

**République Algérienne Démocratique et Populaire**  
**Université des Sciences et de la Technologie Houari Boumediene**  
**Faculté d'Electronique et d'Informatique**  
**Département Informatique**



# Langage Algorithmique (4)

**Cours Algorithmique de 1ere Année MI**  
**Cours: Mr Bessaa Présenté par : S.Medjadba**

## Exemple

## Suite 3...

Ecrire un algorithme qui affiche le nom du mois en ayant son numéro.

Algorithme NomMois;

Var N:entier;

Debut

Ecrire('Donner le numéro du mois:');

Lire(N);

Si N=1 Alors Ecrire('Janvier'); Fsi;

Si N=2 Alors Ecrire('Fevrier'); Fsi;

Si N=3 Alors Ecrire('Mars'); Fsi;

Si N=4 Alors Ecrire('Avril'); Fsi;

Si N=5 Alors Ecrire('Mai'); Fsi;

Si N=6 Alors Ecrire('Juin'); Fsi;

Si N=7 Alors Ecrire('Juillet'); Fsi;

Si N=8 Alors Ecrire('Aôut'); Fsi;

Si N=9 Alors Ecrire('Septembre'); Fsi;

Si N=10 Alors Ecrire('Octobre'); Fsi;

Si N=11 Alors Ecrire('Novembre'); Fsi;

Si N=12 Alors Ecrire('Décembre'); Sinon Ecrire('Erreur') Fsi;

Fin.

On a fait combien de tests ?

12

## Exemple

Ecrire un algorithme qui affiche le nom du mois en ayant son numéro.

Algorithme NomMois;

Var N:entier;

Debut

Ecrire('Donner le numéro du mois:');

Lire(N);

Si N=1 Alors Ecrire('Janvier');

Sinon Si N=2 Alors Ecrire('Fevrier');

Sinon Si N=3 Alors Ecrire('Mars');

Sinon Si N=4 Alors Ecrire('Avril');

Sinon Si N=5 Alors Ecrire('Mai');

Sinon Si N=6 Alors Ecrire('Juin');

Sinon Si N=7 Alors Ecrire('Juillet');

Sinon Si N=8 Alors Ecrire('Aôut');

Sinon Si N=9 Alors Ecrire('Septembre');

Sinon Si N=10 Alors Ecrire('Octobre');

Sinon Si N=11 Alors Ecrire('Novembre');

Sinon Si N=12 Alors Ecrire('Décembre');

Sinon Ecrire('Erreur')

Fsi;

Fsi; Fsi; Fsi; Fsi; Fsi; Fsi; Fsi; Fsi; Fsi; Fsi; Fsi;

Fin.

**Une meilleure solution  
utilisant l'alternative complète  
Si ... Alors ... Sinon ... Fsi;  
et Imbrication des tests**

**On a fait combien de tests ?**

**Entre 1 et 12**

**Il y a moins de tests, mais trop d'écriture !!!**

**Solution ?**

## Structure Alternative à choix multiple : Cas ... Vaut

Une action alternative à choix multiple permet d'exécuter un bloc d'actions **parmi plusieurs** suivant la **valeur** produite par une **expression** qui joue le rôle d'un **sélecteur**.

### Sa syntaxe



**Cas** <Expression> **Vaut**

**Val1** : <Bloc Action1> ;

**Val2** : <Bloc Action2> ;

---

**ValN** : <Bloc ActionN> ;

**Sinon** <Bloc Autres> ;

**Fincas;**

C'est une instruction structurée **équivalente** à une série de tests **Si...Alors...Sinon** imbriqués.

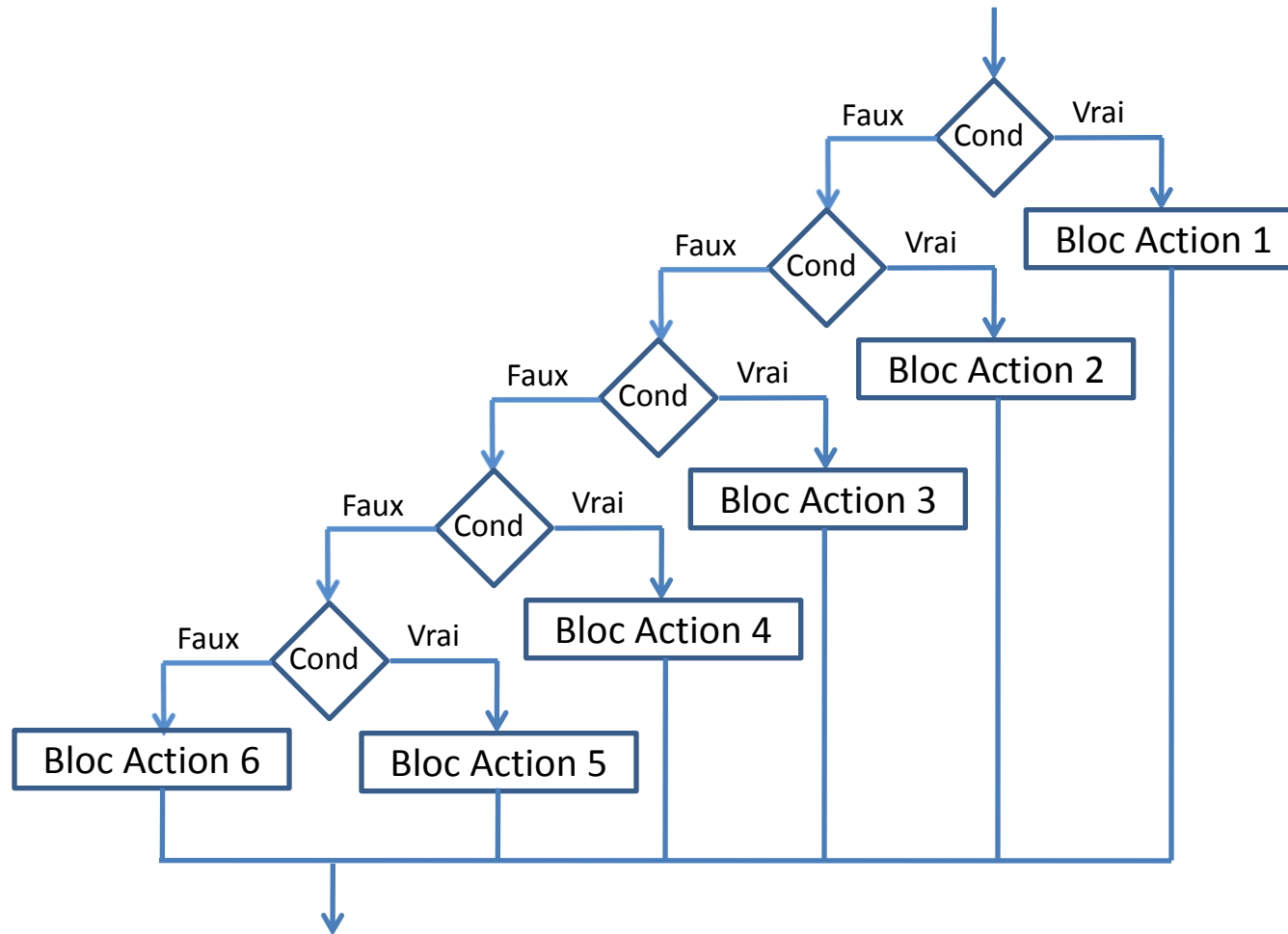
<Expression> : est une variable de type : **Entier**, **Booleen** ou **Caractère**

