

LES INSTRUCTIONS

- L'ordre des instructions est primordial
- Le processeur exécute les instructions dans l'ordre dans lequel elles apparaissent dans le programme
- L'exécution est séquentielle : une fois que le programme a fini une instruction, il passe à la suivante
- Tant qu'une instruction n'est pas terminée, il attend avant de continuer (exemple : Saisir)

INSTRUCTIONS CONDITIONNELLES ET BOUCLES

Il peut-être nécessaire pour résoudre un problème

- de n'exécuter les instructions que sous certaines conditions
- de recommencer plusieurs fois les mêmes instructions

Il existe des instructions particulières appelées structures de contrôle qui le permettent :

- instructions conditionnelles : exécuter certaines instructions uniquement sous certaines conditions
- instructions répétitives (boucles) : répéter des instructions un certain nombre de fois (sous certaines conditions)

INSTRUCTIONS CONDITIONNELLES

Elles permettent d'exécuter des instructions différentes en fonction de certaines conditions.

Une condition est évaluée et elle est :

- soit vraie
- soit fausse

Selon le résultat, les instructions à réaliser ne sont pas les mêmes.

Il y a 3 types d'instructions conditionnelles :

- instruction conditionnelle au sens strict : **Si ... Alors**
- instruction alternative : **Si ... Alors ... Sinon**
- instruction conditionnelle multiple : **Selon ... Faire**

INSTRUCTION CONDITIONNELLE STRICTE

- Elle permet d'exécuter une ou plusieurs instructions si une condition est respectée, et ne rien faire si la condition est fausse

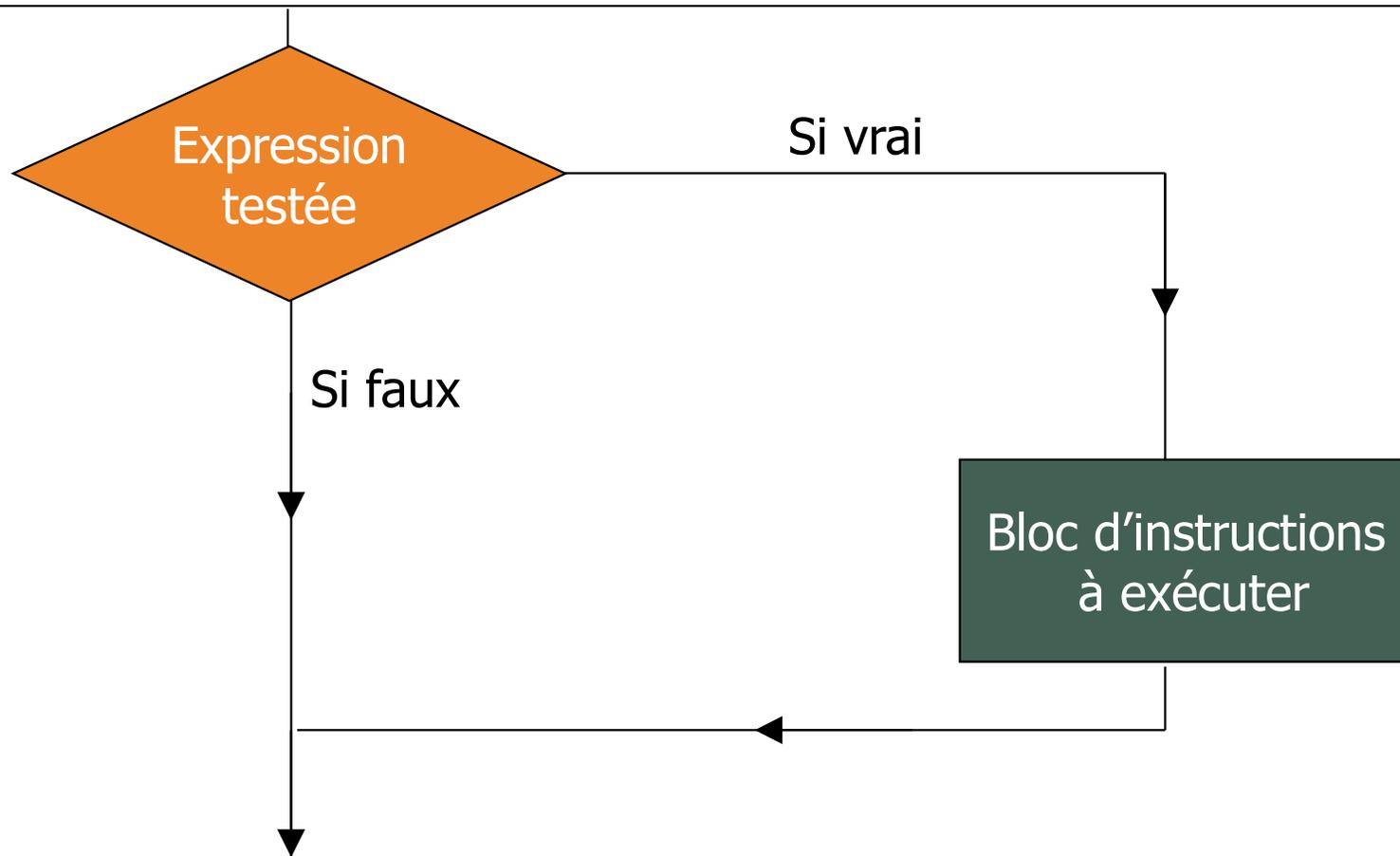
► Une instruction :

```
Si <condition>  
    Alors <instruction>  
Fin Si
```

