



Урок 21. C++: Поиск, вставка и замена

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 21 из 33 · 45 минут · Модуль «Символьные строки»

Уровень: начинающий программист; C++17.

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

Редактор должен превратить первую метку : в [. Как найти метку, не испортить строку без неё и учесть сдвиг после вставки?

Результат урока: Найти фрагмент, обработать pros и выполнить вставку и замену с учётом новых индексов.

01. Проблемная ситуация

0–5 минут. Для "ab:cd", ":ab", "abcd" предскажите нужный результат. Отметьте, где окажется двоеточие, если непосредственно перед ним вставить левую скобку.

02. Понятия и разобранный пример

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

После изменения позиции могут сдвинуться

find возвращает позицию первого совпадения или std::string::npos. Проверяйте именно равенство pros: нулевой индекс — успешный поиск в начале. Нельзя использовать результат отсутствующего поиска как позицию изменения.

insert(p, "[") вставляет перед p. Бывший символ на p переходит на p+1. replace(p+1, 1, "[") заменяет один старый символ правой скобкой. Здесь строка передаётся функции по значению: меняется копия; результат нужно присвоить или вывести.

При повторных заменах задайте правило: заменять найденные неперекрывающиеся вхождения слева направо. После замены "aa" на "b" продолжайте поиск после вставленной b. Иначе можно повторно обработать уже изменённый фрагмент или пропустить часть остатка. Все входы урока — ASCII длиной до 40.

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 "ab:cd" p=2. После insert получаем "ab[:cd": двоеточие на 3. Замена одного символа на 3 даёт "ab[]cd". Для "abcd" результат find равен pros, поэтому возвращаем исходную строку.

```
#include <iostream>
#include <string>
std::string bracket(std::string s) {
    auto p = s.find(':');
    if (p == std::string::npos) { return s; }
    s.insert(p, "[");
    s.replace(p + 1, 1, "[");
    return s;
}
int main() {
    std::string s;
    std::getline(std::cin, s);
    std::cout << bracket(s) << "\n";
}
```



Исходная строка	ab:cd; позиция : равна 2
Вставка на 2	ab[:cd; позиция : стала 3
Замена на 3	ab[]cd
Длина	5 → 6 → 6

Сдвиг индекса после вставки перед первым двоеточием; замена сохраняет длину полученной строки.

03. Практика 1. Проверяем материал

Первая метка и случай без метки

14–25 минут. Для шести входов составьте журнал: `p`, промежуточная строка после `insert`, итог. Вводите без окружающих кавычек.

Вход	Выход bracket
"ab:cd"; ":ab"	"ab[]cd"; "[]ab"
"ab:"; "a:b:c"	"ab[]"; "a[]b:c"
"abcd"; ""	"abcd"; ""

2 минуты — прогноз, 4 — запуски, 3 — замените `p+1` на `p` в `replace` и зафиксируйте ошибку на "ab:cd", 2 — восстановите код. Критерии: `pros`, начало строки, промежуточный индекс и исправление.

04. Практика 2. Перенос способа

Замены слева направо без перекрытия

25–33 минуты. Замените все неперекрывающиеся "aa" на "b" слева направо. За 2 минуты предскажите результаты "aaaaa", "baaa", "aa", "aba", ""; за 4 напишите функцию с `find` и `replace`, за 2 проверьте. Критерии: правильная позиция продолжения, перекрытия не учитываются, отсутствие совпадения и пустая строка.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните поиск первой метки и всех совпадений. За 3 объясните, почему после изменения нельзя без проверки использовать старые индексы. За 2 предложите журнал, который обнаружит ошибку даже при похожем итоговом тексте.

06. Индивидуальная проверка

40–45 минут. Для ":aa:aa" запишите `bracket`, затем результат `compress` от исходной строки, число выполненных замен `compress` и объяснение роли `pros`. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ошибочная замена на `p` уничтожает только что вставленную левую скобку: получится `"ab]:cd"`. Правильный результат `"ab[]cd"`. Только первое двоеточие меняется; `"a:b:c"` даёт `"a[]b:c"`. Полный правильный исходник приведён выше.

Практика 2: разбор и критерии

Получается `"bba"`, `"bba"`, `"b"`, `"aba"`, `""`. В `"aaaaa"` два вхождения `"aa"` и одна оставшаяся `a`. После вставленной `b` продолжаем с `p+1`.

```
#include <iostream>
#include <string>
std::string compress(std::string s) {
    auto p = s.find("aa");
    while (p != std::string::npos) {
        s.replace(p, 2, "b");
        p = s.find("aa", p + 1);
    }
    return s;
}
int main() {
    std::string s;
    std::getline(std::cin, s);
    std::cout << compress(s) << "\n";
}
```

Обсуждение: возможный вывод

Журнал должен показывать `p`, строку до изменения, после изменения и начало нового поиска. Равенство длины не доказывает правильность: `"ab]:cd"` и `"ab[]cd"` одинаковой длины. Вход без совпадения нужен отдельно.

Индивидуальная проверка: ответ и критерии

bracket: `"[]aa:aa"`; compress: `":b:b"`; две замены. `pros` означает отсутствие совпадения и останавливает цикл, а не обозначает пригодный для изменения индекс.

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: строки ASCII, индексы, цикл `while`, функции. 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	Найти фрагмент, обработать <code>pros</code> и выполнить вставку и замену с учётом новых индексов.	Исходник, трассировка и таблица проверок
Обращение к символам; поиск, замена и преобразования	Замены слева направо без перекрытия	Самостоятельный алгоритм, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

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

C++: вставка и замена в строке

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

C++: поиск и `pros`

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

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.