

## Les listes chaînées

### Solutions:

Soit les déclarations de types et la procédure TRUC suivantes :

**Type** ptElement = ^Liste

**Type** Liste= **Structure**

Info : Entier

Suivant : ptElement

**Fin Structure**

**Procédure** TRUC ( E/S p : ptElement, S x : Entier)

**Déclaration** p1 : ptElement

**Début**

si (p<>null) alors

**début**

    x ← p^.info

    p1 ← p

    p ← p^.suivant

    libérer (p1)

**fin**

sinon

**écrire** ('la pile est vide')

**Finsi**

**Fin Procédure**

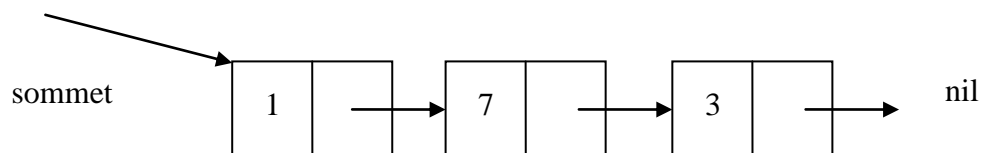
### Solution à la question 1 :

Réalisez un schéma de la structure déclarée avant la procédure.

(Liste)

### Solution à la question 2 :

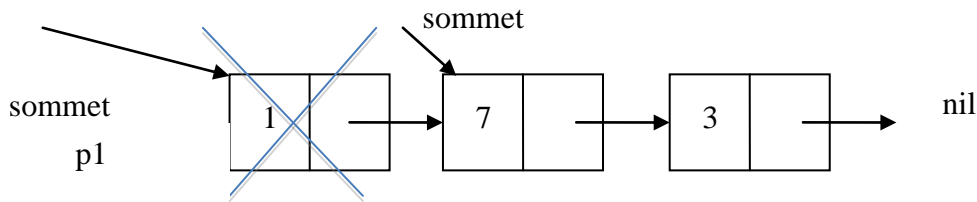
Réaliser un schéma de la structure résultant de l'application de la procédure après chaque instruction (numéro d'instruction et schéma de la liste) sachant que la structure contient déjà les éléments suivant.



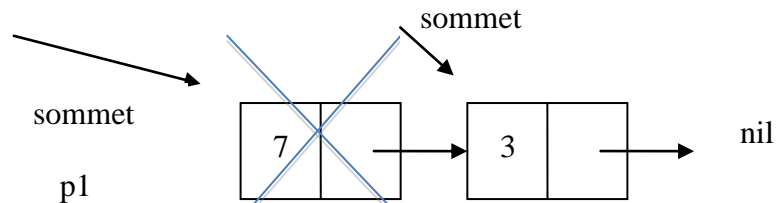
1. Vous justifierez votre schéma (déroulement du code de la procédure « à la main »)
2. truc (q, x) ;
3. truc (q, x) ;

1. L'appel de la procédure TRUC se fait avec q (soit sommets) qui ne vaut pas nil et x qui vaut 1  
⇒ on affecte à x la valeur 1 (sommets^.info)  
⇒ p1 pointe sur le sommets

- ⇒ *sommet* passe au suivant (*sommet* <- *sommet*^.suivant)
- ⇒ On libère la cellule *p1*
- ⇒ La valeur de *x* qui est renvoyée est 1.



2. On dépile la valeur 7.
  - ⇒ on affecte à *x* la valeur 7 (*sommet*^.info)
  - ⇒ *p1* pointe sur le *sommet*
  - ⇒ *sommet* passe au suivant (*sommet* <- *sommet*^.suivant)
  - ⇒ On libère la cellule *p1*
  - ⇒ La valeur de *x* qui est renvoyée est 7.



**Question 3 :**

Que fait cette procédure ? Comment pourrait-on la renommer pour qu'elle soit plus explicite ?

Cette procédure dépile un élément de la Liste qui en fait est une pile. Rappel : On dépile toujours en enlevant l'élément qui est *sommet*  
 On pourrait la renommer Procédure dépile ((E/S) p: ptelement, (S) x : entier)).

**Question 4 :**

Documenter cette procédure :

**Type** ptElement = ^Liste

**Type** Liste= **STRUCTURE**

Info : Entier

Suivant : ptElement

**Fin STRUCTURE**

**Procédure** TRUC ( E/S p : ptElement, S x : Entier)

// procédure qui dépile un élément en début de pile

// une entrée résultat p de type pointeur

// un résultat x de type entier qui peut être affiché

**Déclaration** p1 : ptElement

**Début**

si (p<>null) alors // si la liste est non vide

**début**

$x \leftarrow p^{\wedge}.info$  // *x prend la valeur de l'entier contenu dans la cellule*

$p1 \leftarrow p$  // *p1 pointe sur p*

$p \leftarrow p^{\wedge}.suivant$  // *p avance d'une cellule*

libérer (p1) // *on libère la cellule*

**fin**

**sinon**

écrire ('la pile est vide')

**Finsi**

**Fin Procédure**