

# Rapport TP 1 SDD

## Sommaire

Descriptions des fichiers .....	2
- valCell.c .....	2
- valCell.h .....	2
- linkedList.c .....	2
- linkedList.h .....	2
- polynomial.c .....	2
- Polynomial.h .....	2
- linkedList_main.c .....	2
- polynomial_main.c .....	2
- Creation_Polynome_1.txt .....	2
- Ecriture_Polynome_1.txt .....	2
- Insertion_Fichier_Vers_Polynome.txt .....	2
- Polynome_Vers_Fichier.txt .....	2
- poly1.txt .....	2
- Test_Derive_Vers_Fichier.txt .....	2
- Test_Addition_Vers_Fichier.txt .....	2
- Test_Produit_Vers_Fichier.txt .....	2
- README.md .....	2
Explication des algorithmes avancés .....	3
1. poly_derive : .....	3
2. poly_add .....	3
3. poly_prod .....	3
Structure de données des fonctions .....	4
create_cell .....	4
add_cell .....	4
search_prev .....	5
poly_deriv .....	5
poly_add .....	6
poly_prod .....	6

## Descriptions des fichiers

- [valCell.c](#) : Contient les fonctions de comparaisons de monômes et la fonction d’affichage de monômes
- [valCell.h](#) : Contient la structure monôme et la déclarations des fonctions de valCell.c
- [linkedList.c](#) : Contient toutes les fonctions de manipulation de listes chaînées
- [linkedList.h](#) : Contient la structure cell\_t et la déclarations des fonctions de linkedList.c
- [polynomial.c](#) : Contient les fonctions de dérivée d’un polynôme, d’addition de deux polynômes et de multiplication de deux polynômes.
- [Polynomial.h](#) : Contient les déclarations des fonctions de polynomial.c
- [linkedList\\_main.c](#) : Contient tous les tests pour les fonctions de linkedList.c
- [polynomial\\_main.c](#) : Contient tous les tests pour les fonctions de polynomial.c
- [Creation\\_Polynome\\_1.txt](#) : Fichier pour tester la fonction LL\_create\_list\_fromFileName qui crée un polynôme à partir d’un fichier, c’est-à-dire on crée un polynôme grâce à ce fichier.
- [Ecriture\\_Polynome\\_1.txt](#) : Fichier pour tester la fonction LL\_save\_list\_toFile qui écrit un polynôme dans un fichier.
- [Insertion\\_Fichier\\_Vers\\_Polynome.txt](#) : Fichier pour tester LL\_add\_celln qui insère plusieurs cellules et ensuite insère la liste dans un fichier.
- [Polynome\\_Vers\\_Fichier.txt](#) : Fichier pour tester la fonction LL\_save\_list\_toFileName qui enregistre un polynôme dans un fichier.
- [poly1.txt](#) : Fichier contenant le premier polynôme qu’on utilisera afin de tester la fonction de dérivée
- [Test\\_Derive\\_Vers\\_Fichier.txt](#) : Fichier contenant la dérivée d’un polynôme afin de tester la fonction LL\_save\_list\_toFileName.
- [Test\\_Addition\\_Vers\\_Fichier.txt](#) : Fichier contenant la somme de deux polynômes afin de tester la fonction LL\_save\_list\_toFileName.
- [Test\\_Produit\\_Vers\\_Fichier.txt](#) : Fichier contenant le produit de deux polynômes afin de tester la fonction LL\_save\_list\_toFileName.
- [README.md](#) : Fichier qui explique comment compiler et exécuter

## Explication des algorithmes avancés

Ce TP consiste à implémenter des fonctions pour manipuler des polynômes en utilisant des pointeurs de pointeurs. Les fonctions comprennent la création et l'insertion de cellules, la lecture et l'affichage de polynômes, ainsi que la recherche et la suppression de cellules. Le TP couvre également des opérations mathématiques telles que la dérivation, l'addition et la multiplication de polynômes.

