

# Structures Répétitives: Les boucles

# Structures Répétitives: Les boucles

- Une structure répétitive (Boucle) est un ensemble d'actions qui se répète toujours dans un ordre précis, un nombre **finis** de fois
- En algorithmique nous disposons de 3 structures:
  - La boucle Tantque.... Faire
  - La boucle Repeter ....Jusqu'à
  - La boucle Pour.....Faire

# 1- La boucle Tantque... Faire

# La boucle Tantque.... Faire

Syntaxe:

**TantQue** (condition) **Faire**

Bloc d'instructions



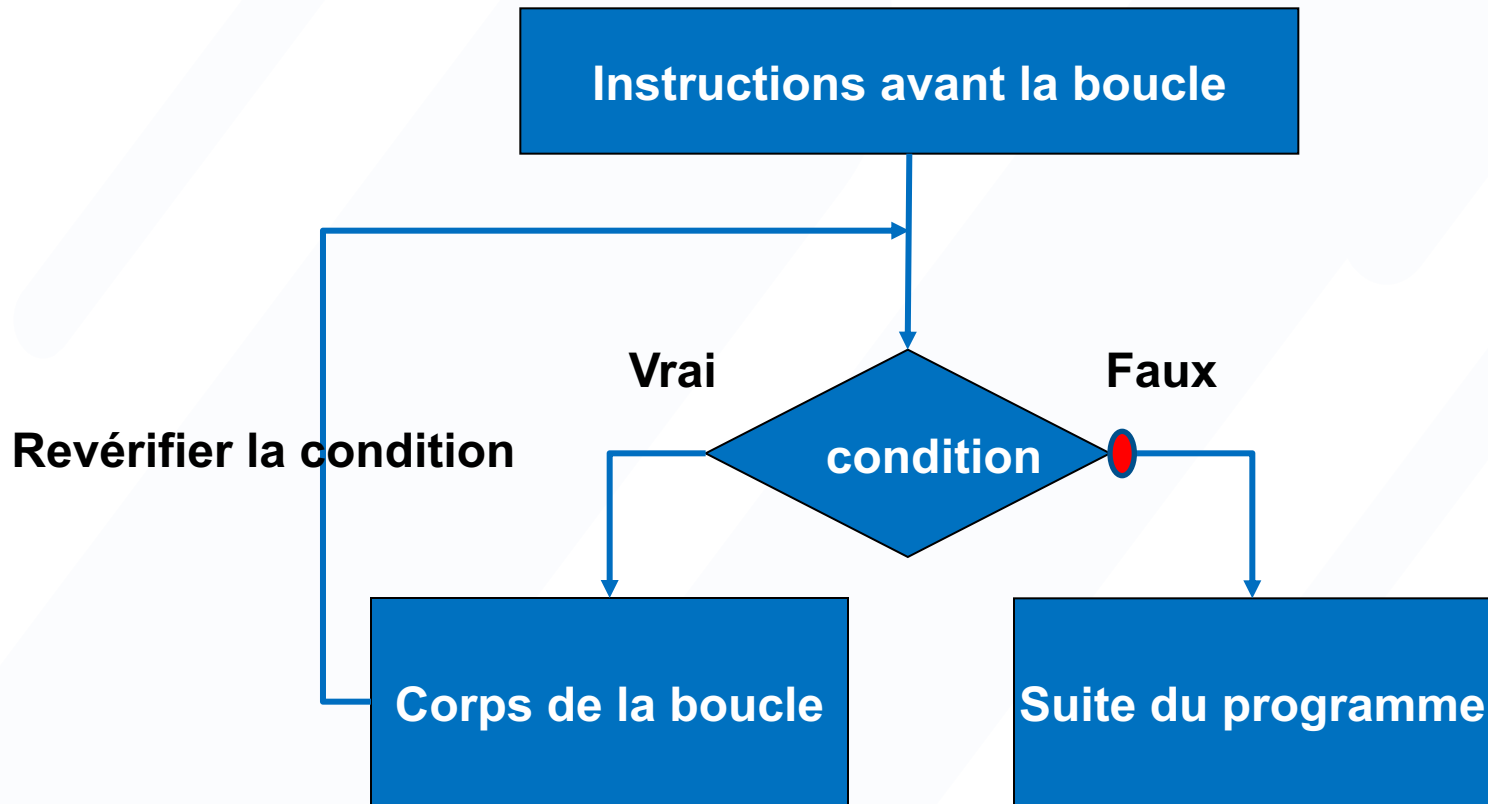
**Corps de la boucle**

**FinTantQue**

## Exemple 1 : Afficher tout les nombres pairs $\leq 1000$

```
Algorithme paire
Variables : i: entier
Début
i  $\leftarrow$  0
Tantque (i  $\leq$  1000)
    Ecrire(i)
    i  $\leftarrow$  i+2
Fintantque
Ecrire( " c'est fini " )
Fin
```

# Algorithme de la boucle Tantque



# Remarques

- La condition (**dite condition de contrôle de la boucle**) est évalué avant chaque itération
  - Si la condition est **vrai** : on exécute le corps de la boucle, puis on retourne vérifier la condition, si elle est encore vrai on répète l'exécution du corps de la boucle.

