

La semaine dernière, nous avons simplement vu comment notre nouvel ami Pascal s'y prenait pour communiquer avec nous, à savoir sa façon un peu particulière de lire et d'écrire. Comme nous aurons du mal à faire des programmes intéressants, uniquement avec ces deux concepts, je vous donne tout de suite une liste d'instructions qui, contrairement aux Read et Write, ne vont rien effectuer de visible à l'exécution du programme, mais seront la plupart du temps des calculs intermédiaires menant à la résolution du problème posé.

- La plus importante est l'instruction d'affectation, qui permet de stocker une valeur dans une variable (préalablement déclarée, naturellement). Vous me direz « Ne fait-on pas un ReadLn pour ce genre de choses ? ». Oui, si vous voulez que ce soit l'utilisateur qui choisisse la valeur à stocker, mais pas si celle-ci découle d'un calcul. Reprenons l'exemple d'une résolution d'équation du second degré : c'est l'utilisateur qui choisit a , b et c , mais ensuite la machine doit calculer la valeur de Δ . Les instructions d'affectation se font en Pascal sous la forme **a := valeur** ; ce texte signifiant qu'on affecte la valeur à droite du signe := dans la variable portant le nom indiqué à gauche. Ainsi, si on veut faire calculer un discriminant à Pascal, on lui dira **delta := b*b-4*a*c** ; . Pascal est naturellement capable d'effectuer les opérations élémentaires, mais comprend aussi les fonctions usuelles comme ln ou exp.
- Les instructions conditionnelles permettent d'effectuer une action (un calcul par exemple) si une certaine condition est vérifiée, et de faire autre chose sinon. La syntaxe intuitive est à peu près celle-ci : SI la condition est vérifiée, ALORS il faut faire ceci, SINON il faut faire cela. Ça tombe bien, la syntaxe Pascal est essentiellement la même, utilisant les trois mots-clés **IF**, **THEN** et **ELSE**. Il est à noter que l'ensemble d'une instruction conditionnelle est considérée comme une seule commande, il n'y a donc qu'un seul ; à mettre à la fin de l'instruction suivant le **ELSE**. Le **ELSE** en question est d'ailleurs facultatif : on peut décider de donner un ordre à Pascal si une certaine condition est vérifiée, mais de ne rien faire dans le cas contraire, auquel cas un **IF** suivi d'un **THEN** suffit. Un exemple pour clarifier les choses :

```
PROGRAM valeur_absolue ;
VAR x : real ;
BEGIN
WriteLn('Entrez la valeur de x. ');
ReadLn(x) ;
IF x < 0 THEN WriteLn(-x) ELSE WriteLn(x) ;
END.
```

Ce programme calcule la valeur absolue d'un nombre entré par l'utilisateur. Les conditions apparaissant derrière le **IF** seront la plupart du temps du style $x > a$, ou $x = a$ (notez que dans ce dernier cas, le signe = est bien compris par Pascal comme une demande de vérifier une égalité, et non pas une affectation qui utiliserait un :=).

- rien à voir avec les instructions, je vous donne une petite liste des types les plus fréquemment utilisés en Pascal (je vous rappelle que le type est le mot qu'on indique après le : lors d'une déclaration du variable pour indiquer de quel sorte d'objet il s'agit. Nous avons déjà vu **STRING** (chaîne de caractère) et **REAL** (nombre réel), il existe aussi **INTEGER** (nombre entier), **LONGINT** (gros nombre entier, nous verrons un jour l'utilité de ce truc), et bien d'autres que nous verrons au cours de l'année.

Exercices

- Écrire un programme demandant trois nombres à l'utilisateur et calculant leur moyenne.
- Écrire un programme calculant la plus grande de deux valeurs saisies par l'utilisateur.
- Faire la même chose avec trois valeurs.
- Écrire un programme effectuant la résolution des équations du premier degré, en faisant bien attention aux cas particuliers.
- Écrire un programme effectuant la résolution d'équations du second degré.