

| | |
|--|-----------|
| Programozás alapjai II. 5. ellenőrző dolgozat. 2013.05.02. Kurz/Terem: G2/ | 20 perc |
| Név: | Összpont: |
| Neptun: | |

1. feladat

3 pont

Tételezze fel, hogy rendelkezésére áll egy sorozattároló (*Vector*)! A tároló a szokásos vektorműveletek egy részhalmazával rendelkezik (*push_back*, *pop_back*, *front*, *back*, *size*, *resize*), és csak default konstruktora van. Ezen osztály felhasználásával készítsen egy véges méretű generikus tárolót (*MyVector*)! A maximális méretet a konstruktorban lehessen megadni, amelynek alapértelmezett értéke legyen 150! A *MyVector* osztály a *Vector* osztállyal **teljesen azonos** módon viselkedjen, kivéve akkor, amikor eléri a maximális méretet. Ekkor dobjon *std::out_of_range()* hibát!

Örökléssel egyszerűbb, de tartalmazott objektummal is megoldható volt.

Ez utóbbi esetben a *Vector* osztály minden függvényét delegálni kellett.

```
template<typename T>
class MyVector :public Vector<T> {
    size_t maxsiz;
public:
    MyVector(size_t n = 150) : maxsiz(n) {}
    void push_back(const T& a) {
        if (Vector<T>::size() >= maxsiz) throw std::out_of_range("MyVector: push");
        Vector<T>::push_back(a);
    }
    // a teljes megoldáshoz a resize() is hozzátartozik, de elfogadtuk enélkül is
    void resize(size_t n, const T& a = T()) {
        if (n > maxsiz) throw std::out_of_range("MyVector: resize");
        Vector<T>::resize(n, a);
    }
};
```

2. feladat

3 pont

Írjon függvénysablont (*myfind_if*), ami az *std::find_if* sablonhoz hasonlóan megkeresi egy sorozatban azt az elemet, ami a függvény 3. paramétereként átadott predikátumot kielégíti. A függvény első két paramétere két iterátor (*InputIterator*), ami kijelöli a jobbról nyílt intervallum kezdetét és végét. A függvény visszatérési értéke egy iterátor, ami a megtalált elemre vagy a sorozat végére mutat. Amennyiben helyesen oldja meg a feladatot, akkor az alábbi kódrészlet a következőt írja ki: 17

```
using namespace std;
int v[] = { 1, 2, 2, 8, 3, 17, 28 };
cout << *myfind_if(v, v+7, bind2nd(greater<int>()), 10) <<endl;
```

```
template <class Iter, class Func>
Iter myfind_if(Iter first, Iter last, Func func) {
    while (first != last) {
        if (func(*first))
            return first;
        ++first;
    }
    return last;
}
```