

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

### Детали реализации

Идея описана выше. Особенности реализации нет (как слышится так и пишется).

Реализуйте класс `StringView` с поддержкой:

- Конструктора по умолчанию (`nullptr` строка).
- Конструктора от `const char*` (с подсчетом длины). Этот конструктор должен допускать неявные преобразования. Чтобы на это не ругался линтер, добавьте к строке с объявлением комментарий `// NOLINT`
- Конструктора от `const char*` и `size_t` (длина передается вторым параметром).
- Операции `[]`.
- Метода `At` (при выходе за границы бросает исключение `StringViewOutOfRange`, которое написано за вас).
- Методов `Front()` и `Back()`, которые возвращают первый и последний символ строки соответственно.
- Методов `Size()` и `Length()`, которые возвращают длину строки.
- Метод `Empty()` проверяет строку на пустоту.
- Метод `Data()` возвращает указатель на начало строки.
- Метода `Swap(StringView&)`.
- Метода `RemovePrefix(size_t prefix_size)`, который сдвигает начало строки на `prefix_size`.
- Метода `RemoveSuffix(size_t suffix_size)`, который сдвигает конец строки на `suffix_size`.
- Метода `Substr(size_t pos, size_t count = -1)`, который формирует из вызывающего объекта `StringView` на подстроку с началом `pos` и длины `min(count, Size() - pos)` (если `pos` выходит за границы, то бросает `StringViewOutOfRange`).