<Bloc Action*i*> : suite d'actions séparées par des ;

Val **i** : les différentes valeurs possibles de <Expression>

**Remarque** : l'action **Sinon** <Bloc Autres> est **facultative**.

## Organigramme équivalent



## Remarques

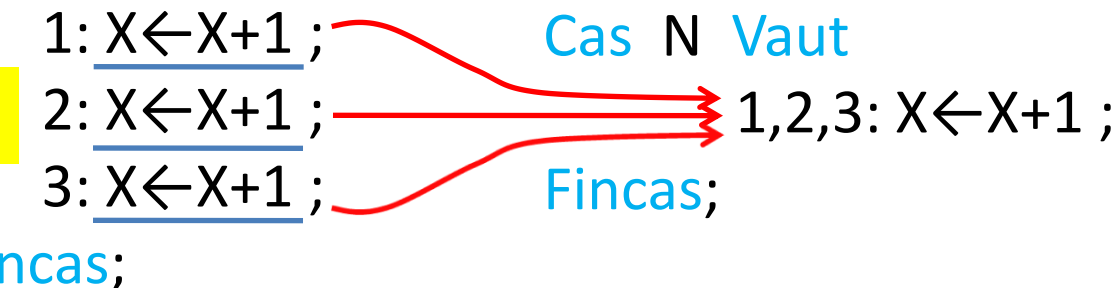
Si plusieurs Valeurs déclenche le même bloc d'actions, on peut les regrouper (séparées avec des virgules).

Exemple : Cas N Vaut

Même Action

1:  $X \leftarrow X+1$  ;  
2:  $X \leftarrow X+1$  ;  
3:  $X \leftarrow X+1$  ;  
Fincas;

Cas N Vaut  
1,2,3:  $X \leftarrow X+1$  ;  
Fincas;



Si ces valeurs sont consécutives (forment un **intervalle**), on peut les écrire sous forme : Val1 .. ValN : <Bloc Action>;

Exemple : Cas N Vaut

1,2,3,4,5,6:  $X \leftarrow X+1$  ;  
Fincas;

Cas N Vaut

1..6:  $X \leftarrow X+1$  ;  
Fincas;

Les différentes valeurs possible ne peuvent être que des **constantes**

Et si on revient à l'exemple précédent (affichage des mois), on peut présenter la solution avec **Cas ... Vaut**

Algorithme NomMois;

Var N:entier;

Debut

Ecrire('Donner le numéro du mois:');

Lire(N);

Cas N Vaut

1 : Ecrire('Janvier');

2 : Ecrire('Fevrier');

3 : Ecrire('Mars');

4 : Ecrire('Avril');

5 : Ecrire('Mai');

6 : Ecrire('Juin');

7 : Ecrire('Juillet');

8 : Ecrire('Août');

9 : Ecrire('Septembre');

10 : Ecrire('Octobre');

11 : Ecrire('Novembre');

12 : Ecrire('Décembre');

Sinon Ecrire('Erreur');

Fincas;

Fin.

**Avantage**

**Ecriture simple avec le minimum  
de tests.**

**Entre 1 et 12**

## Exemple

Ecrire un algorithme qui affiche le nom de la saison en ayant le numéro d'un mois.

Algorithme NomMois;

Var N:entier;

Debut

    Ecrire('Donner le numéro du mois:');

    Lire(N);

    Cas N Vaut

        1,2,12 : Ecrire('Hiver');

        3,4,5 : Ecrire('Printemps');

        6,7,8 : Ecrire('Eté');

        9,10,11 : Ecrire('Automne');

    Sinon Ecrire('Erreur');

    Fincas;

Fin.

3..5 : Ecrire('Printemps');

6..8 : Ecrire('Eté');

9..11 : Ecrire('Automne');



## Quelque chose de nouveau ?!

Ecrire un algorithme qui calcule puis affiche la somme de 5 entiers.

Algorithme Somme5;

Var A1,A2,A3,A4,A5,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

Lire(A1);

Lire(A2);

Lire(A3);

Lire(A4);

Lire(A5);

$S \leftarrow A1 + A2 + A3 + A4 + A5;$

Ecrire('La somme est : ',S);

Fin.

A1	2
A2	5
A3	8
A4	3
A5	7
S	25

## Exécution :

Soit à calculer la somme  $S = 2+5+8+3+7$  ?

La somme est : 25

Mais il n'y a rien de nouveau ?!

Et ben alors tant mieux ?!

