

Programming Assignment: Текстовый редактор

✓ Passed • 1/1 points

**Deadline** The assignment was due on May 12, 11:59 PM PDT  
You can still pass this assignment before the course ends.

**Instructions** My submission Discussions

Рассмотрим упрощённый текстовый редактор, поддерживающий следующий набор команд:

- Перемещение курсора влево (**Left**) и вправо (**Right**) на одну позицию.
- Ввод символа в текущую позицию курсора (**Insert**).
- Копирование фрагмента текста, **начинающегося в текущей позиции курсора**, в буфер обмена (**Copy**).
- Вырезание фрагмента текста аналогично копированию с последующим удалением скопированных символов из текста (**Cut**).
- Вставка содержимого буфера обмена в текущую позицию курсора (**Paste**).

Эти команды действительно поддерживаются практически любым текстовым редактором. Также для команд действуют стандартные правила, определяющие их эффект:

- Если редактор содержит текст длиной  $n$  символов, то курсор может находиться в одной из  $(n + 1)$  возможных позиций. Например, обозначим курсор вертикальной чертой `|` (это обозначение мы будем использовать в дальнейшем), тогда в тексте `abc` курсор может располагаться в 4-х позициях: `|abc`, `a|bc`, `ab|c`, `abc|`. Поэтому команда **Left** не имеет эффекта, когда курсор расположен в начале текста, а **Right** не имеет эффекта, когда курсор находится в конце. В частности, ни **Left**, ни **Right** не имеют эффекта, когда редактор не содержит текста.
- Введённый символ располагается в позиции курсора, сдвигая курсор и весь текст справа от него на одну позицию вправо. Аналогично, при вставке фрагмента длиной  $n$  курсор и текст справа от него смещаются на  $n$  позиций вправо. В таблице ниже приведены примеры, демонстрирующие данное правило.

До вставки	Введённый символ/вставляемый фрагмент	После вставки
	a	a
ab	c	abc
bc	a	a bc
world	hello	hello_ world
hello_	world	hello_world
123 78	456	123456 78

- Буфер обмена изначально пуст. Вставка пустого фрагмента не имеет эффекта. Содержимое буфера не сбрасывается после вставки, а остается неизменным до следующей команды **Copy** или **Cut**. Копирование или вырезание фрагмента нулевой длины не оказывает влияние на текст, но **опустошает буфер обмена**. Курсор не смещается ни при копировании, ни при вырезании текста. Например, после вырезания из текста `abcdef` фрагмента из трёх символов, получим текст `ab|f`.

Вам предстоит реализовать «ядро» текстового редактора, поддерживающего все описанные операции, в виде класса `Editor`, имеющего следующий интерфейс:

```
1 class Editor {
2 public:
3     Editor();
4     void Left(); // сдвинуть курсор влево
5     void Right(); // сдвинуть курсор вправо
6     void Insert(char token); // вставить символ token
7     void Copy(size_t tokens); // скопировать
8         // не более tokens символов,
9     void Cut(size_t tokens); // вырезать не более tokens символов,
10        // начиная с текущей позиции курсора
11     void Paste(); // вставить содержимое буфера
12        // начиная с текущей позиции курсора
13     string GetText() const; // получить текущее содержимое
14        // текстового редактора
15 };
```

Как можно заметить, каждой описанной команде сопоставлен метод класса `Editor`. Также введён метод `GetText()`, позволяющий получить текущее содержимое редактора в виде строки. В комментариях к методам `Cut(size_t tokens)` и `Copy(size_t tokens)` указано, что в буфер обмена попадает фрагмент длиной не более `tokens` символов — это означает, что, если справа от курсора располагается менее, чем `tokens` символов, методы `Cut()` и `Copy()` должны вырезать/скопировать все символы справа.

Заготовка решения

text\_editor.cpp

Ограничения

Реализация класса `Editor` должна обрабатывать  $10^6$  запросов не более чем за 1 секунду. При этом гарантируется, что:

- длина текста никогда не превышает  $10^6$  символов;
- методы `Copy()` и `Cut()` **суммарно** копируют/вырезают не более  $10^6$  символов;
- метод `GetText()` вызывается один раз в каждом из тестов.

Пример

Код

```
1 int main() {
2     Editor editor;
3     const string text = "hello, world";
4     for (char c : text) {
5         editor.Insert(c);
6     }
7     // Текущее состояние редактора: "hello, world"
8     for (size_t i = 0; i < text.size(); ++i) {
9         editor.Left();
10    }
11    // Текущее состояние редактора: "hello, world"
12    editor.Cut(7);
13    // Текущее состояние редактора: "world"
14    // в буфере обмена находится текст "hello, "
15    for (size_t i = 0; i < 5; ++i) {
16        editor.Right();
17    }
18    // Текущее состояние редактора: "world"
19    editor.Insert(" ");
20    editor.Insert(" ");
21    // Текущее состояние редактора: "world, "
22    editor.Paste();
23    // Текущее состояние редактора: "world, hello, "
24    editor.Left();
25    editor.Left();
26    // Текущее состояние редактора: "world, hello, "
27    editor.Cut(3); // Будут вырезаны 2 символа
28    // Текущее состояние редактора: "world, hello"
29    cout << editor.GetText();
30 }
```

Вывод

1 world, hello

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.