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	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
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	DEL 007F
80	Ç 00C7	ü 00FC	é 00E9	â 00E2	ä 00E4	à 00E0	å 00E5	ç 00E7	ê 00EA	ë 00EB	è 00E8	ï 00EF	î 00EE	ì 00EC	Ä 00C4	Å 00C5
90	É 00C9	æ 00E6	Æ 00C6	ô 00F4	ö 00F6	ò 00F2	û 00FB	ù 00F9	ÿ 00FF	Ö 00D6	Ü 00DC	¢ 00A2	£ 00A3	¥ 00A5	₧ 20A7	ƒ 0192
A0	á 00E1	í 00ED	ó 00F3	ú 00FA	ñ 00F1	Ñ 00D1	ª 00AA	º 00BA	¿ 00BF	⌐ 2310	¬ 00AC	½ 00BD	¼ 00BC	¡ 00A1	« 00AB	» 00BB
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	α 03B1	ß 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	∩ 2229
F0	≡ 2261	± 00B1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00F7	≈ 2248	° 00B0	∙ 2219	· 00B7	√ 221A	ⁿ 207F	² 00B2	■ 25A0	NBSP 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	є 0454	Ї 0407	ї 0457	Ў 040E	ў 045E	° 00B0	∙ 2219	· 00B7	√ 221A	№ 2116	¤ 00A4	■ 25A0	NBSP 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	Ђ 0402	Ѓ 0403	‚ 201A	ѓ 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	‹ 2039	Њ 040A	Ќ 040C	Ћ 040B	Џ 040F
90	ђ 0452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122	љ 0459	› 203A	њ 045A	ќ 045C	ћ 045B	џ 045F
A0	NBSP 00A0	Ў 040E	ў 045E	Ј 0408	¤ 00A4	Ґ 0490	¦ 00A6	§ 00A7	Ё 0401	© 00A9	Є 0404	« 00AB	¬ 00AC	- 00AD	® 00AE	Ї 0407
B0	° 00B0	± 00B1	І 0406	і 0456	ґ 0491	µ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	є 0454	» 00BB	ј 0458	Ѕ 0405	ѕ 0455	ї 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
