**Массивы. Часть 4. Примеры решения задач с использованием строк символов**

В данной теме решаются некоторые задачи с использованием строк символов. При решении этих задач используются стандартные функции из библиотеки «string.h».

Данная тема есть продолжением изучения тем: [**одномерные массивы**](https://www.bestprog.net/ru/2017/03/01/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-1-%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D0%B0-%D0%BE/) и [**многомерные массивы**](https://www.bestprog.net/ru/2017/03/04/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-2-%D0%B4%D0%B2%D1%83%D0%BC%D0%B5%D1%80%D0%BD%D1%8B%D0%B5-%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D0%BC%D0%B0%D1%81/).

**Содержание**

1. [Задачи на **поиск** символа в строке. Примеры](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q01)
2. [Задачи на **замену** символов. Пример](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q02)
3. [Задачи на **определение позиции** заданного символа. Примеры](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q03)
4. [Задачи на **преобразование текста**. Примеры](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q04)
5. [Задачи на **подсчет числа символов** в тексте. Примеры](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q05)
6. [**Сортировка** строк символов методом «пузырька». Пример](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#q06)

**1. Задачи на поиск символа в строке. Примеры**

**Пример 1.** **Задана строка символов. Определить, есть ли заданный символ c в этой строке символов.**

**// поиск символа в строке**

**char S[50]; // строка символов**

**char c; // искомый символ**

**int i; //для организации цикла**

**bool f\_is;** **// f\_is=true - символ есть в строке, иначе f\_is=false**

**// ввод строки S**

**// ...**

**// ввод символа c**

**// ...**

**for (i=0; i<strlen(S); i++)**

 **if (S[i]==c)**

 **{**

 **f\_is = true; // символ c есть в строке S**

 **break;**

 **}**

**if (f\_is)**

 **label1->Text = "Символ " + c.ToString() + " есть в строке";**

**else**

 **label1-> Text = "Символа " + c.ToString() +** "**нет в строке**";

В данном примере для определения длины строки S используется функция

**strlen(S)**

Эта функция возвращает число символов в строке без учета последнего символа **‘\0’**. Чтобы использовать эту функцию в Visual C++ нужно в начале файла перед определением пространства имен подключить одну из библиотек <string.h> или <cstring>. Для этого нужно набрать одну из строк:

**#include <string.h>**

или

**#include <cstring>**

**Пример 2.** Пусть задан некоторый текст. Вычислить, сколько раз повторяется наперед заданный символ **a**.

// нахождение числа вхождений символа в строке

**char S[50]; // строка символов**

**char a; // заданный символ**

**int i;**

**int k;** // результат - число вхождений символа a в строке S

// ввод строки S

// ...

// ввод символа a

// ...

**k = 0; // в начале обнулить счетчик k**

**for (i=0; i<strlen(S); i++)**

 **if (S[i]==a)**

 **k++; // увеличить счетчик на 1**

**Пример 3.** В данном тексте **посчитать число символов ‘+’ и ‘ — ‘.**

// подсчет числа символов в строке

**char str[50]; // заданный текст**

**int i;**

**int n\_p; // результат - число символов '+'**

**int n\_m; // результат -** **число символов '-'**

// ввод массива str

// ...

**n\_p = 0;**

**n\_m = 0;**

**for (i=0; i<strlen(str); i++)**

**{**

 **if (str[i] == '+')**

 **n\_p++;**

 **if (str[i] == '-')**

 **n\_m++;**

**}**

**2. Задачи на замену символов. Примеры**

**Пример 1.** **В заданном тексте заменить все символы** **‘+’ на ‘ — ‘.**

**// замена символов**

**char str[50]; // заданный текст**

**int i;**

**// ввод текста**

**// ...**

**for (i=0; i<strlen(str); i++)**

 **if (str[i] == '+')**

 **str[i] = '-';**

**Пример 2.** **Пусть дано несколько строк текста (двумерный массив символов). Заменить все символы ‘+’ на ‘ — ‘.**

// замена символов в массиве строк

**char str[5][50]; // заданный массив, который содержит 5 строк**

**int i, j; // для организации цикла**

**// ввод массива из 5 строк**

**// ...**

**for (i=0; i<5; i++)**

 **for (j=0; j<strlen(str[i]); j++)**

 **if (str[i][j] == '+')**

 **str[i][j] = '-';**

**Пример 3.** В заданном тексте заменить все символы **‘+’** на **‘+++’**.

Данная задача легко решается с помощью введения вспомогательного массива в текст программы.

