

## Nemes Tihamér Online Programozási Verseny 2017/2018

Készült az Eötvös Loránd Tudományegyetem, EFOP-3.4.4-16-2017-00006 számú, „Belépő a tudás közösségébe, MTMI szakok és pályák népszerűsítése a középiskolások körében” című projektje keretében.



## Hajók

Egy szállítmányozási vállalat hajókkal szállít konténereket különböző városokba. Minden hajóra csak ugyanoda viendő konténereket lehet felrakni, legfeljebb  $K$  darabot. Minden hajóra a lehető legtöbb konténert tesznek fel, de ha egy városba akár csak egyetlen konténert kell szállítani, akkor is kell oda küldeni hajót.

Készíts programot, amely megadja azt a három várost, ahova a hajókon a legtöbb szabad hely marad!

### Bemenet

A *standard bemenet* első sorában a konténerek száma ( $1 \leq N \leq 10\,000$ ) és az egyes hajókra pakolható konténerek maximális száma ( $1 \leq K \leq 100$ ) van. A további  $N$  sor mindegyikében egy-egy város neve van (az angol ábécé betűivel írva), ahova az egyes konténereket vinni kell.

### Kimenet

A *standard kimenet* három sorába három város nevét, az oda küldendő hajók számát és a szabad helyek számát kell írni, ahova a hajókon a legtöbb szabad hely marad, a szabad helyek száma szerint csökkenő sorrendben! Az azonos szabad hely számúakat hajók száma szerint csökkenő sorrendben! Több megoldás esetén bármelyik kiírható, kevesebb város esetén kevesebb sort kell kiírni!

### Példa

Bemenet

```
13 2
London
Rotterdam
London
Hamburg
Hamburg
Stockholm
London
Stockholm
Hamburg
Stockholm
London
London
Stockholm
```

Kimenet

```
London 3 1
Hamburg 2 1
Rotterdam 1 1
```

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Kerékpárverseny

Egy útvonal Budapestről indul és  $N$  további várost érint. Ismerjük minden szomszédos várospár távolságát. Egy kerékpárversenyt szeretnénk szervezni, amelyet két tetszőleges város között rendezünk, de pontosan  $K$  kilométeres verseny legyen! A kerékpárosok biztonsága érdekében azokat az útszakaszokat le kell zárni, ahol a verseny folyik – a cél, hogy a lezárt útszakaszok száma minimális legyen.

Készíts programot, amely megadja, hogy mely városból induljon a verseny és hány útszakaszt kell lezárni!

### Bemenet

A *standard bemenet* első sorában a további városok száma ( $1 \leq N \leq 100\,000$ ) és a kerékpárverseny hossza ( $1 \leq K \leq 100\,000$ ) van. A további  $N$  sor mindegyikében egy-egy város előző várostól vett távolsága van ( $1 \leq T_i \leq 1000$ ).

### Kimenet

A *standard kimenet* első sorába a lezárandó útszakaszok  $U$  minimális számát kell írni, amelyek összhossza  $K$ . A második sorba pedig az első város sorszámát kell írni ( $1 \leq A \leq N$ ), ahonnan a verseny kezdődik! Több megoldás esetén bármelyik kiírható. Ha nincs megoldás, akkor mindkét sorban 0 szerepeljen!

### Példa

Bemenet	Kimenet
8 50	3
40	2
12	
28	
10	
10	
10	
10	
10	

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Születésnapok

Ismerjük  $N$  ember születésnapját és névnapját.

Készíts programot, amely megadja azokat az embereket, akinek egy adott évben a születésnapja a legközelebb van a legközelebbi névnapjához (a legközelebbi névnap lehet az előző és a következő évben is)!

### Bemenet

A *standard bemenet* első sorában az emberek száma ( $1 \leq N \leq 1000$ ) és egy évszám van ( $1900 \leq \text{ÉV} \leq 2100$ ). A további  $N$  sor mindegyikében egy-egy ember születésnapja ( $1 \leq \text{SHó}_i \leq 12$ ,  $1 \leq \text{SNap}_i \leq \text{az SHó}_i$  hónap napszáma) és névnapja ( $1 \leq \text{NHó}_i \leq 12$ ,  $1 \leq \text{Nnap}_i \leq \text{az SHó}_i$  hónap napszáma) van.

