

TÖBBDIMENZIÓS TÖMBÖK

GYAKORLÓ FELADATOK C#-BAN

TARTALOM

Kétdimenziós tömbök	1
1. Véletlen mátrix	1
2. Transzponált mátrix	1
3. Mátrix-összeg	1
4. Karaktermátrix	2
5. Elemcsere	2
6. Sorcsere	3
7. Véletlenkeverés	3
8. Szintvonalas terkep	4
Algoritmusok kódolása	5
9. Levenshtein-távolság	5

KÉTDIMENZIÓS TÖMBÖK

1. VÉLETLEN MÁTRIX

Töltsünk fel egy 3x4-es mátrixot véletlen számokkal, majd írassuk ki a képernyőre (mátrix formában).

2. TRANSZPONÁLT MÁTRIX

Töltsünk fel egy 3x4-es mátrixot a programkódban egész számokkal. Írassuk ki mátrix formátumban, majd generáljuk a transzponáltját, és ezt is írassuk ki. (A transzponált mátrixot az eredeti mátrix sorainak és oszlopainak felcserélésével kapjuk.)

A két mátrix kiíratását paraméteres eljárás segítségével oldjuk meg!

3. MÁTRIX-ÖSSZEG

Egy 4x4-es mátrixot töltsünk fel egészekkel a billentyűzetről. Ezen a mátrixon hajtsuk végre az alábbi feladatokat:

- Írassuk ki a mátrixot.
- Írassuk ki a mátrixban lévő számok összegét, átlagát.
- Számoljuk ki a főátló összegét, majd írassuk ki.
- Kérjünk be a felhasználótól egy sorszámot (csak olyan értéket fogadjunk el, ami a mátrixban létezik), majd ennek a sornak az átlagát írassuk ki.
- Írassuk ki sorban egymás után, vesszővel elválasztva minden oszlop legkisebb elemét.

Segítség: Egy kétdimenziós **m** nevű mátrix sorainak számát az **m.GetLength(0)**, oszlopainak számát pedig az **m.GetLength(1)** parancs segítségével kérdezhetjük le.

A program egy lehetséges kimenete:

```

1. sor, 0. oszlop: 2
1. sor, 1. oszlop: 3
1. sor, 2. oszlop: 4
1. sor, 3. oszlop: 5

2. sor, 0. oszlop: 3
2. sor, 1. oszlop: 4
2. sor, 2. oszlop: 5
2. sor, 3. oszlop: 6

3. sor, 0. oszlop: 4
3. sor, 1. oszlop: 5
3. sor, 2. oszlop: 6
3. sor, 3. oszlop: 7

A mátrix így néz ki:
1 2 3 4
2 3 4 5
3 4 5 6
4 5 6 7

A mátrix elemeinek összege: 64.
A mátrix elemeinek átlaga: 4.
A főátló összege: 16
Kérem a sor számát: 5
Kérem a sor számát: -3
Kérem a sor számát: 2
A mátrix 2. sorában lévő elemek átlaga: 4,5.
Oszlopok minimumértékei:
1, 2, 3, 4,

```

4. KARAKTERMÁTRIX

Egy 15x15-ös mátrixba generáljunk véletlenszerű karaktereket (segítség: generáljunk egy számot pl. 65-255 között, majd tegyük a tömbbe ennek az ASCII kódtábla szerinti karakterét). Az így kapott mátrixot írassuk a képernyőre, majd kérjünk be a felhasználótól egy karaktert. Ezt a karaktert írjuk be a mátrix főátlóiba. A végeredmény mátrixot is írassuk a képernyőre.

A mátrix kiíratását függvény segítségével oldjuk meg.

5. ELEMCSERE

Kérjük be a mátrix sor- és oszlopszámát, majd töltsük fel 0-tól kezdődően 1-gyel növekvő egészekkel. Függvény segítségével írassuk ki a mátrixot mátrix formában.

Kérjük be két elem indexét (sor-oszlopazonosító), majd ezeken az indexeken lévő elemeket cseréljük meg a mátrixban. Az eredménymátrixot is írassuk a képernyőre. Csak létező index-számokat fogadjon el a programunk!

Egy lehetséges kimenet:

```

Elemcsere program

Sorok száma: 4
Oszlopok száma: 4

A generált növekvő mátrix:
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
a1= 3
b1= 3
a2= 2
b2= 2
Cserélni fogjuk matrix[3,3]=15 elemet erre: matrix[2,2]=10

A csere után előállt új mátrix:
0 1 2 3
4 5 6 7
8 9 15 11
12 13 14 10

```

6. SORCSERE

Töltsünk fel egy 11x11-es mátrixot soronként egyforma, de föntről lefelé 50-től 2-essel csökkenő számokkal. Kérjünk be két (létező) sorszámot a felhasználótól, majd ezeket a sorokat cseréljük meg. Mindkét mátrixot írassuk a képernyőre.

7. VÉLETLENKEVERÉS

Egy 8x8-as mátrixot töltsünk fel 99-től egyesével csökkenő elemekkel. Kérdezzük meg a felhasználótól, hány keverést kér. Oldjuk meg a kért darabszámú keverést.

A keverés úgy történik, hogy a program véletlenszerűen kisorsol két megcserélendő elemet, amit felcserélünk. A cserélni kívánt elemeket is, és az eredménymátrixot is írassuk a képernyőre a keverés sorszámával együtt.

