

# Plan

- ▶ Les structures de données
- ▶ Les structures conditionnelles
- ▶ Les structures itératives
- ▶ Les tableaux et les enregistrements
- ▶ Les fonctions et les procédures
- ▶ La récursivité
- ▶ Les pointeurs

# Les structures itératives

## ► Les structures itératives ou répétitives

La structure itérative répète l'exécution d'une opération ou d'un traitement.

On considérera 2 cas :

### 1. Le nombre de répétitions est connu

#### Structure itérative complète POUR (pour ... de ... à ... .. faire)

Dans cette structure, la sortie de la boucle d'itération s'effectue lorsque le nombre souhaité de répétition est atteint.

On utilise donc une variable (ou indice) de contrôle d'itération caractérisée par sa **valeur initiale**, sa **valeur finale** et son **pas** de variation.

Si la **valeur finale de l'indice** est inférieure à **sa valeur initiale** le pas de variation est **négatif**, la structure "pour" est dite «**décroissante** »; dans le cas contraire, le pas est **positif** et la structure est dite «**croissante** ».

# Les structures itératives

## ► Notation :

**POUR** **compteur** de  $V_{\text{initiale}}$  à  $V_{\text{finale}}$  [**pas** = p] faire

Action ;

**Fin pour** ;

## Exemple:

Pour c de 1 à 10 faire  
écrire (" Bonjour ",c);  
Fin pour

# Les structures itératives

## 2. Le nombre de répétitions n'est pas connu (ou variable)

On distingue 2 structures de base :

### ► **Structure RÉPÉTER JUSQU'À**

Dans cette structure, le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée.

Un traitement peut être soit une structure isolée, soit une succession d'instructions.

# Les structures itératives

→ L'action est toujours exécutée au moins une fois.

► Notation :

Répéter

Action;

Jusqu'à (condition vraie) ;

Exemple :

Répéter

écrire (" Taper un entier positif ");

lire(n)

Jusqu'à ( $n > 0$  );

# Les structures itératives

## ► Structure TANT QUE ... FAIRE

Dans cette structure, on commence par tester la condition ; si elle est vérifiée, le traitement est exécuté.

Un traitement peut être une structure isolée ou une succession d'instructions.

→ L'action peut ne jamais être exécutée.

## ► Notation :

TANT QUE (condition) faire

Action;

Fin tant que

En C :  
**While(Condition) {**  
**Traitement;**  
**}**

# Les structures itératives

## → Application

Ecrire un algorithme qui affiche la plus grande valeur parmi vingt nombres saisis successivement à partir du clavier. Ne pas utiliser la structure tableau.

### ► Exemple :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

....

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

# Les structures itératives

## Travail à faire (1)

### Exercice 1

Écrire un algorithme qui permet de calculer  $x^y$  avec  $x$  et  $y$  deux entiers saisis au clavier

### Exercice 2

Écrire un algorithme qui permet de calculer  $n!$ , avec  $n$  un entier saisi au clavier

# Les structures itératives

## Travail à faire (2)

### Exercice 3

- ♦ Un nombre parfait est un nombre présentant la particularité d'être égal à la somme de tous ses diviseurs, excepté lui-même.
- ♦ Le premier nombre parfait est :  $6 = 3 + 2 + 1$ .

Écrire un algorithme qui affiche tous les nombres parfaits inférieurs à 1000.

### Exercice 4

Ecrire un algorithme qui lit deux entiers A et B puis calcule et affiche leur PGCD en utilisant la méthode suivante :

Si  $A = B$  ;  $\text{PGCD}(A,B) = A$

Si  $A > B$  ;  $\text{PGCD}(A,B) = \text{PGCD}(A-B,B)$

Si  $B > A$  ;  $\text{PGCD}(A,B) = \text{PGCD}(A, B-A)$