



Universidad Tecnológica Nacional
Facultad Regional Resistencia
Técnico Universitario en Programación

Lenguaje De Programación C

***Entrada
y
Salida
de
Datos***



Funciones de Entrada y Salida

- **getchar**
 - **putchar**
 - **scanf**
 - **printf**
 - **gets**
 - **puts.**
- Estas seis funciones permiten la transferencia de información entre la computadora y los dispositivos de entrada/ salida estándar (por ejemplo, un teclado y un monitor).
 - **getchar y putchar**, permiten la transferencia de caracteres individuales hacia dentro y hacia fuera de la computadora;
 - **gets y puts** permiten la entrada y salida de cadenas de caracteres.
 - **scanf y printf** son más complicadas, pero permiten la transferencia de caracteres individuales, valores numéricos y cadenas de caracteres;

- **getchar y putchar**, permiten la transferencia de caracteres individuales hacia dentro y hacia fuera de la computadora;

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char c;  
    c=getchar();  
    putchar (c);
```

```
    getch();  
    return 0;  
}
```

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char c;  
    printf("Ingrese un caracter por teclado");  
    printf("\n");  
    c=getchar();  
    printf("El caracter ingresado es:");  
    putchar (c);  
    return 0;  
}
```

- **gets y puts** permiten la entrada y salida de cadenas de caracteres.

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char c[10];  
    printf("Ingrese su nombre por teclado");  
    printf("\n");  
    gets(c);  
    printf("El nombre ingresado es:");  
    puts(c);  
  
    return 0;  
}
```



Función printf()

Es la función de salida de datos con formato .

La sintaxis es :

printf (texto,cadena de control, lista de argumentos);

donde texto y cadena de control de tipo son opcionales, dependiendo de lo que se desee mostrar. Cadena de control es una cadena de caracteres “%tipo” que indica el tipo de dato a desplegar (lo requiere la función *printf()*). Por otro lado, argumento o argumentos es el valor o los valores que se pretende mostrar, y pueden ser variables, constantes, expresiones aritméticas, resultados de funciones o simplemente texto que el programa debe mostrar al usuario. A continuación se muestra un ejemplo del uso y la explicación.

```
int a;  
a= 2;  
...  
printf (“el numero es :%d”, a);
```



Función printf() – Cadena de Control

<i>Cadena de tipo</i>	<i>Descripción</i>
<code>%d</code>	<i>El dato es un entero decimal (int).</i>
<code>%i</code>	<i>El dato es un entero.</i>
<code>%o</code>	<i>El dato es un entero octal.</i>
<code>%X</code>	<i>El dato es un entero hexadecimal.</i>
<code>%u</code>	<i>El dato es un entero sin signo en decimal (unsigned int).</i>
<code>%c</code>	<i>El dato es un carácter (char).</i>
<code>%e</code>	<i>El dato es un real expresado en base y exponente (float).</i>
<code>%f</code>	<i>El dato es un real escrito con punto decimal con signo (float).</i>
<code>%g</code>	<i>El dato es un real (float).</i>
<code>%s</code>	<i>El dato es una cadena de caracteres que finaliza con el carácter nulo \0.</i>
<code>%lf</code>	<i>El dato es real de tipo long double.</i>



Función printf()-Ejemplos

```
int a=7;  
float b=8.2;  
char c='s';
```

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

Se visualiza un 7, que es el contenido de la variable a.

```
printf("%d", a+b);
```

Se visualiza un 15, ya que es la suma de a + b mostrada como valor entero.

```
printf("%f", a+b);
```

Se visualiza un 15.2, ya que es la suma de a + b mostrada como valor real.

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

Se visualiza la letra 's' que es el contenido de la variable c.

```
printf("%c %d %f", c, a,b);
```

Se visualiza s 7 8.2 que son los valores de las variables c, a y b respectivamente.