→ Si la condition est **fausse** : on sort de la boucle et on exécute l'instructions qui vient après TantQue

- Le nombre d'itérations dans une boucle TantQue n'est pas connu au moment d'entrée dans la boucle. Il dépend de l'évolution de la valeur de condition
- Une des instructions du corps de la boucle doit absolument changer la valeur de condition de vrai à faux (après un certain nombre d'itérations), sinon le programme tourne indéfiniment

# Exemple 2

- Calculer la somme des N premiers nombres entiers

$$S=1+2+3+4.....+N$$

Algorithme Somme

Variables : S,i,n: entier

Début

Ecrire (" Entrez la valeur de N ")

Lire (N)

$S \leftarrow 0$

$i \leftarrow 1$

Tantque ( $i \leq N$ )

$S \leftarrow S + i$

$i \leftarrow i+1$

Fintantque

Ecrire (" La somme de ", N, " Premiers entier est " , S)



## Exemple 3

Contrôle de saisie d'une lettre majuscule jusqu'à ce que le caractère entré soit valable

Algorithme : Majuscule

Variable c : caractère

Début

Ecrire (" Entrez une lettre majuscule ")

Lire (c)

TantQue (c < 'A' ou c > 'Z')

    Ecrire ("Saisie erronée. Recommencez")

    Lire (c)

FinTantQue

Ecrire ("Saisie valable")

Fin

## 2- La boucle Repeter... Jusqu'à

# Boucle Répéter ... jusqu'à ...

Syntaxe:

