# Задание по программированию: Текстовый редактор

Условие

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 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** | **Вводимый 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 влияния на текст, но **опустошает буфер обмена**. Курсор не смещается ни при копировании, ни при вырезании текста. Например, после вырезания из текста ab|cdef 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:

class Editor {

public:

  Editor();

  void Left();   // сдвинуть курсор влево

  void Right();  // сдвинуть курсор вправо

  void Insert(char token);   // вставить символ token

  void Copy(size\_t tokens);  // cкопировать

                             // не более tokens символов,

                             // начиная с текущей позиции курсора

  void Cut(size\_t tokens);  // вырезать не более tokens символов,

                            // начиная с текущей позиции курсора

  void Paste();  // вставить содержимое буфера

                 // в текущую позицию курсора

  string GetText() const;  // получить текущее содержимое

                           // текстового редактора

};

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](https://d3c33hcgiwev3.cloudfront.net/unT46Hq0EeiLrA7SIbVCTg_bae4d1207ab411e8b20dcf3779aa5f1e_text_editor.cpp?Expires=1617062400&Signature=MyCu0oMYp2DAfZr7TWzd3slZ4d6NCQxgTNq1KogJsvvTRgoFJ7W~wm1OB-rQRiV4vuWZNDIdfOACqYVijftoA~uo-iHSryc6TEDKm30Z6t5zhF5Q8up9iGPjvCD26TlYM1XuDwODfESTzFAdkRk1CE1Pd9wxKkS0cCS9O88xXrs_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

### 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

int main() {

  Editor editor;

  const string text = "hello, world";

  for (char c : text) {

    editor.Insert(c);

  }

  // Текущее состояние редактора: `hello, world|`

  for (size\_t i = 0; i < text.size(); ++i) {

    editor.Left();

  }

  // Текущее состояние редактора: `|hello, world`

  editor.Cut(7);

  // Текущее состояние редактора: `|world`

  // в буфере обмена находится текст `hello, `

  for (size\_t i = 0; i < 5; ++i) {

    editor.Right();

  }

  // Текущее состояние редактора: `world|`

  editor.Insert(',');

  editor.Insert(' ');

  // Текущее состояние редактора: `world, |`

  editor.Paste();

  // Текущее состояние редактора: `world, hello, |`

  editor.Left();

  editor.Left();

  //Текущее состояние редактора: `world, hello|, `

  editor.Cut(3); // Будут вырезаны 2 символа

  // Текущее состояние редактора: `world, hello|`

  cout << editor.GetText();

}

Вывод

world, hello