### Kimenet

A *standard kimenet* első sorába azon emberek DB számát kell írni, akiknek az  $\text{ÉV}$  évbeli születésnapjához legközelebb van névnapjuk, a következő sorba pedig DB egész számot szóközzel elválasztva, ezen emberek sorszámait, növekvő sorrendben!

### Példa

Bemenet	Kimenet
5 2017	2
1 20 6 18	2 4
12 24 1 5	
7 12 12 10	
6 22 6 10	
6 22 7 10	

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Vitorlásverseny

Ismerjük az időjárás előrejelzést  $N$  napra. Vitorlásversenyt szeles időben lehet lebonyolítani, szélcsendben vagy viharban nem. Egy versenyt pontosan egymás utáni  $K$  napon lehet megrendezni

Készíts programot, amely megadja, hogy az  $N$  nap alatt legfeljebb hány  $K$  napos versenyt lehet rendezni és közülük legfeljebb hányat hosszabbíthatnak meg a rendezők egy nappal!

### Bemenet

A *standard bemenet* első sorában a napok száma ( $1 \leq N \leq 100\,000$ ) és a versenyek hossza van ( $1 \leq K \leq N$ ). A második sorban pontosan  $N$  karakter van, az  $i$ . karakter az  $S$  betű, ha az  $i$ . napon szeles idő várható, szélcsend esetén  $C$  betű, vihar esetén pedig  $V$  betű.

### Kimenet

A *standard kimenet* első sorába a megrendezhető  $K$  napos versenyek számát kell írni! A második sorba ezek közül azon versenyek száma kerüljön, amelyek egy nappal meghosszabbíthatók!

### Példa

Bemenet	Kimenet
15 3	3
SSSCSSVSSSSSSSS	2

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Hegycsúcsok

Egy útvonal során adott távolságoként  $N$  helyen mérték a felszín tengerszint feletti magasságát.

Csúcsnak nevezzük azon köztes pontokat, ahol a mért érték nagyobb az előző és a következő mérésnél is. Egymáshoz legközelebbi  $K$  csúcsnak azon csúcsokat nevezzük, amelyek közül a két legtávolabbi távolsága a lehető legkisebb.

Készíts programot, amely meghatározza az egymáshoz legközelebb levő  $K$  hegycsúcsot!

### Bemenet

A *standard bemenet* első sorában a mérések száma ( $1 \leq N \leq 10\,000$ ) és a  $K$  ( $2 \leq K \leq N/2$ ) érték szerepel. A következő sorban az  $N$  mérés értéke van ( $0 \leq M_i \leq 10\,000$ ).

### Kimenet

A *standard kimenet* első sorába az egymáshoz legközelebb levő  $K$  csúcs legkisebb távolságát kell írni! A második sorba egy ilyen  $K$  csúcsból álló sorozat tagjainak mérési sorszáma kerüljön, növekvő sorrendben! Több megoldás esetén bármelyik megadható. Ha nincs  $K$  csúcs, akkor az egyetlen sorba  $-1$ -et kell kiírni!

### Példa

Bemenet

20 3

1 1 5 4 3 5 3 5 1 1 1 1 2 1 2 1 2 1 1 1

Kimenet

4

13 15 17

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Kincsek a hegyoldalon

Egy jobbra-lefelé lejtő hegyoldalon kincseket helyeztünk el, amelyekből a lehető legtöbbet egyetlen szánkóval szeretnénk összegyűjteni. A szánkóval a bal felső sarokból indulhatunk, és lejtő irányba (azaz vagy jobbra, vagy lefelé) haladhatunk. Amelyik mezőn átmegyünk, az ott levő kincset felvesszük. A hegyoldalon kijelöltek néhány gyűjtőpontot, a szánkóval valamelyikhez el kell jutnunk (és onnan tovább nem mehetünk).

