Chapitre Troisième : Les Instructions

I. Introduction
II. Les instructions
II.1. L'instruction d'affectation
II.2. L'instruction de fin d'algorithme
II.3. L'instruction de commentaires
II.4. Les instructions d'Entrée/Sortie
II.4.1. Les instructions de Lecture (Entrées)
II.4.2. Les instructions d'Ecriture(Sorties)
II.5. Les structures de contrôles
II.5.1 Le branchement inconditionnel
II.5.2. Les structures répétitives (les boucles)
II.5.3. Les structures alternatives
Série Exercices

Chapitre Troisième

Les instructions

I. Introduction

Un algorithme est une séquence d'instructions.

Une instruction est une opération élémentaire (qui sera exécutée dans le cadre d'un programme). Elle peut être une opération d'affectation, une opération de lecture, une opération arithmétique, une opération d'écriture, ….

II. Les instructions

II.1. L'instruction d'affectation

Cette instruction permet d'affecter à une **variable**, une constante, une variable ou une expression.

Pour désigner une affectation, on utilise le symbole : ←

Cependant, il existe un autre symbole qui désigne l'affectation : **:=** (en Pascal)

Syntaxe :

Var ← Constante
Var ← Var1
Var ← Expression

Où :

Var, Var1 : des variables
Constante : une constante
Expression : expression non évaluée

Remarque : à gauche du symbole d'affectation←, on doit **toujours** utiliser une variable.

Exemples :

```
A ← 200 ;
B ← A ;
C ← (A+B)/2 ;
```

II.2. L'instruction de fin d'algorithme

Tout algorithme se termine par l'instruction **Fin** qui indique la fin de l'algorithme suivi d'un point.

Syntaxe :

Fin.

Exemple :

```
Algorithme exemple ;
      .....
Début
      .....
Fin.
```

II.3. L'instruction de commentaires

Un commentaire est une chaîne de caractères comprise entre les symboles (* et *). Un commentaire peut être inséré n'importe où dans un algorithme. Il sert à commenter l'algorithme. Le commentaire n'est pas exécutable, ainsi, sa présence (ou son absence) ne change pas la logique de l'algorithme.

Exemple:

```
Algorithme exemple ;(*Algorithme qui permet de calculer les racines d'une équation*)
  Var          DELTA : Réel ; (* le discriminant*)
               X1, X2  : Réel ; (* les racines de l'équation *)
     ……..
Début (*Partie Traitement*)

     …….

Fin. (*Fin de l'algorithme*)
```

Remarque : Il faut toujours s'assurer de la fin du commentaire.

II.4. Les instructions d'Entrée/Sortie

II.4.1. Les instructions de Lecture (Entrées)

Syntaxe :

LIRE (liste_de_variables) ;

Où

Liste_de_variables : désigne les variables à lire(valeurs à introduire)(séparées par des virgules)

Remarque :

L'instruction de lecture pourrait être accompagnée d'un message qui s'affichera à l'exécution.

LIRE ('Message',liste_de_variables) ;

Exemple :

```
LIRE(R,T) ;
…
…
LIRE ('Donner les valeurs de A,B et C : ', A, B, C) ;
```

A l'exécution la machine affiche :

```
76, 98
Donner les valeurs de A,B et C :
12, 54, 23
```

II.4.1.1. Lecture des éléments d'un tableau

La lecture d'un tableau consiste à remplir ses cases par des valeurs (saisies depuis l'extérieur (à partir du clavier) à l'aide de l'instruction **LIRE**).

La lecture des éléments d'un tableau se fait élément par élément. Elle se fait à l'aide de la boucle **Pour**

II.4.1.1.1. Lecture des éléments d'un tableau à une Dimension

Exemple :

```
Algorithme Lecture_Tableau_une_dimension ;
Var TAB(10)  : Entier ;
           I     : Entier ;
Début
     Pour i←1 Jusqu'à 10 Faire
          Lire TAB(I) ;
     Fpour ;
Fin.
```

II.4.1.1.2 Lecture des éléments d'un tableau à Deux Dimensions

Exemple :

```
Algorithme Lecture_Tableau_deux_dimensions ;
Var MAT(4,5)      : Entier ;
                I,J    : Entier ;
Début
     Pour i←1 Jusqu'à 4 Faire
          Pour j← 1 Jusqu'à 5 Faire
               Lire MAT(I,J) ;
          Fpourj ;
     Fpouri ;
Fin.
```

II.4.2. Les instructions d'Ecriture(Sorties)

Syntaxe :

ECRIRE (liste_de_variables)

Où

Liste_de_variables : désigne les variables à écrire

Remarque :

L'instruction d'écriture pourrait être accompagnée d'un message qui s'affichera à l'exécution.

