UNIVERSIDAD DE MURCIA

Fundamentos de Informática
5. Operadores, expresiones
(y su aplicación)

Fundamentos de Informática
Grado en Ingeniería Química

2

Contenidos

- Operadores
  - De asignación
  - Aritméticos
  - De incremento
  - Relacionales
  - Lógicos
- Tipos de datos
  - Otros tipos de datos
  - Conversiones de tipos de datos

3

Operadores, expresiones y tipos de datos

Operadores de asignación

4

Asignación

=

asigna a la variable de la izquierda la expresión de la derecha

```
int contador;
float grados, f;

contador = 45;
grados = 32.58;

contador = contador +1;
```

5

Asignación

También puedo hacer esto

```
int i,j,k;

i = j = k = 10;
```

6

Asignación

Y esto

```
int i=10, j, k=8;

i = i + 1;
j = i + 2;
k = k  - 1;
```

7

Asignación

Otros operadores de asignación son

+= , -= , *=, /=, %=

```
i += 5;    →   i=i+5;
i -= 5;    →   i=i-5;
i *= 5;    →   i=i*5;
i /= 5;    →   i=i/5;
i %= 5;    →   i=i%5;
```

Ahorran escritura

8

Asignación

Otro ejemplo

```
i += (5*a+b)/(3*c);    →   i = i+ ((5*a+b)/(3*c));
```

9

Operadores, expresiones y tipos de datos

Operadores aritméticos

10

Operadores . . .

Aritméticos

+ - * / %

¿Cómo se evalúa 3+5*2?

(3+5) * 2 → 16

3 + (5*2) → 13

Prioridad

- (unario)

* / %

+ -

A igual prioridad estos operadores se evalúan de izquierda a derecha

Es recomendable poner paréntesis en expresiones largas

11

Operadores, expresiones y tipos de datos

Operadores de incremento / decremento

12

Incremento

++ --

Suman o restan uno a la variable que se aplican.

n++;	→	n=n+1;
n--;	→	n=n-1;
++n;	→	n=n+1;
--n;	→	n=n-1;

13

Incremento

n++ parece lo mismo que ++n, pero no es así cuando están en una expresión de asignación:

b = a++; b= ++a;

Si ++ es prefijo, el incremento se realiza antes que la asignación. Si es sufijo, despues.

```
int a=1, b;
b = ++a;
```

a vale 2

b vale 2

```
int a=1, b;
b = a++;
```

a vale 2

b vale 1

14

Operadores, expresiones y tipos de datos

Operadores relacionales y lógicos

15

Operadores . . .

Relacionales

Son los que se usan en comparaciones e iteraciones

Producen valores booleanos (binarios)

Devuelven un valor ***int***

1 si cierto

0 si falso

==	igual a	a == b
!=	no igual a	a != b
>	mayor que	a > b
<	menor que	a < b
>=	mayor o igual	a >= b
<=	menor o igual	a <= b

```
if (a==b) printf…;
if ((b-5)>(c*3+1)) z=8;
```

```
c= (3<7); /* c vale 1 */
'A' < 'C'; /* 1, porque 65<67 */
```

Tienen menor prioridad que los aritméticos

m+5 <= 2*n equivale a (m+5) <= (2*n)

16

Operadores . . .

Lógicos

&&	**AND**
||	**OR**
!	**NOT**

```
if ( (m<n) && (i>j) ) printf….;
c = ( (m<n) || (i>j) );
while (!(x+7= =5)) { …
```

- **&&** tiene más prioridad que ||
- **A igual prioridad, se evalúa de izquierda a derecha**
- **!** Tiene la misma prioridad que **–** (unario)

Prioridad:
1º Matemáticos y ***!***
2º Relacionales
3º Lógicos

```
if ( v < s * 3 && a > 10 * iva ) …
equivale a
if ( (v < (s * 3)) && (a > (10 * iva)) )
```

17

Operadores . . .

