

Structures Conditionnelles en Algorithmique

Qu'est-ce qu'une structure conditionnelle ?

Une structure conditionnelle permet à un programme de **prendre des décisions** et d'exécuter différentes instructions selon que certaines conditions sont vraies ou fausses.

Analogie : Imaginez que vous décidez quoi faire en fonction de la météo.

- **Si** il pleut, **alors** je prends mon parapluie.
- **Sinon** (s'il ne pleut pas), **alors** je ne le prends pas.

Pourquoi utiliser les structures conditionnelles ?

- **Contrôle de Flux :** Elles dirigent le chemin d'exécution du programme.
- **Adaptabilité :** Le programme s'adapte aux entrées de l'utilisateur, aux données externes ou à l'état du système.
- **Validation :** Elles permettent de vérifier la validité des données (ex: un âge ne peut pas être négatif).
- **Gestion d'Erreurs :** Elles peuvent déclencher des actions spécifiques en cas d'erreur.

Structures Conditionnelles en Algorithmique

Qu'est-ce qu'une structure conditionnelle ?

Une structure conditionnelle permet à un programme de **prendre des décisions** et d'exécuter différentes instructions selon que certaines conditions sont vraies ou fausses.

Les trois cas possibles:

Type	Description
SI (IF)	Exécuter une action si une condition est vraie
SI...SINON (IF...ELSE)	Choisir entre deux alternatives
SI...SINON SI (IF...ELSE IF)	Tester plusieurs conditions

Structure SI (IF)

Syntaxe Algorithmique

SI (condition) **ALORS**

 bloc *// Instructions à exécuter si la condition est vraie*

FIN SI

Exemple 1: Verifier si un Nombre est Positif

ALGORITHME VerifierNombrePositif

VARIABLES

 nombre : ENTIER

DEBUT

 ECRIRE ("Veuillez saisir un nombre :")

 LIRE (nombre)

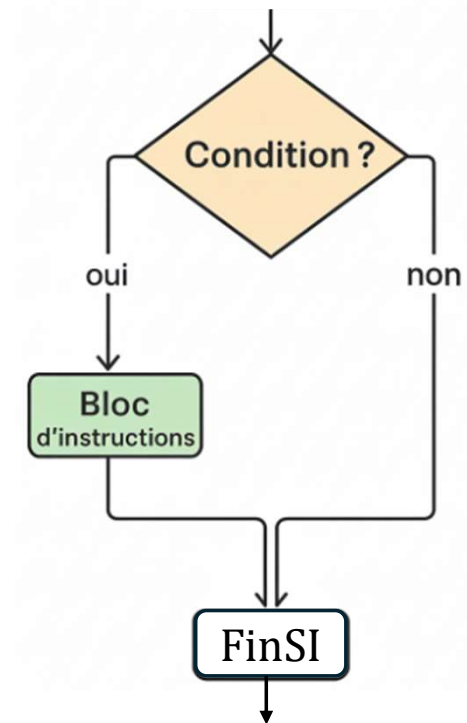
SI (nombre > 0) **ALORS**

 ECRIRE ("Le nombre est positif.")

FIN SI

 ECRIRE ("Fin du programme.")

FIN



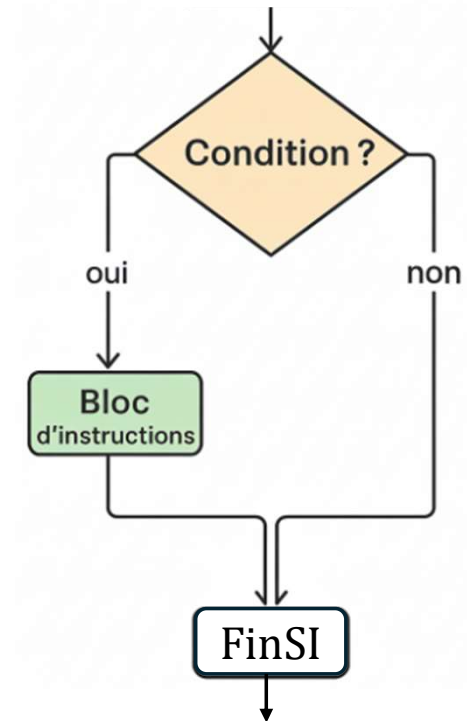
Structure SI (IF)

