

LES FONCTIONS

Les fonctions

1- Introduction

Les fonctions sont utilisées pour décrire les différentes étapes d'un algorithme complexe. On peut alors appeler ces fonctions pour réaliser les étapes de calcul correspondantes : lors de l'appel, on indique grâce aux paramètres les données sur lesquelles la fonction doit travailler, puis on récupère le résultat retourné.

Beaucoup de langages distinguent deux sortes de sous-programmes : les fonctions et les procédures. l'appel d'une fonction retourne un résultat, alors que l'appel d'une procédure ne retourne rien.

Les fonctions

2- Déclaration des fonctions

Une fonction se définit par la construction :

```
fonction NomFonction(NomArg1 :TypeArg1, ...) : TypeRetour  
    déclarations des variables  
    début  
        Bloc d'instructions  
    fin
```

Les fonctions

Exemple :

fonction somme(x : entier, y : entier) : entier

debut

 retourne(x+y)

fin

Les fonctions

Remarque:

Une procédure est une fonction qui ne retourne rien:

```
procédure  nom_procedure(type parametre1,...)
  debut
    Instructions
  fin
```

Les fonctions

Exercice d'application :

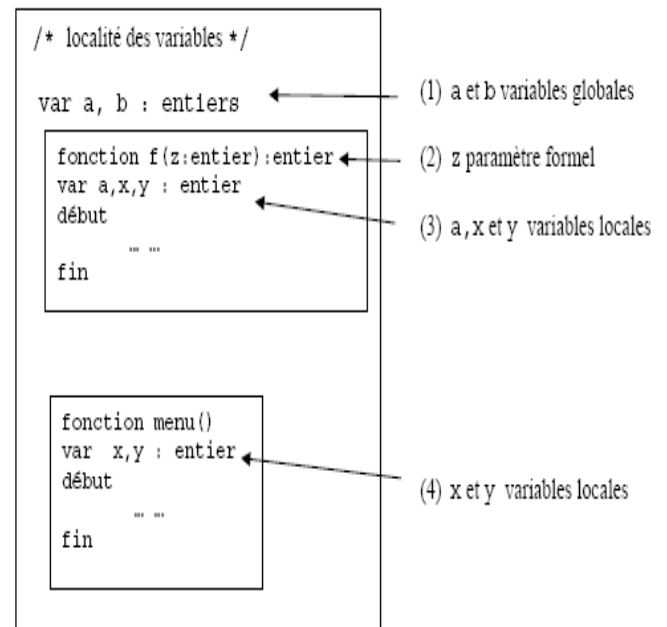
Ecrire une fonction qui prend pour argument un entier n et retourne la somme des n premier entier.

Les fonctions

3- Variables locales et globales

Il peut y avoir différents niveaux de déclaration de variables.

- Les variables définies dans le programme hors des fonctions sont dites **variables globales**. Les variables globales sont connues dans tout le programme.
- Les variables déclarées dans le corps d'une fonction sont dites **variables locales**. Les variables locales sont connues uniquement à l'intérieur de la fonction, c'est-à-dire qu'elles sont propres à la fonction et invisibles de l'extérieur.



Les fonctions

4- Fonctions récursives

Il arrive, en mathématique, que des suites soient définies de la manière suivante :

$u_0 = \text{constante}$

$u_n = f(u_{n-1})$

Exemple :

La suite factorielle : $n! = n \cdot (n-1)!$, pour $n \geq 1$ avec $0! = 1$, peut s'écrire :

$f(0) = 1$

$f(n) = n \cdot f(n-1)$

Ce que l'on peut traduire par :

$f(n) = (\text{si } n=0 \text{ alors } 1 \text{ sinon } n \cdot f(n-1)).$

Cela peut se traduire en algorithmique par :

fonction fact(n :entier) : entier

début

si (n=0) alors

retourne(1)

sinon

retourne(n*fact(n-1))

finsi

fin

Exercices
