

Рядки в C++

Оголошення рядка як масив символів

```
char имя_переменной [размер_строки];
char str[20];
char string[10] = "abcdefghf"
```

Оголошення рядка як об'єкт класу string

```
string имя_переменной;
string str;
string str = "Hello, world!";
```

Таблиця 1. Функції для роботи з текстовими рядками

| Функция | Заголовок | Описание |
|---------------|-----------|------------------------------------------------------------------------------|
| strcpy(s1,s2) | <cstring> | Копирование строки s2 в строку s1 |
| strcat(s1,s2) | <cstring> | Строка s2 присоединяется к строке s1 |
| strcmp(s1,s2) | <cstring> | Сравнение строк s1 и s2: если строки равны, возвращается значение 0 |
| strchr(s,ch) | <cstring> | Указатель на первую позицию символа ch в строке s |
| strstr(s1,s2) | <cstring> | Указатель на первую позицию подстроки s2 в строке s1 |
| atoi(s) | <cstdlib> | Преобразование состоящей из цифр строки s в целое число типа int |
| atol(s) | <cstdlib> | Преобразование состоящей из цифр строки s в целое число типа long |
| atof(s) | <cstdlib> | Преобразование состоящей из цифр строки s в действительное число типа double |
| tolower(ch) | <cctype> | Преобразование буквенного символа ch к строчному формату |
| toupper(ch) | <cctype> | Преобразование буквенного символа ch к прописному формату |

strlen()
size()

Чтобы получить размер строки (количества символов), хранимой строковым объектом (класса `string`) используйте функцию `size()`. Напишем программу, подсчитывающую сколько раз искомый символ встречается в заранее определенной строке. Программа в цикле "проходится" по строке и подсчитывает частоту символа, то есть считает, сколько раз в строке встречается искомый символ.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string str="Hello, world!";
6      char findCh='l';
7      int count = 0;
8      for (int i=0; i< str.size();i++)
9      {
10         if (str[i]==findCh)
11         {
12             ++count;
13         }
14     }
15     cout << "Символ " << findCh << " встречается " <<
16     count << " раз\n";
17     return 0;
18 }
```

Символ l встречается 3 раз

В языке C используется несколько иной стиль строки. Типа `string` нет и строка представляет собой массив символов, то есть элементов типа `char`. Все строки в C заканчиваются знаком `\0`, означающим конец строки.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      char c[] = "Hello, world!";
6      char findCh='l';
7      int count = 0;
8      for (int i=0; i< c[i]!='\0';++i)
9      {
10         if (c[i]==findCh)
11         {
12             ++count;
13         }
14     }
15     cout << "Частота " << findCh << " = " << count << " \n";
16     return 0;
17 }
```

```
Частота l = 3
```

```
Частота o = 2
```

Подсчет количества цифр и пробелов

Задан текст и нужно вычислить: общее количество символов; количество пробелов в тексте; количество цифр в тексте.

```
1 // Стастика про рядок
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     char line[150];
7     int total, digits, spaces;
8     total=digits=spaces=0;
9     cout << "Введите строку:";
10    cin.getline (line, 150);
11    for (int i=0; line[i]!='\0';++i)
12    {
13        ++total;
14        if((line[i]>='0') && (line[i]<='9'))
15        {
16            ++digits;
17        }
18        else if(line[i]==' ')
19        {
20            ++spaces;
21        }
22    }
23    cout << "Цифры: " << digits << endl;
24    cout << "Пробели: " << spaces << endl;
25    cout << "Всего: " << total << endl;
26    return 0;
27 }
28 }
```

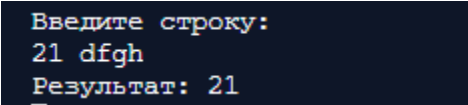
```
1 // Стастика про рядок
2 #include <iostream>
3 #include <cstring>
4 using namespace std;
5
6 int main() {
7     char line[150];
8     int total, digits, spaces;
9     total=digits=spaces=0;
10    cout << "Введите строку:\n";
11    cin.getline (line, 150);
12    //for (int i=0; line[i]!='\0';++i)
13    for (int i=0; i<strlen(line);++i)
14    {
15        ++total;
16        if((line[i]>='0') && (line[i]<='9'))
17        {
18            ++digits;
19        }
20        else if(line[i]==' ')
21        {
22            ++spaces;
23        }
24    }
25    cout << "Цифры: " << digits << endl;
26    cout << "Пробели: " << spaces << endl;
27    cout << "Всего: " << total << endl;
28    return 0;
29 }
```

Введите строку:2020 Good luck
Цифры: 4
Пробели: 2
Всего: 14

Введите строку:
Test 0510 2019 test
Цифры: 8
Пробели: 3
Всего: 19

Удаляем все символы в строке, кроме цифр

```
1  // Удаляемо символы крім цифр
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main() {
7      string line;
8      int total, digits, spaces;
9      total=digits=spaces=0;
10     cout << "Введите строку:\n";
11     getline (cin, line);
12     for (int i=0; i<line.size();++i)
13     {
14         if (((line[i]>='a' && line[i]<='z') || (line[i]>='A' && line
15             [i]<='Z'))))
16         {
17             line[i]='\0';
18         }
19     }
20     cout << "Результат: " << line << endl;
21     return 0;
22 }
```

A screenshot of a terminal window showing the program's execution. The prompt 'Введите строку:' is followed by the input '21 dfgH'. The output line shows 'Результат: 21', where the 'H' has been removed and the string is left-aligned.

Введите строку:
21 dfgH
Результат: 21

В цикле мы проверяем, является ли символ буква. Если да, то все символы после него, включая нулевой символ, смещаются на 1 позицию влево.

Удаляем все символы в строке, кроме цифр

```
1  // Видаляємо символи крім цифр
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main()
7  {
8      string line;
9      int total, digits, spaces;
10     total=digits=spaces=0;
11     cout << "Введіть строку:\n";
12     getline (cin, line);
13     for (int i=0; i<line.size();++i)
14     {
15         if (!(line[i]>='0' && line[i]<='9'))
16         {
17             line[i]='\0';
18         }
19     }
20     cout << "Результат: " << line << endl;
21     return 0;
22 }
```

```
Введіть строку:
gdgh 121 ////tryt....5454
Результат: 1215454
```

Определение длины строки

Для определения длины строки используется функция `strlen()` либо метод `size()`

```
string str = "Hello, world!";           char str[] = "Hello, world!";
cout << "String Length = " << str.size();  cout << "String Length = " << strlen(str);
// или метод str.length()
```

Функция `getline()` принимает два параметра - поток ввода (`cin` - стандартный ввод) и название переменной, в которую будет записана строка. Если использовать оператор `>>`, то строка будет введена до первого пробельного символа. Функция `getline()` читает строку со всеми пробельными символами.

```
1  // Кількість символів у рядку
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main() {
7      string s;
8      int i;
9      cout << "Введіть строку:\n";
10     getline (cin, s);
11     for (i=0; s[i] != '\0'; ++i);
12
13     cout << "Довжина: " << i << endl;
14     return 0;
15 }
16
```

```
Введіть строку:
hello world
Довжина: 11
```

Объединение нескольких строк в одну

Для конкатенации (объединения) двух строк обычно используется функция `strcat()`.

```
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main() {
7      string s1, s2, result;
8
9      cout << "Введите строку s1:\n";
10     getline (cin, s1);
11     cout << "Введите строку s2:\n";
12     getline (cin, s2);
13     result=s1+s2;
14     cout << "Результат: " << result<< endl;
15     return 0;
16 }
```

```
Введите строку s1:
Будь здоровый
Введите строку s2:
сказав Петро
Результат: Будь здоровыйсказав Петро
```

```
1  // Конкатенация
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main() {
7      char s1[50], s2[50], result[100];
8
9      cout << "Введите строку s1:\n";
10     cin.getline (s1, 50);
11     cout << "Введите строку s2:\n";
12     cin.getline (s2, 50);
13     strcat(s1, s2);
14     cout << "Результат: " << s1<< endl;
15     cout << "Результат: " << s2 << endl;
16     return 0;
17 }
```

```
Введите строку s1:
6546
Введите строку s2:
213
Результат: 6546213
Результат: 213
```


Копирование двух строк

При использовании объекта типа `string` для копирования строк можно использовать оператор присваивания `=`. При использовании строк в стиле С лучшим решением является использование функции `strcpy`.

```
1  // Копіювання рядків
2  #include <iostream>
3  #include <cstring>
4  using namespace std;
5
6  int main() {
7      string s1, s2;
8
9      cout << "Введіть строку s1:\n";
10     getline (cin, s1);
11     s2=s1;
12
13     cout << "s1 " << s1<< endl;
14     cout << "s2 " << s2 << endl;
15     return 0;
16 }
```

```
Введіть строку s1:
World suite
s1 World suite
s2 World suite
```

Операторы сравнения строк

```
1 // сортування
2 #include <iostream>
3 #include <cstring>
4 using namespace std;
5
6 int main() {
7     string str[10], tmp;
8
9     cout << "Введіть 10 слів:\n";
10    for (int i=0; i<10; ++i)
11    {
12        getline (cin, str[i]);
13    }
14    for (int i=0; i<9; ++i)
15        for (int j=i+1; j < 10; ++j)
16        {
17            if (str[i]>str[j])
18            {
19                tmp=str[i];
20                str[i]=str[j];
21                str[j]=tmp;
22            }
23        }
```

```
24
25     cout << "Сортируем: " << endl;
26     for (int i=0; i<10; ++i)
27     {
28         cout << str[i] << endl;
29     }
30
31     return 0;
32 }
```

```
Введіть 10 слів:
work
job
cat
dog
milk
mouse
teacher
policmen
bus
peace
Сортируем:
bus
cat
dog
job
milk
mouse
peace
policmen
teacher
work
```

Для сравнения строк используется функцию `strcmp()`, а функция `strcpy()` используется для копирования строки в переменную `temp` в процессе сортировки. Будем работать со строками, как массивами символов. Алгоритм прост: во время перебора двухмерного массива строк `str` сравниваем две строки. Если первая строка больше (лексикографически), чем вторая, то функция `strcmp()` возвращает 1. При этом мы меняем местами эти две строки и так до тех пор, пока строки не будут расположены в лексикографическом порядке.

Функція `strcmp ()` починає порівнювати по одному символу і як тільки будуть знайдені перші неоднакові символи, функція проаналізує числові коди цих символів. Чий код виявиться більше, той рядок і вважатиметься більшим.

```
1  //пример использования функции strcmp
2
3  #include <iostream>
4  #include <cstring>
5  using namespace std;
6
7  int main ()
8  {
9      char fruit[] = "яблоко";
10     // загаданный фрукт
11     char answer[80];
12     do
13     {
14         cout << "Угадайте мой любимый фрукт! >> ";
15         cin >> answer;
16     } while (strcmp(fruit, answer) != 0);
17     // пока фрукт не отгадан, цикл будет работать
18     cout << "Правильный ответ!";
19     return 0;
20 }
```

```
Угадайте мой любимый фрукт! >> груша
Угадайте мой любимый фрукт! >> тиква
Угадайте мой любимый фрукт! >> кавун
Угадайте мой любимый фрукт! >> айва
Угадайте мой любимый фрукт! >> абрикос
Угадайте мой любимый фрукт! >> яблоко
Правильный ответ!
```

Завдання

Вхідні дані: 3 числа X, Y та Z. X, Y -- невід'ємні цілі числа, Z дорівнює 0 або 1. X не дорівнює 0. Передаються як аргументи командного рядка.

Вихідні дані: рядок "Everybody sing a song: <текст пісеньки>.", де <текст пісеньки> формується з у куплетів, розділених пробілами. Всі куплети однакові і складаються з X 'la' через дефіс та Y повторів. Якщо Z дорівнює одиниці, в кінці ставиться знак оклику, інакше крапка. За відсутності куплетів пробіл перед крапкою/ знаком оклику не ставиться.

```
Вхідні дані (вводити через пробіл):
```

```
2 3 1
```

```
Результат: Everybody sing a song: la-la la-la la-la!
```

```
Вхідні дані (вводити через пробіл):
```

```
1 0 0
```

```
Результат: Everybody sing a song:.
```

```
Вхідні дані (вводити через пробіл):
```

```
3 4 1
```

```
Результат: Everybody sing a song: la-la-la la-la-la la-la-la la-la-la!
```