**Ya pas mieux que ça ???**

**Mieux que ça, on peut utiliser moins de variables !**

```
Algorithme Somme5;  
Var A1,A2,A3,A4,A5,S:entier;  
Debut  
    Ecrire('Donner les 5 entiers:');  
    Lire(A1);  
    Lire(A2);  
    Lire(A3);  
    Lire(A4);  
    Lire(A5);  
    S ← A1 + A2 + A3 + A4 + A5;  
    Ecrire('La somme est : ',S);  
Fin.
```

**Voilà une solution meilleure.**

On utilise 2 au lieu de 6 variables

```
Algorithme Somme5;  
Var A,S:entier;  
Debut  
    Ecrire('Donner les 5 entiers:');  
    S ← 0;  
    Lire(A); S ← S + A;  
    Lire(A); S ← S + A;  
    Lire(A); S ← S + A;  
    Lire(A); S ← S + A;  
    Lire(A); S ← S + A;  
    Ecrire('La somme est : ',S);  
Fin.
```

**Et comment ça va marcher ?**

**Allez, on va voir...**

**N'oubliez pas,** ont veut calculer la somme  $S = 2+5+8+3+7$  ?

Algorithme Somme5;

Var A,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

$S \leftarrow 0$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Ecrire('La somme est : ',S);

Fin.

A **2**      S **0**

**N'oubliez pas,** ont veut calculer la somme  $S = 2+5+8+3+7$  ?

Algorithme Somme5;

Var A,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

$S \leftarrow 0$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Ecrire('La somme est : ',S);

Fin.

A	8	S	15
---	---	---	----

**N'oubliez pas,** ont veut calculer la somme  $S = 2+5+8+3+7$  ?

Algorithme Somme5;

Var A,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

$S \leftarrow 0$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Ecrire('La somme est : ',S);

Fin.

A	8	S	18
---	---	---	----

**N'oubliez pas,** ont veut calculer la somme  $S = 2+5+8+3+7$  ?

Algorithme Somme5;

Var A,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

$S \leftarrow 0$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Lire(A);  $S \leftarrow S + A$ ;

Ecrire('La somme est : ',S);

Fin.



A	<b>3</b>	S	<b>28</b>
---	----------	---	-----------

La somme est : 25

**Bien,** c'est intéressant.

**Alors** écrivez moi un algorithme qui calcule la somme de **500 entiers**

**Ahhh Noooooon, pitié...**

**On va écrire 500 lignes !? Mais c'est une punition...**

**Voilà le NOUVEAU**

**Non, non c'est pas le moment de lâcher.**



Quand la vie te  
donne une centaine  
de raisons de  
pleurer, montre-lui  
que tu as mille  
raisons de sourire

**Réfléchissons ?!**

## Revenons à la deuxième solution

Algorithme Somme5;

Var A,S:entier;

Debut

Ecrire('Donner les 5 entiers:');

S ← 0;

Lire(A); S ← S + A;

Lire(A); S ← S + A;

Lire(A); S ← S + A;

Lire(A); S ← S + A;

Lire(A); S ← S + A;

Ecrire('La somme est : ',S);

Fin.

### Pour la somme de 5 entiers

On a **répété** 5 fois la même chose.

### Donc

Si **nous** voulons calculer la somme de 500 entiers, il faudra **répéter** les actions **Lire(A); S ← S + A;** **500** fois ?

**Mais** pourquoi **nous** ?

Pourquoi ne pas **demande**r à la machine de **le faire** ? **ELLE**, elle peut le **répéter** plus que **500** fois sans **se fatiguer**.

**Et comment ?**

**C'est simple**, il faut juste lui apprendre à le faire !



## Les Structures Répétitives ou Itératives (Boucles)

La structure répétitive permet d'exécuter **plusieurs** fois une suite d'actions (Bloc Action). On distingue trois structures itératives.

### 1- La boucle Pour ...

Elle permet d'exécuter un Bloc Action un **nombre N** de fois **connu** à l'avance ( $N \geq 0$ ) en utilisant un **compteur**.

#### Sa syntaxe

**Pour** <IdCpt> **←** <Val\_Init> **à** <Val\_Fin> **Faire** <Bloc Action> **Fait;**

#### Ou

**Pour** <IdCpt> **De** <Val\_Init> **à** <Val\_Fin> **Faire** <Bloc Action> **Fait;**

<IdCpt>: identificateur (nom) du compteur.

<Val\_Init> : valeur initiale du compteur  
<Val\_Fin> : valeur finale du compteur

(entier, caractère)

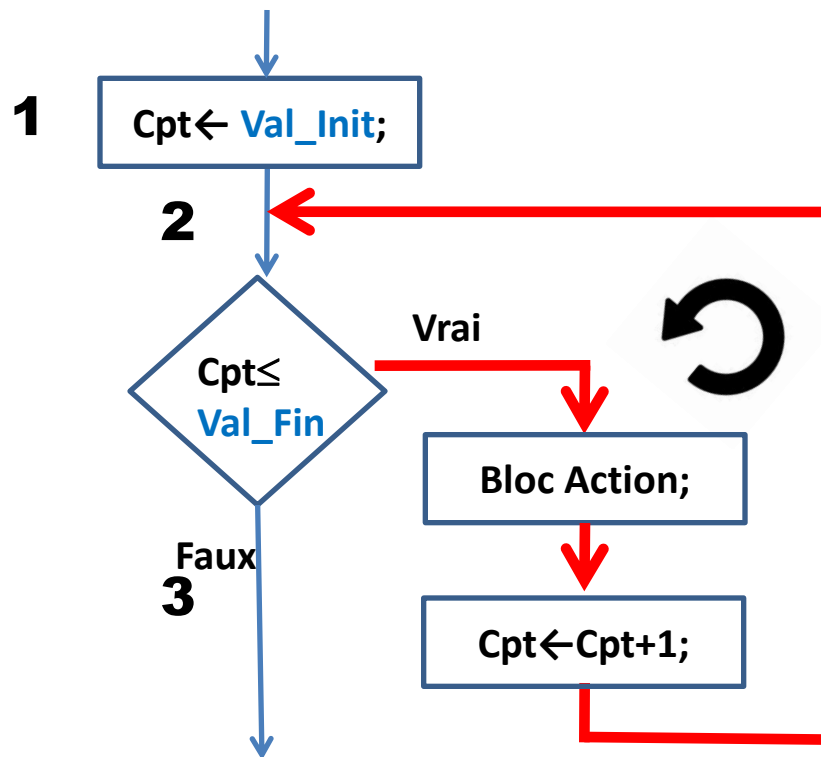
<Bloc Action> : suite d'actions séparées par des ;

