**ЛЕКЦИЯ 8. СТРОКИ КАК МАССИВЫ СИМВОЛОВ**

Содержание

**[1 Строки как массивы символов 1](#_Toc148099382)**

**[1.1 Представление строки 1](#_Toc148099383)**

**[1.2 Передача строки в качестве параметра 2](#_Toc148099384)**

**[1.3 Список функций 7](#_Toc148099385)**

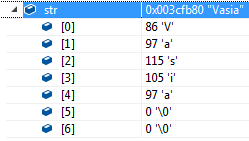
**[1.4 Массивы строк 9](#_Toc148099386)**

# Строки как массивы символов

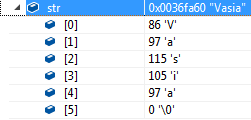
* 1. Представление строки

**Строка представляет собой массив символов, заканчивающийся нуль-символом**.

Нуль-символ — это символ с кодом, равным 0, что записывается в виде управляющей последовательности **‘\0'**. По положению нуль-символа определяется фактическая длина строки. Строку можно инициализировать строковым литералом:

**char str[7] = "Vasia";**

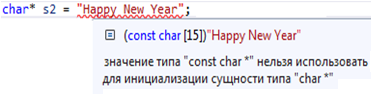
В этом примере под строку выделяется 7 байт, 5 из которых занято под символы строки, шестой — под нуль-символ, оставшийся (не использованный) байт содержит нуль-символ.

Если строка при определении инициализируется, размерность можно опускать (компилятор сам выделит соответствующее количество байт):

**char str[]{ "Vasia" };**

// выделено и заполнено 6 байт

**ВНИМАНИЕ:**

****

Оператор **char \*s2 = (char\*) "** **Happy New Year";** создает не строковую переменную, а **указатель на строковую константу, изменить которую невозможно**!

* 1. Передача строки в качестве параметра

**Пример 1**

**#include <iostream>**

**using namespace std;**

**//функции копирование строк**

**void copy1(char\*, const char\*);**

**void copy2(char\*, const char\*);**

**int main()**

**{**

**char s1[10], \*s2 = (char\*)("Hi"),**

**s3[10], s4[] = "Good Bye";**

**copy1(s1, s2);**

**copy2(s3, s4);**

**cout << "s1= " << s1 << endl;**

**cout << "s2= " << s2 << endl;**

**cout << "s3= " << s3 << endl;**

**cout << "s4= " << s4 << endl;**

**return 0;**

**}**

**void copy1(char\* str1, const char\* str2) {**

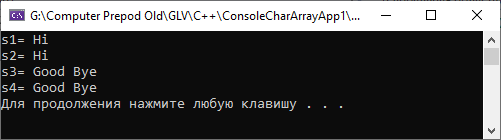
**for (int i = 0; (str1[i] = str2[i]) != '\0'; ++i);**

**}**

**void copy2(char\* str1, const char\* str2) {**

**for (; (\*str1 = \*str2) != '\0'; str1++, str2++);**

**}**

****

**Самостоятельно:**

* будет ли работать следующий код?

**char\* str1 = (char\*)"Vasia", \*str2;**

**for (int i = 0; (str2[i] = str1[i]) != '\0'; ++i);**

**cout << str2 << endl;**

* чем отличается определение строк?

