

CARACTERISTICAS PRINCIPALES DEL LENGUAJE C

- Se puede usar para codificar tanto en alto como en bajo nivel**
- Tiene una gran variedad de operadores y tipos de datos**
- No esta orientado a ningún área en particular, lo que lo hace útil para codificar cualquier problema**
- Esta diseñado para permitir la programación estructurada**
- Facilidad para aprenderlo**

EDITAR UN PROGRAMA EN C

Para editar un programa en C, se debe usar alguno de los tantos editores de C disponibles, nosotros vamos a usar uno llamado Dev-C++:



Esta aplicación permite efectuar las 3 etapas necesarias para producir un programa:

- **Edición del programa**
- **Compilación del programa y generación de archivo ejecutable**
- **Ejecución del programa, prueba y optimización del mismo**

TIPOS DE DATOS EN C

Tipos de datos enteros

Nombre	Rango de valores
<i>char</i>	-128 a 127
<i>unsigned char</i>	0 a 255
<i>int</i>	-32768 a 32767
<i>unsigned int</i>	0 a 65535
<i>long</i>	-2147483648 a 2147483647
<i>unsigned long</i>	0 a 4294967295

Tipos de datos reales

Nombre	Rango de valores
<i>float</i>	-3,402823E+38 a 3,402823E+38
<i>double</i>	-1,79769E+308 a 1,79769E+308
<i>long double</i>	-1,189731E+4932 a 1,189731E+4932

Funciones matemáticas

Las declaraciones para las funciones matemáticas que a continuación se describen, están en la librería a incluir **math.h**

Quiere decir que cuando se utilice una función matemática en un programa, debe especificarse el comando:

```
#include<math.h>
```

Los argumentos para estas funciones son de tipo **double** y el resultado devuelto es también de tipo **double** (números reales de precisión doble)

FUNCIONES MATEMATICAS PREDEFINIDAS EN C

NOMBRE FUNCION	ARGUMENTO/S	EJEMPLO
acos (arco coseno) Da el arco coseno en el intervalo $(0,\pi)$	x, real doble precisión	y=acos(-0.7) Da y=2.346 rad
asin (arco seno) Da el arco seno en el intervalo $(-\pi/2,\pi/2)$	x, real doble precisión	y=asin(0.5) Da y=0.523 rad
atan (arco tangente) Da el arco tangente en el intervalo $(-\pi/2,\pi/2)$	x, real doble precisión	y=atan(1) Da y=0.785
atan2 (arco tangente de y/x) Da el arcotangente del cociente de y/x en el intervalo $(-\pi,\pi)$	x, y: reales doble precisión	Θ =atan2(y,x) Con x=3, y=4 da Θ =0.927 rad

NOMBRE FUNCION	ARGUMENTO/S	EJEMPLO
cos (coseno)	x, real doble precisión	y=cos(2) Da y=-0.416
sin (seno)	x, real doble precisión	y=sin(1.5) Da y=0.997
tan (tangente)	x, real doble precisión	y=tan(2) Da y=-2.185
cosh (coseno hiperbólico)	x, real doble precisión	y=cosh(4) Da y=27.308
sinh (seno hiperbólico)	x, real doble precisión	y=sinh(1) Da y=1.175
tanh (tangente hiperbólica)	x, real doble precisión	y=tanh(0.8) Da y=0.664
exp (e elevado a x)	x, real doble precisión	y=exp(3) Da y=20.085

NOMBRE FUNCION	ARGUMENTO/S	EJEMPLO
log (logaritmo natural)	x, real doble precisión	y=log(1450) Da y=7.279
log10 (logaritmo decimal)	x, real doble precisión	y=log10(2000) Da y=3.301
ceil (redondea al entero mas pequeño que sea $\geq x$)	x, real doble precisión	y=ceil(3.2826) Da y=4
fabs (valor absoluto)	x, real doble precisión	y=fabs(-25) Da y=25
floor (redondea al entero mas grande que sea $\leq x$)	x, real doble precisión	y=floor(7.962) Da y=7

NOMBRE FUNCION	ARGUMENTO/S	EJEMPLO
pow (eleva x a la potencia y)	x,y: reales doble precisión	p=pow(x,y) Con x=4 e y=3 da p=64
sqrt (raíz cuadrada)	x, real doble precisión	y=sqrt(2) Da y=1.414

FUNCIONES PARA GENERAR NUMEROS ALEATORIOS

La función `rand()` siempre genera un entero positivo aleatorio en base a una “semilla”.

