



2

Tema 2.- Introducción a la Programación



4

Características del Lenguaje C++



Información I
n Estadística Aplicada. EUE.

La programación en C++

prácticos: introducción al entorno de C++
a de un programa
s básicos: palabras reservadas, doadores, constantes literales, es y delimitador, y comentarios
datos básicos (carácter, enteros, valores de verdad, cadenas de s): dominio y operaciones; ilidad; prioridades.
, constantes y expresiones
ones básicas: asignación (y), entrada y salida



Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

www.cartagenag9.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.



Compiación

en 3 partes
o (Dev-C ++)

www.bloodshed.net/devcpp.html

El programa tiene las siguientes fases:
1. Traducción del programa (.cpp, .cxx, .cc o .C) a lenguaje máquina.
2. Enlazado del código a lenguaje máquina.
3. Ejecución del objeto para producir un programa (.exe)

C++

Características del lenguaje C++

- Se distingue entre mayúsculas y minúsculas.
- Palabras clave: siempre en minúsculas.
- Lenguaje estructurado pero no estrictamente estructurado en bloques (no se pueden definir funciones dentro de otras funciones).
- Todas las sentencias y declaración de variables terminan en punto y coma.
- La ejecución siempre comienza con la función main().



C++

Estructura básica de un programa

- Un programa en C++ consta de uno o más archivos de texto, cuyas líneas forman:
 - Instrucciones, separadas por ;**
 - Declarativas**, definen variables, tipos, clases, funciones,...
 - Ejecutables**, son las que se convierten en código ejecutable: operaciones, sentencias de flujo, asignaciones, llamadas a funciones, etc...
 - Compuestas**, grupo de sentencias, encerrados entre `{ }`
 - Directivas**, información que le pasamos al compilador, llevan una `#` delante
 - Comentarios**, `//` una línea y `/* */` varias líneas
 - Expresiones**, combinación de constantes, variables, operadores, funciones y paréntesis. Todas las expresiones tienen un tipo



C++

de un programa

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

--

Programa

```

s de preprocesador
iones globales ( variables globales, funciones, ...)

main()

encia de declaraciones e instrucciones

)

encia de declaraciones e instrucciones

)

encia de declaraciones e instrucciones
    
```

C++

Mi primer programa "Hola Mundo!"

```

//Isabel Riomoros
// Este es mi primer programa
#include <iostream>
using namespace std;
int main()
{
    cout << "Hola Mundo!" << endl;
    system("pause");
    return 0;
}
    
```



Hola Mundo!
Pulse enter para continuar!

```

#include <iostream> using namespace
std;
int main(){cout<<"Hola Mundo!"<<endl;
system("pause");return 0;}
    
```

C++

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

Comentarios

en C++ **Comentario**- Propósito del programa

Directiva del preprocesador

Declaración del programa

<< **Operador de inserción de flujo**

\\ **carácter de escape**

\\n **línea nueva**

Indica que el programa termina de forma exitosa

Fin del programa

C++

Ejemplo: Intercambiar el valor de dos variables

```

#include <iostream>
using namespace std;

int main()
{
    int x, y;
    int aux;

    cin >> x >> y;

    aux = x;
    x = y;
    y = aux;

    cout << x <<y;

    return 0;
}
    
```

Directiva de preprocesamiento

Declaración de variables locales

Flujo de entrada

Flujo de salida por estándar (pantalla)

<< : operador de inserción para flujos de salida

Valor que se devuelve al S.O. Es la salida de la función main.

C++

na en el que te presentes a tus
decir que salga por pantalla.

n
)
do

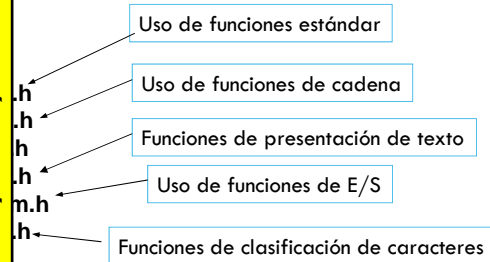
na que intercambie dos
carácter, sabiendo que una
e la denomina **char**

C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Preprocesador

becera estándar muy utilizados



Las directivas del preprocesador es la
ra, pero también se usan para definir macros,

C++

Directivas del preprocesador

- Los compiladores de C++ proporcionan bibliotecas de funciones.
- Cada biblioteca de funciones tiene asociada un archivo de definición que se denomina **cabecera**.

include <iostream.h>

Indica al compilador que lea las directivas antes de compilar la función principal

Si las librerías son heredadas de C se pone de forma obligatoria **.h**, o delante del nombre de la librería la letra **c**, p.e, **stdlib.h** o **cstdlib**

Las directivas más usuales son:

include
define



C++

Directivas del preprocesador

- Nombres de cabeceras
 - las viejas (de C o de C++ no estándar)
 - las nuevas (de C++ estándar que usan namespace std)

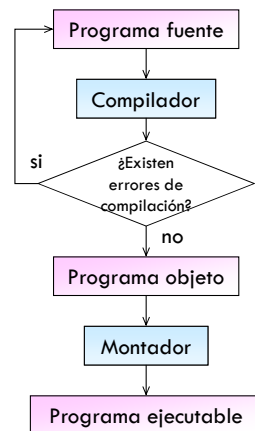
C C++ no estándar	C++ estándar (using namespace std)	
#include <stdlib.h>	#include <cstdlib>	Funciones de propósito general
#include <math.h>	#include <cmath>	Funciones matemáticas
#include <stdio.h>	#include <cstdio>	Manipular datos de E/S
#include <iostream.h>	#include <iostream>	Uso de funciones de E/S
#include <string.h>	#include <string>	Uso de funciones de cadena



C++

Fases en la ejecución de un programa

18



1. Escritura del programa fuente mediante un editor.
2. Traducir el programa mediante un compilador.
3. Verificar y corregir los errores de compilación.
4. Obtención del programa objeto.
5. Obtener el programa ejecutable mediante el montador.
6. Se ejecuta el programa y si no existen errores, se tendrá una salida.