**char str[10]; и**  **char str[10]{};**

**Пример 2**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**setlocale(LC\_ALL, "Russian");**

**char s1[] = "Happy Birthday to You";**

**char s2[25], s3[15];**

**cout << "строка1: " << s1 << endl;**

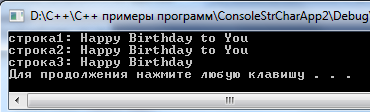
**strcpy\_s(s2, s1);**

**cout << "строка2: " << s2 << endl;**

**strncpy\_s(s3, s1, 14);**

**s3[14] = '\0';**

**cout << "строка3: " << s3 << endl;**

****

**return 0;**

**}**

**Пример 3**

**#include <iostream>**

**using namespace std;**

**int main(){**

**setlocale(LC\_ALL, "Russian");**

**char s1[] = "Happy New Year";**

**char\* s2 = (char\*)"Happy New Year";**

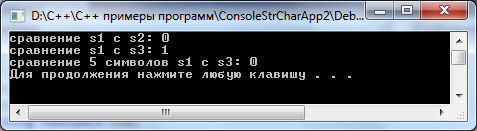
**char s3[] = "Happy Holidays";**

**cout << "сравнение s1 с s2: " << strcmp(s1, s2)<< endl;**

**cout << "сравнение s1 с s3: " << strcmp(s1, s3)<< endl;**

**cout << "сравнение 5 символов s1 с s3: " << strncmp(s1, s3, 5) << endl;**

**return 0;**

**}**

**Пример 4**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**setlocale(LC\_ALL, "Russian");**

**char s1[20] = "Happy ";**

**char s2[] = "New Year ";**

**char s3[40] = "";**

**strcat\_s(s1, s2); // функция strcat\_s дописывает в конец ‘\0’**

**cout << "добавление s2 в s1: " << s1<< endl;**

**strncat\_s(s3, s1, 6);**

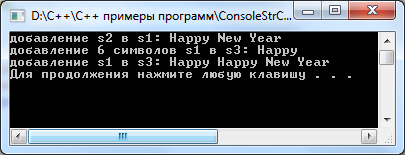
**cout << "добавление 6 символов s1 в s3: " << s3 << endl;**

