

# Programación en C

Francisco Martínez & César Sánchez

October 11, 2018

# Aritmética en C

# Operadores Aritméticos

Operación C	Operador aritmético	Expresión algebraica	Expresión
Suma	+	$f + 7$	$f + 7$
Resta	-	$p - c$	$p - c$
Multiplicación	*	$bm$ $bm \mid b \ m$	
División	/	$x/y$	$x / y$
Módulo	%	$r \bmod s$	$r \% s$

# Operadores Aritéticos

Operación en C	Operador aritmético	Expresión
Suma	+	$f + 7$
Resta	-	$p - c$
Multiplicación	*	$bm$
División	/	$x / y$ o $x / y$
Módulo	%	$r \bmod s$

Figure : Operadores Aritméticos

# Precedencia de Operadores

Operador(es)	Operación(es)	Orden de e
*	Multiplicación	Se evalúan a derecha.
/	División	
%	Módulo	
+	Suma	Se evalúan a derecha.
-	Resta	

Figure : Precedencia de Operadores

## Operadores de Igualdad y Relación

Operador algebraico estándar de igualdad o de relación	Operador de igualdad o de relación en C	Ejemplo de condición en C
<i>Operadores de igualdad</i>		
=	==	x == y
≠	!=	x != y
<i>Operadores de relación</i>		
>	>	x > y
<	<	x < y
≥	>=	x ≥ y
≤	<=	x ≤ y

# Ejercicios.

- Hacer un programa que pida dos números y te diga si son iguales o no.

# Sentencias de Control



## Sentencia if

Permite ejecutar o no una sentencia o bloque, en función de si una expresión es cierta o no.

- Una sentencia:

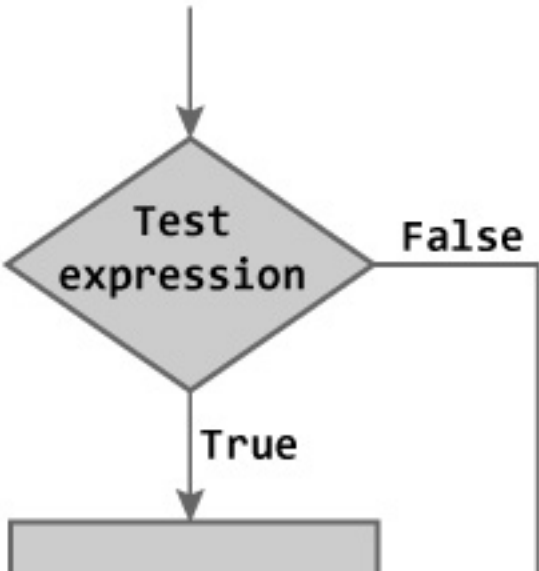
```
if (expresion)  
    sentencia;
```

- Un bloque:

```
if (expresion)  
{  
    //bloque  
    ...  
}
```

“expresion” se construye con operadores lógicos y relacionales.

# Diagrama de Flujo if

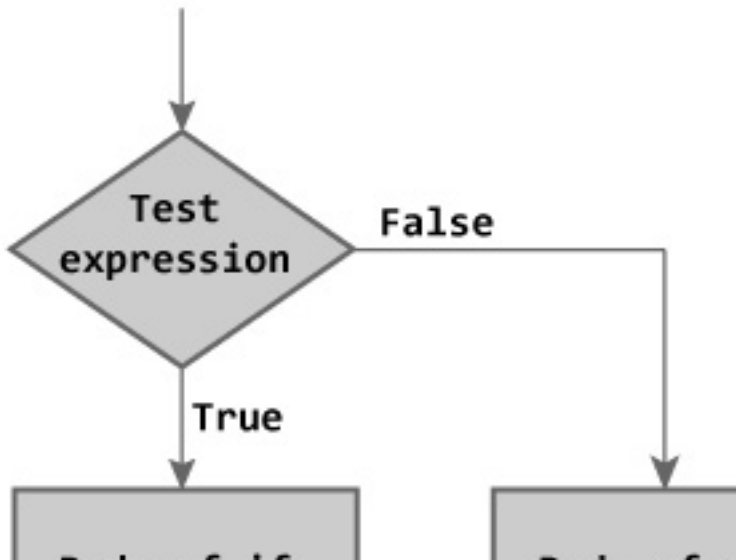


## Sentencia if-else

Permite ejecutar una sentencia o bloque u otra sentencia o bloque en función de si una expresión es cierta o no.

- `if (expresion)`  
    `sentencia;`  
`else`  
    `sentencia;`
- `if (expresion)`  
    {  
    *//bloque*  
    ...  
    }  
`else`  
    `sentencia;`
- Blablabla...