Syntaxe Algorithmique

SI (condition) **ALORS**
instructions
FIN SI

Applications Pratiques

- Vérifier si un étudiant est admis ou non
- Vérifier l'âge d'un utilisateur (majeur/mineur)
- Valider un mot de passe
- Afficher un message si le stock est faible
- Vérifier si un utilisateur est connecté
- Activer une alerte si la température dépasse un seuil



Structure SI...SINON (IF...ELSE)

Choisir entre deux alternatives

Syntaxe Algorithmique

SI (condition) **ALORS**

 bloc 1

SINON

 bloc 2

FIN SI

Exemple 2: Déterminer si un nombre est pair ou impair

nombre: entiere

Nombre=5

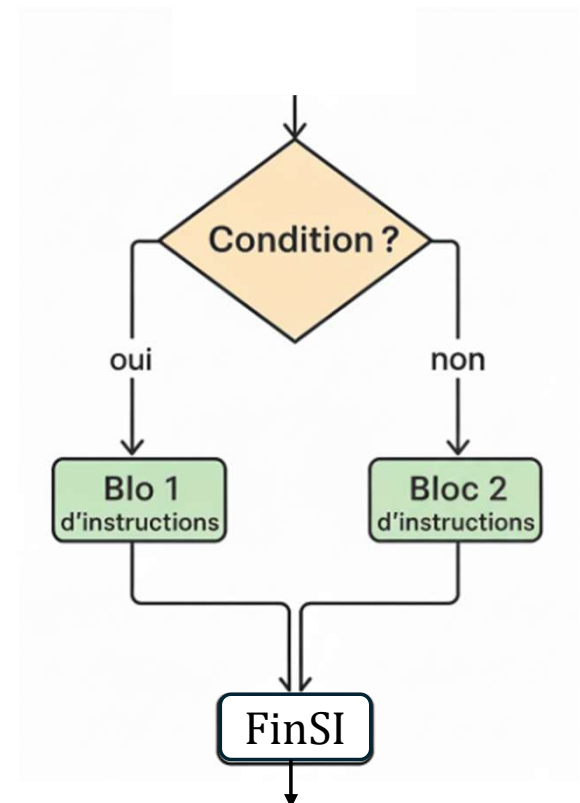
Si (nombre % 2 =0) **Alors**

 ecrire("Nombre pair")

Sinon

 ecrire("Nombre impair")

FinSi



Structure SI...SINON (IF...ELSE)

