

Programación en C

Francisco Martínez & César Sánchez

Aritmética en C

Operadores Aritméticos

Operación	Operador aritmético	Exp. algebraica	Exp. C
Suma	+	$f + 7$	$f + 7$
Resta	-	$p - c$	$p - c$
Multiplicación	*	bm o $b*m$	$b*m$
División	/	x/y	x / y
Módulo	%	$r \text{ mod } s$	$r \% s$

Operadores Aritéticos

Operación en C	Operador aritmético	Expresión algebraica	Expresión en C
Suma	+	$f + 7$	<code>f + 7</code>
Resta	-	$p - c$	<code>p - c</code>
Multiplicación	*	bm	<code>b * m</code>
División	/	x / y o $\frac{x}{y}$ o $x \div y$	<code>x / y</code>
Módulo	%	$r \bmod s$	<code>r % s</code>

Figure 1: Operadores Aritméticos

Precedencia de Operadores

Operador(es)	Operación(es)	Orden de evaluación (precedencia)
*	Multiplicación	Se evalúan primero. De izquierda a derecha.
/	División	
%	Módulo	
+	Suma	
-	Resta	

Precedencia de Operadores

Operador(es)	Operación(es)	Orden de evaluación (precedencia)
*	Multiplicación	Se evalúan primero. Si hay muchas, se evalúan de izquierda a derecha.
/	División	
%	Módulo	
+	Suma	Se evalúan después. Si hay muchas, se evalúan de izquierda a derecha.
-	Resta	

Figure 2: Precedencia de Operadores

Operadores de Igualdad y Relación

Operador en C	Ejemplo	Significado
==	x == y	x es igual que y
!=	x != y	x no es igual que y
>	x > y	x mayor que y
<	x < y	x menor que y
>=	x >= y	x mayor o igual que y
<=	x <= y	x menor o igual que y

Operadores de Igualdad y Relación

Operador algebraico estándar de igualdad o de relación	Operador de igualdad o de relación en C	Ejemplo de una condición en C	Significado de la condición en C
<i>Operadores de igualdad</i>			
=	==	x == y	x es igual que y
≠	!=	x != y	x no es igual que y
<i>Operadores de relación</i>			
>	>	x > y	x es mayor que y
<	<	x < y	x es menor que y
≥	>=	x >= y	x es mayor o igual que y
≤	<=	x <= y	x es menor o igual que y

Figure 3: Operadores de Igualdad y Relación

- 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

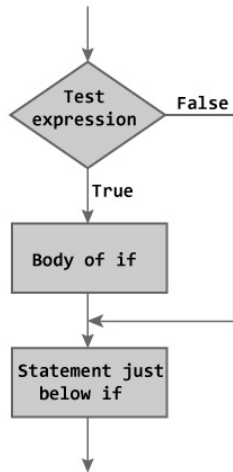


Figure: Flowchart of if Statement

Figure 4: 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

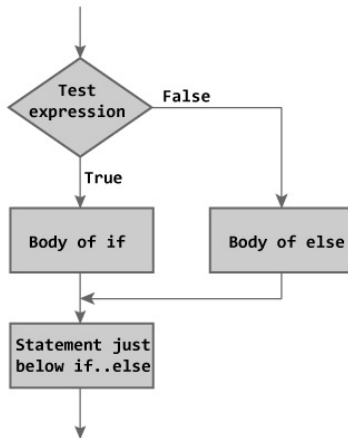


Figure: Flowchart of if...else Statement

Figure 5: 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

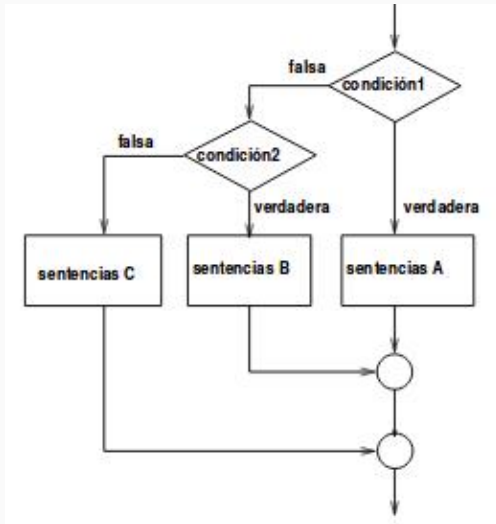


Figure 6: Diagrama de Flujo if-else-anidados por el lado del else

Diagrama de Flujo if-else-anidados

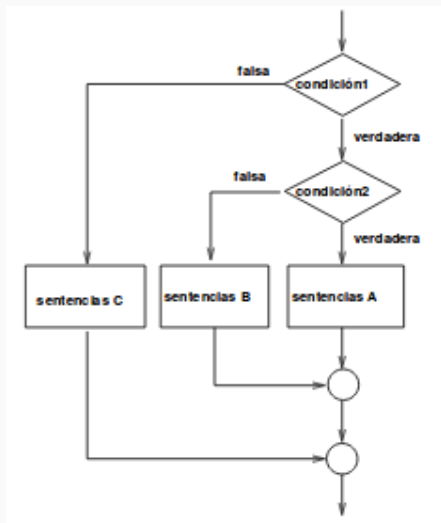


Figure 7: Diagrama de Flujo if-else-anidados por el lado del if

- 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: ...
}
```

- if

```
if (expresion == cte1)
{
    ...
}
else
if (expresion == cte2)
{
    ...
}
...
else
    ...
```

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.

- 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.