

NÉV (olvasható):

NEPTUN kód:

Készíts ShortText osztályt maximum 6 karakterből álló szövegek tárolására! Az osztálynak két adattagja van: egy 6 elemű karaktertömb és a szöveg hossza, ezeket az osztályon kívülről elérhetetlenül tárolja.

- Legyen alapértelmezett konstruktor, amely a szöveghosszt 0-ra állítja.
- Legyen egy olyan konstruktor is, amely egy C sztringet vesz át, és azt másolja be a tárolóba. Ha a C sztring 6 karakternél hosszabb, akkor csak az első 6 karaktert.
- Legyen olyan konstans függvény, amellyel a ShortText hossza lekérdezhető.
- Készíts [] operátort, amely a ShortText megadott indexű karakterének referenciáját adja. Hibakezelés most nem kell.
- Készíts += operátort, amely a ShortText végéhez fűz egy karaktert. Ha nem fér bele, ne csináljon semmit.
- Készíts olyan -- operátort, amely törli az utolsó karaktert a ShortText-ből. Üres ShortText esetén ne csináljon semmit. (Eldöntheted, hogy a predekremens vagy a posztdrekremens operátort definiáld-e.)

Készíts main függvényt, melyben bemutatom a megírt függvények működését!

max. 10p, megoldási idő: 15 perc

```
#include <iostream>

class ShortText { // jó osztálydeklaráció: 1p
    char t[6], n; // privát tagváltozók: 1p
public: // publikus függvények: 1p
    ShortText() :n(0) {} // 0.5p
    ShortText(const char* s) { // 1p
        for (n = 0; n < 6 && s[n] != '\0'; n++)
            t[n] = s[n];
    }
    int getLength()const { return n; } // 0.5p
    char& operator[](int i) { return t[i]; } // 1p
    const ShortText& operator+=(char c)
        { if (n < 6)t[n++] = c; return *this; } // void is jó // 1p
    ShortText& operator--() { if (n > 0)n--; return *this; } // 1p
};

int main() { // 2p (Mind a 6 függvényt be kell mutatni. Rossz / hiányzik: -0.5p / db.)
    ShortText a, b("abrakadabra");
    for (int i = 0; i < b.getLength(); i++)
        std::cout << b[i];
    std::cout << std::endl;
    --b;
    b += '!';
    for (int i = 0; i < b.getLength(); i++)
        std::cout << b[i];
    std::cout << std::endl;
    return 0;
}
```