

## Chapitre 6 : Fonctions et Procédures

### Problématique :

Ecrire un programme qui permet de remplir un tableau de n cellules d'entiers. Le programme affiche le tableau avant et après décalage cyclique d'un rang vers la droite.

### Solution :

Pour résoudre ce problème, il faudra :

- ✚ Remplir le tableau
- ✚ Afficher le tableau
- ✚ Décaler les valeurs
- ✚ Afficher à nouveau le tableau

On remarque que l'affichage du tableau revient deux fois : ce qui est à éviter. Pour pallier à cela, on utilise les sous-programmes.

### Introduction

Un programme est un ensemble d'instructions afin de résoudre un problème. Dès fois, plusieurs instructions sont nécessaires pour effectuer une seule action. En général, il est préférable de regrouper ces instructions dans un bloc : module ou sous-programme.

Les sous-programmes vont nous permettre entre autres de ne pas se répéter dans le code, de l'organiser, d'avoir une meilleure lisibilité... . Nous avons deux types de modules ou sous-programmes : **FONCTION** et **PROCEDURE**.

## I. Fonctions

### 1. Définition

Un sous-programme est une fonction. Une fonction est sous-programme qui doit obligatoirement retourner un et un seul résultat.

### 2. Syntaxe

*Type\_de\_retour nomFonction([type1 param1,..., typen paramn])*

*{*

*//Corps de la fonction*

*return expression ;*

*}*

**NB : Le type de *expression* doit être toujours identique au type de retour.**

### 3. Prototype

Il représente l'entête de la fonction.

```
Type_de_retour nomFonction (paramètres) ;
```

#### Remarque

- Les noms des variables des paramètres sont facultatifs pour les prototypes.
- Il est impossible de regrouper les paramètres par type

4. Corps de la fonction :

```
Type_de_retour nom_Fonction (paramètres)
{
    Instructions
}
```

#### Remarque

1. Les noms de variables sont obligatoires lors de l'implémentation
2. Il existe deux types de variables avec les sous-programmes :
  - ✚ Les variables globales : elles sont déclarées en dehors de toutes les fonctions et généralement en haut
  - ✚ Les variables locales : elles sont déclarées au sein des fonctions. Elles ont une portée

## II. Procédure

C'est une fonction qui ne retourne rien.

### Remarque :

- + Dans une fonction, on doit obligatoirement avoir le mot ***return***.
- + Une procédure commence toujours par ***void***.
- + Pour appeler une procédure, il faut écrire le nom de la procédure et les paramètres réels s'ils existent.
- + Pour appeler une fonction, il faut le faire soit :
  - ✓ Dans une expression d'affichage
  - ✓ Dans une expression d'affectation
  - ✓ Dans une expression de calcul
  - ✓ Dans une condition