Feladat

Madarak életének kutatásával foglalkozó szakemberek n különböző településen m különböző madárfaj előfordulását tanulmányozzák. Egy adott időszakban megszámolták, hogy az egyes településen egy madárfajnak hány egyedével találkoztak. Melyik az a madárfaj, amelyik a leggyakrabban (azaz a legtöbb településen) fordult elő?

# Specifikáció

Tegyük fel, hogy madárfajokra lebontva rendelkezésre állnak az előfordulások településenként. Ki kell számolni, hogy madárfajonként hány településen fordult elő az adott madárfaj, majd az előfordulások közül kiválasztani a legnagyobbat. Ez tehát egy *maximum-kiválasztás*.

A bemenő adatokat fogjuk fel tehát úgy, hogy adott egy -es mátrix, ahol a települések száma, a madárfajok száma, és a mátrix i-edik sorának j-edik eleme tartalmazza az i-edik madárfaj j-edik településen való előfordulását.

ahol a egy olyan függvény lesz, ami megszámolja madárfajonként a településenkénti előfordulásokat. Mivel a bemenő mátrix nem lehet üres, ezért biztosan lesz nem-üres bemenete a függvénynek.

Ez alapján:

és

# Visszavezetés és algoritmus

## Külső tétel:

maximum-kiválasztás

|  |  |  |
| --- | --- | --- |
|  | ~ |  |
|  | ~ |  |
|  | ~ |  |
|  | ~ |  |
|  | ~ |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | |  | |  |
|  | | | | | | |  |
|  | |  |  | | |  |  |
|  | |  | |  | | |  |

## Belső tétel:

számlálás

|  |  |  |
| --- | --- | --- |
|  | ~ |  |
|  | ~ |  |
|  | ~ |  |
|  | ~ |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | |  | |  |
|  |  | | |  | |  | |  |
|  |  | | | | |  | |  |
|  | | | | | | | |  |
|  | |  |  | | | |  |  |
|  | |  | | |  | | |  |

# Implementáció

A programot C++ nyelven valósítottuk meg.

## Adattípusok

A települések számát egy változóba, a madárfajok számát pedig egy változóba kérjük be.

A feladat megfogalmazásakor definiált „listák tömbjét” egy „kesztyűmátrixszal”, azaz egy olyan std::vectorral fogjuk reprezentálni, aminek elemei maguk is vectorok.

Az adatokat az változóba olvassuk be, melynek típusa tehát vector<vector<int> >. Az előfordulások természetes számok lehetnek, ezt a feltételt a beolvasás során ellenőrizzük.

A beolvasás során a soron következő előfordulást a v[i][j] indexszel szúrjuk a megfelelő madárfajt jelképező vector végére.

A struktogrambeli nem más, mint az mérete, azaz az v.size(), az egyes sorozathossz pedig az i-edik sorozatra az v[i].size() kifejezés értékével fogalmazható meg.

A tömbindexelést minden esetben 0-tól végezzük.

## Adatok beolvasása

Először a települések és a madárfajok számát kérjük be, ebből felépítünk egy v-vektort, majd ennek adatai alapján fel tudjuk építeni az előfordulások „mátrixát” is.

### Település- és madárfajszám:

Ennél a bekérésénél a települések és a madárfajok számát előfordulását kérjük be. A felhasználó tetszőlegesen nagy számot megadhat, amíg az nagyobb nullánál, a beolvasás folyamatát hibásan megadott értékkel szakíthatja meg.

input:

5

Ekkor ez a szám a települések, illetve a madárfajok száma lesz, attól függően, hogy éppen melyiket kérte a program.

Billentyűzetről adhatjuk meg. Amíg megfelelő adatot olvasunk, addig azt elmenti a megfelelő változóba, amennyiben hibázunk, azt jelzi a program, majd újra kéri az adatot.

Hibalehetőségek:

* Nem szám
* Nem pozitív egész szám

### Előfordulások:

Az előfordulások bekérésénél az aktuális madárfaj előfordulását kérjük be. A felhasználó tetszőlegesen sok előfordulást megadhat, a beolvasás folyamatát hibásan megadott értékkel szakíthatja meg.

input:

5

Ekkor ez a szám az i-edik madárfaj j-edik településen való előfordulása lesz.

Az inputot billentyűzetről adhatjuk meg. Amíg megfelelő adatot olvasunk, addig azt folyamatosan beszúrja a vektorba, amennyiben hibázunk, azt jelzi a program, majd újra kéri az adatot.

Hibalehetőségek:

* Nem szám
* Nem természetes szám

## A kimenet

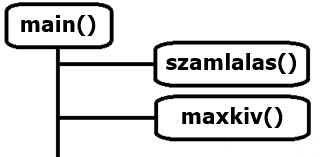
A sikeres beolvasás és feldolgozás után a feladatra adott válasz kerül kiírása a konzolra.

## A projekt felépítése

A következő modulokat használjuk:

* *main* – a main.cpp forrásfájl, a program belépési pontja (main függvény) valamint a két specifikált részfeladat implementációja található itt. A main végzi a program vezérlését, az adatok beolvasását, a tételek hívását, és az eredmény kiírását
* *iostream* – külső könyvtár, mely a *konzolos kommunikáció* eszközeit teszi elérhetővé.
* *vector* – külső könyvtár, az std::vector típushoz

Függvényhívások hierarchiája:



## Tesztelés

Az alábbiakban megadok néhány érvényes és érvénytelen tesztesetet. (A .txt fájlokban az átláthatóság kedvéért az 1-2. sor a települések, illetve madárfajok számát, a többi *m* sor pedig a madárfaj sorszámának megfelelő előfordulásokat tartalmazza, ahol egy sorban *n* előfordulás van, a településszámnak megfelelően):

* *adatbekérés* tesztelése billentyűről
  + település-/madárfajszám - *számként* *értelmezhetetlen* adatok megadása [**t0.txt**] 🡪 az első ilyenig olvasás, hibás adat újra bekérése
  + település-/madárfajszám - *string*eket tartalmazó fájl [**t1.txt**] [**f2.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
  + település-/madárfajszám - megadása *nem pozitív egész* számmal [**t2.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
  + település-/madárfajszám – *negatív* szám [**t3.txt**] [**t4.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
  + előfordulások - *számként* *értelmezhetetlen* adatok megadása [**t5.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
  + előfordulások – *string*eket tartalmazó fájl [**t6.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
  + előfordulások – *negatív* szám [**t7.txt**] 🡪 első ilyenig olvasás, hibás adat újra bekérése
* a *beolvasás* *ellenőrzése* az ellenőrző kiírással
* *maximum-kiválasztás* (külső tétel) fekete doboz tesztelése
  + az összes előfordulás 0 [**f8.txt**] 🡪 az 1. madárfaj a megoldás
  + több azonos maximum előfordulás [**f9.txt**] 🡪 az első azonos madárfaj a megoldás
* számlálás (belső tétel) fekete doboz tesztelése
  + az összes előfordulás 0 [**f8.txt**] 🡪 az összes madárfajonkénti előfordulás száma 0