

Информационные технологии

Лекция №4




Тема: «Функции вывода»

План лекции


- *Функции*
 - *_putch*
 - *puts*
 - *printf*

Кодовая таблица 437

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	<u>NUL</u> 0000	<u>STX</u> 0001	<u>SOT</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
10	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
20	<u>SP</u> 0020	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	:	;	<	=	>	?
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	<u>DEL</u> 007F
80	Ç 00C7	ü 00FC	é 00E9	ã 00E2	ä 00E4	à 00E0	å 00E5	ç 00E7	ë 00EA	ë 00EB	è 00E8	ï 00EF	î 00EE	ì 00EC	Ä 00C4	Å 00C5
90	É 00C9	æ 00E6	Æ 00C6	ö 00F4	ö 00F6	ò 00F2	û 00FB	ù 00F9	ÿ 00FF	Ö 00D6	Ü 00DC	¢ 00A2	£ 00A3	¥ 00A5	€ 20A7	f 0192
A0	á 00E1	í 00ED	ó 00F3	ú 00FA	ñ 00F1	Ñ 00D1	ª 00AA	º 00BA	¿ 00BF	¬ 2310	¬ 00AC	¼ 00BD	½ 00BC	¡ 00A1	« 00AB	» 00BB
B0	▒ 2591	▒ 2592	▒ 2593	 2502	 2524	 2561	 2562	 2556	 2555	 2563	 2551	 2557	 255D	 255C	 255B	 2510
C0	L 2514	L 2534	T 252C	 251C	- 2500	† 253C	† 255E	† 255F	L 255A	† 2554	 2569	† 2566	 2560	= 2550	† 256C	⊥ 2567
D0	 2568	† 2564	† 2565	L 2559	L 2558	F 2552	† 2553	† 256B	† 256A	J 2518	† 250C	■ 2588	■ 2584	■ 258C	■ 2590	■ 2580
E0	α 03B1	β 00DF	Γ 0393	π 03C0	Σ 03A3	σ 03C3	μ 00B5	τ 03C4	Φ 03A6	Θ 0398	Ω 03A9	δ 03B4	∞ 221E	φ 03C6	ε 03B5	Π 2229
F0	≡ 2261	± 00B1	≥ 2265	≤ 2264	 2320	 2321	÷ 00F7	≈ 2248	° 00B0	· 2219	· 00B7	√ 221A	² 207F	² 00B2	■ 25A0	<u>NBSP</u> 00A0

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
90	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
A0	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
B0	 2591	 2592	 2593	2502	2524	2561	2562	2556	2555	2563	2551	2557	255D	255C	255B	2510
C0	┌ 2514	└ 2534	┐ 252C	┑ 251C	─ 2500	┼ 253C	┆ 255E	┇ 255F	┈ 255A	┉ 2554	┊ 2569	┋ 2566	┌ 2560	┐ 2550	┑ 256C	┒ 2567
D0	┓ 2568	└ 2564	┐ 2565	┑ 2559	┒ 2558	┓ 2552	└ 2553	┑ 256B	┒ 256A	┓ 2518	└ 250C	┑ 2588	┒ 2584	┓ 258C	└ 2590	┑ 2580
E0	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F
F0	Ё 0401	ё 0451	Є 0404	е 0454	Ï 0407	ï 0457	ÿ 040E	ÿ 045E	° 00B0	· 2219	· 00B7	√ 221A	№ 2116	* 00A4	■ 25A0	<u>NBSP</u> 00A0

Кодовая страница OEM 866 (командная строка MS-DOS)

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
80	Ъ 0402	Ѓ 0403	Ѕ 201A	Ї 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	< 2039	Њ 040A	Ѓ 040C	Ѕ 040B	Ї 040F
90	Ђ 0452	Ѕ 2018	Ї 2019	„ 201C	“ 201D	• 2022	– 2013	— 2014	 2122	™ 2122	Љ 0459	> 203A	Њ 045A	Ѓ 045C	Ѕ 045B	Ї 045F
A0	<u>NBSP</u> 00A0	Ÿ 040E	ÿ 045E	Ј 0408	* 00A4	Г 0490	І 00A6	Ѕ 00A7	Ё 0401	@ 00A9	Є 0404	« 00AB	¬ 00AC	– 00AD	@ 00AE	İ 0407
B0	° 00B0	± 00B1	І 0406	і 0456	Г 0491	μ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	е 0454	» 00BB	ј 0458	Ѕ 0405	Ѕ 0455	İ 0457
C0	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
D0	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
E0	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
F0	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F

