

# INF155 SÉANCE 2 - RAPPELS

Anis Boubaker, Ph.D.  
Maître d'enseignement  
École de Technologie Supérieure



# OPÉRATEURS MATHÉMATIQUES

- Le résultat d'une expression comportant un opérateur mathématique est une valeur numérique.

Opérateur	Nom ou signification	Exemple	Résultat
*	Opérateur de multiplication	$x * y$	La multiplication de x et y
/	Opérateur de division	$x / y$	Le quotient de x et y
%	Opérateur de reste de la division entière	$x \% y$	Reste de la division entière de x par y
+	Opérateur d'addition	$x + y$	L'addition de x et y
-	Opérateur de soustraction	$x - y$	La soustraction de x et y
++	Opérateur d'incrémenter post-fixe,	$x ++$	La valeur de x est d'abord utilisée telle quelle, puis incrémentée
--	Opérateur de décrémenter post-fixe,	$x --$	La valeur de x est d'abord utilisée telle quelle, puis décrémentée
++	Opérateur d'incrémenter préfixe,	$++ x$	La valeur de x est incrémentée puis alors utilisée.
--	Opérateur de décrémenter préfixe,	$-- x$	La valeur de x est décrémentée puis alors utilisée.

# OPÉRATEURS DE COMPARAISON

- Le résultat d'une expression comportant un opérateur de comparaison est une valeur booléenne (0 ou 1)

Opérateur	Nom ou signification	Exemple	Résultat
<	Opérateur d'infériorité	$x < y$	Vrai si x est inférieur à y
>	Opérateur de supériorité	$x > y$	Vrai si x est supérieur à y
<=	Opérateur d'infériorité ou d'égalité	$x <= y$	Vrai si x est inférieur ou égal à y
>=	Opérateur de supériorité ou d'égalité	$x >= y$	Vrai si x est supérieur ou égal à y
==	Opérateur d'égalité	$x == y$	Vrai si x est égal à y
!=	Opérateur d'inégalité	$x != y$	Vrai si x est différent de y

- **En C:**
  - 0 c'est FAUX
  - Toute autre valeur, c'est VRAI

# OPÉRATEURS LOGIQUES

!	Opérateur NOT logique	!x	Vrais si x est égal à faux. Dans tous les autres cas, retourne faux
&&	Opérateur AND logique	x && y	Vrais si x et y sont vrai
	Opérateur OR logique	x    y	Vrai si x et/ou y sont vrai

# OPÉRATEURS BINAIRES

~	Opérateur de NOT binaire	~x	1 si le bit de x est 0
&	Opérateur AND binaire	x & y	Opération AND bit à bit
^	Opérateur XOR binaire	x ^ y	Opération XOR bit à bit
	Opérateur OR binaire.	x   y	Opération OR bit à bit

# OPÉRATEURS BINAIRES

- Les résultat d'une expression comportant un opérateur binaire est une valeur numérique.
- Les opérateurs binaires effectuent un traitement bit-à-bit sur les opérandes.

~	Opérateur de NOT binaire	~x	1 si le bit de x est 0
&	Opérateur AND binaire	x & y	Opération AND bit à bit
^	Opérateur XOR binaire	x ^ y	Opération XOR bit à bit
	Opérateur OR binaire.	x   y	Opération OR bit à bit

- Ainsi que le opérateurs de décalage binaire (explication en classe)
  - >> : décalage vers la droite
  - << : décalage vers la gauche

# OPÉRATEURS D'AFFECTATION

Opérateur	Nom ou signification	Exemple	Résultat
=	Opérateur d'affectation	$x = y$	Assigne la valeur de $y$ à $x$
*=	Opérateur de multiplication et d'affectation	$x *= y$	Équivalent de: $x = x * y$
/=	Opérateur de division et d'affectation	$x /= y$	Équivalent de: $x = x / y$
%=	Opérateur de modulo et d'affectation	$x %= y$	Équivalent de: $x = x \% y$
+=	Opérateur d'addition et d'affectation	$x += y$	Équivalent de: $x = x + y$
-=	Opérateur de soustraction et d'affectation	$x -= y$	Équivalent de: $x = x - y$
<<=	Opérateur de décalage à gauche et d'affectation	$x <<= y$	Équivalent de: $x = x << y$
>>=	Opérateur de décalage à droite et d'affectation	$x >>= y$	Équivalent de: $x = x >> y$
&=	Opérateur de et binaire et d'affectation	$x \&= y$	Équivalent de: $x = x \& y$
=	Opérateur de ou inclusif binaire et d'affectation	$x  = y$	Équivalent de: $x = x   y$
^=	Opérateur de ou exclusif binaire et d'affectation	$x ^= y$	Équivalent de: $x = x \wedge y$

# PRIORITÉ DES OPÉRATEURS

priorité	Opérateur	Associativité
16	() [] -> . ++ <sup>1</sup> -- <sup>2</sup>	G
15	! ~ ++ <sup>3</sup> -- <sup>4</sup> - <sup>5</sup> + <sup>6</sup> * <sup>7</sup> & <sup>8</sup> sizeof	D
14	<i>conversion</i>	D
13	* <sup>9</sup> / %	G
12	+ -	G
11	<< >>	G
10	< <= > >=	G
9	== !=	G
8	& <sup>10</sup>	G
7	^	G
6		G
5	&&	G
4		G
3	?: (opérateur ternaire)	D
2	= += -= *= /= %= >>= <<= = &= ^=  =	D
1	,	G

- 
- 1 postfixé
  - 2 postfixé
  - 3 préfixé
  - 4 préfixé
  - 5 unaire
  - 6 unaire
  - 7 indirection
  - 8 adresse de
  - 9 multiplication
  - 10 et bit bit

**Source:**  
*matthieu-moy.fr*

# ERREURS FRÉQUENTES ET DIFFICILES À DÉTECTER !!



- Ne pas mélanger les opérateurs binaires et les opérateurs logiques:

$$a \&\& b \neq a \& b$$

- Ne pas mélanger opérateur d'égalité et opérateur d'affectation

$$a == b \neq a = b$$

- L'opérateur de négation logique, c'est ! **et non** ~.

- L'opérateur  $\geq$  **ne s'écrit pas**  $\Rightarrow$

- Ne pas mélanger ou-exclusif binaire avec la puissance (l'opérateur puissance n'existe pas en C, c'est la fonction *pow*)

$$a \wedge b \neq a^b$$



# IF...ELSE IF...ELSE...

## Syntaxe:

```
if( condition1 )
{
    /*
    Instructions à exécuter si condition1 est vraie
    */
}
else if( condition 2)
{
    /*
    Instructions à exécuter si condition1 est fausse
    ET condition2 est vraie
    */
}
else
{
    /*
    Instructions à exécuter si condition1 est fausse
    ET condition2 est fausse
    */
}
```

# BOUCLE WHILE

- Répète un traitement, tant qu'une condition est vérifiée:

## Syntaxe:

```
while( condition )  
{  
    //Traitement répétée tant que  
    //la condition est vraie  
}
```

} Corps de la boucle

- Dès que l'instruction **while** est rencontrée, la condition est évaluée.
- Si la condition est vraie ( $\neq 0$ ), le corps de la boucle est exécuté. Sinon la boucle est sautée (on passe à l'instruction suivante);
- Une fois que le corps de la boucle s'est **terminé**, on revient à l'étape précédente.