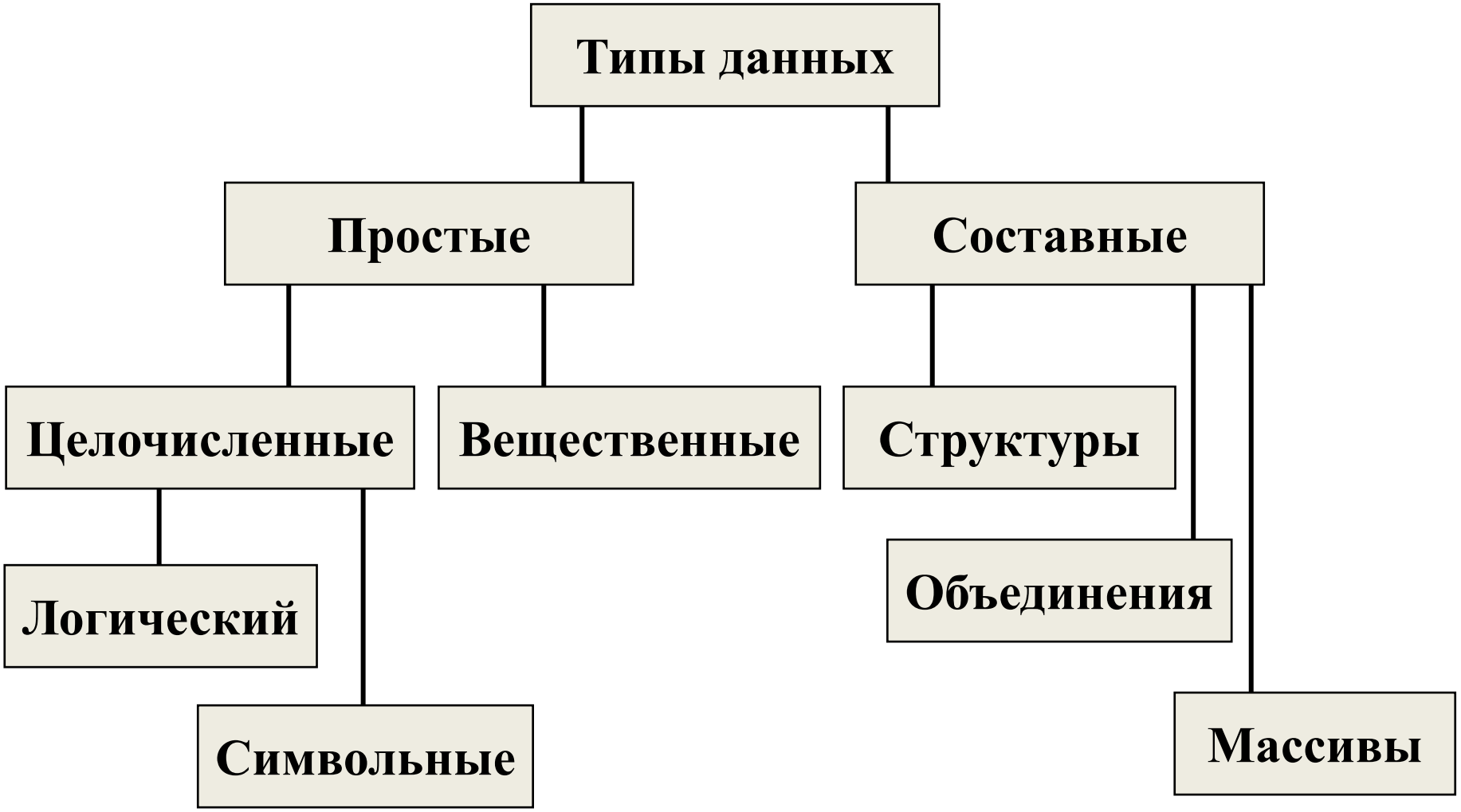
Тема 2.
Типы данных

Классификация типов данных С



Основные (встроенные) типы данных

Переменная – это ячейка в памяти компьютера, которая имеет имя (**идентификатор**) и хранит некоторое значение определенного типа.

