TD 3 & 4 – Travail sur les listes

Exercice 1 – test de la présence d'un élément

On veut étendre le type **liste récursive** avec l'opération **est_présent**, qui teste la présence d'un élément donné, et dont la signature est :

est_présent : liste x élément → boolean

et dont la sémantique est :

```
est_présent(l, e) =
      si l = liste_vide alors faux
      sinon si premier(l) = e alors e
      sinon est_présent(fin(l))
```

a) Ecrire la fonction est_présent dans le cas d'une représentation contiguë

```
Class Liste {
       int[] tab ;
}
```

b) Ecrire la fonction est_présent dans le cas d'une représentation chaînée

```
Class Place {
       int valeur;
       Place suivant;
}

Class Liste {
       int longueur ;
       Place tete;
}
```

Exercice 2 – inversion d'une liste

On veut étendre le type **liste** avec l'opération **renverse**, qui inverse l'ordre de ses éléments, et dont la signature est :

renverse : liste → liste

a) On définit un premier axiome trivial :

renverse(listevide) → listevide

Parmi les quatre propriétés suivantes, expliquer laquelle doit être utilisée comme axiome supplémentaire pour définir la sémantique.

1. ∀ l : liste, renverse(l) = renverse(fin(l)) & [premier(l)]
2. ∀ l : liste, renverse(l) = fin(renverse(l)) & [premier(l)]
3. ∀ l : liste, renverse(l) = fin(l) & [premier(l)]
4. ∀ l : liste, renverse(l) = renverse([premier(l)] & fin(l))

b) Ecrire le programme qui réalise cette opération dans le cas d'une représentation contiguë (cf exo 1a)
c) Ecrire le programme qui réalise cette opération dans le cas d'une représentation chaînée (cf exo 1b)

Exercice 3 – course de ski

Une course de ski fait concourir n skieurs qui se sont préalablement inscrits. Le problème est de mettre à jour le classement après chaque arrivée.

a) proposer une structure de données pour ce problème ;
b) écrire l'algorithme permettant d'afficher le classement ;
c) écrire l'algorithme permettant de mettre à jour la liste après l'arrivée d'un coureur ;
d) écrire l'algorithme permettant de supprimer du classement un coureur disqualifié

Exercice 4 – le palindrome

Il existe dans la langue française des mots qui se lisent de la même façon de gauche à droite et de droite à gauche, comme par exemple « elle » ou « rotor ». Il est également ainsi de certaines phrases comme « Esope reste ici et se repose ». On appelle *palindrome* ces mots ou ces phrases qu'on peut lire dans les deux sens. Plus généralement, sur des listes d'éléments quelconques, on peut chercher à savoir si une liste est égale à sa liste inversée, opération que nous appellerons **`palindrome`**.

`palindrome : liste → booléen`

a) En utilisant la spécification des liste itératives, donner les axiomes de l'opération palindrome
b) En utilisant la spécification des liste récursives, donner les axiomes de l'opération palindrome
c) Donner un algorithme possible dans le cas de l'implantation contiguë
d) Expliquer pourquoi l'implantation chaînée simple des listes n'est pas satisfaisante pour réaliser cette opération. Qu'en est-il des variantes « liste chaînée avec tête fictive » et « liste chaînée » circulaire
e) Donner un algorithme possible dans le cas de la variante « liste doublement chaînée »