## Sémantique

- 1- Initialiser le compteur à la valeur **initiale** **Val\_Init**
- 2- Si Compteur  $\leq$  à la valeur **finale** **Val\_Fin**
  - Exécute <Bloc Action> ;
  - Incrémenter le compteur (**ajouter 1** au compteur);
  - Aller à 2;
- 3- Si Compteur  $>$  Val\_Fin, on continue après **la boucle** ;

### Remarque

Si  $\text{Val\_Init} > \text{Val\_Fin}$ , la boucle ne s'exécute pas, et donc on ne fait rien



Boucle

**Maintenant**, on peut écrire notre fameux algorithme qui fait la somme de 500 entiers.

**Algorithme** Somme500;

**Var** A,S,I:entier;

**Debut**

**Ecrire**('Donner les **500** entiers:');

$S \leftarrow 0$ ;

**Pour** I  $\leftarrow$  1 à 500

**Faire**

**Lire**(A);

$S \leftarrow S + A$ ;

**Fait**;

**Ecrire**('La somme est : ',S);

**Fin**.

On ajoute un compteur I aux objets utilisés.

La valeur de **Val\_Init** = 1

La valeur de **Val\_Fin** = 500

**Et voilà**, les **500 lignes** qui vous ont fait **peur**, sont devenues **5 lignes**.

On peut encore **généraliser** la solution pour calculer la somme de **N** entiers. Il suffit de donner sa valeur.

**Dans ce cas, il faut ajouter encore une autre variable, le N**

**Algorithme** SommeN;

**Var** A,S,I,N:entier;

**Debut**

**Ecrire**("Donner le nombre d'éléments N ");

**Lire**(N);

$S \leftarrow 0$ ;

**Pour** I  $\leftarrow$  1 à N

**Faire**

**Lire**(A);

$S \leftarrow S + A$ ;

**Fait**;

**Ecrire**('La somme est : ',S);

**Fin.**

## **Un autre exemple**

Ecrire un algorithme qui calcule le **produit** de deux entiers positifs en utilisant uniquement l'opération d'**addition** (+).

Soient **A** et **B** ces deux entiers. On veut calculer le produit **P**.

La solution consiste à transformer le **produit** (\*) en **addition** (+).

$$\text{On a : } P = A * B = \underbrace{A + A + A + \dots + A}_{\text{B fois}}$$

**B fois**

**Donc on va boucler sur l'addition de A.** Mais si A est nul, on n'a pas besoin de boucler, le résultat sera nul.

**Algorithme** ProdSomme;

**Var** A,B,P,I:entier;

**Debut**

**Ecrire**("Donner deux entiers positifs A et B");

**Lire**(A,B);

$P \leftarrow 0$ ;

**Si**  $A \neq 0$  **Alors** **Pour**  $I \leftarrow 1$  **à** B

**Faire**  $P \leftarrow P + A$ ; **Fait**;

**Fsi**;

**Ecrire**('Le produit  $A*B =$ ',P);

**Fin.**

## Encore un autre exemple

Ecrire un algorithme qui calcule la somme des N premiers entiers:

$$S = 1 + 2 + 3 + 4 + \dots + N$$

**Alors**, on voit qu'on boucle sur l'addition (+), mais le éléments à additionner coïncident avec le compteur.

Algorithme SommeN;

Var S,I,N:entier;

Debut

Ecrire("Donner le nombre N");

Lire(N);

S ← 0;

Pour I ← 1 à N

Faire

S ← S + I;

Fait;

Ecrire('La somme S= ',S);

Fin.

## Déroulement pour N=6

$$S = 1 + 2 + 3 + 4 + 5 + 6$$

Etape	N	I	S
Initiale	6		0
1	6	1	1
2	6	2	3
3	6	3	6
4	6	4	10
5	6	5	15
6	6	6	21

# Un problème un peu différent !

Soit un dé à 6 faces, chacune porte un numéro de 1 à 6

On va jouer un jeu: chaque joueur lance **plusieurs fois** le dé d'une manière aléatoire, et **ne s'arrête** que s'il **obtient** la face de **numéro 1**. La note du joueur est la **somme** des nombres des faces obtenues.



Ecrire un algorithme permettant de **calculer la note** d'un joueur en ayant les différentes faces obtenues.

Dans ce jeu (problème), le dé est lancé **plusieurs fois**, donc on a un phénomène qui **se répète** : **BOUCLE**

**C'est simple**, on va lire les résultats obtenus, et à chaque étape on ajoute, comme l'exemple précédent !!!

**Ok**, mais le compteur de la boucle il va de quelle à quelle valeur ?  
Quelles sont les valeurs **Val\_Init** et **Val\_Fin** ?

**Mais, on ne peut pas le savoir, c'est aléatoire !**

**Donc**, on ne peut pas utiliser la boucle **Pour**.

**Et alors la solution ????**

**La solution**, vous vous souvenez, on a dit qu'on a **trois** structures répétitives.

## **2- La boucle Tantque ...**

Elle permet d'exécuter **plusieurs fois** un Bloc Action suivant la valeur **d'une condition**. Le nombre de fois n'est **pas forcément** connu.

### **Sa syntaxe**

**Tantque** <Condition> **Faire** <Bloc Action> **Fait;**

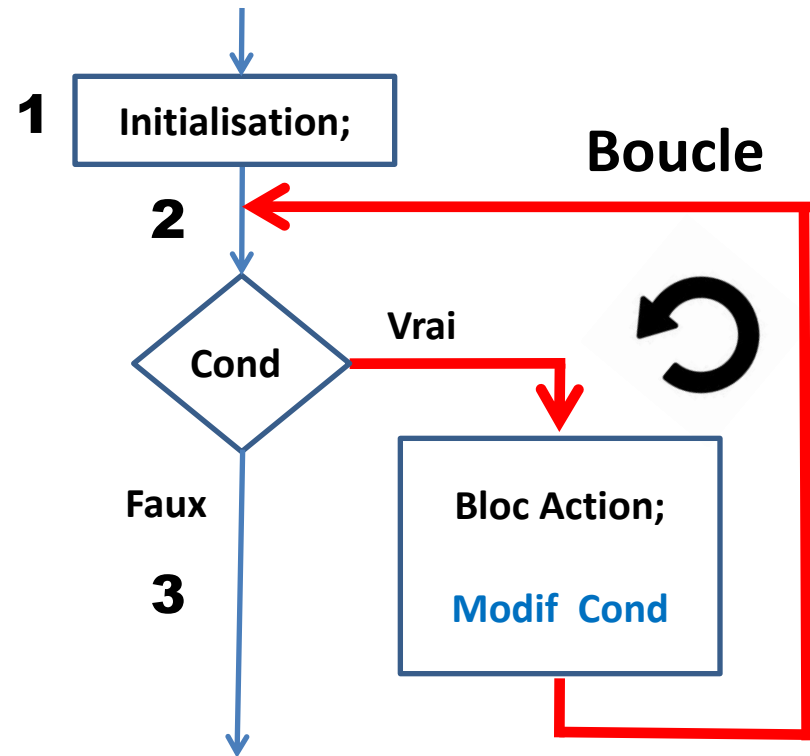
<Condition>: est une expression logique.

