Langage C

4.2. Les opérateurs arithmétiques : + - * / % (modulo)

Les opérateurs arithmétiques : + - * / % (modulo)

Ils s'appliquent à tous les types numériques (*int, double*...), à l'exception de l'opérateur modulo qui ne concerne que les entiers.

 **Un opérateur ne fournit pas le même résultat s'il est appliqué à des entiers ou à des réels !** C'est en particulier le piège des divisions entière/réelle avec l'opérateur quotient.

Opérateur	Rôle	Exemples
+	addition	2+3 vaut 5
-	soustraction	2-3 vaut -1
*	produit	2*3 vaut 6
/	quotient **(entier ou réel !)**	11./2. vaut 5.5 (division réelle) mais 7/3 vaut 2 (division entière) !
%	**modulo** (reste de la division entière)	11%3 vaut 2 24%8 vaut 0 *Très utile en informatique !*

Les opérateurs mathématiques

A ces opérateurs binaires, il convient d'ajouter les deux opérateurs unaires (un seul opérande) qui sont l'opposé - et l'identité +.

La division entière fournit **deux** résultats : le quotient (opérateur /) et le reste (opérateur modulo, % en Langage C). Tous deux sont très utilisés en informatique…

Attention : les opérateurs binaires, c'est-à-dire agissant sur deux opérandes, ne sont a priori définis que pour des opérandes de même type et ils fournissent un résultat de ce type. Par exemple, 5./2. est le quotient de deux valeurs de type *double* et l'opérateur quotient / fournit le résultat 2.5 de type *double*. Par contre, 5/2 est le quotient de deux entiers et le résultat est l'entier 2 !

Cet exemple a priori évident peut avoir des effets surprenants : une mise à l'échelle par une simple règle de trois (formule du style n/NMAX*100) fournit un résultat presque toujours nul si elle est effectuée sur des opérandes entiers sans précautions ! La solution est un *cast* (conversion explicite).

Quand les deux opérandes ne sont pas du même type, une opération de conversion implicite est mise en oeuvre par le compilateur afin que le calcul soit fait dans le **type dominant**. La hiérarchie des types est :

char < short int < long int < float < double

Quand plusieurs opérateurs apparaissent dans une même expression, les règles traditionnelles de **priorité** de l'algèbre s'appliquent (voir tableau des priorités) : d'abord les opérateurs unaires + et -, puis les opérateurs *, /, et %, puis enfin les opérateurs binaires + et -. Des parenthèses permettent de s'affranchir des priorités.

Exemple 8, Opérateurs mathématiques :

res = 5+9/4	9/4 qui vaut 2 est ajouté à 5 → res vaut 7 au final.
res = (5+9)/4	5+9 qui vaut 14 est divisé par 4 (division entière) → res vaut 3 au final.
res = (5+9.)/4	5+9. qui vaut 14. (conversion implicite en *double* de 5 et résultat *double*) est divisé par 4 (division réelle) → res vaut 3.5 au final.
res = 4*2+9%4	4*2 qui vaut 8 est ajouté à 9%4 qui vaut 1 → res vaut 9 au final.
i = (i+1)%10	permet d'incrémenter i « modulo 10 » : i prend successivement les valeurs 0, 1, 2,...8, 9, 0, 1... Très utile pour effectuer automatiquement la remise à zéro de i quand il arrive à sa valeur maximale.

« Précédent | Suivant »