystere*:

```
void mystere(struct Liste** headRef) {  
    struct Liste* premier;  
    struct Liste* reste;  
    if (*headRef == NULL) return;  
    first = *headRef;  
    reste = premier->next;  
    if (reste == NULL) return;  
    mystere(&reste);  
    premier->next->next = premier;  
    premier->next = NULL;  
    *headRef = reste;  
}
```

Aidez-vous d'un dessin en prenant la liste {1,2,3} par exemple.