

Feladat

Madarak életének kutatásával foglalkozó szakemberek n különböző településen m különböző madárfaj előfordulását tanulmányozzák. Egy adott időszakban megszámozták, hogy az egyes településen egy madárfajnak hány egyedével találkoztak.

Hány településen fordult elő mindegyik madárfaj?

Megoldási Terv:

A megfigyelések számát egy $n \times m$ -es mátrixban tároljuk.

$$A = (\text{adatok} : N^{n \times m}, db : N, \text{felcsut} : L)$$

$$ef = (\text{adatok} = \text{adatok}')$$

$$uf = \left(ef \wedge db = \sum_{i=1}^n \text{mindElofordult}(\text{adatok}, i) \quad 1 \right)$$

ahol $\text{mindElofordult} : N^{n \times m} \times [1..n] \rightarrow L$

és $\text{mindElofordult}(\text{adatok}, i) : \text{felcsut} := \bigvee_{j=1}^m \text{SEARCH} \text{adatok}[i, j] > 0$

Számlálás
$[m..n] \sim [1..n]$ $c \sim db$ $\beta(i) \sim \text{mindElofordult}(\text{adatok}, i)$

db=0	
i=1..n	
↑	mindElofordult(i)
	↓
db=db+1	SKIP

Opt. linker
$[m..n] \sim [1..m]$ $l \sim \text{felcsut}$ ind elhagyva $\beta(i) \sim \text{adatok}[i, j] > 0$

felcsut, j := 1, 1	
felcsut \wedge j ≤ m	
felcsut := adatok[i, j] > 0	
j = j + 1	

Implementáció

Adattípusok megvalósítása

A tervben szereplő mátrixot `vector<vector<int>>` -ként deklaráljuk. Mivel a vektor 0-tól indexelődik, azért a tervbeli ciklusok indextartományai a $0..n-1$ és a $0..m-1$ intervallumra módosulnak, ahol a n -re `adatok.size()` alakban, m -re pedig `adatok[i].size()` alakban hivatkozhatunk.

Bemenő adatok formája

Az adatokat be lehet olvasni egy szöveges állományból vagy meg lehet adni billentyűzetről. Indítást követően a program először megkérdezi az adatbevitel módját, majd a szöveges állományból való olvasást választva bekéri az állomány nevét. A billentyűzetről vezérelt adatbevitelt a program párbeszéd-üzemmódban irányítja, és azt megfelelő adat-ellenőrzésekkel vizsgálja.

A szöveges állomány formája kötött, arról feltesszük, hogy helyesen van kitöltve, ezért ezt külön nem ellenőrizzük. A szöveges állomány első sorában a települések száma, második sorában pedig a madárfajok száma szerepel. Utána minden sor egy szóközzel vagy tabulátorokkal elválasztott számsorozatként kell szerepeljen, a sorok végén egy-egy sorvége jellel, ahol a sorok maguk a falvakat jelképezik, a számok pedig az egyes madárfajok megfigyeléseinek darabszámát.

Példa:

```
4
3
10 14 0
5 4 3
2 2 2
0 6 9
```

Program váz

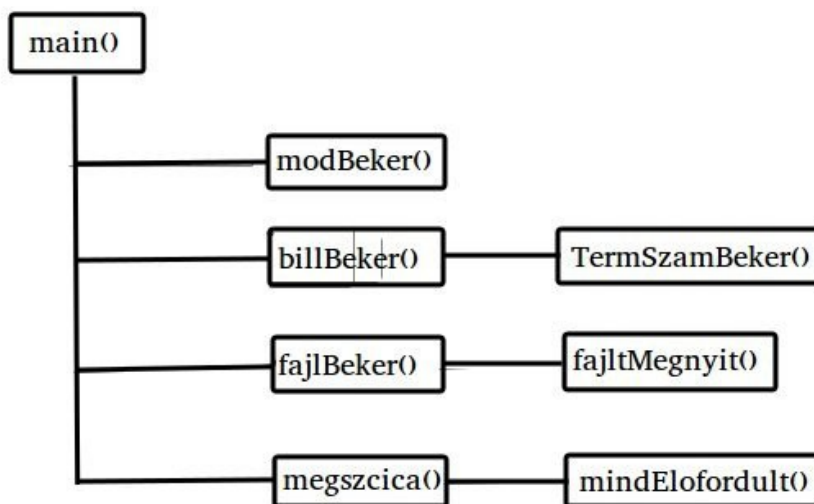
A program több állományból áll. A beolvasás csomagban (`beolvasas.h`, `beolvasas.cpp`) a `modBeker()`, a `TermSzamBeker()`, a `billBeker()`, a `fajlBeker()` és a `fajltMegnyit()` függvényeket találjuk (ezek az adatok beolvasását támogatják) , a többi függvény a `main.cpp` állományban van.

A `modBeker()` segítségével azt a felhasználói döntést olvassuk be, hogy fájlból vagy billentyűzetről történjen-e az adatok bevitele.

