

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 20 perc

1. Készíts generikus függvényt, amely kiszámítja a kapott tömb elemeinek összegét! Írj kódrészletet, melyben bemutatom a függvény használatát! (5p)

```
template<typename T>
T sum(T*t, int n) {
    T s = 0;
    for (int i = 0; i < n; i++)
        s += t[i];
    return s;
}

#include <iostream>

int main() {
    double tomb[5] = { 1.1, 2.2, 3.3, 4.4, 5.5 };
    std::cout << sum(tomb, 5) << std::endl;
    return 0;
}
```

2. Hozz létre irodaszer osztályt! Minden irodaszernek van mennyisége és ára. Az ezeket az adatokat tároló tagváltozók legyenek kívülről elérhetetlenek, viszont legyenek lekérdezhetők és beállíthatók (tagfüggvényekkel)! Legyen konstruktor, amely inicializálja a tagváltozókat! Legyen tagfüggvény, amely kiírja a tagváltozókat!

Származtass papír és ceruza osztályt! A papír rendelkezik tömeggel, a ceruza pedig egy logikai értékkel, amely azt mondja meg, hogy ki van-e hegyezve. Mindkettőhöz készíts konstruktort, amely az ősből örökölt és a saját tagváltozókat egyaránt beállítja! Készíts kiíró függvényeket is, amelyek kiírják a tagváltozók értékét, az ősből örökölteket is! (Ehhez felhasználhatod az ős osztály kiíró függvényét.) (6p)

```
#include <iostream>

class Irodaszer {
    int menny, ar;
public:
    Irodaszer(int menny, int ar) :menny(menny), ar(ar) {}
    void setMenny(int m) { menny = m; }
    void setAr(int a) { ar = a; }
    int getMenny()const { return menny; }
    int getAr()const { return ar; }
    void print()const { std::cout << menny << ' ' << ar; }
};

class Papir : public Irodaszer {
    int tomeg;
public:
    Papir(int menny, int ar, int tomeg):Irodaszer(menny,ar),tomeg(tomeg){}
    void print()const { Irodaszer::print(); std::cout << ' ' << tomeg; }
};

class Ceruza : public Irodaszer {
    bool ki_van_hegyezve;
public:
    Ceruza(int menny, int ar, bool kvh) :Irodaszer(menny, ar), ki_van_hegyezve(kvh) {}
    void print()const { Irodaszer::print(); std::cout << ' ' << ki_van_hegyezve; }
};
```