

Programozás Beadandó

Tanuló adatai

Név: Berezvai Dániel

ETR-azonosító: BEDTACI.ELTE

E-mail: 3ice@3ice.hu

Kurzuskód: IP-08PAEG/18

Gyakorlatvezető neve: Menyhárt László

Feladatsorszám: 35

Feladat szövege: A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet. **Készíts programot, amely megad egy települést, amely hőmérsékletátlaga hidegebb egy tetszőleges másik település leghidegebb napjánál!**

Megjegyzés: Ennek a feladatnak ritkán van megoldása, mert az átlagok nem lehetnek hidegebbek, mint a legalacsonyabb hőmérsékletek. Sivatagi települések és északi sarkon levő települések összehasonításának pedig nincs értelme.

Felhasználói dokumentáció

A program célja

A beolvasott időjárás előrejelzések alapján keresni egy olyan települést, amelynek előre jelzett hőmérsékleteinek átlaga a megadott település várható leghidegebb napjánál alacsonyabb. *(Akár több száz fájl kezelése, vektorok és asszociatív tömbök használata, kis fájl méret, telepítő.)*

Rendszerkövetelmények

- 53 kb HDD
- 1 MB RAM
- Egy- vagy többmagos x86 vagy x64 processzor
- Billentyűzet

(Képernyő, egér nem feltétlen szükséges. Billentyűzet nélkül is futtatható, automatizált környezetben.)

Támogatott operációs rendszerek

- ✓ Windows
- ✓ UNIX (Linux, Unix-Like, *NIX)
- ✓ Mac
- ✓ Minden más operációs rendszer, ahol c++ kódot fordítani lehet. (Portabilitási problémák felléphetnek.)

Telepítés (Windows)

Futtassa a BEDTACI_Bead.exe fájlt és kövesse a képernyőn megjelenő utasításokat.

(A telepítő szkriptemet az INNO Setup Compiler program segítségével fordítottam.

Van hozzá uninstaller is.)

A folyamat során ön dönti el, hova kerül telepítésre a program.

Telepítés (Egyéb)

Más operációs rendszerhez nincs telepítő, a programot a mellékelt C++ forráskódból kell fordítani, majd elhelyezni a kívánt elérési úton.

Használat (Windows)

Telepítés után a program a start menüből vagy parancssorból indítható.

Start > Programok > idojaras

A PATH környezeti változóba felveheti a program elérési útját, így bárholnan meghívható lesz.

A program automatizálásra lett tervezve (nincs konzolos interfésze), ezért érdemesebb batch scriptből futtatni. (Részletek a következő oldalon.)

Használat (Egyéb operációs rendszereken)

Sok sikert!

Bemenet / Kimenet

A program több bemeneti fájlt képes beolvasni egy futás során, ezért érdemes minden nap előrejelzését külön fájlban tárolni. (Természetesen egy nagy fájlból is be tudja olvasni az összes napot.)

Hasonló módon több kimenetet is kérhetünk, egy vagy több fájlba.

Bemeneti fájl formátuma (települések és hőmérséklet előrejelzések)

Minden sor egy nap, a települések és hőmérsékletek egymás mögött, szóközzel vannak elválasztva. A bemeneti fájlok kiterjesztése általában: "*.be".

A település neve maximum 255 karakter hosszú és nem lehet benne szóköz.

Példa

Budapest 5 Kecskemét 5 Pécs 6 Békéscsaba 4 Miskolc 6 Szeged 5 Székesfehérvár 5 Győr 8 Debrecen 5 Eger 4 Szolnok 5 Tatabánya 6 Salgótarján 5 Kaposvár 6 Nyíregyháza 6 Szekszárd 6 Szombathely 8 Veszprém 5 Zalaegerszeg 6

Kimenetkérő fájl formátuma (településnevek)

A program a megadott kimenetkérő fájl(oka)t beolvassa, majd felülírja a kért kimenettel. A települések neveit új sorba (soronként egy település) vagy szóközzel elválasztva kell írni. (A program kevert fájlformátumot is elfogad.) A kimeneti fájlok általában "*.ki" kiterjesztéssel rendelkeznek.

Itt is érvényes a 255 karakter limit és a szóköztílalom településnevekre.

Példák

Kimenetkérés.ki tartalma:	Kapott kimenet:
Pécs Tatabánya Veszprém Budapest Szombathely	Budapest Budapest Békéscsaba Békéscsaba Szombathely

Parancssoros meghívás

A programot az idojaras.exe paranccsal lehet meghívni. Kötelező parancssori kapcsolók a -be és a -ki, ezek után lehet felsorolni a bemeneti illetve kimeneti fájlok neveit. A kapcsolók sorrendje tetszőleges.