<Bloc Action> : suite d'actions séparées par des ;



## Sémantique

- 1- Initialiser la condition
- 2- Si la condition vaut **Vrai**
  - Exécute <Bloc Action> ;
  - **Modifier la condition**
  - Aller à 2;
- 3- Si la condition vaut **Faux**,  
on continue après **la boucle** ;



## Comment modifier la condition ?

La condition est une **expression logique**, donc elle contient des identificateurs d'objets(variables). Sa modification consiste à **modifier au moins un** de ces objets.

Donc le **bloc action** doit **contenir** une action qui **change** la valeur d'au moins un objet de la condition(par **affectation** ou **lecture**).

## Attention

La modification de la condition n'est pas réalisée automatiquement, elle doit être faite par l'utilisateur.

## Exemple

Revenons au problème du lancer de dé. La condition d'arrêt du jeu était **d'avoir une face égale à 1**. La valeur de la face obtenue à chaque itération est introduite par une **lecture**, ce qui **modifie** la condition.

La **donnée en entrée** de ce problème est la valeur de la **face**.

Le **résultat** est la **note** du joueur.

La note est **calculée par un cumul répétitif**. L'opération **s'arrête** une fois la face obtenue = **1**. Au début la Note est initialisée à **la valeur de la première face obtenu**.

Soit **Face** et **Note** les deux objets manipulés.

L'algorithme est le suivant:

Algorithme JeuDe;

Var Face,Note:entier;

Debut

Ecrire("Donner les valeurs des faces obtenues ");

Lire(Face);

Note ← Face;

Tantque Face <> 1

Faire



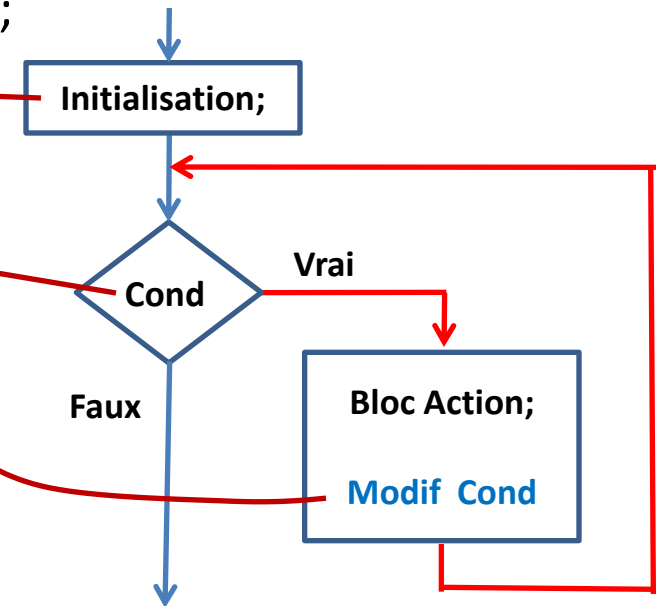
Lire(Face);

Note ← Note + Face;

Fait;

Ecrire('La note du joueur est : ',Note);

Fin.



Algorithme JeuDe;

Var Face, Note: entier;

Debut

Ecrire("Donner les valeurs des faces obtenues ");

Lire(Face);

Note ← Face;

Tantque Face ≠ 1

3 ≠ 1

Faire

Lire(Face);

Note ← Note + Face;

Fait;

Ecrire('La note du joueur est : ', Note);

Fin.

Face

2

Note

3

## Exécution

Supposons que le joueur a obtenu les faces suivantes: **3-2-3-4-1**

```

Algorithme JeuDe;
Var Face,Note:entier;
Debut
  Ecrire("Donner les valeurs des faces obtenues ");
  Lire(Face);
  Note←Face;
  Tantque Face≠1  2 ≠ 1
  Faire
    Lire(Face);
    Note←Note + Face;
  Fait;
  Ecrire('La note du joueur est : ',Note);
Fin.

```

Face **2**      Note **8**

## Exécution

Supposons que le joueur à obtenu les faces suivantes: 3-2-3-4-1

```

Algorithme JeuDe;
Var Face,Note:entier;
Debut
  Ecrire("Donner les valeurs des faces obtenues ");
  Lire(Face);
  Note←Face;
  Tantque Face≠1  3 ≠ 1
  Faire
    Lire(Face);
    Note←Note + Face;
  Fait;
  Ecrire('La note du joueur est : ',Note);
Fin.

```

Face **2**      Note **12**

## Exécution

Supposons que le joueur à obtenu les faces suivantes: 3-2-3-4-1

```

Algorithme JeuDe;
Var Face,Note:entier;
Debut
  Ecrire("Donner les valeurs des faces obtenues ");
  Lire(Face);
  Note←Face;
  Tantque Face≠1  4 ≠ 1
  Faire
    Lire(Face);
    Note←Note + Face;
  Fait;
  Ecrire('La note du joueur est : ',Note);
Fin.

