



# Chapitre 2: les fonctions et les procédures



# Problème

- Nous voulons calculer la somme  $a^n + b^m$

Algorithme Somme

**Variables:** a,n,b,m,P1,P2,i: entier

Début

Lire(a,n)

Lire(b,m)

P1  $\leftarrow$  1

**Pour** i allant de 1 à n faire

    P1  $\leftarrow$  P1\*a

**FinPour**

P2  $\leftarrow$  1

**Pour** i allant de 1 à m faire

    P2  $\leftarrow$  P2\*b

**FinPour**

Ecrire(P1+P2)

Fin



Répétition du Code



Ecrire le code une fois et l'appeler plusieurs fois



La notion de fonction

# Introduction

- Les fonctions et les procédures sont des sous programmes (un ensemble d'instructions) auxquels on peut faire référence à l'intérieur d'un programme.
- Quel est l'**intérêt** de l'utilisation des fonctions et des procédures:
  - Permettre d'éviter les répétitions
  - Permettre une structuration et une meilleure lisibilité des programmes
  - Faciliter la maintenance du code (il suffit de modifier une seule fois)
  - ces procédures et fonctions peuvent éventuellement être réutilisées dans d'autres programmes

# Les fonctions

## ▪ Définition:

- Une fonction est un sous programme qui reçoit des valeurs lors de son appel et retourne un résultat à la fin de son exécution

## ▪ Déclaration:

- Une fonction est déclarée par le mot clé « **Fonction** » en précisant
- son nom (appelé aussi identificateur): il doit respecter les mêmes règles que celles pour les noms de variables
- une liste de paramètres (arguments) avec leur type
- le type du résultat qu'elle retourne
- L'instruction **retourne** qui sert à retourner la valeur du résultat

# Déclaration d'une fonction

**Fonction** nom\_fonction (paramètres et leurs types) : type\_de\_la\_valeur\_retournée

**Début**

Instructions constituant le corps de la fonction

**retourne** Résultat

**Fin**

# Exemple

**Fonction** puissance (a:entier, n:entier) : entier

**Variables:** P,i:entier

**Début**

$P \leftarrow 1$

**Pour** i allant de 1 à n faire

$P \leftarrow P * a$

**FinPour**

**retourne P**

**Fin**

# L'appel d'une fonction

- Une fois la fonction est définie, il est possible de l'appeler une ou plusieurs fois dans le programme principal en utilisant son nom avec le même nombre de paramètres que dans la définition et du même type prévu.
- On peut appeler une fonction de deux manières:
  - En affichant sa valeur
  - En l'affectant à une variable

# Exemple complet $x^7 + y^9$

Algorithme somme\_puissance

Variables : x,y,R entier

Fonction puissance (a:entier, n:entier) : entier

Variables: P,i:entier

Début

$P \leftarrow 1$

Pour i allant de 1 à n faire

$P \leftarrow P * a$

FinPour

retourne P

Fin

Début

Ecrire ("Entrez la valeur de x et de y")

Lire(x,y)

$R \leftarrow$  puissance (x,7)+ puissance (y,9)

Ecrire (« le résultat est », R)

Fin

Paramètres Réels

Programme Principal



# Exemple complet $x^7 + y^9$

Algorithme somme\_puissance

Variables : x,y,R entier

Fonction puissance (a:entier, n:entier) : entier

Variables: P:entier

Début

$P \leftarrow 1$

Pour i allant de 1 à n faire

$P \leftarrow P * a$

FinPour

retourne P

Fin

Paramètres Formels

Début

Ecrire ("Entrez la valeur de x et de y")

Lire(x,y)

$R \leftarrow$  puissance (x,7)+ puissance (y,9)

Ecrire (« le résultat est », R)

Fin

# Exemple complet $x^7 + y^9$

Algorithme somme\_puissance

Variables : x,y,R entier

Fonction puissance (a:entier, n:entier) : entier

Variables: P:entier

Début

P ← 1

Pour i allant de 1 à n faire

P ← P\*a

FinPour

retourne P

Fin

Début

Ecrire ("Entrez la valeur de x et de y")

Lire(x,y)

R ← puissance (x,7)+ puissance (y,9)

Ecrire (« le résultat est », R)

Fin

Variables Globales

Variables Locales

# Exercice 1

- Ecrire une fonction qui calcule la somme des nombres paires inférieurs à un entier  $n$  saisi par l'utilisateur. Et un programme principal qui appelle la fonction avec la valeur de  $n=100$  puis la valeur  $=200$

## Exercice 2

- Ecrire une fonction pair qui retourne Vrai si le nombre passé en paramètre est pair et Faux dans le cas contraire.
- Appeler la fonction plusieurs fois dans le programme et demander à chaque fois à l'utilisateur s'il veut refaire la saisie