**Gyakorló feladatok**

## 1. Völgyszámok és hegyszámok

## Írjatok programot, amely eldönti egy adott számról, hogy völgyszám-e vagy hegyszám. Egy völgyszám számje­gyei szigorúan csökkenő sorrendben követik egymást egy bizonyos elemmel bezárólag, azután szigorúan növekvő sorozatot alkotnak. A hegyszám számjegyei szigorúan növekvő sorrendben követik egymást egy bizo­nyos elemmel bezárólag, azután szigorúan csökkenő sorozatot alkotnak.

Írjatok ki egy megfelelő üzenetet aszerint, hogy az adott szám völgyszám-e vagy hegyszám. Ha a szám nem völgyszám és nem hegyszám, vágjátok le a szám *első néhány* számjegyét amíg az így kapott szám völgyszámmá vagy hegyszámmá válik. Ebben az esetben írjátok ki az adott szám azon részét, amely völgyszám vagy hegyszám, ha létezik ilyen, vagy megfelelő üzene­tet, ha nem.

**Példák:**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 13752 | hegyszam |
| 85369 | volgyszam |
| 913752 | 13752 |
| 1234 | nincs benne hegyszam vagy volgyszam |

## 2. *n* szám legnagyobb közös osztója

## Írjatok programot, amely kiszámítja és kiírja n darab természetes szám legnagyobb közös osztóját!

## ****Példa:****

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 3  16 24 32 | 8 |

**3. „Majdnem prímszámok”**

Egy természetes számot *majdnem prím*nek nevezünk, ha egyenlő *két különböző prímszám szorzatával.* Például, a 15 *majdnem prím*, mivel egyenlő a 3 és 5 prímszámok szorzatával.

Legyen egy ***n*** természetes szám(1 ≤ ***n*** ≤ 1 000) és egy ***n*** elemű ***x*** sorozat, amelyben az ele­mek 1-nél szigorúan nagyobb és 30 000-nél kisebb természetes számok.

Írjatok programot, amely meghatározza az adott sorozat *leghosszabb* tömbszakaszát, amely csak *majdnem prím*eket tartalmaz, és kiírja az illető tömbszakasz kezdőindexét (***balMax***) és végsőindexét (***jobbMax***). Ha több ilyen tömbszakasz létezik, a *legelső* leghosszabb tömbsza­kasz kezdőindexét és végsőindexét kell kiírnotok. Ha a sorozatban nem létezik egyetlen *majdnem prím* sem, ***balMax*** és ***jobbMax*** értéke egyenlő lesz -1-gyel.

## ****Példák:****

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 10  24 34 35 11 8 77 35 26 28 30 | 6 8 |
| 3  24 11 8 | -1 -1 |

**4. Bűvös négyzet**

Vizsgáljuk meg, hogy egy adott, ***n*** x ***n*** méretű négyzetes tömb, amelynek elemei 260-nál kisebb természetes számok, bűvös négyzet-e vagy sem (***n*** ≤ 10)! Egy különböző természetes számokkal feltöltött négyzetes tömböt akkor nevezünk bűvös négyzetnek, ha a számok összege soronként és oszloponként, valamint a két átló mentén azonos.

**Példák**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 3  4 9 2  3 5 7  8 1 6 | igen |
| Bemenet | Kimenet |
| 2  1 1  1 1 | nem |

**5. Hasonló számok**

Legyen ***x*** és ***y*** két természetes szám. Írjatok programot, amely eldönti, hogy a két szám *hasonló-e*. Két számot hasonlónak nevezünk, ha számjegyeik halmaza azonos: 2131 és 32211 hasonló mivel számjegyeik halmaza ugyanaz: {1,2,3}.

a) Tömbhasználattal (egy tömb, két tömb)

b) Tömbhasználat nélkül

**Példák**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 31177  7133 | igen |
| Bemenet | Kimenet |
| 123  1214  1 1 | nem |

**6. Páros tömbszakasz**

Egy sorozat egymás után elhelyezkedő elemekből álló részét, amely legkevesebb két páros számot tartalmaz, *páros tömbszakasz*nak nevezzük. A sorozatnak 1 ≤ ***n*** ≤ 106 természetes szám eleme van, és 0 ≤***szám***≤ 109.

