

Informatikából kitűzött feladatok



I. 586. Judit és Gábor az órák közötti szünetben a táblánál játszanak. Az üres táblára felírnak háromjegyű pozitív egész számokat, összesen N darabot. Ezután felváltva letörölnék egy általuk választott számból egy számjegyet a szám elejéről vagy végéről, de csak akkor, ha a megmaradt szám osztója valamelyik táblán lévő számnak. Ha a törlés során olyan kétjegyű szám maradna, ami 0-val kezdődne, akkor a 0-t is letörlik. Tehát a 205 számból a 2-es törlése után 5 lesz. Az egyjegyű számokat nem törlik le.

A játék véget ér, ha már csak egyjegyű számok vannak, illetve akkor is, ha már nem lehet a tábláról újabb számot letörölni. Készítsünk programot, amely modellezi a fenti játékot úgy, hogy a bemenetként megadott N számhoz megad egy tetszőleges törléssorozatot egészen a játék végéig, vagyis amikor már nem lehet újabb számot törölni.

A program a standard bemenet első sorából olvassa be N értékét ($3 \leq N \leq 20$), majd a második sorából az N darab számot. A program a standard kimenet egymást követő soraiba írja ki a játék egy-egy törlése után a táblán látható számokat. A számok sorrendje a kiírásakor ne változzon. Amennyiben nincs törlési lehetőség, úgy a kimenetre ne írjunk semmit.

Példa:

Bemenet	Kimenet (a / jel sortörést helyettesít)
5 / 941 517 467 983 708	
5 / 120 250 340 270 155	120 250 340 270 155 / 120 250 340 270 15 120 250 40 270 15 / 20 250 40 270 15 20 250 40 270 1 / 2 250 40 270 1

Beküldendő egy tömörített `i586.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 587. Adott a síkon 6 darab (P_1, P_2, \dots, P_6) pont, mindegyik pont x és y koordinátája a $[-500; 500]$ (méter) intervallumba esik. Minden pont pillanatnyi sebességvektora a rákövetkező pont irányába mutat, tehát mindig afelé mozog: a P_1 a P_2 irányába, a P_2 a P_3 irányába, ..., a P_5 a P_6 irányába, végül a P_6 a P_1 irányába, méghozzá azonos nagyságú $0 < v \leq 1,5 \left(\frac{\text{m}}{\text{s}}\right)$ sebességgel. Készítsünk szimulációt, amely másodperc időközönként közelíti a fenti folyamatot. Számítsuk ki az egyes pontok koordinátáit a kiindulási pontból a $t = 0$ kezdőértékkel a következő 1000 másodpercben és jelenítsük meg diagramon a mozgásukat.

1. Hozzuk létre a táblázatkezelő egy üres munkafüzetében az `Alapadatok` munkalapot.
2. Készítsük el az `A1:M1` cellák szövegét a `minta` szerint. Állítsuk az `A2:M2` cellák formátumát a `minta` szerint, adataik két tizedes pontossággal jelenjenek meg.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	x_5	y_5	x_6	y_6	v
2													

- Hozzuk létre a Pozíciók munkalapot. A munkafüzetben ne legyen több munkalap.
- Írjunk be a feltételeknek megfelelő adatokat az Alapadatok munkalap A2:M2 celláiba. A Pozíciók munkalap A1:M1002 celláiba kerüljenek a pontok kiszámított koordinátái az első 1000 másodpercre. Segédszámításokat az M oszloptól jobbra végezhetünk. Ezen a munkalapon nem követelmény a minta szerinti formázás.

