Algorithmique

1 Squelette d'un algorithme

Comme il existe mille et une façons d'écrire un algorithme, notre choix s'est porté sur les algorithmes structurés en bloc.

Donc, pour nous un algorithme est sous la forme suivante :

```
Algorithme Nom
Déclaration
Constante
    ….          /*Déclaration des constante*/
    …
Variable
    …           /*Déclaration des variables*/
    …
Début
    …
    …           /*Instructions*/
    …
Fin.
```

C'est-à-dire qu'il y a trois zones : un entête, une zone déclarative et le corps de l'algorithme.

On va détailler chaque compartiment.

2 Déclaration

Une déclaration est considérée comme une réservation d'un emplacement physique en mémoire centrale.

Pour le moment, on va déclarer des constantes et des variables.

2_1 Constante

Une constante est déclarée par son nom (identificateur) égal une information (une donnée) suivie d'un point-virgule.

Exemple

A = 4 ;

Pi = 3.14 ;

Alpha = A*Pi ;

…

2_2 Variable

On commence par étudier seulement trois (03) types de variables : entier (entier relatif), réel et caractère.

Une variable est déclarée par son nom (identificateur) suivi de deux points ' :' puis son type et enfin un point-virgule.

Exemple

A : réel ;

B : réel ; ou bien A, B : réel ;

C : entier

F : caractère ;

3 Instruction ≡ ordre ≡ opération

Il y a les instructions de base (dites simples) et les primitives.

3_1 Instructions de base

Il en existe deux genres : Les entrées/sorties (I/O) et l'affectation.

3_1_1 Les input/output

Pour faire entrer une information, on utilisera le verbe « lire » et pour la faire sortir, c'est le verbe « écrire ».

Exemple 1 : Input

Lire(a) ;

Lire(X, Y) ;

…

Si on fait un parallèle avec la narration, on remarque l'utilisation des parenthèses et le point-virgule.

Question : Qu'est-ce qu'on lit ?

On ne lit que les variables et rien d'autre.

Dans cet exemple, on a utilisé trois variables a, X et Y, ils doivent être déclarés dans leur compartiment sous le vocable « variable ».

Exemple 2 : Output

Ecrire (F) ;

Ecrire (‘’ Le résultat est : ‘’, R) ;

Ecrire (X, Y, Z) ;

…

Question : Qu'est-ce qu'on écrit ?

Contrairement à la lecture, on peut écrire le contenu d'une variable, d'une constante ou encore un message.

Tout ce qui se trouve entre deux côtes est considéré comme un message.

3_1_2 Affectation

L'opération d'affectation consiste à donner une information à une variable préalablement déclarée.

C'est une opération simple mais ô combien elle est importante !

Exemple

D ← 2 ;

X ← -b/a ;

Co ← Co + 1 ;

Par rapport à la narration, le signe « = » a été remplacé par la flèche « ← » et bien sur le point-virgule.

3 2 Les primitives

Dans les narrations, on a utilisé un verbe très spécial qui est le verbe « tester », celui-ci sera remplacé par des primitives en algorithme.

On va se contenter d'étudier seulement quelques primitives :

- Primitives conditionnelles.
- Primitives alternatives.
- Primitives itératives.

4 Primitives conditionnelles

Les mots clés utilisés sont : Si, alors, fin si

4-1 1ier format d'écriture.

```
Si (Condition(s)) alors
    Instruction ;
Fin si
```

Si on a plus d'une instruction conditionnée entre le « si » et le « fin si », on utilisera un bloc (début fin) d'où la nécessité du 2ième format.

4-2 2ième format d'écriture.

```
Si (Condition(s) alors
  Début
      Instruction 1 ;
      Instruction 2 ;
      …
  Fin
Fin si
```

5 Primitives alternatives.

Les mots clés utilisés sont : si, alors, sinon, fin si

5-1 1ier format d'écriture

```
Si (Condition(s)) alors
    Instruction 1 ;
Sinon
        Instruction 2 ;
Fin si
```

L'instruction 1 et l'instruction 2 ne sont jamais exécutées en même temps ; c'est ou bien l'une ou bien l'autre d'où le mot « alternative ».

