



LABORATORIS

PROGRAMACIÓ CIENTÍFICA

Sessió 2B: Operadors

OPERADORS

Operadors

Aritmètics

Op.		E.g.
+	Suma	a+b
-	Resta	a-b
*	Multiplicació	a*b
/	Divisió (compte!)	a/b
%	Mòdul (residu de la divisió)	a%b
++	Incrementa l'enter en una unitat	a++
--	Decrementa l'enter en una unitat	a--

Relacionals

Op.		E.g.
==	Comparació d'igualtat (=)	a==b
!=	'≠'	a != b
>	>	a > b
<	<	a < b
>=	≥	a >= b
<=	≤	a <= b

Lògics

Op.		E.g.
&&	and	a && b
	or	a b
!	not	! a

Operadors d'assignació


Op.		E.g.
=	Assignació simple	C = A + B
+=	Suma i assignació	C += A equival a: C = C + A
-=	Resta i assignació	C -= A equival a: C = C - A
*=	Multiplicació i assignació	C *= A equival a: C = C * A
/=	Divisió i assignació	C /= A equival a: C = C / A
%=	Mòdul i assignació	C %= A equival a: C = C % A

- N'hi ha més....

Operadors

Divisió

- S'ha d'anar en compte amb la divisió perquè es comporta diferent segons el tipus dels seus operands.
- **Si dividim dos enters, es fa divisió entera, independentment d'on guardem el resultat.**



```
#include <stdio.h>

int main(){
    int e1 = 10;
    int e2 = 3;
    /* El resultat es 3.0000000*/
    printf("El resultat es %f", e1/e2)
}
```

Operadors

Divisió

- S'ha d'anar en compte amb la divisió perquè es comporta diferent segons el tipus dels seus operands.
- Si dividim dos enters, es fa divisió entera, independentment d'on guardem el resultat.
- **Si com a mínim un dels operands és un decimal , es fa divisió amb decimals...**
 - **Si dessem el resultat d'aquesta operació en un real, OK.**

```
#include <stdio.h>

int main(){
    float d1 = 10;
    float d2 = 3;
    /*El resultat es 3.333333*/
    printf("El resultat es %f\n", d1/d2);
}
```

Operadors

Divisió

- S'ha d'anar en compte amb la divisió perquè es comporta diferent segons el tipus dels seus operands.
- Si dividim dos enters, es fa divisió entera, independentment d'on guardem el resultat.
- **Si com a mínim un dels operands és un decimal , es fa divisió amb decimals...**
 - Si dessem el resultat d'aquesta operació en un real, OK.
 - **Si dessem el resultat d'aquesta operació en un enter, perdem els decimals.**

```
#include <stdio.h>

int main(){
    int r;
    float d1 = 10;
    float d2 = 3;

    r = d1/d2;

    /*El resultat es 3*/
    printf("El resultat es %d\n", r);

}
```

Operadors

Divisió

- S'ha d'anar en compte amb la divisió perquè es comporta diferent segons el tipus dels seus operands.
- Si dividim dos enters, es fa divisió entera, independentment d'on guardem el resultat.
- **Si com a mínim un dels operands és un decimal , es fa divisió amb decimals...**
 - Si dessem el resultat d'aquesta operació en un real, OK.
 - **Si dessem el resultat d'aquesta operació en un enter, perdem els decimals.**

```
#include <stdio.h>

int main(){
    int r;
    float d1 = 10;
    float d2 = 3;

    r = d1/d2;


    /*El resultat es 3*/
    printf("El resultat es %d\n", r);
    /*El resultat es 0.000000
    (comportament indefinit)*/
    printf("El resultat es %f\n", r);
}
```

Operadors

Typcasting

- Entre tipus compatibles, es pot fer una conversió de tipus anomenada **typecast**.

```
(nom_del_tipus) expressió
```



```
#include <stdio.h>

int main(){
    int a = 17;
    int b = 5;
    float c;
    c = (float) a/b;
    printf("El resultat es %f\n",c);
}
```

El resultat es 3.400000

Operadors

Typcasting

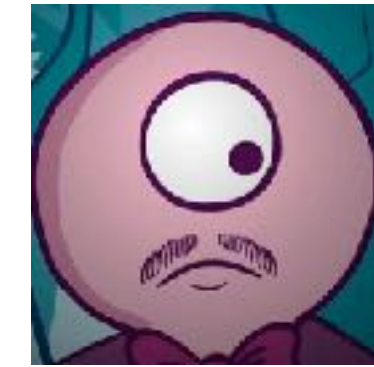
- Entre tipus compatibles, es pot fer una conversió de tipus anomenada **typecast**.

