Копирование строк

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | // str\_cpy.cpp: определяет точку входа для консольного приложения.    #include "stdafx.h"  #include <iostream>  using namespace std;    int main(int argc, char\* argv[])  {      char s2[27] = "Counter-Strike 1.6 forever";          // инициализация строки s2      char s1[27];                                         // резервируем строку для функции strcpy()      cout << "strcpy(s1,s2) = " << strcpy(s1,s2) << endl; // содержимое строки s2 скопировалось в строку s1, возвращается указатель на s1      cout << "s1=             " << s1            << endl; // вывод содержимого строки s1      char s3[7];                                          // резервируем строку для следующей функции      cout << strncpy(s3, s2, 7) << endl;      // копируем 7 символов из строки s2 в строку s3      system("pause");      return 0;  } |

В **строках 9, 10**создаём строковые массивы на 27 символов, словосочетание "Counter-Strike 1.6 forever" содержит 26 символов, последнее место в массиве займет нуль-символ. В **строке 11** функция strcpy(s1,s2) копирует  значение строки  s2 в строку s1 и возвращает указатель на строку s1. Если строка s1 будет меньше строки s2, то скопируется то количество символов, которое вместится в  строку s2**. Строка 10**всего лишь показывает, что в строке s1содержится скопированное значение. В **строке 14**функция strncpy(s3, s2, sizeof(s3)) выполняет копирование 7 символов строки s2в строку s3 (см. Рисунок 2).

CppStudio.com

strcpy(s1,s2) = Counter-Strike 1.6 forever

s1= Counter-Strike 1.6 forever

CounterCounter-Strike 1.6 forever

Для продолжения нажмите любую клавишу . . .

Рисунок 2 — Символы и строки

**Конкатенация строк**

Использование функций strcat() и strncat(), для объединения строк, то есть для их конкатенации.

* [MVS](http://cppstudio.com/post/437/#tab42022114)
* [Code::Blocks](http://cppstudio.com/post/437/#tab42891274)
* [Dev-C++](http://cppstudio.com/post/437/#tab62982295)
* [QtCreator](http://cppstudio.com/post/437/#tab44676282)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | // str\_cat.cpp: определяет точку входа для консольного приложения.    #include "stdafx.h"  #include <iostream>  using namespace std;    int main(int argc, char\* argv[])  {      char s1[30] = "I am ";      char s2[] = "programmer on the C++!!!!";      cout << strcat(s1,s2) << endl;               // объединяем строки s1 и s2, результат сохраняется в строке s1      char s3[23] = "I am a good ";      cout << strncat(s3,s2,10) <<  "!!!" << endl; // объединяем 10 символов строки s2 со строкой s3      system("pause");      return 0;  } |

В **строке 11**функция strcat(s1,s2)объединяет строки s1и s2, результат сохраняется в строке s1. По этому при объявлении строки s1 её размер установлен на 30 символов. В **строке 13**функция**strncat(s3,s2,10)**объединяет 10 символов из строки s2(как раз помещается слово programmer) со строкой s3, результат сохраняется в строке s3. И по этому размер строки s3также задан фиксировано (см. Рисунок 3).

CppStudio.com

I am programmer on the C++!!!!

I am a good programmer!!!

Для продолжения нажмите любую клавишу . . .

Рисунок 3 — Символы и строки

Сравнение строк

Рассмотрим работу функции strcmp(), остальные функции используются аналогично, так что каждую рассматривать не будем.

* [MVS](http://cppstudio.com/post/437/#tab5157094)
* [Code::Blocks](http://cppstudio.com/post/437/#tab57335498)
* [Dev-C++](http://cppstudio.com/post/437/#tab41445695)
* [QtCreator](http://cppstudio.com/post/437/#tab97485783)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | // str\_cmp.cpp: определяет точку входа для консольного приложения.    #include "stdafx.h"  #include <iostream>  #include <iomanip>  using namespace std;    int main(int argc, char\* argv[])  {      char s1[] = "www.cppstudio.com";      char s2[] = "http://www.cppstudio.com";      cout << " s1 == s1 -> " << setw(2) << strcmp(s1,s1) << endl; // строка s1 = s1      cout << " s1 <  s2 -> " << setw(2) << strcmp(s1,s2) << endl; // строка s1 < s2      cout << " s2 >  s1 -> " << setw(2) << strcmp(s2,s1) << endl; // строка s2 > s1      system("pause");      return 0;  } |

В**строках 12, 13, 14**функция strcmp()выполняет сравнение строк, в различном порядке, таким образом были рассмотрены все возможные варианты работы данной функции. В первом случае строку s1сравнивали с самой собой соответственно и результат равен 0. Во втором случае строка s1 оказалась меньше строки s2поэтому результат равен **1**. В третьем случае строка s1по-прежнему меньше строки s2, но мы поменяли эти строки местами, таким образом, порядок параметров функции изменился и результат стал равен **-1** (см. Рисунок 4).