Choisir entre deux alternatives

Syntaxe Algorithmique

SI (condition) **ALORS**

 bloc 1

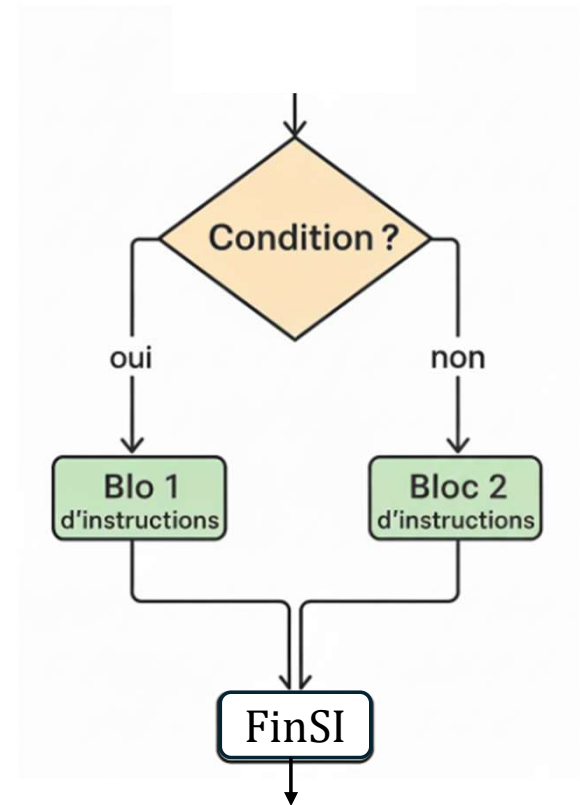
SINON

 bloc 2

FIN SI

Applications Pratiques

- Déterminer si un étudiant a réussi ou échoué
- Vérifier si un produit est en stock ou en rupture
- Autoriser ou refuser l'accès à une ressource
- Afficher "jour" ou "nuit" selon l'heure
- Calculer une remise (client VIP ou standard)
- Ticket de transport (étudiant ou non)



Structure SI...SINON SI (IF...ELSE IF)

Tester plusieurs conditions successives

Syntaxe Algorithmique

SI (condition1) **ALORS**

 bloc 1

SINON SI (condition2) **ALORS**

 bloc 2

SINON SI (condition3) **ALORS**

 bloc 3

SINON bloc_par_defaut

FINSI

FINSI

FINSI

Example 3: Attribuer une mention selon la note
note : réel

Note=14.5

Si (note >= 16) **Alors**

 ecrire("Très bien")

Sinon Si (note >= 14) **Alors**

 ecrire("Bien")

Sinon Si (note >= 12) **Alors**

 ecrire("Assez-bien")

Sinon Si (note >= 10) **Alors**

 ecrire("Passable")

Sinon

 ecrire("Insuffisant")

FinSi

FinSi

FinSi

FinSi

Structure SI...SINON SI (IF...ELSE IF)



Applications Pratiques

- Système de notation (A, B, C, D, F)
- Calcul de l'impôt par tranches
- Tarification selon l'âge (enfant, adulte, senior)
- Classification de la température (froid, tiède, chaud)
- Attribution de badges selon le score (bronze, argent, or)
- Facture d'eau de SNDE, prix par tranches de consommation

Conditions Imbriquées

Des conditions à l'intérieur d'autres conditions

```
SI (condition1) ALORS
    SI (condition2) ALORS
        bloc 1
    SINON
        bloc 2
    FIN SI
SINON
    bloc 3
FIN SI
```

Exemple 4: Vérifier l'accès à une salle de cinema. Les moins de 15ans doivent être accompagnés pour accéder à la salle.

Réponse

ALGORITHME AccesCinema

VARIABLES

age : ENTIER

est_accompagné : BOOLEEN

DEBUT

ECRIRE "Quel est votre âge ?"

LIRE age

ECRIRE ("Êtes-vous accompagné ? (oui/non)")

LIRE(est_accompagné)

SI (age < 15) **ALORS** // L'utilisateur est mineur

SI (est_accompagné = oui) **ALORS**

 ECRIRE ("Accès autorisé (enfant accompagné).")

SINON

 ECRIRE ("Accès interdit (enfant non accompagné).")

FIN SI

SINON // L'utilisateur est majeur ou a 15 ans et plus

 ECRIRE ("Accès autorisé (adulte).")

FIN SI

ECRIRE ("Bonne séance !")

FIN

Exercice 1 : Convertisseur de Devises (Euro/Dollar → Ouguiya)

Écrire un algorithme qui :

1. Demande à l'utilisateur de saisir un **montant**
2. Demande de choisir la **devise** (Euro ou Dollar)
3. Convertit le montant en **Ouguiya mauritanien (MRU)**
4. Affiche le résultat de la conversion
5. Affiche un **message d'erreur** si la devise n'est ni Euro ni Dollar

Sachant que l'utilisateur peut saisir *EUR* ou *eur* pour dire la devise euro ou bien *USD* ou *usd* pour dire le dollar.

