

Fiche C12

Les pointeurs

Eddy Caron
École Normale Supérieure de Lyon

1 Description de la fiche C12

Objectif : L'objectif de cette fiche est de manipuler les pointeurs. On en profite également pour voir les listes chaînées.

Difficulté : ***

Liens support de cours : Tuto sur les listes chaînées <http://sdz.tdct.org/sdz/les-listes-chainees-2.html>

2 Dans l'arène

Exercice C12.1

Recopier et remplir le tableau Table 1 et complétez-le pour chaque instruction du programme ci-dessous :

```
main()
{
    int A = 1; int B = 2; int C = 3;
    int *P1, *P2;

    P1=&A;
    P2=&C;
    *P1=(*P2)++;
    P1=P2;
    P2=&B;
    *P1-=*P2;
    ++*P2;
    *P1*=*P2;
    A=++*P2**P1;
    P1=&A;
    *P2=*P1/=*P2;

    return 0;
}
```

	A	B	C	P1	P2
Init.	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C					
*P1=(*P2)++					
P1=P2					
P2=&B					
*P1-=*P2					
++*P2					
P1=*P2					
A=++*P2**P1					
P1=&A					
*P2=*P1/=*P2					

TABLE 1 – Tableau à compléter.

3 La tête à l'envers

Exercice C12.2

Écrire un programme qui inverse les éléments d'un tableau **tab1** du type `int`. Le programme utilisera des pointeurs `p1` et `p2` et une variable numérique **permut** pour la permutation des éléments.

4 Al Zeï meurt

Exercice C12.3

Écrire la fonction qui alloue la mémoire d'un vecteur de taille **dimension**, puis qui l'initialise à la valeur **val**,

```
int * alloue_vecteur(int dimension, int val)
```

Écrire la fonction

```
void libere_vecteur(int * vecteur)
```

qui libère le vecteur **vecteur**. Afficher ce vecteur pour tester vos fonctions.

5 Ah ! La famille...

Exercice C12.4

On a retrouvé à la mairie de Trifouilly-les-Oies un registre de déclarations de naissances dont le contenu est un peu sommaire¹ :

1. Fortement inspiré D. Lazure. UPJV.

Brandon	Jihair	Saoul-Hélène
Jessica	Jihair	Saoul-Hélène
Jessica	Jihair	Pamella
Jonathan	Huggy	Pamella
Brenda	Huggy	Saoul-Hélène
Jennifer	Jihair	Pamella
Kevin	Huggy	.
Jacques	.	.
Valery	Jonathan	Jennifer
Georges	Jonathan	Jennifer
François	Jonathan	Jessica
Charles	Kevin	Jessica

Après un gros effort de recherche, on a fini par comprendre l'histoire et Netflix en a fait un feuilleton qui déchaîne les passions du village tout entier. Vous consultez le résumé de l'histoire :

Malgré les guerres à Port-Réal, Jihair et Saoul-Hélène eurent trois enfants : Brandon, Jessica et Jennifer. Alors que la fabrique de bonnets ignifugés pour dragons est au bord de la faillite, Huggy et Pamella élèvent quant à eux Jonathan, Brenda et Kevin. La vie est dure dans les douves du palais des Sans-Visage mais l'élection du 1024e Lord Commandant de la Garde de Nuit se passe sans aucun problème. Quelques années plus tard, Jonathan et Jennifer sont sur la paille et décident de faire un petit frère à Valery, leur premier enfant. Ainsi naquit Georges. Mais le malheur s'abat sur la famille et Jennifer décède dans des circonstances terribles (sauvagement poignardée aux côtés de son époux.). Jonathan tente d'oublier et refait sa vie avec Jessica. Ils ont un enfant qu'ils appellent François, en souvenir du temps passé. Mais la vertu de Jessica n'est pas à la hauteur réglementaire et ce qui devait arriver arriva : elle s'en va avec Kevin, pour élever leur enfant né le soir du mariage de Joffrey et Margaery : Charles. Entre temps Jacques, Immaculé de profession, arrive à Astapor et décide alors de s'entraîner pour les championnats de bouclier boomerang.

Entre deux épisodes, vous décidez de développer un outil informatique qui vous permettra de suivre en temps réel les évolutions démographiques des épisodes restants. Pour cela, vous résoudrez les problèmes suivants :

1. Comprenez le graphe (Figure 1 et la signification des 6 pointeurs de chaque nœud).
2. A-t-on besoin de mémoriser pour chaque nœud le sexe de la personne correspondante ? Pourquoi ?
3. À quoi sert le chaînage général de tous les nœuds (celui en petits pointillés) ?
4. Quel est l'intérêt d'avoir les chaînages cycliques horizontaux sur le schéma ?
5. Écrivez les déclarations de types nécessaires au problème.
6. Écrire une fonction qui prend en paramètre un nom et qui renvoie un pointeur sur le nœud correspondant (ou `null` si le nom n'existe pas dans le graphe).
7. Écrivez une fonction qui crée un nouveau nœud, lui donne un nom passé en paramètre, insère le nœud dans la liste de tous les nœuds présents dans le graphe et renvoie un pointeur sur ce nœud. Ce nouveau nœud sera considéré (pour l'instant) comme n'ayant pas de parent, pas de frère ni de soeur (comme Jacques dans l'exemple). On pourra considérer qu'il n'existe pas deux personnes de même nom.

