

Вариант 2

1. Создайте функцию `backspace`, удаляющую символ в данной C-строке по указателю на него. (Замечание: указатель на начало строки не требуется. Алгоритм: необходимо сдвинуть все символы справа от удаляемого, включая завершающий нулевой байт, на одну позицию влево). Создайте функцию, удаляющую с помощью `backspace` все заглавные латинские в данной C-строке.

2. «Амперсанды атакуют!»

Назовём три подряд идущих амперсанда атакующей группой, а все символы, расположенные в строке правее амперсандов (вплоть до следующей атакующей группы), атакуемой группой, например:

&&&xxx&&&yyyyy&&&hhhhh

Определите структуру `DreadAmpsInfo` для хранения информации об атакующих амперсандах:

- индекс начальной позиции атакующей группы;
- индекс конечной позиции атакуемой группы;
- собственно атакуемая группа (строкового типа).

Создайте функцию, которая по заданной строке (`std::string`) формирует вектор структур `DreadAmpsInfo` с информацией об атакующих амперсандах в строке.

Ограничение. Запрещается использовать операцию `[]` для поиска «начала» и «конца» атакующих и атакуемых групп и выделения подстрок.

3. Перегрузите операции `+=` и `+` для `DreadAmpsInfo`, которые конкатенируют поля атакуемых групп. (Результат — типа `DreadAmpsInfo`, возможно ссылка).
4. Перегрузите операцию `<` для `DreadAmpsInfo`: она должна сравнивать строки атакуемых групп лексикографически (`c_str`, `strcmp`).
5. Создайте класс `DreadAmpsInfoArray`, хранящий вектор элементов `DreadAmpsInfo`, перегрузите операцию `[]` для него: по данному целому индексу нужно возвращать строку атакуемой группы соответствующего элемента хранимого вектора `DreadAmpsInfo`.