

Informatikából kitűzött feladatok



I. 439. Egy hosszú egyenes folyón utazunk folyásirányban lefelé egy kevés benzinnel rendelkező motorcsónakon. A folyószakasz hosszú, ezért csak időnként kapcsoljuk be a motort, különben csak az áramlás sebességével utazunk (sodródunk) lefelé. Amikor a motor működik, akkor a parthoz viszonyított sebesség az áramlás sebességének és a motorcsónak vízhez viszonyított sebességének összege lesz, mivel a csónak mindig az áramlás irányában áll.

Készítsünk programot, amely a különböző folyószakaszok v_i áramlási sebességének, a motorcsónak vízhez viszonyított v_m sebességének és maximális működési idejének ismeretében megadja a kiindulási helytől L távolságra lévő célba érkezés minimális idejét.

A standard bemenet első sora négy számot tartalmaz: a megteendő folyószakasz L ($1 \leq L \leq 100$) hosszát (km), a motorcsónak $1 \leq v_m \leq 10$ sebességét (km/h) a vízhez képest, a motor T ($1 \leq T \leq 10$) maximális üzemeltetési idejét (h) és a különböző sodrási sebességű folyószakaszok N ($1 \leq N \leq 50$) számát. Az ezt követő N sor soronként két számot tartalmaz: az adott sodrású folyószakasz elejének a kiindulási helytől mért távolságát ($0 \leq E_i \leq L$) (km) és a folyószakasz sodrási sebességét v_i ($1 \leq v_i \leq 10$) (km/h).

A standard kimenetre írjuk ki a célba érkezés minimális idejét órában, három tizedesjegy pontossággal.

Példa bemenet (a / jel új sort helyettesít)	Kimenet
20 2 3 5 / 0 2 / 3 1 / 7 2 / 12 1 / 15 2	8.167

Beküldendő egy tömörített `i439.zip` állományban a program forráskódja, valamint a program rövid dokumentációja, amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 440 (É). A feladat a `gsmotthon.hu` (utolsó letöltés: 2017.10.22.) weboldalról származó mobiltelefonok adatait összefoglaló táblázat feldolgozása lesz táblázatkezelő program segítségével. A forrásállomány egy-egy sorában rendelkezésre állnak különböző telefonok legfontosabb adatai.

1. Töltsük be a honlapunkról letölthető `mobiltelefonok.txt` szövegfájlt, mely UTF-8 kódolású és tabulátorokkal tagolt, a táblázatkezelő egy munkalapjára az A1-es cellától kezdődően. A megjelenítés év.hónap formátumban történjen (pl. 2016.08.). Munkánkat `i440` néven mentjük a táblázatkezelő alapértelmezett formátumában.
2. Azokban a cellákban, ahol a nyers táblázat nem tartalmazott adatot, és még nincsen benne, jelenjen meg az „na”, nincs adat felirat. Kivétel a kódnév oszlop,

ahol ezt nem kell megtenni, hiszen megszokott dolog, hogy egy telefonnak nem adnak másodlagos nevet.

3. A rendelkezésre álló szöveges adatok között előfordulnak elírások, például fekete helyett fekete, Android helyett Andorid, ezeket javítsuk ki.
4. Oszloponként jelöljük a megfelelő mértékegységeket. A RAM és a háttértár mérete GB-ban, az ár forintban (Ft), a kamerák képfelbontása megapixelben (MP), a kijelző inchben ("), a súly grammban (g), a méretek milliméterben (mm) vannak megadva. A bővíthetőségnél elég azt megjeleníteni, hogy hány GB-ig bővíthető (pl. igen, 32 helyett, 32 GB-ig), és az akkumulátor kapacitása milliampérórában van megadva (mAh).
5. A telefonok árát a készpénz kerekítés szabályai szerint kerekítsük 0-ra, vagy 5-re egy új oszlopba a jelenlegi ár mellé. (A kerekítés szabályai szerint az 1-re, 2-re végződőket 0-ra, a 3-ra, 4-re, 6-ra és 7-re végződőket 5-re, míg a 8-ra és 9-re végződő árakat 10-re kell kerekíteni.)