Кодовая страница Windows 1251

Управляющие последовательности

Escape-последовательность	Представляет
<code>\a</code>	Звонок (предупреждение)
<code>\b</code>	Удаление предыдущего символа
<code>\f</code>	Перевод страницы
<code>\n</code>	Новая строка
<code>\r</code>	Возврат каретки
<code>\t</code>	Горизонтальная табуляция
<code>\v</code>	Вертикальная табуляция
<code>\'</code>	Одиночная кавычка
<code>\"</code>	Двойная кавычка
<code>\\</code>	Обратная косая черта
<code>\?</code>	Литерал вопросительного знака
<code>\ooo</code>	Символ ASCII в восьмеричной нотации
<code>\xhh</code>	Символ ASCII в шестнадцатеричной нотации

Функции вывода

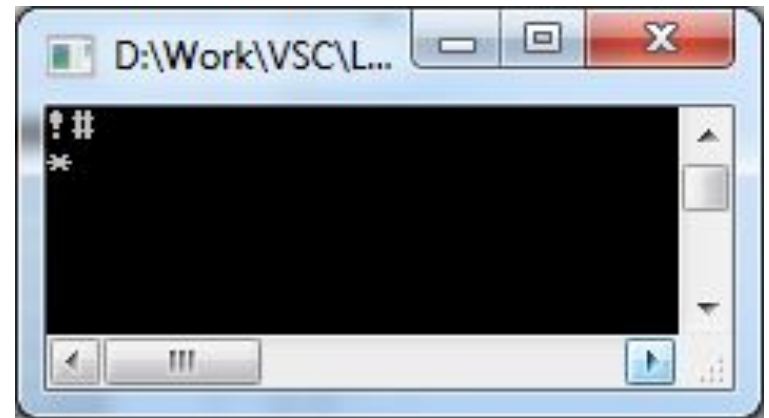
Функция вывода символа `_putch`

```
#include <conio.h> // В файле conio.h находится её объявление  
int _putch(int c); // Прототип функции
```

При ошибке возвращает EOF (-1). При успешном выполнении возвращает неотрицательное значение.

Пример использования:

```
// Демонстрация работы функции _putch  
#include <conio.h>  
void main()  
{  
    char ch = 0x21;  
    _putch(ch); // Выводит символ !  
    _putch(35); // Выводит символ #  
    _putch('\n'); // Переводит строку  
    _putch('*'); // Выводит символ *  
}
```



Функция вывода строки puts

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

```
int puts(const char *str); // Прототип функции
```

При ошибке возвращает EOF (-1). При успешном выполнении возвращает неотрицательное значение.

Пример использования:

```
// Демонстрация работы функции puts
```

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

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    char *str = "Hello";
```

```
    puts(str);
```

```
    puts("World");
```

```
}
```



Функция форматированного вывода printf

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

```
int printf(const char *format, ...);
```

Первый аргумент функции – строка формата.

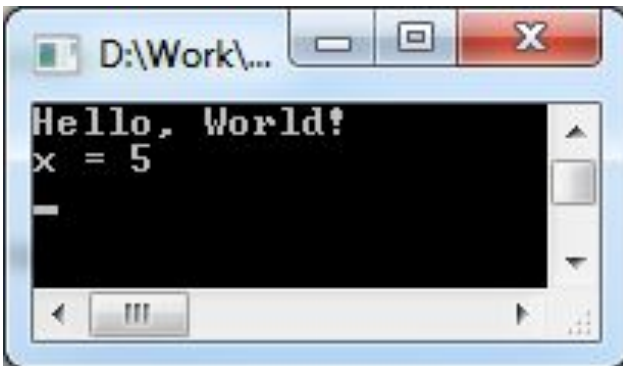
Строка формата может содержать:

- Текст
- Escape-последовательности (начинается с \)
- Спецификации формата (начинается с %)

