Programozási alapismeretek  
komplex beadandó feladat

Készítette: Széplaki Bence  
Neptun-azonosító: STM3ML  
E-mail: [szeplakibence@gmail.com](mailto:szeplakibence@gmail.com)  
 [stm3ml@inf.eltel.hu](mailto:stm3ml@inf.eltel.hu)

Kurzuskód: IP-18PROGEG  
Gyakorlat vezető neve: Klettner Péter

2021. január 10.

Tartalom:

Felhasználói dokumentáció 3

Feladat 3

Futási környezet 3

Használat 3

A program indítása 3

A program bemenete 3

A program kimenete 4

Minta bemenet és kimenet 4

Hibalehetőségek 4

Mintafuttatás hibás adatok esetén 5

Fejlesztői dokumentáció 6

Feladat 6

Specifikáció 6

Fejlesztői környezet 6

Forráskód 6

Megoldás 7

Programparaméterek 7

Programfelépítés 7

Függvénystruktúra 7

Algoritmus 7

A kód 9

Tesztelés 11

Érvényes tesztesetek 11

Érvénytelen tesztesetek 13

Fejlesztési lehetőségek 13

Felhasználói dokumentáció:

Feladat:

Az ország néhány helységében madármegfigyelést végeztünk. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk.  
Készíts programot, amely megadja azokat a madárfajokat, amelyek egyedei több mint 90%-ban egy helységben voltak megtalálhatóak és hogy hány ilyen eset volt!

Futási környezet:

IBM PC, exe futtatására alkalmas, 32-bites operációs rendszer (pl. Windows 10).  
Nem igényel egeret.

Használat:

A program futtatása:

A program az *STM3ML.rar\Komplex\bin\Release\Komplex.exe* néven található a tömörített állományban. A *Komplex.exe* fájl kiválasztásával indítható.

A program bemenete:

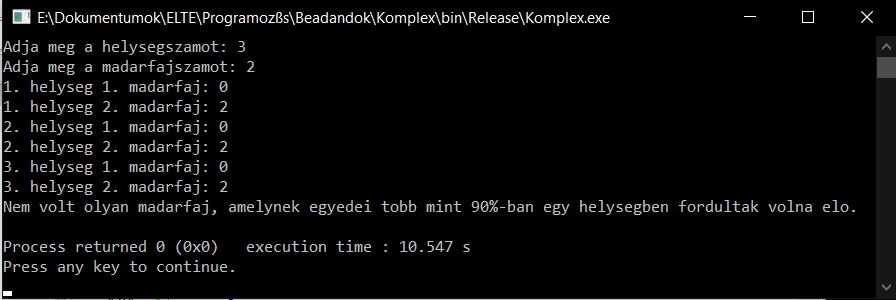
|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *M* | Azon helységek száma, ahol madármegfigyeléseket végeztünk. (0<M<=50) |
| **2.** | *N* | Azon madárfajok száma, amelyeket megfigyeltünk a madármegfigyelés alatt. (0<N<=200) |
| **3.** | *Darab1 1* | Első helység, első madárfajának darabszáma. (1<=Darab<=1000) |
| **...** | *...* | ... |
| **N+2** | *Darab1-N* | Első helység, N-dik madárfajának darabszáma. (1<=Darab<=1000) |
| **N+3** | *Darab2-1* | Második helység, első madárfajának darabszáma. (1<=Darab<=1000) |
| **...** | *...* | ... |
| **N+M+2** | *DarabM-N* | M-dik helység, N-dik madárfajának darabszáma. (1<=Darab<=1000) |

A program kimenete:

A program kettő lehetséges kimenettel rendelkezik:

1. Ha nem volt olyan madárfaj, amelynek egyedi több mint 90%-ban egy helységben fordultak volna elő, akkor a program azt fogja nekünk kiírni :  
   „Nem volt olyan madarfaj, amelynek egyedei tobb mint 90%-ban egy helysegben fordultak volna elo.”
2. Ellenkező esetben a program kiírja, hogy hány darab ilyen eset volt, és a következő sorban pedig azt, hogy melyik madárfajok voltak ezek.

