

## Sintaxis Básica

### Recomendaciones:

### Comprende a fondo el problema antes de delinear la solución

1. Establece cuales son los datos, si los conoces o no, y si puedes averiguarlos de algún modo.
2. Analiza las condiciones que deban ser tenidas en cuenta.
3. Planea la solución.
4. Para orientarte, considera las siguientes preguntas:
  - ✓ ¿Es conocido el problema?
  - ✓ Si es desconocido: ¿conoce la solución de alguno similar?
  - ✓ ¿Puede resolver el problema en su totalidad?
  - ✓ Si no es así: ¿Puede resolver parte del problema?
  - ✓ Si no es así: ¿Puede resolverlo en distintas condiciones?
5. Es importante que desarrolles todo lo que puedas de la solución y que determines con claridad cuál es la dificultad que no puedes salvar. Explicitarla, en muchos casos, orienta hacia la solución de la misma.
6. Desarrolla el plan de la solución (algoritmo), chequeando cada paso.
7. Examina la solución en su conjunto. Recién entonces, escribe el código.

## Variables, operadores y tipo de datos. Especificadores de formato. Entrada y salida estándar.

1. Indicar cuál es la salida por pantalla para cada sentencia printf, siendo: `int a = 5;`

`float b = 13.546;`

`char c = 'A'; char`

`d = 'a';`

`int e = 5>1;`

`printf("a vale %d", a);`

`printf("a vale %o", a);`

`printf("a vale %x", a);`

`printf("a vale %f", a);`

`printf("a vale %c", a);`

`printf("b vale %d", b);`

`printf("b vale %5.2d", b);`

`printf("b vale %f", b);`

`printf("b vale %.1f", b);`

`printf("b vale %.2f", b);`

`printf("b vale %6.4f", b);`

`printf("b vale %6.1f", b);`

`printf("b vale %c", b);`

`printf("b vale %o", b);`

`printf("b vale %x", b);`

`printf("c vale %d", c);`

`printf("c vale %f", c);`

`printf("c vale %c", c);`

`printf("c vale %o", c);`

`printf("c vale %x", c);`

`printf("d vale %d", d);`

`printf("d vale %f", d);`

`printf("d vale %c", d);`

`printf("d vale %o", d);`

`printf("d vale %x", d);`

`printf("e vale %d", e);`

`printf("e vale %f", e);`

`printf("e vale %c", e);`

`printf("e vale %o", e);`

`printf("e vale %x", e);`

2. Indicar qué queda almacenado en la variable con la que se ingresan datos en cada caso, si se han definido las variables de la siguiente forma:

`int a; float b; char c;`

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

`scanf ("%f", &a);`

`scanf ("%c", &a);`

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

`scanf ("%f", &b);`

`scanf ("%c", &b);`

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

`scanf ("%f", &c);`

`scanf ("%f", &c);`

3. Indicar qué es lo que se almacena en cada una de las variables indicadas a continuación, si se realizan las operaciones que se indican, sabiendo que se han definido las variables usadas de la siguiente forma:

`int a = 3, b = 2, c = 1, d, e;`

`float m = 2.5, n = 5.4, r, s;`

```
d = m;
d = n;
e = 3.7;
d = a;
d = a + b;
d = a * b;
d = a / b;
d = b / 4;
d = a % b;
d = a % 2;
e = b / c;
```

```
e = c / b;
r = a + b;
r = a / b;
r = a / 2;
r = a / 2.0;
s = m + n;
s = 3 + 4 - 1;
s = 3.0 + 4.0 - 1;
a = m;
a = m / 2;
a = m / 2.0;
```

```
a = 3.0 + 4.0 - 1;
r = a + 1;
r = a + 1.0;
a++;
r++;
b--;
a+=5;
s*=5;
```

4. Indicar qué valor se almacena en cada variable de la izquierda de cada asignación, siendo:

```
int a, b, c, d = 0;
int e = 1; /*0 (cero) corresponde a falso*/
int x = 2, y = 3;
```

<pre>a = 5 &gt; 3; b = ( ( 4 &lt; 5 ) &amp;&amp; ( 2 &gt; 2 ) ); a = ! e; b = ( ( x % 2 == 0 )    ( x - y &lt; 10 ) ); c = ( d &amp;&amp; e );</pre>	<pre>c = ( d    e ); c = ! ( d &amp;&amp; e ); c = ( ! d ) &amp;&amp; ( ! e ); c = ( a &amp;&amp; ( ! a ) ); c = ( ( ( x &lt;= ( y * 3.2 ) ) &amp;&amp; ( y % 2 != 0 ) )    ( 1 ) );</pre>
--	--