Példák

(atl.bat) idojaras.exe -be 1.be 2.be 3.be 4.be 5.be 6.be 7.be -ki atl.ki

(vilag.bat) idojaras.exe -be vilag.be -ki vilag.ki

(min.bat) idojaras.exe -be min.be -ki min.ki

(A fent használt összes bemeneti és kimenetkérő fájlt mellékeltem tesztelésre.

További tesztesetek lentebb.)

Hibakódok

A program futása során a következő hibákkal találkozhat:

0 (0x0) ERROR_SUCCESS

Nincs hiba, a program sikeresen lefutott.

1 (0x1) ERROR_INVALID_REQUEST

Helytelen kimenetkérési fájlformátum. Győződjön meg arról, hogy a kért település(ek) nevét helyesen írta!

2 (0x2) ERROR_MISSING_PARAM

Egy kötelező paraméter vagy kapcsoló nincs megadva. Kötelező parancssori kapcsolók a -be és a -ki.

3 (0x3) ERROR_MISSING_FILE

A bemenetként vagy kimenetként megadott fájl nem létezik.

13 (0xD) ERROR_INVALID_DATA

Helytelen bemeneti fájlformátum. Ellenőrizze, hogy a bemenet(ek) formátuma megfelelő-e! (településnév hőmérséklet településnév hőmérséklet...)

18 (0x12) ERROR_NO_MORE_FILES

Túl kevés bemeneti vagy kimeneti fájl. Legalább egy be- és egy kimeneti fájlnak jelen kell lennie.

Egyéb

Ha ezeken kívül más hibával találkozik, az a program helytelen működéséből adódik és pontlevonást érdemlek érte.

Fejlesztői dokumentáció

Specifikáció

Be

N, M : Egész (Nem kell megadni, a program kitalálja.)

X : Tömb[1.. N , 1.. M : Szöveg (településnév), Valós(hőmérséklet)]

Ef

$N, M > 0$

$\forall i (1 \leq i \leq N) \forall x \in X[i]: x \in \mathbb{R}$

Ki

Településnév: Szöveg

Uf

$\forall x \in X[1..N]: \forall y \in X[x][1..M]: y < k \mid k = \text{átlag}(X[\text{Településnév}][1 \dots M])$

Algoritmusok

$$\text{átlag}(X[1..N]) = \frac{\text{sum}(1..N)}{N}$$

$$\text{sum}(X[1..N]) = \sum_{i=1}^N X[i]$$

Fordítóprogram, IDE

MinGW GCC-vel fordítunk, Code::Blocks-szal fejlesztünk.

Ideológiám

Lehető legkevesebb függőség (dependency). Csak az `#include <stdio.h>`-val akartam dolgozni, de később jött a string, vector, map is.

Lehető legkisebb fájlméret (Produce debugging symbols -g kapcsoló eltávolítása, Strip all symbols from binary -s kapcsoló hozzáadása)

Lehető leggyorsabb futásidő (Optimize fully for speed -O3 kapcsoló)

Változók

```
//Struct belsejeben vector tarolja a napi előrejelzéseket, minimum, atlag
ebbol lesz kiszámolva.
struct telepules{
    vector<float> homersekletek;
    float minimum;
    float atlag;
};

//Asszociativ tomb, majd így kell elerni a belsejet:
//adatIterator->second.homersekletek[j]
map<string, telepules> adat;
map<string, telepules>::iterator adatIterator;

//Nehany hibakod es -nev hivatalos (0, 13, 18), a tobbi saját gyartmany.
const int ERROR_SUCCESS = 0;
const int ERROR_INVALID_REQUEST = 1;
const int ERROR_MISSING_PARAM = 2;
const int ERROR_MISSING_FILE = 3;
const int ERROR_INVALID_DATA = 13;
const int ERROR_NO_MORE_FILES = 18;
```

```

//argc: parameterek szama, argv: atalakitom string tombbe (args) egyszeru
kezelhetosegert.
int argc
char *argv[]
string args[argc-1]

//Szamlalok
int i, j, x, index

//Ideiglenes valtozok
char tmpChr[255]
string tmpStr
float tmpFlt
telepules tmpTelepules
int/string devnull
float tmpFlt, sum, min, avg

//Tovabbi szamlalok
int bemenetekSzama, kimenetekSzama
int bemenetIndex, kimenetIndex
int bemenetFelsoIndex, kimenetFelsoIndex

//Parameterek kettevalogatas utan
string bemenetek[bemenetekSzama], kimenetek[kimenetekSzama]

