L'opérateur virgule

- Une expression peut être constituée d'une suite d'expressions séparées par des virgules :

  expression1, expression2, ..., expressionN

- Cette expression est alors évaluée **de gauche à droite**. Sa valeur sera la valeur de l'expression de droite. Le programme suivant:

```
main()
{
    int a, b;
    b = ((a = 3), (a + 2));
    printf(''\n b = %d \n'',b);
}
```

  Affiche à l'écran : b = 5

- La virgule séparant les arguments d'une fonction ou les déclarations de variables ne représente pas l'opérateur virgule.

Classes d'opérateurs

les opérateurs **unaires** : qui précèdent un identificateur, une expression ou une constante. Exp: -x

- les opérateurs **binaires** : qui mettent en relation 2 termes ou expressions. Exp: a+b
- les opérateurs **ternaires** : qui mettent en relation 3 termes . Exp : « ? : ». Exp:

  printf("%s", 1==2 ? "1 est égal à 2" : "1 est différent de 2") ;

Opérateur ternaire (ou **affectation conditionnelle**

- L'expression ***(expr1) ? (expr2) : (expr3)*** renvoie la valeur de l'expression **expr2** si l'expression **expr1** est vraie, et l'expression **expr3**si l'expression **expr1** est fausse.

Opérateur ternaire (ou **affectation conditionnelle**

```
/* Renvoyer la valeur max de a et b */

max(a, b)
int a;
int b;
{
  int m;

  if (a > b)
    m = a;
  else
    m = b;
  return m;
}
```

```
/* Renvoyer la valeur max de a et b */

max(a, b)
int a;
int b;
{
   return (a > b) ? a : b;
}
```

Priorité des opérateurs(1)

Exemple

- Soit l'instruction: A=5, B=10, C=1;
- Pour calculer la valeur de X après l'instruction:

X = 2*A+3*B+4*C;

L'ordinateur évalue d'abord les multiplications:

2*A ==> 10 , 3*B ==> 30 , 4*C ==> 4

Ensuite, il fait l'addition des trois résultats obtenus:

10+30+4 ==> 44

A la fin, il affecte le résultat général à la variable:

X = 44

Priorité des opérateurs(2)

- Si on veut forcer l'ordinateur à commencer par un opérateur avec une priorité plus faible, nous devons (comme en mathématiques) entourer le terme en question par des *parenthèses.*

- ***Exemple***

  Dans l'instruction: **X = 2*(A+3)*B+4*C;**

  (En reprenant les valeurs du dernier exemple, le résultat sera 164)

Priorité des opérateurs(3)

Classes de priorités

Priorité 1 (la plus forte):	**( )**
Priorité 2:	**! ++ --**
Priorité 3:	*** / %**
Priorité 4:	**+ -**
Priorité 5:	**< <= > >=**
Priorité 6:	**== !=**
Priorité 7:	**&&**
Priorité 8:	**||**
Priorité 9 (la plus faible):	**= += -= *= /= %=**

Priorité des opérateurs(4)

- ***Evaluation d'opérateurs de la même classe***

  Dans chaque classe de priorité, les opérateurs ont la même priorité. Si nous avons une suite d'opérateurs binaires de la même classe, l'évaluation se fait en passant *de la gauche vers la droite* dans l'expression.
- Pour les opérateurs unaires (**!,++,--**) et pour les opérateurs d'affectation (**=,+=,-=,*=,/=,%=**), l'évaluation se fait *de droite à gauche* dans l'expression.

Priorité des opérateurs(5)

- ***Exemples***

  L'expression **10+20+30-40+50-60** sera évaluée comme suit:

  10+20 ==> 30 - **30+30 ==> 60 - 60-40 ==> 20** - **20+50 ==> 70** - **70-60 ==> 10**

  Pour A=3 et B=4, l'expression **A *= B += 5** sera évaluée comme suit: B=9 puis A=27