```
printf("HOLA");
```

Se visualiza la palabra HOLA.

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char car;  
    printf("Ingrese un caracter por teclado");  
    printf("\n");  
    car=getchar();  
    printf("El caracter ingresado es:");  
    putchar (car);  
    printf("\n");  
    printf("El caracter ingresado es %c:",car);  
    return 0;  
}
```

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char car[10];  
    printf("Ingrese su nombre por teclado");  
    printf("\n");  
    gets(car);  
    printf("El nombre ingresado es:");  
    puts(car);  
    printf("\n");  
    printf("El nombre ingresado es %s:",car);  
    return 0;  
}
```



Secuencia de Escape

Las secuencias de escape son cadenas de caracteres que tienen distintos significados dependiendo de la cadena que se utilice.

La forma más sencilla de escribir una secuencia de escape es con el *carácter barra invertida* (\), seguido de un carácter especial. Por tanto, cuando en la cadena de control de la función `printf` se escriba una secuencia de escape, o bien se mostrará un carácter gráfico por pantalla, o bien se realizará una acción.



Universidad Tecnológica Nacional
Facultad Regional Resistencia
Técnico Universitario en Programación



Secuencia de Escape

<i>Secuencia de escape</i>	<i>Descripción</i>
<code>\a</code>	<i>Alarma</i>
<code>\b</code>	<i>Retroceso</i>
<code>\f</code>	<i>Avance de página</i>
<code>\n</code>	<i>Retorno de carro y avance de línea</i>
<code>\r</code>	<i>Retorno de carro</i>
<code>\t</code>	<i>Tabulación</i>
<code>\v</code>	<i>Tabulación vertical</i>
<code>\\</code>	<i>Diagonal invertida</i>
<code>\?</code>	<i>Signo de interrogación</i>
<code>\"</code>	<i>Comillas dobles</i>
<code>\ooo</code>	<i>Octal</i>
<code>\xhh</code>	<i>Hexadecimal</i>
<code>\0</code>	<i>Carácter nulo</i>



- Permite escribir datos en el dispositivo de salida estándar utilizando la función de biblioteca `printf`.
- Permite mostrar una cadena con formato y muestra la misma por la pantalla.
- La función `printf` del ejemplo muestra el mensaje “Mi primer programa” aparece el símbolo `'\n'`; este hace que después de mostrar el mensaje se pase a la línea siguiente, corresponde a un carácter Ascii no imprimible.
- La cadena con formato provee una descripción de la salida con el uso de un atributo marcador de posición específica que describe el valor esperado de un campo de entrada usando caracteres de escape “%” para especificar la posición relativa y el tipo de salida que la función debe producir.

```
printf("Color %s, numero1 %d, numero2 %05d, hex %x, real %5.2f.\n", "rojo", 12345, 89, 255, 3.14);
```

imprimirá la siguiente línea (incluyendo el caracter de nueva línea `\n`):

```
Color rojo, numero1 12345, numero2 00089, hex ff, real 3.14.
```



Secuencia de Escape- Ejemplo

```
int a=7;  
float b=8.2;  
char c='s';
```

```
printf("%d \n \t %f \n \t\t %c", a,b,c);
```

7

8.2

s

Formateadores: Permite dar formato específico a la salida.



Formateador	Salida
%d ó %i	entero en base 10 con signo (int)
%u	entero en base 10 sin signo (int)
%o	entero en base 8 sin signo (int)
%x	entero en base 16, letras en minúscula (int)
%X	entero en base 16, letras en mayúscula (int)
%f	Coma flotante decimal de precisión simple (float)
%lf	Coma flotante decimal de precisión doble (double)
%e	La notación científica (mantisa / exponente), minúsculas (decimal precisión simple ó doble)
%E	La notación científica (mantisa / exponente), mayúsculas (decimal precisión simple ó doble)
%c	caracter (char)
%s	cadena de caracteres (string)

Ejemplo del uso de Formateadores:

%.7i	largo mínimo de 7 dígitos, justificado a la derecha, rellena con ceros
%8.2f	tamaño total de 8 dígitos, con dos decimales



Ejemplo de Funciones de Entrada y Salida

