

4. labor kis ZH, 2017. április 13., csütörtök 10-12

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 20 perc

1. Ebben a feladatban a szövegek tárolására `std::string`-et használj! Olvass be három sor szöveget a szabványos bemenetről (`std::getline()`)! Az első sorban egy ember neve van (vezetéknév és egy vagy több keresztnév), a második sorban, hogy „ferfi” vagy „no”, a harmadik sorban, hogy „igen” vagy „nem”. Utóbbi válasz arra vonatkozik, hogy házas-e vagy sem. Hozz létre egy sztringet amelyben egy levél megszólítása van a következő formában „Tisztelt VEZETÉKNÉV MEGSZÓLÍTÁS!”. A vezetéknévet a teljes névből kell kinyerni, a megszólítás ür/asszony/kisasszony lehet, a második és harmadik sorban megadott adatok alapján. Pl. „Gipsz Jakab, ferfi, igen” esetén „Tisztelt Gipsz úr!”, „Álmos Elvira Eulália, no, nem” esetén „Tisztelt Álmos kisasszony!”. Írd is ki a megszólítást tartalmazó sztringet! (max. 6p)

```
#include <string>
#include <iostream>
```

```
int main() {
    using namespace std;
    string s1, s2, s3;
    getline(cin, s1);
    getline(cin, s2);
    getline(cin, s3);
    string megszolitas;
    megszolitas = "Tisztelt "
        + s1.substr(0, s1.find(' '))
        + (s2 == "ferfi"
            ? " ur!"
            : (s3 == "igen" ? " asszony!" : " kisasszony!"));
    cout << megszolitas << endl;
    return 0;
}
```

2. A feladatod, hogy olvass be hőmérséklet értékeket, melyeket két nap alatt mértek. Mindkét nap ugyanannyiszor és ugyanakkor mértek hőmérsékletet, de előre nem tudjuk, hogy hányszor. A hőmérséklet értékeket a szabványos bemeneten kapjuk, szóközzel elválasztva, °C-ban értendő valós számként. A két nap adatait egy -300-nál kisebb érték választja el, ami nem tárolandó. Pl. bemenet: 8.1 12.2 16.5 14.2 11.1 -333 7.3 10.4 17.5 16.8 13.2. Az egyes részfeladatokra csak akkor jár pont, ha pontosan a következő módon oldod meg. (max. 10p)

- a) Olvasd be a két nap adatait az `std::cin`-ről két `std::vector` típusú tárolóba, a tárolók `double` típusú értékeket tároljanak!
- b) A két nap hőmérséklet értékeit tartalmazó vector-okat add össze `std::transform` algoritmus segítségével, az eredmény egy harmadik vector-ba kerüljön. Ehhez vagy az `std::plus<double>()` műveletet használd, vagy írd saját összeadó függvényt/funktort.
- c) Felezd el az összeg vector minden elemét az `std::transform` segítségével és saját felező függvényt/funktor segítségével!
- d) Írd ki az így kapott vector-t!
- e) Tetszőleges módszerrel számítsd ki az így kapott vector-ban lévő hőmérsékletek átlagát (egy db. átlaghőmérsékletet kell kapnod), és írd ki! Ha a számítást helyesen végzed el az `std::accumulate` segítségével vagy iterátorral, +1 bónuszpontot kapsz.

```
#include <vector>
#include <iostream>
#include <algorithm>
#include <numeric>

double felez(double a) { return a / 2; }

int main() {
    using namespace std;
    vector<double> nap1, nap2;
    double adat;
    cin >> adat;
    while (adat > -300) {
        nap1.push_back(adat);
        cin >> adat;
    }
    nap2.resize(nap1.size());
    for (size_t i = 0; i < nap2.size(); i++)
        cin >> nap2[i];
    vector<double> osszeg(nap1.size());
    transform(nap1.begin(), nap1.end(), nap2.begin(), osszeg.begin(), plus<double>());
    transform(osszeg.begin(), osszeg.end(), osszeg.begin(), felez);
    for (size_t i = 0; i < osszeg.size(); i++)
        cout << osszeg[i] << ' ';
    cout << '\n' << accumulate(osszeg.begin(), osszeg.end(), 0.0) / osszeg.size() << endl;
    return 0;
}

// iterátorral kiírás
for (vector<double>::iterator it = osszeg.begin(); it != osszeg.end(); it++)
    cout << *it << ' ';
```