

Liste des sujets à revoyer

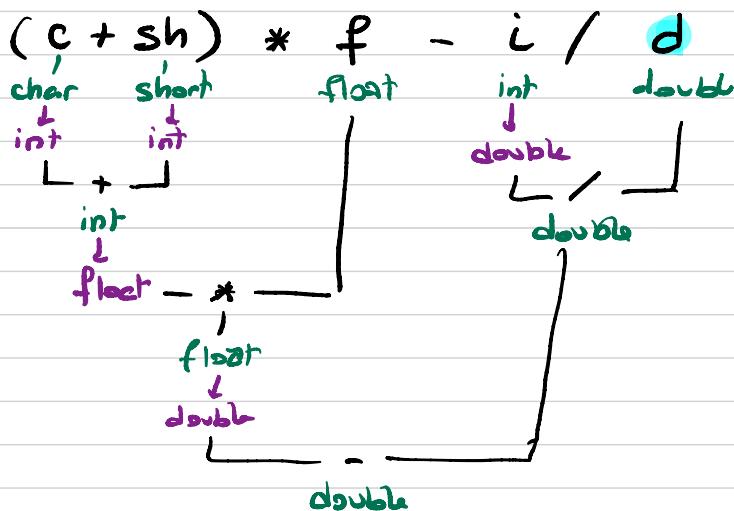
- Changement de types lors des opérations  
(promotion numérique)
- identificateurs des variables (règles)
- Passage par adresse (fonction)
- Conversion de bases numériques
- Quelques opérations mathématiques bâties
- Type en fonction du valeur

Promotion numérique

```
char c;
short sh;
float f;
int i;
double d;
```

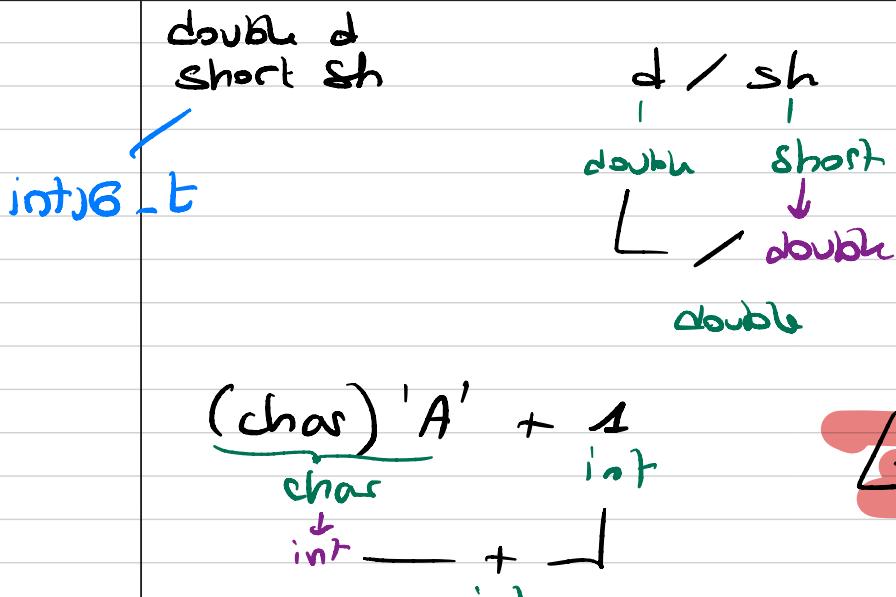
$$(c + sh) * f - i / d$$

1                    2                    3
   
 ——————      ——————      ——————



Résultat: type **l'opérande** → double

## autre exemple



(1) short: pas de promotion numérique vers ce type

**Construction des identificateurs**  
(variables, constantes, fonctions).

**Caractères autorisés:**

- lettres minuscules et Majuscules
- chiffres 0...9
- caractères \_ (souligné, underscore)

**Caractères non autorisés:**

- lettres accentuées
- espace
- symbole (# " / )

**Règles**

- longueur max de 32 caractères.
- minuscules ≠ Majuscules  
Ex: TOTO
- le n° const. n'a pas à être un chiffre.

