

Informatikából kitűzött feladatok



I. 457. Egy síkon K darab pálcika fekszik – a *Marokkó* nevű játékhoz hasonlóan – melyeket pozitív egész számokkal azonosítunk. A pálcikák elhelyezkedése véletlenszerű, egymást úgy keresztezhetik, hogy a nagyobb azonosítójú van mindig feljebb. A pálcikák végpontjainak koordinátái egész számok. A pálcikák egyesével gyűjthetők össze úgy, hogy egy pálcika elvételekor a többi pálcika nem mozdulhat meg: az a pálcika vehető el, amelyet felülről nem keresztez másik. Két pálcika végpontjának találkozása nem számít keresztezésnek.

Készítsünk programot `i457` néven, amely a pálcikák azonosítójának egy olyan sorrendjét adja meg, amellyel a pálcikák mindegyike elvehető úgy, hogy minden lépésben az elvehető pálcikák közül a legkisebb sorszámút választjuk.

A program standard bemenetének első sorában a pálcikák K ($2 \leq K \leq 50$) számát és ezt követő K sorban a pálcikák azonosítóját és végpontjainak (x_1, y_1) és (x_2, y_2) ($1 \leq x_1, y_1, x_2, y_2 \leq 50$) koordinátáit adjuk meg. A program írja ki a standard kimenetre a pálcikák azonosítójának szóközzel elválasztott sorrendjét, amely megadja az összes pálcika elvételének megfelelő sorrendjét.

Példa a bemenetre (a / sortörést jelöl):	Kimenet
9	4 3 1 5 6 8 2 7 9
1 17 29 18 19 / 2 26 27 19 20 / 3 22 29 15 22	
4 18 14 15 24 / 5 20 14 18 24 / 6 20 22 22 12	
7 25 19 19 11 / 8 23 14 21 24 / 9 29 28 27 38	

Beküldendő egy tömörített `i457.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 458 (É). Az éjszakai égbolt csillagai közti könnyebb eligazodás érdekében az emberek már több ezer évvel ezelőtt is az egymáshoz közel látszó, fényesebb csillagokat emberi vagy isteni lények, állatok vagy tárgyak képével azonosították. Egy-egy ilyen, égen látható csillagcsoportot az oda gondolt alakzattal együtt csillagképnek hívtak. Feladatunk a ma használatos, modern és hivatalosan elfogadott 88 csillagkép adatainak feldolgozása adatbázis-kezelő program segítségével.

Az adatok a `csillagkephely.txt` és `szomszedoscs.txt` állományokban állnak rendelkezésünkre. Az állományok tabulátorral tagolt, UTF-8 kódolású szövegfájlok, az első sorok a mezőneveket tartalmazzák.

1. Készítsünk új adatbázist `csillagkepek` néven. A mellékelt adatállományokat importáljuk az adatbázisba a fájlnevével azonos nevű táblákba. Beolvasáskor állítsuk be a megfelelő típusokat és kulcsokat. A második táblánál hozzunk létre kulcsot.

Táblák:

csillagkephely (cskhely, csknev, latinnev, nterület, szterület, tlateszaktol, tlatdelig, legfenyescs)

cskhely	az adott csillagkép azonosítója (szám), ez a kulcs;
csknev	az adott csillagkép magyar neve (szöveg);
latinnev	az adott csillagkép latin neve (szöveg);
nterület	az adott csillagkép területe négyzetfokban megadva (szám);
szterület	az adott csillagkép területe hány százaléka az égbolt területének (szám);
tlateszaktol	megadja, hogy az adott csillagkép hányadik foktól látható teljes egészében az északi féltekén (szám);
tlatdelig	megadja, hogy az adott csillagkép hányadik fokig látható teljes egészében a déli féltekén (szám);
legfenyescs	az adott csillagképben látható legfényesebb csillag fényessége magnitúdóban kifejezve (szám).

szomszedoscs (cskhely, szomszedoshely)

cskhely	az adott csillagkép azonosítója (megegyezik a csillagkephely táblában szereplő azonosítóval) (szám);
szomszedoshely	a cskhely azonosítójú csillagképpel szomszédos csillagkép azonosítója (szám).

Készítsük el a következő feladatok megoldásait. Az egyes lekérdezéseknél ügyeljünk arra, hogy mindig csak a kért értékek jelenjenek meg és más adatok ne. Megoldásainkat a zárójelben lévő néven mentjük el.

2. Bővítsük az adatbázisunkat a 88. *Dél Keresztje* csillagképpel és adataival. A hiányzó adatok megtalálhatók a feladat forrását képező weboldalon*.
3. Módosítsuk az **szterület** megjelenési formátumát úgy, hogy az a százalékjellel együtt százalék formátumban jelenjen meg.
4. Adjuk meg annak a csillagképnek a nevét, amelynek a legtöbb szomszédos csillagképe szerepel az adatbázisban. Írassuk ki azt is, hogy hány szomszédja van. Ha több azonos számú is van, jelenítsük meg mindegyiket. (4szomszed)
5. Adjuk meg, hogy mekkora területet fednek le a csillagképek összesen. Az eredményt függvény segítségével kerekítsük egészre. (5egnagysag)
6. Melyek azok a csillagképek, amelyekben van a Vízöntő legfényesebb csillagánál fényesebb csillag? Jelenjen meg a csillagkép neve és a benne található legfényesebb csillag magnitúdója. A magnitúdó kisebb értéke jelenti a nagyobb fényességet. (6fenyes)
7. Határozzuk meg, melyik három csillagkép látható a legnagyobb tartományban. Adjuk meg a csillagkép latin nevét és a teljes láthatóság szögtartományának nagyságát. (7fok)
8. Számoljuk össze, hány csillagkép nevében szerepel az „északi” előtag. (8eszaki)

*A feladat forrása: https://hu.wikipedia.org/wiki/Csillagképek_méret_szerinti_listája (utolsó letöltés: 2017. 11. 12.).