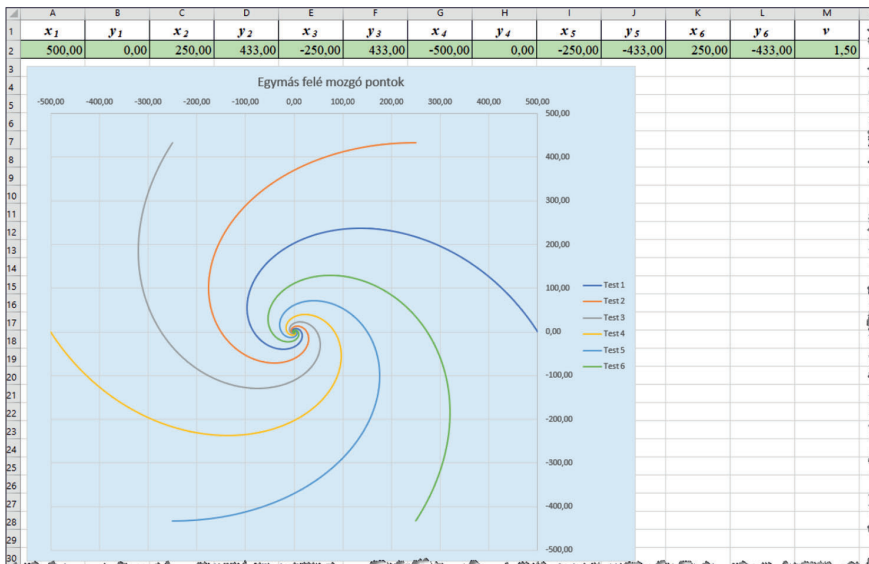
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	t	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	x_5	y_5	x_6	y_6
2	0	500,00	0,00	250,00	433,00	-250,00	433,00	-500,00	0,00	-250,00	-433,00	250,00	-433,00
3	1	499,25	1,30	248,50	433,00	-250,75	431,70	-499,25	-1,30	-248,50	-433,00	250,75	-431,70
4	2	498,50	2,60	247,00	433,00	-251,50	430,40	-498,50	-2,60	-247,00	-433,00	251,50	-430,40
5	3	497,74	3,89	245,50	432,99	-252,24	429,10	-497,74	-3,89	-245,50	-432,99	252,24	-429,10
6	4	496,98	5,18	244,00	432,98	-252,98	427,79	-496,98	-5,18	-244,00	-432,98	252,98	-427,79
7	5	496,22	6,48	242,50	432,96	-253,72	426,49	-496,22	-6,48	-242,50	-432,96	253,72	-426,49
8	6	495,45	7,76	241,00	432,94	-254,45	425,18	-495,45	-7,76	-241,00	-432,94	254,45	-425,18
9	7	494,68	9,05	239,50	432,92	-255,18	423,87	-494,68	-9,05	-239,50	-432,92	255,18	-423,87
10	8	493,91	10,34	238,00	432,89	-255,90	422,55	-493,91	-10,34	-238,00	-432,89	255,90	-422,55
11	9	493,13	11,62	236,50	432,86	-256,63	421,24	-493,13	-11,62	-236,50	-432,86	256,63	-421,24

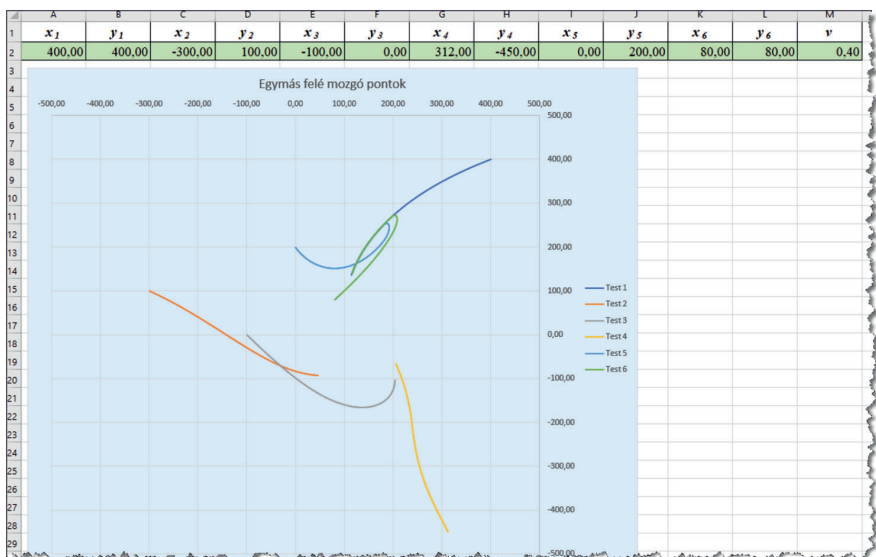
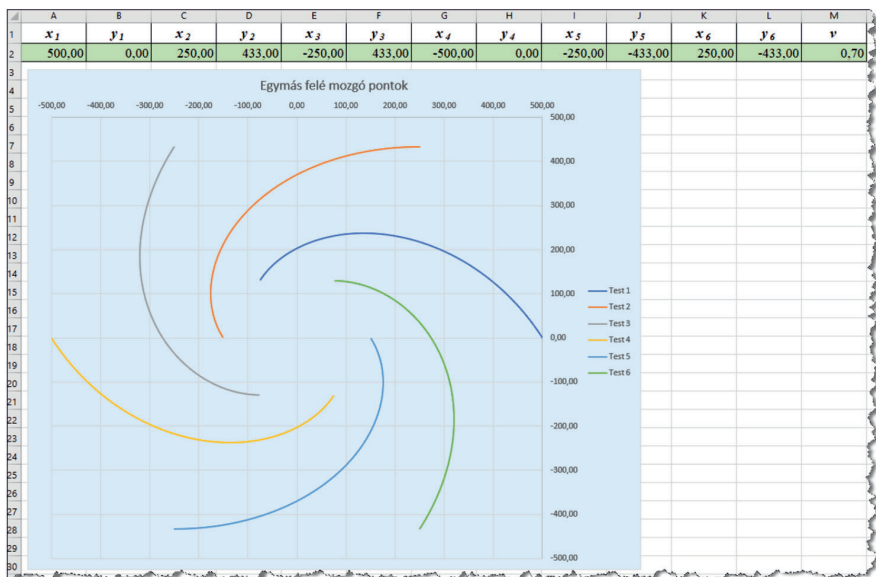
- Végül az Alapadatok munkalapon maximum az A:J oszlopok szélességében az alapadatok alatt jelenítsük meg a mozgásvonalakat diagramon, úgy, hogy lehetőleg görgetés nélkül teljesen megjelenjen a képernyőn.