C++

Palabras reservadas

20

- Son un conjunto de palabras que tienen un significado predeterminado para el compilador, y solo pueden ser utilizadas con dicho sentido.
- Se escriben en minúscula

Por ejemplo:

using, namespace, const, double, int, char, bool, void, for, while, do, if, switch, case, default, return, typedef, enum, struct, etc.



C++

Un subprograma que devuelve un valor o realiza una tarea específica. Tiene una única función main() que es la entrada al programa.

Las declaraciones escritas entre corchetes se denominan **bloque**.

Llamadas a otras funciones

```
#include <iostream.h>

int main()
{
    entrada_datos();
    proceso_datos();
    return 0;
}
```

Se ejecutan las funciones main() dentro del programa. Si hay un error, se muestra el mensaje de error.

C++

Palabras básicas

Las palabras reservadas son: palabras clave, identificadores, constantes literales, operadores y delimitadores, comentarios y espacios en blanco.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70



17



19



www.cartagena99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.



por el programador para representar tantes, variables, funciones, etc) en

te una secuencia de letras y dígitos, **ter una letra.**

dera como una letra, sin embargo, enzan con dicho carácter se reservan iales, por lo que no deberán is.

rar a:



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

dores, delimitadores

cen explícitamente en el programa, y méricos, caracteres y cadenas. , 25, 166.386, " Pts", ' ', etc.

do propio según el contexto en el que

/ % + - < > <= >= == != ++ -- . ,

comienzo o fin de una entidad.



Elección del identificador

- ❑ No pueden contener espacios en blanco, ni ciertos caracteres especiales como letras acentuadas, la letra eñe, las barras \ o /, etc.
- ❑ El compilador determina la máxima longitud que pueden tener (por ejemplo, 31 caracteres)
- ❑ Sensibles a mayúsculas y minúsculas.
- ❑ No se podrá dar a un dato el nombre de una palabra reservada.
- ❑ No es recomendable usar el nombre de algún identificador usado en las bibliotecas estándar (por ejemplo, cout)
- ❑ El identificador de un dato debe reflejar su semántica (contenido).
- ❑ Usaremos minúsculas, salvo la primera letra de nombres compuestos (y los identificadores de las constantes).
- ❑ No se nombrarán dos datos con identificadores que difieran únicamente en la capitalización, o un sólo carácter.



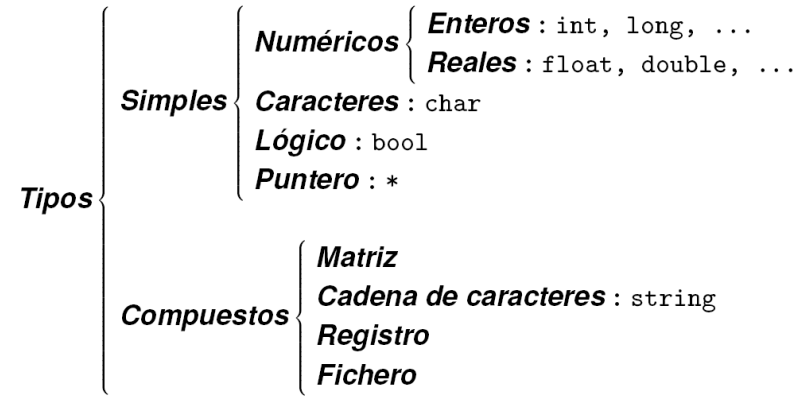
Comentarios y espacios en blanco

- ❑ Los espacios en blanco, tabuladores, nueva línea, retorno de carro, avance de pagina y los comentarios son ignorados por el compilador, excepto en el sentido en que separan elementos.
- ❑ Los comentarios en un programa es texto que el programador escribe para facilitar la comprensión, o remarcar algún hecho importante a quien lea nuestro programa, y son, por lo tanto, ignorados por el compilador.

Los comentarios en C++ se expresan de dos formas diferentes:

- ❑ **Comentarios hasta fin de línea:** los símbolos // marcan el comienzo del comentario, que se extiende hasta el final de la línea.
// acumula el siguiente numero
suma = suma + n; // acumula el valor de 'n'
- ❑ **Comentarios enmarcados:** los símbolos /* marcan el comienzo del comentario, que se extiende hasta los símbolos del fin del comentario */
/*
* acumula el siguiente numero
*/
suma = suma + n; /* acumula el valor de 'n' */





C++

Tipo entero: int

int <identificador>

Tamaño en bytes: 4 bytes (32 bits)
Dominio: son todos los números enteros entre los valores
 - 2.147.483.648 y 2.147.483.647

Operaciones:

+	Suma	int × int → int
-	Resta	
*	Producto	
/	División entera	
%	Resto de la división entera (módulo)	
- , +	Signo negativo, positivo	int → int
++	Incrementación	
--	Decrementación	

Otros enteros

- unsigned char
- char (signed)
- short (signed)
- unsigned short
- unsigned (int)
- int (signed)
- unsigned long
- long (int)



C++



Datos básicos

carácter,
 enteros,
 reales,
 valores de verdad
 cadenas de caracteres
 incompatibilidad de tipos

Datos básicos

Define la naturaleza del valor que puede tomar un tipo de dato define un *dominio de valores* que se pueden realizar con éstos

Existen tantos tipos de datos predefinidos que el programador crear otros tipos de datos

Tipos de datos básicos

Tipos de datos básicos
 (Tipos de datos más grandes que float)
 (Tipos de datos de cualquier cantidad de 8 bits)
 (Se usa & para indicar que una función no devuelve valores)

C++

CLASES PARTICULARES, TUTORIAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



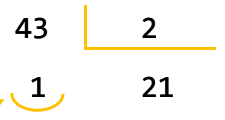


ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70
 CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