`(nom_del_tipus) expressió`

```
#include <stdio.h>
```

```
int main(){  
    int a = 17;  
    int b = 5;  
    float c;  
    c = (float) a/b;  
    printf("El resultat es %f\n",c);  
}
```

El resultat es 3.400000



OJO CUIDAO!

- C ens deixarà fer typecast sempre, però a vegades implicarà pèrdua d'informació.
- No es perd informació si transformem un tipus més específic ("narrower") a un tipus més general ("wider").
- E.g. podem transformar un enter (16 bits) a un long (32 bits) sense perdre informació.
- E.g. no podem transformar un enter (16 bits) a un char (8 bits) sense perdre informació.

La doble personalitat dels caràcters

Caràcters

- Els caràcters es poden imprimir com a caràcters o com a nombres:

lName	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

lName	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

lName	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

lName	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

La doble personalitat dels caràcters

Caràcters

- Els caràcters es poden imprimir com a caràcters o com a nombres:

```
#include <stdio.h>

int main(){
    char c1 = 'A';
    char c2 = '4';
    /* Imprimint com a caràcter */
    printf("Els caràcters són %c i %c\n", c1, c2);
    /* Imprimint com a nombre */
    printf("Els caràcters són %d i %d\n", c1, c2);
}
```

lName	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

lName	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

lName	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

lName	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

FRAGMENT DE LA TAULA ASCII

La doble personalitat dels caràcters

Caràcters

- Els caràcters es poden imprimir com a caràcters o com a nombres:

```
#include <stdio.h>

int main(){
    char c1 = 'A';
    char c2 = '4';
    /* Imprimint com a caràcter */
    printf("Els caràcters són %c i %c\n", c1, c2);
    /* Imprimint com a nombre */
    printf("Els caràcters són %d i %d\n", c1, c2);
}
```

Els caràcters són A i 4
Els caràcters són 65 i 52

lName	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

lName	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075


lName	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

lName	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

Transformar ASCII a numèric

Operar amb caràcters

- Com que “darrere” de l’aspecte de caràcter, són nombres, podem operar amb ells:



```
#include <stdio.h>

int main( ){
    char c1 = 'A';
    char c2 = '4';

    /* Operació vàlida (que no té sentit en aquest cas) */
    printf("Resultat de %c + %c = %d\n", c1, c2, c1+c2);
}
```

Transformar ASCII a numèric

Operar amb caràcters

- Com que “darrere” de l’aspecte de caràcter, són nombres, podem operar amb ells:

```
#include <stdio.h>

int main( ){
    char c1 = 'A';
    char c2 = '4';

    /* Operació vàlida (que no té sentit en aquest cas) */
    printf("Resultat de %c + %c = %d\n", c1, c2, c1+c2);
}
```

Resultat de A + 4 = 117

Transformar ASCII a numèric

Operar amb caràcters

- Com que “darrere” de l’aspecte de caràcter, són nombres, podem operar amb ells:
- Una operació útil és transformar un caràcter a un nombre. Com?