A megoldásban saját függvény vagy makró nem használható.

Beküldendő egy tömörített i587.zip állományban a táblázatkezelő munkafüzet, illetve egy rövid dokumentáció, amelyben szerepel az egymás utáni adatsorok kiszámításának elve, a megoldáskor alkalmazott táblázatkezelő neve, verziószáma.

Minták:





I. 588 (É). A teniszezők világranglistáját heti gyakorisággal frissítik. A rendelkezésre álló `vilagelso.txt` és `jatekos.txt` állományok a ranglista bevezetése óta napjainkig tartó időszak férfi első helyezettjeinek adatait tartalmazzák.

A feladat megoldásához a digitális kultúra emelt szintű érettségien használható XAMPP használatát javasoljuk.

1. Készítsünk új adatbázist **tenisz** néven. A mellékelt két – tabulátorokkal tagolt, UTF-8 kódolású – szöveges állományt importáljuk az adatbázisba az általunk létrehozott táblákba, a fájlnevel azonos néven (**vilagelso**, **jatekos**). Az állományok első sora a mezőneveket tartalmazza. A létrehozás során állítjuk be a megfelelő típusokat és a kulcsokat. A táblák kialakításához vegyük figyelembe az alábbi táblaleírásokat és kapcsolatokat.

Táblák:

vilagelso (id, jatekosid, kezdete, vege)

- id a folyamatos világsőrség azonosítója (szám), ez a kulcs;
- jatekosid az aktuális világsőrső játékos azonosítója (szám), idegen kulcs;
- kezdete a folyamatos világsőrség kezdő dátuma (dátum);
- vege a folyamatos világsőrség befejező dátuma (dátum);

jatekos (id, nev, orszag)

- id a világsőrső férfi játékos azonosítója (szám), ez a kulcs;
- nev a játékos neve (szöveg), az adatbázisban névrokonok nincsenek;
- orszag a játékos születési országa (szöveg).



A következő feladatokat megoldó SQL parancsokat rögzítsük a

`tenisz_megoldas.sql`

nevű állományban a feladatok végén zárójelben megadott névvel. A javítás során csak ennek az állománynak a tartalma lesz értékelve. Ügyeljünk arra, hogy a lekérdezésekben pontosan a kívánt mezők szerepeljenek, felesleges mezőket ne jelenítsünk meg.

2. A világranglistát hetente frissítik és a listavezetést hetekben mérik. Adjuk meg lekérdezés segítségével időrendben, hogy ki hány hétig volt ranglistavezető. (2listavezetes)
3. Általában a játékosok megszakításokkal, de többször kerülnek ranglistavezető pozícióba. Listázzuk lekérdezés segítségével azokat a játékosokat, akik csak egyetlenegy időszakban vezették a világranglistát. A listában a játékos neve, országa, a ranglistavezetés kezdő és befejező dátuma jelenjen meg. (3egyszer)
4. Lekérdezéssel adjuk meg az első 20 – összesítve legtöbb héten át – ranglistavezető játékos nevét, országát és a hetek számát, utóbbi szerint csökkenő sorrendben. (4legnagyobb)
5. Adjuk meg lekérdezés segítségével, hogy 2000-ben, év végén melyik játékos vezette a világranglistát. (5evvege)

6. Lekérdezéssel határozzuk meg, hogy *Roger Federer* utolsó világrangsorosa után melyik játékos volt először listavezető. (6federerutan)
7. Készítsünk lekérdezést, amely kelistázza azokat a játékosokat, akik 2010. és 2020. között voltak ranglistavezetők. A listában a játékosok neve és országa jelenjen meg ismétlődés nélkül, az előbbi szerint növekvő sorrendben. (7tobb)
8. Adjuk meg lekérdezés segítségével azt az évet, amikor a világrangsor vezetője legtöbbször változott. (8mozgalmas)
9. Lekérdezés segítségével adjuk meg azokat a férfi teniszeseket, akik egy naptári év minden napján világrangsorosa voltak. (9rekorderek)

Beküldendő egy tömörített `i588.zip` állományban az adatbázis exportját tartalmazó `tenisz.sql` és a feladatok megoldását tartalmazó `tenisz_megoldas.sql` nevű állomány.