```

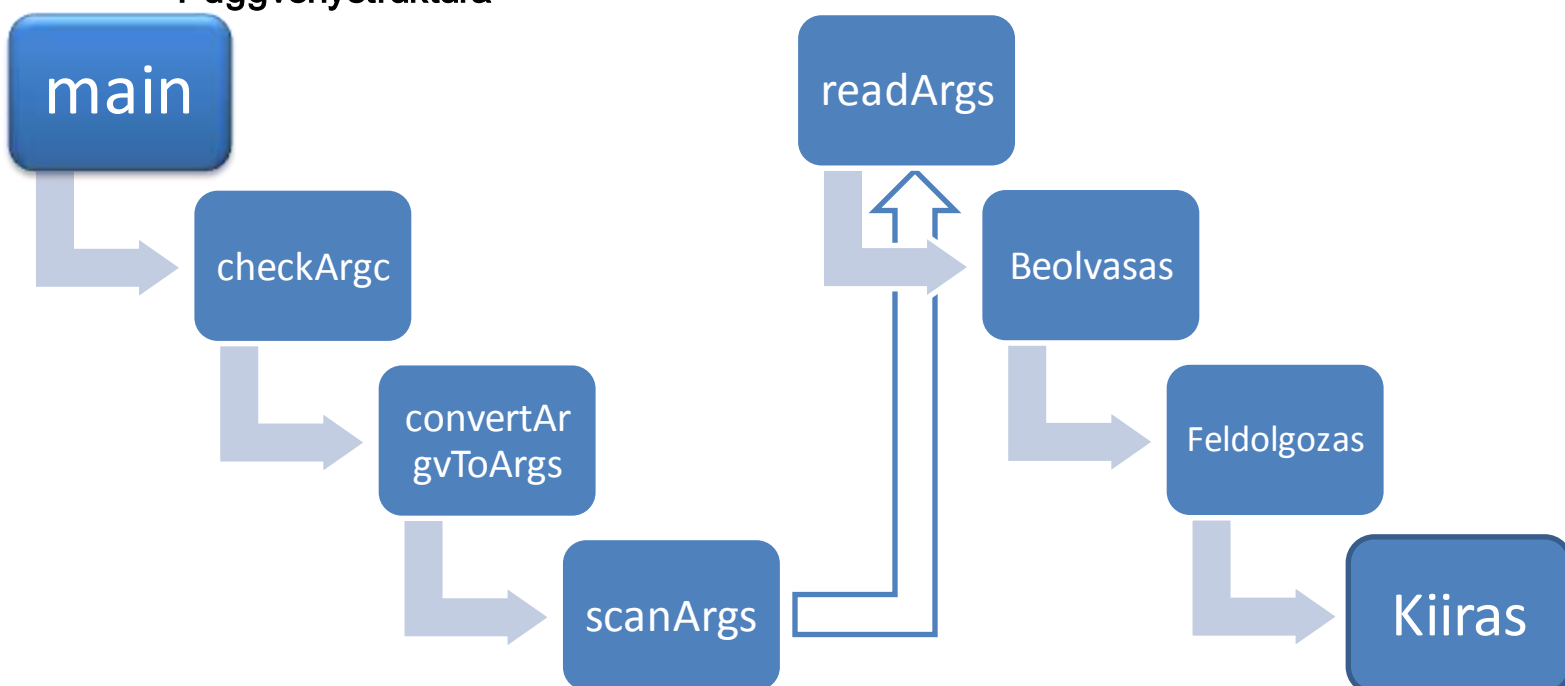
Includes

```
#include <stdio.h> //C fájl I/O
```

Ez volt az egyetlen tervezett include, hogy csak pár kilobyte legyen a program a megszokott 500 kb helyett. (C++ istream például fél MB, míg C stdio mindössze 5 kb lefordítva.)

```
string, vector, map
```

Függvénystruktúra



Algoritmusok, alkalmazott programozási tételek

Minimumkeresés, Sorozatösszeg (átlagot számoltam)

Forráskód

Nyisd meg a main.cpp fájlt! (Vagy az idojaras.cbp Code::Blocks projektet.

Not good enough? Well, if you insist...