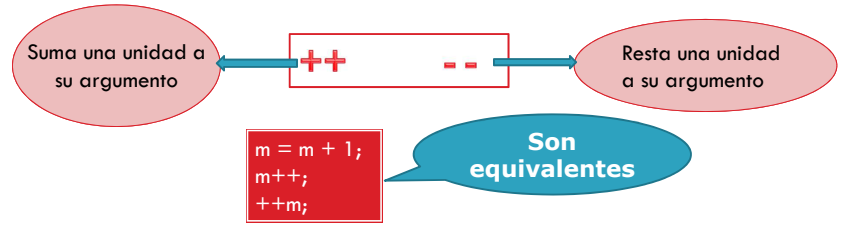
ite y resto de la división

```
2;  
// Muestra 21
```

males → Queda un resto



Operadores de incremento y decremento



- Si precede al operando se realiza la operación ++ o -- y luego se realiza la asignación (**Prefijo**)
- Si sigue al operando, se realiza la asignación y posteriormente se realiza la operación ++ o - (**Postfijo**)

```
x = 10;  
y = ++x; // y vale 11
```

```
x = 10;  
y = x++; // y vale 10
```



float <identificador>

bytes
 los números reales que contienen una
 endidos entre los valores:
 10^{-38} y $3,4 \times 10^{38}$

float × float → float

coma flotante
 o, positivo
 on
 ón

float → float



double <identificador>

Tipo real: double

- Tamaño en bytes: 8 bytes**
- Dominio:** son todos los números reales que contienen una coma decimal comprendidos entre los valores:

$$1,7 \times 10^{-308} \quad \text{y} \quad 1,7 \times 10^{308}$$

- Operaciones:**

+	Suma	double × double → double
-	Resta	
*	Producto	
/	División en coma flotante	
- , +	Signo negativo, positivo	
++	Incrementación	
--	Decrementación	double → double



división real?

166 → División entera
166.667 → División real

Resultado

166
166.667
166.667
166.667

la división es el que quieres



Tipo real

Además de éstas operaciones, C++ dispone de un gran conjunto de **funciones matemáticas**.

Funciones

abs: int → int	Calcula el valor absoluto de un número
ceil: double → double	Calcula el número entero mayor o igual que el dado
floor: double → double	Redondea por defecto el valor de un número
fmod: double × double → double	Calcula el resto de la división real de dos números
sqrt: double → double	Calcula la raíz cuadrada de un número
pow: double × double → double	Calcula la potencia de un número

```
#include <math.h>
{
x = abs(-7)           // x vale 7
y = ceil (5.2)       // y vale 6
z = floor (5.2)      // z vale 5
resto = fmod(5.0, 2.0) // resto vale 1
}
```

math.h
float.h
complex.h



la centésima cifra=3

```
* pow(10,cifras))/pow(10,cifras);
- 'fabs(7.5): Valor absoluto'"<<endl;
- 'floor(7.5): Redondea hacia abajo'"<<endl;
- 'fabs(0.0) Valor absoluto'"<<endl;
- 'ceil(0.0): Redondea hacia arriba'"<<endl;
- 'fabs(-6.4): Valor absoluto'"<<endl;
- 'ceil(-6.4): Redondea hacia arriba'"<<endl;
+ floor(-5.5))<<" - ceil(-fabs(-8 + floor(-
```



Ceil (2.3)= 3.0
Ceil(3.8)= 4.0
Ceil(-2.3)= -2.0
Ceil(-3.8)=-3.0

```
F:\c++ clases\programas\funcionesMath.exe
x = 7.5 - 'fabs<7.5>: Valor absoluto'
x = 7 - 'floor<7.5>: Redondea hacia abajo'
x = 0 - 'fabs<0.0> Valor absoluto'
x = 0 - 'ceil<0.0>: Redondea hacia arriba'
x = 6.4 - 'fabs<-6.4>: Valor absoluto'
x = -6 - 'ceil<-6.4>: Redondea hacia arriba'
x = -14 - 'ceil<-fabs<-8 + floor<-5.5>>'
Presione una tecla para continuar . . .
```



ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

Rango de los enteros y los reales

38

Tipo	bits	Rango / Tipo de uso	Constante
unsigned char	8	$0 \leq X \leq 255$	UCHAR_MAX
char (signed)	8	$-128 \leq X \leq 127$	SCHAR_MIN; SCHAR_MAX
short (signed)	16	$-32.768 \leq X \leq 32.767$	SHRT_MIN; SHRT_MAX
unsigned short	16	$0 \leq X \leq 65.535$	USHRT_MAX
unsigned (int)	32	$0 \leq X \leq 4.294.967.295$	UINT_MAX
int (signed)	32	$-2.147.483.648 \leq X \leq 2.147.483.647$	INT_MIN; INT_MAX
unsigned long	32	$0 \leq X \leq 4.294.967.295$	LONG_MAX
long (int)	32	$-2.147.483.648 \leq X \leq 2.147.483.647$	LONG_MIN; LONG_MAX
float	32	$1.18e-38 \leq X \leq 3.40e38$	
double	64	$2.23e-308 \leq X \leq 1.79e308$	
long double	80	$3.37e-4932 \leq X \leq 1.18e4932$	

En limits.h están definidas las constantes que delimitan los rangos de los enteros. #include <limits.h>

C++



antes en math.h

```

seno, sin r (en radianes)
coseno, cos r (en radianes)
tangente, tan r (en radianes)
arco seno, arcsin x, x ∈ [-1, 1]
arco coseno, arc cos x, x ∈ [-1, 1]
arco tangente, arctan x
arco tangente, arctan y/x
seno hiperbólico, sinh r
coseno hiperbólico, cosh r
tangente hiperbólica, tanh r
√x, x ≥ 0
xy
ex
logaritmo neperiano, ln x, x > 0
logaritmo decimal, log x, x > 0
menor entero ≥ x, [x]
mayor entero ≤ x, [x]
valor absoluto de x, |x|
x2n
inversa de ldxp
parte entera y fraccionaria
resto de x/y
    
```