CppStudio.com

s1 == s1 -> 0

s1 < s2 -> 1

s2 > s1 -> -1

Для продолжения нажмите любую клавишу . . .

Рисунок 4 — Символы и строки

Обработка символов

Функции из данной группы умеют различать, к какому типу знаков относятся те или иные символы, например буквы, цифры, специальные знаки.

* [MVS](http://cppstudio.com/post/437/#tab81828863)
* [Code::Blocks](http://cppstudio.com/post/437/#tab31148748)
* [Dev-C++](http://cppstudio.com/post/437/#tab40033413)
* [QtCreator](http://cppstudio.com/post/437/#tab5949778)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | // issss.cpp: определяет точку входа для консольного приложения.    #include "stdafx.h"  #include <iostream>  using namespace std;    int main(int argc, char\* argv[])  {      char symbol    = 'd'; // буква      char digit     = '9'; // цифра      char space     = ' '; // пробел      char character = '#'; // знак      // функция isalnum() проверяет является ли её аргумент буквой или цифрой      cout << symbol    << " - it is digit or alpha?: "; isalnum(symbol)    ? cout << "true\n": cout << "false\n";      // функция isalpha() проверяет является ли её аргумент буквой      cout << symbol    << " - it is alpha?:          "; isalpha(symbol)    ? cout << "true\n": cout << "false\n";      // функция isdigit() проверяет является ли её аргумент цифрой      cout << digit     << " - it is digit?:          "; isdigit(digit)     ? cout << "true\n": cout << "false\n";      // функция isspace() проверяет является ли её аргумент пробелом      cout << space     << " - it is space?:          "; isspace(space)     ? cout << "true\n": cout << "false\n";      // функция islower() проверяет является ли её аргумент буквой нижнего регистра      cout << symbol    << " - it is lower alpha?:    "; islower(symbol)    ? cout << "true\n": cout << "false\n";      // функция isupper() проверяет является ли её аргумент буквой верхнего регистра      cout << symbol    << " - it is upper alpha?:    "; isupper(symbol) ? cout << "true\n": cout << "false\n";      system("pause");      return 0;  } |

В данной программе по использованию функций вопросов возникать не должно. Функции сами по себе возвращают целочисленное значение, положительное – true, отрицательное или  ноль – false. В формировании результата работы функций участвовал тернарный оператор, то есть выводилось сообщение true или false без всяких числовых значений (см. Рисунок 5).

CppStudio.com

d - it is digit or alpha?: true

d - it is alpha?: true

9 - it is digit?: true

- it is space?: true

d - it is lower alpha?: true

d - it is upper alpha?: false

Для продолжения нажмите любую клавишу . . .

Рисунок 5 — Символы и строки

Итак, мы рассмотрели больше половины функций, из тех, что есть в таблице. Остальные функции используются аналогично, так что по ним примеры программ разбирать не будем.

[Замена символов в строке](https://ru.stackoverflow.com/questions/435385/%d0%97%d0%b0%d0%bc%d0%b5%d0%bd%d0%b0-%d1%81%d0%b8%d0%bc%d0%b2%d0%be%d0%bb%d0%be%d0%b2-%d0%b2-%d1%81%d1%82%d1%80%d0%be%d0%ba%d0%b5)

[Задать вопрос](https://ru.stackoverflow.com/questions/ask)

Вопрос задан 4 года 8 месяцев назад

Последняя активность [2 года назад](https://ru.stackoverflow.com/questions/435385/%d0%97%d0%b0%d0%bc%d0%b5%d0%bd%d0%b0-%d1%81%d0%b8%d0%bc%d0%b2%d0%be%d0%bb%d0%be%d0%b2-%d0%b2-%d1%81%d1%82%d1%80%d0%be%d0%ba%d0%b5?lastactivity)

