```
BUT DU TD : ETUDE DU CODE (INTERFACE) DE DEUX MODULES DE LISTE CHAINEE

Question 1 :
-------------

Ecrire les prototypes des fonctions principales pour un module de
liste avec curseur.

Specification de la liste :

 - un pointeur sur le premier (resp. dernier élément)
 - un curseur définissant la notion d'élément courant de la liste
 - les opérations classiques sont :
     - créer ou détruire la liste
     - insérer, supprimer ou modifier un élément de la liste
     - se déplacer dans la liste
     - tester la position du curseur
     - examiner un élément
     - tester l'état de la liste

     Pour la suppression, il faut bien veiller à supprimer l'élément,
     et il faut donc utiliser une fonction de libération ... à voir
     avec eux, notamment quelles seront leurs idées.
     Idée retenue par Strandh et Olivier : définir une fonction de
     libération dans la structure liste.


Quelles modifications faut-il apporter si on veut que la liste soit
polymorphe hétérogène ?

           Mon idée : définir une liste chaînée de fonctions de
           libération et faire en sorte que chaque élément de la liste
           ait un pointeur vers la fonction correspondante (liste non
           obligatoire).

Justifier l'utilisation du curseur (quelles sont les opérations
intéressantes qu'il permet d'effectuer).

              Insertion au milieu de la liste.

Décrivez des mécanismes permettant de généraliser ces opérations.

         cf le code à venir : une pile ou une notion de position.


Question 2 :
-------------

Lecture de gqueue.h et de list.h et description des fonctions offertes
par ces deux implantations de listes.

Quelques remarques :

 - gestion de la déallocation : statique dans list et dynamique dans
   gqueue => deux types de polymorphismes (hum hum).

 - gestion des accès type tableau : pile dans gqueue et position dans
   list => un élément peut se retirer lui-même de la liste, mais il
   faut définir une structure de données destinée à interagir avec la
   liste.

 - position du curseur et conséquences sur la structure interne de la
```



```
   liste : gqueue, le curseur pointe sur un lien et non sur un
   élément.

 - gestion de la fonction map : quelles sont les solutions possibles
   pour passer des arguments supplémentaires à la fonction map (static
   ou void *)

 - notion de beginning dans list.h


Question 3 :
-------------

Leur demander de coder la structure de données pour gqueue (voir
notamment qu'on peut utiliser une gqueue pour la pile des curseurs) et
pour list et, si ils ont été rapides leur demander, pour gqueue de
programmer quelques fonctions simples.


Question 4 :
-------------

Leur demander de programmer, avec le module list, les fonctions
suivantes :

 - récupérer une liste d'entiers en ligne de commande et
   - la transformer par la transfo de Syracuse
   - produire la liste des éléments pairs

Question maison :
------------------

Programmer la fonction map chez gqueue en utilisant les fonctions de
bibliothèque et l'appliquer aux deux exemples précédents.
```