

FEJLETT PROGRAMOZÁSI NYELVEK, 2009

3. GYAKORLAT – Bevezetés a standard könyvtárba

Cél:

- szöveges állományok olvasása és írása
- standard tárolók: `vector<T>`, `string`
- standard algoritmusok: `sort`, `unique`, `equal`, `binary_search`

I. Szöveges állományok feldolgozása

Szöveges állomány olvasása:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

int main() {
    vector<string> v;
    ifstream ifs("szoveg.txt");
    string word;
    while( ifs >> word ){
        v.push_back( word );
    }
    //...
    return 0;
}
```

Szöveges állomány írása:

```
ofstream ofs("out.txt");
ofs<<"Ez egy szam: "<<(2+3)<<endl;
```

II. Feladatok

1. A standard könyvtár használata (`vector`, `sort`, `unique`, `string`)

Adott egy szöveges állomány, amely szóközzel elválasztott egész számokat tartalmaz. Készítsen programot, amelynek bemenete a fenti állomány, kimenete pedig a számsorozatban szereplő számok növekvő sorrendben, úgy, hogy minden szám csak egyetlen egyszer szerepeljen.

Megoldás:

1. lépés: olvassa be a számokat elhelyezve egy `vector` dinamikus tömbben

2. lépés: rendezze a számokat a `sort` algoritmussal (`#include <algorithm>`)

3. GYAKORLAT – Bevezetés a standard könyvtárba

3. lépés: hívja meg a `unique` algoritmust majd írassa ki az eredményt

```
vector<int>::iterator itend, it;
...
itend = unique(v.begin(), v.end());

for( it = v.begin(); it != itend; ++it )
    cout<<*it<<endl;
```

2. Palindroma - A standard könyvtár használata (`string`, `equal`)

Töltse le a következő angol szótárt: <http://www.ms.sapientia.ro/~manyi/dict.txt>

Készítsen programot, amely a fenti szótárból kiírja az összes palindromát. A szó palindroma tulajdonságának eldöntéséhez használja az `equal` algoritmust!

3. Szótár - A standard könyvtár használata (`vector`, `string`, `sort`, `binary_search`)

Használva a fenti szótárt, készítsen egy egyszerű helyesírás ellenőrző programot.

1. lépés: Készítsen egy osztályt, amely egy szótárt ábrázol és implementálja a legszükségesebb műveleteket. A szótárt tárolja dinamikus tömbben, majd rendezze alfabetikusan. A keresést végezze bináris kereséssel.

2. lépés: Készítsen programot a következő funkcionalitással:

Indításkor megadjuk a szótárt tartalmazó állomány nevét, majd folyamatosan parancsokat olvasunk egészen a kilépés parancsig.

Parancsok:

-f filename: kapcsolóval ellenőrzi a kapcsoló utáni állomány helyességét, egyszerű szóellenőrzést végezve az állomány minden egyes szavára
-w word: a kapcsoló utáni szót ellenőrzi
-t text: a kapcsoló utáni szöveg helyességét ellenőrzi
-q: kilépés