# Diagrama de Flujo if-else



## Sentencia if e if-else anidados

Entre las sentencias después de `if` o `else` pueden haber más `if` e `if-else`. Por el lado del `else`:

```
if (condicion_1)
{
    ...
}
else
    if (condicion_2)
    {
        ...
    }
    else
        if (condicion_3)
        {....
        }
```

## Sentencia if e if-else anidados

Entre las sentencias después de `if` o `else` pueden haber más `if` e `if-else`. Por el lado del `if`:

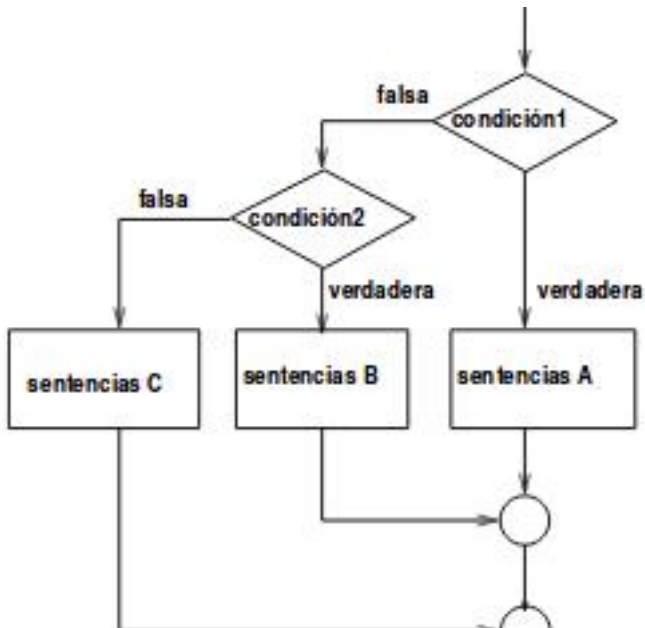
```
if (condicion_1)
{
    if (condicion_2)
    {
        ...
    }
    else
    {
        ...
    }
}
else ...
```

## Diagrama de Flujo if-else-anidados

Pero tambien puede haber más if e if-else por ambos lados...

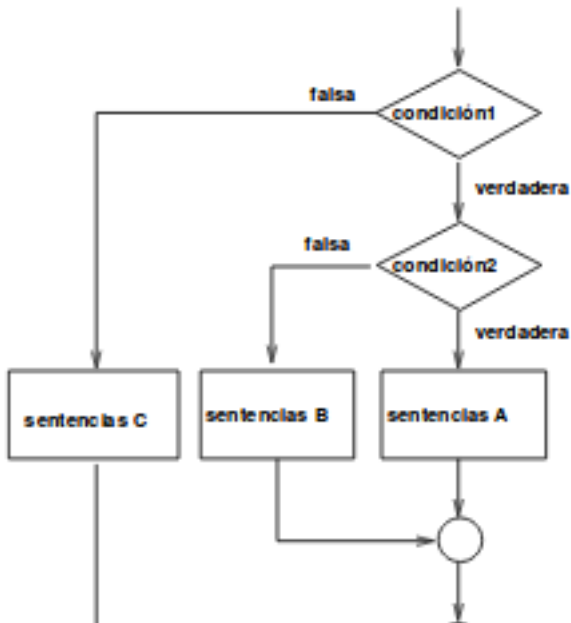
```
if (condicion_1)
{
    if (condicion_2)
    {
        ...
    }
    else
    {
        ...
    }
}
else
{...}
```

## Diagrama de Flujo if-else-anidados





## Diagrama de Flujo if-else-anidados



# Ejercicios

- Hacer un programa que diga si un caracter es vocal o consonante.
- Hacer un programa que diga el número más grande entre cuatro números.

# Sentencia Switch

Es como tener varios if-else restringiendo la condición a la comparación de la igualdad entre la expresión y constantes.

- Switch

```
switch (expresion)
{
    case cte1: ...
        break;
    case cte2: ...
        break;
    ...
    default: ...
}
```

# Sentencia Switch

- if

```
if (expression == cte1)
{
    ...
}

else
    if (expression == cte2)
    {
        ...
    }

...

else
    ...
```

# Sentencia Switch

## Cosas a considerar

- Si se omite `break` se ejecuta todo el código que siga hasta encontrar el siguiente.
- La expresión es de tipo entero o carácter.
- Después de `case` solo pueden ir constantes de este tipo.
- La condición es, implícitamente, la comparación a igualdad entre `expresion` y las constantes. No se pueden hacer otro tipo de comparación.

# Ejercicios

- Hacer una calculadora, la calculadora debe de recibir dos numeros tipo float y un caracter para decidir que operación se quiere hacer. Se le debe de mostrar al usuario un menú de opciones para realizar la operación. Acto seguido el usuario ingresa los números y el programa devuelve el resultado.