

## TPN°2: Expresiones en C - Entradas y Salidas

El código fuente de la biblioteca **getnum.c**, junto con su header, se encuentra en ITBA Online.

Para leer por entrada estándar valores de tipo entero usar la función **getint**.

Para leer valores numéricos reales, usar la función **getfloat** o **getdouble**.

Al invocar cualquiera de estas funciones, cuando el usuario ingrese un número seguido de caracteres extras, el número será aceptado y el resto del buffer será consumido.

Hay una variante de la biblioteca, el archivo **getnum2.c**, que contiene las mismas funciones que **getnum.c**, pero en caso de haber recibir caracteres extras después del número –excepto espacios- el mismo es rechazado como error.

### Ejercicio 1

Decidir si en los siguientes pares de códigos cada una de las variables termina con el mismo valor numérico independientemente de la plataforma. Indicar en las evaluaciones el tipo de aritmética utilizada. **Justificar todas las respuestas.**

- |    |   |  |
|----|---|--|
| a) | <pre>int letra, flag;<br/>flag = 0;<br/>if ( letra == (int)'ñ')<br/>    flag=1;</pre> | <pre>int letra, flag;<br/>flag = 0;<br/>if ( letra == 'ñ')<br/>    flag=1;</pre> |
| b) | <pre>float x;<br/>int y;<br/>y = 5;<br/>x = y / 3.0 ;</pre>                           | <pre>float x;<br/>int y;<br/>y = 5;<br/>x = (float)( y / 3);</pre>               |
| c) | <pre>float x;<br/>int y;<br/>y = 5;<br/>x = y / 3.0 ;</pre>                           | <pre>int x;<br/>int y;<br/>y = 5;<br/>x = (float)y / 3;</pre>                    |
| d) | <pre>int x;<br/>float y;<br/>y = 5.5;<br/>x = y / 3 ;</pre>                           | <pre>int x;<br/>float y;<br/>y = 5.5;<br/>x = (int)y / 3.0;</pre>                |
| e) | <pre>double x;<br/>int y;<br/>x = 25;<br/>y = x / 5;</pre>                            | <pre>double x;<br/>float y;<br/>x =25.0 ;<br/>y = x / 5;</pre>                   |

f)	<pre>int x,y; x = (char) 5; y = x++;</pre>	<pre>int x,y; x = 5.0; y = x++;</pre>
g)	<pre>int i,j; int k; i = j = 20000; k = i + j;</pre>	<pre>int i,j; unsigned k; i = j = 20000; k = i + j;</pre>
h)	<pre>char a,b; char c; a = b = 60 ; c = a + b;</pre>	<pre>char a,b; unsigned char c; a = b = 60; c = a + b;</pre>
i)	<pre>char a,b; char c; a = b = 100 ; c = a + b;</pre>	<pre>char a,b; unsigned char c; a = b = 100; c = a + b;</pre>
j)	<pre>int x,y; x = (char) 300; y = x++;</pre>	<pre>int x,y; x = 300.0; y = x++;</pre>
k)	<pre>int x,y,z; z = ( x = 2 ) + ( y = x )</pre>	<pre>int x,y,z; x = y = 2; z = x + y;</pre>

### Ejercicio 2

a) Escribir un programa que imprima el mensaje **"Este es un programa en C"** en la salida estándar, colocando cada palabra en una línea. (Usar un único printf)

b) Redireccionar la salida del programa hacia el archivo **output.txt**. Una vez terminada la ejecución del programa, editar dicho archivo y verificar que contenga la salida correcta.

### Ejercicio 3

¿Es válida la siguiente sentencia ?

```
printf ( "Hola que
        tal \n" );
```

#### Ejercicio 4

Probar las siguientes instrucciones e investigar cuales son equivalentes entre sí. Sacar conclusiones.

```
printf ("\a\n");
printf ("%c\n", 7);
printf ("%d\n", 7);
printf ("%f\n", 7);
printf ("%g\n", 7);
```

#### Ejercicio 5

Dada la siguiente declaración de variable :

```
char c = 't';
```

armar un programa que imprima en la salida estándar dicha variable, con los siguientes formatos: %c, %d, %f. Indicar qué valores imprime y de dónde obtiene dichos valores.

