

# Informatique 2

## Fonctions et procédures

Etabli par : Dr KHETIR Hafid

## Problème

- Dès qu'on commence à écrire des programmes sophistiqués, il devient difficile d'avoir une vision globale sur son fonctionnement
- Difficulté de trouver des erreurs
- **Solution : décomposer le problème en sous problèmes**
  - Trouver une solution à chacun
  - La solution partielle donne lieu à un sous-programme

## Programmation procédurale

### Principe:

- Il s'agit d'écrire des programmes en utilisant des sous-programmes
- Forme générale d'un programme

Programme P

Sous-programme SP  
1

...

Sous-programme SP  
n

FinP

### Exemple

- Algorithme qui teste si M est la matrice inverse de N
- Algorithme A
  - A1: Lecture de M
  - A2: Lecture de N
  - A3: Affecter à O le résultat de  $M*N$
  - A4: Tester si O est l'identité
  - A5: Affichage du message adéquat

Fin A

## Procédures & Fonctions

- En algorithmique, on distingue deux types de sous-programmes
  - Les procédures
  - Les fonctions

### Plan

- Fonctions
  - Paramètres
  - Type retourné
- Procédures
  - Paramètres
  - Appel par variable
  - Appel par valeur

### Fonctions : structure

- Une fonction est un sous-programme qui :
  - A un nom
  - Peut avoir des paramètres
  - Qui retourne une valeur d'un certain type
  - Qui peut avoir besoin de variables
  - Qui est composé d'instructions

### Fonctions : déclaration

Fonction **nomf** (<paramètres>): type

Déclaration des variables

Début

instructions

**nomf** ← expression

Fin fonction

### Fonctions : Exemple

- Fonction qui retourne le carré d'un entier :

Fonction carré(n : entier): entier

Début

carré ← n \* n

fin fonction

## Fonction : utilisation dans un algorithme

Algorithme ex1

Variable i, j: entier

Fonction carré(n : entier): entier

Début

carré  $\leftarrow$  n \* n

fin fonction

Début

Lire (i)

Ecrire(carré(i))

j  $\leftarrow$  carré(i)

Ecrire(j)

Fin

## Fonctions : A retenir

- Une fonction retourne toujours une valeur
- Une fonction NomF contient toujours une instruction de la forme

NomF  $\leftarrow$  Expression

- Il ne faut jamais utiliser d'instructions de la forme

f(paramètres)  $\leftarrow$  expression

- En général, l'utilisation d'une fonction se fait
  - Soit par une affectation: v  $\leftarrow$  f(paramètres)

Soit dans l'écriture: Ecrire ( f(paramètres) )

## Fonctions : quelques exercices

- Ecrire une fonction qui
  - Prend un tableau de 5 entiers, puis
  - Retourne la valeur Vraie ou Faux selon que le tableau est trié par ordre croissant ou non

## Fonctions : Principe du test

- On suppose d'abord que le tableau est trié
- Ensuite on compare chaque case à sa suivante:
  - Si l'ordre n'est pas respecté alors on conclut que le tableau n'est pas trié

## Fonction : test du tri

Fonction **trié**( T: Tableau[5] d'entiers): Booléen

Variable i : entier

Variable b : booléen

Début

b ← Vrai

Pour i = 1 à 4

    Si  $T(i) > T(i+1)$  alors

        b ← Faux

    FinSi

FinPour

**trié** ← b

Fin Fonction

## Fonction : exemple

- Ecrire un algorithme qui
  - lit un tableau de 5 entiers puis
  - teste s'il est trié ou pas

## Fonction : exemple

Algorithme ex1

Variable T1 : tableau[5] d'entier

Variable i : entier

**Fonction Trié(...)**

...

**Fin fonction**

Début

    Pour i = 1 à 5

        Lire( T1(i))

    Fin Pour

    Si Trié(T1) = Vrai Alors

        Ecrire(« c'est trié »)

    FinSi

    Sinon

        Ecrire(« Non trié »)

    FinSinon

Fin

## Fonctions : en VBA (Visuel basic)

```
Function trié(t() As Integer) As Boolean
```

```
Dim b As Boolean
```

```
Dim i as Integer
```

```
b = True
```

```
For i = 1 To 4
```

```
    If t(i) > t(i + 1) Then
```

```
        b = False
```

```
    End If
```

```
Next
```

```
trié = b
```

```
End Function
```

## Procédures : définition

- Une procédure est un sous-programme qui ne retourne pas de valeur
- C'est donc un type particulier de fonction
- En général, une procédure modifie la valeur de ses paramètres
  - Je dis bien « en général », ce n'est pas toujours le cas

## Procédures : structure

- Tout comme les fonctions, une procédure est un sous-programme qui :
  - A un nom
  - Peut avoir des paramètres
  - Qui retourne une valeur d'un certain type
  - Qui peut avoir besoin de variables
  - Qui est composé d'instructions
- **Procédures : déclaration**

```
Procédure nomf (<paramètres>)
```

```
    Déclaration des variables
```

```
    Début
```

```
        instructions
```

```
    Fin procédure
```

## Procédures : exemple

- Une procédure qui ajoute 2 à un entier

procédure aug2(n : entier)

Début

$n \leftarrow n+2$

Fin Procédure

## Procédures : dans les algorithmes

- Ecrire un algorithme qui
  - Lit un entier positif n puis
  - Affiche tous les nombres impaires inférieurs à n

## Procédure : dans les algorithmes

Algorithme ex1

Variable i,n: entier

Procédure Aug2(..)