Exemples d'exécution

```
=====
CONVERTISSEUR DE DEVICES
EUR/USD vers Ouguiya (MRU)
=====

Entrez le montant à convertir : 100
Entrez la devise (EUR ou USD) : EUR

Résultat : 100.00 EUR = 4400.00 MRU
=====
```



Taux de Change (valeurs approximatives)

- 1 Euro (EUR) = 44 MRU
- 1 Dollar (USD) = 40 MRU

```
=====
CONVERTISSEUR DE DEVICES
EUR/USD vers Ouguiya (MRU)
=====

Entrez le montant à convertir : 75
Entrez la devise (EUR ou USD) : GBP

✗ ERREUR : Devise non reconnue !
Veuillez entrer EUR ou USD uniquement.
=====
```

ALGORITHME ConversionDevise

VARIABLES

montant : RÉEL
devise : CHAÎNE
resultat : RÉEL

CONSTANTES

TAUX_EURO : RÉEL = 44
TAUX_DOLLAR : RÉEL = 40

DÉBUT

ÉCRIRE "===== "
ÉCRIRE " CONVERTISSEUR DE DEVISES"
ÉCRIRE " EUR/USD vers Ouguiya (MRU)"
ÉCRIRE "===== "

ÉCRIRE "Entrez le montant à convertir : "
LIRE montant

ÉCRIRE "Entrez la devise (EUR ou USD) : "
LIRE devise

→ **SI** (devise = "EUR" **OU** devise = "eur") ALORS
 resultat = montant × TAUX_EURO
 ÉCRIRE montant, " EUR = ", resultat, " MRU"
 SINON SI (devise = "USD" **OU** devise = "usd") ALORS
 resultat = montant × TAUX_DOLLAR
 ÉCRIRE montant, " USD = ", resultat, " MRU"
 SINON
 ÉCRIRE "ERREUR : Devise non reconnue !"
 FIN SI
FIN SI
FIN

Exercice 2 : Calculer l'IMC (Indice de Masse Corporelle) d'un individu en fonction de son poids et de sa taille et afficher la catégorie. $IMC = \text{Poids (kg)} / \text{Taille}^2 \text{ (m)}$

- $IMC < 18.5$: "Insuffisance pondérale"
- $18.5 \leq IMC < 25$: "Poids normal"
- $25 \leq IMC < 30$: "Surpoids"
- $IMC \geq 30$: "Obésité"

Exercice 3 : Déterminer si une année est bissextile. Une année est bissextile si :

- Elle est divisible par 4 ET
- (est non divisible par 100 OU est divisible par 400)

Il est de bonne pratique d'utiliser les opérateurs logiques **ET** et **OU** pour combiner plusieurs conditions et d'éviter trop d'imbrications.

Algorithme Calcule_IMC

Variables:

poids, taille, imc: réel

Debut

ecrire("Entrez votre poids (en kg) : “)

Lire(poids)

ecrire("Entrez votre taille (en m) : “)

lire(taille)

$imc = poids / (taille * taille);$

Ecrire("Catégorie : “)

SI ($imc < 18.5$) Alors

ecrire(" Insuffisance pondérale ")

Sinon SI ($imc < 25$) ALors

ecrire(" Poids normal")

Sinon SI ($imc < 30$) Alors

ecrire(" Surpoids ")

Sinon

ecrire(" Obésité ")

FinSI

FinSI

FinSI

Fin

Solution Exercice 2 : Calculer l’IMC.

Solution Exercice 3: Déterminer si une année est bissextile.

Une année est bissextile si :

1.  Elle est **divisible par 4 ET**
2.  (Elle n'est **PAS divisible par 100 OU** elle est **divisible par 400**)

Exemples :

Année	Divisible par 4 ?	Divisible par 100 ?	Divisible par 400 ?	Bissextile ?
2024	 Oui	 Non	-	 OUI
2000	 Oui	 Oui	 Oui	 OUI
1900	 Oui	 Oui	 Non	 NON
2023	 Non	-	-	 NON
2100	 Oui	 Oui	 Non	 NON

ALGORITHME AnneeBissextile

VARIABLES

annee : ENTIER

DÉBUT

ÉCRIRE "Entrez une année : "

