



Entrada e Saída formatada em C:

 *printf(...)*

e

 *scanf(...)*

Ao contrário doutras linguagens, as facilidades de entrada/saída não fazem parte de Linguagem C.

Estas funções encontram-se especificadas numa biblioteca standard que existe em todos os compiladores de C - “stdio.h”

Nesta biblioteca também estão especificadas certas macros, constantes simbólicas (exemplo EOF).



Entrada / Saída Formatada

Saída Formatada

int printf(string-control [, arg1, arg2, ...])

printf converte, formata e imprime os seus argumentos na saída standard (stdout) de acordo com a string-control.

A string-control pode ter:

- ☞ caracteres “ordinários” -> copiados para a saída
- ☞ especificações de conversão -> provoca a conversão e impressão do argumento correspondente

Especificação de conversão:

% [-] [largura mínima do campo] [.precisão] [modificador de comprimento] caracter de conversão

Caracteres de conversão:

TABLE 7-1. BASIC PRINTF CONVERSIONS

CHARACTER	ARGUMENT TYPE; PRINTED AS
d, i	int; decimal number.
o	int; unsigned octal number (without a leading zero).
x, X	int; unsigned hexadecimal number (without a leading 0x or 0X), using abcdef or ABCDEF for 10, ..., 15.
u	int; unsigned decimal number.
c	int; single character.
s	char *; print characters from the string until a ‘\0’ or the number of characters given by the precision.
f	double; [-]m.ddddd, where the number of d’s is given by the precision (default 6).
e, E	double; [-]m.dddddE±xx or [-]m.dddddE±xx, where the number of d’s is given by the precision (default 6).
g, G	double; use %e or %E if the exponent is less than -4 or greater than or equal to the precision; otherwise use %f. Trailing zeros and a trailing decimal point are not printed.
p	void *; pointer (implementation-dependent representation).
%	no argument is converted; print a %.



Entrada Formatada

int scanf(string-control [, arg1, arg2, ...])

scanf lê caracteres da entrada standard (stdin), interpreta-os de acordo com o formato especificado na string-control e guarda os resultados nos correspondentes argumentos (que devem ser ponteiros).

A string-control pode ter:

- ☞ **espaços, tabs, mudanças de linha** -> espaços em branco, que são ignorados
- ☞ **caracteres “ordinários”** -> coincidem com o esperado à entrada
- ☞ **especificações de conversão** -> dirige a conversão do correspondente argumento de entrada

Especificação de conversão:

% [*] [largura máxima do campo] [especificação de tamanho] caracter de conversão

Caracteres de conversão:

TABLE 7-2. BASIC SCANF CONVERSIONS

CHARACTER	INPUT DATA; ARGUMENT TYPE
d	decimal integer; int *.
i	integer; int *. The integer may be in octal (leading 0) or hexadecimal (leading 0x or 0X).
o	octal integer (with or without leading zero); int *.
u	unsigned decimal integer; unsigned int *.
x	hexadecimal integer (with or without leading 0x or 0X); int *.
c	characters; char *. The next input characters (default 1) are placed at the indicated spot. The normal skip over white space is suppressed; to read the next non-white space character, use %1s.
s	character string (not quoted); char *, pointing to an array of characters large enough for the string and a terminating '\0' that will be added.
e, f, g	floating-point number with optional sign, optional decimal point and optional exponent; float *.
%	literal %; no assignment is made.