Minta bemenetre és kimenetre:

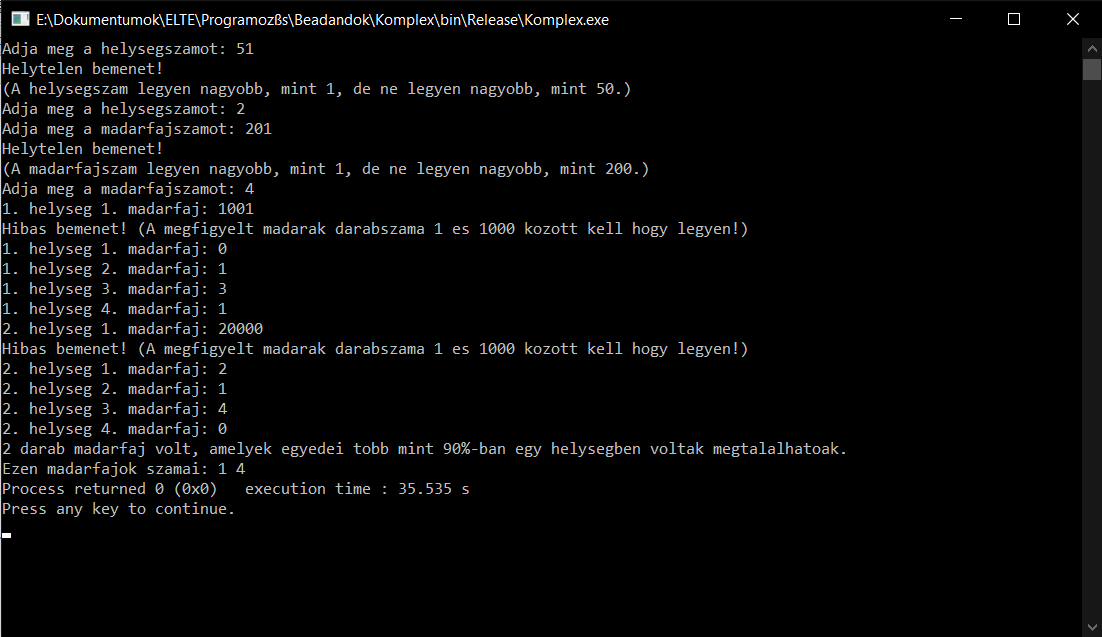




Hibalehetőségek:

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a helységek száma nem egész szám, vagy nem esik a 1..50 intervallumba; vagy ha a madárfajok száma nem egész szám, vagy nem esik a 1..200 intervallumba, vagy valamely darabszám nem egész szám, vagy nem esik az 1..1000 intervallumba. Hiba esetén a program jelzi a hibát és az elgépelt/elírt/helytelen értéket újra kérdezi addig, amíg a felhasználó helyesen be nem viszi azt.

Mintafutás hibás bemeneti adatok esetén:



Fejlesztői dokumentáció:

Feladat:

Az ország néhány helységében madármegfigyelést végeztünk. Mindegyikben megadtuk, hogy milyen fajú madárból hányat láttunk.  
Készíts programot, amely megadja azokat a madárfajokat, amelyek egyedei több mint 90%-ban egy helységben voltak megtalálhatóak és hogy hány ilyen eset volt!

Specifikáció:

**Bemenet:**

* M∈N [helységek száma]
* N∈N [madárfajok száma]
* matrix[1..M][1..N]∈R [a madarak számát tartalmazó tömb]

**Kimenet:**

* Db∈N
* Madarfajok[1..Db]∈N

**Előfeltétel:** 1 ≤ M ≤ 50 ∧ 1 ≤ N ≤ 200 ∧ 1 ≤ Mátrix[1..M][1..N] ≤ 1000

**Utófeltétel:** 0 ≤ Db ∧ 0 ≤ Madarfajok[1..Db] ≤ N

Környezet:

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 10). mingw32-g++.exe c++ fordítóprogram (v4.7), Code::Blocks (v20.03) fejlesztői környezet.

