Université de Perpignan — Année 2003-2004
Licence 2e année Informatique — S. Graillat

Programmation impérative

TD n° 5 : Gestion des tableaux non contraints et des articles

1 Les tableaux non contraints

Le langage Ada possède le concept de tableau non contraint : dans celui-ci, les contraintes d'indices ne sont pas données. Considérons par exemple,

```
type Vecteur is array(Integer range <>) of Float;
```

(<> se lit "boîte"). On établit ainsi que `Vecteur` est le nom d'un type tableau unidimensionnel de composants `Float`, muni d'indices de type `Integer`. Cependant, les bornes inférieures et supérieures ne sont pas données : `range <>` est là pour représenter la notion d'information à fournir ultérieurement.

Quand nous déclarons des objets de type `Vecteur`, nous devons fournir les bornes. Par exemple,

```
V : Vecteur(1..5);
```

Exercice 1. Écrire une procédure qui remplace chaque valeur d'un tableau d'entiers par son carré. Utilisez cette procédure sur un tableau de 5 et un tableau de 10 entiers.

Exercice 2. Écrire une procédure qui prend en paramètre un tableau de string de taille quelconque et qui échange l'ordre de la première moitié et de la deuxième moitié du tableau. Si le tableau contient un nombre impair d'éléments, l'élément du milieu restera à sa place.

Exemples :

```
(1,2,3,4,5,6) --> (4,5,6,1,2,3)
(1,2,3,4,5,6,7) --> (5,6,7,4,1,2,3)
```

Vous testerez cette procédure sur plusieurs tableaux de tailles variées, paires et impaires.

2 Les articles

Exercice 3. On définit le type `Point` par

```
type Point is record
   X : Float;
   Y : Float;
end record;
```

Écrire une procédure `Distance_entre_Points` qui donnera la distance entre deux éléments de type `Point`. On rappelle que si $P_1 = (X_1, Y_1)$ et $P_2 = (X_2, Y_2)$ sont deux points, on a alors

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}.$$

Cette procédure devra contenir la procédure `EntrerPoint` qui demande d'entrer les coordonnées d'un point et la fonction `Distance` qui prend en paramètre deux points et renvoie la distance entre ces deux mêmes points. Le prototype du programme est le suivant.

Listing 1 – Distance_entre_Points.adb

```
1  with Ada.Text_IO, Ada.Float_Text_IO, Ada.Numerics.Elementary_Functions;
2  use Ada.Text_IO, Ada.Float_Text_IO, Ada.Numerics.Elementary_Functions;
3
4  procedure Distance_entre_Points is
5
6  type Point is record
7     X : Float;
8     Y : Float;
9  end record;
10
11 procedure EntrerPoint(P : out Point) is
12
13 begin
14    -- A REMPLIR
15 end EntrerPoint;
16
17 function Distance(P1 : in Point; P2 : in Point) return Float is
18
19 begin
20    -- A REMPLIR
21 end Distance;
22
23
24 begin
25    -- A REMPLIR
26 end Distance_entre_Points;
```