Transformar ASCII a numèric

Operar amb caràcters

- Com que “darrere” de l’aspecte de caràcter, són nombres, podem operar amb ells:
- Una operació útil és transformar un caràcter a un nombre. Com?
 - Valor ASCII del nostre caràcter ‘4’ = 52
 - Valor ASCII del caràcter ‘0’ = 48
 - Resta ‘4’ - ‘0’ = 52 - 48 = 4 (el valor numèric que volem!)

lName	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

lName	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

lName	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

lName	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

TAULA ASCII INCOMPLETA

Transformar ASCII a numèric

Operar amb caràcters

- Com que “darrere” de l’aspecte de caràcter, són nombres, podem operar amb ells:
- Una operació útil és transformar un caràcter a un nombre. Com?
 - Valor ASCII del nostre caràcter ‘4’ = 52
 - Valor ASCII del caràcter ‘0’ = 48
 - Resta ‘4’ - ‘0’ = 52 - 48 = 4 (el valor numèric que volem!)

```
#include <stdio.h>

int main(){
    char c2 = '4';
    int num;
    /* Transformar caràcter a enter */
    num = c2 - '0';
    printf("El num es %d\n", num);
}
```

El num es 4

lName	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

lName	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

lName	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

lName	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

TAULA ASCII INCOMPLETA

Altres operacions interessants

Transformar a minúscules

- Fent servir la mateixa lògica, podem convertir lletres de minúscula a majúscula i al revés.

```
#include <stdio.h>

int main(){
    char c = 'H';
    char min;
    min = c - 'A' + 'a';
    printf("Caracter original:  %c\n",c);
    printf("Caracter minúscula: %c\n",min);
}
```

Caracter original: H
Caracter minúscula: h

a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Operadors

Precedència d'operadors

Categoria	Operador
Postfix	() ++ --
Unary	+ - ! ~ ++ -- (type)* & sizeof
Multiplicative	* / %
Additive	+ -
Shift	<< >>
Relational	< <= > >=
Equality	== !=
Bitwise AND	&
Bitwise XOR	^
Bitwise OR	
Logical AND	&&
Logical OR	
Conditional	?:
Assignment	+= -= *= /= %= >>= <<= &= ^= =
Comma	,

Per practicar a casa...

- Escriviu un programa que demani a l'usuari dos valors enters i n'imprimeixi per pantalla la seva suma, la seva resta, la seva multiplicació i la divisió.

ALTRES OPERACIONS

MATEMÀTIQUES

Operacions matemàtiques

Llibreria `<math.h>`

- La biblioteca estàndard, definida per l'estandard ANSI, ens proporciona altres funcions més complexes molt útils per escriure programes en C. En el cas d'operacions matemàtiques, la llibreria a incloure és **`<math.h>`**
- <https://en.cppreference.com/w/c/numeric/math>

Funcions interessants (una selecció)

<code>sin(x)</code>	<code>exp(x)</code>
<code>cos(x)</code>	<code>log(x)</code>
<code>tan(x)</code>	<code>log10(x)</code>
<code>asin(x)</code>	<code>pow(x,y)</code>
<code>acos(x)</code>	<code>sqrt(x)</code>
<code>atan(x)</code>	<code>ceil(x)</code>
<code>atan2(x,y)</code>	<code>floor(x)</code>
<code>sinh(x)</code>	<code>fabs(x)</code>
<code>cosh(x)</code>	<code>fmod(x,y)</code>
<code>tanh(x)</code>	

Operacions matemàtiques

Llibreria <math.h>

- La biblioteca estàndard, definida per l'estandard ANSI, ens proporciona altres funcions més complexes molt útils per escriure programes en C. En el cas d'operacions matemàtiques, la llibreria a incloure és **<math.h>**
- <https://en.cppreference.com/w/c/numeric/math>

Funcions interessants (una selecció)

sin(x)	exp(x)
cos(x)	log(x)
tan(x)	log10(x)
asin(x)	pow(x,y)
acos(x)	sqrt(x)
atan(x)	ceil(x)
atan2(x,y)	floor(x)
sinh(x)	fabs(x)
cosh(x)	fmod(x,y)
tanh(x)	

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(){
5
6     // Calcul d'una potencia
7     int base, exp, res;
8     base = 2;
9     exp = 10;
10    res = pow(base,exp);
11    return 0;
12 }
```

A CASA...

EXERCICIS L2B del Moodle

A CONTINUACIÓ

CONDICIONALS + BUCLES