Forráskód:

A teljes fejlesztői anyag –kicsomagolás után– az STM3ML nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| STM3ML\bin\Release\Komplex.exe | futtatható kód |
| STM3ML\obj\Release\main.o | félig lefordított kód |
|  |  |
| STM3ML\main.cpp | C++ forráskód |
|  |  |
| STM3ML\teszt1.txt | teszt-bemeneti fájl |
| STM3ML\teszt2.txt | teszt-bemeneti fájl |
| STM3ML\teszt3.txt | teszt-bemeneti fájl |
| STM3ML\teszt4.txt | teszt-bemeneti fájl |
| STM3ML\teszt5.txt | teszt-bemeneti fájl |
|  |  |
| STM3ML\dokumentacio\STM3ML\_Dokumentacio.docx | dokumentáció |

Megoldás:

Programparaméterek:

Változók:

M : **Egész**(50)

N : **Egész**(200)

db : **Egész**

matrix : **Tömb**[M:Egész][N:Egész]

oszloposszeg : **Tömb**[N:Egész]

madarfajok : **Tömb**[N:Egész]

Programfelépítés

A program által használt modulok (és helyük):

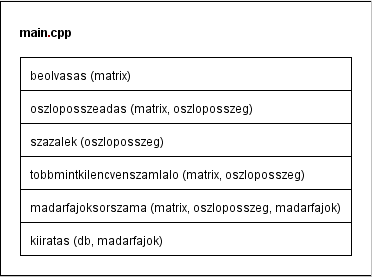
*main.cpp* – program, a forráskönyvtárban

*iostream* – képernyő-, és billentyűkezelés, a C++ rendszer része

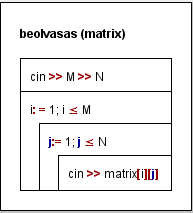
Függvénystruktúra:

A teljes program algoritmusa:

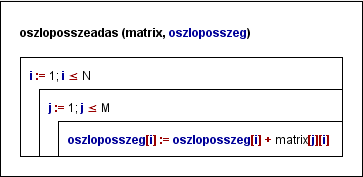
Főprogram:

****

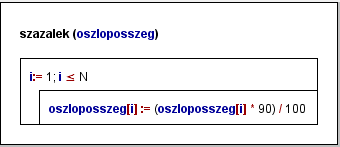
Beolvasás:



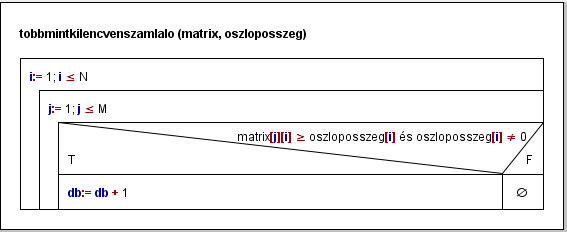
Oszloposszeadas:



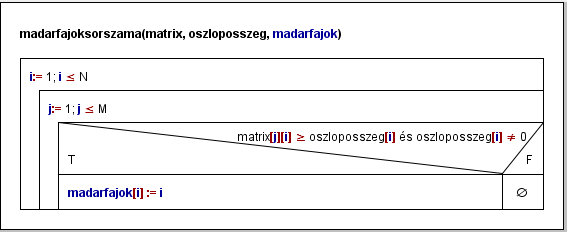
Szazalek:



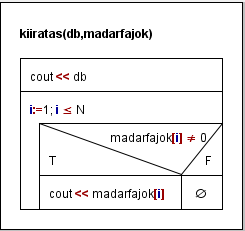
Tobbmintkilencvenszamlalo:



Madarfajoksorszama:



Kiiratas:



A kód:

A *main.cpp* fájl tartalma:

/\*

Készítette: Széplaki Bence

Neptun-kód: STM3ML

E-mail: szeplakibence@gmail.com

stm3ml@inf.elte.hu

Feladat: "ProgAlap beadandó feladatok" téma "Madárfajok egyedei több mint 90%-ban egy helységben" feladat

\*/

#include <iostream>

using namespace std;

int N, M, db = 0;

void beolvasas(float matrix[51][201])

{

...

}

void oszloposszeadas(float matrix[51][201], float oszloposszeg[201])

{

...

}

void szazalek(float oszloposszeg[201])

{

...