Készíts programot, amely megadja azt a gyűjtőpontot, ahova a legtöbb kincset vihetjük, valamint az oda vihető kincsek maximális számát!

### Bemenet

A *standard bemenet* első sorában a hegyoldal sorai és oszlopai száma ( $1 \leq N, M \leq 1000$ ), a kincsek száma ( $1 \leq K \leq 100\,000$ ), valamint a gyűjtőhelyek száma ( $1 \leq G \leq 1000$ ) van. A következő  $K$  sor mindegyikében egy-egy kincs sor- és oszlopindexe van ( $1 \leq S_i \leq N, 1 \leq O_i \leq M$ ), ahol kincs található. Az utolsó  $G$  sorban a gyűjtőhelyek sor- és oszlopindexei vannak ( $1 \leq GS_i \leq N, 1 \leq GO_i \leq M$ ). A gyűjtőhelyeken biztosan nincs kincs.

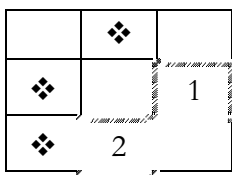
### Kimenet

A *standard kimenet* első sorába a valamely gyűjtőhelyre szállítható kincsek maximális számát kell írni! A második sorba annak a gyűjtőhelynek a sor- és oszlopindexét kell írni, ahova a legtöbb kincset lehet vinni (több megoldás esetén bármelyiket)!

### Példa

Bemenet	Kimenet
3 4 3 2	2
1 2	3 2
2 1	
3 1	
3 2	
2 3	

Az alábbi mintán látható, hogy a két gyűjtőhelyre hány kincset lehet vinni:



### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Sikertelen napok

Egy üzletben minden nap feljegyezték a napi forgalmat. Különösen sikeres napnak tekintik azokat a napokat, amelyeken a napi forgalom nagyobb volt, mint bármely korábbi napon.

Készíts programot, amely megadja napok leghosszabb egymásutáni sorozatát, ahol nem volt különösen sikeres nap!

### Bemenet

A *standard bemenet* első sorában a napok száma van ( $1 \leq N \leq 100\,000$ ). A következő  $N$  sor mindegyikében egy-egy nap bevétele van ( $0 \leq B_i \leq 100\,000$ ).

### Kimenet

A *standard kimenet* első sorába az egymásutáni nem különösen sikeres napok maximális számát kell írni! A második sorba egy ilyen napsorozat első napja sorszáma kerüljön (több megoldás esetén bármelyik)! Ha nem volt ilyen napsorozat, akkor a második sorba -1-et írjunk!

### Példa

Bemenet	Kimenet
8	2
100	5
120	
110	
130	
<b>120</b>	
<b>125</b>	
150	
100	

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB



## Virágárusok

Egy osztály a szalagavató bálra virágokat szeretne tenni a díszterembe. Eldöntötték, hogy milyen virágból hány darabra van szükségük. A közelben két virágárus van, akitől vehetnek virágokat. Mindkettőről tudjuk, hogy milyen virágból hány van az üzletükben. A két árus közül az egyiket kell választanunk!

Készíts programot, amely megadja, hogy melyik árustól vegyék a virágokat, hogy a tervezetthez képest a lehető legkevesebb virág hiányozzon!

### Bemenet

A *standard bemenet* első sorában a megvenni szándékozott virágfajták száma van ( $1 \leq N \leq 10\,000$ ). A következő  $N$  sor mindegyikében egy-egy virág neve (az angol ábécé betűivel írva) és a belőle vásárolni kívánt virágok száma van (név szerinti ábécésorrendben). Ezután következő sorban az első árus virágfajtái száma ( $1 \leq A_1 \leq 10\,000$ ) van, amit  $A_1$  sorban a virágai neve és darabszáma követ (név szerinti ábécésorrendben). Az utolsó csoport első száma a második árus virágfajtái száma ( $1 \leq A_2 \leq 10\,000$ ), amit  $A_2$  sorban a virágai neve és darabszáma követ (szintén név szerinti ábécésorrendben).