5. Supongamos que tenemos estas variables:

```
int a = 10; float b = 19.3; double d = 64.8; char c = 64;
```

Indica el tipo resultante para las expresiones siguientes:

a) a+b	b) c+d	c) (int)d+a	d) d+b	e) (float)c+d
<input type="radio"/> char <input type="radio"/> int <input type="radio"/> float <input type="radio"/> double	<input type="radio"/> char <input type="radio"/> int <input type="radio"/> float <input type="radio"/> double	<input type="radio"/> char <input type="radio"/> int <input type="radio"/> float <input type="radio"/> double	<input type="radio"/> char <input type="radio"/> int <input type="radio"/> float <input type="radio"/> double	<input type="radio"/> char <input type="radio"/> int <input type="radio"/> float <input type="radio"/> double

## Operador sizeof()

6. Qué muestra el siguiente programa:

```
int main() {
    char c; short s; int i; long l; float f; double d; long double ld;

    printf("\nLARGOS DE TIPOS ESCALARES PREDEFINIDOS\n");
    printf("El largo de char es %d\n", sizeof(c));
    printf("El largo de short es %d\n", sizeof(s));
    printf("El largo de int es %d\n", sizeof(i));
    printf("El largo de long es %d\n", sizeof(l));
    printf("El largo de float es %d\n", sizeof(f));
    printf("El largo de double es %d\n", sizeof(d));
    printf("El largo de long double es %d\n", sizeof(ld));
    getchar();
    return 0; }
```

## Operadores de asignación compacta

7. Suponiendo los siguientes valores iniciales para las variables:

$x = 2;$

$z = 9;$

$s = 10;$

$y = 6;$

$r = 100;$

$a = 15; b = 3;$

¿Cuáles son los valores correctos en cada expresión?

a)  $x += 10;$

12      10      11

b)  $s *= b;$

9      13      30

c)  $r /= 0;$

infinito      1      error

d)  $y += x + 10;$

8      12      18

e)  $z -= a * b;$

-36      -18      36

8.Cuál es la expresión equivalente para las siguientes operaciones, usando operadores mixtos:

a)  $x = 10 + x - y;$

$x += 10 - y$

$x -= y + 10$

$x += 10 + y$

b)  $r = 100 * r;$

$r *= 100 * r$

$r *= 100$

$r += 100$

c)  $y = y / (10 + x);$

$y /= 10 * x$

$y /= 10 + y / x$

$y /= 10 + x$

d)  $z = 3 * x + 6;$

$z += 6$

$z *= 3$

no es posible

9. Siendo "y" una variable int y "x" una variable float, indica si son correctas las sentencias:

$x = .25 + y$

$y = 0.25 + x$

$y = 0,25 + x$

10. Evalúa cada una de las siguientes expresiones y luego comprueba el resultado en el laboratorio.

$8 * 6 / 3 * 4$

$(8 * 6) / 3 * 4$

$8 * 6 / (3 * 4)$

$(8 * 6 / 3 * 4)$

$(8 * 6) / (3 * 4)$

$1 + 4 * 5 + 8 / 4 + 4$

$1 + 4 * 5 - 8 / 4 + 4$

11. Supone que las variables a, b y c tienen asignados los valores 49, 5 y 3 respectivamente. Encuentra:

$a \% b * c + 1$

$a \% (b * c) + 1$

$24 / c * 4$

$(\text{int})(a / b) \% 2$

$48 / (c * 2) * 4$

12. Describe los tipos de errores que se pueden encontrar en un programa (Ver módulo I).

13. Desarrolla un algoritmo que permita leer por teclado 2 valores y emitir por pantalla: la suma de los dos, la resta, producto, división, promedio y el doble producto del primero menos la mitad del segundo. Análisis: Para dar solución a este ejercicio es necesario leer los

valores que para el caso concreto del ejemplo son dos, calcular las operaciones con dichos valores y por último escribir el resultado. Los cálculos se realizarán y los valores serán almacenados en las variables correspondientes, por último se emitirán en la pantalla.

14. Encuentra el error en cada uno de los siguientes programas e indica de qué tipo es.

a)	b)
<pre>#include &lt;stdio.h&gt;  int main(){     integer x;     real y;         scanf("%D", y);         printf("%f",x);     return 0; }</pre>	<pre>#include &lt;stdio.h&gt;  int main(){     int n,total;     float promedio;     n=0;     promedio=total/n;         printf("El promedio es: %f\n",promedio); return 0; }</pre>

15. Desarrolla un algoritmo que permita, dados ciertos centímetros como entrada de tipo flotante, emitir por pantalla su equivalencia en pies (enteros) y en pulgadas (flotante, con 1 decimal).
16. Construye un programa que pregunte los años que tienes y emita como respuesta el número de días (sólo en años).
17. Construye un programa que dados el costo de 1 kg de determinado producto y la cantidad comprada, emita el valor del total a pagar.
18. Construye un programa que permita ingresar los valores de 2 de los ángulos interiores de un triángulo, y se emita por pantalla el valor del restante.
19. Construye un programa que permita ingresar las medidas de los lados de un rectángulo; el mismo debe emitir por pantalla su superficie y perímetro.
20. Construye un programa que permita ingresar la *superficie* de un cuadrado (en m2), el mismo debe emitir por pantalla su perímetro.