**Repete**

Bloc d'instructions

**Jusqu'à (condition)**



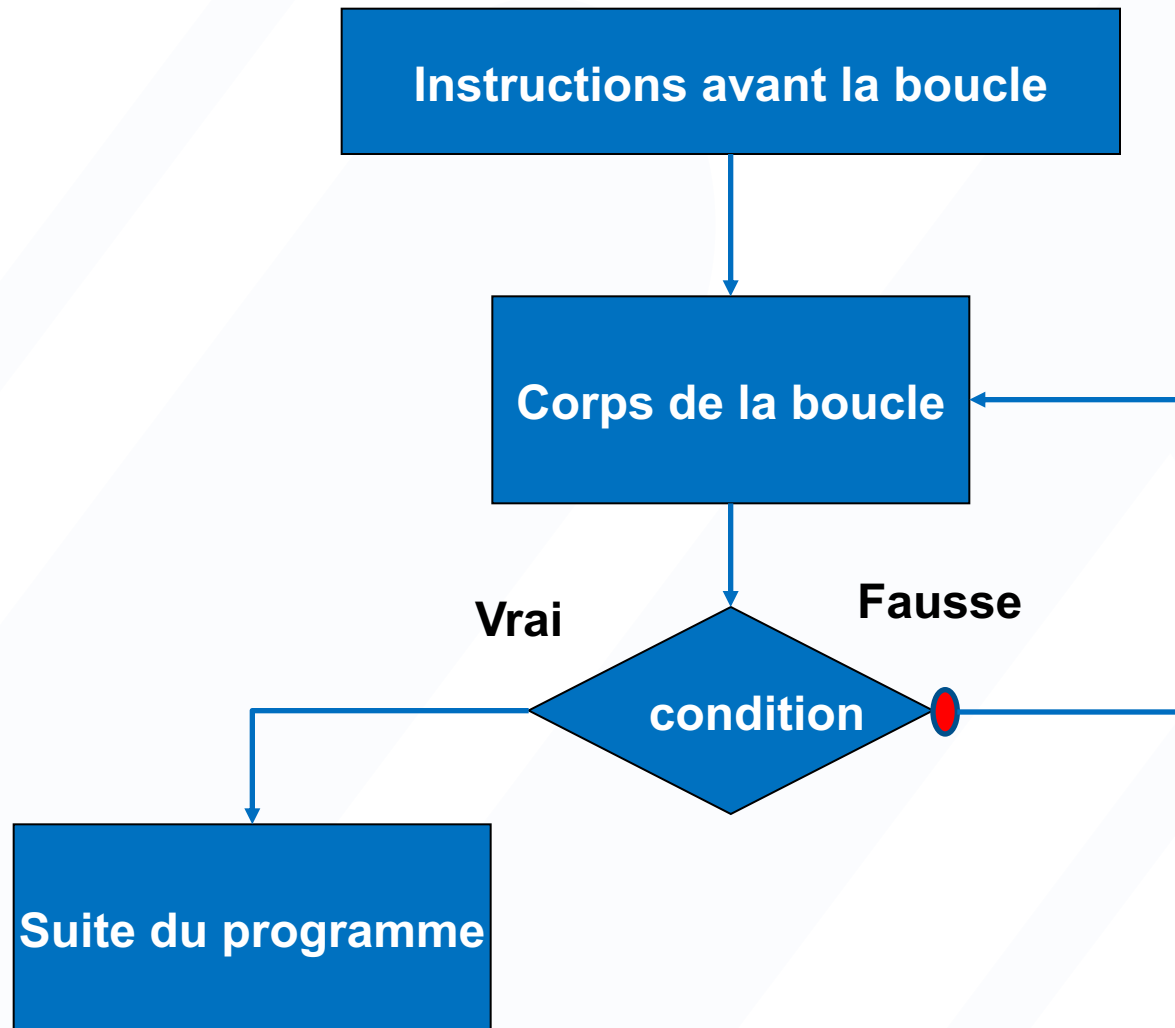
**Corps de la boucle**

## Exemple 1 : Afficher tout les nombres pairs $\leq 1000$

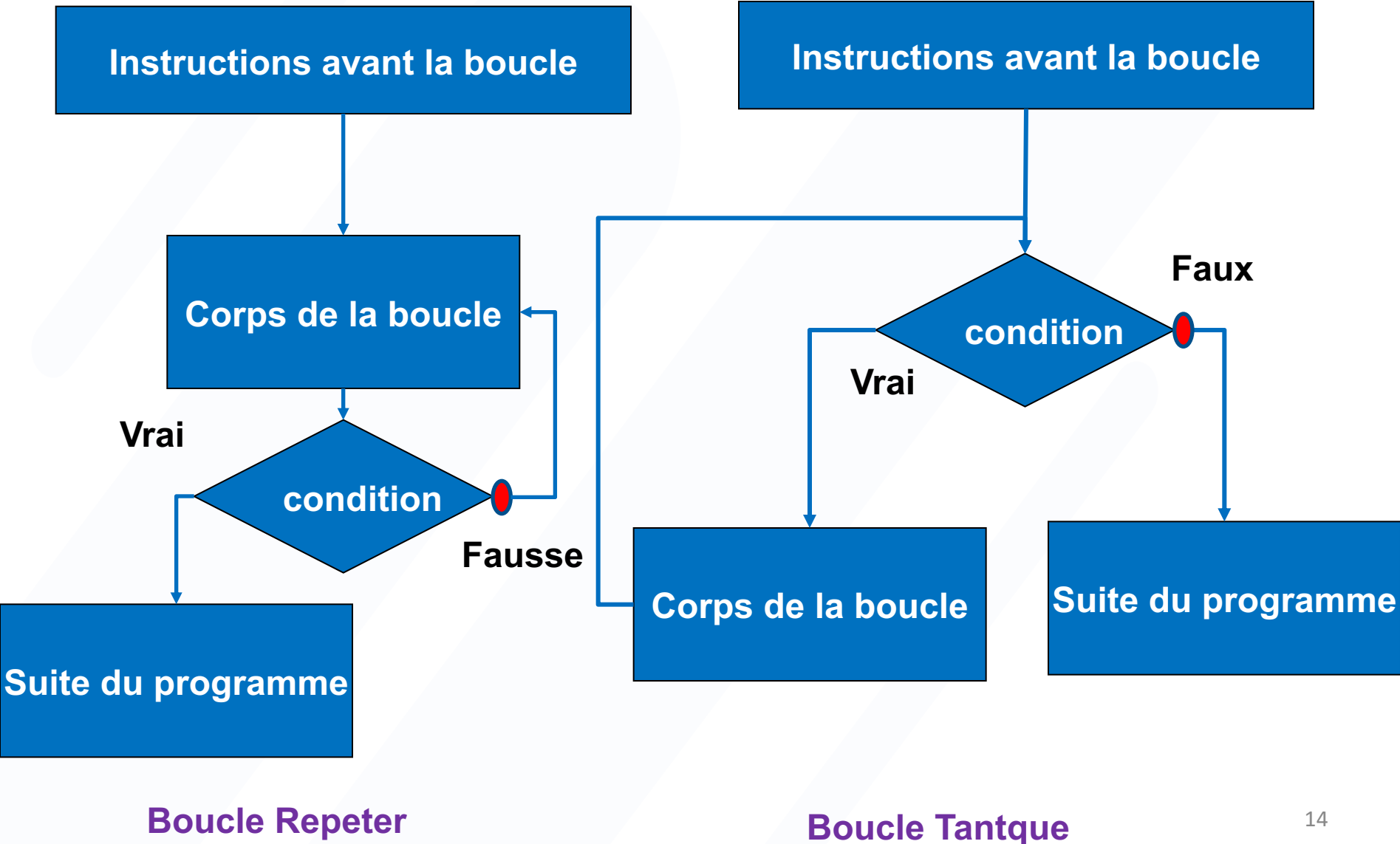
```
Algorithme paire
Variables : i: entier
Début
i  $\leftarrow$  0
Repeter
    Ecrire(i)
    i  $\leftarrow$  i+2
Jusqu'à (i > 1000)
Ecrire( " c'est fini " )
Fin
```

La condition de la boucle Tantque est la négation de la condition de la boucle Repeter

# Algorithme de la boucle Repeter



# Comparaison des algorithme Repeter et Tantque



# Exemple 2

- Calculer la somme des N premiers nombres entiers

$$S=1+2+3+4.....+N$$

Algorithme Somme

Variables : S,i,n: entier

Début

Ecrire (" Entrez la valeur de N " )

Lire (N)

$S \leftarrow 0$

$i \leftarrow 0$  // Traiter le cas de N=0

**Repeter**

$S \leftarrow S + i$

$i \leftarrow i+1$

**Jusqu'à (i > N)**

Ecrire (" La somme de ", N, " Premiers entier est " , S)

## Exemple 3

Contrôle de saisie d'une lettre majuscule jusqu'à ce que le caractère entré soit valable

Algorithme : Majuscule

Variable c : caractère

Début

Repete

    Ecrire (" Entrez une lettre majuscule ")

    Lire (c)

    jusqu'à ( c >= 'A' et c <= 'Z' )

Fin



# Remarques

- la boucle **Repeter** est utilisée lorsqu'on a des instructions qui se répètent plusieurs fois jusqu'à ce que la condition associée soit vraie
- Dans la boucle Repeter le test est placé en fin d'itération par conséquent le corps de la boucle est exécuté **au moins une fois**
- Dans la structure Tantque , le test est placé avant le corps de la boucle, il se peut que le corps de la boucle ne soit jamais exécuté.

## 2- La boucle Pour..... Faire

# Boucle *Pour*

## Syntaxe

**Pour**  $i \leftarrow$  allant de valeur\_initial à valeur finale **Faire**

Bloc d'instructions

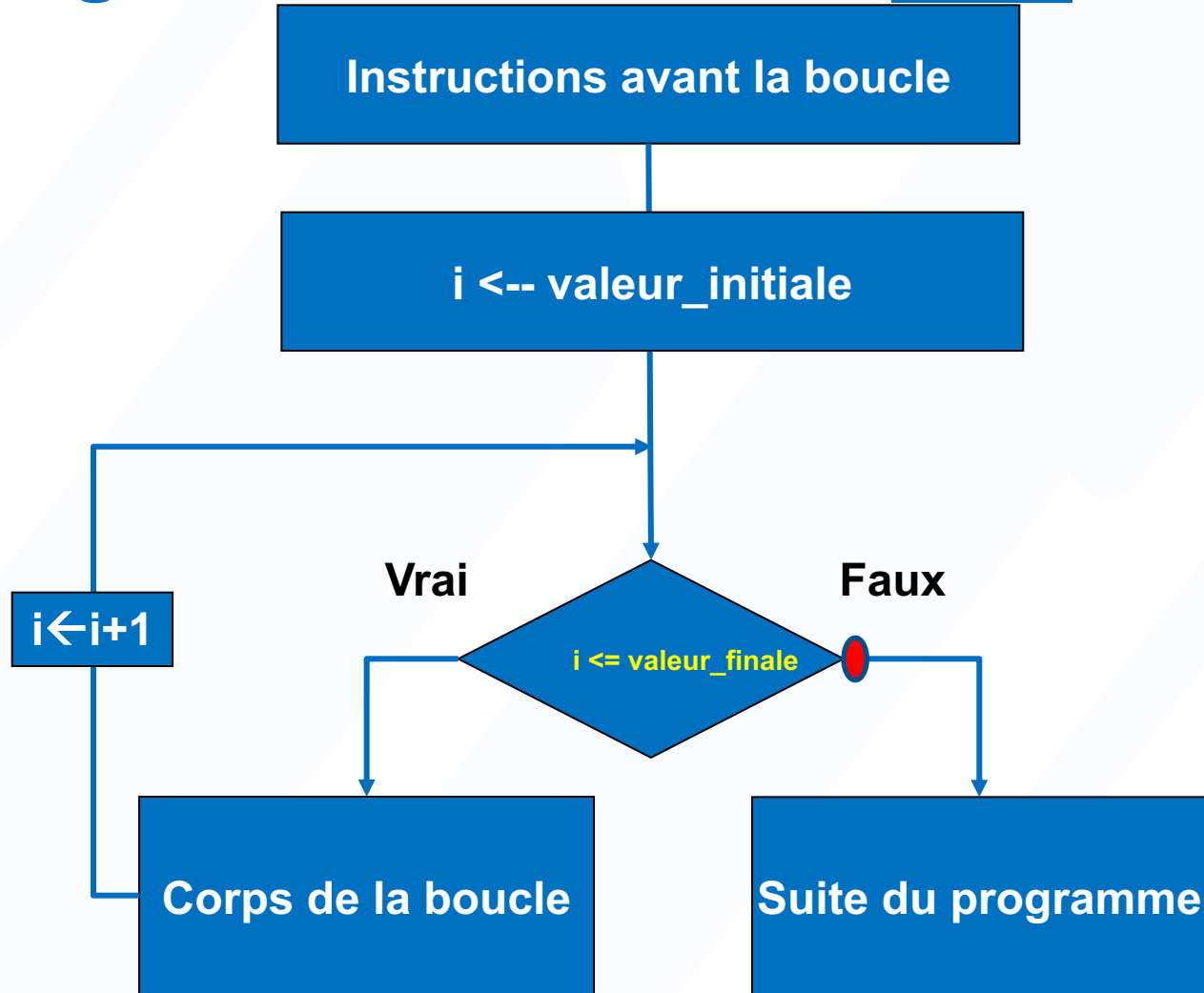
**FinPour**

## Remarque

La structure « pour » est utilisée pour traiter un bloc d'instructions qui se répète plusieurs fois et **lorsqu'on connaît à l'avance le nombre d'itération**

La variable  $i$  est appelé Compteur de la boucle : le compteur est une variable de type entier .Elle doit être déclarée.

# Algorithme de la boucle Pour



## Exemple 1 : Afficher tout les nombres pairs $\leq 1000$

Algorithme paire

Variabes : i: entier

Début

**Pour** i  $\leftarrow$  0 à 1000 pas 2 **Faire**

    Ecrire(i)

**FinPour**

    Ecrire( " c'est fini " )

Fin

**Le Pas** est un entier qui peut être positif ou négatif. **Pas** peut ne pas être mentionné, car par défaut sa valeur est égal à 1.

Ici on a la valeur du pas est égale à 2 donc la boucle exécutera automatiquement l'instruction  $i \leftarrow i+2$  avant de passe à l'itération suivante

## Exemple 2

- Calculer la somme des N premiers nombres entiers

$$S=1+2+3+4.....+N$$

Algorithme Somme

Variables : S,i,n: entier

Début

Ecrire (" Entrez la valeur de N ")

Lire (N)

$S \leftarrow 0$

**Pour i <-allant de 1 à N Faire**

$S \leftarrow S + i$

**FinPour**

Ecrire (" La somme de ", N, " Premiers entier est " , S)

# Comparaison entre Pour et TantQue

- La boucle Pour est un cas particulier de la boucle Tant Que (cas où le nombre d'itérations est connu et fixé) .