► Un bloc d'instructions :

```
Si <condition>  
Alors  
    <instruction1>  
    <instruction2>  
    <instruction3>  
Fin si
```

EXÉCUTION



EXEMPLE

ALGORITHME voter

CONSTANTE majorite \leftarrow 18 : entier

VARIABLE age : entier

DEBUT

Afficher («Quel est votre âge ?»)

Saisir (age)

Si age \geq majorite

Alors

Afficher («Vous pouvez voter car vous êtes majeur depuis : », age – 18, « ans »)

Fin si

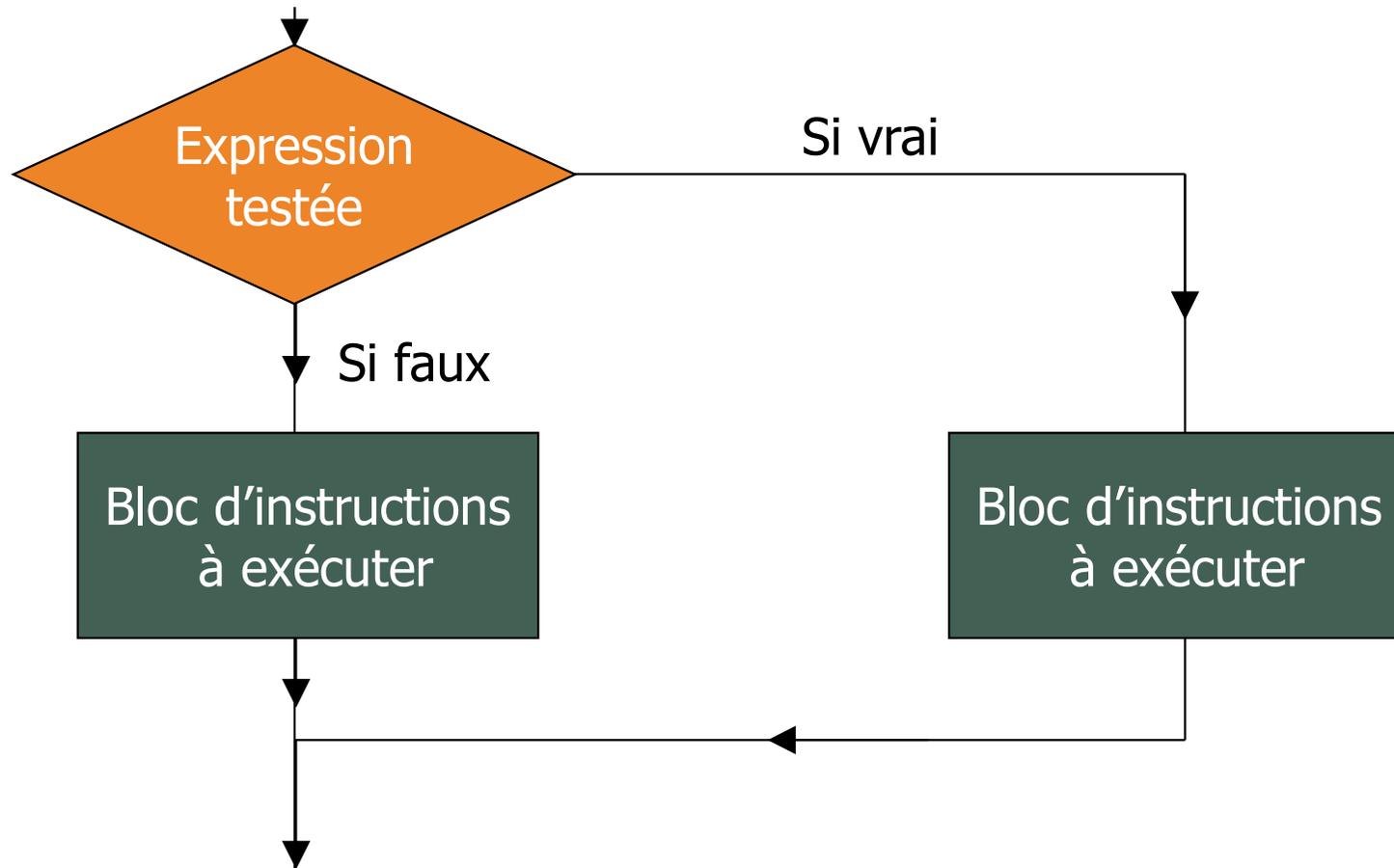
FIN

INSTRUCTION ALTERNATIVE

- Elle permet de choisir entre deux actions, suivant une condition.
- L'instruction alternative va permettre d'effectuer des choix.

```
Si <condition>  
  Alors  
    <instruction1>  
    <instruction2>  
  Sinon  
    <instruction3>  
    <instruction4>  
Fin si
```

EXÉCUTION



EXEMPLE

ALGORITHME voter

CONSTANTE majorite \leftarrow 18 : entier

VARIABLE age : entier

DEBUT

Afficher («Quel est votre âge ?»)

Saisir (age)

Si age \geq majorite

Alors

(Afficher «Vous pouvez voter car vous êtes majeur depuis : », age – 18, « ans»)

Sinon

(Afficher «Vous ne pouvez pas voter, vous devez attendre : », 18 - age , « ans»)

Fin si

FIN

EXPRESSION CONDITIONNELLE

Une expression conditionnelle (ou expression booléenne) est :

- soit VRAIE
- soit FAUSSE

Il y a plusieurs types :

- condition simple
- condition complexe
- variable booléenne

CONDITION SIMPLE

- Une condition simple est une comparaison de deux expressions de même type
- Les symboles de comparaison sont :
 - $<$
 - $>$
 - $=$
 - \leq
 - \geq
 - \neq

EXEMPLES

Si $c = 'a'$

Alors (Afficher « le caractère est a »)

Fin Si

Si $r = 3,3 * x$

Alors (Afficher « l'expression est vraie »)

Fin Si

Si $(x - 3 + y) * a \leq z - 2 + b / 3$

Alors (Afficher « l'expression est vraie »)

Fin Si

CONDITION COMPLEXE

- Une condition complexe est une comparaison formée de plusieurs conditions simples ou variables booléennes reliées entre elles par les opérateurs logiques
- Les opérateurs logiques sont :
 - **ET**
 - **OU**
 - **NON**

