

Programozás alapja 2.	2. ellenőrző dolgozat.	2016.03.22.	Kurz/Terem: G4/	Elért pontszám:
Név:			Neptun:	

Az alábbi osztályt szövegek tárolására készítettük. Az osztályból létrehozott objektumok a tárolt szöveget – egy közös beállítástól függően – kis, vagy nagybetűs formában írják ki.

```
class MyStr {
    char *p;
    static bool nagybetus;
public:
    MyStr(const char *s) { p = new char[strlen(s)+1]; strcpy(p, s); }
    std::ostream& kiir(std::ostream& os) const {
        int len = strlen(p);
        for (int i = 0; i < len; i++) {
            char ch = nagybetus ? toupper(p[i]) : tolower(p[i]);
            os << ch;
        }
        return os;
    }
    static void setNagybetus(bool b) { nagybetus = b; }
    MyStr(const MyStr& s) {
        p = new char[strlen(s.p)+1];
        strcpy(p, s.p);
    }
    ~MyStr() { delete[] p; }
};
```

```
};
bool MyStr::nagybetus = false;

std::ostream& operator<<(std::ostream& os, const MyStr& s) {
    return s.kiir(os);
}
```

1. Egészítse ki az osztályt úgy, hogy az alábbi kódrészlet hiba nélkül (ide értve a memóriaszivargást is), az elvárásoknak megfelelően működjön! Ha szükséges, használja az üres helyeket az osztály deklarációjában! Az elvárások megértéséhez a megjegyzések segítenek! (10p)

```
{
    MyStr s1("Hello ");
    MyStr s("World");
    const MyStr s2 = s;
    std::cout << s1 << s2 << std::endl; // kiír: hello world
    MyStr::setNagybetus(true); // nagybetűsre állítja a kiírást
    std::cout << s1 << s2 << std::endl; // kiír: HELLO WORLD
}
```

Programozás alapja 2.	2. ellenőrző dolgozat.	2016.03.22.	Kurz/Terem: G4/	Elért pontszám:
Név:			Neptun:	

Az alábbi osztályt szövegek tárolására készítettük. Az osztályból létrehozott objektumok a tárolt szöveget – egy közös beállítástól függően – visszafelé írják ki.

```
class RevStr {
    char *p;
    static bool vissza;
public:
    RevStr(const char *s) { p = new char[strlen(s)+1]; strcpy(p, s); }
    std::ostream& kiir(std::ostream& os) const {
        int len = strlen(p);
        for (int i = 0; i < len; i++) {
            int j = i;
            if (vissza) j = len - i - 1;
            os << p[j];
        }
        return os;
    }
    ~RevStr() { delete[] p; }
    static void setVissza(bool b) { vissza = b; }
    RevStr& operator=(const RevStr& s) {
        if (this != &s) {
            delete[] p;
            p = new char[strlen(s.p)+1];
            strcpy(p, s.p);
        }
        return *this;
    }
};
```

```
};
bool RevStr::vissza = false;
```

```
std::ostream& operator<<(std::ostream& os, const RevStr& s) {
    return s.kiir(os);
}
```

2. Egészítse ki az osztályt úgy, hogy az alábbi kódrészlet hiba nélkül (ide értve a memóriaszivárgást is), az elvárásoknak megfelelően működjön! Ha szükséges, használja az üres helyeket az osztály deklarációjában! Az elvárások megértéséhez a megjegyzések segítenek! (10p)

```
{
    using std::cout;
    using std::endl;
    const RevStr s1("Indul a gorog aludni");
    RevStr s2(""), s("Az nem menza");
    s2 = s;
    cout << s1 << " " << s2 << endl;    // kiír: Indul a gorog aludni Az nem menza
    RevStr::setVissza(true);
    cout << s1 << " " << s2 << endl;    // kiír: indula gorog a ludnI aznem men zA
}
```