...

Fin Procédure

Début

Lire(n)

$i \leftarrow 1$

Tant que  $i \leq n$

Ecrire(i)

aug2(i)

Fin TantQue

Fin

## Procédures : A retenir

- Une procédure ne retourne pas de valeur
- Il est donc faux de l'affecter à une variable
  - Ne pas écrire :  $j \leftarrow \text{aug2}(i)$

## Procédures : en VBA

Sub aug2(n as integer)

$n = n + 2$

End sub

- Tous les programmes qu'on a écrits jusqu'à présent étaient en fait
  - Des procédures sans paramètres

## Procédures : appel en VBA(Visuel basic)

```
Sub aug2(n as integer)
    n = n + 2
End sub
Sub ex()
Dim i, n as integer
n=InputBox(« donner n »)
i = 1
While i <= n
    MsgBox(i)
    call aug2(i)
Wend
End Sub
```

## Procédures & fonctions : appels imbriqués

- Dans la définition d'une procédure, on peut faire appel à une autre procédure ou fonction déjà définie
- Même remarque pour les fonctions

## Procédures & fonctions : appels imbriqués

```
Procédure aug4(n : entier)
Début
    aug2(n)
    aug2(n)
Fin Procédure
```

## Procédures & fonctions : appels imbriqués

```
Fonction Puiss4(n : entier) : entier
Début
    Puiss4 ← Carré(Carré(n))
Fin Fonction
```

## Procédure : Appel par variable versus appel par valeur

- En général, les procédures modifient leurs paramètres.
- Ceci à cause du fait que *par défaut*, elles travaillent sur les variables elles même

- Dans certains cas, on ne veut pas que la procédure modifie ses paramètres
  - on lui précise qu'elle doit travailler sur leurs **valeurs**
  - Dans ce cas, la procédure travaille sur une **copie** des paramètres

## Procédure: mode d'appel

- Exemple : on veut écrire un algorithme qui
  - saisit un tableau d'entiers puis
  - affiche ses éléments dans l'ordre croissant

## Procédure : mode d'appel

- Idée :
  - On lit le tableau T
  - On fait appel à une procédure qui trie T
  - On parcourt ensuite les éléments du premier jusqu'au dernier
- Il ne faut par contre pas que le tableau soit trié définitivement
- La procédure doit donc travailler sur une « copie » non pas sur le tableau lui même

## Procédure : Appel par valeur

- Si on veut qu'un des paramètres ne soit pas modifié par la procédure, il faut le faire précéder par le terme
  - Val
- Exemple:

**Procédure TrierEtAfficher(val t:tableau[5] d'entiers)**

...

**Fin procédure**

## Procédures : appel par valeur en VBA

- Il faut faire précéder les paramètres par le mot réservé
  - **ByVal**

**Sub ab(ByVal i as integer)**

**i=Sqr(i)**

**MsgBox(i)**

**End Sub**

## Procédures & fonctions: exemple complet

- Reprendre l'algorithme de tri et le développer cette fois-ci en utilisant des fonctions et des procédures
- Procédure de saisie du tableau
- Fonction qui retourne l'indice de la valeur max dans une partie du tableau
- Procédure qui échange les valeurs de deux cases

- Procédure qui fait le tri en utilisant la fonction et les 2 procédures ci-dessus

## Procédure de saisie

Procédure saisir(t:tableau[5] d'entiers)

variable i: entier

Début

Pour i = 1 à 5

Lire(t(i))

Fin Pour

Fin Procédure

## Fonction qui retourne l'indice de la valeur max dans une partie du tableau

Fonction IndMax (t: tableau[5] d'entiers,

i: entier)

variable j, Max: entier

Début

Max  $\leftarrow$  i

Pour j = i à 5

Si  $t(\text{Max}) < t(j)$  Alors

Max  $\leftarrow$  j

FinSi

Fin Pour

IndMax  $\leftarrow$  Max

Fin Fonction

## Procédure qui échange les cases i et j

Procédure échanger(i, j : entier, t: tableau[5] d'entiers)

Variable Z : entier

Début

Z  $\leftarrow$  t(i)

t(i)  $\leftarrow$  t(j)

t(j)  $\leftarrow$  Z

Fin Procédure

## Procédure de Tri

Procédure Trier(T: Tableau[5] d'entiers)

Variable i: entier

Pour i = 1 à 4

Echanger(i, IndMax(i,T), T)

FinPour

FinProcédure

## Algorithme de saisie et de tri

Algorithme ex

Variable T: Tableau[5] d'entiers

Début

    Saisir(T)

    Trier(T)

Fin