

## Informatikából kitűzött feladatok



**I. 487.** Adott egy  $N$  elemű, pozitív egészekből álló számhalmaz ( $2 \leq N \leq 20$ ). Készítsünk programot, amely

a) megkeresi a legnagyobb olyan  $a$  számot a halmazban, amely minden nála kisebb halmazbeli számhoz relatív prím;

b) megadja a legkisebb olyan  $a$ -nál nagyobb  $b$  számot, amellyel kibővítve a halmazt az a) feladatrészt megoldása a hozzávett  $b$  szám lesz – illetve 0-t ad, ha nincs ilyen  $b$  szám.

A program a standard bemenet első sorából olvassa be  $N$  értékét, majd a következő sorból a halmazt alkotó  $N$  darab egész számot. A standard kimenet első sorába írja az a) feladatrészt keresett számot, a kimenet második sorába a b) feladatrészt megoldását.

Beküldendő egy `i487.zip` tömörített állományban a program forráskódja és egy rövid leírás, ami megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

Letölthető állomány: `i487beki.zip`.

**I. 488.** A biológiai kísérletek kiértékelését igyekeznek automatizálni. Egy négyzet alakú táptalajon tenyésztett baktériumtörzs példányait lefényképezik, majd a fotókat több lépésben digitálisan feldolgozzák. A baktériumpéldányok különböző méretűek, alakúak és helyzetűek.

A táptalaj fényképét egy képzeletbeli négyzetháló segítségével cellákra osztják. Egy  $50 \times 50$  cellából álló, az egyes baktériumpéldányokat már számokkal azonosítottan megjelenítő táblázat áll rendelkezésre a `meres.txt` tabulátorral tagolt, UTF-8 kódolású állományban.

A baktériumok az előfeldolgozott képen 1 és  $N$  közötti egész számmal ( $N \leq 50$ ) vannak azonosítva. Egy-egy példány összefüggő területet alkot, de egy cella csak egy baktériumhoz tartozik. Ha egy cellában nincs baktérium, akkor ott a táblázatban nincs adat. A mintán két baktériumpéldány látható, az 1-es és a 4-es sorszámú.

		4	4	4
			4	4
		1	1	4
1	1	1	1	4
	1	1		

Értékeljük ki és segítsük a további munkát táblázatkezelővel.

- Töltsük be a `meres.txt` szövegfájlt a táblázatkezelő egy munkalapjára az A1-es cellától kezdődően. A munkalap neve legyen **kep**. Munkánkat **bakterium** néven mentjük el a táblázatkezelő alapértelmezett formátumában.
- Hozzunk létre még két munkalapot **szamolas** és **eredmeny** néven. A **szamolas** munkalapon végezzük el minden, a megoldáshoz szükséges számolást.

Az **eredmény** munkalapon jelenítsük meg a kérdésekre adott válaszokat és eredményeket. Mind a két munkalap tartalmának értelmezését feliratokkal segítsük.

3. A **kep** munkalapon az  $A:AX$  oszlopok szélességét és az 1:50 sorok magasságát állítsuk be úgy, hogy a cellák (normál nézetben) négyzetek legyenek.
4. A mérési eredményeket szemléltessük feltételes formázással. A különböző baktériumpéldányok celláit más-más kitöltőszínnel jelenítsük meg, az üres cellák maradjanak fehérek.
5. Ha az **eredmény** munkalap egy adott cellájába beírunk egy sorszámot 1 és 50 között, akkor a mellette lévő cellában jelenjen meg, hogy ilyen sorszámú baktérium szerepel-e a képen.
6. Adjuk meg, hogy összesen hány baktérium van a képen.
7. Adjuk meg, hogy hányas sorszámú baktérium foglalja el a legnagyobb területet a képen és ez hány cella.
8. Adjuk meg annak a minimális méretű téglalapnak a szélességét és magasságát, amelyben a képen látható összes baktérium benne van.
9. Ha van, akkor adjuk meg két érintkező baktérium sorszámát, ha nincs, akkor írjuk ki, hogy „**Nincsenek egymással érintkező baktériumok.**”

Beküldendő egy tömörített `i488.zip` állományban a munkafüzet, valamint egy rövid leírás, amelyben szerepel az alkalmazott táblázatkezelő neve és verziószáma.

