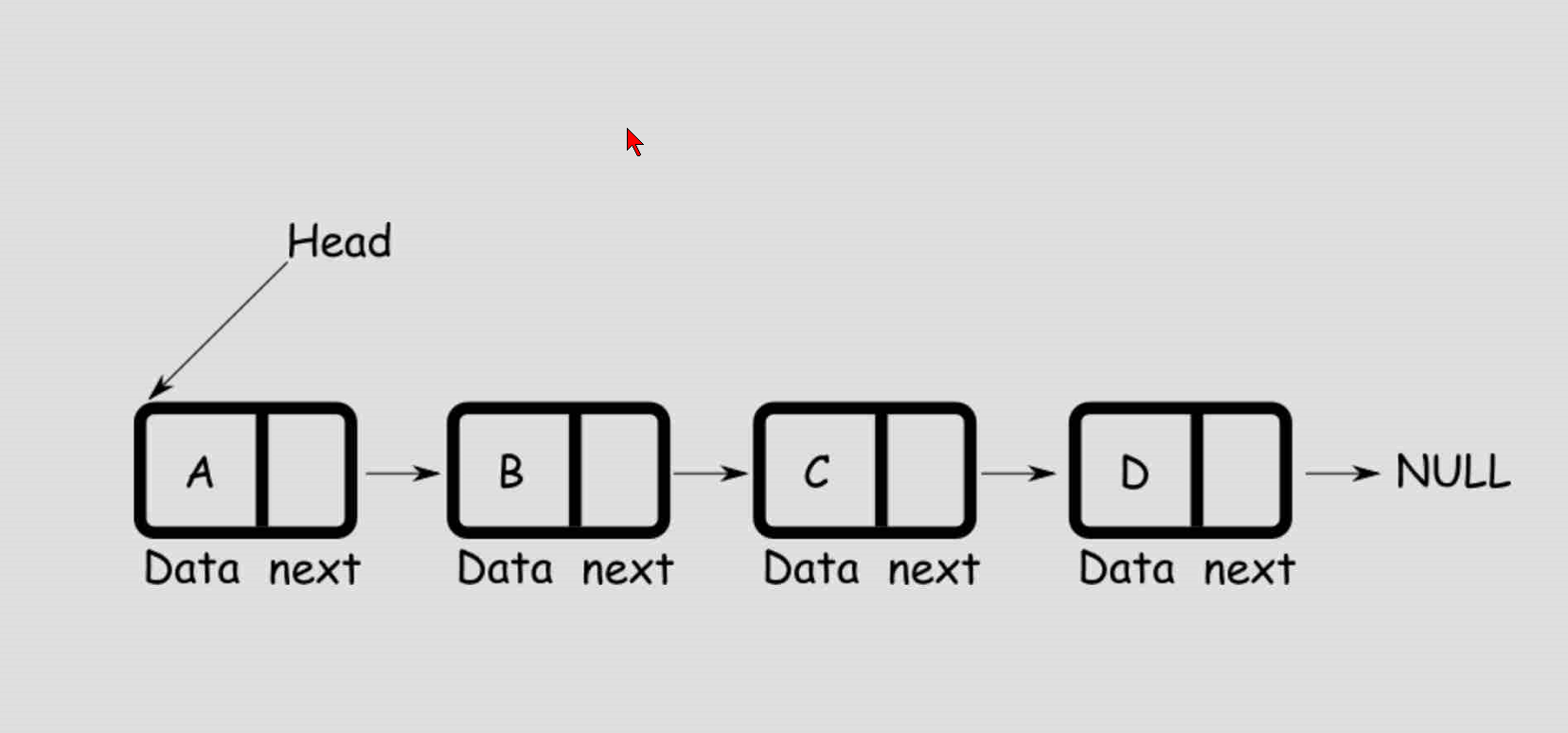
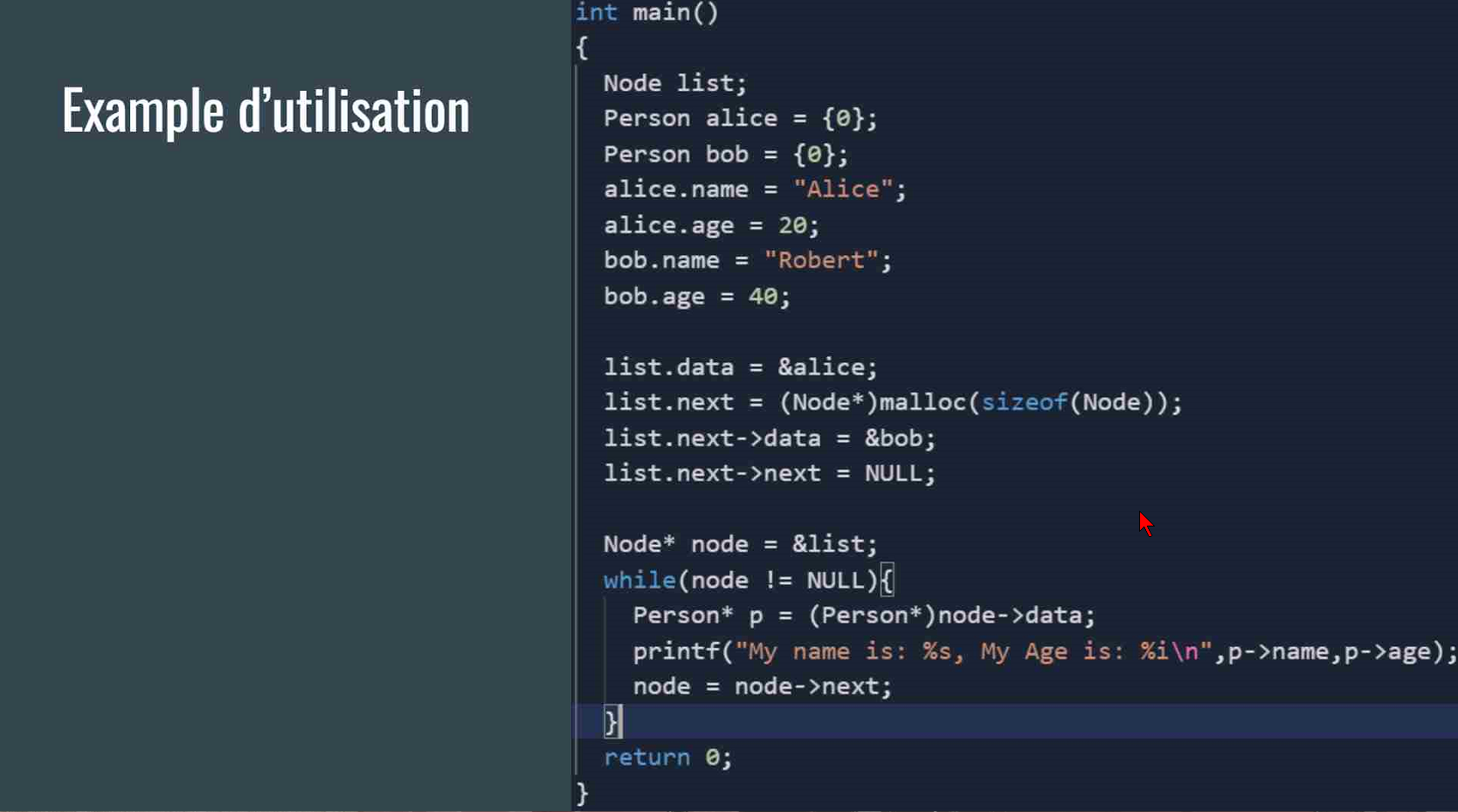
Semaine 4 cours tech prog