Les fonctions dans `LinkedList.c` ne nécessitent pas d'explications plus détaillées que celles fournies dans les commentaires des fonctions, car ce sont des fonctions présentées dans le cours.

1. `poly_derive` : Cette fonction `poly_derive` prend en entrée un pointeur `head` vers le premier élément d'une liste chaînée représentant un polynôme. Elle applique la règle de dérivation de base aux monômes du polynôme, c'est-à-dire qu'elle multiplie le coefficient de chaque monôme par son degré et soustrait 1 à son degré. Ensuite, la fonction vérifie si le degré du monôme dérivé est devenu négatif. Si c'est le cas, la cellule correspondante est supprimée de la liste chaînée. Si la cellule à supprimer est la première de la liste, le pointeur de tête de la liste est mis à jour pour pointer sur la cellule suivante. Si la cellule à supprimer est une cellule de la liste, mais pas la première, le pointeur de la cellule précédente est mis à jour pour pointer sur la cellule suivante. Si le degré du monôme dérivé est positif ou nul, la fonction avance simplement dans la liste chaînée en mettant à jour les pointeurs `current` et `prev`.
2. `poly_add` : La fonction `poly_add` réalise la somme de deux polynômes représentés par des listes chaînées. Pour cela, la fonction parcourt les deux listes chaînées qui représentent les polynômes. Elle somme les coefficients de chaque monôme de degré égal dans la liste 1 et ajoute à la liste 1 les monômes qui n'existent pas dans la liste 2. Si les degrés des termes actuels des deux listes sont égaux, les coefficients des deux termes sont ajoutés. Si le coefficient est nul, la cellule actuelle est supprimée. Si le coefficient est non nul, la fonction continue à parcourir la liste. Si le degré du terme actuel de la liste 1 est inférieur au degré du terme actuel de la liste 2, la fonction continue à parcourir la liste 1. Si le degré du terme actuel de la liste 1 est supérieur au degré du terme actuel de la liste 2, la cellule actuelle de la liste 2 est insérée avant la cellule actuelle de la liste 1. Si la liste 1 n'est pas vide, la liste 2 est ajoutée à la fin de la liste 1. En fin de parcours, la fonction supprime la liste 2 puisqu'elle est maintenant vide.
3. `poly_prod` : La fonction `poly_prod` prend deux listes chaînées `head1` et `head2` qui représentent deux polynômes et renvoie une nouvelle liste chaînée qui représente le produit de ces deux polynômes. La fonction commence par initialiser des variables pour parcourir les deux listes et pour stocker le résultat. Elle parcourt ensuite chaque terme de la première liste et pour chaque terme, elle parcourt chaque terme de la deuxième liste. Pour chaque paire de termes, elle calcule le produit des coefficients et la somme des degrés, crée une nouvelle cellule contenant ces informations, et l'ajoute à la liste de résultats. Une fois que tous les produits ont été calculés et ajoutés à la liste de résultats, la fonction fusionne les termes de même degré dans la liste de résultats en additionnant les coefficients correspondants. Enfin, la fonction renvoie la liste de résultats.