- Значение переменной может меняться во время выполнения программы.
- При записи в ячейку нового значения старое стирается.

bool - логические значения

int - целые числа

char - символы

float, double - вещественные числа (числа с плавающей запятой)

Основы синтаксиса языка С

Идентификаторы

Имена, использующиеся для переменных, функций, меток и других определяемых пользователем объектов, называются **идентификаторами**. В идентификаторе могут использоваться буквы латинского алфавита, цифры и знак подчеркивания "_", но он не может начинаться с цифры. Прописные и строчные буквы различаются. Идентификаторы не должны совпадать с ключевыми словами языка программирования.

Примеры идентификаторов

```
_code
TEXT
Text
icon16_16
iCountPersonOfBase
screen_width
```

Правильные и неправильные идентификаторы

`Name`	правильно
`Имя`	неправильно
`12z`	неправильно
`_12z`	правильно
`dValue`	правильно
`B34_X`	правильно
`VAL_ПЕРЕМ`	неправильно
`__M2__`	правильно

Логический тип данных

Тип данных bool (включен в С только начиная с С99)

Переменным типа **bool** могут быть присвоены только значения **false** или **true**, которые являются зарезервированными словами. Если переменной логического типа присвоено целое значение, то 0 интерпретируется как **false**, а значение, не равное нулю, как **true**. В памяти **bool** занимает 1 байт.

Пример кода

```
bool var1 = false;

bool var2 = true;
```

Целочисленные типы данных

Тип данных char	Тип данных int
Переменным типа **char** могут быть присвоены целые значения. В памяти **char** занимает 1 байт.	Переменным типа **int** могут быть присвоены целые значения. Размер занимаемой памяти зависит от платформы.