LIRE annee

SI (annee%4 = 0) ET ((annee%100 ≠ 0) OU (annee%400 = 0)) ALORS

ÉCRIRE annee, " est une année bissextile"

SINON

ÉCRIRE annee, " n'est pas une année bissextile"

FIN SI

FIN

📄 Tableau de Test

Année	Calcul	Résultat
2020	2020 % 4 = 0, 2020 % 100 ≠ 0	✅ Bissextile
2021	2021 % 4 = 1	❌ Non bissextile
2024	2024 % 4 = 0, 2024 % 100 ≠ 0	✅ Bissextile
2000	2000 % 400 = 0	✅ Bissextile
1900	1900 % 100 = 0, 1900 % 400 ≠ 0	❌ Non bissextile
2100	2100 % 100 = 0, 2100 % 400 ≠ 0	❌ Non bissextile
2400	2400 % 400 = 0	✅ Bissextile



Les boucles (Structures Répétitives)

Après avoir maîtrisé les structures conditionnelles pour prendre des décisions, la prochaine étape cruciale en algorithmique est d'apprendre à répéter des actions. C'est le rôle des **boucles**, également appelées structures répétitives.

Pourquoi utiliser les boucles ?

- **Répétition d'actions** : Exécuter le même bloc de code plusieurs fois sans le réécrire.
- **Traitement de collections** : Parcourir des listes, des tableaux, des chaînes de caractères.
- **Attente d'une condition** : Répéter une action jusqu'à ce qu'une certaine condition soit remplie

3 boucles: TANT QUE, POUR et REPETER-JUSQU'A

La boucle TANT QUE

Syntaxe Algorithmique

TANT QUE (condition) **FAIRE**

 bloc // Instructions à répéter

FIN TANT QUE

Exemple 1 : afficher les nombres entre 1 et 10.

compteur=1

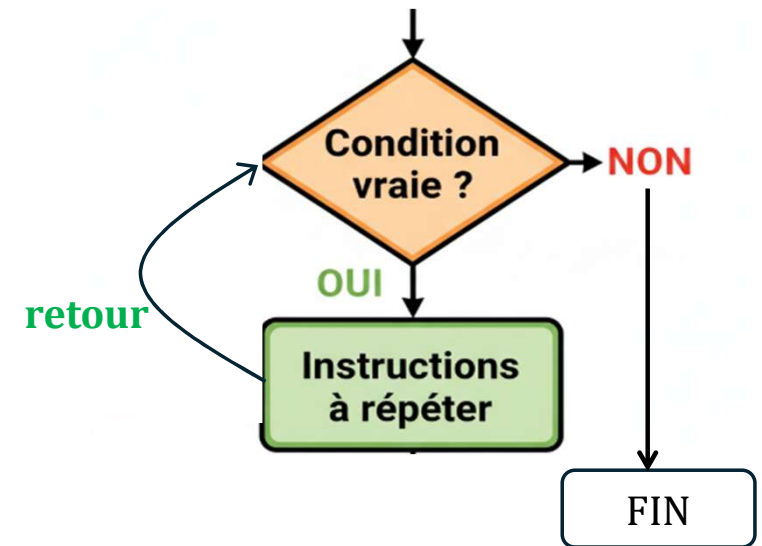
TANT QUE compteur≤10 **FAIRE**

 écrire(compteur)

 Compteur= compteur+1

FIN TANT QUE

écrire (“On a terminé de compter.”)



Caractéristiques

- ☒ Condition testée AVANT
- ☒ Peut ne jamais s'exécuter
- ☒ Nombre d'itérations peut être inconnu
- ☒ Risque de boucle infinie si la condition ne change jamais

La boucle TANT QUE

Syntaxe Algorithmique

TANT QUE (condition) **FAIRE**

 bloc // Instructions à répéter

FINTANT QUE

Exemple 2 : Demander un nombre N à l'utilisateur et afficher un compte à rebours de N jusqu'à 0.