ECRIRE ('Message'**,**liste_de_variables) **;**

Exemple :

```
.......
ECRIRE (A, B, C) ;
........
ECRIRE ('Le Résultat est : ',RES) ;
```

A l'exécution, la machine affiche :

12 54 23

Le Résultat est : 3432

Où

12 54 et 23 correspondent aux valeurs de A, B et C

3432 est la valeur de la variable RES

Exercices d'application

Exercice 01 :
Ecrire un algorithme qui permet de Lire 03 valeurs et de les afficher.

Exercice 02 :
Réécrire l'algorithme précédent en intégrant des messages (en Lecture et en Ecriture).

II.4.2.1. Ecriture des éléments d'un tableau

L'écriture des éléments d'un tableau se fait élément par élément. Elle se fait à l'aide de la boucle **Pour**

II.4.2.1.1 Ecriture des éléments d'un tableau à une Dimension

Exemple :

```
Algorithme Ecriture_Tableau_une_Dimension ;
Var TAB(10)  : Entier ;
           I     : Entier ;
Début
      ……..
      Pour i←1 Jusqu'à 10 Faire
            Ecrire TAB(I) ;
      Fpour ;
Fin.
```

Cette procédure permet d'afficher les 10 éléments du tableau TAB.

II.4.2.1.2 Ecriture des éléments d'un tableau à Deux Dimensions

Exemple :

```
Algorithme Ecriture_Tableau_deux_dimensions;
Var MAT(4,5)       : Entier ;
           I,J           : Entier ;
Début
      ……..
      Pour i←1 Jusqu'à 4 Faire
            Pour j← 1 Jusqu'à 5 Faire
                  Ecrire MAT(I,J) ;
            Fpourj ;
      Fpouri ;
Fin.
```

Exercices d'application

Exercice 01 :
Ecrire un algorithme qui permet de lire les éléments d'un tableau de 10 éléments de types entier et de les afficher.

Exercice 02 :
Ecrire un algorithme qui permet de lire un tableau de 10 éléments de type caractères et d'afficher seulement les 05 premières valeurs.

II.5. Les structures de contrôles

Cette famille d'instructions comporte essentiellement les branchements conditionnels et les branchements inconditionnels.

II.5.1 Le branchement inconditionnel : Aller à

Syntaxe :

Aller à Etiq ;

Où

Etiq : une étiquette dans l'algorithme

N.B : Les étiquettes doivent être déclarées par **Etiquette** dans la partie déclaration avant **Var**

Exemple :

```
Algorithme Exemple ;
Etiquettes : A1, B1
Var
     ..........
Début
     ........
Si X >= 10 Alors
                    Aller à A1 ;
Sinon
     RES ← (X + Y)^2/3
Fsi ;
.........
B1 : RES ← (X + B)/3
........
A1 : Ecrire ('Pas de solution') ;
.......
Fin.
```

II.5.2. Les structures répétitives (les boucles)

Ils existent essentiellement 03 structures répétitives.

II.5.2.1. La boucle POUR

Syntaxe1 :

Pour Var ← Val1 **Jusqu'à** Val2 **Pas** Val3 **Faire**
    Début
        Instruction1 ;
        Instruction2 ;
        Instruction3
    Fin ;
Fpour ;

Quand le nombre d'instructions contenu dans un bloc est supérieur à 1, les instructions doivent être écrites entre :
Début
    Bloc d'instructions ;
Fin ;

Où :

Var : variable d'incrémentation (compteur de passage)
Val1 : valeur de début (initiale)
Val2 : valeur terminale (fin)
Val3 : valeur du pas d'incrémentation (=1 par défaut)

Le programme exécute les instructions comprises entre **Début** et **Fin** autant de fois qu'il est spécifié par les valeurs **Val1** et **Val2**.

Exemple 1 :

```
Pour I ← 1 Jusqu'à 10 Faire
    Début
        Ecrire ('Bonne Année')
    Fin ;
Fpour ;
```

A l'exécution, on obtient :

```
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
Bonne Année
```

Exemple 2 :

```
Pour K←1 Jusqu'à 5 Faire
    Début
        A ← K^2 + 10
    Fin ;
Fpour ;
Ecrire (A) ;
```

A l'exécution on obtient :

35

qui est la valeur de la variable A.

Exemple 3 :

```
M ← 0 ;
Pour i←1 Jusqu'à 5 Faire
    Début
        M ← M + 100
    Fin ;
Fpour ;
Ecrire (M) ;
```

A l'exécution on obtient :

500

qui est le contenu de la variable M.

Exercices d'application

Exercice 01 :

Ecrire un algorithme qui permet de lire un nombre entier et d'afficher son double.

Exercice 02 :

Ecrire un algorithme qui permet de lire un nombre et d'afficher son double dans une série de 10 nombres entiers.

II.5.2.2. La boucle TANTQUE

Syntaxe :

```
Var ← Val1
TQ Var <= Val2 Faire
    Début
        Instruction1 ;
        Instruction2 ;
        Instruction3
        Var←Var + Pas
    Fin ;
FTQ ;
```

Où :

Var : variable d'incrémentation (compteur de passage) ;
Val1 : valeur de début (initiale) ;
Val2 : valeur terminale (fin) ;
Pas : valeur du pas d'incrémentation.

