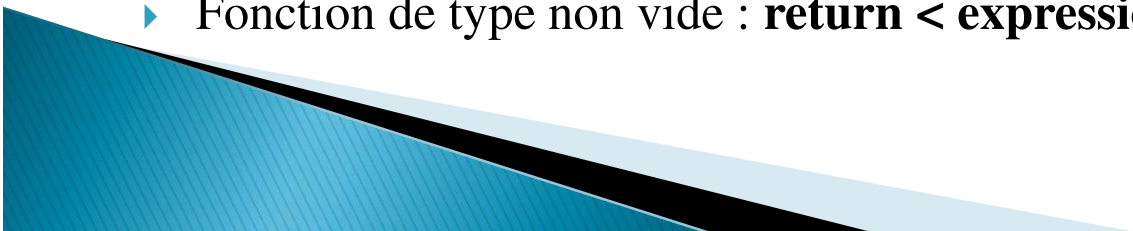


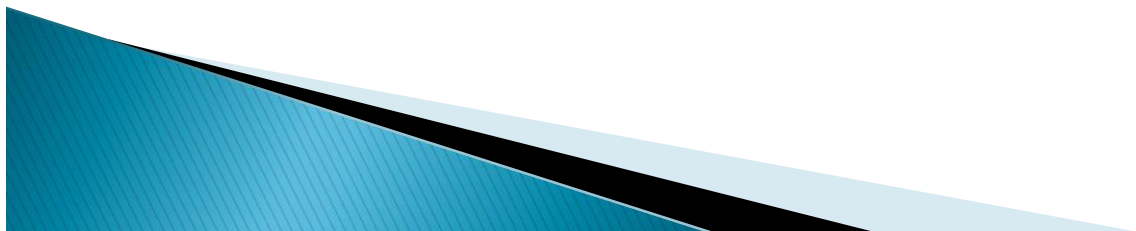
# Les Fonctions

- ▶ un sous-programme qui effectue un traitement puis renvoie ou non un résultat
- ▶ définitions :
  - **< type > < identificateur > (< déclaration-ident > ,..., < déclaration-ident >)**  
**instruction-bloc**
  - **< type > < identificateur > (void)**  
**instruction-bloc**
  - **void < identificateur > (< déclaration-ident > ,..., < déclaration-ident >)**  
**instruction-bloc**
  - **void < identificateur > (void)**  
**instruction-bloc**
- ▶ Fonction de type non vide : **return < expression > ;**



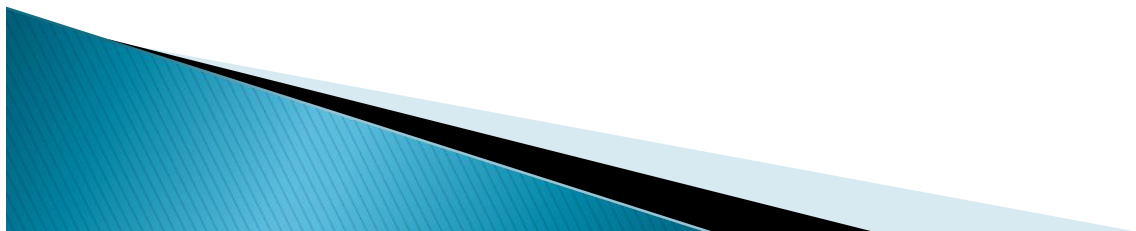
# Appel de Fonctions

- ▶ Appel de fonction :
  - **Syntaxe :**
    - **< identificateur > (< argument >, ..., < argument >)**
    - **< variable > = < identificateur > (< argument >, ..., < argument >)**
- ▶ Les types des arguments doivent être compatibles avec ceux des paramètres formels
- ▶ Par défaut, toute fonction peut en appeler une autre
- ▶ A chaque appel, le type de chaque argument est converti au type du paramètre formel correspondant.



# Déclaration de fonction

- ▶ Si le compilateur rencontre une référence à une fonction dont il connaît pas encore la définition : il lui attribue le type int
- ▶ Déclaration de la fonction :
  - définition
  - son prototype :
    - `< type > < identificateur > (< déclaration-ident >, ..., < déclaration-ident >);`



# Passage de Paramètres

- ▶ La transmission des paramètres se fait par valeur
- ▶ Pas de modifications des arguments
- ▶ Exception : les tableaux

