4. labor kis ZH, 2015. április 16., csütörtök 10-12

NÉV (olvasható): NEPTUN kód:

max. 10p, megoldási idő: 15 perc

1. Adott a következő komplex szám osztály. Készítsd el hozzá az ostreamre kiíró << is az istreamről olvasó >> operátort! A Komplex osztály deklarációját nem módosíthatod. A kiírás/beolvasás formátuma: 3.5+4i. (3p)

Tipp: beolvasás: két számot és az „i” betűt kell beolvasni.
Kiírás: …`<< std::showpos << szam << std::noshowpos <<`… módon a szám előtt nem csak a -, hanem a + előjel is megjelenik.

```
class Komplex{
   double re, im;
public:
   Komplex(double re = 0, double im = 0) : re(re), im(im){}
   void get(double re = 0, double im = 0){ this->re = re; this->im = im; }
   double getRe()const{ return re; }
   double getIm()const{ return im; }
   double getAbs()const{ return sqrt(re*re + im*im); }
   Komplex operator+(const Komplex &b)const{ return Komplex(re + b.re, im + b.im); }
};
```

(A 2. feladat a lap túlolddalán!)

```
std::istream & operator>>(std::istream &is, Komplex & c){
    double re, im;
    char ch;
    is >> re >> im >> ch;
    c.set(re, im);
    return is;
}

std::ostream &operator<<(std::ostream &os, const Komplex & c){
    os << c.getRe() << std::showpos << c.getIm() << std::noshowpos << 'i';
    return os;
}
```

2. Készíts programot, amelyben létrehozol egy std::vector-t a fenti komplex típusú értékek tárolására!
- Töltsd fel a vektort 5 komplex számmal, amelyeket a felhasználótól olvasol be a fenti >> operátorral!
- Keresd meg a vektor maximális abszolútértékű elemét, és írd ki azt! A vektor bejárásához nem használhatsz [] operétort, **használj iterátort!**
- Definiálj még egy vector-t, melybe az eredeti vektor minden elemének a duplája kerüljön! Az eredeti vektor bejárásához [] operátort használj! A duplázához a + operátort használd (y=x+x)!
- Írd ki a második vektorban tárolt számokat, soronként egyet!

```
int main(){
    std::vector<Komplex> a(5);

    for (int i = 0; i < 5; i++)
        std::cin >> a[i];

    Komplex max = a[0];
    for (std::vector<Komplex>::iterator it = a.begin(); it != a.end(); it++)
        if (it->getAbs() > max.getAbs())
            max = *it;
    std::cout << "Max: " << max << std::endl;

    std::vector<Komplex> b(5);
    for (int i = 0; i < 5; i++)
        std::cout << (b[i] = a[i] + a[i]) << std::endl;

    return 0;
}
```