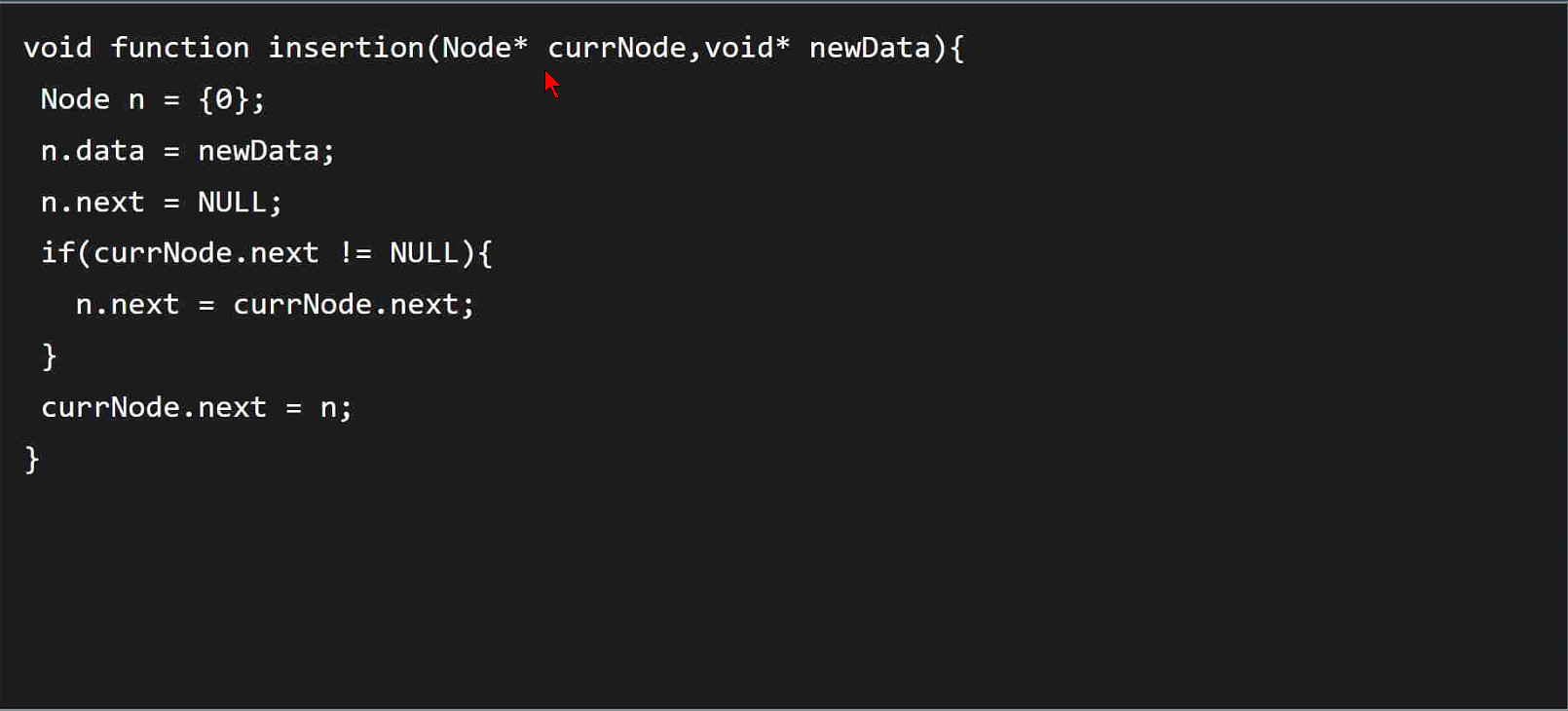
Différence entre liste et tableau

Liste simplement chinée : une chaine de nœud

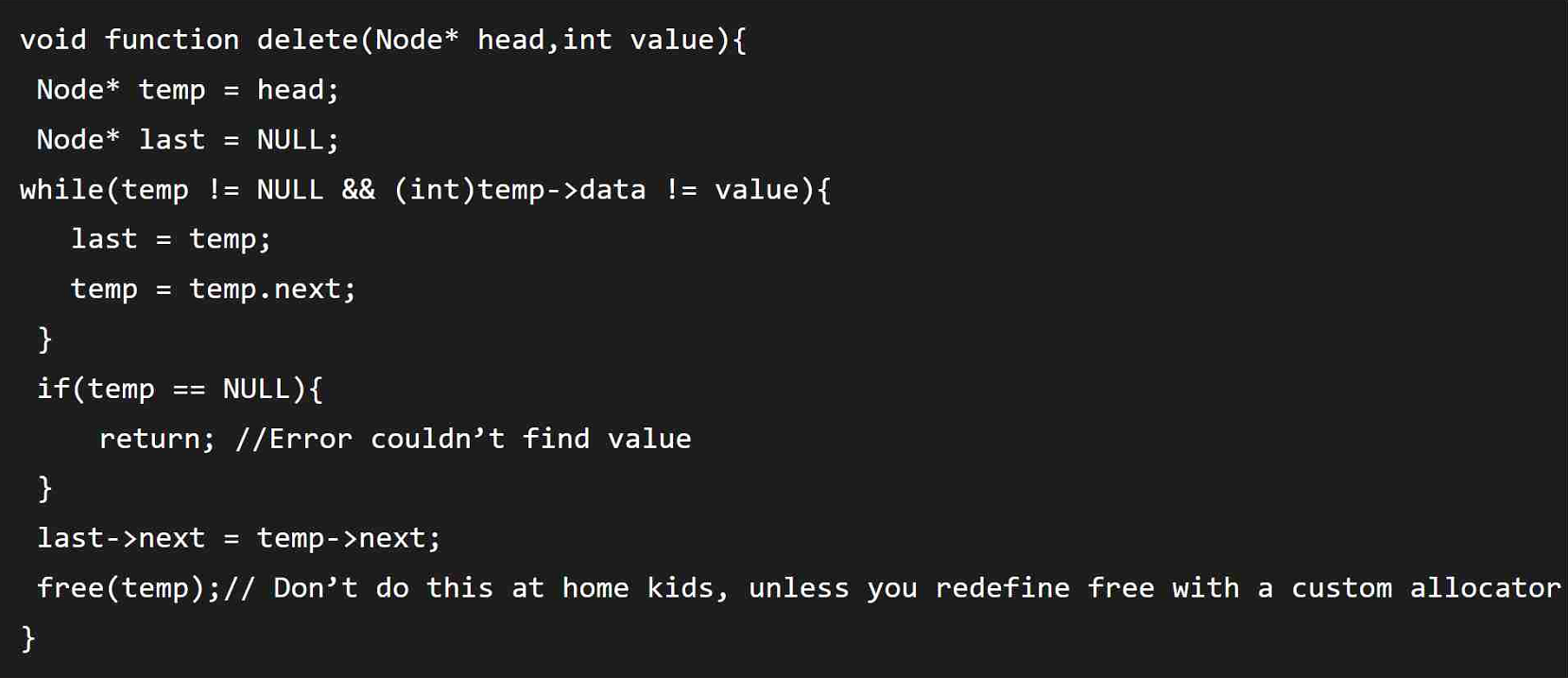




Lorsqu’on veut rajouter un nœud on doit juste dire que W est derrière B et avant C.



