« RÈGLES » d'ÉCRITURE d'un algorithme en pseudo-code

1. SCHÉMA D'ÉCRITURE D'UN ALGORITHME

L'écriture d'un algorithme obéit à des règles précises et doit comporter :

- un **nom** (répondant aux mêmes contraintes que les noms de variables)
- la **déclaration des variables** utilisées (noms et types)
- un **début** et une **fin** encadrant la suite d'instructions constituant l'algorithme.
- des **commentaires** afin d'être facilement lisible et compréhensible. On signalera un commentaire par le symbole #. Tout ce qui suit ce symbole sur la même ligne est ignoré par la machine.

2. DÉCLARATION

nom : entier
nom : réel
nom : booléen
nom : Chaine
nom : tableau [Taille] : Type ou nom[] : tableau de Type
nom : tableau[Lignes][Colonnes] : Type

3. AFFECTATION D'UNE VARIABLE

Le fait d'attribuer un contenu à une variable s'appelle l'affectation Dans l'écriture d'un algorithme, elle est symbolisée par ← selon la syntaxe :

NomVariable ← Valeur

4. ENTRÉES-SORTIES

Afficher(« texte »)
Afficher (variable)
Afficher(« texte »+variable+ « texte »)

Saisir(variable) ou variable ← Saisir(« Invitation de saisie »)

5. TRAITEMENT CONDITIONNEL SIMPLE

Le traitement simple consiste à tester une condition et à réaliser une action (suite d'instructions) lorsqu'elle est vraie Sa structure est la suivante :

Si (condition) **Alors :**

action si condition vraie

Fin Si

6. TRAITEMENT CONDITIONNEL ÉTENDU

Le principe reste identique au traitement simple, mais on est dirigé vers une autre suite d'instructions lorsque la condition n'est pas réalisée La structure est la suivante :

Si (condition) **Alors :**
action si condition vraie
Sinon
action si condition fausse
Fin Si

7. IMBRICATION

Si (Condition1) **Alors :**
    action si Condition1 vraie
Sinon
    Si (Condition2) **Alors :**
        action si Condition2 vraie
    Sinon
        action si condition2 fausse
    Fin Si
Fin Si

8. LA BOUCLE "TANT QUE"

La structure est la suivante :

Tant que (condition) **Faire :**
    action
Fin Tant que

9. LA BOUCLE "POUR"

Il s'agit d'une boucle "comptée" dans laquelle le programme s'occupe de tout : initialisation de la variable de comptage, son évolution et la condition d'arrêt Sa structure est la suivante :

Pour VariableComptage **de** ValeurDébut **à** ValeurFin **Faire :**
    action ….
FinPour

10. PROCÉDURES ET FONCTIONS

PROCÉDURE nom(arguments : type)
    Instructions
FIN PROCÉDURE

FONCTION nom (arguments : type)
    renvoyer …….
FIN FONCTION

11. OPÉRATIONS ET FONCTIONS UTILES

x mod y ou x modulo y	donne le reste de la division euclidienne de x par y
aléa()	qui renvoie un nombre réel aléatoire de l'intervalle [0; 1[
arrondi(x,n)	qui arrondit un réel x à un certain nombre n de décimales
long(ch) ou long(tableau)	nombre de caractères de la chaine ch ou taille d'un tableau
majus(ch) , minus(ch)	mise en majuscule ou minuscule de ch
position(ch_1,ch_2,n)	renvoie la position de la chaîne ch_1 dans la chaîne ch_2 à partir de la position n renvoie −1 si la chaîne ch_1 n'est pas trouvée
extrait(ch,n,p)	renvoie la sous-chaine de p caractères extraite de ch à partir du caractère de rang n

carASCII(n)	renvoie le caractère dont le code ASCII est n
codeASCII(caract)	renvoie le code ASCII du caractère caract
entier(ch)	convertit la chaine ch en un nombre entier
réel(ch)	convertit la chaine ch en un nombre réel
chaine(x)	convertit le nombre x (entier ou réel) en chaine
étendre(NomTableau,Valeur)	ajoute une cellule au tableau NomTableau avec valeur comme contenu