6. Android telefonok esetében az operációs rendszer típusánál a verziószám helyett jelenjen meg az elnevezése. Az alábbi táblázatot illesszük a megoldásba, és használjunk megfelelő függvényeket a megoldáshoz.
7. Feltételes formázást használva változtassuk meg a betűszínét zöldre azon soroknak, ahol minden adat meg van adva. (Ha csak a kódnév nincsen megadva, attól még jelöljük ki a sort.)
8. Rendezzük a táblázatot a márka, azon belül pedig a típus neve szerinti ABC sorrendbe.

Verzió	Elnevezés
2.2	Froyo
2.3.3-2.3.7	Gingerbread
4.0.3-4.0.4	Ice Cream Sandwich
4.1	Jelly Bean
4.4	KitKat
5.0.0-5.0.2 5.1-5.1.1	Lollipop
6.0	Marshmallow
7.0 7.1	Nougat
8.0	Oreo

9. A következő feladatot egy új munkalapon oldjuk meg. Gyűjtsük ki kategóriák szerint az előző munkalap adatait felhasználva, hogy összesen hány készülék tartozik az egyes típusokhoz. Ily módon készítsünk – a színek és operációs rendszer mellett – még három általunk választott kategóriát. Minden kategóriának adjunk oszlopcímet és formázzunk a mintán látható módon. Színeknél a kötőjellel jelölt színek azt jelzik, hogy a telefon elő- és hátlapja eltérő színű, ezek a kötőjeles színek egy külön kategóriaként jelenjenek meg. A dupla hátlapi kamerával rendelkező mobiloknál a készülék jelenjen meg mindkét kategóriánál, ha azt választjuk.

	A	B
1	dual vagy sim	
2	dual	db
3	sim	db
4	na	db
5		
6	RAM	
7	1	dh

10. Készítsünk diagramot új munkalapra, mely szemlélteti, hogy melyik évben hány telefont adtak ki a rendelkezésre álló adatok alapján.

Beküldendő egy tömörített `i440.zip` állományban a megoldást adó táblázatkezelő munkafüzet és egy rövid dokumentáció, amely megadja a felhasznált táblázatkezelő nevét és verzióját.

I. 441. Az Első Rend kutatói egy veszélyes sugárfejevert fejlesztettek ki, amely minden eddig ismert védópajzson áthatol. Az Ellenállás egy olyan szilárd anyagból álló pajzson kísérletezik, amely a sugárzást visszaveri vagy elnyeli. Tudósai megállapították, hogy a sugárzás csak két anyag részecskéivel lép reakcióba. Amikor a sugárzás nagy energiájú, akkor a T anyag részecskéi, amikor már kisebb energiájú, akkor az N anyag részecskéi tudják befogni. Mindkét esetben a befogás után nem sokkal a részecskék is kibocsátanak sugárzást: az N anyag sugárzása veszélytelen, míg a T anyag kibocsát egy, a fejevertől származó sugárzással egyező tulajdonságú, de kisebb energiájú sugárzást. Úgy is tekinthetjük, hogy a T anyag részecskéi visszaverik a nagy energiájú sugarakat, míg a kis energiájúak áthatolnak rajta. Az N anyag részecskéi elnyelik a kis energiájú sugarakat, de a nagy energiájúak áthatolnak rajta. Megmérték, hogy a fejever sugárzása a T részecskékkel történő ötszöri reakció után válik kis energiájúvá.

A T és az N anyag nagyon ritka a Galaxisban, az Ellenállás csak igen keveset tudott beszerezni belőlük. A tudósok terve az, hogy az N és a T anyag részecskéit elkeverik egy olyan anyagba, amelyen áthatol a sugárzás. Az így kialakított pajzs – megfelelő vastagság, illetve megfelelő számú N és T részecske esetén – alkalmas lenne arra, hogy a sugárzás döntő részét elnyelje, vagy szétszórja, visszaverje. Mivel sem idő, sem megfelelő mennyiségű anyag nem áll rendelkezésre, ezért számítógépes szimulációval vizsgálják, hogy adott N és T részecskemennyiség, valamint falvastagság mellett a bejövő sugárzás mekkora része hatolna át a falon.