Pour supprimer C par exemple il faut juste dire que le next de B est le next de C



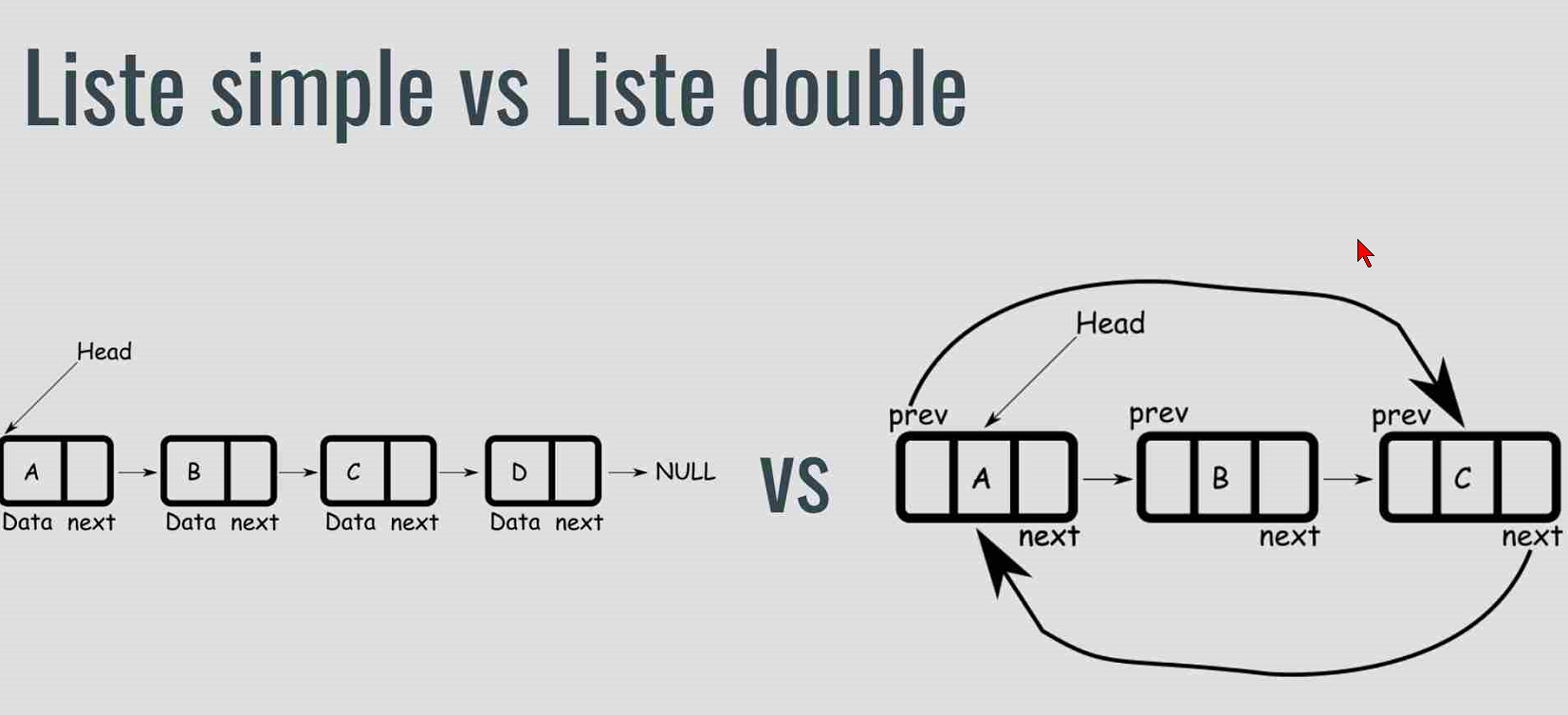
A chaque fois on va changer de nœud

Head = le debut du nœud

Value = le truc que tu cherche

Pour chercher notre nœud