// замена символов '+' на '+++'

char str[50]; // заданный текст

int i, j;

char str2[150]; // вспомогательный массив

// ввод массива str

// ...

j = 0; // текущая позиция в массиве str2

for (i=0; i<strlen(str); i++)

 if (str[i] == '+')

 {

 str2[j++] = '+';

 str2[j++] = '+';

 str2[j++] = '+';

 }

 else

 {

 str2[j] = str[i];

 j++;

 }

// добавить символ конца строки

str2[j] = '\0';

strcpy(str, str2); // скопировать str2 в str

В листинге используется функция strcpy() из библиотеки <string.h> (или cstring), которая копирует одну строку в другую.

**Пример 4.** В заданном тексте заменить последовательность символов ‘…’ на ‘ . ‘.

// замена символов '...' на '.'

char str[50]; // заданный текст

int i, j;

char str2[50]; // вспомогательный массив

int d;

// ввод массива str

// ...

j = 0; // текущая позиция в массиве str2

d = strlen(str); // длина строки str

for (i=0; i<d; i++)

 if (i<d-2)

 {

 if ((str[i]=='.')&&(str[i+1]=='.')&&(str[i+2]=='.'))

 {

 str2[j++] = '.';

 i+=2;

 }

 else

 {

 str2[j++] = str[i];

 }

 }

 else

 {

 str2[j++] = str[i];

 }

// добавить символ конца строки

str2[j] = '\0';

strcpy(str, str2); // скопировать вспомогательный массив в исходный

Так же, как и в предшествующем примере, вводится вспомогательный массив. В исходном массиве происходит проверка на последовательность символов **‘…’**

if ((str[i]=='.')&&(str[i+1]=='.')&&(str[i+2]=='.'))

Если такая последовательность найдена, то к вспомогательному массиву добавляется ‘.’

str2[j++] = '.';

 i+=2;

В другом случае, символ из исходного массива копируется в вспомогательный.

str2[j++] = str[i];

[**⇑**](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#contents)

**3. Задачи на определение позиции заданного символа. Примеры**

Пример 1. Дана некоторая строка символов. В заданном тексте определить позицию первой точки **‘ . ‘**. Считать, что первый символ в строке имеет позицию 1.

// поиск первого вхождения символа в тексте

char str[50]; // заданный текст

int i;

int pos; // искомое значение - позиция

// ввод массива str

// ...

pos = -1;

for (i=0; i<strlen(str); i++)

 if (str[i]=='.')

 {

 pos = i+1; // запомнить позицию

 break; // выход из цикла, дальнейшее выполнение цикла не имеет смысла

 }

if (pos == -1)

 label1->Text = "Символ отсутствует в тексте.";

else

 label1->Text = pos.ToString();

**Пример 2.** Задана некоторая строка символов. Определить позицию последней точки ‘ . ‘ в тексте.

// поиск последнего вхождения символа в тексте

char str[50]; // заданный текст

int i;

int pos; // искомое значение - позиция

// ввод массива str

// ...

pos = -1;

for (i=strlen(str)-1; i>=0; i--) // просмотр строки от конца к началу

 if (str[i]=='.')

 {

 pos = i+1; // запомнить позицию

 break; // выход из цикла, дальнейшее выполнение цикла не имеет смысла

 }

if (pos == -1)

 label1->Text = "Символ отсутствует в тексте.";

else

 label1->Text = pos.ToString();

[**⇑**](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#contents)

**4. Задачи на преобразование текста. Примеры**

**Пример 1.** **Задана некоторая строка символов. Создать новую строку, которая образована из данной чтением от конца до начала.**

// обращение строки

**char str[50]; // заданная строка**

**char str2[50]; // результирующая строка**

**int i;**

**int d;**

**// ввод массива str**

**// ...**

**d = strlen(str);**

**for (i=0; i<d; i++)**

 **str2[d-i-1] = str[i];**

**// добавить символ конца строки**

**str2[d] = '\0';**

**Пример 2.** Задано слово. Проверить, читается ли это слово слева направо и наоборот.

// проверка строки

char str[50]; // заданная строка

int i;

int d;

bool f\_yes; // результирующая переменная

// ввод массива str

// ...

d = strlen(str);

f\_yes = true;

for (i=0; i<d/2; i++) // просмотр строки от конца до начала

 if (str[i] != str[d-i-1])

 f\_yes = false;

if (f\_yes)

 label1->Text = "Строка читается слева направо и наоборот";

else

 label1->Text = "Строка не читается слева направо и наоборот";

**Пример 3.** Проверить, есть ли данные слова обратными между собой, то есть первое слово читается слева направо так же, как второе слово справа налево.