}

void tobbmintkilencvenszamlalo(float matrix[51][201], float oszloposszeg[201])

{

...

}

void madarfajoksorszama(float matrix[51][201], float oszloposszeg[201], int madarfajok[201])

{

...

}

void kiiratas(int db, int madarfajok[201])

{

...

}

int main()

{

float matrix[51][201], oszloposszeg[201] = {0};

int madarfajok[201];

beolvasas(matrix);

oszloposszeadas(matrix,oszloposszeg);

szazalek(oszloposszeg);

tobbmintkilencvenszamlalo(matrix,oszloposszeg);

madarfajoksorszama(matrix,oszloposszeg,madarfajok);

kiiratas(db,madarfajok);

return 0;

}

Tesztelés:

Érvényes tesztesetek:

1. Teszteset: be1.txt

|  |
| --- |
| Bemenet – *nincs olyan madárfaj* |
| M = 3  N = 3  matix 1,1 = 0  matix 1,2 = 0  matix 1,3 = 0  matix 2,1 = 0  matix 2,2 = 0  matix 2,3 = 0  matix 3,1 = 0  matix 3,2 = 0  matix 3,3 = 0 |
| Kimenet |
| 0 |

1. Teszteset: be2.txt

|  |
| --- |
| Bemenet – *egy olyan madárfaj van* |
| M = 4  N = 3  matix 1,1 = 1  matix 1,2 = 0  matix 1,3 = 1  matix 2,1 = 2  matix 2,2 = 0  matix 2,3 = 1  matix 3,1 = 3  matix 3,2 = 5  matix 3,3 = 2  matix 4,1 = 4  matix 4,2 = 0  matix 4,3 = 1 |
| Kimenet |
| 1  2 |

1. Teszteset: be3.txt

|  |
| --- |
| Bemenet – *több olyan madárfaj van* |
| M = 2  N = 4  matix 1,1 = 0  matix 1,2 = 1  matix 1,3 = 3  matix 1,4 = 1  matix 2,1 = 5  matix 2,2 = 1  matix 2,3 = 4  matix 2,4 = 0 |
| Kimenet |
| 2  1 4 |

1. Teszteset: be4.txt

|  |
| --- |
| Bemenet – *mindegyik madárfajra igaz* |
| M = 4  N = 4  matix 1,1 = 1  matix 1,2 = 0  matix 1,3 = 0  matix 1,4 = 0  matix 2,1 = 0  matix 2,2 = 2  matix 2,3 = 0  matix 2,4 = 0  matix 3,1 = 0  matix 3,2 = 0  matix 3,3 = 3  matix 3,4 = 0  matix 4,1 = 0  matix 4,2 = 0  matix 4,3 = 0  matix 4,4 = 4 |
| Kimenet |
| 4  1 2 3 4 |

1. Teszteset: be5.txt

|  |
| --- |
| Bemenet – *MAX érték M és N-re* |
| M = 50  N = 200  matix 1,1 = 0  matix 1,2 = 0  ...  matix 50,200 = 0 |
| Kimenet |
| 67  1 2 4 6 8 9 10 13 19 21 23 24 26 34 37 40 43 44 49 51 53 60 61 62 66 67 68 70 74 77 79 81 84 88 90 91 92 100 102 108 111 112 114 116 121 122 123 126 133 136 139 141 142 147 149 160 161 162 169 172 175 182 184 190 192 196 198 |

Érvénytelen tesztesetek:

1. Teszteset:

|  |
| --- |
| Bemenet – *Negatív M értékadás* |
| M = -2 |
| Kimenet |
| Ujrakérdezés  M = ? |

1. Teszteset:

|  |
| --- |
| Bemenet – |
| M = 1  N = 2  matrix 1,1 = -5 |
| Kimenet |
| Ujrakérdezés  matrix 1,1 = ? |

Fejlesztői lehetőségek:

* Adatok –a felhasználó igénye szerint– akár fájlból is fogadása.
* Hibás fájl-bemenetek felismerése, és a hiba helyének (sor sorszámának) kiírása.
* Többszöri futtatás megszervezése
* Helységszám kiírása