

## Les listes

### Matériel fourni :

- les notes de cours sur les listes ;
- une unité `U_Exemples_Listes.pas`.

## 1 L'unité `U_Liste`

### 1.1 Programmation de l'unité

**Question 1.** Réalisez l'unité `U_Liste`. La partie interface doit comprendre

1. la déclaration du type `LISTE` ;
2. la déclaration de la constante `LISTEVIDE` ;
3. les déclarations des six opérations primitives.

Par la suite, vous ajouterez d'autres opérations, non primitives mais utiles.

### 1.2 Vérification

L'unité `U_Exemples_Listes.pas` définit une variable qui est un tableau de indexé de 1 à `NBLISTES` dont chaque élément est une liste de longueur égale à l'indice, et dont les éléments ont été choisis au hasard (sauf pour la dernière dont les éléments sont fixés). Les éléments sont nécessairement des entiers (positifs ou nuls) et par conséquent le type `ELEMENT` (défini dans l'unité `U_Element`) doit être `CARDINAL` (ou `INTEGER`).

**Question 2.** Vérifiez la validité de votre unité en utilisant l'unité `U_Exemples_Listes.pas`. Votre programme de test doit afficher pour chacune des listes, le premier et le deuxième élément (lorsqu'il existe) de la liste.

## 2 Autres opérations sur les listes

### 2.1 Longueur d'une liste

**Question 3.** Programmez la longueur d'une liste.

**Question 4.** Vérifiez votre calcul de longueur avec les listes de l'unité `U_Exemples_Listes.pas`. La longueur de la liste numéro  $n$  est  $n$ .

### 2.2 Affichage des éléments d'une liste

**Question 5.** Programmez l'affichage à l'écran des éléments d'une liste.

**Question 6.** Testez votre programme avec les listes de l'unité `U_Exemples_Listes.pas`.