écrire ("Entrer un nombre entier:")
lire(nombre)

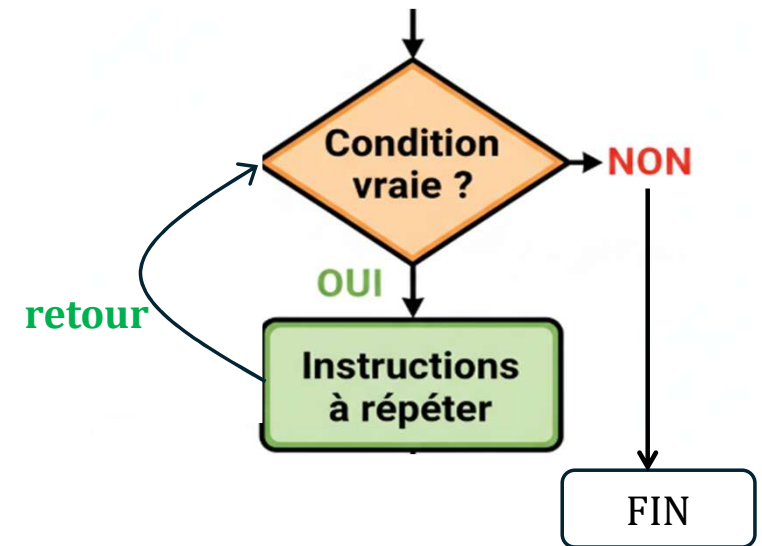
i=nombre

TANT QUE $i \geq 0$ **FAIRE**

 écrire(i)

$i = i - 1$

FINTANT QUE



La boucle TANT QUE

Exemple 3: Validation de Saisie ★

Écrire un algorithme qui demande à l'utilisateur de saisir un nombre entre 1 et 10. Tant que le nombre n'est pas dans cet intervalle, redemander la saisie.

Exemple d'exécution :

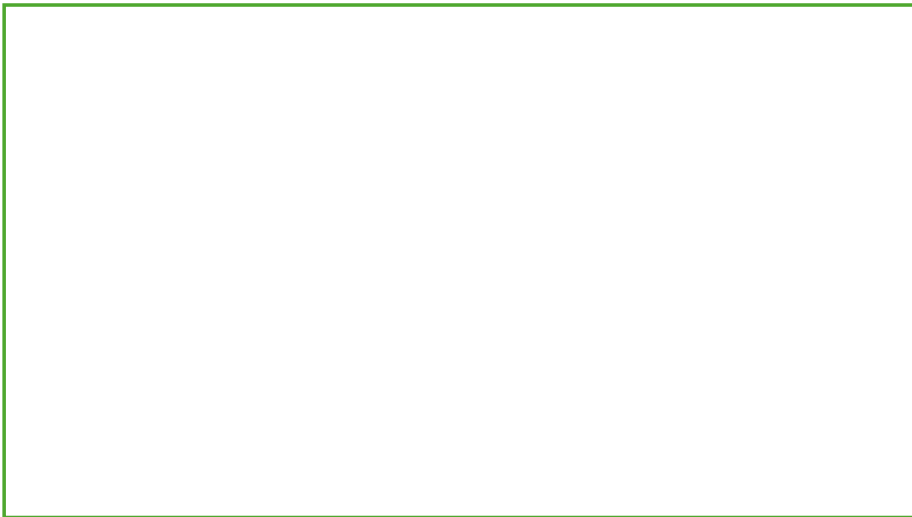
```
Entrez un nombre entre 1 et 10 : 15
Erreur ! Réessayez.
Entrez un nombre entre 1 et 10 : -3
Erreur ! Réessayez. Entrez un nombre entre 1 et 10 : 7
Merci !
```

Reponse:

La boucle TANT QUE

Exemple 4: Retrait bancaire avec solde vérifié

L'utilisateur doit donner son *solde* et le *montant* à retirer. L'algo doit forcer l'utilisateur de fournir un $0 < \text{montant} \leq \text{solde}$ pour pouvoir retirer de l'argent.



