**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. Ismerjük egy osztály tanulóinak neveit (családnév + keresztnév) és év végi átlagait. Állapítsuk meg, hogy egy adott nevű tanuló az első három díjazott között van-e? A bemenet alakja: egy ***n*** szám (1 ≤ ***n*** *≤* 100), amely a diákok számát jelöli, az ezt követő *n* sor a diákok adatait tartalmazza *Családnév Keresztnév* *Átlag* alakban. Az ezt követő sorban a keresett diák neve található *Családnév Keresztnév* alakban. A családnév, illetve a keresztnév maximális hossza 50 karakter.

**Példa**:

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 10  Albert Henrietta  5.2 Ambrus Anna  7.8 Bartok Boglar  6.3 Benedek Andor  5.2 Berkeczi Aron  9.8 Biro Daniel  6.8 Csoregi Norbert  3.5 Denes Eduard  7.2 Demeter Henrietta  5.1 Fazekas Gergo  5.8  Biro Daniel | nem |

1. Számoljuk meg egy adott szövegben a magánhangzók számát! A szöveg maximális hossza 1000 karakter.

**Példa**:

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| Szamoljuk meg: A maganhangzok szamat! | 11 |

1. Adott egy csoport év végi vizsgajegyeinek listája. Töröljük ki az 5-ösnél kisebb jegyeket a listából. (A végső listában a jegyek sorrendje nem lényeges). A módosítást a beolvasott tömbben kell elvégezni, nem elég csak kiírni az eredményt. A jegyek *double* típusú valós számok. Legtöbb 1 000 000 jegy lehet.

**Példa:**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 15  3 6 7 5 4 8 10 2 5 4 9 8 3 6 1 | 6 6 7 5 8 8 10 9 5 |

1. Adott ***n*** programozó fizetése (1 ≤ ***n*** ≤ 1 000 000). Válogassuk szét őket a fizetésük alapján. Az első csoportba azok kerülnek, akiknek a fizetésük nagyobb mint 1000 EUR, a másodikba a megmaradtak közül azok, akiknek a fizetése nagyobb mint 700 EUR, a harmadikban maradnak a többiek (a sorrend nem fontos). Törekedjünk egyetlen tömbbel dolgozni és abban elvégezni a szétválogatást, nem elég a kiíratásban csoportokra osztani a számokat.

**Példa:**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 15  990 360 870 1150 620 1300 750 890 560 720 940 310 420 510 1400 | 1300 1150 1400  990 870 750 720 890 940 360 620 560 310 420 510 |

1. Ismerjük két szám prímszám osztóit, melyeket 64-bites előjel nélküli egész típusban tárolhatunk. Határozzuk meg a két szám legkisebb közös többszörösének prímszám osztóit! (A két számot nem ismerjük, csupán a prím osztóikat)

**Példa:**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 2 5 11 29 3 5 11 | 2 3 5 11 29 |

1. Adottak az elsőéves egyetemisták névsorai (ábécé sorrendben), csoportonként. Állítsunk elő ezekből egyetlen névsort, mely szintén ábécé sorrendben van! Legtöbb 10 csoport lehet és egy csoportban nem lesz több, mint 100 diák. Egy diák neve legtöbb 100 karakterből áll.

**Példa:**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 3  5 Ambrus Anna Bartok Hanna Boglar Berkeczi Aron Kovacs David Sukosd Mate  7 Biro Daniel Csoregi Norbert Drig Eduard Dulf Henrietta Fazekas Gergo Gere Edit Pandi Aliz  6 Albert Nagy Henrietta Benedek Andor Gere Andor Lupás Monika Pulbere David Sallai Mark | Albert Nagy Henrietta Ambrus Anna Bartok Hanna Boglar Benedek Andor Berkeczi Aron Biro Daniel Csoregi Norbert Drig Eduard Dulf Henrietta Fazekas Gergo Gere Andor Gere Edit Kovacs Zoltan David Lupás Monika Pandi Aliz Pulbere David Sallai Mark Sukosd Mate |

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