### Kimenet

A *standard kimenet* első sorába a virágárus sorszámát kell írni, akitől a virágokat vásároljuk! A második sorba azon virágok minimális száma kerüljön, amiket így nem tudunk megvenni!

### Példa

Bemenet

```
3
ibolya 5
kankalin 2
orgona 3
4
ibolya 4
kankalin 1
liliom 20
orgona 2
3
amarillisz 3
kankalin 20
orgona 30
```

Kimenet

```
1
3
Így 1 ibolya, 1 kankalin, 1 orgona hiány lesz.
Ha a 2. árust választanánk, akkor 5 virág (mind
ibolya) hiányozna.
```

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Betűk intervalluma

Ismerünk egy szöveget és betűknek egy  $R$  részhalmazát.

Készíts programot, amely megadja egy hosszú szöveg legrövidebb olyan intervallumát, ahol  $R$  összes betűje előfordul!

### Bemenet

A *standard bemenet* első sorában a szöveg hossza van ( $1 \leq H \leq 10\,000$ ). A második sor a szöveg karaktereit tartalmazza (az angol ábécé kis- és nagybetűit, továbbá szóközöket és írásjeleket tartalmazhat). A harmadik sorban az  $R$  betűhalmaz karakterei találhatók (legfeljebb 26 betű az angol ábécé kisbetűi közül).

### Kimenet

A *standard kimenet* első sorába a legrövidebb olyan intervallum első és utolsó karakterének sorszámát kell írni, amelyben az  $R$  betűhalmaz minden betűje előfordul! (A nagy- és kisbetűk ekvivalensek, elég minden betűből az egyik.) Ha nincs ilyen intervallum, akkor a sor egyetlen -1-et tartalmazzon!

### Példa

Bemenet	Kimenet
20	11 19
Abcde, Fghuoi aeiou	EDCBA.

### Korlátok

Időlimit: 1 mp.

Memórialimit: 32 MiB

## Legtöbb csúcsot tartalmazó terület

Egy téglalap alakú területen ismerjük  $N \cdot M$  pont tengerszint feletti magasságát. Csúcsnak nevezük azokat a pontokat, amelyek nagyobbak négy szomszédjuknál. A terület szélein biztos nincs csúcs.

Készíts programot, amely megad egy pontosan  $K$  pontot tartalmazó téglalap alakú területet, amelyen a lehető legtöbb csúcs van!

### Bemenet

A *standard bemenet* első sorában a terület sorai és oszlopai száma ( $1 \leq N, M \leq 1000$ ), valamint a  $K$  érték ( $1 \leq K \leq 1000$ ) van. A következő  $N$  sor mindegyikében egy-egy pont tengerszint feletti magassága található ( $1 \leq \text{Mag}_{i,j} \leq 9000$ ).

### Kimenet

A *standard kimenet* első sorába egy pontosan  $K$  pontot tartalmazó téglalap alakú terület bal felső és jobb alsó sarkának sor- és oszlopindexeit kell írni, amelyen a lehető legtöbb csúcs van! Több megoldás esetén bármelyik kiírható.

### Példa

Bemenet					Kimenet					
5			6			8	1	2	4	3
1	<b>1</b>	<b>1</b>	1	1	1	1				
1	<b>1</b>	<b>2</b>	1	1	1	1				
1	<b>2</b>	<b>1</b>	1	2	1	1				
1	<b>1</b>	<b>2</b>	1	1	1	1				
1	1	1	1	1	1	1				

### Korlátok

Időlimit: 1 mp.

Memórialimit: 32 MiB

## Földrengések

Földrengéseket követnek utórengések. Az utórengés a földrengéstől vagy az előző utórengéstől legfeljebb  $K$  percen belül történik és a földrengésnél kisebb erősségű (azaz egy földrengés egyes utórengései a földrengéstől  $K$ -nál több percre is lehetnek). Ha  $K$  percen belül egy földrengésnél erősebb vagy vele egyforma rengést mérnek, az már új földrengésnek számít. Percenként mérjük a rengések erősségét (ez 0, ha épp akkor nincs rengés).

