

### A) Calcul d'expressions : priorités des opérateurs

On distingue les expressions arithmétiques et les expressions booléennes (ou logique) :

- Une expression arithmétique est constituée d'opérandes numériques reliés par des opérateurs arithmétiques.
- Une expression booléenne (ou expression logique) est une expression dont le résultat est de type booléen. Elle peut comporter des opérateurs arithmétiques, des opérateurs de relation et des opérateurs booléens.

La priorité des opérateurs arithmétiques et logiques est dans l'ordre suivant :

- 1) Les parenthèses
- 2) Les fonctions
- 3) Le moins unaire, le NOT
- 4) \*, /, DIV, MOD, AND
- 5) +, -, OR
- 6) =, <>, <, >, <=, >=

#### Exercice 1 :

Evaluer les expressions suivantes en respectant l'ordre de priorité des opérateurs :

**Expression 1 :**  $a+b/c+((d/3+4)/3+a)/b$  ;

**Expression 2 :**  $(a>b)\text{or not}(c>=d)\text{and}(b<c)$  ; Avec  $a=1; b=2; c=4; d=6$

**Rép.:** Expression 1 = 3      Expression 2 = TRUE

### B) Instructions d'Entrées/Sorties (Lecture / Écriture) et instruction d'affectation

**Variables :** Les variables sont des objets contenant des valeurs pouvant être modifiées.

**Constantes :** Les constantes sont des objets contenant des valeurs non modifiables.

#### 1) Instruction d'entrée (Lecture)

Une instruction d'entrée nous permet dans un programme de donner une valeur quelconque à une variable.

#### 2) Instruction de sortie (écriture)

Une instruction de sortie nous permet dans un programme d'afficher un résultat.

#### 3) Instruction d'affectation

Une affectation consiste à donner une valeur (immédiate, constante, variable ou calculée à travers une expression) à une variable.

Algorithme	PASCAL	Signification
Lire(<id_var>)	read(<id_var>); readln(<id_var>);	Donner une valeur quelconque à la variable dont l'identifiant <id_var>.
Lire(<id1_var>, <id2_var>,	read(<id1_var>, <id2_var>,	Donner des valeurs aux variables <id1_var>, <id2_var>, etc.
Ecrire(<id_var>,<id_const>,<valeur>,<expression>)	write(<id_var>,<id_const>,<valeur>,<expression>); writeln(<id_var>,<id_const>,<valeur> ,<expression>;	Afficher une valeur d'une variable, d'une constante, valeur immédiate ou calculée à travers une expression.
<id_varialbe> ← <valeur>   <id_const>   <id_variable>   <expression>	<id_varialbe>:=<valeur>   <id_const>   <id_variable>   <expression>;	donner une valeur (immédiate, constante, variable ou calculée à travers une expression) à une variable.

**Exercice 2:**

A)

Dérouler l'algorithme suivant :

**Algorithme permutation****Variables A,B:entier****Début**

A←5

B←8

A←A+B

B←A-B

A←A-B

Écrire(A,B)

**Fin**

B) Soit l'algorithme suivant

**Algorithme** exo2-serie1**Variables** R : entier

Surface: réel

**Constante** P=3.14**Début**

Ecrire ('Introduire le rayon R ')

**Lire** (R)

Surface ← P\*sqr( R );

**Ecrire** ('La surface du cercle de rayon

R=',R,'est égale à', surface)

**Fin.****PROGRAM** exo2-serie1;**uses** wincrt;**Var** R : integer ;

Surface: real;

**Const** P=3.14;**BEGIN****Writeln**('Introduire le rayon R ');**read**(R);

Surface:=P\*sqr( R );

**write**('La surface du cercle de rayon R=',R,'est égale à', surface);**END.**3) Enregistrer le programme sous le nom **Exo\_1** dans le dossier **A1** (pour le groupe **A1**).

4) Compiler le programme et corriger les erreurs éventuelles.

5) Exécuter le programme.