

Tema 2: Fundamentos de C

Oscar Perpiñán Lamigueiro - David Álvarez

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Aritméticos

```
x + y  
x - y  
x / y // OJO con el tipo de datos  
x * y  
x % y // módulo o resto de división de entera
```

Aritméticos con enteros

```
#include <stdio.h>

int main()
{
    int x, y, sum;
    x = 10;
    y = 15;
    sum = x + y;

    printf("La suma de %i con %i es %i\n", x, y, sum);
    printf("La resta de %i con %i es %i\n", x, y, x-y);
    // los demás?
    // y con decimales?

    return 0;
}
```

Ejercicios

- Escriba un programa que lea del teclado una letra minúscula e imprima esa misma letra en mayúscula.
- Escriba un programa que pida un peso y altura de una persona y calcule su índice de masa corporal (imc). El imc se calcula dividiendo el peso en (kg) por la altura (m) al cuadrado.

Asignaciones especiales

- Recomendando no usar mientras se aprende

```
x = y
x += y // x = x + y
x -= y // x = x - y
x *= y // x = x * y
x /= y // x = x / y
x %= y // x = x % y
```

```
// ERROR: en el lado izquierdo no puede ir una expresión
x + y = 1
```

Ejercicio operaciones de asignación

```
#include <stdio.h>
int main() {
    int a, b = 3;

    a = 5;
    printf("a = %i\n", a);
    a *= 4; // a = a * 4
    printf("a = %i\n", a);
    a += b; // a = a + b
    printf("a = %i\n", a);
    a /= (b + 1); // a = a / (b+1)
    printf("a = %i\n", a);
    a = b = 1;
    printf("a = %i, b = %i\n", a, b);

    return 0;
}
```


Incrementales - solo usar en bucles

```
y = ++x // x = x + 1; y = x (preincremento)
y = x++ // y = x; x = x + 1 (postincremento)

y = --x // x = x - 1; y = x (predecremento)
y = x-- // y = x; x = x - 1 (postdecremento)
```

Ejercicio operaciones de incremento

```
#include <stdio.h>

int main()
{
    int b = 2, r;
    //Preincremento
    r = ++b;
    printf("b = %i, r = %i\n", b, r);
    //Postincremento
    r = b++;
    printf("b = %i, r = %i\n", b, r);

    return 0;
}
```

Ejercicio operaciones de incremento 2

```
#include <stdio.h>

int main()
{
    int a = 0;
    printf("a = %i\n", ++a);
    printf("a = %i\n", a++);
    printf("a = %i\n", a);
    printf("a = %i\n", --a);
    printf("a = %i\n", a--);
    printf("a = %i\n", a);

    return 0;
}
```

Ejercicio precedencia y asociatividad

```
#include <stdio.h>

int main() { // Y si son enteros?
    float a = 4, b = 7, c = 3, g = 9, result;

    result = a + b * c;
    printf( "resultado = %f\n", result);

    result = (a + b) * c;
    printf( "resultado = %f\n", result);

    result = a * b / c * g;
    printf("resultado = %f\n", result);

    result = (a * b) / (c * g);
    printf("resultado = %f\n", result);

    return 0;
}
```

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Relacionales

- El resultado es verdadero o falso.
- En programación esto significa 1 ó 0 respectivamente.

```
x == y  
x != y  
x > y  
x >= y  
x < y  
x <= y
```

Ejercicio operaciones comparadores

```
#include <stdio.h>
int main()
{
    int x = 10, y = 3;

    printf("x igual a y = %i\n", (x == y));
    printf("x distinto a y = %i\n", (x != y));
    printf("x mayor que y = %i\n", (x > y));
    printf("x menor o igual a y = %i\n", (x <= y));
    printf("x mayor o igual que y = %i\n", (x >= y));

    return 0;
}
```

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

- AND, OR, NOT

```
x && y //AND  
x || y //OR  
!x //NOT, operador unario
```

- Operador condicional ? (ternario)

```
// expresión booleana ? valor si cierto : valor si falso  
x > y ? "cierto" : "falso"  
x == y ? "true" : "false"
```

Ejercicio operaciones lógicas

```
#include <stdio.h>
int main()
{
    int x, resto;

    printf("Escribe un número entero: ");

    scanf("%i", &x);

    // Calcula el resto de dividir por 2
    resto = x % 2;

    // Si el resto es 0, x es par.
    printf("Es un número %s\n", (resto == 0) ? "par" : "impar");

    return 0;
}
```

- 1 Primeros pasos
- 2 Datos en C
 - Almacenamiento de la información
 - Datos en un programa en C
 - Caracteres
- 3 Interacción entrada/salida (I/O)
 - Especificaciones de formato
 - Imprimir en el terminal
 - Lectura de datos desde el teclado
- 4 Operadores
 - Aritméticos
 - Comparadores
 - Lógicos
- 5 Conversión de tipos de datos

Conversión implícita

Asignaciones el valor de la derecha se convierte al tipo de la variable de la izquierda (posible aviso o error). NO RECOMENDABLE.

```
#include <stdio.h>

int main()
{
    float f1 = 3.7, f2;
    int i1 = 2, i2;
    // Real a entero: pierde decimales
    i2 = f1;
    printf("Un real %f convertido a entero %i\n", f1, i2);
    // Entero a real: no cambia valor
    f2 = i1;
    printf("Un entero %i convertido a real %f\n", i1, f2);

    return 0;
}
```

Conversión explícita

Conversión explícita o forzada (tipo) expresión

```
#include <stdio.h>

int main()
{
    int i1 = 5, i2=2;
    float f1;

    printf("La división entre enteros: %i/%i = %f\n", i1, i2, (float)(i1/i2));

    // Completar el último campo
    //printf("El resto de %i entre %i es: %i", i1, f1, ...);

    return 0;
}
```

Conversión en expresiones

Expresiones los valores de los operandos se convierten al tipo del operando que tenga la precisión más alta.

```
#include <stdio.h>

int main()
{
    double f1 = 100;
    int i1 = 150, i2 = 100;
    printf("Un entero, %i, dividido por un real, %f,", i1, f1);
    printf(" produce un real, %f\n", i1 / f1);
    printf("Un entero, %i, por un entero, %i: %i\n", i1, i2, i1 / i2);

    return 0;
}
```