Exercice 1 :

Soient L1 et L2 deux listes. On suppose que les listes L1 et L2 contiennent respectivement n et m éléments. Donner en pseudo-code un algorithme pour trouver l'intersection des deux listes.

Quel est le temps d'exécution de votre algorithme ? Faire deux pseudo-code :

1- Deux Listes chaînées simples

Solution avec 2 boucle for imbriqué O(n*m):

- Temps d'exécution : O(n*m)
- Pseudocode :

On commence par mettre le head de la première liste dans une variable nœud (headListe1)
Boucle tant que le pointeur suivant de headListe1 n'est pas égale à null
    Mettre le head de la deuxième liste dans une variable nœud (headListe2)
    Boucle tant que le pointeur suivant de headListe2 n'est pas égale à null

        Si headListe1 == headListe2 et si premierNoeudTrouvé (par défaut False) est False
            premierNoeudTrouvé = True
            noeudActuel = headListe1
            Insérer noeudActuel dans listeIntersection
            noeudPrecedent = noeudActuel

        Sinon si headListe1 == headListe 2
            noeudActuel = headListe1
            Lier pointeur suivant de noeudPrecedent à nœudActuel
            Insérer noeudActuel dans listeIntersection
            noeudPrecedent = noeudActuel

        headListe2 = le suivant de headListe2
    headListe1 = le suivant de headListe1
lier le pointeur suivant de noeudActuel à null
Return listeIntersection (vide par défaut)

Exercice 1 (Suite) :

2 - Deux Listes doublement chaînées

- Temps d'exécution : O(n*m)
- Pseudocode :

On commence par mettre le head de la première liste dans une variable nœud (headListe1)
Boucle tant que le pointeur suivant de headListe1 n'est pas égale à null
    Mettre le head de la deuxième liste dans une variable nœud (headListe2)
    Boucle tant que le pointeur suivant de headListe2 n'est pas égale à null

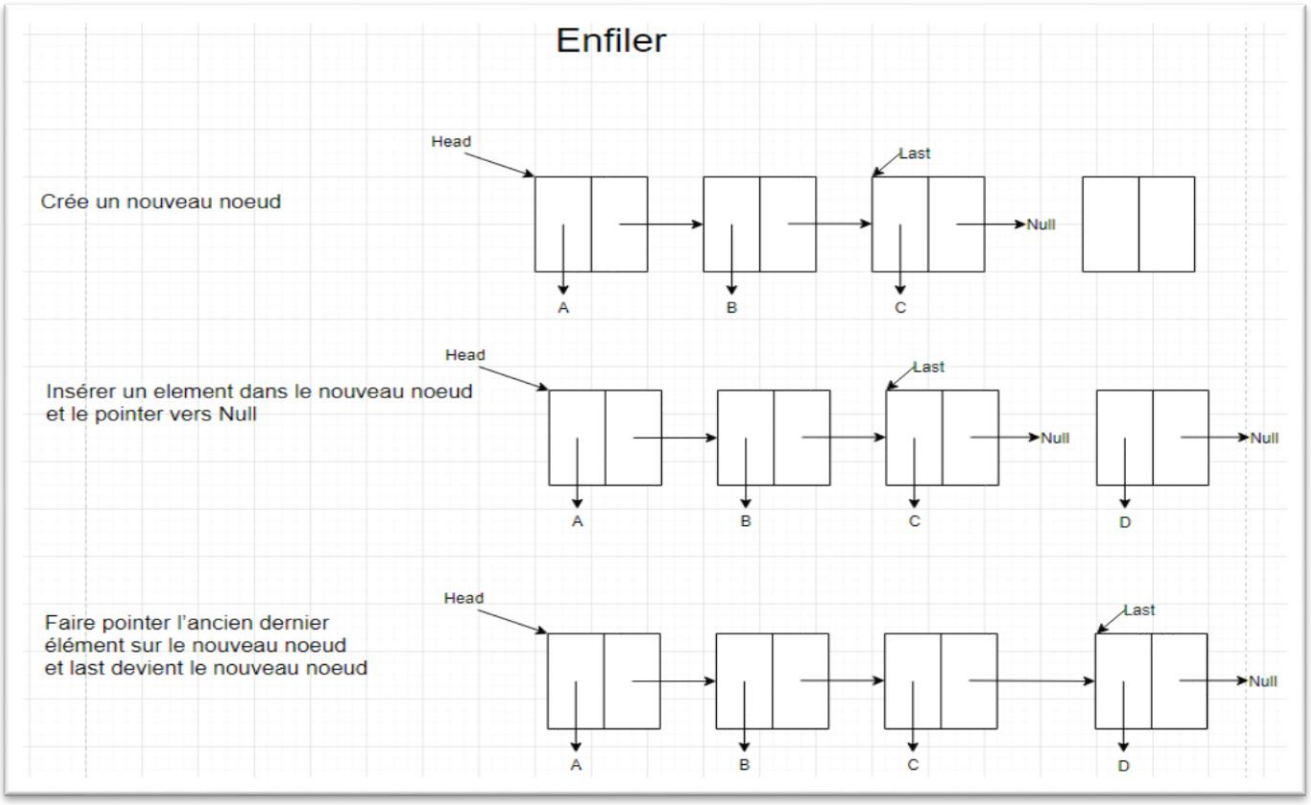
        Si headListe1 == headListe2 et si premierNoeudTrouvé (par défaut False) est False
            premierNoeudTrouvé = True
            noeudActuel = headListe1
            Lier pointeur précédent de noeudActuel à null
            Insérer noeudActuel dans listeIntersection
            noeudPrecedent = noeudActuel