#### Ejercicio 6

Indicar qué salida se obtiene en cada *printf* del siguiente programa:

```
#include <stdio.h>
#define palabra "ultima prueba"

int
main(void)
{
    int  num1= 53, num2=4;
    float num3= 6.874;

    printf ("num1= %10d\n", num1);
    printf ("num2= %010d\n", num2);
    printf ("num1= %-10d\n", num1);
    printf ("num1+num2= %5i\n", num1 + num2);
    printf ("num1+num2= %5f\n", num1 + num2);
    printf ("num3= %-3.2f\n", num3);
    printf ("num1= %-4d\nnum2= %-4d\nnum3= %3.1f\n", num1,num2, num3);
    printf ("num3(como entero)= %d\n", num3);
    printf ("num1 / num2 = %d\n", num1 / num2);
    printf ("num2 / num1 = %d\n", num2 / num1);
    printf("esta es la %s\n", palabra);

    return 0;
}
```

**Ejercicio 7**

Dada la siguiente definición de variables

```
int    edad= 25;
float  longitud= 185.654;
char   letra= 'Z';
```

imprimir en la salida estándar:

- a) la variable **edad**, encolumnada a izquierda, en un ancho de campo de 5 espacios.
- b) la variable **edad**, encolumnada a derecha, en un ancho de campo de 10 espacios.
- c) la variable **longitud**, encolumnada a la izquierda en un ancho de campo de 10 espacios, con 2 dígitos decimales.
- d) la variable **letra**, como número entero, en un ancho de campo de 8 espacios.

**Ejercicio 8**

Escribir un programa que lea una variable tipo **float** representando una velocidad en m/s, e imprima en la salida estándar el equivalente en km/h. **Utilizar una única variable.**

**Ejercicio 9**

Leer un numero entero positivo y escribir en la salida estándar el valor de dicho número dividido 2, usando decalaje.

**Ejercicio 10**

Escribir un programa que lea una variable entera que represente un intervalo de tiempo expresado en segundos. Realizar la conversión para imprimir en la salida estándar el equivalente en horas, minutos y segundos. **No utilizar variables auxiliares.**

**Ejercicio 11**

Leer dos caracteres desde la entrada estándar e imprimir en la salida estándar el de mayor valor ASCII, usando el operador condicional. Redireccionar la entrada de manera tal que **getchar** lea desde el archivo **entrada.txt**. Dicho archivo será escrito con un editor de texto sin formato y deberá tener el siguiente contenido:

- a) **ABCD**
- b) **A**  
**B**  
**C**  
**D**
- c) **A    B** (usando tabulador entre A y B)

Indicar la salida generada en cada punto.

**Ejercicio 12**

Leer un caracter desde la entrada estándar e imprimir en la salida estándar si el mismo es una letra o no. Considerar que el conjunto de caracteres es el del lenguaje inglés, o sea ignorar la ñ y vocales acentuadas. Usar el operador condicional.

**Ejercicio 13**

Escribir un programa que lea dos enteros, e imprima si el segundo es múltiplo del primero. Usar el operador condicional.

**Ejercicio 14**

Escribir un programa que lea dos enteros y a continuación imprima el promedio, la suma, el menor y el mayor de ellos (o indicar que son iguales). Usar el operador condicional.

*Ejemplo:*

Si tuviésemos `int a, b;`  
y se leyeran los valores **5** y **8**, se debería imprimir:

**El promedio es 6.5**  
**La suma es 13**  
**El menor es 5**  
**El mayor es 8**  
**no son iguales**

**Ejercicio 15**

Leer un caracter e imprimirlo en mayúscula. Considerar que el conjunto de caracteres es el del lenguaje inglés, o sea ignorar la ñ y vocales acentuadas. (Usar operador condicional).

**Ejercicio 16**

Leer dos caracteres de la entrada estándar e imprimir en la salida estándar si son iguales, si el primero es mayor o si el primero es menor, utilizando operadores condicionales:

- a) desde teclado, hacia pantalla
- b) redireccionando la entrada desde archivo, hacia pantalla
- c) desde teclado, redireccionando la salida a un archivo
- d) redireccionando la entrada y la salida con archivos

*Ejemplo:*

Si se ingresaran los caracteres **'a'** y **'g'**  
se debería imprimir **El caracter 'a' es menor al caracter 'g'**

**Ejercicio 17**

Indicar, para aquellas expresiones que compilen correctamente, el valor que devuelve.

```
int
main(void)
{
    "3" + "4";
    '3' + '4';
    3 + 4;
    '3' + 4;
    3 + '4';
    "3" + '4';
    "3" + 4;
    return 0;
}
```

### **Ejercicio 18**

Indicar si las siguientes afirmaciones son verdaderas o falsas, justificar

- a) La función `printf` siempre empieza a imprimir al comienzo de una línea nueva.
- b) El operador de módulo (%) puede ser usado sólo con operandos enteros
- c) Las declaraciones de variable pueden aparecer en cualquier parte dentro de la función **main**.
- d) Un programa C que imprime tres líneas de texto debe contener tres sentencias con **printf**

### **Ejercicio 19**

Realizar los siguientes ejercicios del libro de texto básico:

**1.1, 1.2, 1.6 y 1.7.**