

NÉV (olvasható):

NEPTUN kód:

Készíts 3D vektor osztályt, amely az x, y, z koordinátákat valós típusú tagváltozóban, az osztályon kívülről elérhetetlenül tárolja! Ami lehet és van értelme, legyen konstans.

- Készíts getter és setter függvényeket a tagváltozók lekérdezéséhez és beállításához!
- Legyen alapértelmezett konstruktor, amely 0-ra állítja a számot, valamint olyan konstruktor is, amelynek megadható a három koordináta értéke! (Egyben is lehet.)
- Írj globális függvényt, amely kiszámítja egy 3D vektor hosszát!
- Írj tagfüggvény + operátort, amely a + operátortól elvárt módon adja össze a két vektort!
- Készíts main függvényt, melyben bemutatom a megírt függvények működését!

max. 10p, megoldási idő: 18 perc

```
#include <cmath>
#include <cstdio>

class Vec3d { // jó osztályszerkezet: 1p
    double x, y, z; // private: 1p, nem valós: -1p
public:
    Vec3d(double X = 0, double Y = 0, double Z=0) :x(X), y(Y), z(Z) {} // 0 és 2 paraméteres: 2p
    double getX()const { return x; } // 1p
    double getY()const { return y; }
    double getZ()const { return z; }
    void setX(double X) { x = X; } // 1p
    void setY(double Y) { y = Y; }
    void setZ(double Z) { z = Z; }
    const Vec3d & operator+(const Vec3d &b)const { // 2p
        return Vec3d(x + b.x, y + b.y, z + b.z);
    }
};

double hossz(const Vec3d &a) { // 2p
    return sqrt(a.getX()*a.getX()+a.getY()*a.getY()+a.getZ()*a.getZ());
}

int main() { // -0,5p/db, ha hiányzik valamelyik, max. -2p
    Vec3d a(3, 4, 5), b(-1, 6, 3.5), c;
    c = a + b;
    b.setX(0);
    b.setY(0);
    b.setZ(0);
    printf("A (%g,%g,%g) vektor hossza %g\n", c.getX(), c.getY(), c.getZ(), hossz(c));
    return 0;
}
```