Le programme exécute les instructions comprises entre **Début** et **Fin** tant que la condition (Var <= Val2) est vérifiée.

Exemple 1 :

```
K ←1 ;
TQ K <= 5 Faire
    Début
        A ← K^2 + 10 ;
        K ← K+1
    Fin ;
FTQ ;
Ecrire (A) ;
```

A l'exécution on obtient :

35

qui est la valeur de la variable A.

Exercices d'application

Exercice 01 :

Réécrire l'algorithme précédent en utilisant la boucle **TantQue**.

II.5.2.3. La structure REPETER

Syntaxe :

```
Répéter
    Début
        Instruction1 ;
        Instruction2 ;
        Instruction3 ;
        ..
    Fin
Jusqu'à  Condition ;
```

Où

Instructions **:** sont les instructions à répéter
Condition **:** Expression conditionnelle (condition d'arrêt de la boucle **Répéter**)

Exemple :

```
Début
    Répéter
        Ecrire ('Saisir le nombre A ')
        Lire (A)
        Ecrire ('Saisir le nombre B ')
        Lire (B)
        P ←A * B
        Ecrire (P)
        Ecrire ('Encore un calcul ? Non touche N ; Oui autre touche')
        Lire (C)
    Jusqu'à C= 'N'
Fin
```

II.5.3. Les structures alternatives

Syntaxe1 :

```
Si Condition Alors
                Début
                    Instruction1 ;
                    Instruction2 ;
                Fin
Fsi ;
Instruction3 ;
Instruction4 ;
```

Où :

Condition : expression conditionnelle

- Si la condition **est vérifiée,** le programme exécute Instruction1**,** instruction2 puis Instruction3**,** instruction4.
- Si la condition **n'est pas vérifiée,** le programme exécute Instruction3 et instruction4.

Exemple :

```
A ←200 ;
B ← 100 ;
C ← 0 ;
D ← 0 ;
Si A > B Alors
                Début
                    C ← B - A ;
                    D← C^2 *10
                Fin
Fsi ;
X ← C + D ;
Ecrire (A,B) ;
Ecrire (C, D, X) ;
```

A l'exécution, on obtient :

```
200  100
-100 100000 99900
```

Syntaxe2 :

```
Si Condition Alors
                Début
                    Instruction1 ;
                    Instruction2 ;
                Fin
Sinon
    Début
        Instruction3 ;
        Instruction4 ;
    Fin ;
Fsi ;
Instruction5 ;
Instruction6 ;
```

Où :

Condition : expression conditionnelle

- Si la condition **est vérifiée,** le programme exécute Instruction1**,** instruction2 puis Instruction5**,** instruction6.
- Si la condition **n'est pas vérifiée,** le programme exécute Instruction3 et instruction4**,** puis Instruction5**,** instruction6.

Exemple :

```
A ←10 ;
B ← 20
Si A > B Alors
                Début
                    C←B - A ;
                    D← C^2 *10
                Fin
Sinon
     Début
          C←A - B ;
          D ← C *100
     Fin ;
Fsi ;
X ← C + D ;
Ecrire (C, D, X)
```

A l'exécution, on obtient :

```
????????
```

Série Exercices

Exercice 01 :

Soit la structure conditionnelle composée ci-dessous :

```
Si (Cond1) alors
        Si (Cond2) alors
                Instr1 ;
                Instr2
        Sinon
                Instr3 ;
                Instr4 ;
        Fsi ;
        Instr5
Sinon
        Si (Cond3) alors
                Instr6 ;
                Instr7
        Sinon
                Instr8 ;
        Fsi ;
Fsi ;
```

Quelles sont les instructions qui s'exécutent dans les cas suivants :

1) **Cond1** =Vrai **Cond2** =Vrai **Cond3** =Vrai
2) **Cond1** =Faux **Cond2** =Vrai **Cond3** =Vrai
3) **Cond1** =Faux **Cond2** =Faux **Cond3** =Faux

Exercice 02 :

Ecrire un Algorithme qui lit le potentiel hydrogène **Ph** d'une solution puis afficher sa nature selon les cas suivants :

Ph inférieur à 7 est **acide**
Ph supérieur à 7 est **basique**
Ph égal de 7 est **neutre**

***Exercice 03* :**

Ecrire un Algorithme qui affiche les nombres entiers de 1 à 50, par l'utilisation de la boucle :
Pour, puis ***Tant que***.

Exercice 04:

Ecrire un algorithme qui permet de lire un nombre et d'afficher son double pour les nombres positifs et la moitié pour les nombres négatifs dans une série de 10 nombres entiers.

Exercice 05 :

Ecrire un algorithme qui permet d'afficher seulement les nombres compris entre 9 et 12 d'un tableau de 10 éléments entiers.

Exercice 06 :

Ecrire un Algorithme qui calcule la somme suivante :

S=1+2+3+.....+100