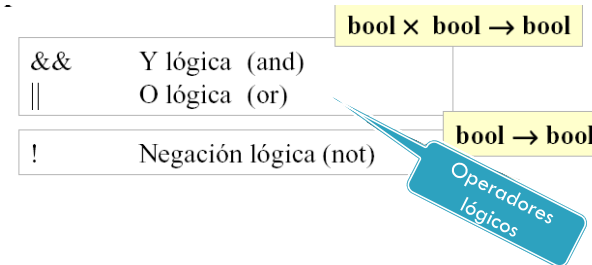
C++

Tipo booleano: bool

bool <identificador>

40

- ❑ **Tamaño en bytes:** 1 byte
- ❑ **Dominio:** dos únicos valores: { true, false }
- ❑ **Operaciones:**



- ❑ No todos los compiladores de C++ tienen éste tipo de dato

Falso → Cero
Verdadero → Distinto de cero

C++



C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99

37

39

```

C:\Programas\pruebaConstantes.exe
7483647
7483647
35
67
para continuar . . .
    
```



Tipo booleano: bool

Escritura de valores bool

- Raramente se tiene la necesidad de escribir o leer valores de tipo bool ya que éste tipo de datos se utiliza sobre todo para evaluar expresiones lógicas.
- En caso necesario, si escribimos un dato de tipo bool cuyo valor es true, en consola se visualiza el valor 1. Para visualizar **true** podemos utilizar **boolalpha**.

```
#include <iostream>
using namespace std;
int main () {
    bool b=true;
    cout << boolalpha << b << endl;
    cout << noboolalpha << b << endl;
    return 0; }
```

true
1

- La lectura es análogo.



C++

Tipo carácter: char

char <identificador>

- Tamaño en bytes:** 1 byte
- Dominio:** dígitos, letras mayúsculas, letras minúsculas y signos de puntuación. Tipos de datos básicos en C++
 $0 < 1 < 2 \dots < 9 < A < B < \dots < Z < a < b < \dots < z$
- Internamente, los caracteres se almacenan como números.
- El tipo **char** representa valores en el rango -128 y 127 y se asocian con el código ASCII.
 - Así, el carácter 'A' se almacena como el número 65, etc ...
- Operaciones:**
Dado que los caracteres se almacenan internamente como números enteros, se pueden realizar operaciones aritméticas con los datos de tipo char. Se puede sumar un entero a un carácter para obtener otro código ASCII diferente.
- Ejemplos:**
 - Para convertir una letra minúscula en mayúscula basta con restar 32.
'a' - 32 = 'A'
 - Para convertir una letra mayúscula en minúscula basta con sumar 32.
'B' + 32 = 'b'
 - Para convertir el carácter '4' en el número 4 basta con restar 48.
'4' - 48 = 4



C++

	! A	A && B	A B
	F	T	T
	F	F	T
	T	F	T
	T	F	F

es

a
to
r que
r que
r o igual que
r o igual que

C++

anas para indicar:

probar si forman un triangulo
ro de tres cifras es capicúa

C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

41

43

Tabla ASCII

Standard Code for Information Interchange

- La tabla ASCII es comúnmente utilizada como base para la representación de los caracteres, donde los números del 0 al 31 se utilizan para representar caracteres de control, y los números del 128 al 255 se utilizan para representar caracteres extendidos.

Rep	Simb	Rep	Simb	Rep	Simb	Rep	Simb
0	\0	32	SP	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	\a	39	'	71	G	103	g
8	\b	40	(72	H	104	h
9	\t	41)	73	I	105	i
10	\n	42	*	74	J	106	j
11	\v	43	+	75	K	107	k
12	\f	44	,	76	L	108	l
13	\r	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

C++



Prioridad de operadores

Operador	Tipo de Operador	Asociatividad
! ~ -	Unarios	Dch. a Izq.
* / %	Binarios	Izq. a Dch.
+ -	Binarios	Izq. a Dch.
<< >>	Binarios	Izq. a Dch.
< <= > >=	Binarios	Izq. a Dch.
== !=	Binarios	Izq. a Dch.
&	Binario	Izq. a Dch.
^	Binario	Izq. a Dch.
	Binario	Izq. a Dch.
&&	Binario	Izq. a Dch.
	Binario	Izq. a Dch.
?:	Ternario	Dch. a Izq.

C++



Cartagena99

Devuelve TRUE si el carácter es: '0', ... , '9'
 Devuelve TRUE si el carácter es: 'A', ... , 'Z', 'a', ... , 'z'.
 e TRUE si el carácter es una letra minúscula: 'a', ... , 'z'.
 e TRUE si el carácter es una letra mayúscula: 'A', ... , 'Z'.
 vierte un carácter mayúscula en minúscula.
 vierte un carácter minúscula en mayúscula.

char c = 65;

El archivo de cabecera que contiene éstas funciones es: **ctype.h**

ale 'a'
 ale O (FALSE)

C++

dores

= 10*6

C++



ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70
 CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70



Una fórmula

```
#include <iostream>
using namespace std;
```

$$f(x) = \frac{4x^2}{6} + \frac{9x}{5} - 8$$

```
int main()
{
    double x, f;
    cout << "Introduce el valor de X: ";
    cin >> x;
    f = 4 * x * x / 6 + 9 * x / 5 - 8;
    cout << "f(x) = " << f << endl;
    return 0;
}
```

Usa paréntesis para mejorar la legibilidad:
`f = (4 * x * x / 6) + (9 * x / 5) - 8;`



C++

Expresiones

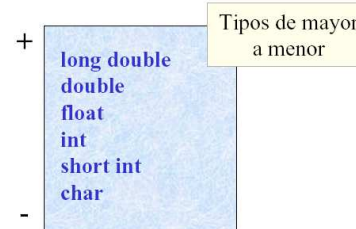
```
- (3 * 2 - 1)
  ↓
(6 - 1)
  ↓
  5
```

Pon espacio antes y después de cada operador binario

C++

Modificadores de tipos de datos

- Los ordenadores realizan numerosas operaciones para la resolución de problemas,
 - Operaciones aritméticas y lógicas.
 - Operaciones condicionales.
- Además, puede ocurrir que en una misma expresión concurren varios tipos de datos. Ante ésta situación, debemos saber cómo se comporta el compilador.
- Cuando los dos operandos son de tipos distintos, el de tipo menor se promociona al de tipo mayor.**



C++

Tipos de datos

double y char tienen variaciones o tipos de datos, permiten un uso más flexible de los datos.

