

# Cours « Algorithmique »

*2<sup>ème</sup> année Licence*

## Chapitre 3 (Suite):

**Notions d' « Algorithmique »**

## Instructions de base

### Instruction d'affectation $\leftarrow$ :

l'affectation est une opération qui fixe une nouvelle valeur à une variable.

Exemple :  $x \leftarrow 5$ .

Dans une instruction d'affectation, on trouve :

- à gauche de la flèche, un **nom de variable** déjà déclarée.
- à droite de la flèche, une **expression**.

**Définition d'une expression** : Une expression est une formule de calcul constituée **d'opérandes** et **d'opérateurs**.

Les opérandes sont soit des **constantes**, des **variables** ou des **fonctions**

**Exemples:**

$$\text{Delta} = (b * b) - 4 * (a * c); \quad // \text{ ax}^2 + \text{bx} + \text{c} = 0$$

$$\text{Surface} = \text{longueur} * \text{largeur};$$

$$Y = \text{Tronc}(x);$$

**Règles d'évaluation des expressions** : Il existe une certaine **priorité** entre les opérateurs



- Le “+” et “-” dans 1 sont Les opérateurs arithmétiques unaires (signes). Exemple : +5 et -5.
- En cas de priorités identiques (1 ou 2 ou 3 ou 4), les calculs s’effectuent de gauche à droite.
- Les parenthèses permettent de surpasser ces règles de priorité, en forçant le calcul préalable de l’expression qu’elles contiennent.

**Exercice** : Evaluer les expressions suivantes sachant que :

$$A = 3, \quad B = 5, \quad C = 2, \quad D = 3, \quad F = -1, \quad H = 1, \quad L = -2$$

$$E \leftarrow A+B*C; \quad \mathbf{E = 13}$$

$$E \leftarrow A+B*C+D; \quad \mathbf{E = 16}$$

$$E \leftarrow (A-B)+C*D; \quad \mathbf{E = 4}$$

$$E \leftarrow (A+B/(H+L))*B-(C+F*D); \quad \mathbf{E = ????$$

$$E = (3 + 5 / -1) * 5 - (2 + -3)$$

$$E = -10 - -1 = -9$$

+	-	Non		*	/	^	div	mod	et		+	-	ou	ouex		=	<>	≤	≥	≠
1				2							3					4				

## Instructions de lecture et écriture

Pour pouvoir **introduire** les données et **afficher** les résultats, la mémoire centrale doit être reliée avec le monde extérieur : le **clavier** et **l'écran**.

Les instructions permettant de gérer le transfert entre ces périphériques et la mémoire centrale sont :

- **lire** : ordonne le transfert d'information saisie par clavier vers la mémoire centrale.
- **écrire** : ordonne le transfert d'information depuis la mémoire centrale vers l'écran.

## Instruction de lecture et écriture

**lire(A)** : implique l'arrêt de l'exécution de l'algorithme. L'utilisateur doit saisir à l'aide du clavier une valeur du même type que A.

Cette valeur sera stockée dans l'espace mémoire désigné par A.

**écrire(A)** : affiche la valeur de A sur l'écran.

## Exemple :

algorithme double ;

var x, double : entier ;

début

    écrire(“donner une valeur entière : ”) ;

    lire(x) ;

    double  $\leftarrow$  x \* 2 ;

    écrire (“le double de x est : ”) ;

    écrire (double) ;

fin .

Le **déroulement** de cet algorithme est :

Instructions	Mémoire		Ecran
variables x, double : réel ;	x = ?	double = ?	
<u>écrire</u> (“donner une valeur réelle ”) ;	x = ?	double = ?	donner une valeur réelle
lire(x) ;	x = 5	double = ?	5 ↵
double $\leftarrow$ x * 2 ;	x = 5	double = 10	
<u>écrire</u> (“le double de ”, x, “ est : ”, double) ;	x = 5	double = 10	le double de 5 est : 10

# **Exemples programmes pratiques**

**(DEV C++)**

## Instructions conditionnelles :

L'instruction conditionnelle détermine si le bloc d'instructions (Début ... Fin) qui la suit sera exécuté ou non selon la valeur de sa condition.

Une condition est une expression booléenne.

- **Syntaxe :**

**Si** (condition) **alors**

Bloc d'instructions 1;

**sinon**

Bloc d'instructions 2 ;

**fin si**

Si la condition est vérifiée alors le bloc d'instructions 1 est exécuté, et si la condition n'est pas vérifiée c'est le bloc d'instructions 2 qui sera exécuté.

## Exemple 1 :

algorithme max\_deux\_entiers;

var x, y, max : entiers ;

début

lire(x,y) ;

si (x > y) alors

max ← x ;

sinon

max ← y ;

fin si

écrire(max) ;

Fin.

## Exemple 2 :

```
algorithme délibérations;  
var moyenne: réel ;  
    delib: chaine ;  
début  
    lire(moyenne) ;  
    si (moyenne >= 10) alors  
        delib ← 'ADMIS' ;  
    sinon  
        delib ← 'AJOURNE' ;  
    fin si  
    écrire (delib);  
Fin.
```

## •Conditionnelle imbriquée :

il est possible d'imbriquer des blocs de programme les uns dans les autres,

**Exemple**: affichage des mentions suivant leurs codes.

- 'A' : "Très Bien".
- 'B' : "Bien".
- 'C' : "Moyen".
- 'D' : "Insuffisant".

La **première solution** utilise des instructions conditionnelles simples :

```
algorithme mentions;
```

```
var x : char ;
```

```
début
```

```
    lire(x) ;
```

```
    si(x = 'A') alors écrire("Très Bien") ;
```

```
    si(x = 'B') alors écrire("Bien") ;
```

```
    si(x = 'C') alors écrire("Moyen") ;
```

```
    si(x = 'D') alors écrire("Insuffisant") ;
```

```
    si((x ≠ 'A') et (x ≠ 'B') et (x ≠ 'C') et (x ≠ 'D')) alors  
        écrire("valeur saisie incorrecte") ;
```

```
fin .
```

## une autre solution :

algorithme mentions;

var x : char ;

début

lire(x) ;

si(x = 'A') alors écrire("Très Bien") ;

sinon

    si(x = 'B') alors écrire("Bien") ;

    sinon

        si(x = 'C') alors écrire("Moyen") ;

        sinon

            si(x = 'D') alors écrire("Insuffisant") ;

            sinon

                écrire("valeur saisie incorrecte") ;

            fin si

        fin si

    fin si

fin si

fin

## Structure à choix multiples (selon) :

La structure de choix multiple permet d'effectuer des actions différentes suivant les valeurs que peut prendre une même variable.

**Selon** (variable ou expression)

**Cas** Valeur1 : action1 ;

**Cas** Valeur2 : action2 ;

**Cas** Valeur3 : action3 ;

**Cas Sinon** action par défaut ;

**Fin Selon**

## Exemple :

algorithme mentions ;

var x : char ;

début

lire(x) ;

selon (x)

cas 'A' : écrire("Très Bien") ;

cas 'B' : écrire("Bien") ;

cas 'C' : écrire("Moyen") ;

cas 'D' alors écrire("Insuffisant") ;

cas sinon écrire("valeur saisie incorrecte") ;

fin selon

Fin.

# LES BOUCLES

Le prochain cours !

