

PROGRAMMATION IMPÉRATIVE

TD n° 6 : Pointeurs et packages

Le but de ce TD est de créer un package `Listes`. La spécification du package est la suivante.

Listing 1 – Listes.ads

```
1 with Ada.Text_IO, Ada.Float_Text_IO;
2 use Ada.Text_IO, Ada.Float_Text_IO;
3
4 package Listes is
5
6 type Couple;
7 type Liste is access Couple;
8 type Couple is record
9     val: Integer;
10    suiv: Liste;
11 end record;
12
13 function Premier(L: Liste) return Integer;
14 — retourne le premier element de la liste
15
16 function Suivant(L: Liste) return Liste;
17 — retourne la liste privee du premier element
18
19 function Longueur(L : Liste) return Integer;
20 — retourne la longueur de la liste
21
22 function Appartient(x : Integer; L : Liste) return Boolean;
23 — retourne True si x est dans la liste L et false sinon
24
25 procedure Ajouter_debut(L : in out Liste; e : in Integer);
26 — Ajoute l'entier e au debut de la liste L
27
28 procedure Ajouter_fin(l : in out liste, e : in integer);
29 — Ajoute l'entier e a la fin de la liste L
30
31 end Listes;
```

Compléter le corps du package suivant.

Listing 2 – Listes.adb

```
1 with Ada.Text_IO, Ada.Float_Text_IO;
2 use Ada.Text_IO, Ada.Float_Text_IO;
3
4 package body Listes is
5
6 function Premier(L: Liste) return Integer is
7 begin
8 — A COMPLETER
```

```

9 end Premier;
10
11
12 function Suivant(L: Liste) return Liste is
13 begin
14 — A COMPLETER
15 end Suivant;
16
17
18 function Longueur(L : Liste) return Integer is
19 begin
20 — A COMPLETER
21 end Longueur;
22
23 function Appartient(x : Integer; L : Liste) return Boolean is
24 begin
25 — A COMPLETER
26 end Appartient;
27
28
29 procedure Ajouter_debut(L : in out Liste; e : in Integer) is
30 begin
31 — A COMPLETER
32 end Ajouter_debut;
33
34 procedure Ajouter_fin(l : in out liste , e : in integer) is
35 begin
36 — A COMPLETER
37 end Ajouter_fin;
38
39 end Listes;

```