Su seteo se establece con la función: `srand(“semilla”);`

Si no se especifica la “semilla”, su valor por default es “0”.

En ese caso, cada vez que se inicie el programa, la secuencia de números aleatorios va a ser siempre la misma. Perdiendo su propiedad de imprevisibilidad

Para que los números generados sean totalmente imprevisibles, se debe setear la “semilla” o valor inicial del generador de números aleatorios, con algo que cambie permanentemente. Por ejemplo: el tiempo.

Esto se logra con la siguiente instrucción:

`srand(time(0));` ó `srand(time(NULL));`

Con éste seteo inicial, cada vez que usemos la función en el resto del programa, va a generar realmente un número aleatorio.

Estas dos funciones están en dos bibliotecas: `stdlib.h` y `time.h`, por lo tanto se debe escribir en la cabecera del programa:

`#include <stdlib.h>`

`#include <time.h>`

EJEMPLOS PARA GENERAR NUMEROS ALEATORIOS

**1- GENERAR UN NUMERO ALEATORIO ENTERO POSITIVO
EN EL INTERVALO (a,b):**

La formula a usar es: $n=a+\text{rand}() \% (b-a+1);$

**2- GENERAR UN NUMERO ALEATORIO ENTERO POSITIVO
DE UN SOLO DIGITO:**

La formula a usar es: $n=\text{rand}() \% 10;$

EJEMPLOS DE GENERAR NUMEROS ALEATORIOS

3- GENERAR UN NUMERO ALEATORIO ENTERO DE 3 DIGITOS

Una de las tantas formas es:

```
do {                                // Filtro
    n=rand() ;
}
while(n<100 || n>999);
```

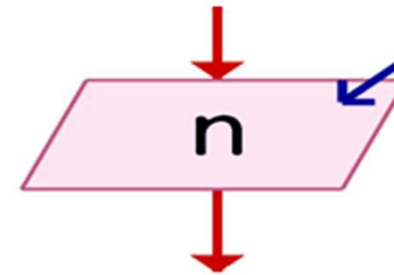
4- GENERAR UN NUMERO ALEATORIO ENTERO (POSITIVO O NEGATIVO)

La formula a usar es: $n = \text{rand}() * \text{POW}(-1, (\text{rand}() \% 10)) ;$

INSTRUCCIONES PARA ENTRADA DE DATOS Y SALIDA DE RESULTADOS

INGRESO DE DATOS

En los diagramas de flujo, se usó:



En el lenguaje C:

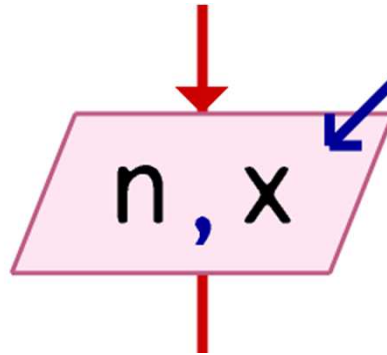
```
scanf("%d", &n);
```

Nombre de la variable donde se almacena el dato

Símbolo que indica al Compilador que debe guardar la dirección del dato para poder buscarlo más adelante

Símbolo que indica el tipo de dato:
%d: dato entero, %f: dato real

Si se desea ingresar mas de un dato de una sola vez:



En el lenguaje C:

`scanf("%d %f",&n,&x);`

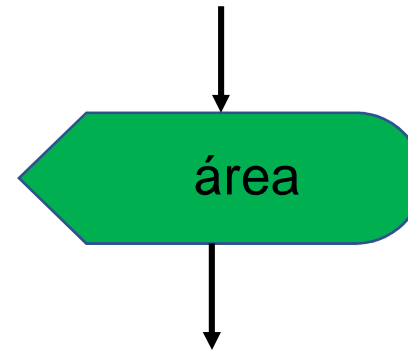
Nombres de las variables
que almacenan cada dato

& es el símbolo que indica al Compilador que
debe guardar la dirección del dato para poder
buscarlo más adelante

Cada símbolo corresponderá al tipo de dato,
respetando el orden

SALIDA DE RESULTADOS POR LA PANTALLA

En el diagrama de flujo se uso:



En el lenguaje C:

```
printf("La superficie del triangulo es %.2f", area);
```

Mensaje de texto al usuario

.2: significa que pido
mostrar 2 decimales en el
resultado

Símbolo correspondiente
al tipo de dato

Variable que contiene el
resultado a mostrar

Si se requiere mostrar mas de un resultado de una sola vez:

