



Урок 20. C++: Символы, индексы и перебор

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 20 из 33 · 45 минут · Модуль «Символьные строки»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

В коде партии A7-77 три цифры 7. Почему цикл до `size()` включительно обращается не к очередной букве, а за конец содержимого строки?

Результат урока: Перебрать строку ASCII без выхода за границы и сосчитать отдельные символы и соседние равные пары.

01. Проблемная ситуация

0–5 минут. Запишите под A7-77 индексы и отметьте три позиции с цифрой 7. Предскажите число проверок для пустой строки. Объясните, почему индекс последнего символа и длина различаются.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Индекс относится к элементу, длина — к строке

`std::string` хранит последовательность элементов `char`. Здесь входы ограничены ASCII: латинские буквы, цифры, пробелы и знаки. В UTF-8 русская буква обычно занимает несколько байтов, поэтому этот алгоритм нельзя объявлять подсчётом букв произвольного русского текста.

Для содержимого строки допустимы индексы от 0 до `size()-1`. Цикл `i < s.size()` не выполняет тело при пустой строке. Тип `std::size_t` подходит для индексов и размера. Не используйте `i <= size()`: элемент с таким индексом не является очередным символом содержимого.

Перебор по индексу нужен, когда важна позиция или сосед. Для соседней пары требуется `i+1 < size()`. Запись `i < size()-1` опасна при пустой строке: размер имеет беззнаковый тип, и вычитание единицы из нуля даёт огромное значение. Счётчик увеличивается только после успешного сравнения.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

В A7-77 длина 5; индексы 0,1,2,3,4. Успехи сравнения с символом '7' происходят на 1,3,4: счётчик 0→1→2→3. Символьный литерал пишется в одинарных кавычках, строковый — в двойных.

```
#include <iostream>
#include <string>
std::size_t count7(const std::string& s) {
    std::size_t count = 0;
    for (std::size_t i = 0; i < s.size(); ++i) {
        if (s[i] == '7') { ++count; }
    }
    return count;
}
int main() {
    std::string s;
    std::getline(std::cin, s);
    std::cout << s.size() << " " << count7(s) << "\n";
}
```



Индекс	0 1 2 3 4
A7-77	A 7 - 7 7
Равно 7?	нет да нет да да
Счётчик	0 1 1 2 3

Трассировка подсчёта цифры 7 в пятисимвольной строке ASCII; индекс начинается с нуля.

03. Практика 1. Проверяем материал

Журнал посещённых индексов

14–25 минут. Сделайте шесть запусков: ввод строки завершается Enter; пустой ввод — сразу Enter. До запуска запишите длину, посещённые индексы и счётчик.

Строка (кавычки не вводить)	Длина; число 7
""	0; 0
"7"; "A7-77"	1; 1 и 5; 3
" 7 "	3; 1
"777"; "abc"	3; 3 и 3; 0

2 минуты — прогноз, 4 — запуск, 3 — замените начало цикла на $i=1$ и найдите два контрпримера, 2 — восстановите код. Продукт: шесть строк журнала. Критерии: длины, индексы, ответы и объяснение ошибки.

04. Практика 2. Перенос способа

Соседи без вычитания из нуля

25–33 минуты. Сосчитайте соседние равные пары любых ASCII-символов, разрешая перекрытия: в "aaa" их две. За 2 минуты разметьте "aabb", "aaa", "ab", "", "x"; за 4 измените функцию; за 2 проверьте все пять входов. Критерии: сравнение соседей, перекрытия, отсутствие обращения вне содержимого, пустой и односимвольный случаи.

05. Обсуждение и границы вывода

33–40 минут. Один ученик выбирает цикл по `char`, второй — по индексу. За 2 минуты приведите задачу для каждого варианта, за 3 объясните риск `size()-1`, за 2 обозначьте ограничение ASCII. Может ли успешный тест "abc" подтвердить правильность работы с пустой строкой?

06. Индивидуальная проверка

40–45 минут. Для "7777" найдите длину, последний индекс, `count7` и `equal_pairs`. Объясните один из четырёх ответов через посещённые индексы. Четыре ответа с объяснением — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

При начале с 1 строка "7" даст 0 вместо 1; "777" даст 2 вместо 3. Для пустой строки тело не выполняется. Исправленный полный исходник дан в примере. Пробелы сохраняются благодаря getline, а не чтению отдельного слова.

Практика 2: разбор и критерии

Ответы: 2,2,0,0,0. Для "aabb" подходят пары индексов (0,1) и (2,3). Не надо перескакивать через найденную пару. Следующий исходник запускается отдельно.

```
#include <iostream>
#include <string>
std::size_t equal_pairs(const std::string& s) {
    std::size_t count = 0;
    for (std::size_t i = 0; i + 1 < s.size(); ++i) {
        if (s[i] == s[i + 1]) { ++count; }
    }
    return count;
}
int main() {
    std::string s;
    std::getline(std::cin, s);
    std::cout << equal_pairs(s) << "\n";
}
```

Обсуждение: возможный вывод

Для подсчёта отдельного символа достаточно char, для сравнения соседей удобнее индекс. У пустого входа собственный граничный случай; вычитание из беззнакового нуля его ломает. Перенос на UTF-8 требует решения, что считать символом, а не простой замены входного текста.

Индивидуальная проверка: ответ и критерии

4; 3; 4; 3. Пары начинаются на индексах 0,1,2; для пары с началом 3 нет второго символа. По баллу за каждый верный ответ; объяснение подтверждает самостоятельность.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Компилятор G++ с поддержкой C++17 и редактор. Сборка: g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson. Запуск ./lesson в Linux/macOS или .\lesson.exe в Windows (при сборке задайте -o lesson.exe). Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: цикл for, if, функции, std::string. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Символьные строки	Перебрать строку ASCII без выхода за границы и сосчитать отдельные символы и соседние равные пары.	Исходник, трассировка и таблица проверок
Обращение к символам; поиск, замена и преобразования	Соседи без вычитания из нуля	Самостоятельный алгоритм, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

C++: доступ к элементам строки

<https://eel.is/c++draft/string.access>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.