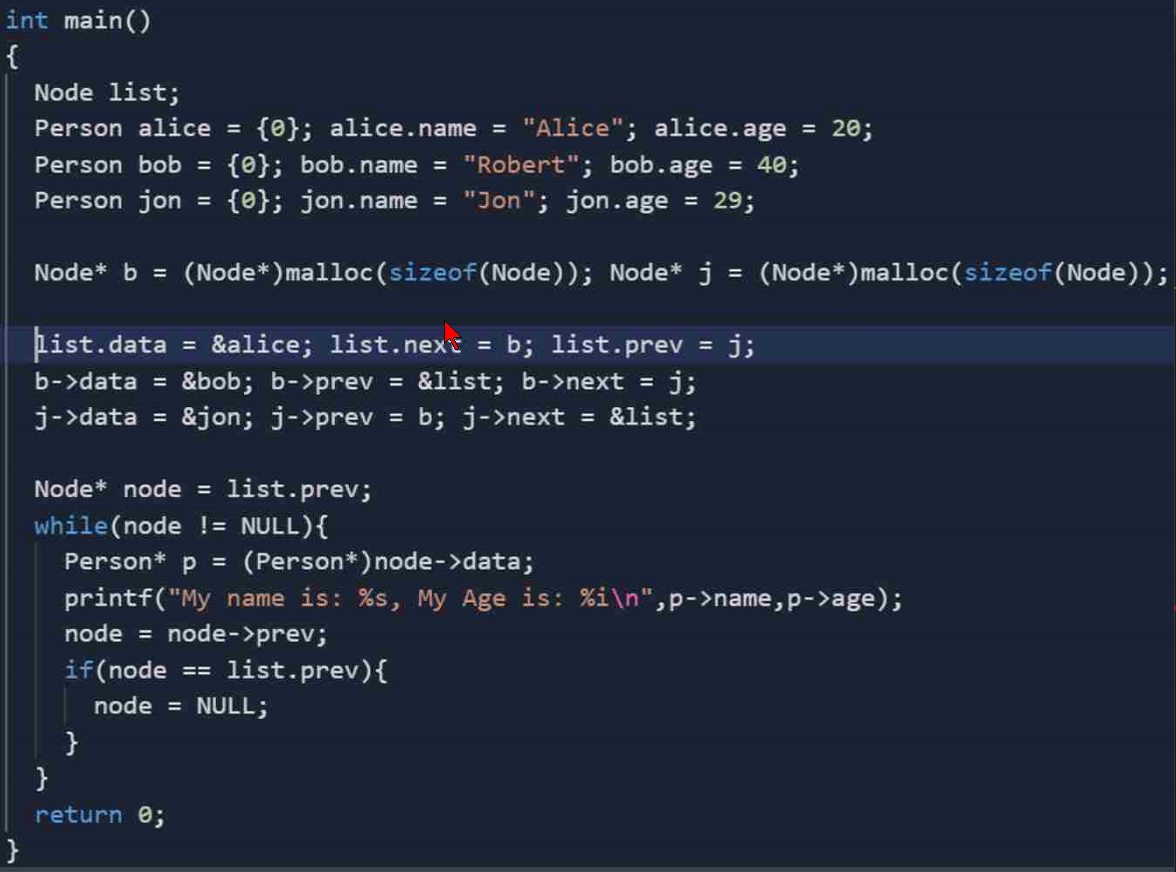
Liste doublement chainer

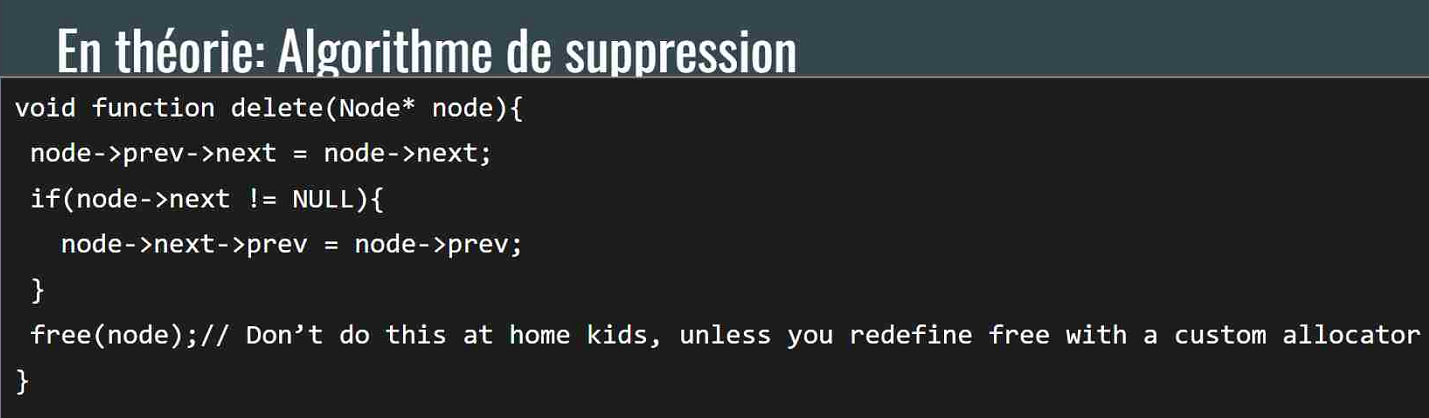
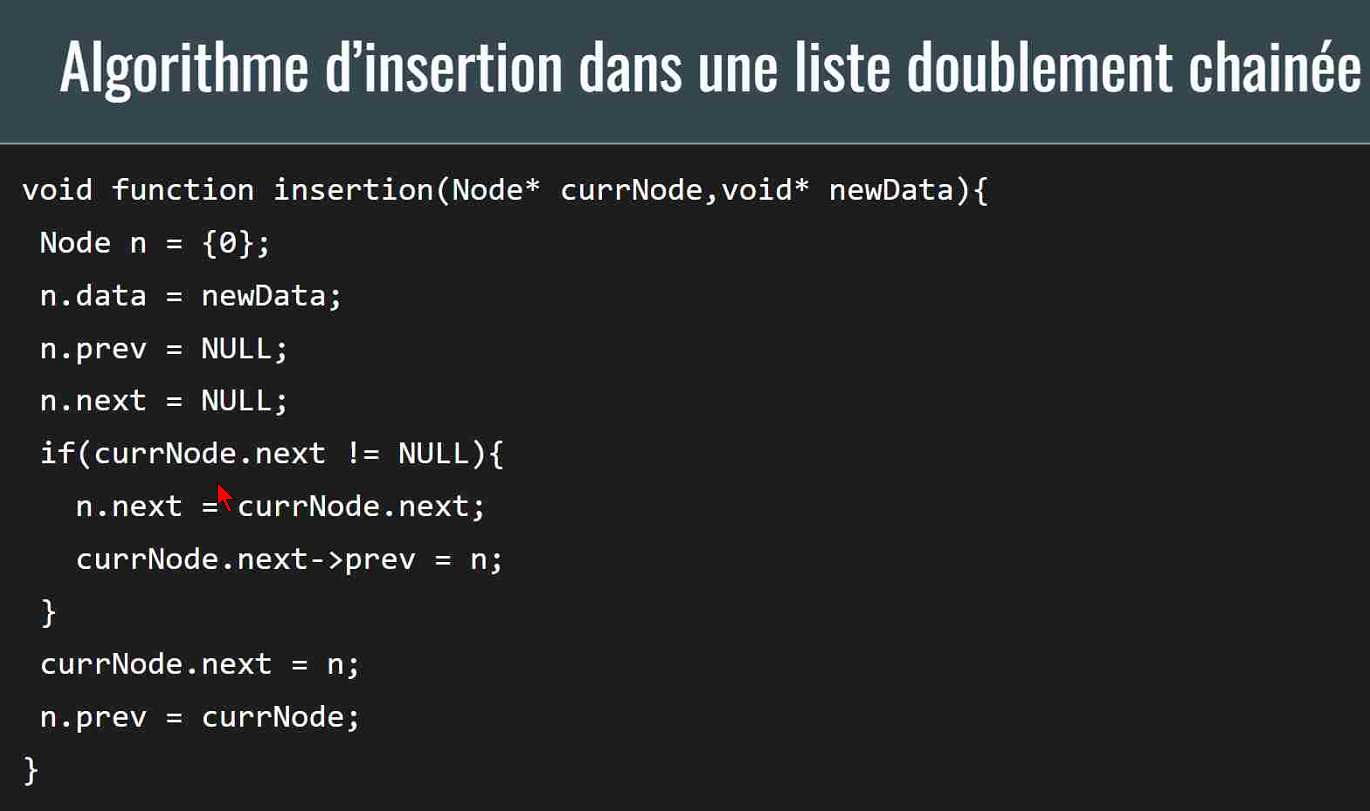


Dans une liste double cyclique il n’y a pas de NULL.

Dans une liste double non cyclique il y a 2 NULL head et last.