EXEMPLES

Si (c1 = 'a') **ET** (c2 = 'a')

Alors (Afficher « Les caractères sont a »)

Fin Si

Si (r = 3,3 * x) **OU** (r = 3,3 * y)

Alors (Afficher « Une expression est vraie »)

Fin Si

Si (((x - 3) * a) **ET** (z + b / 3)) **OU** c < 2

Alors (Afficher « Tout est vrai »)

Fin Si

INSTRUCTIONS CONDITIONNELLES IMBRIQUÉES

Saisir (x)

Si (x < 0)

Alors Afficher (« x est négatif »)

Sinon Si (x < 10)

Alors Afficher (« x est une unité »)

Sinon Si (x < 20)

Alors (Afficher « x est une dizaine »)

Sinon (Afficher « x ≥ 20 »)

Fin si

Fin si

Fin si

INSTRUCTION CONDITIONNELLE MULTIPLE

Elle permet de choisir les instructions à effectuer en fonction de la valeur (ou de l'intervalle de valeur) d'une variable ou d'une expression.

Elle permet de remplacer une succession d'instructions

Si ... Alors

SYNTAXE

Selon expression **Faire**

valeur 1 : bloc d'instructions 1

valeur 2 : bloc d'instructions 2

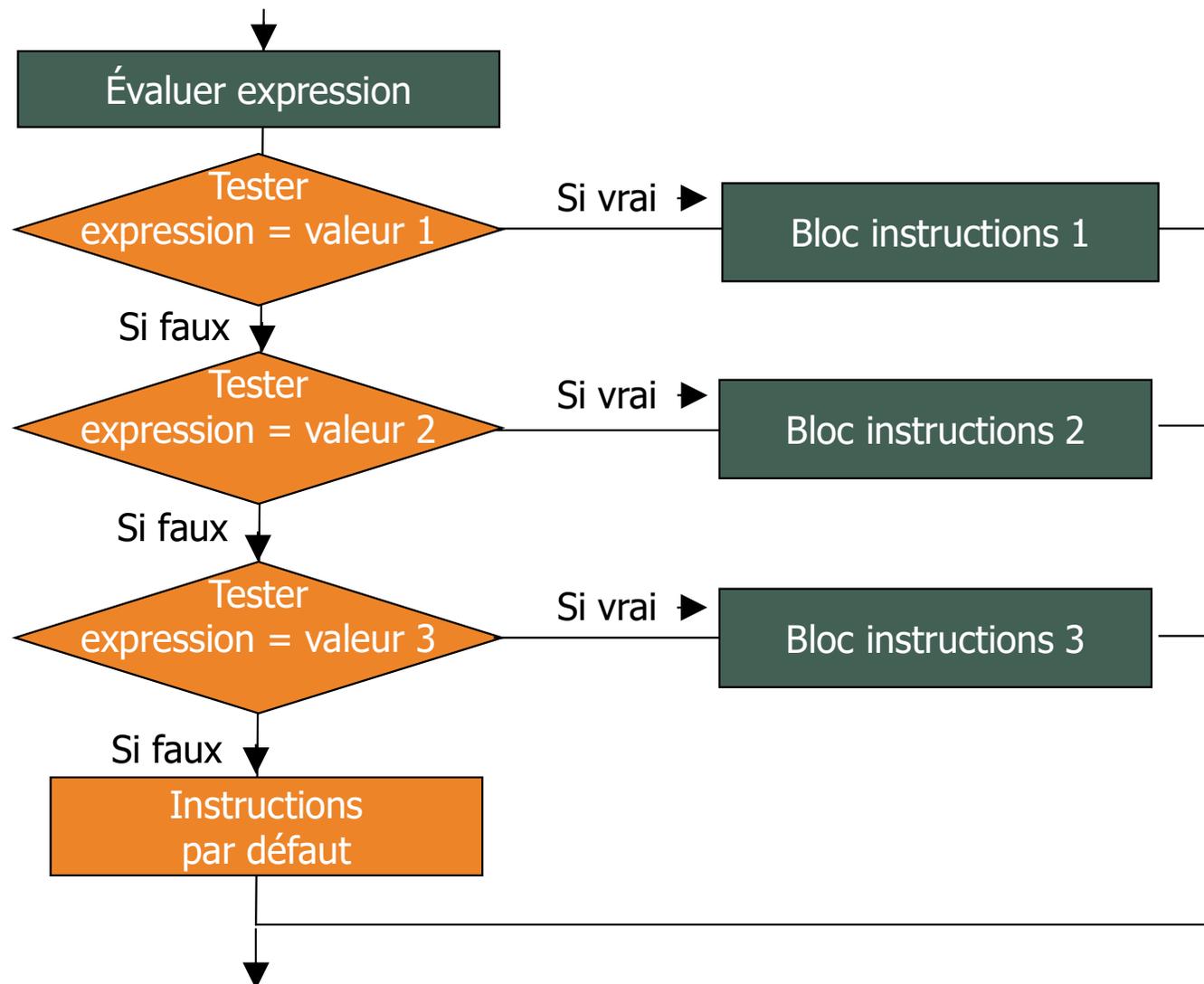
valeur 3 : bloc d'instructions 3

...

Sinon instructions par défaut (facultatif)

Fin selon

EXÉCUTION



EXEMPLE

Saisir (mois)

Selon mois **Faire**

1 : Afficher (« Janvier »)

2 : Afficher (« Février »)

3 : Afficher (« Mars »)

4 : Afficher (« Avril »)

...

11: Afficher (« Novembre »)

12: Afficher (« Décembre »)

Sinon (Afficher "Un numéro de mois doit être compris entre 1 et 12 »)

Fin selon

INSTRUCTIONS RÉPÉTITIVES (BOUCLES)

Les boucles permettent de répéter une instruction (ou un bloc d'instructions) autant de fois qu'il est nécessaire :

- soit tant qu'une condition est vraie
- soit un nombre déterminé de fois

Il y a 3 formes de boucles :

- **Tant que ... Faire**
- **Pour**
- **Répéter ... Jusqu'à**

FORME I

TANT QUE ... FAIRE

Cette forme permet d'exécuter un bloc d'instructions tant qu'une condition est remplie (satisfaite)