Prioridades

0	**(,)**	I - D
1	**-** (unario), **!**, **++**, **- -**	D - I
2	*****, /, **%**	I - D
3	**+**, **-**	I - D
4	**>**, **<**, **>=**, **<=**	I - D
5	**==**, **!=**	I - D
6	**&&**	I - D
7	||	I - D
8	**=**, **+=**, **- =**, ***=**, /=	D - I

18

Operadores, expresiones y tipos de datos

Hay algunos otros operadores, menos importantes para nosotros: manipulación de bits, especiales, etc.

19

Operadores, expresiones y tipos de datos

Otros tipos de datos

20

Otros tipos de datos

Conocíamos los tipos ***int, float y char***.
Hay variaciones sobre ellos, para mejorar la precisión.

Tipo	Tamaño habitual
float	4 byte
double	8 byte
long double	12 bytes

Tipo	Tamaño habitual
char	1 byte
unsigned char	1 byte
int	4 bytes
unsigned int	4 bytes

El número de bytes puede depender del hardware y el compilador.

```
int a;
printf ("%d", sizeof(a));
```

21

Otros tipos de datos

Algunos rangos de tipos de datos

Tipo	Tamaño	Rango
char	1 byte	-128 .. 127
unsigned char	1 byte	0 .. 255
int	4 bytes	-2.147.483.648 .. 2.147.483.647
unsigned int	4 bytes	0 .. 4.294.967.295
float	4 bytes	3.4×10^{-38} .. 3.4×10^{38}
double	8 bytes	1.7×10^{-308} .. 1.7×10^{308}

¡Ojo!, C permite asignaciones erróneas (otros lenguajes controlan rangos).

```
char a;
a=128;
printf ("%d",a);        /* valor impredecible*/
a=127;a*=2;
printf ("%d",a);        /* valor impredecible*/
```

22

Otros tipos de datos

El tipo de dato char

Un caracter se almacena por su código ASCII, un entero de 1 a 127.

char almacena enteros -128..127, por lo que puede almacenar caracteres (el código ASCII del caracter)

```
char c1,c2;
c1=65;
c2='A';
printf ("%d %c" %d %c, c1,c1,c2,c2);
```

```
char c;
c='A';
c=c+1;
printf ("%d %c", c,c);
```

Con **unsigned char** (0..255) podemos almacenar también los caracteres especiales del código ASCII

23

Operadores, expresiones y tipos de datos

Conversiones de tipos de datos

24

Conversiones de tipos de datos

Conversiones

Para evaluar expresiones a veces hay que operar entre distintos tipos de datos, y deben hacerse conversiones.

```
int i;
double x=2.4;
i=2*x;
```

En este ejemplo, C realiza una conversión automática. Los formatos internos de ambos datos son distintos, y si no la hicera se producirían errores.

```
/* C se encarga de esto */
int i; unsigned int j;
double x; float y;
…
if (y == 3*i*(j+x)) …
```

Conversiones de tipos de datos

Conversión automática

Los operandos de tipo más bajo se convierten a tipo más alto.
char se convierten a ***int***, y luego, según la lista:

char → int ⟶ float ⟶ double

```
int i=12;
double x=4;

x=x+i;    // i a double antes de sumarse
x=i/5;    // se hace div entera. 2 se convierte a 2.0 y se asigna a x
x=4.0;    // 4.0 se asigna a x como double
x=x/5;    // 5 se convierte a double y se divide. 0.8 se asigna a x como double
```


Conversiones de tipos de datos

Conversión forzada (cast)

Se puede forzar a convertir de tipo mediante el cast ***(tipo) valor***

```
(int) a;
(int) 3.7;              // se trunca a 3
(int) (3*b+c);
(float) contador;
(unsigned int) vector;
```

```
/* da 30 como resultado */

float x,y, precio;

x = 19.99;
y = 11.99;
precio = (int) x + (int) y;
```

27

Operadores, expresiones y tipos de datos

- - - - - -