Просмотрен 17k раз

4

У меня есть строка: "''Start''". После всех действий над ней, должно получиться: "Start". Т.е удаляются все символы ", а затем все пары символов '' заменяются на ".  
Удаляю из нее все символы ":  
result.erase(std::remove(result.begin(), result.end(), '"' ), result.end());

Затем нужно заменить все пары символов '' на ":  
replace(result.begin(), result.end(), "\'\'", "\"");

Выдает ошибку следующую ошибку:

tools/data\_parser.cpp: In function ‘int main(int, const char\*\*)’:

tools/data\_parser.cpp:21:52: error: no matching function for call to ‘replace(std::\_\_cxx11::basic\_string<char>::iterator, std::\_\_cxx11::basic\_string<char>::iterator, const char [3], const char [2])’

replace(result.begin(), result.end(), "\'\'", "\"");

^

In file included from /usr/include/c++/5/algorithm:62:0,

from tools/data\_parser.cpp:7:

/usr/include/c++/5/bits/stl\_algo.h:4233:5: note: candidate: template<class \_FIter, class \_Tp> void std::replace(\_FIter, \_FIter, const \_Tp&, const \_Tp&)

replace(\_ForwardIterator \_\_first, \_ForwardIterator \_\_last,

^

/usr/include/c++/5/bits/stl\_algo.h:4233:5: note: template argument deduction/substitution failed:

tools/data\_parser.cpp:21:52: note: deduced conflicting types for parameter ‘const \_Tp’ (‘char [3]’ and ‘char [2]’)

replace(result.begin(), result.end(), "\'\'", "\"");

Я так понимаю, что std::replace заменяет только один символ на другой. Как заменить пару символов? Компилирую так: g++-5 -std=c++11 cfg\_parser.cpp -o cfg\_parser

Просто, быстро и (надеюсь) надежно

#include <stdio.h>

#include <stdlib.h>

char \*

replq (char \*str)

{

int i, j;

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

if (str[i] != '"') {

if (str[i] == '\'' && str[i + 1] == '\'') {

str[j++] = '"';

i++;

} else

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

}

str[j] = 0;

return str;

}

int

main (int ac, char \*av[])

{

char str[1000];

while (fgets(str, 1000, stdin))

puts(replq(str));

return puts("End") == EOF;

}

**КАК ПРАВИЛЬНО ЗАМЕНИТЬ СИМВОЛ В СТРОКЕ C ++?**

#include <iostream>

#include <string>

#include <cctype>

using namespace std;

int main() {

string passCode;

passCode = "1 ";

int i;

for(i =0; i < passCode.length();i++){

if(isspace(passCode.at(i)) == true){

passCode.replace(i,1,"\_");

}

}

cout << passCode << endl;

return 0;

}

Код выше, мои указания: [Заменить любой пробел » на ‘\_’ в 2-символьной строке passCode. Если места не существует, программа не должна ничего печатать.]

с моим кодом в настоящее время так оно и есть, он выводит «1». Когда я запускаю его с проверкой состояния на ложь вместо истины, он печатает «\_». Я не понимаю, почему он это делает, кто-нибудь видит проблему, которую я не вижу?  
Мне не разрешено использовать алгоритм. заголовок. Мне также разрешено работать только внутри main, без функций или импортированных заголовков / классов.

**2**

Как найти и заменить все символы в строке конкретными символами C++

Я новичок в программировании, поэтому не волнуйтесь, если я подошел к проблеме неправильно. я делаю это как задание. Моя цель состоит в том, чтобы взять строку от пользователя и заменить все символы другим символом. Код ниже должен найти все As и заменить их на \*s. мой код показывает совершенно неожиданный результат. Также какова цель \_deciphered.length().

например: "I Am A bAd boy" должно превратиться в "I \*m \* b\*d boy"

затем я должен реализовать его для всех прописных и строчных букв и цифр и заменить различными символами и наоборот, чтобы сделать небольшую программу кодирования-декодирования

#include <iostream>

#include <string>

using namespace std;

string cipher (string);

void main ()

{

string ciphered, deciphered;

ciphered="String Empty";

deciphered="String Empty";

cout<<"Enter a string to \"Encode\" it : ";

cin>>deciphered;

ciphered=cipher (deciphered);

cout<<endl<<endl;

cout<<deciphered;

}

string cipher (string \_deciphered)

{

string \_ciphered=(\_deciphered.replace(\_deciphered.find("A"), \_deciphered.length(), "\*"));

return \_ciphered;