```
#include <stdio.h> int main ()  
{  
    char c; float x, y; int i;  
  
    printf("Ingrese un caracter:");  
    c = getchar(); /* entrada de un carácter */  
    printf("Ingrese un valor flotante:");  
    scanf("%f\n",&x);/* entrada de número en coma flotante */  
    printf("Ingrese un valor entero:");  
    scanf("%d\n",&i);/* entrada de enteros */  
    printf("Mostrar los valores ingresados:");  
    putchar(c);/* salida de un carácter */  
    printf("%3d %7.4f", i, x); /* salida de números*/  
}
```



Función scanf()

Es la función de entrada de datos con formato .

La sintaxis es :

scanf (cadena de control, &variable);

El operador & es necesario en scanf() para simular las *llamadas por referencia*, y hace que la función trabaje internamente con la dirección de la variable.

```
int edad;  
  
printf("Teclea tu edad");  
scanf("%d",&edad);
```



Ejemplo de Funciones de Entrada y Salida

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

```
int main ()
```

```
{
```

```
    float x; int k;
```

```
    printf("Ingrese un valor flotante:");
```

```
    scanf("%f",&x);/* entrada de número en coma flotante */
```

```
    printf("Ingrese un valor entero:");
```

```
    scanf("%d\n",&k);/* entrada de enteros */
```

```
    printf("Mostrar los valores ingresados:");
```

```
    printf("Valor flotante: %f" y valor entero:%d", x,k);
```

```
}
```

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char p;  
    printf("Ingresar un caracter por teclado");  
    p=getchar();  
    printf ("El valor ingresado por teclado es:"),  
    putchar (p);  
    printf("Ingresar un caracter por teclado\n");  
    scanf (" %c",&p);  
    printf ("El valor ingresado por teclado es:"),  
    putchar (p);  
    return 0;  
}
```

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */
```

```
int main()  
{  
    char p;  
    printf("Ingresar un caracter por teclado");  
    p=getchar();  
    printf ("El valor ingresado por teclado es:"),  
    putchar (p);  
    printf("Ingresar un caracter por teclado\n");  
    scanf (" %c",&p);  
    printf ("El valor ingresado por teclado es %c",p);  
  
    return 0;  
}
```

EJERCICIOS

Ejercicio: Encontrar los errores y corregir

```
#include <stdio.h> /* necesario para utilizar printf()*/
```

```
int main()  
{  
    char p;  
    p=gets();  
    printf ("El valor ingresado por teclado es:");  
    putchar (p);  
  
    return 0;  
}
```

CORRECCIÓN

```
#include <stdio.h> /* necesario para utilizar printf() */  
#include <conio.h> /* necesario para utilizar getch() */  
  
int main()  
{  
    char p;  
    p=getchar();  
    printf ("El valor ingresado por teclado es:")  
    putchar (p);  
  
    return 0;  
}
```


Ejercicio:

Completar los siguientes trozos de programa e indicar si se obtiene el mismo resultado.

Expresion 1	Expresion 1
<pre>int a,b,c; a= 3; b=4; c=a-b; printf("El resultado de la resta es %d",c);</pre>	<pre>int a,b,c; a= 3; b=4; printf("El resultado de la resta es %d",a-b);</pre>

Ejercicio: Se pide:

- Corregir los errores del siguiente programa y
- documentarlo usando comentarios
- cada mensaje que se muestra por pantalla debe aparecer uno debajo del otro.

#include stdio.h

```
int main()
{
int a,b,c;
printf("Ingrese los valores para realizar la operación resta);
printf ("Ingrese el primer valor");
scanf("%f",&a);
printf ("Ingrese el segundo valor");
scanf("%f",&a);
c=a-b;
printf("El resultado de la resta es %d",c);
return 0;
}
```