# Exercice : La liste chaînée unsafe

## Un peu de théorie ...

### Notion de pointeur

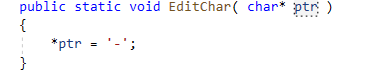
Un pointeur est variable, il est destiné à contenir une adresse mémoire, c'est-à-dire une valeur identifiant un emplacement en mémoire. Techniquement un pointeur est un entier positif sur 32 bits ou 64bits selon le type de système ciblé. Pour différencier un pointeur d'une variable ordinaire, on fait succéder le type du signe « **\*** » lors de sa déclaration.



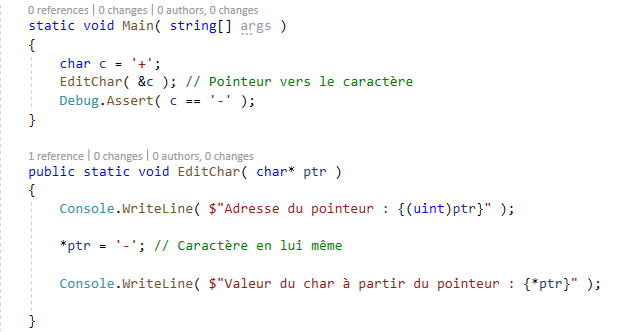
Ceci represente un pointeur vers un caractère.

### Manipuler les pointeurs

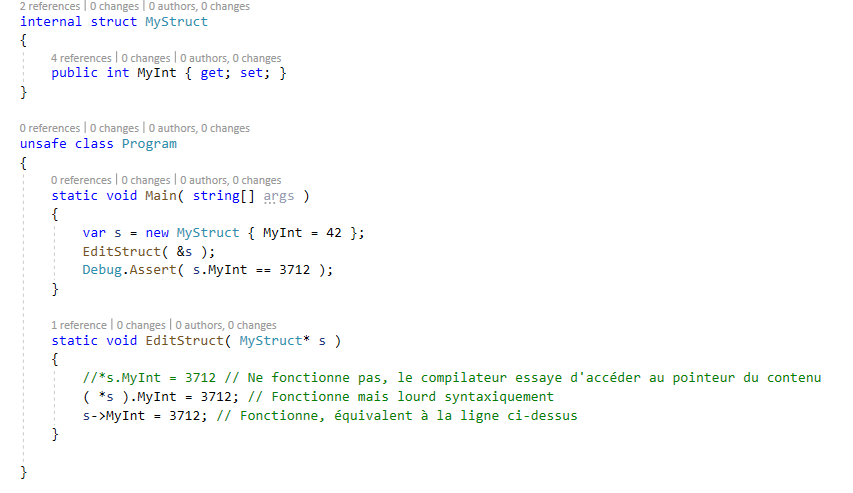
Pour accéder à la variable contenue à l’adresse d’un pointeur, nous utilisons aussi le signe « **\*** » :



Afin d’obtenir l’adresse mémoire d’une variable, nous utilisons cette fois-ci le signe « **&** »



Dans le but d’accéder à un champ ou une propriété d’une struct, nous utilisons le signe « **->** »



## Maintenant la pratique

L’objectif de cet exercice est d’implémenter une liste chaînée basique en utilisant les pointeurs. Les spécifications sont sous la forme de tests unitaires. Autrement dit : l'objectif de l'exercice est de faire passer en vert un jeu de tests.

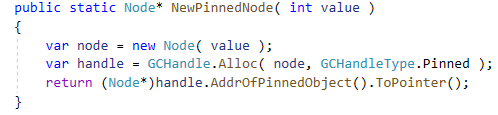
Les tests unitaires sont là pour spécifier de façon détaillée les fonctionnalités attendues. Cependant voici une courte description de ce qui est attendu :

* Créer un nœud
* Créer la liste
* Ajouter des éléments à la liste au début ou à la fin par valeur et par pointeur
* Supprimer des éléments à la liste par valeur et par pointeur
* Nettoyer la liste
* Implémenter un compteur
* Mettre en place un accesseur
* Savoir si la liste contient un élément
* Trouver la première ou la dernière occurrence d’un élément

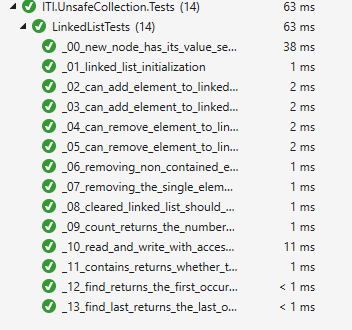
Vous disposez pour cela d'une solution comprenant 2 projets :

* ITI.UnsafeCollection.Tests, contient les tests unitaires
* ITI. UnsafeCollection, une implémentation minimale de la liste (le code écrit ne permet rien de plus que de pouvoir compiler sans erreur)

La fonction ci-dessous a été implémentée pour vous, vous n’avez nullement besoin de la modifier pour réussir cet exercice.



Il y a 14 tests à valider :



Bonne chance !