

Programozás alapja 2.	3. beszámoló dolgozat. 2017.03.16.	Kurz/Terem: Gy1/	Elért pontszám:
Név:	Neptun:		

Tételezze fel, hogy a **Komplex** osztály létezik, tökéletesen működik, és mindenben megfelel a feladat elvárásainak. Az alábbi osztályt komplex számok tárolására készítettük:

```
class Szamok {
    unsigned siz;
    Komplex *p;
public:
    Szamok(unsigned s = 0) :siz(s)          { p = new Komplex[siz]; }
    unsigned size() const                  { return siz; }
    Komplex& operator[](unsigned i)        { return p[i]; }
    ~Szamok() { delete[] p; }
    Szamok& operator=(const Szamok& t) {
        if (this != &t) {
            delete[] p;
            siz = t.siz;
            p = new Komplex[siz];
            for (unsigned i = 0; i < siz; i++)
                p[i] = t.p[i];
        }
        return *this;
    }
};
```

1. Magyarázza meg, hogy az alábbi függvény futtatásakor miért keletkezik memóriakezelési hiba! (1p)

```
Komplex osszeg(Szamok& t) {
    Komplex sum = 0;
    for (unsigned i = 0; i < t.size(); i++) sum += t[i];
    return sum;
}
```

Értékparaméter másoló konstruktorral másolódik be a verembe. Az alapértelmezett másoló konstruktor azonban nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa a fenti **osszeg** nevű függvényt, hogy ne jelentkezzen hiba! (1p)

A paramétert értékparaméter helyett referenciával kell átadni.

2. Magyarázza meg, hogy az alábbi kódrészlet futtatásakor miért keletkezik memóriakezelési hiba! (1p)

```
{ Szamok t1(3); t1 = t1 = Szamok(5); }
```

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

Módosítsa az osztályt, hogy a fenti kódrészlet futtatása ne okozzon hibát, és az általános elvárásoknak megfelelően működjön! Használja az üres helyeket az osztály deklarációjában! (5p)

3. Mi a hiba az alábbi kódrészletben? Adja meg a hiba jellegét is (fordítási vagy futási)! (1p)

```
{ Szamok t1; Szamok const t2; std::cout << t2.size(); }
```

A size tagfüggvény nem konstans, így az konstans objektumra nem alkalmazható. Ez fordítási hibát okoz.

Módosítsa az osztályt, hogy ne legyen hiba! Használja az üres helyeket az osztály deklarációjában! (1p)

Programozás alapja 2.	3. beszámoló dolgozat. 2017.03.16.	Kurz/Terem: Gy1/	Elért pontszám:
Név:	Neptun:		

Tételezze fel, hogy a String osztály létezik, tökéletesen működik, és mindenben megfelel a feladat elvárásainak. Az alábbi osztályt sztringek tárolására készítettük:

```
class StringTomb {
    unsigned siz;
    String *p;
public:
    StringTomb(unsigned s = 0) :siz(s)    { p = new String[siz]; }
    unsigned size()                       { return siz; }
    String& operator[] (unsigned i)       { return p[i]; }
    ~StringTomb() { delete[] p; }
    StringTomb& operator=(const StringTomb& t) {
        if (this != &t) {
            delete[] p;
            siz = t.siz;
            p = new String[siz];
            for (unsigned i = 0; i < siz; i++) p[i] = t.p[i];
        }
        return *this;
    }
};
```

1. Magyarozza meg, hogy az alábbi kódrészlet futtatásakor miért lép fel memóriakezelési hiba! (1p)
- ```
{ StringTomb t1(3), t2(3); t2 = t1 = t1; }
```

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

Módosítsa az osztályt, hogy a fenti kódrészlet futtatása ne okozzon hibát, és az általános elvárásoknak megfelelően működjön! Használja az üres helyeket az osztály deklarációjában! (5p)

2. Magyarozza meg, hogy az alábbi függvény futtatásakor miért keletkezik memóriakezelési hiba! (1p)
- ```
int darab(StringTomb& t1, StringTomb& t2) {
    return t1.size() + t2.size();
}
```

Értékparaméter másoló konstruktorral másolódik be a verembe. Az alapértelmezett másoló konstruktor azonban nem jó, mivel az osztály dinamikus memóriaterületet foglal és tart nyilván.

Módosítsa a fenti **darab** nevű függvényt, hogy ne jelentkezzen hiba! (1p)

Az első paramétert értékparaméter helyett referenciával kell átadni.

3. Mi a hiba az alábbi kódrészletben? Adja meg a hiba jellegét is (fordítási vagy futási)! (1p)
- ```
{ StringTomb t1[10]; StringTomb t2(3); t2[1] = 2; }
```

Az operator[] nem referencia értékű, így bal oldalon nem használható.  
Ez fordítási hibát okoz.

Módosítsa az osztályt, hogy ne legyen hiba! Használja az üres helyeket az osztály deklarációjában! (1p)