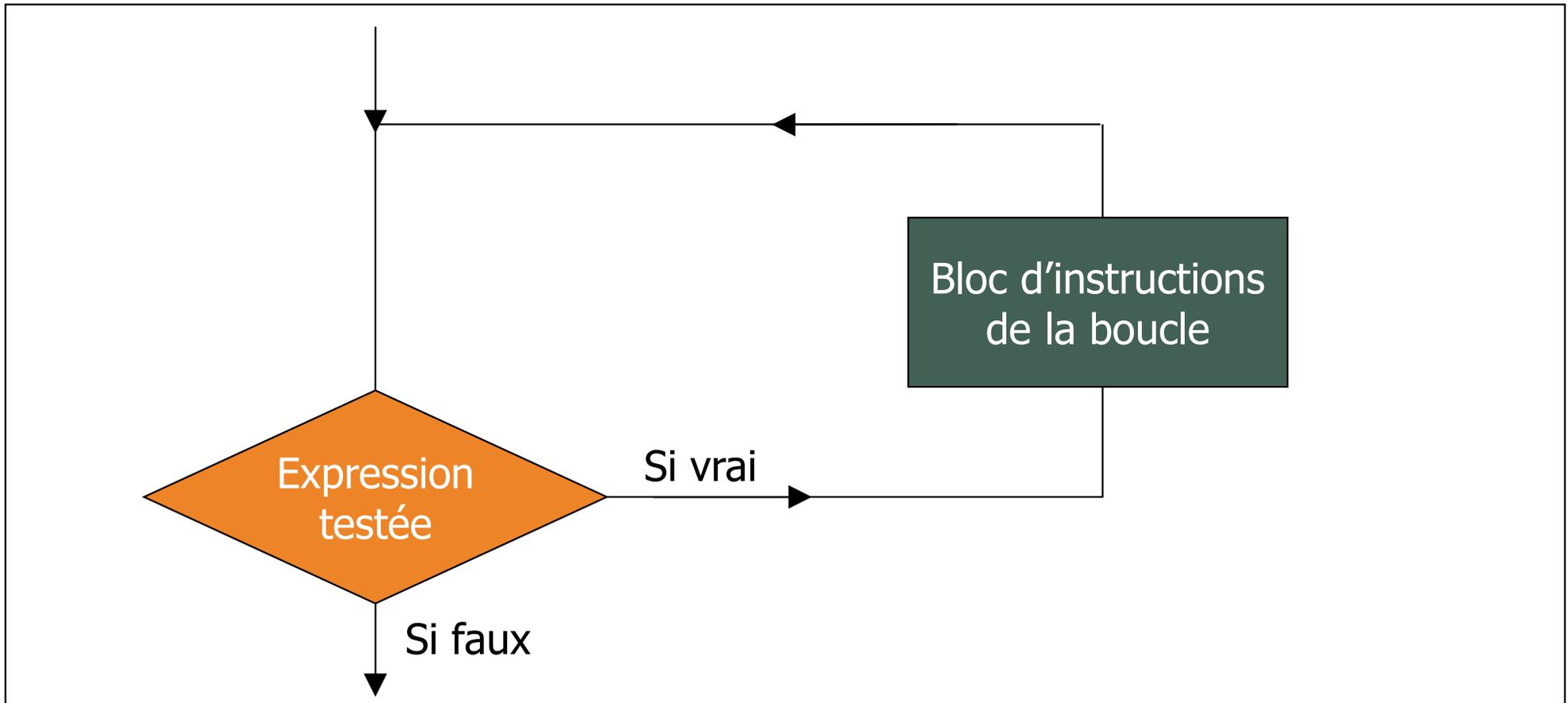
Syntaxe :

Tant que <condition> **Faire**

<instruction>

Fin Tant que

EXÉCUTION



FORME I EXEMPLE

```
/* gestion d'un budget */  
ALGORITHME forme1
```

```
VARIABLES budget, depense : entier
```

```
DEBUT
```

```
  Afficher (« Quel est le budget de départ ? »)
```

```
  Saisir (budget)
```

```
  Tant que budget > 0 Faire
```

```
    Afficher (« Vous ne pouvez pas dépenser plus de », budget)
```

```
    Afficher (« Combien voulez vous dépenser ? »)
```

```
    Saisir (depense)
```

```
    budget ← budget – depense
```

```
  Fin Tant que
```

```
  Afficher (« Le budget a été dépassé de: », budget * (-1))
```

```
FIN
```

FORME 2 BOUCLE POUR

Cette forme permet de répéter un bloc d'instructions un certain nombre de fois

Syntaxe :