Correct → x x1 x1 température xFF

Incorrect → 100W #A1 A# double

identification qui exprime la temp. extérieure

température\_extérieure

température\_Exterieure

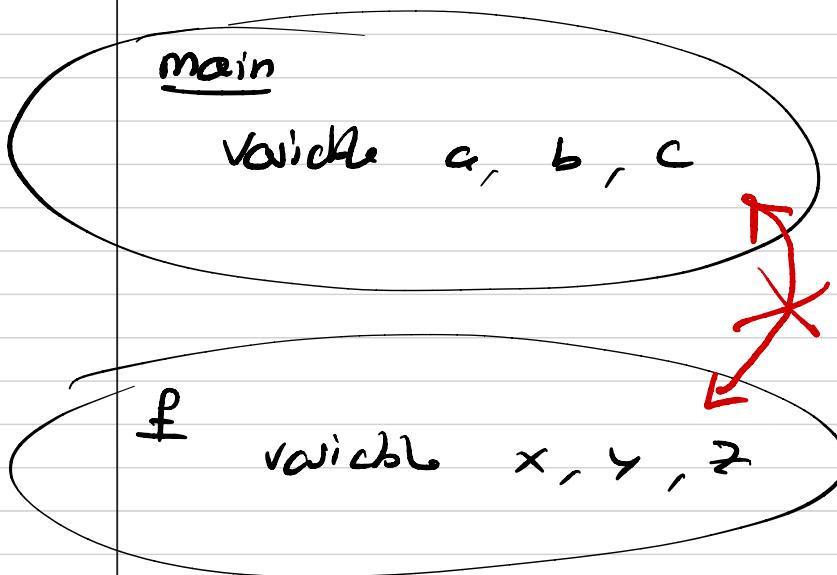
TempératureExtérieure

TEMP\_MAX ← constante

Momogénéité dans l'écriv.

Passage par adresse

Contrôle : donne à une fonction la possibilité de modifier des variables qui lui sont extérieures.



les variables sont  
LOCALES  
aux fonctions

Quand on est dans le main : x, y et z  
n'existent pas.

Quand on est dans f : a, b, c  
n'existent pas.

main: char a=Ø, b=7, c=2

f(&b);

// f(1001);

// b. char

// &b: obtenir d'un  
char.

void

`f( char * z )`

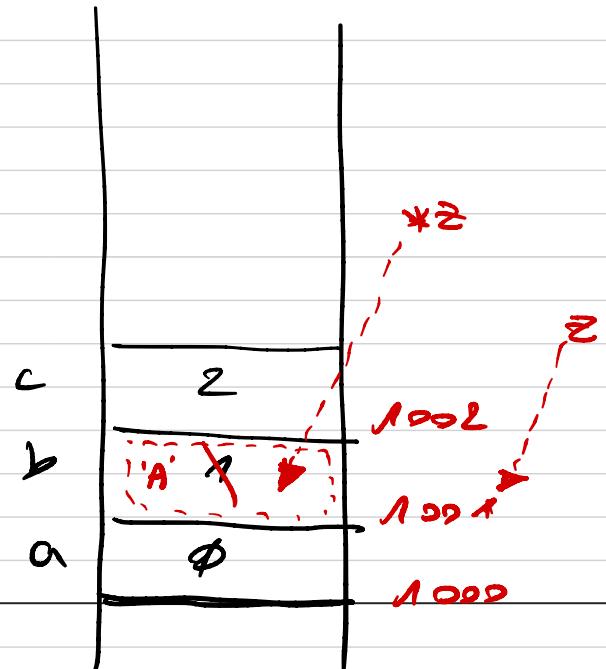
//z : doit pouvoir contenir l'adjectif d'un char.

1/ modifier le contenu de la cellule mémoire à l'adresse Z (1001)

The diagram illustrates the memory representation of a character variable. On the left, a green oval labeled "char" contains the expression "char \*". Below it, another green oval labeled "int" contains the expression "int \*". A green arrow points from the "char" oval to the "char \*" part of the "char \*" expression. Another green arrow points from the "int" oval to the "int \*" part of the "int \*" expression. The entire diagram is set against a background of horizontal blue lines.

Toujours contrôler les types à gauche et à droite du symbole =

The diagram shows a character array `z` of size `N`. A pointer `char *p` points to the first element of the array. The pointer `p` is shown as a blue line pointing downwards from the label `*p` to the first element of the array. The array elements are labeled `z[0], z[1], ..., z[N-1]`.