Forráskód

(100 soros részlet az 500 soros programból. A maradék 400 sor preprocesszor-komment és debug output. A program úgy fut, mintha narrátor mesélné végig.)

```
//Berezvai Dániel
//BEDTACI.ELTE
#include <stdio.h>, <string>, <vector>, <map>
using namespace std;
//Struct belsejében vektor tarolja a napi előrejelzéseket, minimum, átlag
//ebbol lesz kiszámolva.
struct telepules{
    vector<float> homersekletek;
    float minimum;
    float atlag;
};

//Asszociatív tömb, majd így kell elérni a belsejét:
//adatIterator->second.homersekletek[j]
    map<string, telepules> adat;
    map<string, telepules>::iterator adatIterator;

//Néhány hibakód és -név hivatalos (0, 13, 18), a többi saját gyártmány.
const int ERROR_SUCCESS = 0;
const int ERROR_INVALID_REQUEST = 1;
const int ERROR_MISSING_PARAM = 2;
const int ERROR_MISSING_FILE = 3;
const int ERROR_INVALID_DATA = 13;
const int ERROR_NO_MORE_FILES = 18;

//Így az exit funkció használható fölösleges 100 kb-nyi include nélkül
extern "C" void exit(int);

void checkArgc(int argc){
    if(argc<5){exit(ERROR_MISSING_PARAM);}
}

void convertArgvToArgs(int argc, char *argv[], string args[]){
    for(int i=1;i<argc;i++){args[i-1]=argv[i];}
}
```

```

void scanArgs(int argc, string args[], int &bemenetekSzama, int
&kimenetekSzama, int &bemenetIndex, int &kimenetIndex, int
&bemenetFelsőIndex, int &kimenetFelsőIndex){
    int index=0;
    while(index<argc && args[index]!="-be"){index++;}
    bemenetIndex=++index;//Ez az indexet es a bemenetIndexet is növeli.
    while(index<argc && args[index]!="-ki"){index++;}
    bemenetekSzama=index-bemenetIndex;
    if(bemenetekSzama<1){exit(ERROR_NO_MORE_FILES);}
    if(bemenetekSzama!=index-1){index=0;}
    while(index<argc && args[index]!="-ki"){index++;}
    kimenetIndex=++index;//Ez az indexet es a kimenetIndexet is növeli.
    while(index<argc && args[index]!="-be"){index++;}
    kimenetekSzama=index-kimenetIndex;
    if(kimenetekSzama<1){exit(ERROR_NO_MORE_FILES);}
    bemenetFelsőIndex=bemenetIndex+bemenetekSzama;
    kimenetFelsőIndex=kimenetIndex+kimenetekSzama;
}

void readArgs(string bemenetek[], string kimenetek[], string args[], int
bemenetIndex, int bemenetFelsőIndex, int kimenetIndex, int
kimenetFelsőIndex){
    int j=0;
    for(int
i=bemenetIndex;i<bemenetFelsőIndex;i++){bemenetek[j]=args[i];j++;}
    j=0;
    for(int
i=kimenetIndex;i<kimenetFelsőIndex;i++){kimenetek[j]=args[i];j++;}
}

void Beolvasas(string bemenetek[], int bemenetekSzama){
    char tmpChr[255];
    string tmpStr;
    float tmpFlt;
    telepules tmpTelepules;
    for(int i=0;i<bemenetekSzama;i++){
        FILE *inputFile=fopen(bemenetek[i].c_str(), "r");
        if(inputFile==NULL){exit(ERROR_MISSING_FILE);}
        while(fscanf(inputFile, "%s", tmpChr)!=EOF){//régí C típusú
file I/O gyorsabb, mint a C++
            tmpStr=tmpChr;
            if(fscanf(inputFile, "%f", &tmpFlt)==EOF || tmpFlt>500 ||
tmpFlt<-500 || (tmpFlt<0.01 && tmpFlt>-0.01 &&
tmpFlt!=0)){exit(ERROR_INVALID_DATA);}
            adat[tmpStr].homersekletek.push_back(tmpFlt);
        }
        fclose(inputFile);
    }
}

void Feldolgozas(){
    float tmpFlt, sum, min;
    for(adatIterator=adat.begin();adatIterator!=adat.end();adatIterator++){
        sum=0;
        min=2147483647;//Ennel biztos csak kisebb lesz.
        for(unsigned int j=0;j<adatIterator
->second.homersekletek.size();j++){
            tmpFlt=adatIterator->second.homersekletek[j];
            sum+=tmpFlt;
            if(tmpFlt<min){min=tmpFlt;}
            adatIterator->second.minimum=min;
            adatIterator->second.atlag=sum/adatIterator
->second.homersekletek.size();