```

Face **4**      Note **12**

## Exécution

Supposons que le joueur à obtenu les faces suivantes: 3-2-3-4-1

```
Algorithme JeuDe;  
Var Face,Note:entier;  
Debut  
  Ecrire("Donner les valeurs des faces obtenues ");  
  Lire(Face);  
  Note←Face;  
  Tantque Face≠1  
    Faire  
      Lire(Face);  
      Note←Note + Face;  
  Fait;  
  Ecrire('La note du joueur est : ',Note);  
Fin.
```

Face **1**      Note **13**

**La note du joueur est : 13**

## Exécution

Supposons que le joueur à obtenu les faces suivantes: 3-2-3-4-1

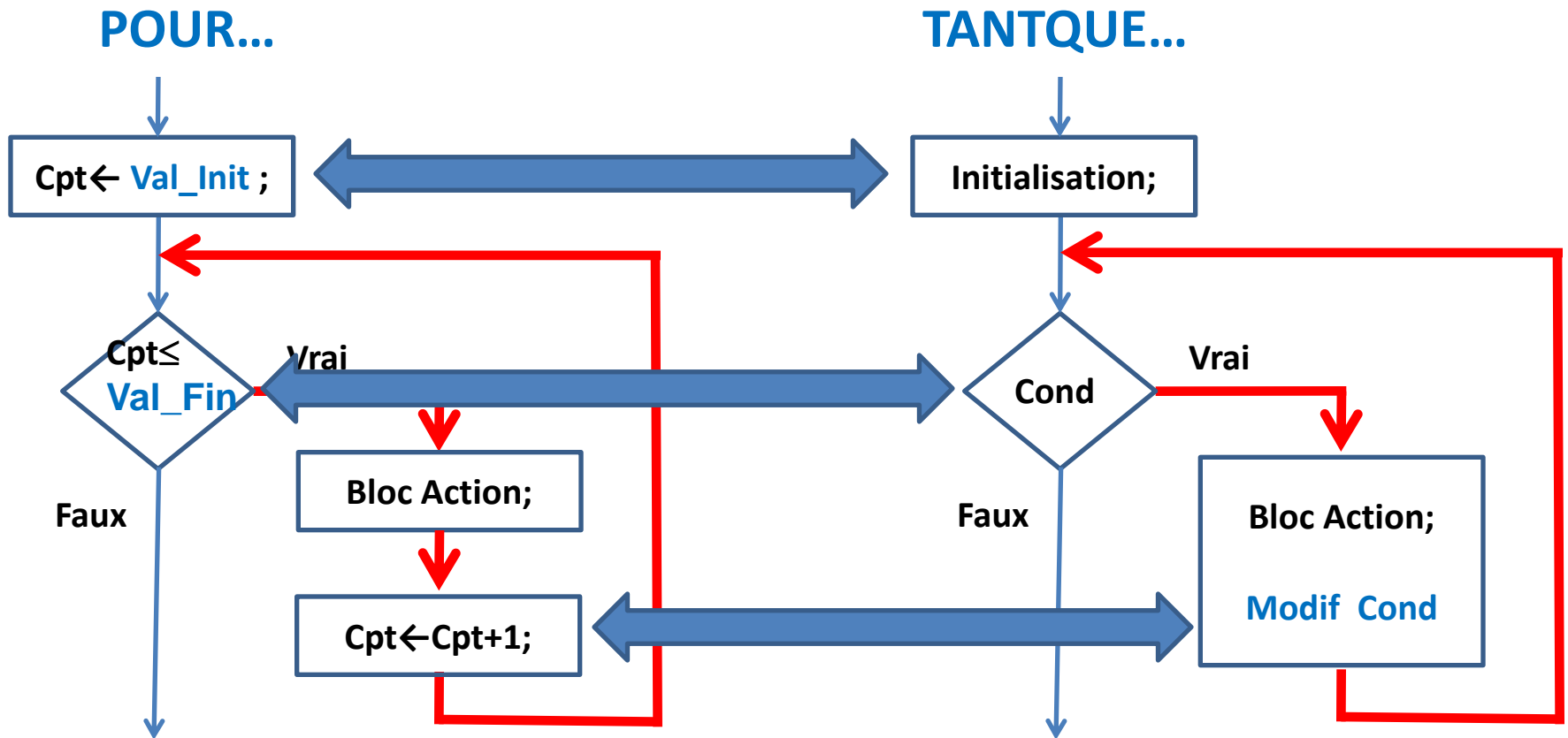


## Retour sur la définition de la boucle Tantque ...

Elle permet d'exécuter **plusieurs fois** un Bloc Action suivant la valeur d'une **condition**. Le nombre de fois n'est pas forcément connu.

**Ca veut dire qu'on peut utiliser la boucle Tantque si le nombre est connu ?**

**Effectivement**, regardons les organigrammes des deux boucles:



**Donc**, toute boucle **POUR** peut être transformé en boucle **TANTQUE**

Mais **ATTENTION** l'inverse n'est pas toujours possible

Il est possible si le nombre d'itérations est connu

**Exemple**, reprenons l'exemple de la somme de N entiers.

## Solution avec POUR

Algorithme SommeN;

Var A,S,I,N:entier;

Debut

Ecrire("Donner le nombre d'éléments N ");

Lire(N);

S ← 0;

Pour I ← 1 à N

Faire

Lire(A); S ← S + A;

Fait; I ← I+1 automatique

Ecrire('La somme est : ',S);

Fin.

## Solution avec TANTQUE

Algorithme SommeN;

Var A,S,I,N:entier;

Debut

Ecrire("Donner le nombre d'éléments N ");

Lire(N);

S ← 0;

I ← 1; initialisation

Tantque I ≤ N

Faire

Lire(A); S ← S + A;

I ← I+1; Faite par le programmeur

Fait;

Ecrire('La somme est : ',S);

Fin.

**Exemple de la somme** des N premiers entiers:

$$S = 1 + 2 + 3 + 4 + \dots + N$$

## Solution avec POUR

```
Algorithme SommeN;  
Var S,I,N:entier;  
Debut  
    Ecrire("Donner le nombre N");  
    Lire(N);  
    S ← 0;  
    Pour I ← 1 à N  
        Faire  
            S ← S + I;  
        Fait;  
    Ecrire('La somme S= ',S);  
Fin.
```

## Solution avec TANTQUE

```
Algorithme SommeN;  
Var S,I,N:entier;  
Debut  
    Ecrire("Donner le nombre N");  
    Lire(N);  
    S ← 0;  
    I ← 1;  
    Tantque I ≤ N  
        Faire  
            S ← S + I;  
            I ← I + 1;  
        Fait;  
    Ecrire('La somme S= ',S);  
Fin.
```

## Enfin la troisième Boucle

### 3- La boucle Repeter ...

Elle permet d'exécuter **plusieurs fois** un Bloc Action suivant la valeur **d'une condition**. Le nombre de fois n'est **pas forcément connu**.

