

Algorithmique

Mr. FAREH Abdelhak

Site personnel : <http://www.fareh-abdelhak.com>

Courriel : fareh.abdelhak@yahoo.fr

Les procédures & les fonctions

- Un programme écrit en un seul bloc devient difficile à comprendre dès qu'il dépasse un certain nombre de lignes.
- Pour éviter un tel problème, il est vivement conseillé d'utiliser des sous-programmes.
- En effet, l'algorithmique dispose de deux types de déclarations qui sont : les **procédures** et les **fonctions**.
- Une décomposition du programme en modules élémentaires facilite la tâche.

Les procédures & les fonctions

- Ce processus pouvant être répété autant de fois que nécessaire, à chaque tâche (module) est associée une procédure ou une fonction.
- On utilise la notion de sous-programme pour les raisons suivantes :
 - Gain de temps et d'espace mémoire.
 - Lisibilité des programmes.
 - Réutilisabilité des sous-programmes sans avoir à les redéclarer.

Les procédures & les fonctions

- Facilité de la maintenance ou de la modification des programmes.
- **Qu'est ce qu'une procédure ?**
- Une procédure est un sous-programme, qui peut être appelé dans un autre programme ou dans différents lieux du même programme.
- Une procédure peut être appelée comme une instruction dans un programme par l'intermédiaire de son nom.

Les procédures & les fonctions

- **Déclaration d'une procédure :**
- Une procédure a la même structure qu'un programme, elle est définie dans la partie déclarative du programme, et est constituée d'un en-tête, des déclarations de variables si elles existent et d'un corps (partie traitement).

En-tête

Déclarations

Corps

Les procédures & les fonctions

Procédure <nom de la procédure> (liste de paramètres) ;

Var <déclaration de variables>

Début

<corps de la procédure>

Fin;



Non obligatoires



Le ';' termine la procédure

Les procédures & les fonctions

- **Exemple :**
- Soit l'algorithme qui fait le tri d'un tableau contenant une liste de nombres réels.

```
Algorithme tri ;  
  
Type vect : tableau [1..10] de réels ;  
  
Var V : vect ;  
    i, j : entier ;  
    x : réel ;  
  
Début  
    i ← 1 ;
```

Les procédures & les fonctions

Présentation de
l'algorithme en un
seul bloc

Tant que $i \leq 9$ faire

Début

$j \leftarrow i + 1;$

Tant que $j \leq 10$ faire

Début

Si $V[i] > V[j]$ alors

Début

$x \leftarrow V[i];$

$V[i] \leftarrow V[j];$

$V[j] \leftarrow x;$

Fin ;

$j \leftarrow j + 1;$

Fin ;

$i \leftarrow i + 1;$

Fin ;

Fin.

Permutation

Les procédures & les fonctions

- En utilisant la notion de sous-programme, l'algorithme précédent se présente de la façon suivante :

```
Algorithme tri ;  
Type vect : tableau [1..10] de réels ;  
Var V : vect ;  
      i, j : entier ;  
Procédure permut (var U : vect ; k, l : entier) ;  
Var x : réel ;  
Début  
    x ← U[k] ;  
    U[k] ← U[l] ;  
    U[l] ← x ;  
Fin ;
```

Algorithme utilisant
une procédure

Définition de la
procédure

Les procédures & les fonctions

Début

$i \leftarrow 1;$

Tant que $i \leq 9$ **faire**

Début

$j \leftarrow i + 1;$

Tant que $j \leq 10$ **faire**

Début

Si $V[i] > V[j]$ **alors**

permut (V, i, j);

$j \leftarrow j + 1;$

Fin ;

$i \leftarrow i + 1;$

Fin ;

Fin.

Programme principal

Appel de la procédure

Les procédures & les fonctions

- Qu'est ce qu'une fonction ?
- Une fonction est un cas particulier de procédures, contrairement à celle-ci, la fonction doit avoir un **type**, c'est-à-dire doit remettre une valeur en sortie.

Les procédures & les fonctions

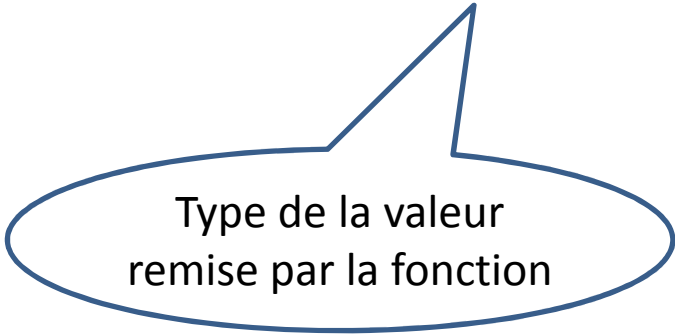
Fonction <nom de la fonction> (paramètres) : <Type> ;

Var <déclarations>

Début

<corps de la fonction>

Fin;



Type de la valeur
remise par la fonction