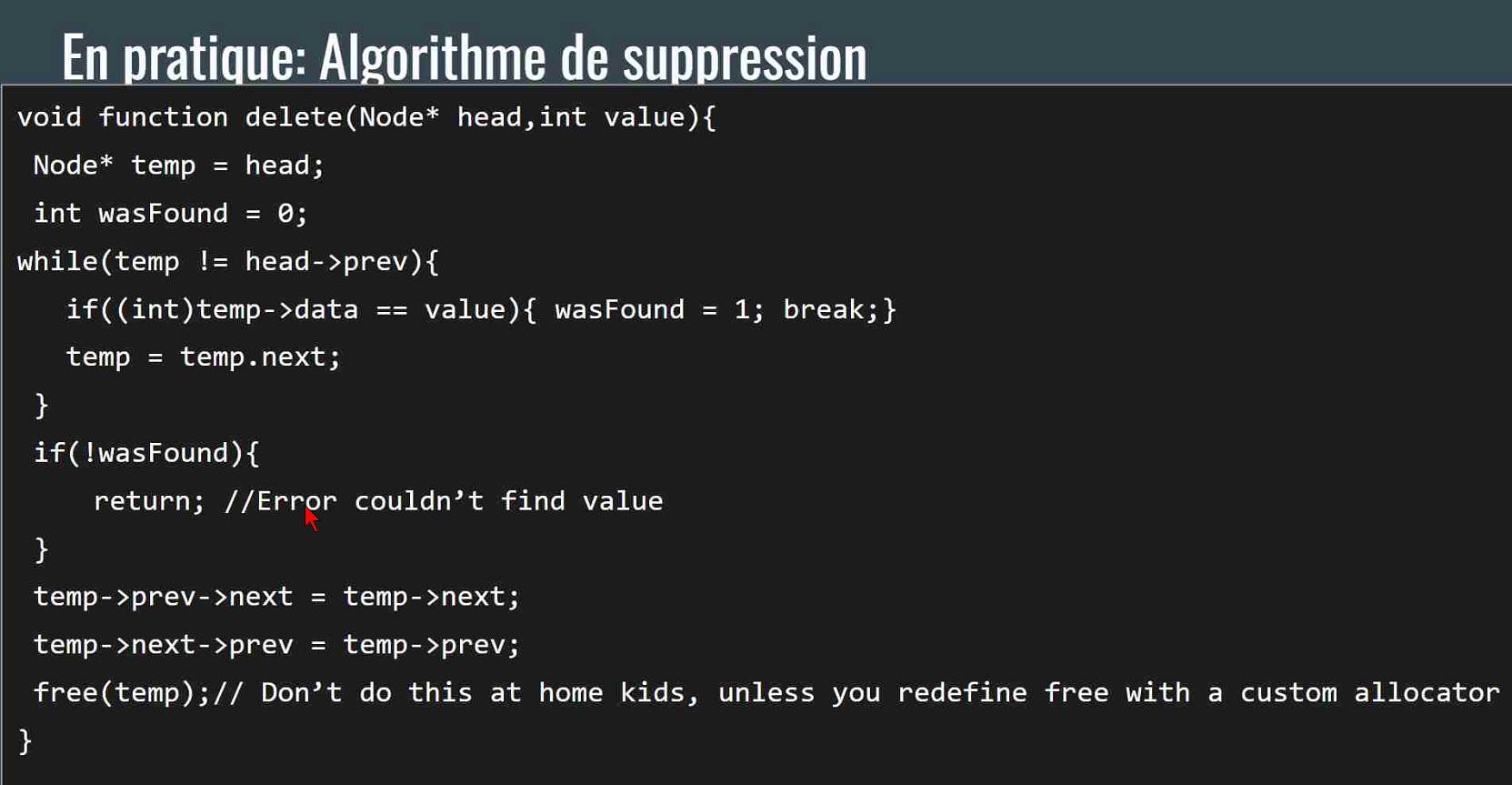
C’est plus efficace pour ajouter

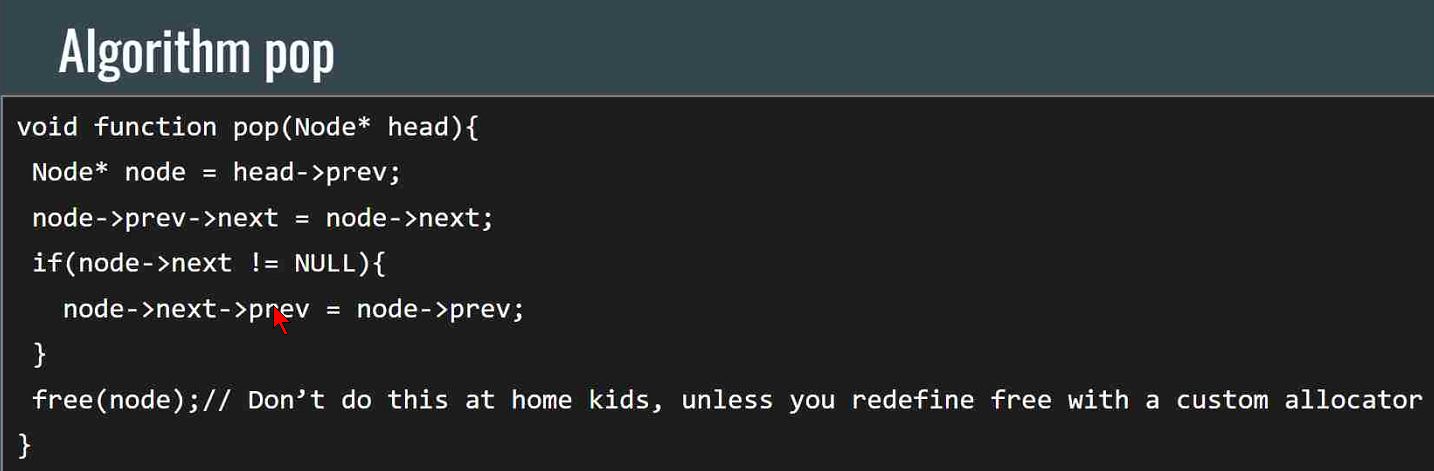


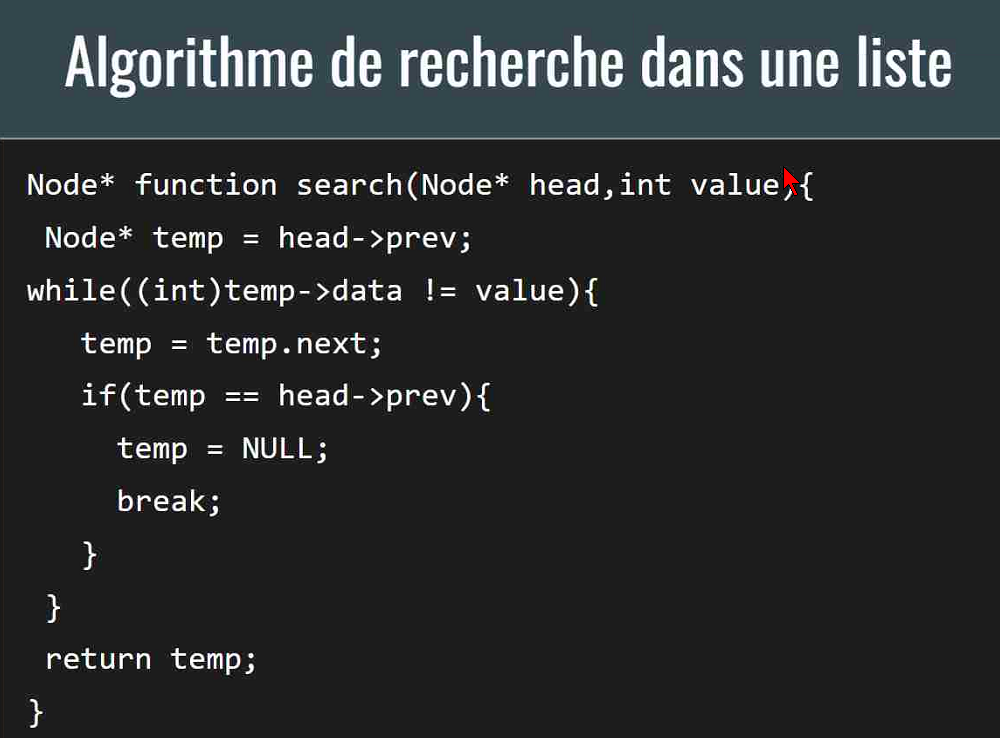


Node->prev->next : le next du nœud d’avand

Node->next : ton next







Pile ou « the stacks »