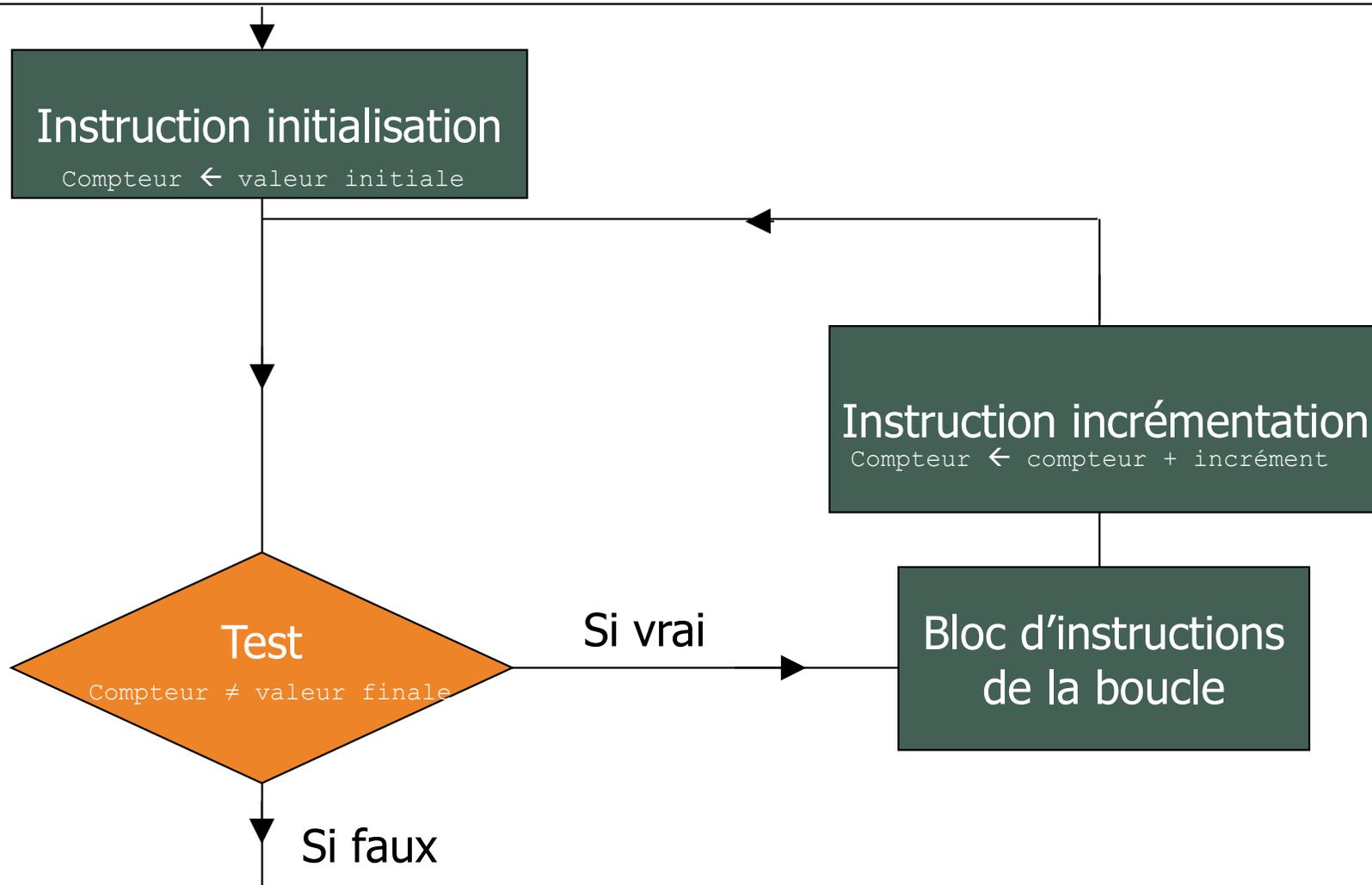
Pour <compteur> **allant de** <valeur initiale> **à** <valeur finale>

par pas de <incrément> **Faire**

<bloc instructions>

Fin Pour

EXÉCUTION



FORME 2

EXEMPLE

/* multiplication par 3 de 10 entiers allant de 1 à 10 */

Algorithme forme2

Variable x : entier

Début

Pour x **allant de** 1 **à** 10 **par pas de** 1 **Faire**

Afficher (x, " * 3 = ", x * 3)

Fin Pour

Fin

FORME 3 BOUCLE RÉPÉTER

Cette forme permet de répéter un bloc d'instructions jusqu'à ce qu'une condition soit satisfaite