```
printf("Hello, "); // Строка формата содержит текст.
```

```
printf("World!\n"); // Текст и escape-последовательность.
```

```
printf("x = %d\n", x); // Текст, escape-последовательность и  
// спецификацию формата.
```



Спецификация формата функции printf

`%[flags] [width] [.precision] [{h | l | ll | w | I | I32 | I64}] type`

`%[флаг][ширина][.точность][размер]тип`

Символ типа	Формат вывода
Целые	
d	Десятичное целое число со знаком.
i	Десятичное целое число со знаком.
o	Восьмеричное целое число без знака.
u	Десятичное целое число без знака.
x	Шестнадцатеричное целое число без знака, использует "abcdef".
X	Шестнадцатеричное целое число без знака, использует "ABCDEF".

Символ типа	Формат вывода
Вещественные	
e	Значение со знаком, имеющее формат [–]d.dddd e [знак]dd[d], где d – одна десятичная цифра, dddd – одна или несколько десятичных цифр, ddd три десятичные цифры и знак – символ + или –.
E	Аналогичен формату e, за исключением того, что вместо e порядок представляет E.
f	Значение со знаком, имеющее формат [–]dddd.dddd где dddd – одна или несколько десятичных цифр. Количество цифр перед десятичной точкой зависит от порядка числа, а количество цифр после десятичной точки зависит от указанной точности.
g	Значения со знаком отображаются в формате f или e, в зависимости от того, какой формат является более компактным для заданного значения и точности. Формат e используется только когда порядок значения меньше –4 или больше или равен значению поля точности. Нули в конце отбрасываются, а десятичная запятая отображается только в том случае, если за ней следует хотя бы одна цифра.
G	Аналогичен формату g, за исключением того, что вместо e порядок представляет E (где это необходимо).

Символ типа	Формат вывода
Символы	
c	Один печатный символ или управляющий код.
Указатели	
p	Отображает аргумент как адрес в шестнадцатеричных цифрах.
Строки	
s	Строку из одного или многих символов или управляющих кодов. Символы отображаются до первого символа null или до тех пор, пока не будет достигнуто значение поля точности.

В спецификации формата может быть несколько флагов, при этом они могут располагаться в любом порядке

Флаг	Назначение
–	Выровнять по левому краю результат в пределах заданной ширины поля.
+	Использование знака (+ или -) как префикс выходного значения, если оно имеет тип со знаком.
0	Если width предшествует префикс 0, начальные нули добавляются до тех пор, пока не будет достигнута минимальная ширина. Если и 0, и – присутствуют, 0 игнорируется. Если 0 указывается как формат целого числа (i, u, x, X, o, d) и присутствует спецификация точности, например, %04.d, — 0 игнорируется.