**Attention**, madame, apparemment vous faites erreur car c'est la définition de la boucle **Tantque...**

**Non**, ce n'est une erreur, effectivement elle ont la même définition.

**Mais** il y a une **petite différence** entre les deux.

#### Sa syntaxe

**Repeter** <Bloc Action> **Jusqu'à** <Condition>;

<Condition>: est une expression logique.

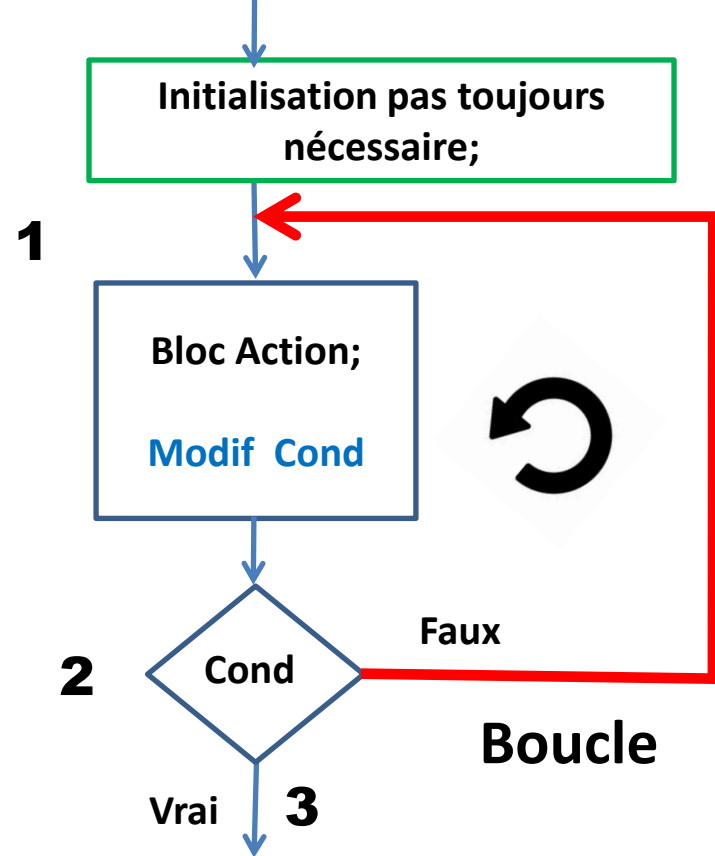
<Bloc Action> : suite d'actions séparées par des ;

**La syntaxe** est la première différence.

La deuxième différence est **La sémantique !**

## Sémantique

- 1- Exécuter Bloc Action  
    **Modifier la condition**
- 2- Si la condition vaut **Faux**  
    Aller à 1;
- 3- Si la condition vaut **Vrai**,  
    on continue après **la boucle** ;



## Remarque

- Le **Bloc Action** s'exécute au moins une fois.
- l'initialisation **n'est pas toujours nécessaire**.
- A l'inverse du Tantque, **si la condition devient Vrai, on arrête la boucle**.

**La condition du Tantque est la négation de la condition du Repeter**

## Exemple

Revenons au problème du lancer de dé. La condition d'arrêt du jeu était **d'avoir une face égale à 1**. La valeur de la face obtenue à chaque itération est introduite par une **lecture**, ce qui **modifie** la condition.

## Solution avec TANTQUE

```
Algorithme JeuDe;  
Var Face,Note:entier;  
Debut  
    Ecrire("Donner les valeurs des faces obtenues ");  
    Lire(Face);  
    Note ← Face;  
    Tantque Face ≠ 1  
        Faire  
            Lire(Face);  
            Note ← Note + Face;  
    Fait;  
    Ecrire('La note du joueur est : ',Note);  
Fin.
```

## Solution avec REPETER

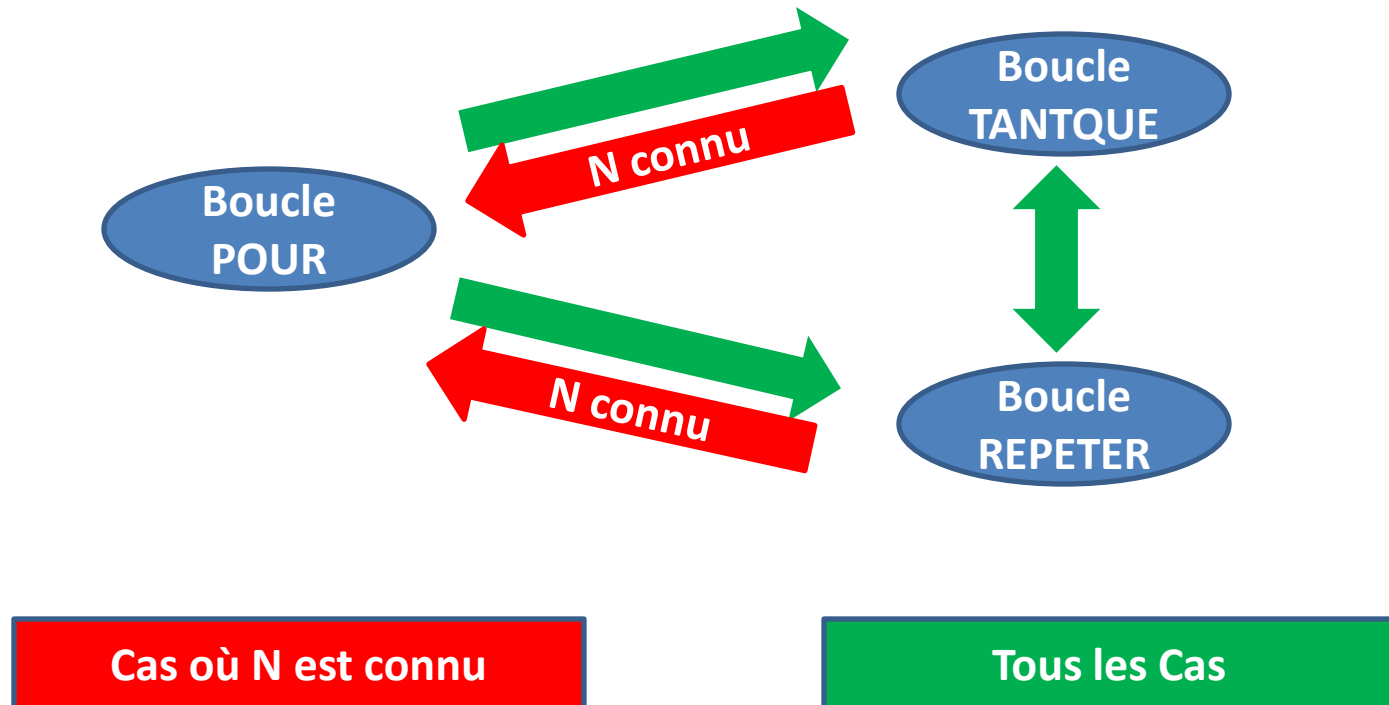
```
Algorithme JeuDe;  
Var Face,Note:entier;  
Debut  
    Ecrire("Donner les valeurs des faces obtenues ");  
    Note ← 0;  
    Repeter  
        Lire(Face);  
        Note ← Note + Face;  
    Jusqu'à Face = 1;  
    Ecrire('La note du joueur est : ',Note);  
Fin.
```