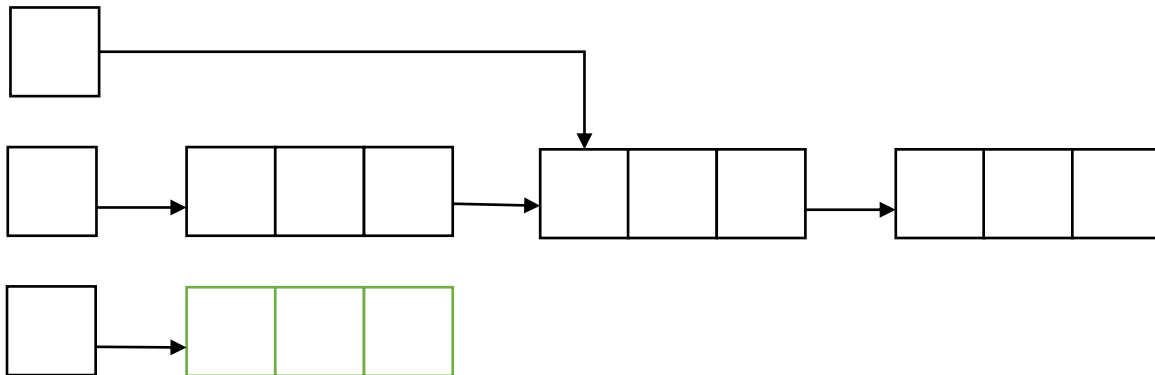
## Structure de données des fonctions

create\_cell

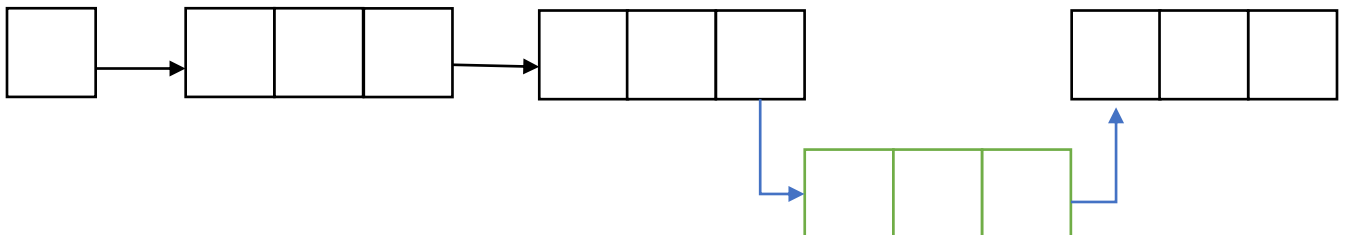
Coefficient	Puissance	Adresse suivant
-------------	-----------	-----------------

add\_cell

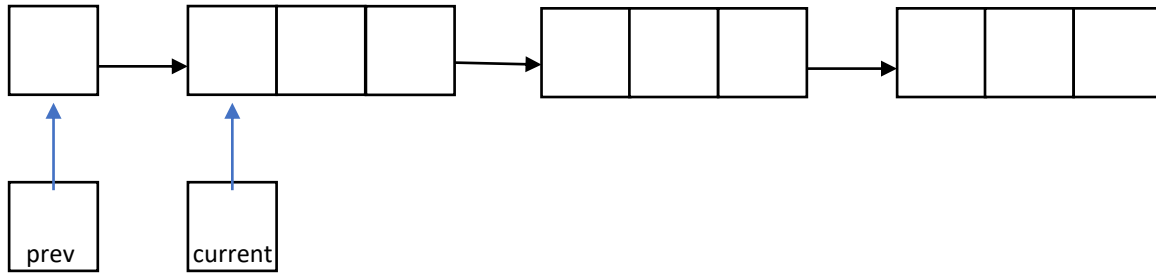
Entrée :



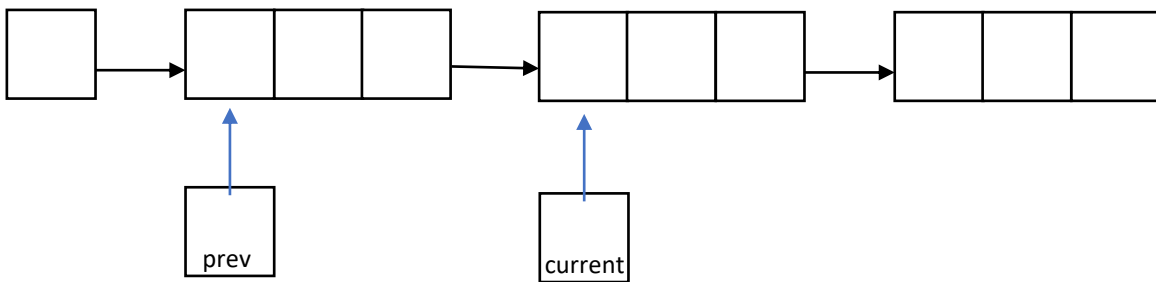
Sortie :