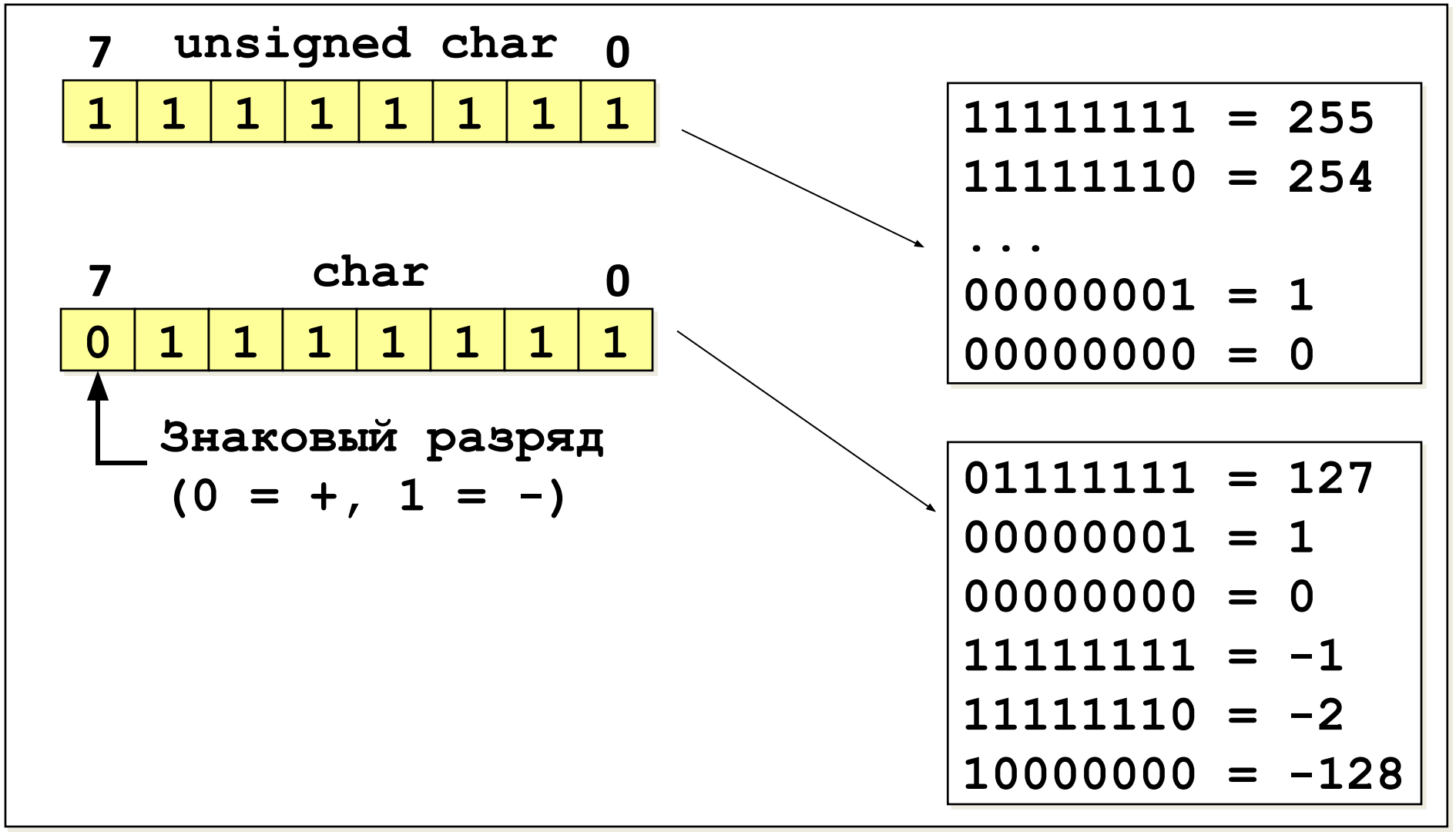
Модификаторы

signed **unsigned** **short** **long**

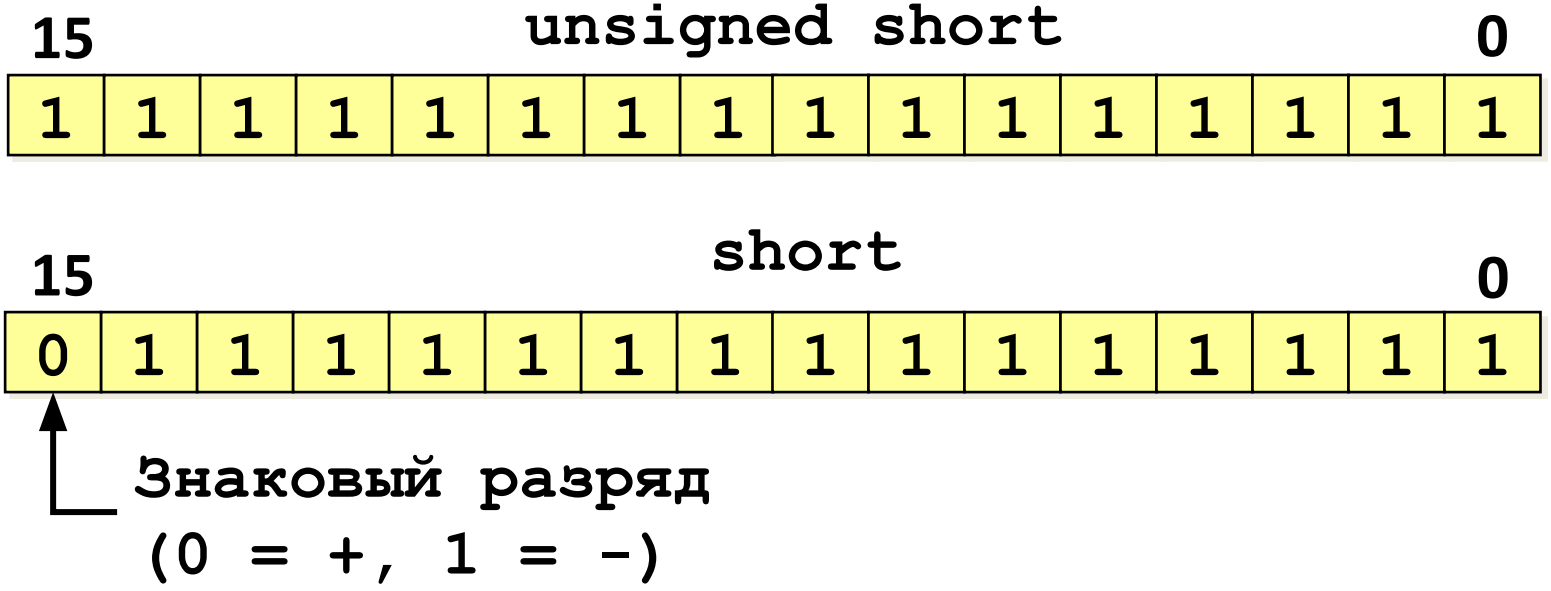
signed означает целое число со знаком, **unsigned** - целое без знака.

