**Algorithmique (suite)**

Table des matières

[4- Procédures et fonctions 2](#_Toc91615674)

[1-Fonctions : 2](#_Toc91615675)

[2- Procédures : 2](#_Toc91615676)

[5-Structure de données : 3](#_Toc91615677)

[1- Les tableaux : 3](#_Toc91615678)

[2- Les structures : 4](#_Toc91615679)

[3- Les structures et tableau : 5](#_Toc91615680)

# 

**Algorithmique**

# 4- Procédures et fonctions

Les procédures et les fonctions servent à rendre l’algorithme plus lisible, ils permettent de diviser ce dernier pour mieux répondre au problème donné.

On fait appel à un bloc compétent pour chaque procédure.

C’est un pseudo algorithme qui a un début et une fin avec une fonction limitée.

## 1-Fonctions :

La fonction est appelée pour remplir une tâche et renvoyer un résultat.

Est une partie de l’algorithme qui a :

* Une finalité précise
* Un nom
* Elle peut communiquer avec l’extérieur ( hors bloc de la fonction) par l’intermédiaire de paramètres typés
* Comprend des variables locale (utilisée dans le corps de la fonction)
* Elle retourne une valeur
* **La déclaration d’une fonction :**

Dans la déclaration de la fonction on doit donner un nom à la fonction, déclarer ses variables et leurs types (réel, entier, …), et le type du résultat à retourner (entier, réel, booléen…)

* **Appel de la fonction :**

Il faut distinguer la différence entre déclarer la fonction et appeler la fonction, la fonction est exécutée au moment de l’appeler et non pas au moment de la déclarer.

**Exemple :**

**Fonction** **somme** (x : entier, y : entier) **: entier** ;

**Var** : somme : entier

**Début**

Ecrire (**"**calcul de la somme en cours… **"**)

**somme** = x + y ;

**Retourner** **somme**

**FinFonction**

* Cette **fonction** prend x et y comme paramètres calcule leur somme et retourne le résultat
* **Somme** est une variable

## 2- Procédures :

C’est une fonction qui ne renvoie pas de résultat (comme programmer l’envoie de mails aux clients) c’est-à-dire elle ne possède pas d’instruction retourne.

Exemple : Procédure somme prend en paramètre 2 entiers x et y calcule leur somme et affiche le résultat sur la console.

**Procedure** **somme** (x : entier, y : entier) **: vide** ;

**Var** : somme : entier

**Début**

Ecrire (**"**calcul de la somme en cours… **"**)

**somme** = x + y ;

écrire (**"La somme des deux nombres est : "  somme)**

**FInProcédure**

# 5-Structure de données :

Une structure de donnée complexe sert à traiter une masse importante de données, qui peut aussi appelée une collection de données (un ensemble d’éléments)

## 1- Les tableaux :

* Quand les données sont nombreuses et la déclaration de toutes les variables devient quasiment impossible, les données seront mieux représentées sous forme de tableaux mais les données sont du même type
* Le type tableau définie le type commun de tous les éléments de ce tableau
* Syntaxe : tabName : TYPE (exemple : note : entier [10] : est une variable de type tableau d’entiers de taille 10

On peut distinguer plusieurs types de tableau selon leur dimension :

* **Tableau à une dimension :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| indexe | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| valeur | 45 | 54 | 1 | -56 | 22 | 134 | 49 | 12 | 90 |

Indexe définit la position, il permet de différencier les éléments du tableau.

-Ce tableau est de longueur 9 car il contient 9 emplacements.

-Chaque nombre du tableau (les 9) est repéré par son rang appelé indice.

-Pour accéder à un élément du tableau il faut mettre entre crochets l’indice de la case qui contient cet élément (pour accéder au 5 ème élément (22), on écrit :Tab [4]

Exemple : réinitialiser toutes les valeurs du tableau précédent à 0 :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| valeur | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Pour se faire on applique la boucle suivante :

AlgoInitTableau

DEBUT

Pour i = de 0 à n-1 faire

Tab[i] = 0

FinPour

FIN

* **Les tableaux à deux dimensions appelés aussi une matrice**

Les données sont de masse mais de même nature et sont représentées selon deux critères, elles sont rangées dans un tableau à deux entrées.

Les tableaux peuvent avoir plusieurs dimensions (n).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T[0, 0] | T[0, 1] | T[0, 2] | T[0, 3] | T[0, 4] |
| T[1, 0] | T[1, 1] | T[1, 2] | T[1, 3] | T[1, 4] |
| T[2, 0] | T[2, 1] | T[2, 2]15 | T[2, 3] | T[2, 4] |

On écrit Tab [2, 2] =

Ce tableau contient 5 colonnes et 3 lignes.

## 2- Les structures :

Une structure de donnée est un ensemble de champs (alphanumérique) qui ont un point en commun. Elles constituent un produit composé de plusieurs types définissant un nouveau type, ce qui peut amener à avoir des variables ayant comme type ce nouveau type.

Toutes les variables de la structure auront la même représentation avec des valeurs différentes.

**Exemples :**

**Struct Person** {

Nom : car (20) ;

Prenom : (20) ;

Age : Entier ;

20 représente le nombre de personnes avec 20 noms et 20 prénoms (chaque personne a un nom et un prénom

**Déclaration de variable :**

P1 : Person ;

P2 : Person ;

P3 : Person ;

**Utilisation :**

P1.Nom = "MOUCHENE"

P1.Prenom = "Djamel"

P1.Age = 25

## 3- Les structures et tableau :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | SEBIH | | Salim | | 34 | | |  | | --- | | ADDA | | Fatima | | 25 | | |  | | --- | | BIZANDRY | | Fortuna | | 22 | |
|  |  |  |

Struct Person{

Nom : car(20) ;

Prenom : car (20)

Age : Entier ;

}

Persons : Person[20] ;

P1 : Person ;

P1.nom = **"SEBIH"**

P1.prenom=**"Salim" ;**

P1.age= 34

Persons : Person[20] ;

P2 : Person ;

P2.nom = **"ADDA"**

P2.prenom=**"FATIMA" ;**

P2.age= 25

Persons [0] = p1 persons[1] = p2

0 représente le rang des trois informations (nom, prenom et age) de la personne N° 1 dans le