Siguientes: de los tipos de datos `signed - short - long`

```
0 ... 65625
3,37 × 10-4932 ... 3,37 × 104932
-2147483648 ... 2147483647
```

C++



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

Conversión de tipos

- Cuando se opera con datos de diferente tipo,
 - ▢ ambos son convertidos al tipo de dato de precisión más alta
 - ▢ El resultado es convertido al tipo de datos que lo almacena

Por ejemplo:

```
int x, z; double y;  
z=x+y;
```

1. convierte x en double
2. Se realiza la suma entre doubles
3. Se guardará la parte entera del resultado en z

No es conveniente dejar en manos del compilador la conversión de tipos, el programador puede forzar a conversión

p.e. (int) y

(nuevoTipoDeDatos) variableAconvertir



C++

56

Variables, constantes y expresiones



Trabaja con los tipos de datos

```
#include <stdio.h>  
#include <iostream.h>  
  
1 int main()  
2 {  
4     int i;  
8     i = sizeof( int ) * 8;  
1  
  
    cout << "Tamaño (en bits) del tipo int = " ;  
    cout << i;  
    return 0;  
}
```

memoria que ocupan los tipos de datos

C++

Conversión de tipo:

La expresión se trata como

```
Muestra 1 (división entera)  
int a, b; // Muestra 1.5 (división real)
```

C++



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

53

55

www.cartagenan99.com no se hace responsable de la información contenida en el presente documento en virtud al Artículo 17.1 de la Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico, de 11 de julio de 2002. Si la información contenida en el documento es ilícita o lesiona bienes o derechos de un tercero háganoslo saber y será retirada.

Constantes

58

```
const <tipo_de_dato> <nombre_de_constante> = <valor>;
```

```
...  
const int DIAS = 7;  
const char VACIO = ' ' ;  
const char PORCENTAJE = '% ' ;
```

Ejemplo

- Constantes cadenas de caracteres literales, la secuencia de caracteres aparece entre comillas dobles
"Hola Pepe"
"Hola\nJuan\n"
"Hola " "Mara"



C++

Constantes

60

- Constantes enteras, pueden ser expresadas en
 - decimal (base 10)
 - hexadecimal (base 16)
 - octal (base 8).
 - El sufijo **L** se utiliza para especificar **long**, el sufijo **LL** se utiliza para especificar **long long**, el sufijo **U** se utiliza para especificar **unsigned**, el sufijo **UL** especifica **unsigned long**, y el sufijo **ULL** especifica **unsigned long long**:
123, -1520, 2345U, 3000L, 5000UL, 0x10B3FC23 (hexadecimal), 0751 (octal)
- Constantes reales, números en punto flotante. El sufijo **F** especifica float, y el sufijo **L** especifica long double:
3.1415, -1e12, 5.3456e-5, 2.54e-1F, 3.25e200L



C++

Cartagena99

57

objeto cuyo valor no puede ser

Constantes se suelen escribir en mayúsculas

no aparecen como
en aquellas cuyo valor aparece directamente

en aquellas cuyo valor se asocia a un
símbolo el cual se representa

pool):

(char), el símbolo constante aparece

C++

59

ante la directiva **#define**

```
<nombre_de_constante> <valor>
```

No se especifica
el tipo de dato

No aparece ;
al final de la sentencia

No aparece el
símbolo =

const en lugar de **#define** ya que el compilador
genera código más eficiente.

C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70



as se declaran indicando la palabra por su tipo, el nombre simbólico (o los referiremos a ella y el valor asociado

mbólicas:

```
e;
= '\a';
ITO = 1000;
: 5000;
= 100000L;
IANO = 1000000LL;
t VALOR = 100U;
AS = 200U;
COLUMNAS = 200UL;
long NELMS = 2000ULL;
7182F;
DE = log(N_E);
_PI = 3.141592L;
DEFECTO = ROJO;
```



Las variables

- Una variable es un espacio reservado en la memoria del ordenador para contener valores que pueden cambiar durante la ejecución de un programa
- Los tipos determinan cómo se manipulará la información contenida en esas variables
- El tipo nos dice a nosotros y al compilador cómo debe interpretarse y manipularse la información binaria almacenada en la memoria de un ordenador
- Una declaración de variables es una instrucción, puede aparecer entremezclada con otras instrucciones
- La variable declarada existe inmediatamente después de la declaración hasta el momento en que se acabe el bloque
- Recién declarada tiene un valor indefinido, es recomendable definirla lo más próximo al lugar dónde se va a utilizar y darle un valor inicial
- Se suele escribir en minúscula



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

programa debe ser declarada previamente. En situarse en cualquier parte del programa.

- Variables globales
- Variables locales
- Parámetros

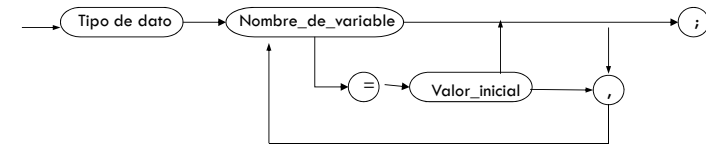
```
variable> ;
ables> ;
variable> = valor;
```

int x;
char x, y, z;
long int i=10, j, k=0;

as variables no se automáticamente



VARIABLES



Declaraciones locales

Son variables locales aquellas que están declaradas dentro de las funciones o de los bloques.