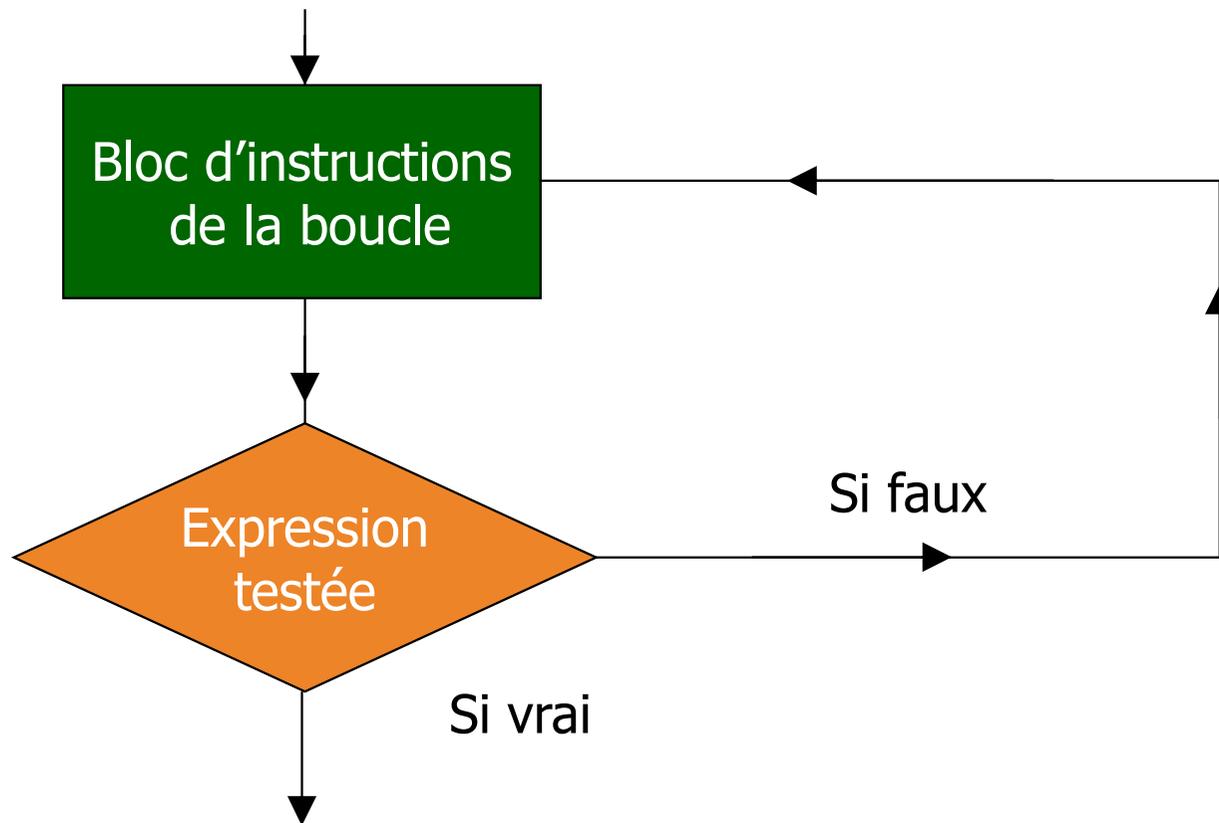
Syntaxe :

Répéter

<instructions>

Jusqu'à *<condition>*

EXÉCUTION



FORME 3

EXEMPLE I

Algorithme carré

Variables x , res : entier

rep : caractère

Début

Répéter

Afficher (« Entrez un entier : »)

Saisir (x)

Afficher (x , « au carré = », $x * x$)

Afficher (« Voulez-vous continuer ? (O/N)»)

Saisir (rep)

Jusqu'à (rep = 'N')

Afficher (« Merci, au revoir »)

Fin

FORME 3

EXEMPLE 2

Algorithme exposant

Variables x, n, res : entier
rep : caractère

Début

Afficher (« Veuillez saisir un entier : »)

Saisir (x)

$res \leftarrow x$

$n \leftarrow 1$

Répéter

$res \leftarrow res * res$

$n \leftarrow n + 1$

Afficher (x , « puissance », n , « = », res)

Afficher (« Voulez-vous continuer ? (O/N)»)

Saisir (rep)

Jusqu'à (rep = 'N')

Afficher (« merci, au revoir »)

Fin