

В этой задаче запрещено пользоваться какими-либо стандартными контейнерами. Выделения памяти нужно делать вручную. Разрешается подключать только `<iostream>` и `<cstring>`.

Напишите класс `String` - упрощенный аналог библиотечного `std::string`. Необходимо поддержать следующий функционал:

- Конструктор от C-style строки (`const char*`).
- Конструктор от двух параметров: число `n` и символ `c` - создает строку из `n` копий символа `c`.
- Конструктор по умолчанию, копирующий конструктор и оператор присваивания.
- Оператор `==`, позволяющий проверять строки на совпадение.
- Квадратные скобки, корректно работающие как для константных, так и для неконстантных строк.
- Метод `length()`, возвращающий размер строки.
- Методы `push_back(char)` и `pop_back()`. Учетное время работы должно быть $O(1)$.
- Методы `front()` и `back()`, возвращающие ссылку на начальный и на последний символы строки соответственно.
- Оператор `+=`, позволяющий добавить к строке символ или другую строку. Добавление символа должно работать за амортизированное $O(1)$.
- Оператор `+`, позволяющий складывать строки с символами, символы со строками и строки со строками.
- Методы `find(substring)` и `rfind(substring)`, позволяющие найти самое левое и самое правое вхождение подстроки `substring` в данную строку. Возвращают индекс найденного вхождения. Если вхождений не найдено, нужно вернуть `length()` (фиктивный индекс).
- Метод `substr(start, count)`, возвращающий подстроку начиная с индекса `start` и состоящую из `count` символов.
- Метод `empty()`, проверяющий, пуста ли строка. Метод `clear()`, позволяющий очистить строку.
- Операторы вывода в поток « и ввода из потока ».

В вашем файле должна отсутствовать функция `main()`, а сам файл должен называться `string.h`. В качестве компилятора необходимо указывать `Make C++17`. Ваш код будет вставлен посредством команды `#include "string.h"` в программу, содержащую тесты; вследствие этого код необходимо отправлять в файле со строго соответствующим именем!