



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

TP 0

LEGAJO: 177500-5

NOMBRE: LUCIANO NAHUEL

APELLIDO: NUÑEZ PEREZ

CORREO INSTITUCIONAL: lnnezperez@frba.utn.edu.ar

USUARIO GIT: LUCHITO12

LINK REPOSITORIO: https://github.com/luchito12/2024_K2102_1775005

2) Investigar el uso de las funciones printf y scanf en C. Busque su especificación en el libro “El lenguaje de Programación C” de Brian Kernighan y Dennis Ritchie. Escriba la especificación.

La función de salida printf traduce valores internos a caracteres

```
int printf(char *format, argv arg2> ...)
```

Printf convierte, da formato e imprime sus argumentos en la salida estándar bajo el control de format. Regresa el número de caracteres impresos. La cadena de formato contiene dos tipos de objetos: caracteres ordinarios, Que son copiados al flujo de salida, y especificaciones de conversión, cada uno de los cuales causa la conversión e impresión de los siguientes argumentos sucesivos de printf.

Un especificador de formato sigue este prototipo:

```
%[ banderas ] [ ancho ] [ .precisión ] [ longitud ] especificador
```

A continuación, una tabla con el especificador, la salida y un ejemplo

especificador	Salida	Ejemplo
d o y	Número entero decimal firmado	392
u	Número entero decimal no firmado	7235
o	Octal sin firmar	610
x	Número entero hexadecimal sin firmar	7fa
X	Número entero hexadecimal no firmado, mayúscula	7FA
f	Punto flotante decimal, minúscula	392.65
F	Punto flotante decimal, mayúscula	392.65
e	Notación científica (mantisa/exponente), minúsculas	3.9265e+2
E	Notación científica (mantisa/exponente), mayúsculas	3.9265E+2
g	Utilice la representación más corta: %e o %f	392.65
G	Utilice la representación más corta: %E o %F	392.65
a	Punto flotante hexadecimal, minúscula	0xc.90fep-2
A	Punto flotante hexadecimal, mayúscula	-0XC.90FEP-2
c	Carácter	a
s	Cadena de caracteres	sample
p	Dirección del puntero	b8000000
n	Nada impreso. El argumento correspondiente debe ser un puntero a un int firmado. El número de caracteres escritos hasta ahora se almacena en la ubicación del puntero.	
%	A % seguido de otro % el personaje escribirá un solo % a la corriente.	%

(tabla sacada de cplusplus.com)

La función `scanf` es la entrada análoga de `printf`, y proporciona muchas de las mismas facilidades de conversión en la dirección opuesta.

```
int scanf(char *format, ...)
```

`scanf` lee caracteres de la entrada estándar, los interpreta de acuerdo con las especificaciones que están en `format`, y almacena los resultados a través de los argumentos restantes.

Un especificador de formato para `scanf` sigue este prototipo:

```
% [ * ] [ ancho ] [ longitud ] especificador
```

A continuación, una tabla con los especificadores, una breve descripción y lo que admite

especificador	Descripción	Caracteres extraídos
i	Entero	Cualquier número de dígitos, opcionalmente precedido por un signo (+ o -). <u>Dígitos decimales</u> asumido por defecto (0-9), pero a 0 el prefijo introduce dígitos octales (0-7), y 0x <u>dígitos hexadecimales</u> (0-f). argumento <i>Firmado</i> .
d o u	Número entero decimal	Cualquier número de <u>dígitos decimales</u> (0-9), opcionalmente precedido por un signo (+ o -). d es para un argumento <i>firmado</i> , y u para un <i>sin firmar</i> .
o	Octal entero	Cualquier número de dígitos octales (0-7), opcionalmente precedido por un signo (+ o -). Argumento <i>Desfirmado</i> .
x	Número entero hexadecimal	Cualquier número de <u>dígitos hexadecimales</u> (0-9, a-f, A-F), opcionalmente precedido por 0x o 0X, y todo opcionalmente precedido por un signo (+ o -). Argumento <i>Desfirmado</i> .
f, e, g	Número de punto flotante	Una serie de dígitos decimales, que opcionalmente contienen un punto decimal, opcionalmente precedido por un signo (+ o -) y opcionalmente seguido por el e o E carácter y un entero decimal (o algunas de las otras secuencias admitidas por <u>strtod</u>). Las implementaciones que cumplen con C99 también admiten formato hexadecimal de punto flotante cuando van precedidas de 0x o 0X.
a		
c	Caracter	El siguiente carácter. Si un <i>ancho</i> que no sea 1 se especifica, la función lee exactamente <i>ancho</i> los caracteres y los almacena en las ubicaciones sucesivas de la matriz aprobada como argumento. No se agrega ningún carácter nulo al final.

s	Cadena de caracteres	Cualquier número de caracteres que no sean de espacio en blanco, deteniéndose en el primer espacio en blanco de un carácter encontrado. Se añade automáticamente un carácter nulo de terminación al final de la secuencia almacenada.
p	Dirección del puntero	Una secuencia de caracteres que representa un puntero. El formato particular utilizado depende de la implementación del sistema y la biblioteca, pero es el mismo que el utilizado para formatear %p en fprintf .
[<i>caracteres</i>]	scanset	Cualquier número de caracteres especificado entre los corchetes. Un guión (-) que no es el primer carácter puede producir un comportamiento no portátil en algunas implementaciones de biblioteca.
[^ <i>caracteres</i>]	Negated scanset	Cualquier número de caracteres ninguno de ellos especificado como <i>caracteres</i> entre los soportes.
n	Count	No se consume ninguna entrada. El número de caracteres leídos hasta ahora stdin se almacena en la ubicación puntiaguda.
%	%	A % seguido de otro % coincide con un sencillo %.

(tabla sacada de cplusplus.com)

3) Realice un primer programa en lenguaje C que muestre por pantalla la leyenda “Hola mundo!”.

```
#include <stdio.h>

int main() {
    printf("Hola mundo!\n");
    getchar();
    return 0;
}
```

4) Realice un programa en lenguaje C que solicite el nombre al usuario y luego muestre por pantalla la leyenda "Hola ABC!", donde ABC es el nombre ingresado por teclado.

```
#include <stdio.h>

int main() {
    char nombre[3];

    printf("Por favor, ingresa tu nombre: ");
    scanf("%3s", nombre);
    printf("Hola %s!\n", nombre);
    getchar();
    return 0;
}
```

(En el código adjunto está en él .c con el código y comentarios en el mismo)

5) ¿Debió declarar alguna variable en el punto 4? De ser así, ¿de qué tipo de dato esa variable?

Si en el punto 4 se debió declarar una variable para almacenar el nombre ingresado por el usuario. La variable declarada fue de tipo char, implemente un arreglo de caracteres, para poder guardar una cadena de texto, que en este caso es el nombre del usuario, el cual por enunciado debería de ser "ABC".

6) Indique la diferencia, si existiera, con lo utilizado en Algoritmos y Estructuras de Datos

Las diferencias entre lo que vimos en este TP y en algoritmos son el `printf` con el `cout` y el `scanf` con el `cin`, la diferencia radica en que `printf` y `scanf` son funciones del lenguaje C y `cout`, `cin` son funciones que vienen incluidas en la biblioteca `ostream` y son específicas de C++. Otra diferencia es que acá usamos `\n` y en algoritmos `endl`.