```

```

    }
}

void Kiiras(string kimenetek[], int kimenetekSzama){
    char tmpChr[255];
    string tmpStr;
    float avg;
    for(int i=0;i<kimenetekSzama;i++){
        FILE *outputFile=fopen(kimenetek[i].c_str(), "rw");
        if(outputFile==NULL){exit(ERROR_MISSING_FILE);}
        while(fscanf(outputFile, "%s", tmpChr)!=EOF){//régí C típusú
file I/O gyorsabb, mint a C++
            tmpStr=tmpChr;
            adatIterator=adat.find(tmpStr);
            if(adatIterator==adat.end()){exit(ERROR_INVALID_REQUEST);}
            avg=adatIterator->second.atlag;
            adatIterator=adat.begin();
            while(adatIterator!=adat.end() && avg>adatIterator
->second.minimum){
                adatIterator++;
            }
            if(adatIterator!=adat.end() && avg<adatIterator
->second.minimum){...}
        }
        fclose(outputFile);
    }
}

//argc: parameterek szama, argv: atalakitom string tombbe (args) egyszeru
kezelhetosegert.
int main(int argc, char *argv[]){
    checkArgc(argc);
    string args[argc-1];
    convertArgvToArgs(argc, argv, args);
    int bemenetekSzama=0;    int kimenetekSzama=0;
    int bemenetIndex=0;      int kimenetIndex=0;
    int bemenetFelsőIndex=0; int kimenetFelsőIndex=0;
    scanArgs(argc-1, args, bemenetekSzama, kimenetekSzama, bemenetIndex,
kimenetIndex, bemenetFelsőIndex, kimenetFelsőIndex);
    string bemenetek[bemenetekSzama];
    string kimenetek[kimenetekSzama];
    readArgs(bemenetek, kimenetek, args, bemenetIndex, bemenetFelsőIndex,
kimenetIndex, kimenetFelsőIndex);
    Beolvasas(bemenetek, bemenetekSzama);
    Feldolgozas();
    Kiiras(kimenetek, kimenetekSzama);
    return ERROR_SUCCESS;
}

```

Tesztelés

Valós adatokkal kezdtem (Magyarország heti időjárás előrejelzése). Majd generáltam 1000000 véletlen számot -40 .. 0 .. +50 körül. A városok listáját egy "top 1000 cities by population" listáról loptam.

Érvényes tesztesetek

Egy város: Mindig az adott bemenet az eredmény, mert inkább az egyenlőséget engedjük meg, mint hogy NULL-t vagy N/A-t dobnánk a felhasználó arcába.

egy.bat (egy.be egy.ki)

Minimalista: Egy nap, két város.

min.bat (min.be min.ki)

Átlagos: 7 nap, 7 fájl, 19 város, 19 kimenet.

atl.bat (1.be .. 7.be atl.ki)

Maximalista: 7 nap, bemenet egy fájlban, 1000 város, 12 kimenet.

vilag.bat (vilag.be vilag.ki)

Maximalista2: 1000 nap, 69 város, 10 kimenet.

max.bat (max.be max.ki)

Kegyetlen: 10000 nap 10000 város 10000 kimenet.

Ne essünk túlzásba... (Ennyi napra ma még nem is tudunk elég gyorsan időjárást előre jelezni)

Felcserélt paraméterek: Kimenettel kezdjük a paraméterek megadását. Ez érvényes.
csere.bat

Érvénytelen tesztesetek

Üres bemeneti fájl, vagy a kimenetben olyan várost kérünk, ami nincs benne egyetlen bemenetben sem: ERROR_INVALID_REQUEST

nemstimmel.bat (nemstimmel.be nemstimmel.ki)

Érvénytelen paraméter megadás, például -be -ki hello.ki world.ki:

ERROR_NO_MORE_FILES

badparam.bat (hello.ki world.ki)

Érvénytelen paraméter megadás, például -be world.be: ERROR_MISSING_PARAM
missingparam.bat (world.be)

"Véletlenül" kifelejtünk egy számot a bemenetből, így két településnév kerül egymás mellé: ERROR_INVALID_DATA
rosszbemenet.bat (rosszbemenet.be)

Vajon mit felejtettem ki? (*"Saját magunk programját sosem tudjuk igazságosan tesztelni."*)

Fejlesztési lehetőségek

- Konzolos (cin, cout) használat lehetővé tétele, ha nincsenek paraméterként fájlok megadva. (Bár ez sosem volt célom, hiánya nemtetszést válthat ki.)
- Bírjon 10000 napos előrejelzést 10000 városra (Future proof)
- Grafikus kezelőfelület
- Térképes integráció színkódolással és animációval
- Olyan feladat megoldása, aminek kézzelfogható haszna is van. (Például a leghidegebb átlaghőmérsékletű város leghidegebb napját sokkal szívesebben számoltam volna ki, mint ezt.)
- És fizetést mikor kapok? (Tudom, jövőre lesz ösztöndíj.)