**strcat\_s(s3, s1);**

**cout << "добавление s1 в s3: " << s3<< endl;**

**return 0;**

**}**

****

**Пример 5**

**#include <iostream>**

**using namespace std;**

**void findS1InS2(char[], char[], char[]);**

**int main()**

**{**

**char string1[] = "1 2 34 5 8 3 9"; //исходная строка**

**char string2[] = "1 2 3 58 5 4 1"; //исходная строка**

**char string3[80] = ""; //результирующая строка**

**findS1InS2(string1, string2, string3);**

**cout << string3<< endl;**

**return 0;**

**}**

**void findS1InS2(char s1[], char s2[], char s3[]){**

**char p3[80]; //копия строки s2**

**s3[0] = NULL; //'\0'//ОБЯЗАТЕЛЬНО ОБНУЛИТЬ!!!**

**char\* p1, \*pp1; //указатели на начало текущего слова и остаток s1**

**char\* p2, \*pp2; //указатели на начало текущего слова и остаток s2**

**p1 = strtok\_s(s1, " ", &pp1);**

**//strtok\_s - функция возвращает указатель на первый**

**//символ из последовательности символов s1 до**

**//до указанного ограничителя " " - пробела**

**//в pp1 записывается указатель на символ s1 после**

**//ограничителя-хвост предложения после текущего слова**

**while(p1 != NULL){**

**strcpy\_s(p3, s2);**

**p2 = strtok\_s(p3, " ", &pp2);**

**while (p2 != NULL) {**

**if (!strcmp(p1, p2)) {**

**strcat\_s(s3, 80, p1);//добавление в конец s3 символов p1**

**strcat\_s(s3, 80, " ");**

**break;**

**}**

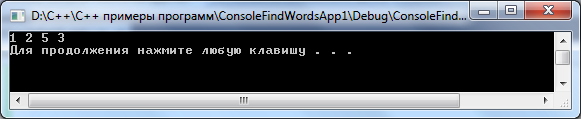
**p2 = strtok\_s(NULL, " ", &pp2);**

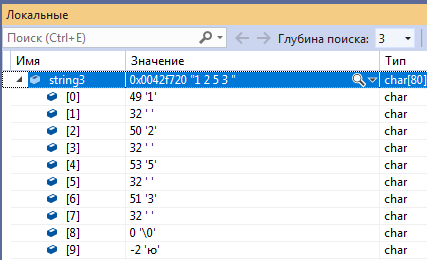
**}**

**p1 = strtok\_s(NULL, " ", &pp1);**

**}**

**}**





**Пример 6**

**#include <iostream>**

**using namespace std;**

**void convertIngToEd(char \*, char (\*)[80]);**

**int main() {**

**char str[80]; //исходная строка**

**char str1[80] = ""; //результирующая строка**

**cin.getline(str, 80); //ввод строки, 80 - ограничитель числа символов**

**//в строке, если использовать cin>>str, то**

**//введутся только символы до первого пробела**

**convertIngToEd(str, &str1);**

**cout << str1 << endl;**

**return 0;**

**}**

**void convertIngToEd(char \*str, char (\*s)[80]) {**

**char ss[80]; //слово без окончания ing**

**char str1[80]=""; //ОБЯЗАТЕЛЬНО ОБНУЛИТЬ!!!**

**char\* p; //указатель на начало текущего слова предложения**

**char\* pp; //указатель на остаток предложения**

**//strtok\_s - функция возвращает указатель на первый**

**//символ из последовательности символов str до до**

**//указанного ограничителя " "- пробела**

**//в pp записывается указатель на символ str после**

**//ограничителя-хвост предложения после текущего слова**

**for (p = strtok\_s(str, " ", &pp); p; p = strtok\_s(NULL, " ", &pp)) {**

**cout << p << " + " << pp << endl; //для отладки**

**if ((strlen(p) >= 3) && (strncmp(&p[strlen(p) - 3], "ing", 3) == 0))**

**//strlen -возвращает количество символов, начиная с символа**

**//по указателю р и до ограничителя**

**//здесь - если длина слова p >= 3**

**//strncmp - сравнение n символов двух символьных**

**//последовательностей. Здесь - если три последних символа р**

**//равны "ing"**

**//&p[strlen(p)-3]-адрес последовательности из 3 последних**

**//символов р**

**{**

**strncpy\_s(ss, p, strlen(p) - 3); //strncpy\_s – копирование**

**//strlen(p) -3 символов р в строку ss. Здесь копирование в**

**//ss символов р без 3 последних символов**

**strcat\_s(ss, "ed "); //добавление в конец ss символов "ed "**

**strcat\_s(str1, ss); //добавление в конец str1 символов ss**

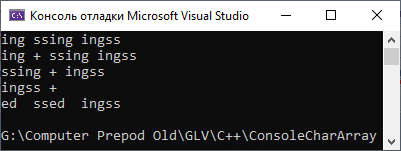
**}**

**else**

**strcat\_s(str1, p); strcat\_s(str1, " "); //если слово меньше**

**//3 символов или в окончании слова не "ing", то в строку**

**//str1 добавляются символы р без изменения**

** }**

**strcpy\_s(\*s, str1);**

**}**

* 1. Список функций

|  |  |
| --- | --- |
| **Функция** | **Пояснение** |
| [strlen](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strlen/)**(**имя\_строки**)** | определяет длину указанной строки, без учёта нуль-символа |
| **Копирование строк** | |
| [strcpy](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcpy/)**(**s1,s2**)** | выполняет побайтное копирование символов из строки  s2 в строку s1 |
| [strncpy](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncpy/)**(**s1,s2, n**)** | выполняет побайтное копирование n символов из строки  s2 в строку s1. возвращает значения s1 |
| **Конкатенация строк** | |
| [strcat(s1,s2)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcat/) | объединяет строку s2 со строкой s1. Результат сохраняется в s1 |
| [strncat(s1,s2,n)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncat/) | объединяет n символов строки s2 со строкой s1. Результат сохраняется в s1 |
| **Сравнение строк** | |
| [strcmp(s1,s2)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcmp/) | сравнивает строку s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 С учётом регистра |
| [strncmp(s1,s2,n)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncmp/) | сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 С учётом регистра |
| **stricmp(**s1,s2**)** | сравнивает строку s1 со строкой s2 и возвращает результат типа **int**: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 Без учёта регистра |
| **strnicmp(**s1,s2,n**)** | сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0 — если s1>s2 Без учёта регистра |
| **Обработка символов** | |
| [isalnum(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isalnum/) | возвращает значение true, если с является буквой или цифрой, и false в других случаях |
| [isalpha(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isalpha/) | возвращает значение true, если с является буквой,  и false в других случаях |
| [isdigit(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isdigit/) | возвращает значение true, если с является цифрой, и false в других случаях |
| [islower(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-islower/) | возвращает значение true, если с является буквой нижнего регистра, и false в других случаях |
| [isupper(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isupper/) | возвращает значение true, если с является буквой верхнего регистра, и false в других случаях |
| [isspace(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isspace/) | возвращает значение true, если с является пробелом, и false в других случаях |
| [toupper(c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-toupper/) | если символ с, является символом нижнего регистра, то функция возвращает преобразованный символ с в верхнем регистре, иначе символ возвращается без изменений. |
| **Функции поиска** | |
| [strchr(s,c)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strchr/) | поиск первого вхождения символа св строке s**.**В случае удачного поиска возвращает указатель на место первого вхождения символа с**.**Если символ не найден, то возвращается ноль. |
| [strcspn(s1,s2)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcspn/) | определяет длину начального сегмента строки s1, содержащего те символы, которые не входят в строку s2 |
| [strspn(s1,s2)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strspn/) | возвращает длину начального сегмента строки s1, содержащего только те символы, которые входят в строку s2 |
| **strprbk(**s1,s2**)**  <http://cppstudio.com/post/732/> | Возвращает указатель  первого вхождения любого символа строки s2 в строке s1 |
| **Функции преобразования** | |
| [atof(s1)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atof/) | преобразует строку s1 в тип double |
| [atoi(s1)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atoi/) | преобразует строку s1 в тип int |
| [atol(s1)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atol/) | преобразует строку s1 в тип long int |
| **Функции стандартной библиотеки ввода/вывода <stdio>** | |
| [getchar(с)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdio-stdio-h/funkciya-getchar/) | считывает символ ссо стандартного потока ввода, возвращает символ в формате int |
| [gets(s)](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdio-stdio-h/funkciya-gets/) | считывает поток символов со стандартного устройства ввода в строку sдо тех пор, пока не будет нажата клавиша ENTER |

