Ecriture des algorithmes

Un *algorithme* est une suite d'actions à effectuer pour obtenir, à partir de données initiales, la solution d'un problème. Comme il existe souvent plusieurs manières de résoudre un problème, on peut imaginer plusieurs algorithmes plus ou moins différents

Les variables

Ces données ainsi que les résultats des calculs intermédiaires ou finaux, sont rangés dans des "cases-mémoires" appelées *variables* que l'on repère par des *identificateurs* (que l'on choisira autant que possible significatifs).
Les contenus des variables sont de nature diverse, évoluent pendant l'exécution des algorithmes, mais une variable ne peut contenir au cours du traitement que des données de même nature :

Le *type* d'une variable est l'ensemble des valeurs possibles de son contenu. On distingue :

Les types élémentaires :

- les types numériques : **ENTIER** et **REEL**.
- le type **BOOLEEN** (deux valeurs possibles : "vrai", "faux")
- le type **CHAÎNE** (ou chaîne de caractère)

Les types structurés :

- le type **TABLEAU** ou **MATRICE** (à une ou plusieurs dimensions)
- le type **ENREGISTREMENT** ou **LISTE** (ou type composé)

Dès le début du traitement, on indique (par exemple, dans un tableau), la liste des variables qui seront utilisées en précisant pour chacune d'elles *le nom*, *le type* ainsi que *le rôle* de cette variable dans l'algorithme.

Les instructions

Les instructions élémentaires

- **La lecture au clavier** du contenu d'une ou plusieurs variables :
  LIRE(variable) ; **LIRE**(A,B,C)
  Remarques : la lecture au clavier s'achève dès que l'on presse la touche "entrée" (ou "retour chariot"). La donnée tapée doit être du même type que la variable qui la reçoit.

- **L'affichage à l'écran** (ou **l'édition sur imprimante**) d'un objet (nombre, chaîne, ...) du contenu d'une ou plusieurs variables, d'une expression, ...
  ECRIRE('Prix de revient = ',P_Achat + Frais)

- **L'affectation** (donner une valeur au contenu d'une variable) :
  Nom de Variable ← Expression (la flèche ← peut se lire reçoit)
  ex : P_Vente ← P_Achat + Frais + Bénéfices

- **L'appel d'une procédure** (algorithme défini par ailleurs)

Les instructions composées

- **Un bloc** d'instructions est une suite d'instructions (élémentaires ou non) séparées par des points-virgules et encadrées des deux mots **DEBUT** et **FIN**. Dans la suite, "instruction" désignera soit une instruction élémentaire soit un bloc.

- **Les instructions conditionnelles :**

L'alternative : On effectue un test pour choisir entre deux instructions possibles :

SI <condition> **ALORS** instruction_1
SINON instruction_2;

La conditionnelle simple : même structure mais la deuxième instruction est vide :

SI <condition> **ALORS** instruction_1;

La conditionnelle multiple :

SELON NomVar
Cas_1 : Instruction_1;
Cas_2 : Instruction_2;
...
Cas_*n* : Instruction_*n*;
FIN;

- **Les instructions répétitives** (ou boucles): une même séquence est répétée un certain nombre de fois.

La boucle POUR : on connaît exactement le nombre de répétitions à effectuer. On utilise un compteur de boucles :

POUR i **VARIANT DE** a **A** b **EFFECTUER** Instruction;

La boucle TANT_QUE : tant que le test d'entrée est vrai, on exécute l'instruction.

TANT_QUE <condition> **EFFECTUER** instruction;

Exercices : Savoir exécuter un algorithme élémentaire

I. Dans cet algorithme *a, b, c* désignent des variables numériques

```
début
    a←5;
    b←12;
    c←2*a-b;
    b←2*b-c*3;
    a←b-a*4+c*5;
    écrire('A=',a,' B=',b,' C=',c);
fin.
```

1) Exécuter cet algorithme
2) Le résultat constaté sur *a* est-il vrai quelles que soient les valeurs initiales des variables *a* et *b* ?

II. Quelle est l'action effectuée par l'algorithme suivant ?

```
début
    lire(a,b);
    a←a+b;
    b←a-b;
    a←a-b;
    écrire('A=',a,' B=',b);
fin.
```

III.1) Effectuer l'algorithme suivant pour les triplets (*a,b,c*) :
a) (2,-1,3) b) (-1,6,0) c) (7,4,3)
2) Que réalise cet algorithme ?

```
début
    lire(a,b,c); {a,b,c sont des entiers}
    si a>b
        alors si a>c
                alors si b>c
                        alors écrire (a,' ',b,' ',c)
                        sinon écrire (a,' ',c,' ',b)
                sinon écrire(c,' ',a,' ',b)
        sinon si a>c
                alors écrire(b,' ',a,' ',c)
                sinon si b>c
                        alors écrire (b,' ',c,' ',a)
                        sinon écrire (c,' ',b,' ',a);
fin.
```

IV. Exécuter le programme suivant pour *n*=5 puis 10. Que réalise-t-il ?

```
Début
    lire(n); {n,p,i sont des entiers}
    p←1;
    Pour i ← 1 à n faire p←p*i;
    écrire('P=',p);
fin.
```