Exercice 1 : Somme de Nombres

Demander à l'utilisateur de saisir des nombres et calculer leur somme. Arrêter quand l'utilisateur entre 0.

Exemple d'exécution:

Entrez un nombre (0 pour arrêter) : 10
Entrez un nombre (0 pour arrêter) : 5
Entrez un nombre (0 pour arrêter) : 3
Entrez un nombre (0 pour arrêter) : 0
Somme totale : 18

```
nombre, somme: entier
somme = 0
ecrire("Entrez un nombre (0 pour arrêter) : " )
lire(nombre)
Tant que (nombre != 0) Faire
    somme= somme+ nombre
    écrire("Entrez un nombre (0 pour arrêter) : " )
    lire(nombre)
FinTantque
ecrire("Somme totale : ", somme )
```

La boucle TANT QUE

Exercice 2: Développez un algo qui fait deviner un nombre secret à l'utilisateur. L'algo doit guider le joueur en indiquant si sa proposition est trop petite ou trop grande jusqu'à ce qu'il trouve le nombre mystère.

- Le nombre secret est un entier aléatoire entre **1 et 100**
- Le joueur dispose d'un nombre maximum d'essais (10 tentatives)

Exemple d'exécution

Bienvenue au Jeu de Devinette ! Trouvez le nombre secret entre 1 et 100

Entrez votre proposition (essai 1/10) : 50

Trop grand ! Essayez plus petit

Entrez votre proposition (essai 2/10) : 25

Trop petit ! Essayez plus grand

Entrez votre proposition (essai 3/10) : 37

Bravo ! Vous avez trouvé le nombre secret en 3 essais

Vous pouvez travailler en petits groupes

La boucle POUR

Syntaxe Algorithmique

POUR variable **DE** début **À** fin [PAR PAS DE pas] **FAIRE**
instructions
FIN POUR

Presque même organigramme que la boucle tant que.

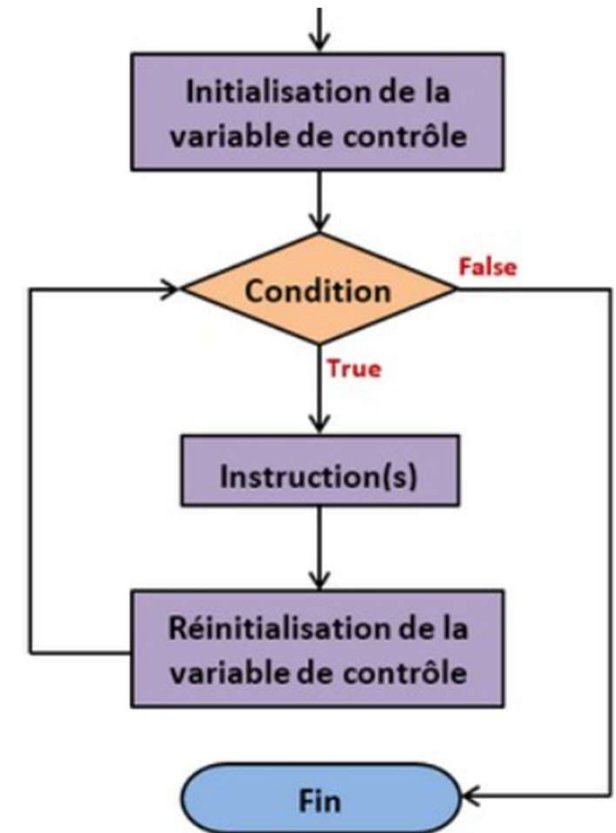
Exemple 1 : afficher les nombres entre 1 et 10.

Pour compteur de 1 A 10 **FAIRE**

écrire(compteur)

FIN Pour

ecrire (“On a terminé de compter.”)