9. Vizsgáljuk meg, hogy a „Déli hal” csillagkép – nevéhez híven – valóban nagyobb szögtartományban látható-e a déli féltekén azoknál a csillagképeknél, amelyek nevében szerepel a „hal”, de nem szerepel a „déli” szórészlet. Válaszként jelelnünk meg a „Déli hal” csillagkép déli féltekén való láthatósága és az összes többi „hal” csillagkép déli féltekén lévő átlagos láthatóságának különbségét. (9tobbe)
10. Készítsünk lekérdezéssel új táblát „allatok” néven, melybe kigyűjtjük az állatöv 12 csillagképeének (Kos, Bika, Ikrek, Rák, Oroszlán, Szűz, Mérleg, Skorpió, Nyilas, Bak, Vízöntő, Halak) legfontosabb adatait: a csillagkép azonosítóját, magyar és latin nevét, a területét és a szomszédos csillagképek számát. (10allatok)

Beküldendő egy tömörített `i458.zip` állományban az adatbázis, valamint egy rövid dokumentáció, amelyből kiderül az alkalmazott adatbázis-kezelő neve és verziószáma.

I. 459. A kenguru nyelvben csak a K, E és N betűket használják. Egyetlen egybetűs értelmes szó van, az E. A két- vagy több-betűs szavak közül azok értelmesek a kenguru nyelvben, amelyek tartalmazznak E betűt, és az utolsó betűjüket elhagyva olyan szót kapunk, amely nem értelmes a kenguru nyelvben.

Írjunk programot, amely elállítja az X betűből ($1 \leq X \leq 12$) álló értelmes szavakat a kenguru nyelvben.

Beküldendő egy `i459.zip` tömörített mappában a program forráskódja és rövid dokumentációja, amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I/S. 27. Egy ország N városa között autóbusszjáratok közlekednek, melyeknek ismerjük a menetrendjét. A városokat pozitív egész számokkal jelöljük. Az 1-es városból szeretnénk eljutni az N -es városba autóbusszok segítségével. Minden járat két város között közlekedik, az egyes járatok azonos időközönként követik egymást. Tudjuk minden járatról a napi első indulási időpontot és a járat menetidejét. A járatok utolsó indulási ideje 20:00, később már nem indulnak autóbusszok. Az alábbi példa első járata az 1-es várostól a 4-es városig közlekedik, az út 80 percig tart, az első járat 7:00-kor indul (a nap 420. percében), majd minden következő 200 perccel az előző után, és így az utolsó 17:00-kor.

Átszálláskor legkorábban a megérkezés után legalább 10 perccel később induló buszokat érjük el biztonságosan. Minden városban van szálloda, így nem jelent gondot valamelyikben megszállni éjszakára. Számítsuk ki, hogy legkevesebb hány percig tart eljutni az induló városból a cél városba, illetve adjuk meg, hogy mely városokat érintünk egy ilyen utazás során. Ha több megoldás is lehetséges, akkor elég egyet megadni. A menetrend csak az eljutás szempontjából fontos járatokat tartalmazza, nem az összes járatot, de az 1-es városból biztosan el lehet jutni buszokkal az N -es városba.

A program a standard bemenet első sorából olvassa be a városok N számát, majd a következő N sorból a járatok induló és cél városát, a menetidőt, a nap első járatának indulási idejét 0:00-tól számítva, illetve az egymást követő buszok indulása közötti eltérés idejét percben. A program írja a standard kimenet első

sorába a legrövidebb eljutás idejét percben, majd a következő sorba az egy ilyen időtartamú út során érintett városokat sorrendben.

Példa:

Bemenet (a / sortörést jelöl):	Kimenet
7 / 1 4 80 420 200 / 1 3 125 380 240 / 4 2 220 340 90	400
2 5 110 360 65 / 2 3 70 320 80 / 3 6 180 510 180	1 3 5 7
3 5 60 430 95 / 5 6 40 420 60 / 6 7 100 390 120	
5 7 160 440 180	

Korlátok: $4 \leq N \leq 100$, a menetidők nem hosszabbak 10 óránál.

Értékelés: a megoldás lényegét leíró dokumentáció 1 pontot ér. További 9 pont kapható arra a programra, amely a korlátoknak megfelelő bemenetekre helyes kimenetet ad 1 másodperc futásidő alatt. Részpontszám kapható arra a programra, amely csak kisebb N értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `is27.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

S. 126. Általánosítsuk az idén februárban kitűzött **C. 1466.** feladatot. Egy bizottság összesen A alkalommal ülésezett. A tagok közül minden ülésen pontosan S személy vett részt, de bármely két tag legföljebb egyszer volt együtt jelen. Legalább hány tagból áll a bizottság?

A program a standard bemenet első sorából olvassa be az ülések A számát és az egy ülésen résztvevő személyek S számát. A program írja a standard kimenet első sorába, hogy legkevesebb hány tagból áll a bizottság.

Példa:

Bemenet	Kimenet
4 3	6

Korlátok: $3 \leq S \leq 10$, $3 \leq A \leq 12$.

Értékelés: a megoldás lényegét leíró dokumentáció 1 pontot ér. További 9 pont kapható arra a programra, amely a korlátoknak megfelelő bemenetekre helyes kimenetet ad 1 másodperc futásidő alatt. Részpontszám kapható arra a programra, amely csak kisebb bemeneti értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `s126.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.



A feladatok megoldásai regisztráció után a következő címen tölthetők fel:

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

Beküldési határidő: 2018. június 10.