short явно задает размер 2 байта, **long** - 4 байта.

Тип данных char



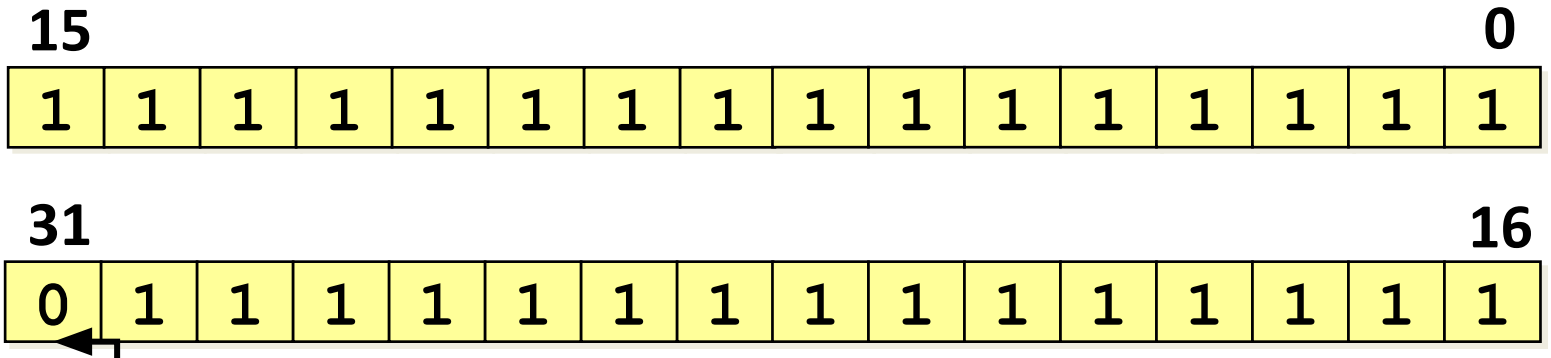
Тип данных short




```
0111111111111111 = 32767
0000000000000001 = 1
0000000000000000 = 0
1111111111111111 = -1
1000000000000000 = -32768
```

```
1111111111111111 = 65535
1111111111111110 = 65534
0000000000000010 = 2
0000000000000001 = 1
0000000000000000 = 0
```

Тип данных long



long

01111111111111111111111111111111 = 2147483647 = $2^{31}-1$
10000000000000000000000000000000 = -2147483648 = -2^{31}

unsigned long

11111111111111111111111111111111 = 4294967295 = $2^{32}-1$
10000000000000000000000000000000 = -4294967296 = -2^{32}

Сводная таблица целочисленных типов данных

Тип	Размер	Диапазон
`bool`	1	`false, true`
`signed char`	1	-128 … 127
`unsigned char`	1	0 … 255
`signed short int`	2	-32768 … 32767
`unsigned short int`	2	0 … 65535
`signed long int`	4	-2^{31} … $2^{31}-1$
`unsigned long int`	4	0 … $2^{32}-1$

`short = short int`

`long = long int`

`signed` – по умолчанию

Целочисленные константы

Целочисленные константы могут записываться в десятичном, восьмеричном и шестнадцатеричном видах. Восьмеричные константы начинаются с **0**, шестнадцатеричные - с **0x** или **0X**.