Készíts programot, amely megszámolja, hogy a vizsgált időszakban hány földrengés volt!

### Bemenet

A *standard bemenet* első sorában a vizsgált percek száma ( $1 \leq N \leq 100\,000$ ) és a  $K$  szám ( $1 \leq K \leq 100$ ) van. A második sor az  $N$  mérés értékét tartalmazza ( $0 \leq S \leq 1000$ ).

### Kimenet

A *standard kimenet* első sorába az érzékelt földrengések számát kell írni!

### Példa

Bemenet	Kimenet
13	3 3
0 5 0 1 0 6 6 2 1 1 0 0 2	

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MiB

## Játékállások

$N$  személy kétszemélyes játékot játszik. Ha  $A$  legyőzi  $B$ -t, akkor  $A$  1 pontot kap,  $B$ -től pedig 1 pontot levonnak.

Készíts programot, amely minden játszma eredménye után megadja az összes maximális pontszámú játékost, sorszám szerinti sorrendben!

### Bemenet

A *standard bemenet* első sorában a játékosok száma ( $2 \leq N \leq 250$ ) és a játszmák száma ( $1 \leq M \leq 10\,000$ ) van. A következő  $M$  sor mindegyike egy játszma eredményét tartalmazza ( $1 \leq A_i \neq B_i \leq N$ ), ami azt jelenti, hogy  $A_i$  legyőzte  $B_i$ -t.

### Kimenet

A *standard kimenet*  $M$  sorába az egyes játszmák utáni első helyen állók sorszámát kell írni, sorszám szerint növekvő sorrendben!

### Példa

Bemenet	Kimenet
4	6 1
1	2 1
3	4 3
3	1 3
2	3 2
2	4 2 3
4 1	

### Korlátok

Időlimit: 1 mp.

Memórialimit: 32 MiB

## Csoportképek

Egy eseményen  $N$  ember vesz részt. Mindenkinek ismerjük az érkezési és távozási időpontját. Az eseményen minden időpontban készül egy csoportkép a jelenlevőkről, de csak akkor, ha a jelenlevők száma legfeljebb  $F$ , mert különben nem férnek rá egy képre. (Nem számít az, hogy ha már korábban készült csoportkép a társaságról, tehát ha két egymást követő időpontban senki sem érkezik vagy távozik, akkor is készül mindkét időpontban csoportkép.) Egy ember az érkezésének és távozásának időpontjában is jelen levőnek számít. Készül  $M$  darab videófelvétel is. A videókon látszódnak a fényképek készülésekor a vaku villanások (ha pont a videó kezdetén vagy végén villan a vaku, akkor is látszik).

Készíts programot, amely megadja minden videóra, hogy hányszor villan a vaku!

### Bemenet

A *standard bemenet* első sorában az emberek száma ( $1 \leq N \leq 100\,000$ ) és az egy képre elférők maximális száma ( $1 \leq F \leq 100\,000$ ) van. A következő  $N$  sorban egy-egy ember érkezési és távozási időpontja található ( $1 \leq E_i < T_i \leq 100\,000$ ). A következő sorban a videófelvételek száma van ( $1 \leq M \leq 50\,000$ ). Az utolsó  $M$  sorban a videófelvételek kezdete és vége található ( $1 \leq K_i < V_i \leq 100\,000$ , ahol  $V_i < K_{i+1}$ ).

### Kimenet

A *standard kimenet*  $i$ -edik sorába az  $i$ -edik videófelvétel alatti vaku villanások számát kell írni!

### Példa

Bemenet	Kimenet
6	2 1
1	3 3
6	7
2	4
5	8
2	4
3	5
2	
1	3
4 7	

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MiB

## Karavánok találkozása

Egy sivatagban  $N$  oázis van, amelyek között több napos utat igénylő karavánutak vezetnek.  $K$  karaván indul útnak adott időpontban. Ha egy karaván egy oázisba érkezik, akkor egy napot ott tölt, azaz két éjszakára szállást kell nekik biztosítani. A karavánoknak az útjuk első és utolsó oázisában nem kell szállást biztosítani (azaz ott már nem találkoznak senkivel)!