\a	Звонок (предупреждение)
\b	Удаление предыдущего символа
\f	Перевод страницы
\n	Новая строка
\r	Возврат каретки
\t	Горизонтальная табуляция
\v	Вертикальная табуляция
\'	Одиночная кавычка
\"	Двойная кавычка
****	Обратная косая черта
\?	Литерал вопросительного знака
\ooo	Символ ASCII в восьмеричной нотации
\xhh	Символ ASCII в шестнадцатеричной нотации

Функции вывода

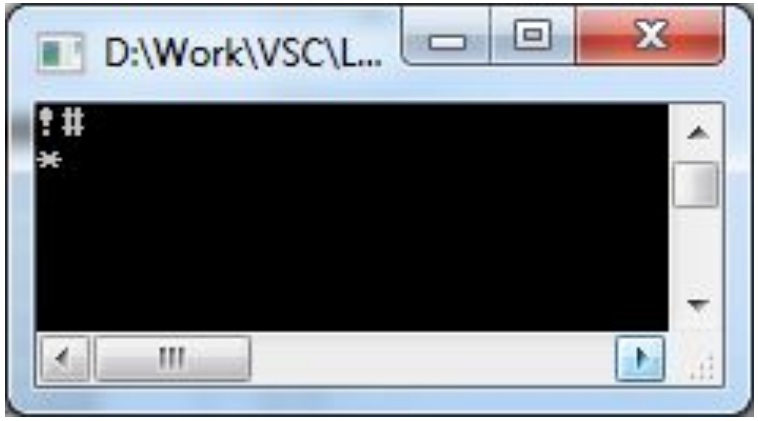
Функция вывода символа _putch

```
#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

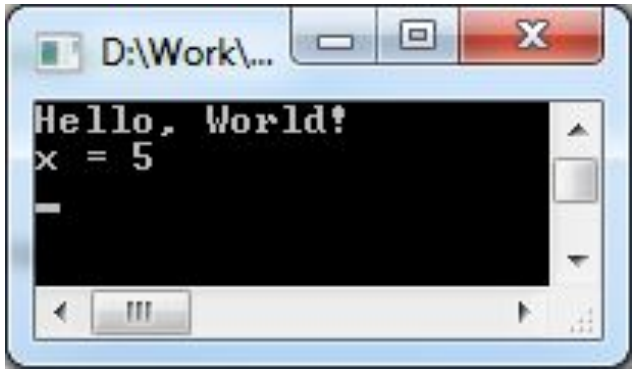
```
#include <stdio.h>
int printf(const char *format, ...);
```

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 края 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, a 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 десятеричное 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

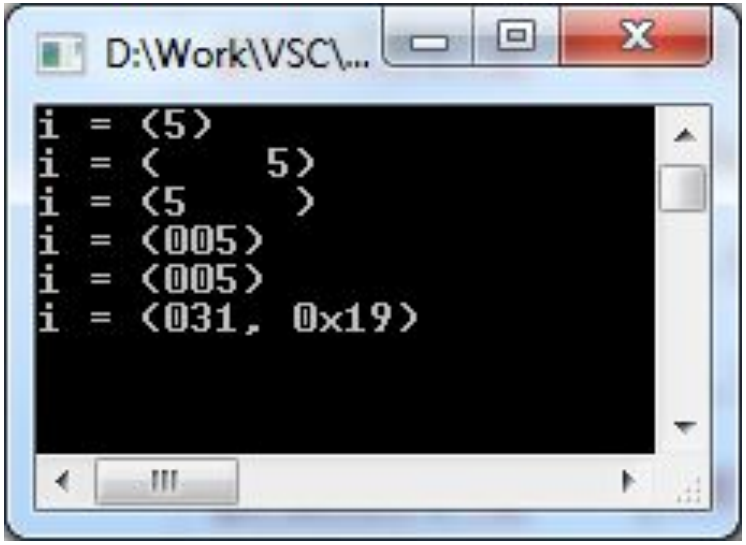
Тип	Назначение	Default
c	Точность не имеет эффекта.	
d, i, u, o, x, X	Точность определяет минимальное число цифр, которые нужно напечатать. Если число цифр в аргументе меньше, чем точность, выходное значение дополняется слева нулями. Значение не усекается, когда число знаков превышает precision.	1
e, 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 числа 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.	-

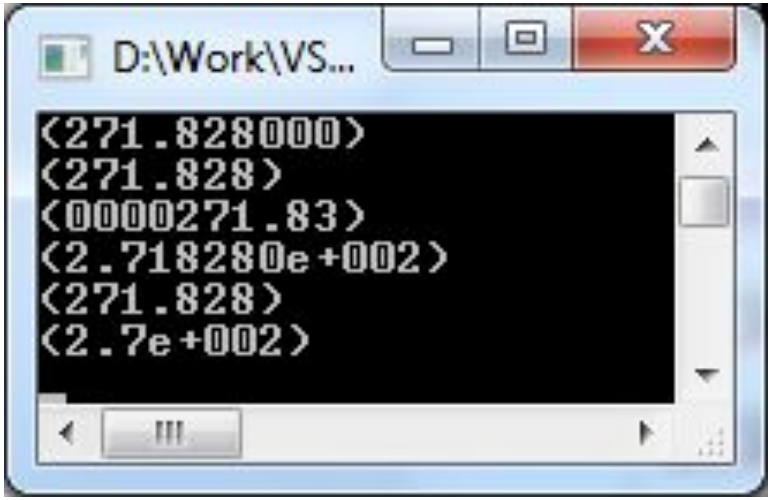
Спецификация размера

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);
```




```
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);
```