

# Технология программирования на ЭВМ

## Символьный тип

Баев А.Ж.

Казахстанский филиал МГУ

08 ноября 2019

## ASCII таблица.

ASCII (American standard code for information interchange) — таблица, в которой представлены основные символы с номерами от 0 до 127. Непечатные 0–31. Печатные 32–127.

номер	0	1	2	3	4	5	6	7	8	9
30	неп	неп		!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			

# Константные символы.

Как можно записать букву в переменную:

Хорошо

```
1 char ch;  
2 ch = 'М';  
3 ch = 'm';  
4 ch = '1';  
5 ch = '!';  
6 ch = '␣';
```

Плохо

```
1 char ch;  
2 ch = 77;  
3 ch = 109;  
4 ch = 49;  
5 ch = 33;  
6 ch = 32;
```



## Окончание файла.

EOF — символ окончания файла.

В терминале такой символ генерируется при нажатии `Ctrl+d` при пустой строке.

Если в строке имеется какой-либо текст и он еще не передан на вход (если `ENTER` еще не нажат), то нажатие отправляет в буфер этот текст. А повторное нажатие генерирует конец файла.

# Тип char — числовой.

Буквы идут в таблице подряд:

```
1      char letter = 'A';  
2      letter = letter + 2;
```

Цифры идут в таблице подряд:

```
1      char digit = '0';  
2      digit = digit + 5;
```

# Буква и её порядковый номер.

Порядковый номер буквы в алфавите

```
1   char letter = 'D';  
2   int index = letter - 'A' + 1;
```

# Строчные и заглавные буквы.

Перевод из строчной буквы в заглавную

```
1 char letter = 'D';  
2 letter = letter - 'A' + 'a';
```

Перевод из заглавной буквы в строчную

```
1 char letter = 'd';  
2 letter = letter - 'a' + 'A';
```



# Символ цифры и её порядковый номер.

Перевод порядкового номера цифры в реальное значение

```
1 char digit = '7';  
2 int value = digit - '0';
```

# Неправильная сумма.

Что окажется в переменной a?

```
1      char a;  
2      a = '2' + '3';
```

## Десятичное число.

Даны две цифры как символы:

```
1 char a = '2', b = '5';
```

Составить из них целое двухзначное число.

```
1 int x = 10 * (a - '0') + (b - '0');
```

## Симметричная с конца буква

Дана буква от 'a' до 'z'. Вывести симметричную с конца букву, например, букву 'd' заменить на 'w'.

```
1 char letter = 'd';  
2 char position = letter - 'a';  
3 char sym = 'z' - position;
```

```
1 position = 100 - 97  
2 sym = 'z' - 3
```

# Какого типа символ

## Цифра

```
1      if (ch >= '0' && ch <= '9')
```

## Строчная буква

```
1      if (ch >= 'a' && ch <= 'z')
```

## Заглавная буква

```
1      if (ch >= 'A' && ch <= 'Z')
```

## Буква

```
1      if ((ch >= 'a' && ch <= 'z') ||  
2          (ch >= 'A' && ch <= 'Z'))
```

# Ввод-вывод символа

## Ввод

```
1 int getchar()
```

## Например

```
1 char ch;  
2 ch = getchar();
```

## Вывод

```
1 int putchar(int)
```

## Например

```
1 char ch = 'A';  
2 putchar(ch);
```

## Новая строка

Нажатие клавиши Enter генерирует символ `\n`. Он тоже считывается `getchar()`.

Чтобы сделать перенос на новую строку:

```
1 putchar( '\n' );
```

## Ввод-вывод символа

Распечатать символ и его код

```
1 char ch = 'A';  
2 printf("%c %d", ch, ch);
```

Вводятся 2 пары шахматных координат, то есть A5 B7:

```
1 char c1, r1, c2, r2;  
2 scanf("%c%c %c%c", &c1, &r1, &c2, &r2);
```



# Вывод таблицы ascii

Распечатать все заглавные буквы

```
1 char ch = 'A';  
2 while (ch <= 'Z') {  
3     putchar(ch);  
4     ch++;  
5 }
```

Распечатать все строчные буквы и их ascii-номера

```
1 char ch = 'a';  
2 while (ch <= 'z') {  
3     print("%c□%d\n", ch, ch);  
4     ch++;  
5 }
```

## Пример. Вывод ASCII кодов.

Дана строка символов. Вывести все ASCII-коды символов и их общее количество.

