

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 15 perc

1. Definiálj két, double típusú elemeket tároló std::vector-t: a-t és w-t! Ezután tételezd fel, hogy valaki feltölti ezt a két vektort (nem kell megvalósítano, írhatod helyette, hogy ...). Számítsd ki az 'a' tömb 'w' súlyok szerinti súlyozott átlagát! ($sa = (\sum a_i \cdot w_i) / \sum w_i$) Ha a számításhoz STL algoritmusokat használsz, és ezt helyesen teszed, +1p pónuszt kapsz algoritmusonként. Képezd az a és w tömbök összegét egy ossz nevű vektorba! Ehhez is használhatsz STL algoritmust 1 bónuszpontért. Ne feledd, hogy az STL algoritmus megfelelő méretű eredmény vektort vár! (4p)

```
std::vector<double> a, w;
double w_a = std::inner_product(a.begin(), a.end(), w.begin(), 0.0) /
std::accumulate(w.begin(), w.end(), 0.0);
std::vector<double> eredmeny(a.size());
std::transform(a.begin(), a.end(), w.begin(), eredmeny.begin(), std::plus<double>());
```

2. Készíts generikus függvényt, amely egy tömböt kap paraméterként, és visszaadja a tömb elemeinek összegét! Készíts specializációt C sztringekre mutató pointerok tömbje esetére, amely a leghosszabb sztring címét adja vissza! Írj kódrészletet, amelyben kipróbálsz a függvényt double tömbre és C sztring-pointer tömbre is! (A kipróbálásnál felhasználhatod a következő sztringpointer tömböt.) (6p)

```
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };

template<typename T> // 1p
T osszead(const T t[], int n) { // 1p
    T sum = T();
    for (int i = 0; i < n; i++) // alg: 1p
        sum += t[i];
    return sum;
}

template<>
const char* osszead(const char* const t[], int n) { // specializáció 1p
    int leghosszabbIndex = 0;
    int leghosszabbHossz = strlen(t[0]);
    for (int i = 1; i < n; i++)
        if (strlen(t[i]) > leghosszabbHossz) { // alg 1p
            leghosszabbHossz = strlen(t[i]);
            leghosszabbIndex = i;
        }
    return t[leghosszabbIndex];
}

double dt[6] = { 10, -8, 0, 99, -11, 3 };
const char* sptomb[4] = { "Dénes", "Aladár", "Cecil", "Béla" };
std::cout << osszead(dt, 6) << std::endl; // ha mindkét példa jó: 1p
std::cout << osszead(sptomb, 4) << std::endl;
```