XIII. Bolyai Farkas Multidiszciplináris Országos Tantárgyverseny
2018. május 4-6.
KALKULUSZ – Programozói verseny

I. Bűvös négyzet

Készítsen programot, amely egy szöveges állományból beolvassa az n értékét (`n>2 és n<51`), majd rendre egy `nxn` méretű négyzetes mátrix elemeit, eredményül kiír egy üzenetet (IGEN/NEM), hogy bűvös négyzetet alkot-e a beolvasott mátrix. Amennyiben nem bűvös négyzet, azt is írja ki, hogy ez a hányadik beolvasott elem után dönthető el.

Egy `nxn` méretű mátrixot bűvös négyzetnek nevezünk, ha rendelkezik a következő tulajdonságokkal:

- elemei az 1,2,3,…,n^2 számsorozat elemei,
- elemei páronként különbözőek,
- minden sora, oszlopa, valamint a fő- és mellék-átlója mentén az elemek összege ugyanannyi.

Példák:

Bemeneti állomány neve	**Bemeneti állomány tartalma**	**Kimenet**	**Magyarázat**
matrix0.txt	3 4 9 2 3 5 7 8 1 6	IGEN	
matrix1.txt	3 5 5 5 5 5 5 5 5 5	NEM, 2	A 2. beolvasott elem (1. sor, 2. oszlop) után már eldönthető, mert az 5 ismétlődik.

Megkötések:

- Az eredményeket írassa ki a standard kimenetre.
- Programját úgy készítse el, hogy az a következő 10 bemeneti állományra végezze el az ellenőrzést: `matrix0.txt, matrix1.txt, …, matrix9.txt` (A program eredménye mindig 10 sor. Minden sor egy adott bemeneti állomány eredményét tartalmazza.)
- A programot lássa el megjegyzésekkel (kommentekkel), amelyekben magyarázza a megoldását.

II. BéKá a szóboltban

BéKá akar egy szót (jelöljük *s*-sel), ezért elmegy a szóboltba, ahol mindenféle szó kapható. Amint belép a boltba, közlik vele, hogy aznap az *n* hosszú szavak kaphatók akciós áron. BéKá el is dönti, hogy ilyet vásárol, és azt is tudja, hogy amennyiben ugyanabból a szóból vásárol többet, az sokkal olcsóbb, mintha különböző szavakat kérne. Minimum hányat kell vásároljon BéKá valamelyik akciós szóból ahhoz, hogy annak betűiből ki tudja rakni a kívánt szót, vagyis az *s*-et.

Input:

A bemeneti állomány (beka.txt) első sorában az *s* karakterlánc található (hossza legalább 1, és legfeljebb 10.000.000), amely kizárólag az angol ábécé kisbetűiből tartalmaz karaktereket. A második sor az *n* értéket tartalmazza ($1 \leq n \leq 10.000.000$).

Output:

A kimenet egyetlen számot tartalmaz, ami azt jelenti, hogy legkevesebb hány azonos akciós szót kell BéKá-nak megvásárolnia. Ha nincs megoldás, akkor ezt a -1 kiírásával jelezzük.

1. példa

INPUT
```
kalkulus
5
```
OUTPUT
```
2
```
Magyarázat: BéKá vásárolhat `2` darabot, például az `5` hosszú `salku` szóból

2. példa

INPUT
```
kalkulus
4
```
OUTPUT
```
-1
```
Magyarázat: BéKá bármely `4` hosszú szóból, bármennyit vásárolna is, nem tudja kirakni a kívánt `kalkulus` szót.

3. példa

INPUT
```
kalkulus
10
```
OUTPUT
```
1
```
Magyarázat: BéKá vásárolhat `1` darabot, például a `10` hosszú `kalkulusxx` szóból

Megkötések:

- Az eredményeket írassa ki a standard kimenetre.
- Programját úgy készítse el, hogy az a következő 10 bemeneti állományra végezze el a feladatot: `beka0.txt, beka1.txt, …, beka9.txt` (A program eredménye mindig 10 sor. Minden sor egy adott bemeneti állomány eredményét tartalmazza.)
- A programot lássa el megjegyzésekkel (kommentekkel), amelyekben magyarázza a megoldását.
- Törekedjen hatékony megoldásra.

III. Regiszter

Egy állomány két listát tartalmaz, amelyeket egy üres sor választ el. Az első listában kétféle típusú sor van:

- három mezőt tartalmazó sor: név, születési év, elhalálozási év
- két mezőt tartalmazó sor: név, születési év.

A második lista minden sora két mezőt tartalmaz: név és elhalálozási év. A mezők fehér karakterekkel vannak elválasztva.

Írassuk ki azon személyek számát, akik teljes adatsorral rendelkeznek.

A bemenet helyes, a bemenő állomány legfeljebb 500.000 sort tartalmaz, a nevek legfeljebb 20 karakter hosszúságúak.

Bemenet	**Kimenet**	**Magyarázat**

`peter 1950`
`robert 1945`
`mary 1950 2002`
`alex 1965 1999`
`bob 1978`
`aliz 1978 2012`
 	
`bob 2000`
`xaver 2010`
`robert 2007`
	5	Teljes adatsorral a következő személyek rendelkeznek:
		
		`alex 1965 1999`
		`aliz 1978 2012`
		`bob 1978 2000`
		`mary 1950 2002`
		`robert 1945 2007`

Megkötések:

- Az eredményeket írassa ki a standard kimenetre.
- Programját úgy készítse el, hogy az a következő 3 bemeneti állományra végezze el a feladatot: `regiszter0.txt, regiszter1.txt, regiszter2.txt` (A program eredménye mindig 3 sor. Minden sor egy adott bemeneti állomány eredményét tartalmazza.)
- A programot lássa el megjegyzésekkel (kommentekkel), amelyekben magyarázza a megoldását.
- Törekedjen hatékony megoldásra.