

# TD Info n°8 : Tableaux

ECE3 Lycée Carnot

8 décembre 2011

## 1 Les tableaux en Pascal

Jusqu'à présent, nous n'avons travaillé qu'avec des types de données en PASCAL qui stockent une seule valeur à la fois (entière, réelle, ...). Il est pourtant souvent utile de stocker d'un seul coup plusieurs valeurs. Nous verrons bientôt que c'est quelque chose que nous aurons naturellement envie de faire quand nous travaillerons avec des matrices. C'est également très utile quand on veut travailler avec des polynômes (pour lesquels il faut stocker un certain nombre de coefficients), exemple que nous utiliserons principalement pour l'instant. La façon la plus simple de représenter informatiquement un polynôme est de ne conserver que les coefficients dans un tableau, par exemple par degré croissant (c'est plus facile si on veut pouvoir manipuler des polynômes de degré variable dans un même programme). Ainsi, le polynôme  $x^4 + 3x^3 + 5x^2 + 9x - 4$  sera représenté par le tableau  $[-4; 9; 5; 3; 1]$  (les programmes que nous écrirons ne feront jamais apparaître la variable  $x$ , comme c'était déjà le cas par exemple dans le programme de résolution d'équations du second degré que nous avons écrit en début d'année). Ça tombe bien, il existe en PASCAL un type ARRAY qui est en fait un tableau de nombres dont on spécifie à l'avance la longueur et le type de données qu'il permettra de stocker. On peut ainsi déclarer des variables de la façon suivante :

```
VAR t : ARRAY[0..10] OF real;
```

Tout ce qui est après les deux points : ARRAY[0..10] OF real désigne simplement le type de la variable t : il s'agit d'un tableau dont les cases seront numérotées de 0 à 10 (autrement dit, un tableau à 11 cases) et dont chaque case contient un réel (on peut naturellement remplacer 0 et 10 par des entiers quelconques, et rela par n'importe quel autre type « simple »). La variable t pourra donc par exemple représenter un polynôme de degré 10 (qui a bien 11 coefficients). Comme toujours, Pascal est très pontilleux sur la syntaxe, à respecter scrupuleusement. Collé au ARRAY se trouvent toujours des crochets à l'intérieur desquels deux entiers séparés par deux points (pas plus ni moins) indiquent la numérotation des éléments du tableau.

Une telle variable nécessite d'être manipulée un peu différemment de ce qu'on a l'habitude de faire avec les types plus simples. Notamment, les opérations d'affectation vont être plus compliquées. Par exemple, Pascal ne comprend pas une instruction comme  $t := 1$ ; (la variable t n'est pas un nombre, on ne pas lui donner une valeur numérique). Si on veut remplir le tableau en mettant des 1 partout, il faut expliquer à Pascal que chaque case devra, l'une après l'autre, recevoir la valeur 1. Pour désigner à Pascal la case numéro  $i$  du tableau  $t$ , on utilisera la notation  $t[i]$ . Ainsi, en tapant une instruction comme  $t[6] := 1$ , on écrit le nombre 1 dans la case numéro 6 du tableau t. Si on veut mettre des 1 dans toutes les cases du tableau, on écrira une boucle FOR, par exemple

```
FOR i := 0 TO 10 DO t[i] := 1;
```

De la même façon, demander à l'utilisateur de saisir successivement tous les éléments d'un tableau demandera en général d'utiliser une boucle. Même le simple affichage des valeurs contenues dans le tableau nécessite une boucle (WriteLn(t) ne marchera pas). Bref, c'est très lourd...

## Définitions de types

On peut aussi, tant qu'on y est, définir nous-même de nouveaux types faisant intervenir des tableaux, histoire d'alléger un peu les notations dans les programmes. Une déclaration de type se fait dans l'en-tête du programme, par une ligne du genre `TYPE blabla=ARRAY[p..n] OF typesimple`, où  $n$  et  $p$  sont deux entiers et `typesimple` un type « classique », réels, entiers etc. Ainsi :

```
TYPE polynome=ARRAY[0..10] OF real; VAR p,q :polynome;
```

définit un nouveau type, le type `polynome`, qui est représenté par des tableaux de 11 réels, puis deux variables  $p$  et  $q$  de ce nouveau type, donc deux tableaux à 11 réels.

## 2 Exercices sur les polynomes

Dans tous les programmes nécessitant d'utiliser des polynomes, on prendra l'habitude de définir le type `polynome` par :

```
TYPE polynome=ARRAY[0..99] OF real;
```

Ceci nous permet de travailler avec des polynomes dont le degré n'excède pas 99. Naturellement, si on travaille avec des polynomes de petit degré, la plupart des éléments de notre tableau seront égaux à 0.

Pour s'entraîner, nous allons écrire quelques petits programmes utilisant ce nouveau type :

1. Écrire un programme permettant de stocker dans un tableau les coefficients d'un polynome  $P$  saisis par l'utilisateur (on demandera d'abord le degré du polynome pour éviter de faire taper 95 zéros à l'utilisateur).
2. Compléter le programme précédent en lui faisant calculer la valeur de  $P(0)$  et de  $P(1)$  (et même de  $P(2)$  si vous êtes courageux).
3. Compléter le premier programme pour lui faire calculer le polynome dérivé du polynome saisi.