

# Introduction au C TP2

## 1 Définition d'un type

1. Écrivez un fichier header pour un type liste simplement chaînée :
  - (a) Utilisez **struct** et **typedef** pour définir une liste d'entiers simplement chaînée.
  - (b) Déclarez les prototype des fonctions usuelles : **creer\_liste**, **liste\_est\_vide**, **ajouter\_debut**, **suivant**, **supprimer\_debut**, **ajouter\_fin**, **supprimer\_fin** .
2. Écrivez le fichier définissant ces fonctions. Assurez vous qu'elle fonctionne correctement en écrivant un programme qui les utilise

## 2 Pointeurs de fonction

1. Ajoutez aux fonctions de manipulation de listes déjà écrites une fonction **map**, qui étant donné une liste d'entiers et une fonction f qui à un entier associe un entier, modifie la liste en appliquant f à chaque élément de la liste de départ.
2. Utilisez cette fonction pour écrire
  - (a) la fonction **doubler** qui étant donné une liste d'entiers la modifie en multipliant par 2 chacun des entiers de la liste de départ
  - (b) la fonction **div2** qui étant donné une liste d'entiers renvoie la modifie en divisant par 2 (division entière) chacun des entiers de la liste de départ
3. Modifiez votre main pour utiliser ces deux fonctions et afficher le résultat

## 3 Menu de sélection

On utilisera ici les fonctions définies dans la première partie.

1. Modifiez votre définition de liste afin de rajouter une chaîne de caractères de manière statique pour chaque élément de la liste. Il faudra alors modifier certaines fonctions liées à la liste :  
Rajouter un argument aux fonctions d'ajout afin de donner la chaîne de caractères associée au maillon rajouté de la liste
2. Ecrivez une fonction qui parcourt la liste, si elle n'est pas vide, et affiche les chaînes de caractères de chaque maillon, une chaîne par ligne.
3. Modifiez la liste pour qu'elle soit doublement chaînée (rajout d'un champ correspondant à l'adresse de la case précédente). Modifiez les fonctions d'ajouts et de suppression pour qu'elles correspondent.
4. On va se servir d'un pointeur qui correspondra à la sélection de la case du menu. Créez les fonctions **item\_suivant** et **item\_precedent** qui renvoie l'adresse la case du nouveau maillon si il existe sinon renvoi l'adresse du maillon actuel.

5. Modifiez la fonction affichage afin de rajouter le maillon selectionné en argument afin d'avoir un affichage qui prenne en compte et mette en évidence l'item sélectionné, ex :

```
item 1
item 2
-> item 3
item 4
```

6. Rajoutez une entrée utilisateur récupérant un caractère afin de manipuler le menu directement, ex : "Entrez 1 pour sélectionner item suivant, 2 pour précédent,0 pour afficher la valeur de l'entier du maillon sélectionné".
7. Remplacez la déclaration statique des chaines de caractères par une déclaration dynamique (malloc).
8. Il ne manque plus qu'à remplacer les entier des maillons par des pointeurs de fonctions et le menu est opérationnel