Declaraciones globales (variables globales, funciones, ...)

La zona de declaraciones globales de un programa puede incluir declaraciones de variables y declaraciones de funciones (prototipos).



Variables

```
// -- Principal -----
int main ()
{
    bool logico = false;
    char caracter = 'z';
    unsigned natural_1, natural_2;
    int entero = 56;
    double real;
    // Acciones
}
```

C++

Operadores y expresiones

Instrucción de asignación
Operador de dirección
Referencias



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Instrucción de asignación

<variable> = <expresión>;

Puede ser otra variable, una constante o una operación entre variables y constantes.

La instrucción de asignación (=) asigna el valor de la expresión situada en la izquierda de la instrucción.

Existen además varios operadores de asignación:

= *= /= %=

C++

Instrucciones de asignación

Ejemplos:

<code>m = n;</code>	<code>// asigna el valor de n a m</code>
<code>m += n;</code>	<code>m = m + n; //suma m y n y lo asigna a la variable m</code>
<code>m -= n;</code>	<code>m = m - n; // resta m menos n y lo asigna a la variable m</code>
<code>m *= n;</code>	<code>m = m * n; //multiplica m por n y lo asigna a la variable m</code>
<code>m /= n;</code>	<code>m = m / n; //divide m entre n y lo asigna a la variable m</code>
<code>m %= n;</code>	<code>m = m % n; //calcula el resto de la div. entera y lo asigna a la variable m</code>

Instrucción abreviada.



C++

Asignación

Asignar valores a varias variables a la vez

Asignar valores a las variables m, n y t el valor 10

Las variables m, n y t toman el valor de la variable a

Asignar a varias variables el valor de otra de un sólo golpe

Ejemplos

70

```
// -- Principal -----
int main ()
{
    int x1 = 4 + 2 * 6; // Inicialización
    x1 = x1 * 3 % 2 - 3; // Asignación
    ++x1;           // x1 = x1 + 1;
    --x1;          // x1 = x1 - 1;
    x1 += 5;       // x1 = x1 + 5;
    x1 -= 3;       // x1 = x1 - 3;
    x1 *= 2 + 5;   // x1 = x1 * (2 + 5);
    x1 /= 2;       // x1 = x1 / 2;
    x1 %= 2;       // x1 = x1 % 2;
}
```

```
#include <iostream>
#include <string>
using namespace std;
// -- Principal -----
int main ()
{
    char letra;
    cin >> letra;
    int valor = int(letra) + 1;
    letra = char(valor);
    letra = char(letra + 1);
    cout << letra << endl;
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Operadores de Entrada / Salida

Instrucciones de Entrada / Salida

- En C++ la entrada y salida se lee y escribe **en flujos**. Cuando se incluye la biblioteca **iostream** en el programa, se definen automáticamente dos flujos:

Flujo **cin** (se utiliza para la entrada de datos)

Permiten la comunicación del ordenador con el exterior para tomar datos o devolver resultados

Flujo **cout** (se utiliza para la salida de datos)

- Esta biblioteca también nos proporciona dos operadores, uno de *inserción* (**<<**), que inserta datos en el flujo **cout** y otro operador de extracción (**>>**) para extraer valores del flujo **cin** y almacenarlos en variables.

```
...
cin >> a;
cin >> a >> b >> c;
...
```

```
...
cout << x;
cout << x << y << z << endl;
cout << " x vale: " << x;
cout << "Hola\n";
...
```

Salto de línea

\n también provoca salto de línea



Para evitar errores, lee cada dato en una instrucción aparte

71

Dividir dos enteros

```
//Sumar dos numeros enteros
#include<iostream>
using namespace std;
int main(){
    int num1,num2,division;
    cout << "Introduce un número entero ";
    cin >> num1;
    cout << endl<<"Introduce un numero entero ";
    cin >> num2;
    producto = num1/num2;
    cout << "\n El producto de "<<num1<<" x "<<num2<<"
        es"<<suma<<endl;
    system("pause");
    return 0;
}
```



C++

Salida formateada

- **setprecision().** Para indicar el número de dígitos significativos en un dato en punto flotante. Afecta a todos los datos que se introduzcan con posterioridad.
- **setw().** Permite indicar el número de espacios que se emplearán para escribir un dato, alineando al mismo a la derecha dentro de dicho espacio.
 - Si el espacio requerido es mayor que el indicado, el modificador se ignora. Sólo afecta al dato que se indica a continuación.
- **setfill().** Para especificar el carácter de relleno que se empleará para los espacios no usados al escribir un dato (según un modificador setw()).



C++

73

lida

se ajustan a un sencillo esquema:



resultado

la programación: Tipos e instrucciones I

C++

hasta pulsar un carácter

hasta pulsar una tecla

getline() Lee todo hasta el siguiente carácter de nueva línea y coloca el resultado en la variable especificada. Lee líneas de texto que pueden contener espacios en blanco.

cout << "\n";

cout << frase;

C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

Ejemplo setw y setfill

