

Nom	
Prénom	
Groupe	

Note	/ 5
------	-----

## Algorithmique Contrôle 1 - Partie 1

INFO-SUP S1

EPITA

9 Nov. 2020 - 8 : 30

- Ceci est la partie 1 de l'épreuve - Vous devez rendre les deux parties !
- Vous devez répondre directement **sur ce sujet**.
  - Répondez dans les espaces prévus, les réponses en dehors ne seront pas corrigées.
  - Aucune réponse au crayon de papier ne sera corrigée.
- La présentation est notée.

### Exercice 1 (Types Abstraits : Listes récursives – 5 points)

Supposons le type abstrait algébrique *Liste récursive* vu en cours et rappelé ci-dessous.

#### TYPES

liste, place

#### UTILISE

élément

#### OPÉRATIONS

$listevide$  :  $\rightarrow$  liste  
 $tête$  : liste  $\rightarrow$  place  
 $contenu$  : place  $\rightarrow$  élément  
 $premier$  : liste  $\rightarrow$  élément  
 $cons$  : élément  $\times$  liste  $\rightarrow$  liste  
 $fin$  : liste  $\rightarrow$  liste  
 $succ$  : place  $\rightarrow$  place

#### PRÉCONDITIONS

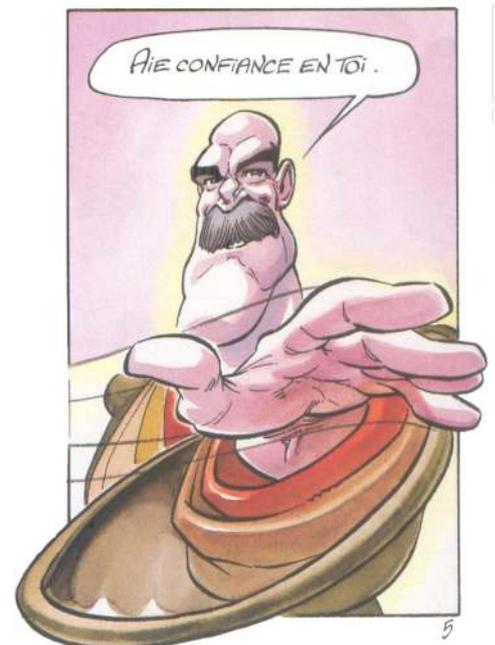
$tête(\lambda)$  est-défini-ssi  $\lambda \neq listevide$   
 $fin(\lambda)$  est-défini-ssi  $\lambda \neq listevide$   
 $premier(\lambda)$  est-défini-ssi  $\lambda \neq listevide$

#### AXIOMES

$premier(cons(e, \lambda)) = e$   
 $fin(cons(e, \lambda)) = \lambda$   
 $contenu(tête(\lambda)) = premier(\lambda)$   
 $succ(tête(\lambda)) = tête(fin(\lambda))$

#### AVEC

liste  $\lambda$   
 élément  $e$



On se propose d'étendre les propriétés de ce type en lui permettant :

- de rechercher un élément dans une liste
- de concaténer deux listes.

La recherche d'un élément parmi ceux d'une liste ne retournera la place correspondante à celui-ci que s'il existe. Dès lors nous avons deux opérations pour la recherche, celle qui détermine la présence effective de l'élément et celle qui détermine la place de ce dernier s'il est présent. La concaténation n'a quant à elle besoin d'aucune opération auxiliaire. Nous considérerons donc les trois opérations suivantes :

**OPÉRATIONS**

*est-présent* : élément  $\times$  liste  $\rightarrow$  booléen

*rechercher* : élément  $\times$  liste  $\rightarrow$  place.

*concaténer* : liste  $\times$  liste  $\rightarrow$  liste

1. Donner les axiomes déduisant une valeur pour la recherche d'un élément  $e$  parmi ceux d'une *liste récursive*  $\lambda$ . Vous préciserez les PRÉCONDITIONS s'il y en a.

---

---

---

---

---

---

---

---

---

---

2. Donner les axiomes déduisant une valeur pour l'opération de concaténation de deux *listes récursives*  $\lambda$  et  $\lambda 2$ . Vous préciserez les PRÉCONDITIONS s'il y en a.

---

---

---

---

---