5-2 2ième format d'écriture.

```
Si (Condition(s)) alors
  Début
        Instruction 11 ;
        Instruction 12 ;
  Fin
Sinon
  Début
        Instruction 21 ;
        Instruction 22 ;
        Instruction 23 ;
  Fin
Fin si
```

6- Primitives itératives.

Ces primitives sont appelées aussi des boucles (loops). On va étudier deux genres de boucles : « tant que » et « pour ».

6-1 La boucle « tant que »

Les mots clés utilisés sont : tant que, faire, fin tant que

6-1-1 1ier format d'écriture.

```
Tant que (Condition(s)) faire
|       Instruction ;
Fin tant que
```

L'instruction qui est à l'intérieur de la boucle va s'exécuter plusieurs fois jusqu'à ce que la où les conditions ne sont plus vérifiées.

6_1_2 2ième format d'écriture

```
Tant que (condition(s)) faire
|       Début
|       |       Instruction 1 ;
|       |       Instruction 2 ;
|       |       …
|       Fin
Fin tant que
```

6_2 La boucle « Pour »

Les mots clés sont : pour, de, à, pas, faire, fin pour

6_2_1 1ier format d'écriture

```
Pour … de … à … pas … faire
|   Instruction ;
Fin pour
```

On constate qu'il y a quatre espace « … », cela veut dire, il y aura :

- Apres **pour**, le nom d'une variable déclarée.
- Après **de**, une information connue : c'est le point de départ ou l'initialisation de la variable.
- Après **à**, une autre information connue : c'est le point d'arrivée ou la condition de sortie de la boucle « pour ».
- Après **pas**, une troisième information connue : c'est le pas d'avancement (incrémentation) ou de recul (décrémentation).

Exemple 1

```
Pour X de 1 à 10 pas 1 faire
    Ecrire (X) ;
Fin pour
```

Dans cet exemple la variable X est initialisée à 1 et doit atteindre la valeur 10 avec un pas égal à 1.

Donc cette boucle va écrire les valeurs : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ; en quelque sorte elle va compter de 1 à 10 avec un pas égal à 1.

Ecrivons son équivalente en utilisant la boucle « tant que ».

```
X ← 1 ;          /* Initialisation*/

Tant que (X ≤ 10) faire
        Début
            Ecrire (X) ;
            X ← X + 1 ;      /* Incrémentation*/
        Fin
Fin tant que
```

NB : Si le pas est égal à 1, on peut ne pas le mentionner. Par conséquent la boucle de l'exemple précédent peut s'écrire comme suit :

```
Pour X de 1 à 10 faire
    Ecrire (X) ;
Fin pour
```

Exemple 2

```
Pour X de 10 à 1 pas -1 faire
    Ecrire (X) ;
Fin pour
```

C'est le compte à rebours de 10 à 1 avec un pas égal à -1. Son équivalent en utilisant la boucle « tant que » est :

```
X ← 10 ;        /* Initialisation*/

Tant que (X ≥ 1) faire
        Début
            Ecrire (X) ;
            X ← X – 1 ;     /* Décrémentation*/
        Fin
Fin tant que
```

Attention !

La variable de la boucle « pour » ne doit pas être modifiée à l'intérieur de la boucle : c'est strictement interdit.

Exemple

```
Pour A de 0 à 20 pas 2 faire
    A ← A + 1 ;          NON !
Fin Pour
```

6 2 2 2ième format d'écriture

```
Pour … de … à … pas … faire
|   Début
|   |   Instruction 1 ;
|   |   Instruction 2 ;
|   |   …
|   Fin
Fin pour
```

Remarque :

- Il est souhaitable de commenter votre algorithme. Un commentaire s'écrit entre /* et */, tout ce qui s'y trouve n'influe en aucun cas sur le déroulement de votre algorithme.

  Plus, un algorithme est commenté, plus il est compris.

- Les mots clés sont appelés aussi des mots réservés, c'est-à-dire ils appartiennent au vocabulaire du langage algorithmique. Pour les mettre en évidence, on les souligne.