Variables : Son adresse

char \* z : z contient l'adresse de l'un des

$\alpha z$  contient le char à l'adresse  $z$ .

65535

main chw a, b, c

(( f(100z) ))

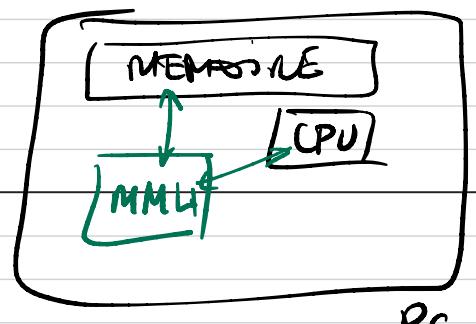
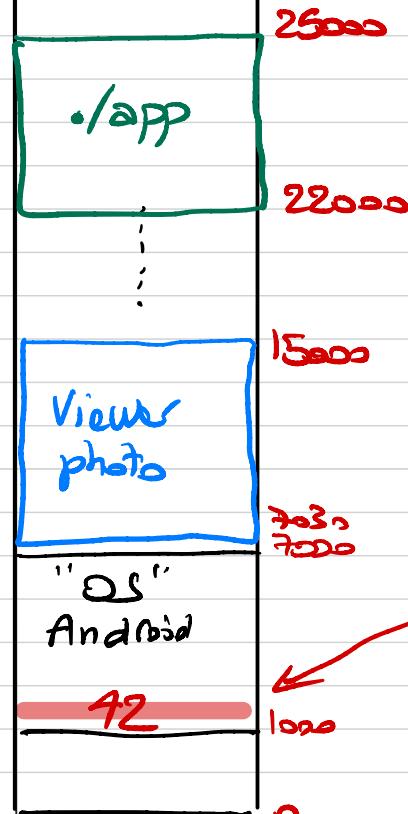
8b

void f ( chw \* z )

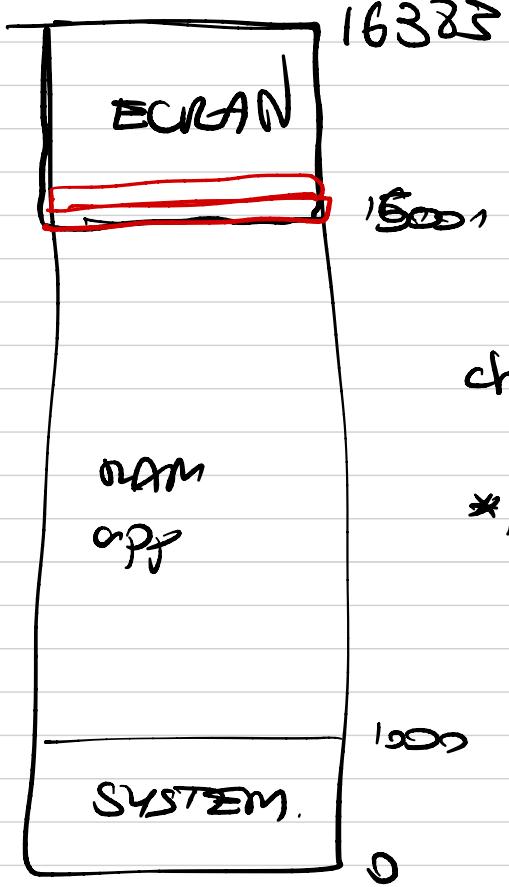
\*z = 72.

z ← 1001

RAM



Memory  
Management  
Unit



`char *p = (char *) 15000;`

$$*p = 255$$

## Conversion de bases numériques

En mémoire : 0 et 1

espace de stockage à 1 bit :  
2 bits

0 ou 1

0  
1  
0  
1  
} 4

7 bits

0000000  
:  
1111111  
} 256

char 8 bit  
short 16 bit  
long 32 bit

256 valeurs ≠  
65536 valeurs ≠  
4G. valeurs ≠

float 32 bit  
double 64 bit  
long double 128 bit

uint8\_t a;

10       $a = 17;$   
8       $a = \phi 7;$   
16      $a = \phi x C;$   
2       $a = \phi b 0111;$