Letölthető állomány: `meres.txt`.

**I. 489 (É).** Tamás a kedvenc fényképeit rendezgette. Ezek mindegyikén szerepelnek emberek, akiket a programban egy ékezetek nélküli keresztnév vagy megszólítás azonosít. Előfordulnak a képeken Tamás és családtagjai (őket a programban az `En`, `Apa`, `Anya`, `Mama`, `Papa`, `Ocsi`, `Hugi` szöveggel jelöljük), valamint Tamás barátai, osztálytársai (a programban például `Anna`, `Zsolt`, `Judit`, `Evi`). Minden képen legalább egy személy szerepel, de egy képen természetesen egy személy csak egyszer. A képeken csak ismert, azonosítóval rendelkező személyek láthatók, minden azonosító egy szóból áll.

A képek némelyikéről tudni lehet, hogy hol vagy milyen alkalommal készült, melyeket a programban szintén ékezetek nélkül azonosítunk, például `Otthon`, `Erdei_suli`, `Tisza-to` (a többszavas neveket aláhúzásjellel kapcsoltuk össze). Sok esetben szerepel a képen a felvétel időpontja, például `2017.3.18`.

A képek készítésének helye, ideje és a rajta szereplő személyek megtalálhatóak a `kepszem.txt` szöveges állományban. A hiányzó időpont vagy helyszín helyett egy ('-') kötőjel szerepel a megfelelő helyeken.

A fájl első sorában a képek  $K$  száma ( $5 \leq K \leq 100$ ), és a következő  $K$  sorban egy-egy kép adatai találhatóak az alábbi minta szerint.

40

Szeged 2016.11.21. Juli Ocsi En

Fociedzes - Mama Ivan En Szabolcs

Otthon 2012.8.8. Szasa Laci Andras Hanna Zsuzsi Benedek Marci

...

Készítsünk programot, amely beolvassa a szöveges állományból az adatokat, és azok földolgozásával megoldja a következő feladatokat. Minden feladat be- és kimenete előtt írjuk ki a feladat sorszámát egy külön sorba a következő formában „3. feladat:”.

- Adjuk meg, hogy Tamás hány ismerőse szerepel a képeken. Az eredmény például a következő szöveg legyen: „Tamásnak 27 ismerőse szerepel a képeken.”.
- Határozzuk meg, hogy hány olyan kép van, ahol nem ismert a készítés helye és időpontja sem. Az eredményt a következőképp írjuk ki: „6 kép készítésének ideje és helye ismeretlen.”.
- Adjuk meg, hogy hányan szerepelnek azon a képen, ahol a legtöbb személy fordul elő. A kiírás a következők szerint történjen: „12 emberrel nincs több egy képen sem.”.
- Írjuk ki a képeken szereplő összes személy azonosítóját ABC-sorrendben egy sorban, vesszővel elválasztva és ponttal a végén. A kiírás formája: „A képeken szerepel: Anna, Balázs, Cecil ...”.
- Kérjük be az egyik ismerős nevét, és adjuk meg azokat az ismert időpontokat (az adatfájlban adott sorrendben), amikor Tamás (azonosítója En) és a bekért személy a fényképek szerint együtt volt. A bekérés formája: „Kérem adj meg egy szereplőt: Ocsi”. Az eredményt a következőképp írjuk ki: „Tamás és Ocsi közös időpontjai: 2019.5.10. 2020.7.7.” Ha nincs közös időpont, akkor a „Tamás és Ocsi nem szerepel közös ismert időpontban készített képen.” mondatot írjuk ki.
- Tamás szeretne egy családi tablót készíteni, ezért összegyűjti azokat a képeket, amelyek legalább két, Tamáson kívüli családtag szerepel (hogy Tamás szerepel-e vagy sem, az nem lényeges). Készítsünk egy listát ezekről a képekről úgy, hogy megadjuk a készítés helyét, a készítés idejét, valamint a családtagok azonosítóját. A listában a képek legyenek az adatfájlban szereplő sorrendben, az egyes képen szereplő személyek ABC-sorrendben.

