

Programozás alapjai 2.	4. ellenőrző dolgozat.	2014.04.23.	Kurz/Terem: G2/
Név:	Neptun:	Összpont:	

1. feladat

3 pont

Írjon függvénysablont (*mytransform*), ami egy adatsorozatból új sortozatot képez! A sablon az új sorozat elemeit az eredeti sorozat megegyező pozíciójú elemeiből a 4. paramétereként megadott egyparaméteres függvénnyel/függvényobjektummal képezzel! A *mytransform* függvény első három paramétere három iterátor, amelyek közül az első kettő a bemeneti adatokat jelöli ki (jobbról nyílt intervallum kezdete és vége). A harmadik iterátor pedig az eredményt tároló sorozat elejére mutat. Ha jól oldja meg a feladatot, akkor az alábbi kódrészlet lefutása után az out tömb tartalma 1,2,3,4,5 lesz.

```
inline double mysqrt(double d) { return sqrt(d); }
double inp[] = { 1, 4, 9, 16, 25}; // bemeneti sorozat
double out[5]; // eredmény helye
...
mytransform(inp, inp+5, out, mysqrt);

template<typename Iterator, typename Function>
void mytransform(Iterator first, Iterator last, Iterator result, Function op) {
    while (first!=last)
        *result++ = op(*first++);
}
```

2. feladat

3 pont

Készítsen olyan függvény-, vagy függvényobjektum sablont, ami kiszámítja a paraméterként kapott generikus adat háromszorosát! Feltételezheti, hogy a generikus adatnak van *T operator*(int) const*; alakú operátora! Mutassa be az elkészített sablon használatát a *mytransform* sablonnal a következő módon:

1. Az *std::vector* sablon felhasználásával hozzon létre kettő darab 100 elemű int értékeket tartalmazó vektort!
2. Másolja át az egyik vektor elemeinek háromszorosát a másik vektorba a *mytransform* felhasználásával!

```
std::vector<int> v1(100), v2(100);

//Függvénysablonnal:
template<typename T>
inline T triplaz(T x) { return x*3; }

mytransform(v1.begin(), v1.end(), v2.begin(), triplaz<int>);

//Függvényobjektummal:
template<typename T>
struct Triplaz {
    T operator()(T x) { return x*3; }
};

mytransform(v1.begin(), v1.end(), v2.begin(), Triplaz<int>());
```