0	0	0	1	0	0	0	1
1	0	0	1	0	1	1	1
0	0	0	0	1	1	1	0
0	0	0	0	0	1	1	1

if ( $\phi 7 == \phi b 111$ ) { - f  
  octal      binaria

00000111      00000111

: if ( $\phi x 3E8 == 1000$ ) { puts("ok"); }

16<sup>h</sup>05



if ( $0 \times 3E8 == 1000$ ) ...

base 16.

base 10

$0 \times 3E8$        $\underbrace{00}_{3} \underbrace{1}_{E} \underbrace{1}_{8}$

$x2^9 \quad x2^8 \quad x2^7 \quad x2^6 \quad x2^5 \quad x2^4 \quad x2^3 \quad x2^2 \quad x2^1 \quad x2^0$

$$2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

$$512 + 256 + 128 + 64 + 32 + 8 = 1000$$

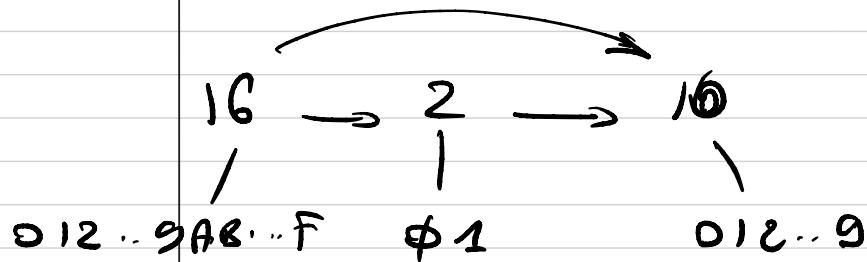
$0 \times 3E8$

$\frac{3}{x16^2}$

$\frac{E}{x16^1}$

$\frac{8}{x16^0}$

$$3 \times 256 + 14 \times 16 + 8 \times 1 = 1000$$



8  
|  
0..7

base 9  
0..8

$$1000_16 \rightarrow (?)_2$$

1000 | 2  
φ | 500 | 2  
0 | 250 | 2  
| 125 | 2  
| 62 | 2  
| 31 | 2  
| 15 | 2  
| 7 | 2  
| 3 | 2  
| 1 | 2

11 | 1110 | 1000

311  
Octal

en base 10:

$$\begin{array}{r} 3 \quad 1 \quad 1 \\ \times 8^2 \quad \times 8^1 \quad \times 8^0 \end{array}$$

$$3 \times 64 + 1 \times 8 + 1 \times 2 = 201$$

201  
<sub>(10)</sub> en base 8

$$\begin{array}{r} 201 \\ 1 \end{array} \left| \begin{array}{r} 8 \\ 25 \\ 1 \end{array} \right. \left| \begin{array}{r} 8 \\ 3 \end{array} \right.$$

311

## Operations math.

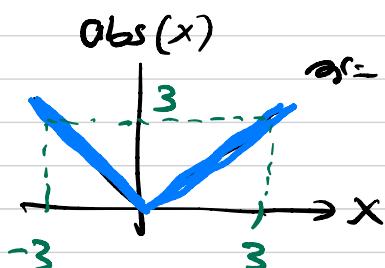
<math.h>

<https://wwwcplusplus.com/reference/cmath/>

trigo	(	cos acos	sin asin	tan atan	atan2	
exp / log	(	exp	log	log10		
	{	pow	$y = x^t$	$y = \text{pow}(x, t)$		
puissance		sqrt	$y = \sqrt{x}$	$y = \text{sqrt}(x)$		
		cbrt	$y = \sqrt[3]{x}$	$y = \text{cbrt}(x)$	<u>C++</u> <u>A</u>	
arround:		ceil	3.51	→ +		
		floor	3.4	→ 3		

absolute value

fabs double



$$x = 3$$

$$|x| = 3$$

$$x = -3$$

$$|x| = 3$$

abs = double | float | long double

{ #define ABS(x) ((x)<0 ? (-x) : (x)) }

int k = -3;

int a = ABS(k); // a ← 3

or sinon

a = (int) abs(((double)k));

## Types de données.

0, 1, -12, 5422	int
3L	long
'A'	int
12u	unsigned int
42UL	unsigned long
53LL	long long
3.14f	float
3.14	double
14.5e-9	double
18.1L	long double