Írjunk programot, amely meghatározza az adott sorozat *leghosszabb* *páros tömbszakaszainak* *darabszámát*. Ha nincs egyetlen páros tömbszakasz sem, 0-t írunk ki.

**Példa**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 26  1 2 3 4 6 10 2 8 5 7 9 4 6 10 121 20 4 11 10 2 5 2 6 8 10 16 | 2 |

**7. Számtani haladvány**

Egy sorozat ***r*** *léptékű számtani haladvány*, ha bármely elemét – kivéve az elsőt – megkapjuk, ha az előtte levőhöz hozzáadunk ***r***-t.

Például, a (12, 14, 16, 18, 20) sorozat egy 2 léptékű számtani haladvány.

Adott legtöbb 3 ≤ ***n*** ≤ 105 természetes szám, ahol 0 ≤ ***szám*** ≤ 103. Írjunk programot, amely eldönti, hogy létezik-e egy ***r*** természetes szám, amely annak a számtani haladványnak a léptéke, amelyet úgy kapnánk, ha az adott sorozat *különböző* értékeit átrendeznénk. Ha nem létezik ilyen ***r*** szám, írjunk ki megfelelő üzenetet, különben írjuk ki a lépték értékét.

A megoldást *rendezés nélkül* végezzük!

**Példa**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 11  180 30 80 280 130 330 230 30 30 330 80 | 50 |

**8. Nyeregpontok**

Adott egy ***n*** soros és ***m*** oszlopos (0 < ***n***, ***m*** ≤ 100), egész számokat tároló kétdimenziós *a* tömb. Írjatok programot, amely megszámolja a tömb nyeregpontjait!

Egy ***aij*** elemet *nyeregpont*nak nevezünk, ha az ***aij*** elem legnagyobb a ***j****.* oszlopban és legki­sebb az ***i****.* sorban, és fordítva. Írjátok ki a nyeregpontok darabszámát és az elemek indexeit.

**Példa**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 2 6  5 2 8 4 9 3  7 1 6 3 8 5 | 1 2  2 5 |

**9. Képfeldolgozás**

Egy fehér-fekete képet egy olyan négyzetes tömbbel kódolunk, amelynek elemei nullák (0 = fehér pixel) és egyesek (1 = fekete pixel). A képet a következő műveletekkel alakíthatjuk át:

* **Megfordítás** (**M**), vagyis a 0-t 1-gyé, az 1-et 0-ává változtatjuk;
* **Forgatás** 90 fokkal (**F**), az óramutató járásával megegyező irányban;
* **Zoom** (**Z**), vagyis minden pixel helyére négy új, az eredetivel azonos értékű pixel kerül.

Egy feldolgozás-sorozatot egy – M, F és Z betűkből álló – karaktersorozattal írunk le, ahol a betűk sorrendje tetszőleges.

Írjunk programot, amely adott, ***m*** sorral és ***m*** oszloppal rendelkező ***kép*** kétdimenziós tömb (***m*** – természetes szám, 2 ≤ ***m*** ≤ 10) és ***s***, legtöbb öt képfeldolgozó műveletet tartalmazó betűsor esetében végrehajtja ezeket a műveleteket és kiírja az átalakított képet.

***Példa*:** ha ***m*** = 3, ***kép*** = és ***k*** = 4, ***s*** = (F, M, F, Z), akkor a feldolgozás eredménye: .

Az egyes feldolgozások eredményei, rendre:

, , , .

A megoldásban írjatok egy-egy alprogramot, amely:

1. beolvassa a bemeneti adatokat a **kep.in** állományból (ezek garantáltan megfelelnek a követelményeknek);
2. megfordít egy képet;
3. elforgat 90 fokkal egy képet;
4. végrehajtja a zoom műveletet egy képre;
5. kiír a képernyőre egy képet.