**Пример 7**

Поиск гласных букв

**#include <iostream>**

**int main()**

**{**

**char str[] = "Police Academy";**

**char key[] = "aeiou";**

**std::cout << "Finding vowels in a string" << str << std::endl;**

**char\* pch = strpbrk(str, key); // первый поиск**

**while (pch != NULL) // пока есть гласные буквы в строке**

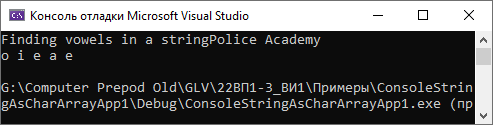
**{**

**std::cout << \*pch << " "; // печать гласного символа**

**pch = strpbrk(pch + 1, key); // поиск гласных букв**

**}**

**std::cout << "\n";**

** return 0;**

**}**

* 1. Массивы строк

**Пример 8**

**#include <iostream>**

**using namespace std;**

**int main() {**

**const int max\_length{ 50 }; // максимальная длина строки**

**//включая нулевой байт \0**

**char langs[][max\_length] {**

**"C++", "C#", "Python", "Java",**

**"Kotlin", "Go", "Dart", "PHP"**

**};**

**cout << langs[0] << endl; // C++**

**cout << langs[1] << endl; // C#**

**cout << langs[2] << endl; // Python**

**for (auto lang : langs) {**

**std::cout << lang << std::endl;**

**}**

**}**

**Пример 9**

Рассмотрим пример, в котором названия дней недели помещаются в массив:

**#include <iostream>**

**using namespace std;**

**int main() {**

**setlocale(LC\_ALL, "Russian");**

**const int DAYS = 7; // количество строк в массиве**

**const int MAX = 12; // максимальная длина каждой из них**

**// массив строк**

**char star[DAYS][MAX] = {**

**"Понедельник", "Вторник", "Среда", "Четверг",**

**"Пятница", "Суббота", "Воскресенье"**

**};**

**// вывод всех строк на экран**

**for (int j = 0; j < DAYS; j++)**

**cout << star[j] << endl;**

**return 0;**

**}**

Программа печатает все строки массива:

**Понедельник**

**Вторник**

**Среда**

**Четверг**

**Пятница**

**Суббота**

**Пример 10**

**#include <iostream>**

**using namespace std;**

**void findS1InS2(char[], char[], char[][80], int&);**

**int main()**

**{**

**char string1[] = "1 2 34 5 8 3 9"; //исходная строка**

**char string2[] = "1 2 3 58 5 4 1"; //исходная строка**

**char string3[20][80]; //результат – массив строк**

**int k = -1;**

**findS1InS2(string1, string2, string3, k);**

**for(int i = 0; i < k+1; ++i)**

**cout << string3[i] << endl;**

**return 0;**

**}**

**void findS1InS2(char s1[], char s2[], char s3[][80], int& n) {**

**char p3[80];**

**char\* p1, \* pp1; //указатели на начало текущего слова и остаток s1**

**char\* p2, \* pp2; //указатели на начало текущего слова и остаток s2**

**p1 = strtok\_s(s1, " ", &pp1);**

**while (p1 != NULL) {**

**strcpy\_s(p3, s2);**

**p2 = strtok\_s(p3, " ", &pp2);**

**while (p2 != NULL) {**

**if (!strcmp(p1, p2)) {**

**strcpy\_s(s3[++n], p1);//добавление в s3 символов p1**

**break;**

**}**

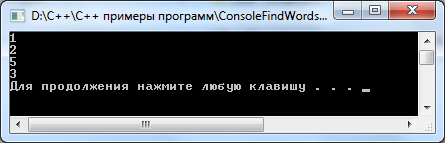
**p2 = strtok\_s(NULL, " ", &pp2);**

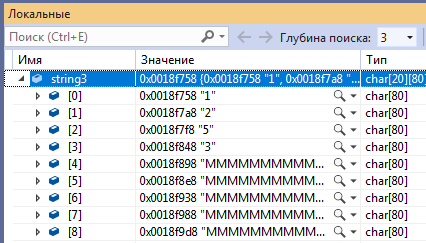
**}**

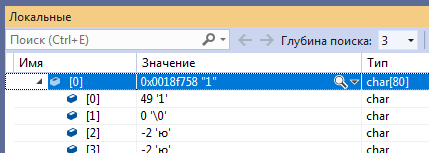
**p1 = strtok\_s(NULL, " ", &pp1);**

**}**

**}**



****

****