Ввод	Astana 2017
Вывод	65 115 116 97 110 97 32 50 48 49 55 46 12

## Решение. Вывод ASCII кодов.

```
1  #include <stdio.h>
2  int main(){
3      int n = 0;
4      char ch = getchar();
5      while (ch != '\n') {
6          n++;
7          printf("%d□", (int)ch);
8          ch = getchar();
9      };
10     printf("\n%d\n", n);
11     return 0;
12 }
```

## Пример. Фильтр цифр.

Дана строка (последовательность символов с переносом строки в конце). Вывести все символы, кроме цифр.

Ввод	Astana 2018-2019!
Вывод	Astana -!

Идея: у не цифр код меньше '0' или больше '9'.

## Решение. Фильтр цифр.

```
1 #include <stdio.h>
2 int main(){
3     char ch = getchar();
4     while (ch != '\n') {
5         if (ch < '0' || ch > '9') {
6             putchar(ch);
7         }
8         ch = getchar();
9     }
10    return 0;
11 }
```



## Решение. Смена регистра.

```
1 #include <stdio.h>
2 int main() {
3     char ch;
4     ch = getchar();
5     while (ch != EOF) {
6         if (ch >= 'a' && c <= 'z') {
7             ch += 'A' - 'a';
8         }
9         putchar(ch);
10        ch = getchar();
11    }
12    putchar('\n');
13    return 0;
14 }
```

## Пример. Статистика.

Дан текст, ввод которого завершается концом файла.  
Посчитать, сколько в этой последовательности букв и  
сколько цифр.

Ввод	Astana 2017 and 2018!
Вывод	9 8



## Пример. Статистика.

```
1 #include <stdio.h>
2 int main(){
3     int letters = 0, digits = 0;
4     char ch = getchar();
5     while (ch != EOF) {
6         if (ch >= 'A' && ch <= 'Z' ||
7             ch >= 'a' && ch <= 'z') {
8             letters++;
9         }
10        if (ch >= '0' && ch <= '9') {
11            digits++;
12        }
13        ch = getchar();
14    }
15    printf("%d□%d\n", letters, digits);
```



## Пример. Однозначный калькулятор.

```
1 #include <stdio.h>
2 int main(){
3     char left, sign, right;
4     int ans = 0;
5     left = getchar() - '0';
6     sign = getchar();
7     right = getchar() - '0';
8     if (sign == '+') {
9         ans = left + right;
10    }
11    if (sign == '*')
12        ans = left * right;
13    }
14    printf("%d\n", ans);
15    return 0;
```

## Подводные камни

Не забывайте, что тип `char` с точки зрения арифметики ограничен диапазоном  $[-128; 127]$ . Например, данный код

```
1      char ch = 'z';
2      ch = ch + 20;
3      if (ch > 'z') {
4          puts("ok");
5      } else {
6          puts("magic");
7      }
```

подаст на вывод «magic», так как  $122 + 20 = 142$ , но ответ берется по модулю 256 из диапазона  $[-128; 127]$  и будет равен  $(-116)$ .

## Подводные камни

Чтобы избежать такой проблемы, используйте беззнаковый тип или тип `int`. Например, данный код

```
1 unsigned char ch = 'z';  
2 ch = ch + 20;  
3 if (ch > 'z') {  
4     puts("ok");  
5 } else {  
6     puts("magic");  
7 }
```

подаст на вывод «ok».

## Умный printf

Если вы хотите, чтобы вывод целого числа занимал фиксированное число ячеек (например 5), то можно использовать спецификатор ширины и нулевого заполнителя. Сравните:

```
1 printf("%d", 12);  
2 printf("%5d", 12);  
3 printf("%05d", 12);
```

```
1 12  
2    12  
3 00012
```

## Умный scanf

Если вы хотите считывать сложную строку (например дату в формате «dd/mm уууу») то можно это описать с помощью форматной строки. Сравните:

```

1      scanf ("%d", &dd);
2      getchar();
3      scanf ("%d", &mm);
4      getchar();
5      scanf ("%d", &уууу);
  
```

Здесь `getchar()` считывает (пропускает) слеш и пробел.

```

1      scanf ("%d/%d□%d", &dd, &mm, &уууу);
  
```

# Летучка

1. В переменной `char ch`; записан некоторый символ. Выписать логическое выражение, которое принимает значение истина, если этот символ — буква.
2. Сколько видимых символов будет напечатано? Сколько строк это займет? Напишите с отступами напечатанный текст:  

```
printf("n\ntnt\tn\n\t\\n");
```
3. Чему равно выражение (без явных значений кодов ASCII таблицы):
  - a) `'3' + 5`;
  - б) `'Z' - 'A'`;
  - в) `'+' - '+'`?
4. Напишите программу, которая печатает на экран все заглавные буквы ASCII таблицы в обратном порядке.
5. Дан текст, который заканчивается точкой. Данный код должен вычислять количество слов. Приведите 2