A lista formája legyen a következő (az adatokat táblázatosan jelenítsük meg, az első két oszlop szélessége 16-16 karakter):

Családi tábla		
Otthon	2019.04.13.	Anya Apa En
Pecs	-	Hugi Ocsi Papa
...		

Beküldendő egy `i489.zip` tömörített állományban a program forráskódja és egy rövid leírás, ami megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

Letölthető állomány: `i489forras_beki.zip`.

**I/S. 37.** Béla a titkosszolgálatnak dolgozik, feladata egy szuperintelligens idegen civilizáció által küldött üzenetek feldolgozása. Az üzenetek  $N$  hosszú jelsorozatok, melyek '0'-t és '1'-et tartalmaznak. Mivel ez egy nem túl izgalmas munka, Béla úgy döntött megkeresi a számára érdekes részeket az üzenetben. Bélának az üzenet azon részei érdekesek, melyek '00'-val kezdődnek és '11'-gyel végződnek. Segítsünk

Bélának megmondani, hogy mennyi érdekes része van az üzenetnek, vagyis hány olyan  $x; y$  ( $x < y$ ) számpár van, amelynél az üzenet  $x$ -edik és  $(x + 1)$ -edik helyén '0',  $y$ -edik és  $(y + 1)$ -edik helyén '1' szerepel.

*Standard bemenet:* az első és egyetlen sora tartalmazza az üzenetet.

*Standard kimenet:* az első sora tartalmazza az érdekes részek számát.

*Korlátok:*  $1 \leq N \leq 10^5$ , időkorlát: 0,3 mp.

*Értékelés:* a pontok 50%-a kapható  $N \leq 10^4$  esetén.

*Példa:*

Bemenet	Kimenet
10001011011	4

Beküldendő egy `is37.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

**S. 136.** Adott egy  $N$  csúcú,  $M$  élű egyszerű gráf (nincs többszörös él vagy hurokél, de nem feltétlenül összefüggő). A csúcsokat 0-tól  $(N - 1)$ -ig indexeljük. A gráf összes csúcsa fekete vagy fehér. Jelölje  $f(x)$  az  $X$  csúcs színét. Cseresznyének hívunk egy  $(A, B, C)$  rendezett csúcshármaszt, ha páronként különbözőek és létezik az  $A - B$ , valamint a  $B - C$  csúcspárok közt él. Egy  $(A, B, C)$  cseresznye finom, ha  $f(A) = f(C)$  és  $f(A) \neq f(B)$ . Adjuk meg, hogy egy él behúzásával legföljebb mennyire növelhető a finom cseresznyék száma. ( $A$  és  $B$  csúcs összeköthető egy éllel, ha  $A \neq B$ , és eddig nem létezett köztük él.)

*Standard bemenet:* az első sor tartalmazza  $N$ -et és  $M$ -et. A következő sor  $N$  darab számot tartalmaz: az  $i$ -edik szám az  $i - 1$  indexű csúcs színét határozza meg, 0 ha fekete, 1 ha fehér. A következő  $M$  sor mindegyike két számot tartalmaz. Az  $i$ -edik sor az  $i$ -edik él két végpontjának csúcindexét adja meg.

*Standard kimenet:* a maximálisan elérhető finom cseresznyék száma.

*Korlátok:*  $3 \leq N \leq 10^5$ ,  $0 \leq M \leq \min(10^5, N^2 - N - 1)$ . Időkorlát: 0,3 mp.

*Értékelés:* A pontok 50%-a kapható, ha a gráf fa.

*Példa:*

Bemenet (a / jel a sortörést helyettesíti)	Kimenet
5 4 0 1 1 1 0 0 1 / 1 4 / 3 0 / 0 2	12

Beküldendő egy `s136.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

**A feladatok megoldásai regisztráció után a következő címen tölthetők fel:**

<https://www.komal.hu/munkafuzet>

**Beküldési határidő: 2019. október 10.**