**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 SP1

…

Sous-programme SPn

FinP

**Procédures & Fonctions**

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

**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 🡨 n \* n**

**fin fonction**

**Début**

**Lire (i)**

**Ecrire(carré(i))**

**Fin**

**Fonctions: A retenir**

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

NomF 🡨 Expression

* Il ne faut jamais utiliser d’instructions de la forme

f(paramètres) 🡨 expression

* En général, l’utilisation d’une fonction se fait
  + Soit par une affectation: v 🡨 f(paramètres)
  + Soit dans l’écriture: Ecrire ( f(paramètres) )

**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 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 🡨 n+2

Fin Procédure

**Procédures : dans les algorithmes**

* Ecrire un algorithme qui
  + Lit un entier positif n puis
  + Affiche tous les nombres impairs inférieurs à n

Algorithme ex1

Variable i,n: entier

Procédure Aug2(..)

…

Fin Procédure

Début

Lire(n)

i 🡨 1

Tant que i ≤ 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 🡨 aug2(i)