Пример кода

```
int a = 15;
int b = 015; //b = 13
int c = 0x15;  //b = 21
int d = 0X15;  //b = 21
```

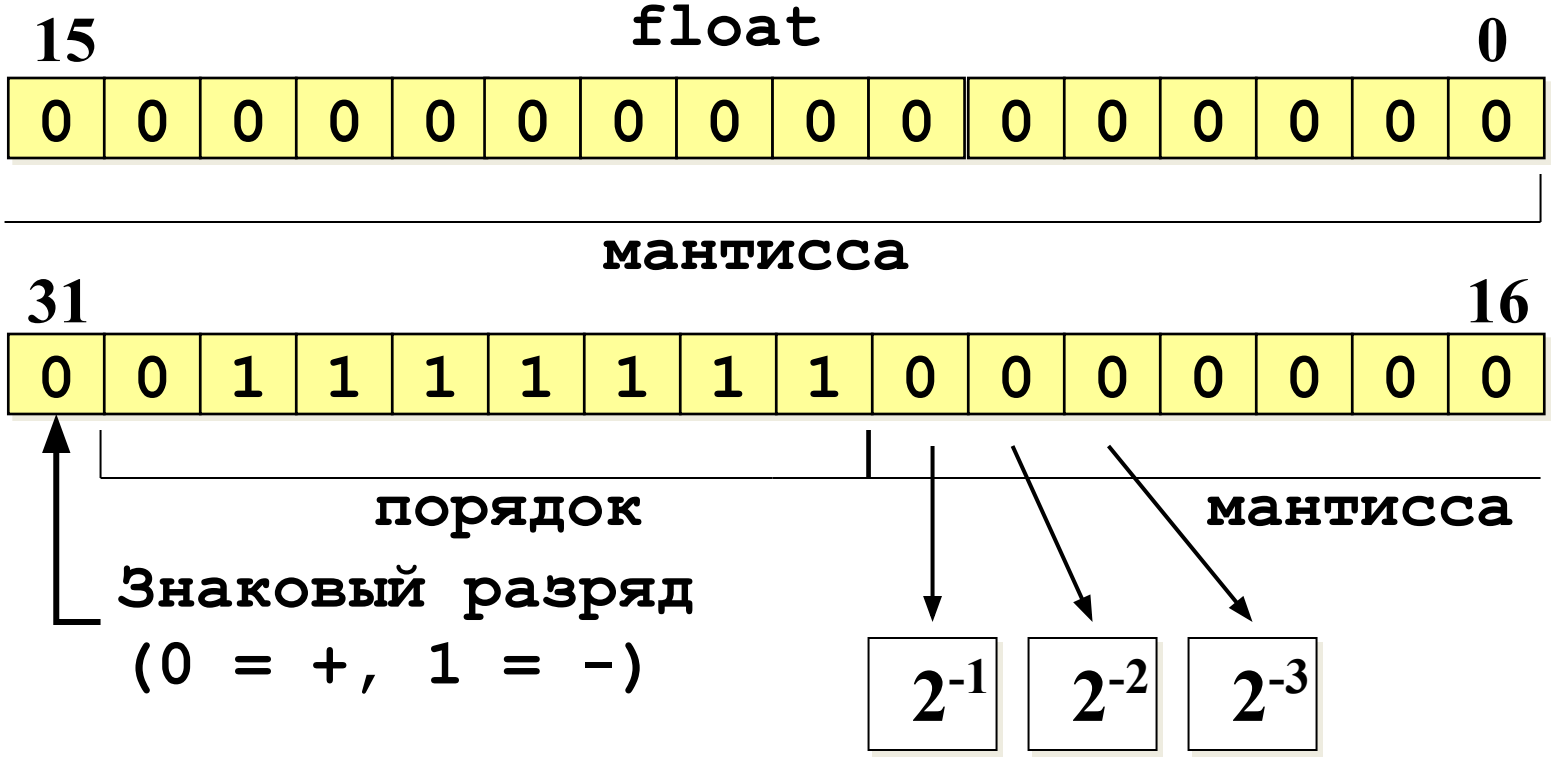
Вещественные типы данных

Тип данных float
Переменным типа **float** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **float** занимает 4 байта.

