

# Fonctions en PASCAL

ECE3 Lycée Carnot

Mardi 24 novembre 2009

## Principe

Le langage PASCAL dispose d'un certain nombre de fonctions prédéfinies qu'on a déjà eu l'occasion d'utiliser dans nos programmes, comme la fonction racine carrée (**sqrt**) ou même les fonctions logarithme ou exponentielle (respectivement notées **ln** et **exp**). Il y a toutefois quelques manques criants (par exemple, pas de façon simple d'écrire des puissances, ou pas de raccourci pour les factorielles), et il est de toute façon utile de temps à autre de définir de nouvelles fonctions pour une utilisation limitée à un seul programme (par exemple, on cherche à effectuer via Pascal un tableau de valeurs pour la fonction  $f : x \mapsto \frac{x^2 - 3}{x^2 + 1}$ ). Cela tombe bien, puisqu'on peut définir à l'intérieur d'un programme Pascal des fonctions de toutes sortes.

## Syntaxe

La définition d'une fonction se fait dans l'en-tête du programme, par exemple à la suite des déclarations de variables. Elle constitue en fait un sous-programme qui peut contenir ses propres déclarations de variables et dont le corps sera encadré par un BEGIN et un END ; (pas de END. qui est réservé pour la fin du programme complet). En effet, les fonctions Pascal sont à prendre dans le sens le plus général possible, et peuvent prendre plusieurs variables de types différents et renvoyer un résultat d'à peu près n'importe quel type également. La première chose à préciser à Pascal quand on définit une fonction sera donc le nombre de variables ainsi que leur type, et le type du résultat. cela se fait de la façon suivante :

```
FUNCTION nomfonction (var1 : type1 ; var2 : type2 ... ; vark : typek) : typeresultat ;
```

Ainsi, la déclaration de la fonction factorielle aura pour en-tête :

```
FUNCTION factorielle (n : longint) :longint ;
```

Les variables apparaissant dans cet en-tête (ici l'entier  $n$ ) sont définies et typées dans cet en-tête, inutile donc de les déclarer à nouveau ensuite. Par contre, on peut avoir besoin de déclarer à l'intérieur de la déclaration de fonction d'autres variables servant à calculer la valeur de la fonction (ici un indice  $i$  pour faire tourner une boucle calculant la valeur de la factorielle). Ces variables seront définies comme d'habitude par une ligne du type VAR nom : type ; à l'intérieur de la déclaration de fonction, et ne seront utilisables qu'à l'intérieur de cette même déclaration (on parle de variables **locales** par opposition aux variables **globales** qu'on définit pour l'intégralité du programme), et « disparaîtront » dès que le calcul de la valeur de la fonction sera terminé. On peut même donner un même nom à une variable globale et à une variable locale, il n'y aura aucune interaction entre les deux (c'est tout de même très fortement déconseillé !). Le corps de la déclaration de fonction proprement est constitué d'instructions (comme n'importe quel corps de programme), mais il est interdit de faire apparaître des WriteLn ou des ReadLn dedans, et il doit par contre **nécessairement** contenir une ligne du type

```
nomfonction :=valeur ;
```

qui sert à définir la fonction. La valeur dépendra a priori des variables qu'on lui a associées, et cette fonction pourra ensuite être appelée à l'intérieur du programme par la commande `nomfonction(var1,var2,..)` (comme vous le noteriez naturellement en maths). Ainsi, la fonction factorielle peut être définie par la déclaration suivante :

```
FUNCTION factorielle (n :integer) : longint ;
VAR i :integer ; p :longint ;
BEGIN
p :=1 ;
FOR i :=1 TO n DO p :=p*i ;
factorielle :=p ;
END ;
```

Si on insère ensuite dans notre programme une ligne du genre `a :=factorielle(14)` ; Pascal calculera  $14!$  à l'aide de la définition précédente et affectera la valeur obtenue à la variable `a`.

## Petits exercices

1. Écrire une déclaration de fonction Pascal prenant comme variables un réel et un entier, et calculant la puissance correspondante.
2. Écrire un programme Pascal affichant les images de tous les entiers compris entre  $-5$  et  $5$  par la fonction  $f : x \mapsto \sqrt{x^2 + 3}$ .
3. Écrire un programme Pascal calculant le coefficient binomial  $\binom{n}{k}$  en utilisant la fonction factorielle décrite ci-dessus ( $n$  et  $k$  étant deux entiers choisis par l'utilisateur).