

# printf(format, arg1, arg2, ...)

`format` – управляющая строка в двойных кавычках, содержащая:

- обычные или управляющие символы: `text \n \' \' %% ...`
- спецификаторы преобразования: `%...specifier`

specifier	ожидаемый тип	примечание, пример
d,i	signed int	десятичные: -10
u	unsigned int	десятичные: 10
o	unsigned int	восмеричные: 012
x,X	unsigned int	шеснадиатиричные: d1ff (or D1FF)
f	double или float	0.123456
e,E	double или float	1.234567e-07 (or 2.32E12)
g,G	≡ %e или %f, какой короче	
c	char	один символ: c
s	C-string	текст: text
p	void*	указатель: 0x220a1b0

## Модификаторы формата

На месте точек в `% ...specifier` может стоять:

- `+` (знак плюс) – печать со знаком
- `-` (знак минус) – выравниванию по левому краю
- `w.p` – **ширина вывода** (`w`) и **точность** (`p`);
- модификаторы длины переменной:

	sizeof	specifiers	example
<code>h</code>	short int	<code>d i o u x</code>	<code>%hd</code>
<code>l(ell)</code>	long int	<code>d i o u x</code>	<code>%li</code>
<code>ll(ell-ell)</code>	long long int	<code>d i o u x</code>	<code>%llu</code>
<code>L</code>	long double	<code>e f g</code>	<code>%Lf</code>

## Управляющие символы

<code>\"</code>	double quote	<code>\\</code>	backslash
<code>\a</code>	alert (BEL)	<code>\b</code>	backspace
<code>\r</code>	carriage return	<code>\n</code>	new line
<code>\t</code>	horizontal tab	<code>%%</code>	a single %

# scanf(format, arg1, arg2, ...)

## Обратите внимание

- в функцию `scanf()` необходимо передавать адреса переменных
- обычные символы в `format` проверяются на совпадение с входным потоком и в случае несовпадения чтение прекращается
- значения спецификаторов не полностью совпадают с `printf`:
  - ✓ `%i` – читает целое число в десятичном, восьмеричном и шестнадцатеричном формате
  - ✓ `%f %e %g` – соответствуют данным типа `float`
  - ✓ `%lf %le %lg` – для `double`
  - ✓ `%Lf %Le %Lg` – для `long double`