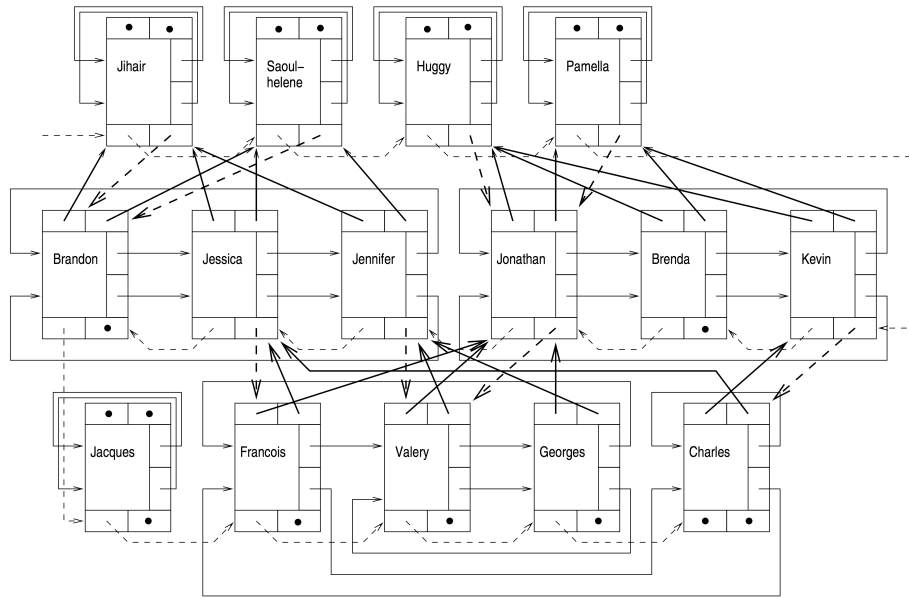


FIGURE 1 – Traduction de ce qui se passe dans votre tête après lecture du texte.

8. Écrivez une fonction qui modifie les différents pointeurs du graphe lors de l'établissement d'une filiation père/enfant. Cette fonction prends deux paramètres : un pointeur sur l'enfant (que l'on vient de créer avec la fonction précédente) et un pointeur sur le père ; et établit la filiation père/enfant en conséquence.
9. Même question avec la filiation mère/enfant.
10. Écrivez une fonction qui affiche l'état du graphe à un instant donné. Par exemple, vous pouvez écrire une fonction qui pour chaque nœud écrit sur `stdout` le nom de la personne concernée et les noms de chacun de ses éventuels 6 voisins.
11. Écrire une fonction qui lit un registre de déclarations de naissances et crée le graphe correspondant.
12. Testez votre programme sur l'exemple donné.
13. Terrible rebondissement durant l'épisode de cette semaine ! Un enfant n'était pas le fils de son père ! Écrire une fonction qui prend en paramètre un pointeur vers un nœud et qui annule la filiation d'avec son père.
14. Même question avec la filiation mère/enfant. (rebondissement certes un peu plus rare, mais à Trifouilly, rien d'impossible).
15. Écrire une fonction qui ordonne le chaînage général de tous les nœuds suivant l'ordre alphabétique des noms. Que faire de tous les autres pointeurs ?
16. Écrire une fonction qui affiche les ascendants successifs d'un nœud (son arbre généalogique).
17. Écrire une fonction qui affiche les descendants d'une personne, pour une génération dont le numéro sera passé en second paramètre.

18. Écrire une fonction qui affiche les descendants d'une personne, jusqu'à une génération donnée, génération par génération.
19. Écrire une fonction qui détecte les éventuelles anomalies généalogiques d'un graphe (comme avoir son père comme fils ou comme frère).
20. Écrire la fonction qui met fin au feuilleton en faisant place nette en mémoire.

6 Question subsidiaire (mais obligatoire)

Exercice C12.5

Selon vous dans le cadre de cette fiche quelle(s) notion(s) serait (seraient) intéressante(s) à aborder au travers d'un exercice. Si vous n'avez rien de particulier à dire indiquez simplement "RAS".

7 Conclusion

Voilà c'est déjà fini. Vous devez à présent rendre le résultat de cette fiche à votre référent² pour passer à la suivante. La prochaine est presque aussi facile.

2. Rappel : Pour toutes correspondances par mail avec votre référent, pensez à ajouter le tag [PROJ1].