пробел	Использование пробела в качестве префикса выходного значения, если оно знаковое и положительное. Пробел игнорируется, если присутствуют и пробел, и флаг +.
#	С форматами o, x или X, флаг # использует 0, 0x или 0X соответственно как префикс любого отличного от нуля выходное значения.
	С форматами e, E, f, а или A флаг # требует присутствие десятичной запятой в выходном значении.
	С форматами g или G, флаг # требует присутствие десятичной запятой в выходном значении и предотвращает усечение хвостовых нулей. Игнорируется при использовании с c, d, i, u или s.

Спецификация ширины

- Неотрицательное десятичное число, определяющее минимальное количество выводимых символов, или звёздочка (*).
- Указание ширины никогда не приводит к отбрасыванию значений.
- Если ширина задана звёздочкой (*), в качестве её значения берётся целый аргумент из списка аргументов.

Спецификация точности

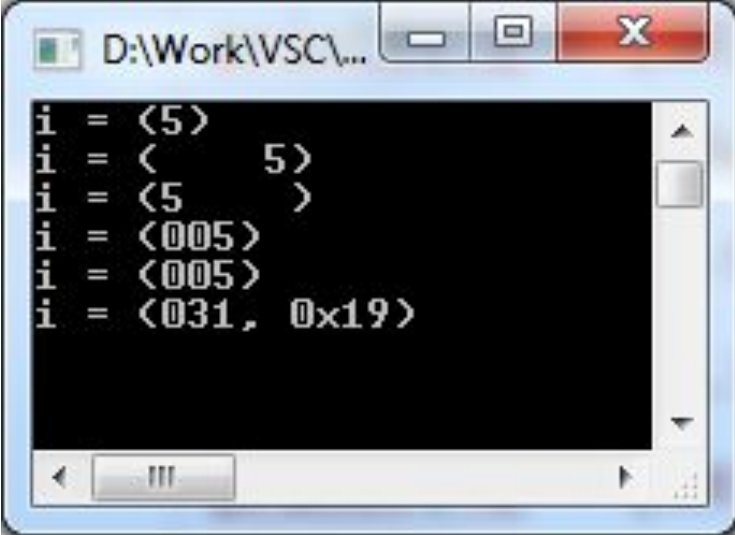
Тип	Назначение	Default
c	Точность не имеет эффекта.	
d, i, u, o, x, X	Точность определяет минимальное число цифр, которые нужно напечатать. Если число цифр в аргументе меньше, чем точность, выходное значение дополняется слева нулями. Значение не усекается, когда число знаков превышает precision.	1
e, E	Число знаков дробной части для печати задается спецификатором точности. Последняя напечатанная цифра округлена.	6
f	Значение точности задает число цифр после десятичной запятой. Если десятичная запятая присутствует, по крайней мере одна цифра отображается после нее. Значение округляется до соответствующему числу цифр.	6
g, G	Точность определяет максимальное количество значащих цифр для печати.	6
s	Точность определяет максимальное число символов для печати. Символы свыше precision не будут напечатаны.	-

Спецификация размера

Тип аргумента	Используемый префикс	Со спецификатором типа.
long int	l (строчная L)	d, i, o, x или X
long unsigned int	l	o, u, x или X
long long	ll	d, i, o, x или X
short int	h	d, i, o, x или X
short unsigned int	h	o, u, x или X
long double	l или L	a, A, e, E, f, g или G

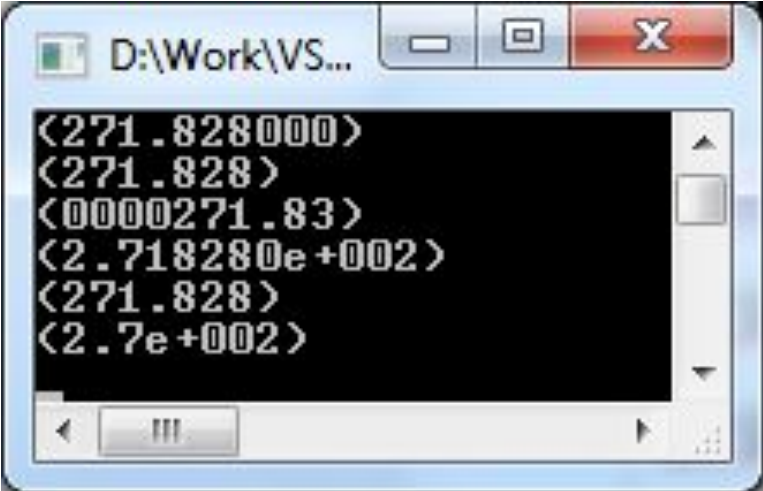
Примеры:

```
int i = 5;
printf("i = (%d)\n", i);
printf("i = (%5d)\n", i);
printf("i = (%-5d)\n", i);
printf("i = (%.3d)\n", i);
printf("i = (%03d)\n", i);
printf("i = (%#o, %#x)\n", i*5, i*5);
```



```
i = <5>
i = < 5>
i = <5 >
i = <005>
i = <005>
i = <031, 0x19>
```

```
double y = 271.828;
printf("(%f)\n", y);
printf("(%.3f)\n", y);
printf("(%010.2f)\n", y);
printf("(%e)\n", y);
printf("(%g)\n", y);
printf("(%.2g)\n", y);
```



```
<271.828000>
<271.828>
<0000271.83>
<2.718280e+002>
<271.828>
<2.7e+002>
```