

|                       |                        |             |                 |                 |
|-----------------------|------------------------|-------------|-----------------|-----------------|
| Programozás alapja 2. | 2. ellenőrző dolgozat. | 2016.03.21. | Kurz/Terem: G2/ | Elért pontszám: |
| Név:                  |                        |             | Neptun:         |                 |

Az alábbi osztályt DNS szekvenciák tárolására készítettük:

```
class DNS {
    char *seq;
public:
    DNS() { seq = new char[1]; seq[0] = '\0'; }
    DNS& operator+=(char ch) {
        int len = strlen(seq);
        char *tmp = new char[len+2];
        strcpy(tmp, seq);
        tmp[len] = ch; tmp[len+1] = '\0';
        delete[] seq;
        seq = tmp;
        return *this;
    }
    const char* c_str() const { return seq; }
    ~DNS() { delete[] seq; }
    DNS(const DNS& d) {
        seq = new char[strlen(d.seq)+1];
        strcpy(seq, d.seq);
    }
};
```

1. Magyarozza meg, hogy az alábbi kódrészlet futtatásakor miért lép fel memóriahiba! (1p)

```
{ DNS d; d += 'A'; d += 'C'; d += 'G'; d += 'T';
  std::cout << d.c_str() << std::endl; }
```

Az alapértelmezett destruktorként nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa az osztályt, hogy a fenti kódrészlet futtatása ne okozzon hibát! Használja az üres helyeket az osztály deklarációjában! (2p)

Mit ír a szabványos kimenetre a fenti kódrészlet? **ACGT** (1p)

2. Milyen problémára kell számítani, ha egy DNS osztályból létrehozott objektumot értékparaméterként akarunk átadni? Mi a probléma forrása? (1p)

Értékparaméter átadásakor a másoló konstruktor hívódik. Az alapértelmezett másoló konstruktor azonban nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa az osztályt, hogy az abból létrehozott objektumok átadhatók legyenek értékparaméterként! Használja az üres helyeket az osztály deklarációjában! (3p)

3. Mi a hiba az alábbi kódrészletben? Adja meg a hiba jellegét is (fordítási vagy futási)! (2p)

```
{ DNS d1('A'), d2; char* p = d2.c_str(); std::cout << p; }
```

Az osztálynak nincs 1 paraméteres konstruktora.

c\_str() const char\* értékű. Ez nem tárolható el char\* típusú változóban.

Ezek fordítási hibát okoznak.

|                       |                        |             |                 |                 |
|-----------------------|------------------------|-------------|-----------------|-----------------|
| Programozás alapja 2. | 2. ellenőrző dolgozat. | 2016.03.21. | Kurz/Terem: G2/ | Elért pontszám: |
| Név:                  |                        |             | Neptun:         |                 |

Az alábbi osztályt szövegek tárolására készítettük:

```
class Str {
    char *p;
public:
    Str(const char *s) { p = new char[strlen(s)+1]; strcpy(p, s); }
    const char* c_str() const { return p; }
    char& operator[](int i) { return p[i]; }
    ~Str() { delete[] p; }
    Str& operator=(const Str& s) {
        if (this != &s) {
            delete[] p;
            p = new char[strlen(s.p)+1];
            strcpy(p, s.p);
        }
        return *this;
    }
};
```

1. Magyarázza meg, hogy az alábbi kódrészlet futtatásakor miért lép fel memóriahiba! (1p)
- ```
{ Str s("Hello"); std::cout << s.c_str() << std::endl; }
```

Az alapértelmezett destruktork nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa az osztályt, hogy a fenti kódrészlet futtatása ne okozzon hibát! Használja az üres helyeket az osztály deklarációjában! (2p)

2. Mi a hiba az alábbi kódrészletben? Adja meg a hiba jellegét is (fordítási vagy futási) (1p)
- ```
{ const Str s("Hello"); std::cout << s[0] << std::endl; }
```

Konstans objektumra csak konstans tagfüggvény alkalmazható. Az operator[] nem konstans.

3. Milyen problémára számíthatunk, ha **str** típusú objektumok értékadás bal és jobb oldalán szerepelnének? Mi a probléma forrása? (1p)

Az alapértelmezett értékadó operátor nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa az osztályt, hogy az abból létrehozott objektumok többszörös értékadás operandusaként is szerepelhessenek! Használja az üres helyeket az osztály deklarációjában! (5p)