- Tout ce qu'on peut écrire avec la boucle **Pour** peut être remplacé avec **TantQue** (la réciproque est fausse)

# Choix d'un type de boucle

- Si on peut déterminer le nombre d'itérations avant l'exécution de la boucle, il est plus naturel d'utiliser *la boucle Pour*
- S'il n'est pas possible de connaître le nombre d'itérations avant l'exécution de la boucle, on fera appel à l'une des *boucles TantQue* ou *répéter jusqu'à*
- Pour le choix entre *TantQue* et *jusqu'à* :
  - Si on doit tester la condition de contrôle avant de commencer les instructions de la boucle, on utilisera *TantQue*
  - Si la valeur de la condition de contrôle dépend d'une première exécution des instructions de la boucle, on utilisera *répéter jusqu'à*



# Exercice 1:

Calcul de  $x$  à la puissance  $n$  où  $x$  est un réel non nul et  $n$  un entier positif ou nul

Algorithme : Puissance

Variables  $x$ ,  $puiss$  : réel

$n$ ,  $i$  : entier

Début

Ecrire (" Entrez la valeur de  $x$  ")

Lire ( $x$ )

Ecrire (" Entrez la valeur de  $n$  ")

Lire ( $n$ )

$puiss \leftarrow 1$

Pour  $i$  allant de 1 à  $n$

$puiss \leftarrow puissance * x$

FinPour

Ecrire ( $x$ , " à la puissance ",  $n$ , " est égal à ",  $puiss$ )

Fin

## Exercice 2

- Ecrire un algorithme qui calcule la factorielle d'un entier n

Algorithme : Factorielle

Variables n, fact : réel

i : entier

Début

Ecrire (" Entrez la valeur de n ")

Lire (n)

fact  $\leftarrow$  1

**Pour** i allant de 1 à n

fact  $\leftarrow$  fact \*i

**FinPour**

Ecrire ("n! = ", fact)

Fin