**Est-t-il toujours possible de passer du Tantque au Repeter ?**

**Oui**, on peut toujours passer de la boucle **Tantque** à la boucle **Repeter** et **vis versa**. Mais parfois on a besoin d'ajouter quelques actions.

**Et de la boucle Pour à la boucle Repeter ?**

**Oui**, toute boucle **POUR** peut être transformé en boucle **REPETER**  
Mais **ATTENTION** l'inverse n'est pas toujours possible



**Exemple,** reprenons l'exemple de la somme de N entiers.

## Solution avec POUR

```
Algorithme SommeN;  
Var A,S,I,N:entier;  
Debut  
  Ecrire("Donner le nombre d'éléments N ");  
  Lire(N);  
  S ← 0;  
  Pour I ← 1 à N  
  Faire  
    Lire(A); S ← S + A;  
  Fait; I ← I+1 automatique  
  Ecrire('La somme est : ',S);  
Fin.
```

## Solution avec REPETER

```
Algorithme SommeN;  
Var A,S,I,N:entier;  
Debut  
  Ecrire("Donner le nombre d'éléments N ");  
  Lire(N);  
  S ← 0;  
  I ← 1; initialisation  
  Repeter  
    Lire(A); S ← S + A;  
    I ← I+1; Faite par l'utilisateur  
  Jusqu'à I > N ;  
  Ecrire('La somme est : ',S);  
Fin.
```



## Les Commentaires

Les Commentaires sont des messages insérés dans l'algorithme pour éclaircir quelques actions.

Les Commentaires ne sont pas exécutable par la machine, ils sont destinés au programmeur afin de l'aider à comprendre le rôle de l'algorithme.

### Syntaxe

Commentaire sur une ligne : `// ceci est un message sur 1 ligne`

Commentaire sur plusieurs lignes : `/* ceci est un  
message sur  
plusieurs  
lignes*/`

## Exemple

Algorithme SommeN;

Var A,S,I,N:entier; //déclaration des variables

Debut

Ecrire("Donner le nombre d'éléments N ");

Lire(N); //lecture de N

$S \leftarrow 0$ ;

Pour  $I \leftarrow 1$  à N

Faire

Lire(A);  $S \leftarrow S + A$ ;

Fait;

/\* affichage du résultat  
de la somme \*/

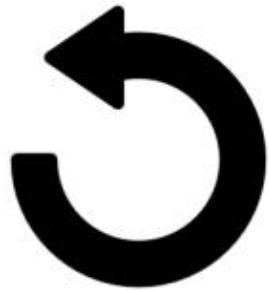
Ecrire('La somme est : ',S);

Fin.

## Un autre type de problème ?!

Revenons au problème du lancer de dé. Et supposons maintenant qu'on a **plusieurs** joueurs. Il s'agit d'afficher la note de **chaque** joueur. Soit par exemple **N** joueurs.

On aura donc



Le joueur **J=1**

Il va jouer son tour (une boucle)



Le joueur **J=2**

Il va jouer son tour (une boucle)



3 ----- N-1

Le joueur **J=N**

Il va jouer son tour (une boucle)



**Boucler  
sur les  
joueurs**

## C'est ce qu'on appelle des boucles imbriquées

**Solution :** puisque le nombre de joueurs est connu (**N**), on peut utiliser une boucle **<Pour>** pour boucler sur les joueurs. On utilise donc un compteur, soit **J** par exemple.

Algorithme JoueursDe;

Var Face,Note,J,N:entier;

Debut

Ecrire("Donner le nombre de joueurs : "); Lire(N);

Pour J ← 1 à N

Faire

Ecrire("Donner les valeurs des faces obtenues par le joueur : ", J);

Lire(Face); Note ← Face;

Tantque Face ≠ 1

Faire

Lire(Face);

Note ← Note + Face;

Fait;

Ecrire('La note du joueur ', J, ' est : ', Note);

Fait;

Fin.



Boucle interne  
Pour un joueur J

Boucle externe

## C'est BON, maintenant on affiche le numéro du vainqueur et sa note ?

Dans ce cas, on doit trouver le maximum des notes. Soit **Nmax** la note maximale et **Jmax** le numéro jour qui la détient.

Algorithme JoueursDe;

Var Face, Note, J, N, Nmax, Jmax:entier;

Debut

Nmax ← 0; *//initialiser la maximum à zéro*

Ecrire("Donner le nombre de joueurs : "); Lire(N);

Pour J ← 1 à N

Faire

Ecrire("Donner les valeurs des faces obtenues par le joueur : ", J);

Lire(Face); Note ← Face;

Tantque Face ≠ 1

Faire

Lire(Face); Note ← Note + Face;

Fait;

Ecrire('La note du joueur ', J, ' est : ', Note);

*//modifier le Maximum si nécessaire*

Si Note > Nmax Alors Nmax ← Note; Jmax ← J Fsi;

Fait;

Ecrire("Le vainqueur est le joueur : ", Jmax, " sa note est : ", Nmax);

Fin.

**A Suivre ...**

*Merci!*