```
#include <iostream> // para operaciones de E/S
#include <iomanip> // define diferentes manipuladores
using namespace std;
int main()
{
    cout << setfill('.'); // rellenar con puntos
    cout << "Lista de notas\n" << endl;
    cout << "Julio Iglesias" << setw(20) << "5" << endl;
    cout << "Shakira Pérez" << setw(21) << "8" << endl;
    cout << "Alejandro Sanz" << setw(20) << "7" << endl;
    cout << "Jarabe de Palo" << setw(20) << "5" << endl;
    cout << "Jhon Travolta" << setw(21) << "8" << endl;
    cout << "Enrique Bunbuny" << setw(19) << "9" << endl;
    cout << setfill('\0'); // se restablece el carácter de llenado
    cout << setw(20) << "Fin" << endl;

    return 0;
}
```



C++

Salida formateada

SALIDA FORMATEADA DE DATOS

formato aplicado a la salida de datos. incluir la biblioteca estándar **iomanip**

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;
// -- Principal -----
int main ()
{
    cout << boolalpha; // escribe los valores booleanos como 'false' o 'true'
    cout << dec << 27; // escribe 27 (decimal)
    cout << hex << 27; // escribe 1b (hexadecimal)
    cout << oct << 27; // escribe 33 (octal)
    cout << setprecision(2) << 4.567; // escribe 4.6
    cout << setw(5) << 234; // escribe 234
    cout << setfill('#') << setw(5) << 234; // escribe ##234
}
```



C++

on

para operaciones de E/S
define diferentes manipuladores

able2=7;

able2 << endl;

```
G:\Isabel\Programacion\programas\setprecision.exe
3
3
3.1
3.14
3.143
3.1429
3.14286
3.142857
3.1428571
3.14285714
3.142857143
3.1428571429
3.14285714286
3.142857142857
3.1428571428571
```

C++

fill

```
manipuladores2.exe
5
8
3
5
8
9
e 0
```

C++

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70



Entrada / Salida

de escape para visualizar caracteres dos por los símbolos tradicionales.

Mostramos en la siguiente tabla:

de carro y avance de línea
ón
omillas
linada

```

"an";
es\t Martes\t Miércoles\t ";
;
    
```

Generar números aleatorios

82

- Activar el generador de números aleatorios
srand(time(NULL));

```
variable = limiteInferior + rand() % (limiteSuperior + 1 -
limiteInferior);
```

Ejemplos

- Número aleatorios entre **0 y 50**:
num=rand()%51;
- Número aleatorios entre **1 y 100**:
num=1+rand()%(101-1);
- Número aleatorios entre **250 y 420**:
num=250+rand()%(421-250);



ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

Errores en un programa

83

- **Errores en tiempo de compilación.** Ocasionados por un fallo de sintaxis en el código fuente. No se genera el programa ejecutable.

```

/* CONTIENE ERRORES */
#include <iostream>
USING namespace std;
int main(){
    double lado1;
    double lado 2,
    double hip:
    lado1 = 2;
    lado2 = 3
    hip = sqrt(lado1**lado1 + ladv2*ladp2);
    cout >> "La hipotenusa vale >> hip;
)
    
```



Compilador

Programa Ejecutable: Pitagoras.exe

```

10011000010000
10010000111101
00110100000001
11110001011110
11100001111100
11100101011000
00001101000111
00011000111100
    
```

```

#i
us:
in:
d:
c:
c:
.
}
    
```





rama

e ejecución. Se ha generado el e, pero se produce un error 1.

ntero;



Errores en un programa

Errores lógicos. Se ha generado el programa ejecutable, pero el programa ofrece una solución equivocada.

```
.....
lado1 = 4;
lado2 = 9;
hip = sqrt(lado1+lado1+lado2*lado2);
.....
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

características del tipo char)

```
isalpha(ch) || isdigit(ch)
isupper(ch) || islower(ch)
caracteres de control
dígito decimal
caracteres imprimibles excepto espacio
letra minúscula
caracteres imprimibles incluyendo espacio
carac. impr. excepto espacio, letra o dígito
espacio, '\r', '\n', '\t', '\v', '\f'
letra mayúscula
dígito hexadecimal
retorna la letra minúscula correspondiente a ch
retorna la letra mayúscula correspondiente a ch
```



Biblioteca cstdlib

```
#include <cstdlib>
using namespace std;

int system(const char orden[]); orden a ejecutar por el sistema operativo
int abs(int n); retorna el valor absoluto del número int n
long labs(long n); retorna el valor absoluto del número long n
void srand(unsigned semilla); inicializa el generador de números aleatorios
int rand(); retorna un aleatorio entre 0 y RAND_MAX (ambos inclusive)

#include <cstdlib>
#include <ctime>
using namespace std;
// -----
// inicializa el generador de números aleatorios
inline unsigned ini_aleatorio()
{
    srand(time(0));
}
// -----
// Devuelve un número aleatorio entre 0 y max (exclusive)
inline unsigned aleatorio(unsigned max)
{
    return unsigned(max*double(rand()/(RAND_MAX+1.0)));
}
// -----
// Devuelve un número aleatorio entre min y max (ambos inclusive)
inline unsigned aleatorio(unsigned min, unsigned max)
{
    return min + aleatorio(max-min+1);
}
// -----
```



Datos de una persona

```
#include <iostream>
using namespace std;
int main() {
    char Nombre[30];
    int Edad;
    char Telefono[9];
    cout << "Introduce tu nombre, edad y número de teléfono" << endl;
    cin >> Nombre >> Edad >> Telefono;
    cout << "Nombre:" << Nombre << endl;
    cout << "Edad:" << Edad << endl;
    cout << "Teléfono:" << Telefono << endl;
    return 0;
}
```



C++

Error sintáctico

```
#include <iostream>
#include <string>
using namespace std;
// -- Principal -----
int main ()
{
    int dividendo, divisor;
    int cociente;
    cin >> dividendo >> divisor;
    cociente = / dividendo divisor;
    cout << cociente << endl;
}
```

ERROR SINTÁCTICO



C++

ces

Edición. Prentice-Hall. 2000.
Encontrada en <http://www.bruceeckel.com/>

<http://www.cplusplus.com/resources/Informat1/AyudaInf/Index.htm#lenguajes>

www.cplusplus.com

www.cplusplus.com

<http://www.cppreference.com>

<http://www.zator.com/Cpp/>

C++

92

89

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

--