Тип данных double
Переменным типа **double** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **double** занимает 8 байтов.

Тип данных long double
Переменным типа **long double** могут быть присвоены вещественные значения в формате с плавающей точкой. В памяти **long double** занимает 10 байтов.

Представление вещественных типов данных



$$Число = (1+мантисса)*2^{(порядок-127)}$$

Сводная таблица вещественных типов данных

Тип	Размер	Диапазон	Точность
`float`	4	$3.4*10^{-38}$ … $3.4*10^{38}$	7
`double`	8	$1.7*10^{-308}$ … $1.7*10^{308}$	15
`long double`	10	$3.4*10^{-4932}$ … $3.4*10^{4932}$	19

Точность чисел с плавающей точкой
123000 - 3 значащих цифры
0.045 - 2 значащих цифры
123000.045 - 9 значащих цифр

Вещественные константы

Вещественные константы включают дециальную точку, либо могут использовать экспоненциальный формат (**e** или **E**). Тип константы по умолчанию – **double**, с помощью модификаторов **F** или **f** можно задать константу типа **float**, а с помощью **L** или **l** – типа **long double**.

Пример кода

```
float a = 5.67F;
float b = .25f;
float c = 5.67e4F;
float d = 2E-2F;
double e = 5.67;
double f = .25;
double g = 5.67e4;
long double h = 2.1234E-2L;
```

Символы. Кодирование символов

Тип данных char
Для представления символов используется тип данных **char**. В памяти **char** занимает 1 байт.

‘0’ = 48	*‘A’ = 65*	*‘a’ = 97*
‘1’ = 49	*‘B’ = 66*	*‘b’ = 98*
…	…	…

Код ASCII

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Расширенные кодировки символов

Кодовая таблица Windows (CP-1251)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				©	Ё	§	Є	·		°						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2		!	"	#	$	%	&		(	)	*	+				/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}]	~	□
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	Ъ	Г	,	ѓ	„	…	†	‡		‰	Љ	‹	Њ	Ќ	Ћ	Џ
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	ђ	‘	’	"	"		–	—		™	љ	›	њ	ќ	ћ	џ
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A		Ў	ў	J	¤	Ґ	¦	§	Ё	©	Є	«		-	®	Ї
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	°	±	І	і	ґ	µ		·	ё	№	є	»	j	S	s	ї
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	▪ 223
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

kod-win1

Символьные константы

Символьные константы состоят из одного символа, заключенного в апострофы. Тип константы – **char**. Для представления непечатаемых символов используется управляющие последовательности, начинающаяся с символа \ (обратный слэш).

Пример кода

```
char a = 'a';
char b = '1';
char c = '\t';    // табуляция
char d = '\r';    // возврат каретки
char e = '\n';    // перевод строки
char f = '\062'; // символ с кодом 062
char g = '\"';    // символ "
char h = '\\';    // символ \
```

Строковые константы (литералы)

Строковые константы состоят из нескольких символов, заключенных в апострофы. Тип константы – **char***. Для представления непечатаемых символов используется комбинация, начинающаяся с символа \ (обратный слэш). В конце каждого строкового литерала компилятор добавляет нуль-терминатором **'\0'**. Поэтому длина строки всегда на 1 байт больше количества символов в ее записи.

Пример кода

```
char* a = "abcdef";
char* b = "a";
char* c = "";     // пустая строка
char* d = "Вывод на терминал\r\n";
char* e = "Оценка \"отлично\"";
```

Определение размера типов данных

Оператор sizeof()
Оператор **sizeof()** позволяет определить размер в байтах, занимаемый в памяти типом данных или переменной.

Пример кода
`int i = sizeof(char);` `int j = sizeof(i)`

Выбор типов данных