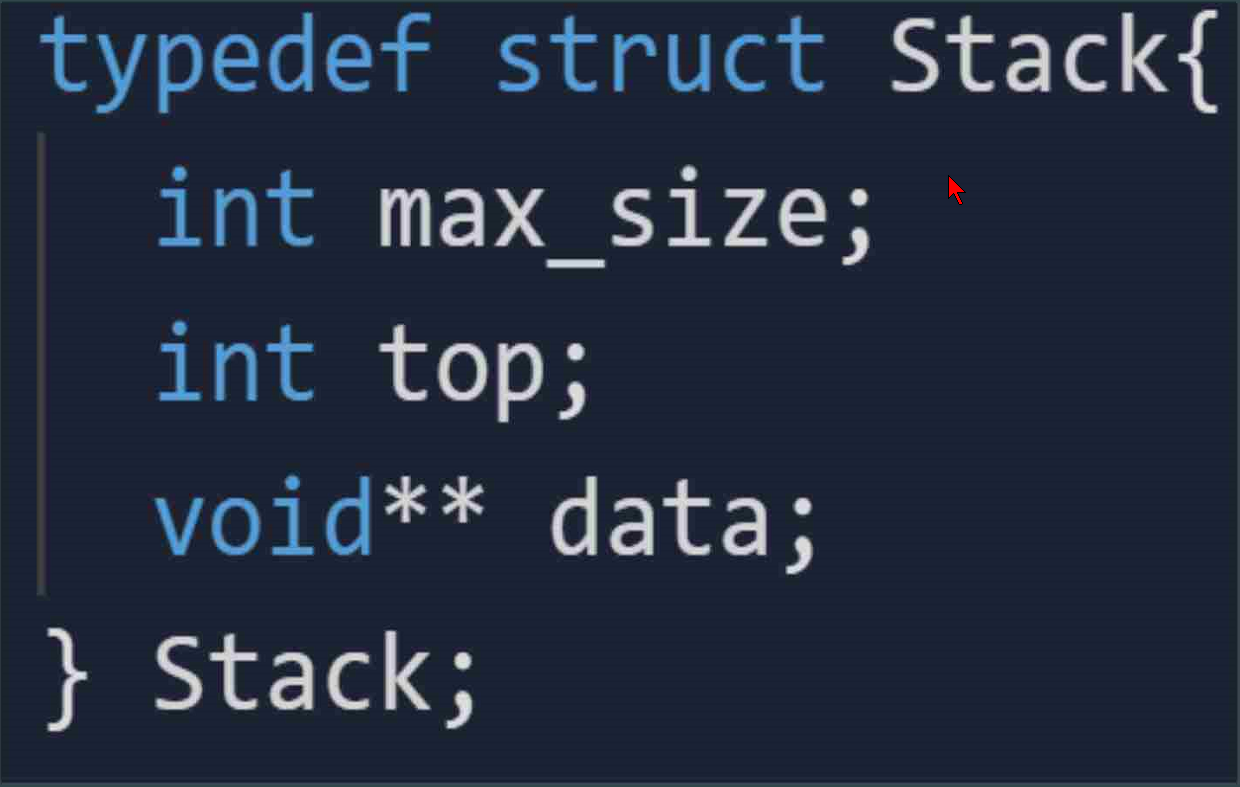
Les stacks sont ‘’last in, first out’’ ou ‘’LIFO’’

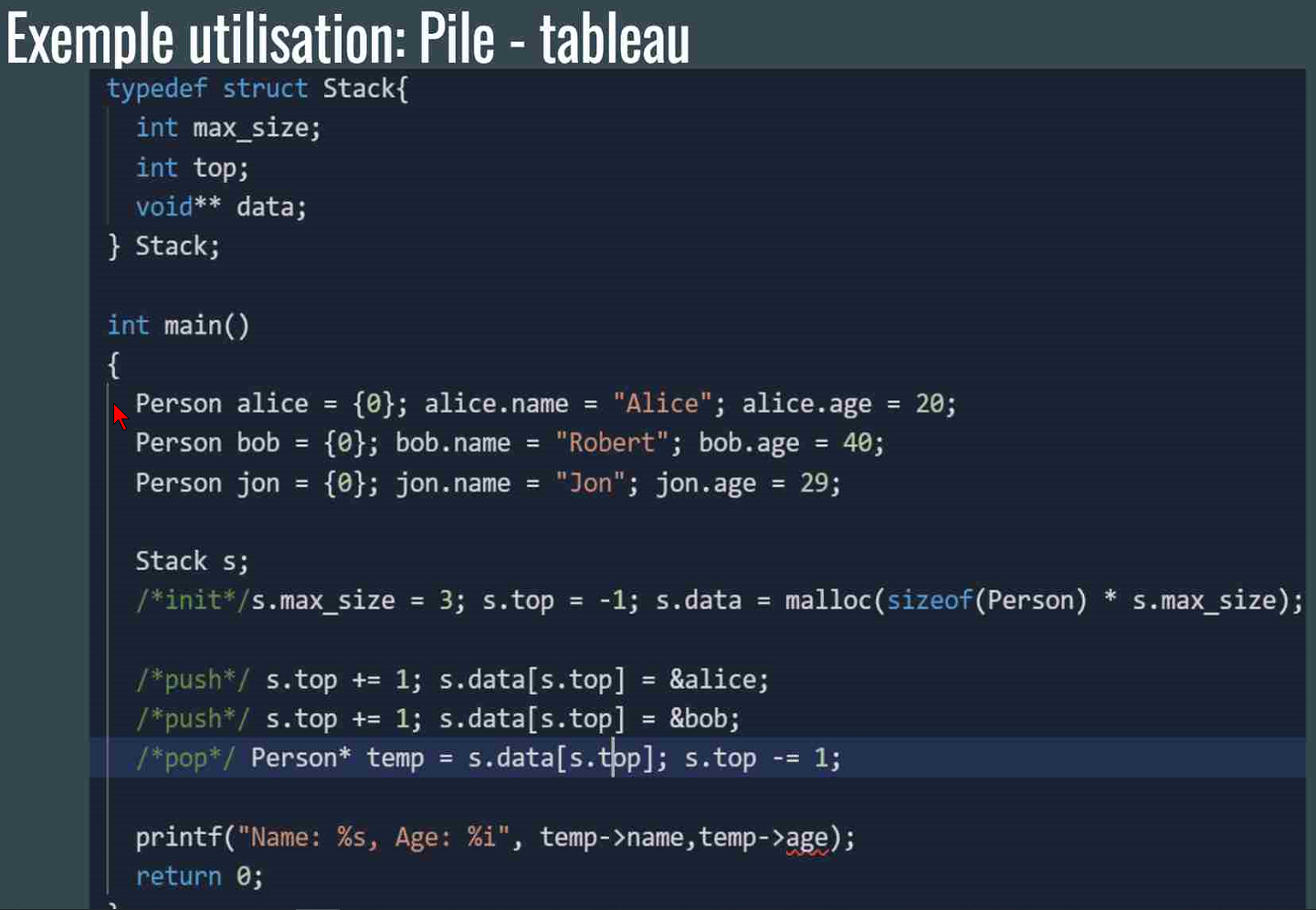
Si on veut aller chercher quelque chose dans un stack par exemple JS