A program egy lehetséges kimenete:

```
Véletlenkeverés program

A generált mátrix:
99 98 97 96 95 94 93 92
91 90 89 88 87 86 85 84
83 82 81 80 79 78 77 76
75 74 73 72 71 70 69 68
67 66 65 64 63 62 61 60
59 58 57 56 55 54 53 52
51 50 49 48 47 46 45 44
43 42 41 40 39 38 37 36
Mennyi keverés legyen?
3

1. keverés: 6,5 --> 4,0
Cserélni fogjuk matrix[6,5]=46 elemet erre: matrix[4,0]=67

2. keverés: 1,6 --> 3,1
Cserélni fogjuk matrix[1,6]=85 elemet erre: matrix[3,1]=74

3. keverés: 1,2 --> 6,5
Cserélni fogjuk matrix[1,2]=89 elemet erre: matrix[6,5]=67

A csere után előállt új mátrix:
99 98 97 96 95 94 93 92
91 90 67 88 87 86 74 84
83 82 81 80 79 78 77 76
75 85 73 72 71 70 69 68
46 66 65 64 63 62 61 60
59 58 57 56 55 54 53 52
51 50 49 48 47 89 45 44
43 42 41 40 39 38 37 36
```

8. SZINTVONALAS TERKEP

A föld felszínének magasságát egy kétdimenziós mátrixban tároljuk. Írjunk olyan programot, amelyik a mátrix alapján kirajzolja a domborzat magasságának megfelelő színkódú karakterekkel.

A mátrix mérete legyen: 20x10-es. A magasság értékeket cm-ben tároljuk, melyet véletlenszám-generátorral állítunk elő. A szám legyen -3.000 és 3.000 között.

A színek:

- -3.000-0: kék
- 0-1.000: sárga
- 1.000-2.000: piros
- 2.000-3.000: sötétpiros

A program egy lehetséges kimenete:

Szintvonalas térkép program

A generált mátrix:

```
76 2107 -2383 2851 706 -1366 -1228 -895 -2354 341
-1679 -357 1118 -211 -167 -1329 1227 -1261 -430 -1180
2599 1012 -1577 -1032 -2270 2014 1196 2529 546 1692
2437 -2233 1569 1625 -1353 1185 2808 2780 872 -2071
-242 2343 1288 2313 370 2975 836 859 -466 -2264
-933 48 -1257 -1222 -1668 2063 684 1649 2120 1692
437 -758 1558 -1046 903 -2446 1073 2493 -1859 1648
-1137 1447 866 -1359 2061 -2745 2724 -890 1598 -2246
-1822 -2663 -6 -190 -376 -611 2023 -209 292 -416
-2499 -1841 -2341 2180 491 -2484 -2216 -666 -1590 -185
-1099 1342 -282 886 1872 620 2182 -2898 -284 -1923
2339 -1427 -2950 1366 514 100 2248 -1252 2330 -1487
-1469 -1718 -800 1557 1146 -2297 787 1686 1149 1545
2470 390 -2300 -1217 1852 -165 619 2107 937 2003
206 1920 -2925 -785 245 1927 -1891 -707 -1335 -2609
-1733 2036 3 2896 -1716 -2381 -858 1161 2739 1169
2833 -2504 -1047 1170 -1394 -1052 -2127 1266 2513 -2321
2481 -2860 810 -2674 1307 -390 -1793 2985 -1689 2219
2594 -1507 20 757 277 -2566 -2614 -2196 -2502 1233
-2307 2458 2368 2768 2170 -290 -33 -1096 -2391 -1704
```

A szintvonalas térkép:

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

ALGORITMUSOK KÓDOLÁSA

9. LEVENSZTEIN-TÁVOLSÁG

A következő algoritmus két, maximum 25 karakter hosszú karakterláncról megállapítja, hogy mekkora a Levenshtein-távolságuk, azaz minimálisan hány karakterenkénti művelet (beszúrás, törlés, csere) kell ahhoz, hogy az egyik karakterláncot a másikra átalakítsuk.

Kódolja az algoritmust a választott programozási nyelven! Az elkészült program forráskódját mentse LD néven!

A megoldás során vegye figyelembe a következőket:

- A "Hossz()" függvény a karakterlánc hosszát adja meg.
- A választott programozási nyelvtől függően eltérő jelölésű operátorokat és függvényeket kell alkalmaznia.
- A "Térj vissza!" utasítás megszakítja a függvény futását, és meghatározza annak visszatérési értékét.

```
Függvény Min(a: Egész, b: Egész): Egész
  Ha a<b akkor
    Min:= a
  különben
    Min:= b
  Elágazás vége
Függvény vége

Függvény LDTav(s1: Szöveg, s2: Szöveg): Egész
  Változó h1, h2:Egész
  Változó c:Egész
  Változó tömb m[0..26, 0..26]:Egész
  h1:= Hossz(s1)
  h2:= Hossz(s2)
  Ha h1= 0 akkor Térj vissza h2-vel
  Ha h2= 0 akkor Térj vissza h1-el
  Ciklus i:= 0-tól h1-ig (+1 lépésközzel)
    m[i,0]:= i
  Ciklus vége
  Ciklus j:= 0-tól h2-ig (+1 lépésközzel)
    m[0,j]:= j
  Ciklus vége
  Ciklus i:= 1-től h1-ig (+1 lépésközzel)
    Ciklus j:= 1-től h2-ig (+1 lépésközzel)
      c:= 1
      Ha s2[j-1]= s1[i-1] akkor c:= 0
      m[i,j]:= Min(Min(m[i-1,j]+1,m[i,j-1]+1),m[i-1,j-1]+c)
    Ciklus vége
  Ciklus vége
  LDTav:= m[h1,h2]
Függvény vége

Program LD
  Ki: LDTav("alma","falja")
Program vége.
```

/Informatikai alapismeretek — emelt szintű érettségi (2015. okt.)/