**Felvételi előkészítő**

**(Tömbök)**

1. Írjatok programot, amely kiszámítja és kiírja ***n*** darab természetes szám legnagyobb közös osztóját!

**Példa:**

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

1. Vizsgáljuk meg, hogy egy adott, ***n*** × ***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 |

1. 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 *nyeregpontnak* 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 |

1. 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:

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

1. Ismert, hogy bármely természetes szám ábrázolható az úgynevezett Fibonacci számrendszerben, ahol a számjegyek 1 és 0:

Egy, a Fibonacci számrendszerben felírt szám alakja: ***xFib*** = ***cncn*-1…*c*3*c*2*c*1**, ahol ***ci*** **1** vagy **0**.

Az ***x*** szám értékét a 10-es számrendszerben a következőképpen számítjuk ki:

***x***10 = ***cn*** ⋅ ***Fn*** + … + ***c*3** ⋅ **3** + ***c*2** ⋅ **2** + ***c*1** ⋅ **1**, ahol ***Fi*** a Fibonacci sorozat ***i***-dik eleme (az **F0** = **0** elemet nem vesszük figyelembe és a sorozat egyetlen **1**-essel kezdődik).

**Példa:**

***x****Fib* = 10101001 = 1 ⋅ 34 + 0 ⋅ 21 + 1 ⋅ 13 + 0 ⋅ 8 + 1 ⋅ 5 + 0 ⋅ 3 + 0 ⋅ 2 + 1 ⋅ 1 = 5310

Azért, hogy az ábrázolás egyedi legyen, a fenti szabályhoz hozzátesszük még a következő követelményt: az ábrázolás ne tartalmazzon egymás után két darab 1-est.

Írjatok programot, amely megszámolja azokat az adott ***n*** (1 < ***n*** ≤ 1 000 000) számnál kisebb (esetleg ***n***-nel egyenlő) számokat, amelyek felírva Fibonacci számrendszerben palindromszámok.

**Példa:**

|  |  |  |
| --- | --- | --- |
| Bemenet | Kimenet | *Magyarázat* |
| 15 | 6 | 6 darab 15-nél kisebb, adott tulajdonságú szám létezik: 1, 4, 6, 9, 12, 14, amelyeknek alakja felírva Fibonacci számrendszerben:  1, 101, 1001, 10001, 10101, 100001. |

Maximális végrehajtási idő: 0,1 szekundum

**Tesztállományok: TesztekFibo.zip**

1. A ***hrs*** sorozat elemeit a következő szabályok alapján számítjuk ki:

…

**Követelmény**

Számítsátok ki, adott ***n*** esetében ***rn*** és ***sn*** értékét.

**Bemeneti adatok**

A **hrs.in** bemeneti állomány első sorában az ***n*** természetes szám található.

**Kimeneti adatok**

A **hrs.out** kimeneti állomány egyetlen sorába két természetes számot kell írnotok, amelyek közül az első az ***r*** sorozat ***n***. eleme és a második az ***s*** sorozat ***n***. eleme.

**Megszorítások és pontosítások**

* 2 ≤ ***n*** ≤ 65 000.

**Példa**

|  |  |
| --- | --- |
| **hrs.in** | **hrs.out** |
| 15 | 150 20 |

**Tesztállományok: Tesztekhrs.zip**

1. Egysorozat ***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 106 természetes szám, ahol 0 ≤ ***szám*** ≤ 1000. Döntsük el, hogy létezik-e egy ***r*** természetes szám, amely annak a számtani haladványnak a léptéke, amelyet úgy kapunk, hogy az adott sorozat különböző értékeit átrendezzük. Ha nem létezik ilyen ***r*** szám, írjunk ki megfelelő üzenetet.

Oldjuk meg a feladatot rendezés nélkül!