

# Conversions

entre types « compatibles »

Implicite

char → short → long

'\7' → 07 → 07L

float → double

Explicite  
(cast)

op\_conversion

expression →

op\_conversion





# Conversions

## Règle de conversion implicite



- ◆ Convertir les éléments de la partie droite d'une expression d'affectation dans le type de la variable ou de la constante le plus riche
- ◆ Faire les opérations de calcul dans ce type
- ◆ Puis convertir le résultat dans le type de la variable affectée (partie gauche de l'affectation)

# Conversions

## notion de richesse d'un type

1. si l'un des deux opérandes est du type **long double** alors le calcul doit être fait dans le type **long double**
2. sinon, si l'un des deux opérandes est du type **double** alors le calcul doit être fait dans le type **double**
3. sinon, si l'un des deux opérandes est du type **float** alors le calcul doit être fait dans le type **float**
4. sinon, appliquer la règle de promotion en entier, puis :
  1. si l'un des deux opérandes est du type **unsigned long int** alors le calcul doit être fait dans ce type
  2. si l'un des deux opérandes est du type **long int** alors le calcul doit être fait dans le type **long int**
  3. si l'un des deux opérandes est du type **unsigned int** alors le calcul doit être fait dans le type **unsigned int**
  4. si l'un des deux opérandes est du type **int** alors le calcul doit être fait dans le type **int**