

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 20 perc

a) Definiálj *Adat* osztályt, amely publikus adatként két egész számot tárol: egy osztályzatot (1-5) és egy kreditet (0-30)! Definiálj $\gg$ és $\ll$ operátort, melyekkel beolvashatsz istream-ről és kiírhatsz ostream-re egy ilyen objektumot!

b) Írj programot, amely a szabványos bemenetről olvas be osztályzat-kredit párokat egészen az adatfolyam végét jelző 0 0 jegy-kredit párig! Használd az a) feladatban megírt beolvasó operátort! Tárold el a beolvasott adatokat egy `std::vector` típusú tárolóban!

c) A program számítsa ki az osztályzatok súlyozott átlagát, majd másolja át a súlyozott átlag feletti adatokat egy másik `std::vector` tárolóba! ($A = \frac{\sum_i (jegy_i \cdot kredit_i)}{\sum_i kredit_i}$)

d) Írja ki a program a szabványos kimenetre az átlag feletti adatokat tartalmazó tároló tartalmát az a) feladatban definiált kiíró operátor segítségével!

```
#include <iostream>
#include <vector>

struct Adat {
    int jegy, kredit;
};

std::istream & operator>>(std::istream & is, Adat & a) {
    is >> a.jegy >> a.kredit;
    return is;
}

std::ostream & operator<<(std::ostream & os, const Adat & a) {
    os << a.jegy << ' ' << a.kredit;
    return os;
}

int main() {
    std::vector<Adat> be, jok;

    Adat a;
    while ((std::cin >> a) && a.jegy != 0) {
        be.push_back(a);
    }

    double sum = 0, sum_kr = 0;
    for (size_t i = 0; i < be.size(); i++) {
        sum += be[i].jegy*be[i].kredit;
        sum_kr += be[i].kredit;
    }
    double atlag = sum / sum_kr;

    for (size_t i = 0; i < be.size(); i++)
        if (be[i].jegy>atalag)
            jok.push_back(be[i]);

    for (size_t i = 0; i < jok.size(); i++)
        std::cout << jok[i] << std::endl;

    return 0;
}
```