```
...eros: ";
... >> (x2 * 3) << endl;
```

C++

91



Error semántico

94

```
#include <iostream>
#include <string>
using namespace std;
// -- Principal -----
int main ()
{
    int dividendo, divisor;
    const int cociente = 3;
    cin >> dividendo >> divisor;
    cociente = dividendo / divisor;
    cout << cociente << endl;
}
```

ERROR SEMÁNTICO



C++

Pasar una cantidad de pesetas a euros

96

```
//- fichero: euros.cpp -----
#include <iostream>
#include <cstdlib>
using namespace std;
const double EUR_PTS = 166.386;
int main()
{
    cout << "Introduce la cantidad (en euros): ";
    double euros;
    cin >> euros;
    double pesetas = euros * EUR_PTS;
    cout << euros << " Euros equivalen a " << pesetas << " Pts" << endl;
    system("pause");
    return 0;
}
//- fin: euros.cpp -----
```



C++

ejecución

93

```
#:
#:
us
//
ir
{
}
}
```

RT_ERROR: DIVISIÓN POR CERO

C++

```
) && (num <= 999);
gitos && par;
= 100) && (num <= 999)) && ((num % 2) == 0);
(num <= 3)) || (num == 5) || (num == 7);
((10 % num) == 0);
```

C++

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

--

s

95



Dada una letra mayúscula convertirla a minúscula

```
#include <iostream>
#include <string>
using namespace std;
// -- Principal -----
int main ()
{
    char letra_mayuscula;
    cin >> letra_mayuscula;
    // Suponemos que es una letra Mayúscula
    int distancia = int(letra_mayuscula) - int('A');
    char letra_minuscula = char(int('a') + distancia);
    // letra_minuscula = char(int('a') + (int(letra_mayuscula) - int('A')));
    // letra_minuscula = char('a' + (letra_mayuscula - 'A'));
    cout << letra_mayuscula << " -> " << letra_minuscula << endl;
}
```



C++

o dado es o no par

```
int x;
int y;
if (x % 2 == 0);
    cout << "Es par: " << boolalpha << par << endl;
```

C++

```
#include <iostream>
using namespace std;
int main ()
{
    int x, y;
    char c;
    cout << " Introduzca dos numeros naturales : ";
    cin >> x >> y;
    cout << " Introduzca un caracter : ";
    cin >> c;
    bool prop_a = (x >= 3 && x <= 7);
    bool prop_b = (x >= 1 && x <= 3) || (x >= 7 && x <= 9);
    bool prop_c = (x >= 1 && x <= 9) && (x % 2 == 1);
    bool prop_d = (x == 2) || (x >= 5 && x <= 9);
    bool prop_e = ((x >= 3 && x <= 9 && x % 2 == 1) && ((y >= 6 && y <= 8) || y == 3));
    bool prop_f = (x <= 10 && y <= 10);
    bool prop_g = ! (y != 0 && x % y == 0);
    bool prop_h = (c >= 'A' && c <= 'Z');
    bool prop_i = (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z');
    bool prop_j = ((c >= 'a' && c <= 'z') || (c >= '0' && c <= '9'));
    cout << boolalpha;
    cout << "(a) " << x << " pertenece a { 3, 4, 5, 6, 7 } : " << prop_a << endl;
    cout << "(b) " << x << " pertenece a { 1, 2, 3, 7, 8, 9 } : " << prop_b << endl;
    cout << "(c) " << x << " pertenece a { 1, 3, 5, 7, 9 } : " << prop_c << endl;
    cout << "(d) " << x << " pertenece a { 2, 5, 6, 7, 8, 9 } : " << prop_d << endl;
    cout << "(e) " << x << " pertenece a { 3, 4, 6, 8, 9 } : " << prop_e << endl;
    cout << "(f) " << x << " pertenece a { 6, 7, 8, 3 } : " << prop_f << endl;
    cout << "(g) " << x << " Ni " << x << " ni " << y << " sean mayores que 10 : " << prop_g << endl;
    cout << "(h) " << x << " no sea multiplo de " << y << " : " << prop_h << endl;
    cout << "(i) " << c << " es una letra mayuscula : " << prop_i << endl;
    cout << "(j) " << c << " es una letra : " << prop_j << endl;
    cout << "(k) " << c << " es un alfanumerico : " << prop_k << endl;
}
```



C++

Introduzca dos números enteros (x e y) escriba true si cumplen las siguientes condiciones, en caso contrario:

- (a) x pertenece a { 3; 4; 5; 6; 7 }
- (b) x pertenece a { 1; 2; 3; 7; 8; 9 }
- (c) x pertenece a { 1; 3; 5; 7; 9 }
- (d) x pertenece a { 2; 5; 6; 7; 8; 9 }
- (e) x pertenece a { 3; 4; 6; 8; 9 }
- (f) x e y pertenecen a { 6; 7; 8; 3 }
- (g) Ni x ni y sean mayores que 10
- (h) x no sea múltiplo de y
- (i) c sea una letra mayúscula
- (j) c sea un carácter alfanumérico (letra o dígito)

C++

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



```

#include <iostream.h>
#include <stdlib.h>
Using namespace std;
int main()
{
// Otra manera de definir constantes
// CONST float PI=3.1415;

float altura, base, radio;

cout << endl << "Introduzca la base del triangulo " << endl;
cin >> base;
cout << endl << "Introduzca la altura del triangulo " << endl;
cin >> altura;
cout << endl << "El area del triangulo es " << base*altura/2;

cout << endl << "Introduzca el radio del circulo " << endl;
cin >> radio;
cout << endl << "El area del circulo es " << PI*radio*radio << endl;

system("PAUSE");
return 0;
}

```

102

```

primer numero: " << endl;

segundo numero: " << endl;

< a+b;
<< a-b;
>n es: " << a*b;
>era es: " << a/b;
>division entera es: " << a%b << endl;

```

Número de tres cifras es

```

) && (numero <= 999) && (digito_1 == digito_3);
tos y es capicua: " << boolalpha << tres_cap << endl;

```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

101

103