Billentyűzetről való adatbevitel esetén egy "b" karaktert vár a program felhasználótól. Ebben az esetben meghívódik a `billBeker()` függvény, amely a felhasználótól bekéri a települések számát, a madárfajok számát, illetve az egyes településeken az egyes madárfajok egyedeinek a számát. A `TermSzamBeker()` függvény ellenőrzi az adatok helyességét.

Fájlból beolvasás esetén a program "f" karaktert vár a felhasználótól. Ebben az esetben a `fajlBeker()` függvény hívódik meg, amely egy szöveges állományból olvassa be a települések számát, a madárfajok számát, illetve az egyes településeken az egyes madárfajok egyedeinek a számát. A `fajltMegnyit()` függvény a felhasználótól bekéri a beolvasni kívánt fájl nevét és ellenőrzi, hogy az adott fájl létezik-e.

A `megszcica()` függvény a feladatot oldja meg a `mindElofordult()` függvény segítségével. A `mindElofordult()` függvény egy logikai értékkel tér vissza, ami akkor igaz, ha az adott sorban(településen) mindegyik madárfajt megfigyelték(minden madárfaj egyedeinek a száma nagyobb, mint 0). A `megszcica()` függvény megszámolja, hogy a `mindElofordult()` függvény hányszor tért vissza igaz értékkel.



Tesztelési terv

Tesztesetek a feladat specifikációja alapján (fekete doboz tesztelés)

Érvényes tesztesetek:

1. Külső programozási tétel(intervallum és megszámlálás):

1.1. Üres adatok esete:

0 település és 0 madárfaj(be1): 0 db település

0 település és több madárfaj(be2): 0 db település

Több település és 0 madárfaj(be3): n db település, ahol n a települések száma

1.2.Egy település és az első madárfaj nincs megfigyelve(be10): 0 db település

1.3.Egy település és az utolsó madárfaj nincs megfigyelve(be11): 0 db település

1.4.Több település és több madárfaj és az első madárfaj egyik településen sincs megfigyelve(be4): 0 db település

1.5.Több település és több madárfaj és az utolsó madárfaj egyik településen sincs megfigyelve(be5): 0 db település

1.6.Több település és több madárfaj és mindegyik településen csak 1 madárfaj van megfigyelve(be6): 0 db település

1.7.Több település és több madárfaj és mindegyik településen van olyan madárfaj, ami nincs megfigyelve(be7): 0 db település

1.8.Több település és több madárfaj és egyik településen sincs egyik madárfaj se megfigyelve(be8): 0 db település

1.9.Több település és több madárfaj és mindegyik településen mindegyik madárfaj meg van figyelve(be9): n db település, ahol n a települések száma

2. Belső programozási tétel(intervallum és opt. Linker):

2.1. Üres adatok esete:

0 település és 0 madárfaj(be1): 0 db település

0 település és több madárfaj(be2): 0 db település

Több település és 0 madárfaj(be3): n db település, ahol n a települések száma

2.2.Egy település és az első madárfaj nincs megfigyelve(be10): 0 db település

2.3.Egy település és az utolsó madárfaj nincs megfigyelve(be11): 0 db település

2.4.Több település és több madárfaj és az első madárfaj egyik településen sincs megfigyelve(be4): 0 db település

2.5.Több település és több madárfaj és az utolsó madárfaj egyik településen sincs megfigyelve(be5): 0 db település

2.6.Több település és több madárfaj és mindegyik településen csak 1 madárfaj van megfigyelve(be6): 0 db település

2.7.Több település és több madárfaj és mindegyik településen van olyan madárfaj, ami nincs megfigyelve(be7): 0 db település

2.8. Több település és több madárfaj és egyik településen sincs egyik madárfaj se megfigyelve (be8): 0 db település

2.9. Több település és több madárfaj és mindegyik településen mindegyik madárfaj meg van figyelve (be9): n db település, ahol n a települések száma

Érvénytelen tesztesetek:

Nem megengedett értékek:

A megadott szám nem természetes szám: lekezelet a program, az adatot újrakéri.

A beolvasandó fájl nem felel meg a példában megadott formátumnak: Nem kezeljük, mert kikötöttük a formátumot.

Tesztesetek a megoldó kód alapján (fehér doboz tesztelés)**A beolvasást végző függvény tesztelése:**

1. Menü választás tesztelése (b, f, más)
2. Beolvasás mindkét módozatának tesztelése
3. Nem létező fájl név megadása
4. Hibás adatok megadása a billentyűzetről (negatív értékek, nem szám karakterek)

A megszűnés() függvény tesztelése:

1. Olyan adatsorra, amikor egyszer sem lép a ciklusba a vezérlés: be1 - válasz: 0
2. Olyan adatsorra, amikor egyszer sem növeljük a darabszámot: be4 - válasz: 0
3. Olyan adatsorra, amikor egyszer növeljük a darabszámot: be12 - válasz: 1
4. Olyan adatsorra, amikor többször növeljük a darabszámot: be9 - válasz: n, ahol n a települések száma.

A mindElofordult() függvény tesztelése:

1. Igen válasz esete minden madárfajra : be9 - válasz: n, ahol n a települések száma.
2. Nemleges válasz az összes első madárfajnál: be4 - válasz: 0
3. Nemleges válasz az összes utolsó madárfajnál: be5 - válasz: 0