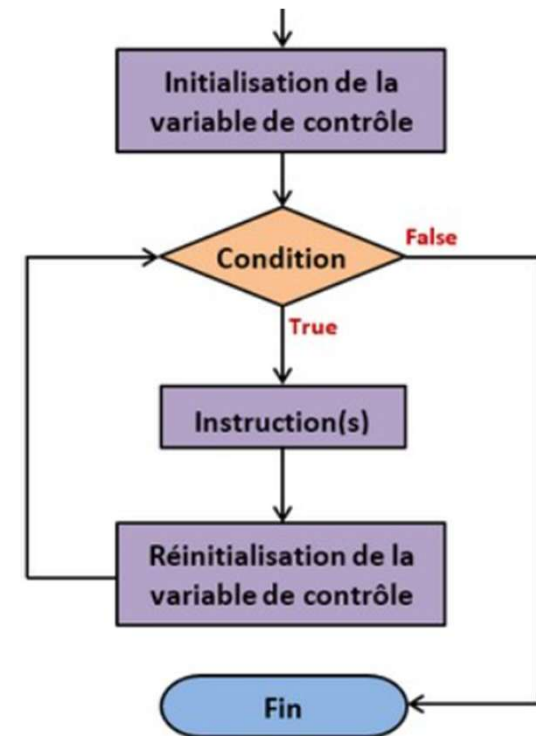
Note : Le *pas* est optionnel, par défaut il vaut 1

La boucle POUR

Syntaxe Algorithmique

POUR variable **DE** début **À** fin [PAR PAS DE pas] **FAIRE**
instructions
FIN POUR

Exemple 2 : Boucle avec pas différent
Afficher les nombres pairs entre 0 et 20 :



La boucle POUR

Exercice 1: Table de Multiplication

Afficher la table de multiplication d'un nombre N saisi par l'utilisateur (de 1 à 10).

Exemple : Si $N = 7$

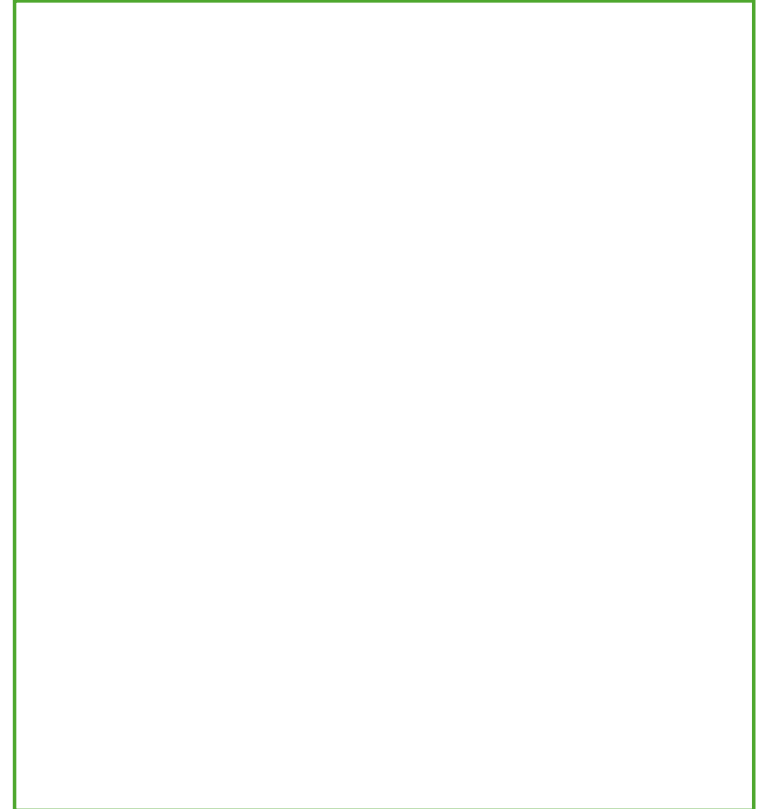
$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$



La boucle REPETER



Syntaxe Algorithmique

RÉPÉTER

instructions

JUSQU'À (condition)

Exemple 1 : Saisie d'un nombre positif. L'utilisateur est forcé de saisir un nombre strictement positif.

FIN