* True
* Js
* Sus
* 42

On doit enlever les deux qui ont été enregistré après pour pouvoir y accédé.

Pour faire une stack on peut utiliser un tableau ou une liste doublement chainé, pas une simplement chainé parce qu’il ne peut pass accéder au donné d’avant.

 void\*\* = tableau de void\*



On doit initialiser la stack et lui donner un taille maximale

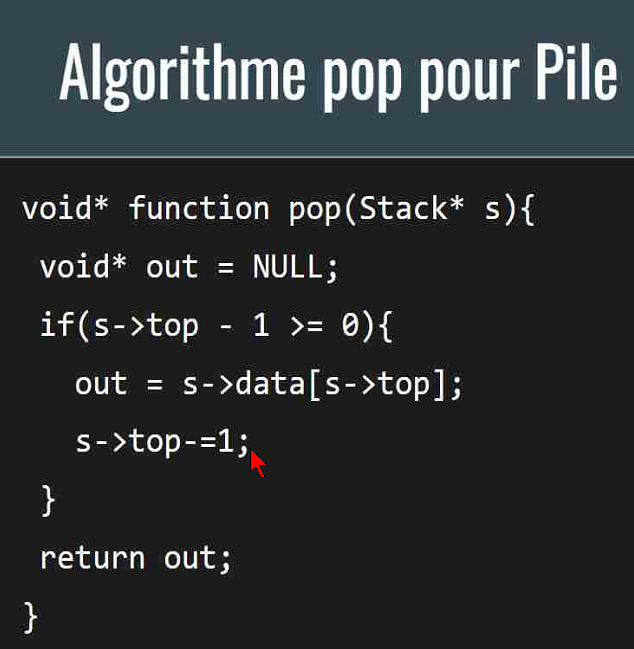
Si on veut savoir si la stack est vide on regarde si le top de la stack est = a -1

Le +1 est nécessaire parce que sinon ton stack va commencer a prendre les info a l’index -1 est ca faire beuguer

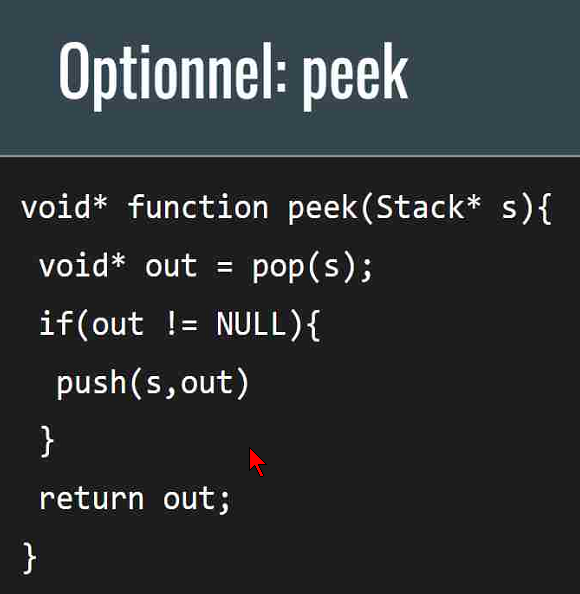
Pour push :



Pour pop :



Pour peek ou top :



Peek permet d’avoir accés a l’éléments sur le top^

Pour chercher un élément dans le stack :



On fait une stack temporaire pour pouvoir garder en mémoire ce que tu as fait pop.

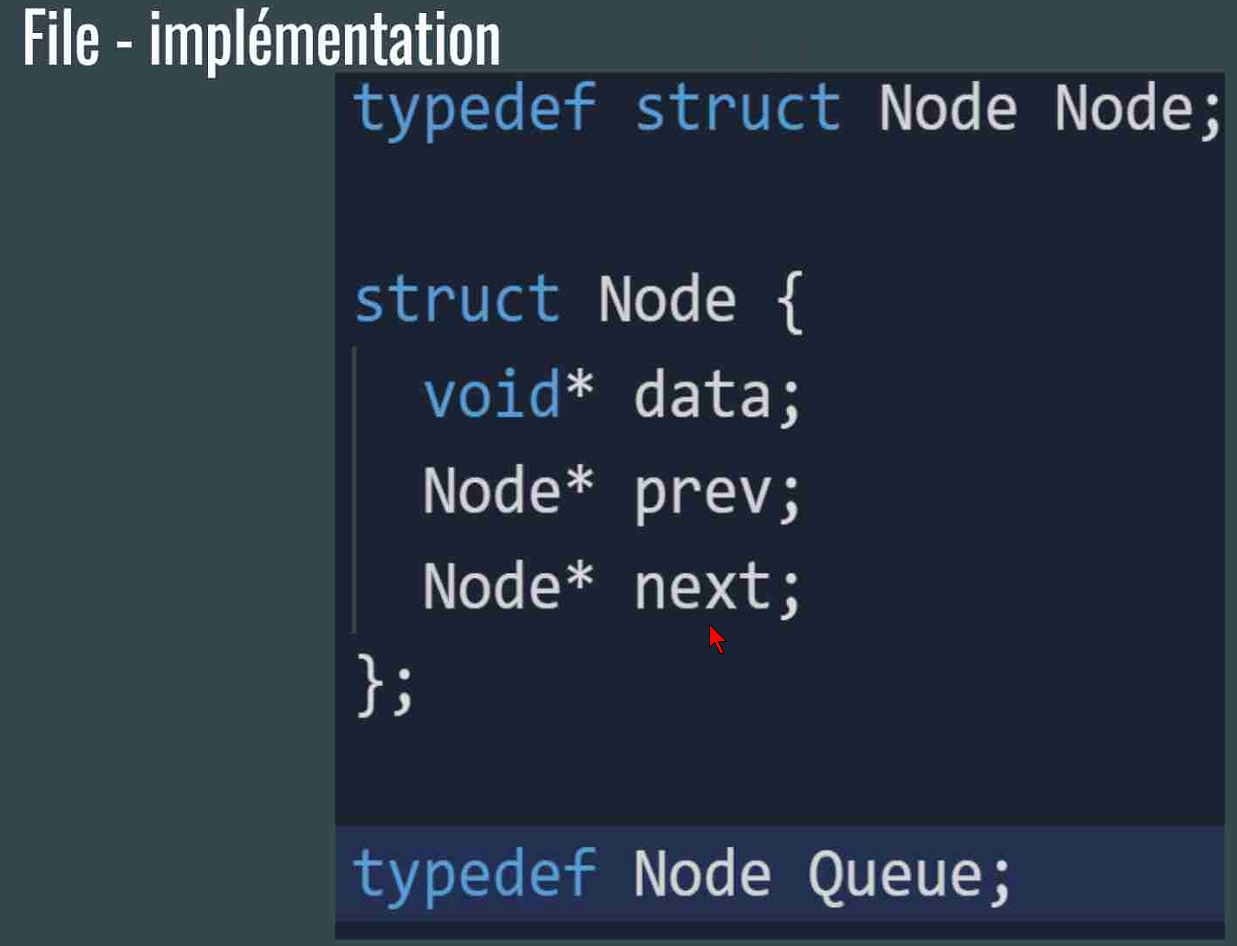
Le stack peut être utiliser pour faire du backtraking (l’action que tu dois faire dans métroidvania)