### Uso de la librería math.h

21. Este programa utiliza la librería math.h y sus funciones incorporadas. Ejecuta, prueba y extrae conclusiones:

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

```
int main(){
    printf( "sqrt(%.1f) = %.1f\n", 900.0, sqrt( 900.0 ) );
    printf( "sqrt(%.1f) = %.1f\n", 9.0, sqrt( 9.0 ) );
    printf( "exp(%.1f) = %f\n", 1.0, exp( 1.0 ) );
    printf( "exp(%.1f) = %f\n", 2.0, exp( 2.0 ) );
    printf( "log(%.1f) = %.1f\n", 2.718282, log( 2.718282 ) );
    printf( "log(%.1f) = %.1f\n", 7.389056, log( 7.389056 ) );
    printf( "log10(%.1f) = %.1f\n", 1.0, log10( 1.0 ) );
    printf( "log10(%.1f) = %.1f\n", 10.0, log10( 10.0 ) );
    printf( "log10(%.1f) = %.1f\n", 100.0, log10( 100.0 ) );
    printf( "fabs(%.1f) = %.1f\n", 13.5, fabs( 13.5 ) );
    printf( "fabs(%.1f) = %.1f\n", 0.0, fabs( 0.0 ) );
    printf( "fabs(%.1f) = %.1f\n", -13.5, fabs( -13.5 ) );
    printf( "ceil(%.1f) = %.1f\n", 9.2, ceil( 9.2 ) );
```

```
printf( "ceil(%.1f) = %.1f\n", -9.8, ceil( -9.8 ) );
printf( "floor(%.1f) = %.1f\n", 9.2, floor( 9.2 ) );
printf( "floor(%.1f) = %.1f\n", -9.8, floor( -9.8 ) );
printf( "pow(%.1f, %.1f) = %.1f\n", 2.0, 7.0, pow( 2.0,7.0 ) );
printf( "pow(%.1f, %.1f) = %.1f\n", 9.0, 0.5, pow( 9.0,0.5 ) );
printf( "fmod(%.3f/%.3f) = %.3f\n", 13.675, 2.333, fmod(
13.675, 2.333 ) );
printf( "sin(%.1f) = %.1f\n", 0.0, sin( 0.0 ) );
printf( "cos(%.1f) = %.1f\n", 0.0, cos( 0.0 ) );
printf( "tan(%.1f) = %.1f\n", 0.0, tan( 0.0 ) ); system( "pause");
return 0; }
/* fin de main */
```

22. Transcribe el siguiente programa y extrae conclusiones:

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define CUAD(x) (x*x) /* Definición de macros */
```

```
int main(){
    float a;
    printf("\nEscriba un número: ");
    scanf("%f",&a);
    printf("\nSu cuadrado es: %.2f\n",CUAD(a));
    getchar();
    return 0;}
```

## Ejercicios integradores

23. Construye un programa que permita ingresar 2 tiempos, expresados en horas, minutos y segundos, el mismo debe emitir por pantalla la suma de ambos (también en horas, minutos y segundos).

24. Desarrolla un algoritmo que le permita leer un valor radio (R), calcular el área (A) de un círculo y emitir valor.

$$A = \pi * R^2$$

25. Determina la hipotenusa de un triángulo rectángulo conocidas las longitudes de sus dos catetos. Desarrolla los correspondientes algoritmos. *Análisis: En el ejercicio se puede definir como tareas las tres acciones solicitadas. "Leer, Calcular y Escribir", Lee cada uno de los valores de los dos catetos y almacenarlos en cada uno de los identificadores definidos para el caso, calcular la hipotenusa aplicando la fórmula correspondiente almacenando su valor en el identificador del caso y escribir el valor encontrado para la hipotenusa como respuesta.*

$$Hip = \sqrt{CatA^2 + CatB^2}$$

26. Ingresar una cantidad entera de segundos y convertirla en horas, minutos y segundos utilizando los operadores de cociente y resto enteros.

27. Desarrolla un algoritmo que permita leer un valor que represente una temperatura expresada en grados Celsius y convierta dicho valor en un valor expresado en grados Fahrenheit.

28. Desarrolla un algoritmo que permita calcular el área de un triángulo en función de las longitudes de sus lados previamente leídos desde el teclado.

$$AREA = \sqrt{p(p-A)(p-B)(p-C)}$$

29. Desarrolla un algoritmo que permita determinar el área y volumen de un cilindro cuyo radio (r) y altura (h) se leen desde teclado.

30. Desarrolla un algoritmo que permita calcular el área (a) de un segmento de círculo. *Análisis: Para calcular el área de un segmento de círculo lo primero que hay que hacer es leer el valor del radio y leer el valor de x que es la distancia del centro al segmento. Una vez leído dichos valores se calcula aplicando la fórmula respectiva y por último se emite el valor del área.*

$$A = (\pi * r^2) / 2 - [x \sqrt{r^2 - x^2} + r^2 \sin^{-1}(x / r)]$$