Létölthető állomány: `vilagelso.txt` és `jatekos.txt`

I/S. 70. Egy T betűsorozatot palindromnak nevezünk, ha az angol ábécé kisbetűiből áll, és visszafele olvasva megegyezik T-vel. Például az „abffba”, „e” és „tt” betűsorozatok palindromok, de az „ab”, „xyz” és „abab” betűsorozatok nem.

Egy T betűsorozatot szuperpalindromnak nevezünk, ha pontosan egy betűből áll, vagy pedig felírható P x P alakban, ahol P egy szuperpalindrom, x pedig az angol ábécé egy tetszőleges kisbetűje. Például az „a”, „xyxhyx” és „aaa” betűsorok szuperpalindromok, de az „aa”, „xyxhyx” és „aabcbaa” betűsorok nem.

Adott egy N darab betűből álló betűsor. Adjuk meg, hogy minimum hány betűjét kell megváltoztatni ahhoz, hogy egy szuperpalindromot kapjunk.

A bemenet első sorában az N szám található, a betűsor hossza. A második sorban az angol ábécé N darab betűje található, szóköz nélkül.

A kimenet egyetlen sorában egyetlen szám szerepeljen: a minimálisan átírandó betűk száma, hogy szuperpalindromot kapjunk. Ha ez nem lehetséges, írjunk ki (-1) -et.

Bemenet (a / jel sortörést helyettesít)	Kimenet
7 / aaabbbb	3
3 / aba	0
2 / ab	-1

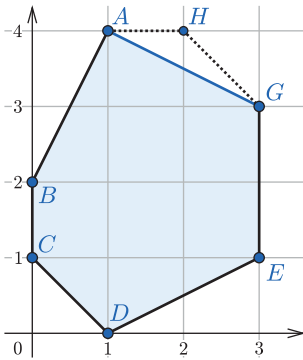
Magyarázat (1. példa): 3 betű megváltoztatásával kaphatjuk például a következő szuperpalindromot: „abababa”.

Korlátok: $1 \leq N \leq 10^5$. Időkorlát: 0,4 mp.

Értékelés: a pontok 50%-a kapható, ha a program helyes kimenetet ad az $N \leq 7$ esetekben.

Beküldendő egy `is70.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható. A dokumen-

táció tartalmazza a megoldás elméleti háttérét, az esetleg felhasznált forrásokat. Ne tartalmazzon kódrészleteket, azok magyarázata kódkommentek formájában a forrásprogramban szerepeljen.



S. 169. 2 egység magasságú egyenes hasáb alakú tárolóedényt szeretnénk készíteni. Az edény alapjának elkészítéséhez kiindulásként adott egy konvex sokszög, melynek csúcsai a koordináta-rendszer rácspontjaira esnek. Ennek a sokszögnek töröljük egy csúcsát és a csúcshoz kapcsolódó két élét, és összekötjük egy éllel a törölt élek megmaradt csúcspontjait. Az így kialakuló sokszöget tekintjük a tárolóedény alapjának.

Feladatunk, hogy a feltételek szerint adott konvex sokszög esetén határozzuk meg, mekkora a belőle készíthető legnagyobb kapacitású edény kapacitása.

A bemenet első sorában a konvex sokszög csúcsainak N száma szerepel. A következő N sorban a csúcsok x és y koordinátája pozitív körüljárási irány szerint. A listában a szomszédos pontok a sokszöget alkotó szakaszok végpontjai. Az első és az utolsó csúcs is össze van kötve.

A kimenet első és egyetlen sorába a konvex sokszögből készíthető legnagyobb kapacitású edény kapacitása kerüljön.

Minta:

Bemenet (a / jel sortörést helyettesít)	Kimenet
7 / 1 4 / 0 2 / 0 1 / 1 0 / 3 1 / 3 3 / 2 4	17

Magyarázat: az ábrán a H csúcsot törölve az edény kapacitása 17.

Korlátok: $1 \leq N \leq 100\,000$, $-10^8 \leq x, y \leq 10^8$. Időkorlát: 1 mp.

Értékelés: A pontok 40%-a kapható, ha a program helyes kimenetet ad az $N \leq 100$ esetekben.

Beküldendő egy `s169.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható. A dokumentáció tartalmazza a megoldás elméleti háttérét, az esetleg felhasznált forrásokat. Ne tartalmazzon kódrészleteket, azok magyarázata kódkommentek formájában a forrásprogramban szerepeljen.



A feladatok megoldásai regisztráció után a következő címen tölthetők fel:

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

Beküldési határidő: 2023. április 15.