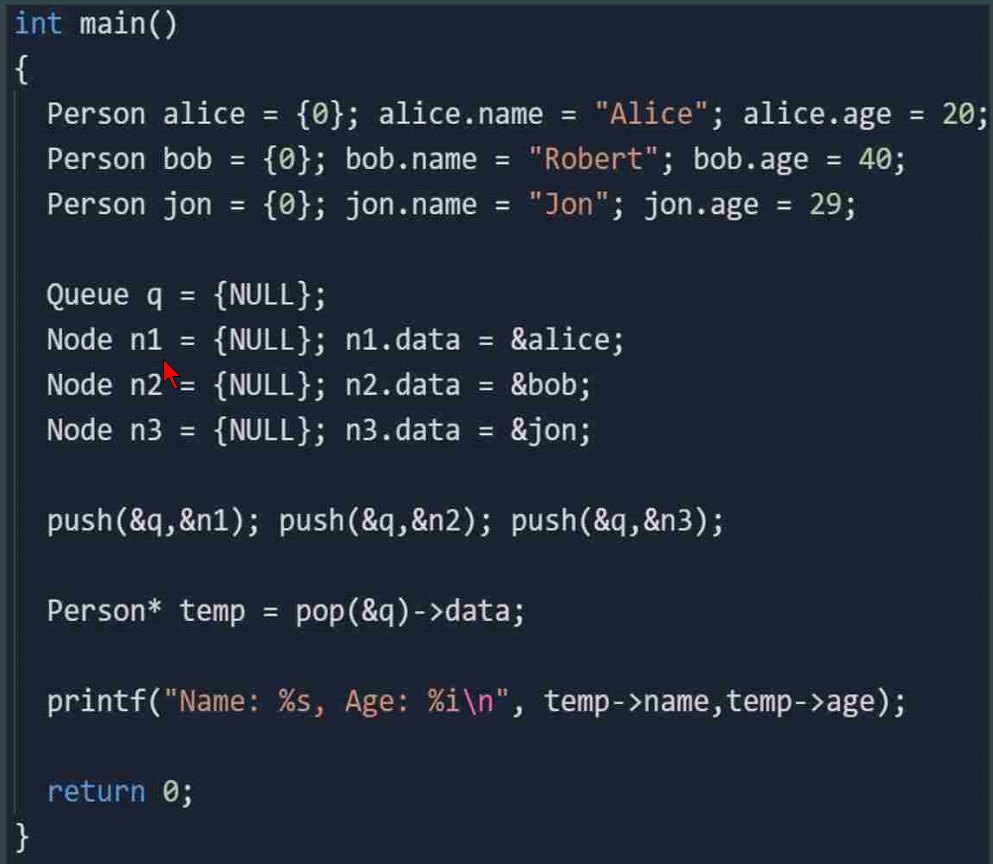
La file ou ‘’a queue’’

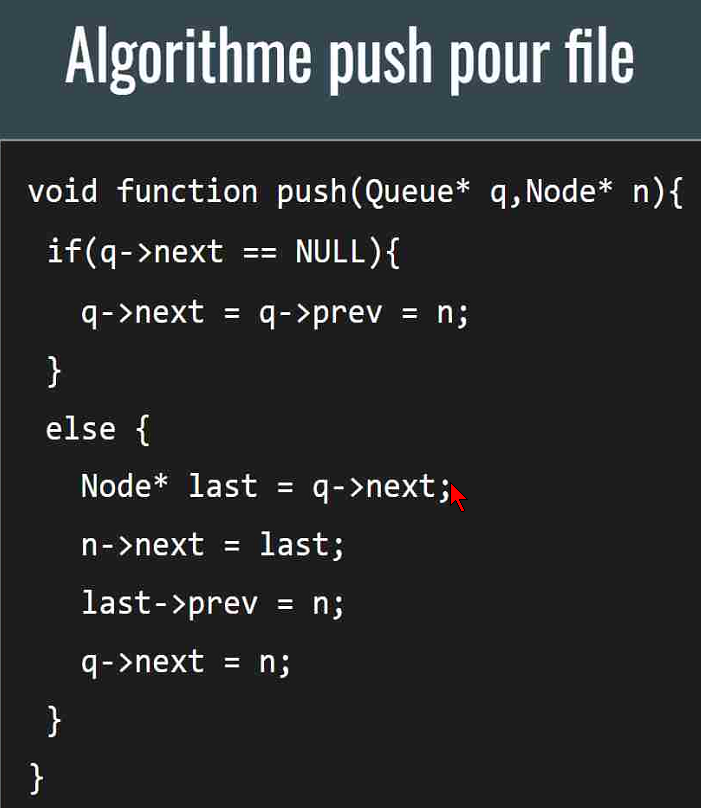
Les Files sont ‘’First in first out’’

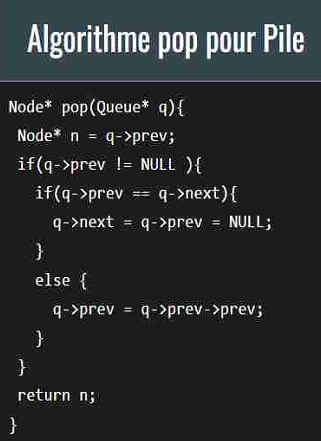
Pour faire une queue il serait mieux de faire une liste doublement chainer qui nous ferait faire une notation grand O : O(1)

Alors qu’avec une liste simplement chainer ca fait O(n)

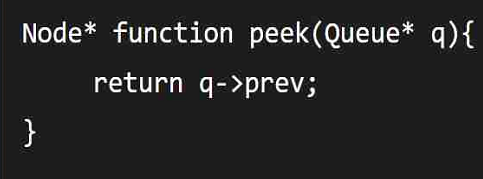








Pour peek:



Les queues servent a faire comme par exemple ‘’des priority queue’’ ca permet de priorisé les sons