search\_prev



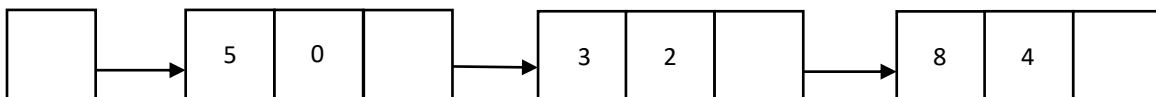
Parcours de la liste chaînée



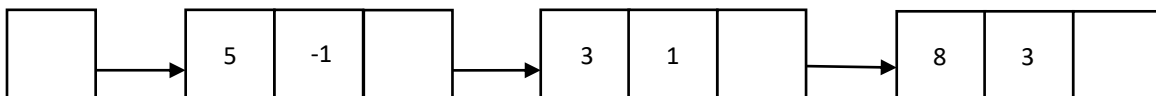
Sortie : Retourne prev

poly\_deriv

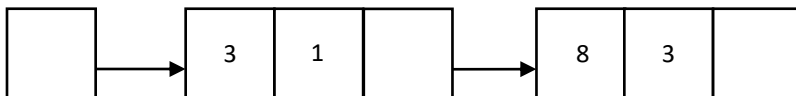
Entrée :



Dérivation :

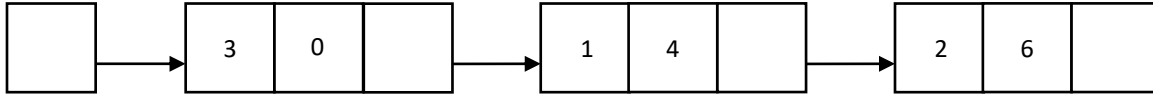
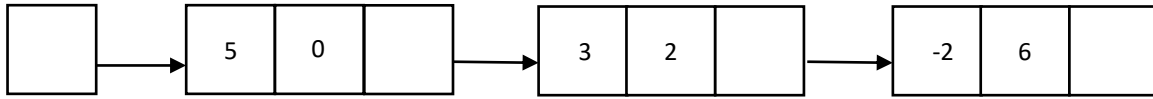


Sortie :

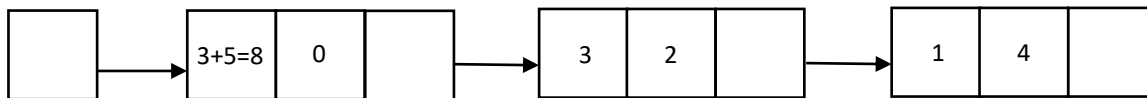


## poly\_add

Entrée :



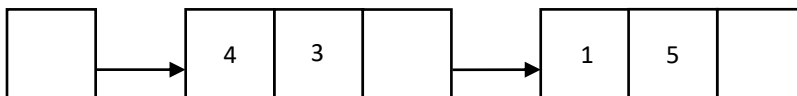
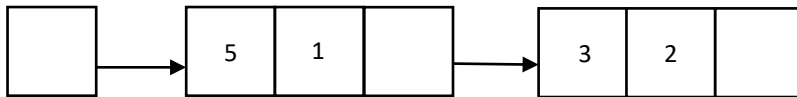
Sortie :



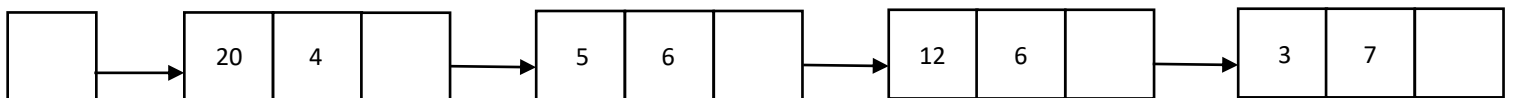
NULL

## poly\_prod

Entrée :



Produit :



Sortie :