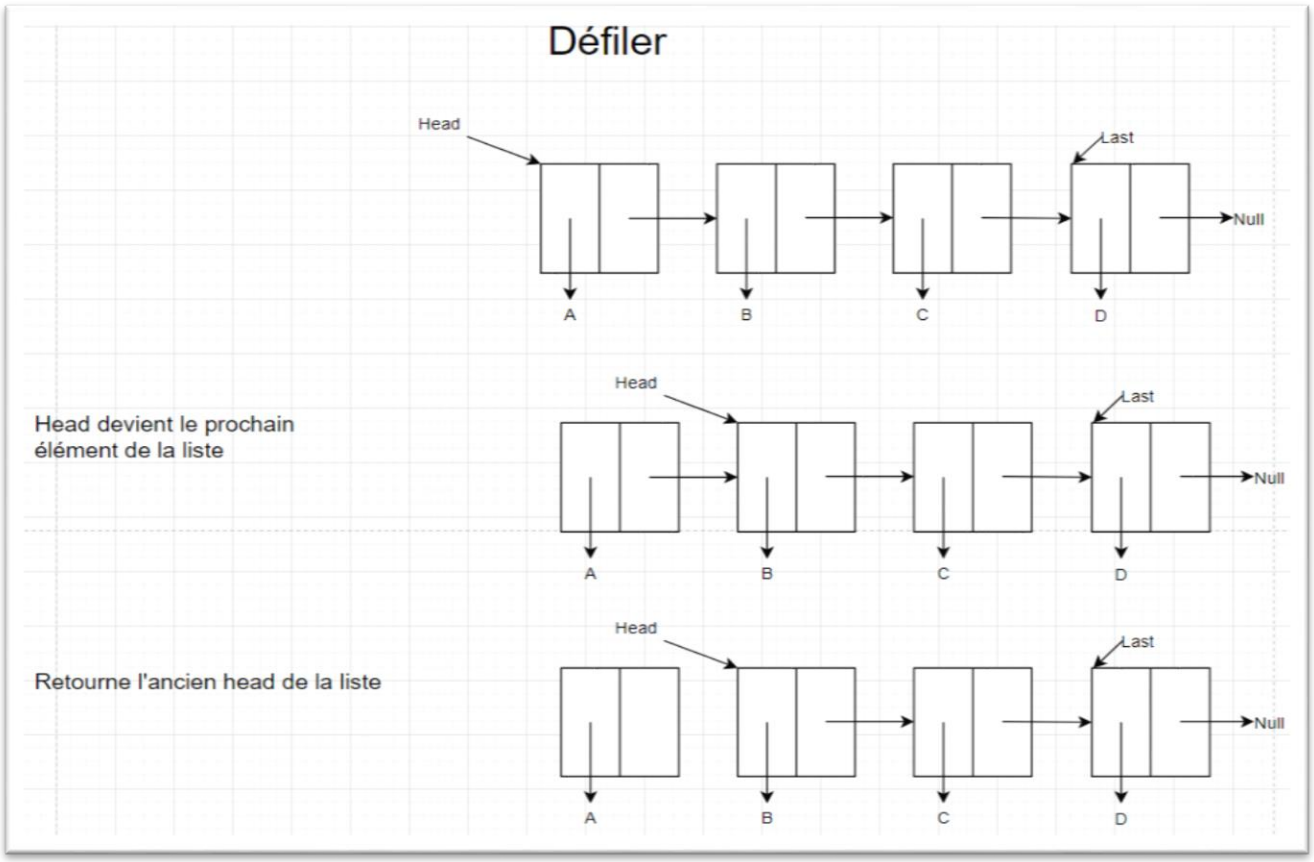
        Sinon si headListe1 == headListe 2
            noeudActuel = headListe1
            lier pointeur suivant de noeudPrecedent à noeudActuel
            lier pointeur precedent de noeudActuel à noeudPrecedent
            Insérer noeudActuel dans listeIntersection
            noeudPrecedent = noeudActuel

        headListe2 = le suivant de headListe2
    headListe1 = le suivant de headListe1
lier pointeur suivant de noeudActuel à null
Return listeIntersection (vide par défaut)

Exercice 2 :

a) Structure *pour implémenter le type abstrait de données file basée sur une liste simplement chaînée.*





Exercice 2 (suite) :

b) *Donnez les algorithmes en **pseudo code** des procédures enfiler et défiler.*

Contrainte: la structure implémentant la file ne doit être pointée que par <u>un seul</u> champ (exemple: tête, queue ou nombre d'éléments, etc.).

- Pseudocode défiler(file):

Vérifier si la file est vide
    retourne
Sinon verifier si le head pointe vers Null
    Retourne
Sinon
    stocker la valeur du head actuel dans une variable temp
    pointer head sur le nœud suivant
retourner temp

- Pseudocode enfiler(file, element):

Cree une variable nouveauElement qui prend l'élément et un pointeur suivant vers Null
Vérifier si la file est vide
    Head devient nouveauElement
Sinon
    Cree variable elementActuel et on lui met le head
    Tant que le pointeur suivant de elementActuel n'est pas egal a Null
        ElementActuel devient elementActuel du suivant
    Quand l'elementActuel est egal a Null alors on associe le suivant de l'elementActuel a nouveauElement

Exercice 3 :

Écrire un programme en pseudo code qui lit, des mots ou des phrases et qui décide si c'est un palindrome ou non

Fonction qui décide si mot est un palindrome ("mot")

Concatenage du string "mot" dans une variable motOrdre en enlevant les espaces

De l'index i : n-1 à 0 du string "mot", concaténer le char d'index i
dans le string motInverse qui est un string null au tout départ en enlevant les espaces

Si motOrdre est égale à motInverse
    retourne true

Sinon
    retourne false