}

[c++](https://coderoad.ru/list/?page=1&sort=view&tag=c%2B%2B) [replace](https://coderoad.ru/list/?page=1&sort=view&tag=replace)

 Поделиться[Источник](https://stackoverflow.com/questions/19546367)**Usama Khurshid**   *23 октября 2013 в 15:46*

3 Ответа

4

Поскольку вы, кажется, уже используете стандартную библиотеку,

#include <algorithm> // for std::replace

std::replace(\_deciphered.begin(), \_deciphered.end(), 'A', '\*');

Если вам нужно сделать это вручную, то имейте в виду , что std::string выглядит как контейнер char, поэтому вы можете перебирать его содержимое , проверять, является ли каждый элемент 'A', и если да, установите его в '\*' .

Рабочий пример:

#include <iostream>

#include <string>

#include <algorithm>

int main()

{

std::string s = "FooBarro";

std::cout << s << std::endl;

std::replace(s.begin(), s.end(), 'o', '\*');

std::cout << s << std::endl;

}

Выход:

FooBarro

F\*\*Barr\*

то я должен реализовать его для всех столичных и маленьких букв и цифр и заменить их разными символами и наоборот сделать небольшую программу Encode-Decode

#include <iostream>

#include <string>

using namespace std;

string cipher (string);

void main ()

{

string ciphered, deciphered;

ciphered="String Empty";

deciphered="String Empty";

cout<<"Enter a string to \"Encode\" it : ";

cin>>deciphered;

ciphered=cipher (deciphered);

cout<<endl<<endl;

cout<<deciphered;

}

string cipher (string \_deciphered)

{

string \_ciphered=(\_deciphered.replace(\_deciphered.find("A"), \_deciphered.length(), "\*"));

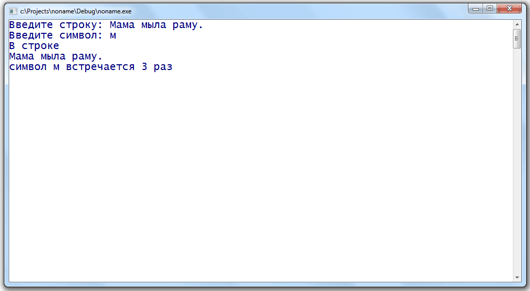
return \_ciphered;

}

***Пример*** Посчитать количество введенных символов во введенной строке.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
int main() {  
  char s[80], sym;  
  int count, i;  
  system("chcp 1251");  
  system("cls");  
  printf("Введите строку : ");  
  gets\_s(s);  
  printf("Введите символ : ");  
  sym = getchar();  
  count = 0;  
  for (i = 0; s[i] != '\0'; i++)  
  {  
    if (s[i] == sym)  
      count++;  
  }  
  printf("В строке\n");  
  puts(s);      // Вывод строки  
  printf("символ ");  
  putchar(sym); // Вывод символа  
  printf(" встречается %d раз", count);  
  getchar(); getchar();  
  return 0;  
}

Результат выполнения  


**Основные функции стандартной библиотеки string.h**

***Пример*** использования функций

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

#define \_CRT\_SECURE\_NO\_WARNINGS  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
int main() {  
  char m1[80] = "Первая строка";  
  char m2[80] = "Вторая строка";  
  char m3[80];  
  system("chcp 1251");  
  system("cls");  
  strncpy(m3, m1, 6);  // не добавляет '\0' в конце строки  
  puts("Результат strncpy(m3, m1, 6)");  
  puts(m3);  
  strcpy(m3, m1);  
  puts("Результат strcpy(m3, m1)");  
  puts(m3);  
  puts("Результат strcmp(m3, m1) равен");  
  printf("%d", strcmp(m3, m1));  
  strncat(m3, m2, 5);  
  puts("Результат strncat(m3, m2, 5)");  
  puts(m3);  
  strcat(m3, m2);  
  puts("Результат strcat(m3, m2)");  
  puts(m3);  
  puts("Количество символов в строке m1 равно  strlen(m1) : ");  
  printf("%d\n", strlen(m1));  
  \_strnset(m3, 'f', 7);  
  puts("Результат strnset(m3, 'f', 7)");  
  puts(m3);  
  \_strset(m3, 'k');  
  puts("Результат strnset(m3, 'k')");  
  puts(m3);  
  getchar();  
  return 0;  
}

Результат выполнения  