// проверка двух слов на обратимость

char str1[50]; // строка 1

char str2[50]; // строка 2

int i;

int d1, d2;

bool f\_yes; // результирующая переменная

// ввод массивов str1, str2

// ...

d1 = strlen(str1);

d2 = strlen(str2);

f\_yes = true;

if (d1 == d2) // проверка длин строк на совпадение

{

 for (i=0; i<d1; i++)

 if (str1[i] != str2[d1-i-1])

 f\_yes = false;

}

else

 f\_yes = false;

if (f\_yes)

 label1->Text = "Слова есть обратными между собой";

else

 label1->Text = "Слова не являются обратными между собой";

[**⇑**](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#contents)

**5. Задачи на подсчет числа символов в тексте. Примеры**

Пример 1. Задана строка символов. **Подсчитать число символов ‘+’ в этой строке.**

**// подсчет числа символов '+' в строке**

**char str[50];**

**int i;**

**int k; // результат - число символов**

**// ввод строки str**

**// ...**

**k = 0; // обнулить счетчик**

**for (i=0; i<strlen(str); i++)**

 **if (str[i] == '+')**

 **k++;// вывод результата**

**label1->Text = k.ToString();**

**Пример 2.** **Задано несколько строк символов. Подсчитать число символов ‘ — ‘ в этих строках.**

// подсчет числа символов '-' в массиве из 6 строк

char s[6][50];

int i, j;

int k; // результат - число символов

// ввод строк s

// ...

k = 0; // обнулить счетчик

// вычисление k

for (i=0; i<6; i++)

 for (j=0; j<strlen(s[i]); j++)

 if (s[i][j] == '-')

 k++;

// вывод результата

label1->Text = k.ToString();

**Пример 3.** Задана строка символов. Подсчитать число слов в этой строке. Считать, что слова разделяются одним из символов ‘ ‘ (пробел), ‘ , ‘ (запятая), ‘ . ‘ (точка).

// подсчет числа слов в строке

char s[50]; // строка символов

int i;

int k; // результат - число слов

bool f; // вспомогательная переменная - определяет конец предшествующего слова

// ввод строки s

// ...

k = 0; // обнулить счетчик

f = true;

// вычисление k

for (i=0; i<strlen(s); i++)

 if ((s[i] == ' ')||(s[i]==',')||(s[i]=='.'))

 {

 f = true; // конец слова

 }

 else

 {

 if (f)

 {

 // начало нового слова

 k++;

 f = false;

 }

 }

// вывод результата

label1->Text = k.ToString();

В вышеприведенном примере с помощью вспомогательной переменной f определяется конец предшествующего слова. Если

f = true

значит был конец предшествующего слова. Иначе, начинается новое слово и счетчик k увеличивается на 1.

**Пример 4.** Задано несколько строк символов (двумерный массив символов). Подсчитать число слов, которые начинаются из символов ‘a’ или ‘A’. Считать, что слово заканчивается на один из символов ‘ ‘ (пробел), ‘ . ‘ (точка) или ‘ , ‘ (запятая).

// подсчет числа слов в массиве строк

char s[5][50]; // массив строк символов

int i, j;

int k; // результат - число слов

bool f; // вспомогательная переменная - определяет конец предшествующего слова

// ввод массива строк s

// ...

k = 0; // обнулить счетчик

// вычисление k

for (i=0; i<5; i++) // перебор строк

{

 f = true;

 for (j=0; j<strlen(s[i]); j++)

 if ((s[i][j] == ' ')||(s[i][j]==',')||(s[i][j]=='.'))

 {

 f = true; // конец слова

 }

 else

 {

 if (f)

 {

 // начало нового слова

 if ((s[i][j]=='a')||(s[i][j]=='A'))

 k++;

 f = false;

 }

 }

}

// вывод результата

label1->Text = k.ToString();

[**⇑**](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#contents)

**6. Сортировка строк символов методом «пузырька». Пример**

**Пример.** Задано несколько строк символов. **Отсортировать эти строки в алфавитном порядке.**

**// сортировка массива строк методом "пузырька"**

**char s[5][50]; // массив строк символов**

**int i, j;**

**char ts[50]; // вспомогательная строка**

**// ввод массива строк s**

**// ...**

**for (i=0; i<4; i++)**

 **for (j=i; j>=0; j--)**

 **if (strcmp(s[j],s[j+1])>0)**

 **{**

 **// обменять строки местами**

 **strcpy(ts, s[j]);**

 **strcpy(s[j], s[j+1]);**

 **strcpy(s[j+1], ts);**

 **}**

В вышеприведенном листинге для сравнения строк в лексикографическом порядке используется функция strcmp() из библиотеки <string.h> (cstring). Функция имеет общую форму:

int strcmp(string1, string2);

где string1, string2 – строки, которые сравниваются между собой.

Функция возвращает число:

* **>0**, если string1 следует после string2 в лексикографическом порядке;
* **=0**, если строки одинаковые;
* **<0**, если string1 следует перед string2 в лексикографическом порядке.

[**⇑**](https://www.bestprog.net/ru/2017/03/13/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-4-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B-%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87/#contents)