Készíts programot, amely megadja, hogy melyik karaván hány másikkal találkozik (együtt tölt éjszakát) az útja során!

### Bemenet

A *standard bemenet* első sorában az oázisok száma ( $1 \leq N \leq 100$ ), a karavánutak száma ( $1 \leq M \leq 10\,000$ ) és a karavánok száma ( $1 \leq K \leq 100$ ) van. A következő  $M$  sorban egy-egy karavánút két végpontjának sorszáma ( $1 \leq A_i \neq B_i \leq N$ ) és az útszakasz hossza ( $1 \leq H_i \leq 1000$ ) van. Az utolsó  $K$  sorban a karavánok útvonala szerepel. Minden sor első száma a karaván indulási napjának sorszáma ( $1 \leq Nap_i \leq 100$ ), az érintett oázisok száma ( $1 \leq O_i \leq N$ ), majd az érintett oázisok sorszámait ( $1 \leq S_{i,j} \leq N$ ), a haladás sorrendjében!

### Kimenet

A *standard kimenet*  $i$ -edik sorába azt a számot kell írni, ahány karavánnal találkozik az útja során az  $i$ -edik karaván!

### Példa

bemenet	kimenet
6 9 4	1
1 2 5	1
1 3 3	0
2 3 2	0
2 4 5	
2 5 10	
3 5 5	
4 5 2	
4 6 4	
5 6 3	
1 4 1 2 4 6	
1 4 1 3 2 5	
3 4 1 3 5 4	
4 3 5 4 6	

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Legyőzött és legyőző versenyző

Egy síugró versenyen minden versenyző akkor indulhat, ha az összes korábban indult versenyző befejezte már a versenyzését. Egy versenyző akkor győz le egy másikat, ha ugrásával magasabb pontszámot ér el.

Készíts programot, amely megadja minden indulóra, hogy az előtte indulók közül ki volt a hozzá legközelebb indult, általa legyőzött versenyző, valamint az utána indulók közül a hozzá legközelebb indult olyan, aki legyőzte őt!

### Bemenet

A *standard bemenet* első sorában a versenyzők száma van ( $1 \leq N \leq 100\,000$ ). A következő  $N$  sor mindegyike egy-egy versenyző pontszámát tartalmazza ( $1 \leq P_i \leq 100\,000$ ), az indulás sorrendjében.

### Kimenet

A *standard kimenet*  $i$ . sorába az  $i$ . induló versenyző előtteiek közül az utolsónak induló, általa legyőzött versenyző sorszáma, valamint az utána indultak közül a leghamarabb induló őt legyőző versenyző sorszáma kerüljön! Ha valaki senki őt megelőzőt nem győzött le, akkor a sor első száma  $-1$  legyen! Ha valakit senki őt követőt nem győzött le, akkor a második szám legyen  $-1$ !

### Példa

Bemenet	Kimenet
7	-1 3
3	-1 3
2	2 4
5	3 -1
6	-1 6
1	5 -1
4	5 -1
3	

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MiB



## Számok összege

Készíts programot, amely megadja egy rendezett számsorozat három elemét, amelyek összege pontosan egy előre megadott  $X$  érték!

### Bemenet

A *standard bemenet* első sorában a számsorozat hossza ( $1 \leq N \leq 1000$ ) és az  $X$  érték ( $1 \leq X \leq 100\,000$ ) van. A következő sor az  $N$  számot tartalmazza, növekvő sorrendben ( $1 \leq s_i \leq 100\,000$ ).

### Kimenet

A *standard kimenet* első sorába a sorozat három olyan elemét kell írni növekvő sorrendben, amelyek összege pontosan  $X$ ! Ha nincs három ilyen szám, akkor egyetlen  $-1$ -et kell írni!

### Példa

Bemenet	Kimenet
7	25 4 6 15
3 4 6 9 11 12 15	

### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MiB