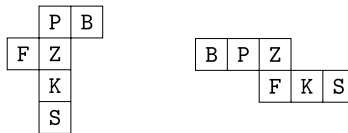
A tudósok szerint a jelenséget jól leírja egy síkbeli modell: a fal egy $a \times b$ oldalú téglalap, a sugárzás a téglalap egyik b hosszúságú oldalán érkezik, és a téglalap belseje felé tart. A téglalapot gondolatban $a \times b$ darab egységnyi négyzetre bontjuk. Mindegyik ilyen négyzet vagy üres (itt áthatol a fejever sugárzása), vagy T típusú (egy T részecskét tartalmaz, amely a nagy energiájú sugárzást elnyeli), vagy N típusú (egy N részecskét tartalmaz, a kis energiájú sugárzást nyeli el). A T típusú szórás azt jelenti, hogy csökkent energiával, ugyanakkor véletlenszerű irányba történik az elnyelést követő kisugárzás. A sugárzás egyenes irányba terjed, minden négyzetet, amelyet érint, vizsgálni kell az előbbieik alapján. Ha a sugárzás a téglalap a hosszú oldalain kilép, akkor az ellenkező oldalon bejövő sugárzásként folytatja útját. Ha a sugárzás elnyelődik, vagy azon a b hosszú oldalon lép ki, amelyen beérkezett, akkor a fal hatékonyan működik. Ha a bejövő sugárzás a b hosszúságú másik oldalon lép ki, akkor átjutott a falon.

Készítsük el a szimulációt végző programot. A program standard bemenete a falat modellező téglalap a szélessége ($10 \leq a \leq 100$), és b magassága ($100 \leq b \leq 10000$), valamint az N és T típusú részecskét tartalmazó négyzetek száma a téglalapon ($a \cdot b / 10 \leq N + T \leq a \cdot b$). A program adja meg a standard kimeneten, hogy 100 000 beérkező sugárzásból átlagosan hány sugár jut át a falon.

Példa bemenetek	Példa kimenetek
10 150 100 100	63495
10 150 100 100	63414
20 150 200 400	23339
20 150 200 400	22930
50 5000 3000 10000	36083
50 5000 3000 10000	35997

Beküldendő egy tömörített `i441.zip` állományban a program forráskódja, valamint a program rövid dokumentációja, amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I/S. 21. Adott két kiszínezett kocka. Mindkét kocka külső oldalai vannak színezve, egy kockán belül nincs két azonos színű oldal. Elkészítjük a két kocka síkbeli hálóját – bizonyos élek mentén történő felvágással – úgy, hogy egy összefüggő síkidomot kapjunk. Két lehetséges hálót mutat az alábbi ábra:



Írjunk programot, amely eldönti, hogy a két hálóból összeállítható-e hajtogatóval két egyező színezésű kocka.

A program a hálók leírását a standard bemenetről olvassa. A bemenet 10, 5 karakter hosszú sorból áll, az első 5 sor az első kocka, a következő 5 sor a második kocka hálóját adja meg. A lapok helyén az angol ábécé egy-egy nagybetűje szerepel, amely a színt jelöli, a többi helyre pedig a `*` (csillag) karakter kerül.

Ha a két háló azonos színezésű kockát határoz meg, akkor a standard kimenet egyetlen sorába az „igen” szó kerüljön, egyéb esetben pedig annyi sorból álljon, ahány lap színének módosítása legalább szükséges a második kockán az elsővel egyező színezés kialakításához. Minden sorba két karakter kerüljön, az első azt a színt adja meg, amit cserélni kell, a második pedig azt, amire változtatni kell.

Példa bemenet	Példa kimenet	Példa bemenet	Példa kimenet
*PB**	B S	*PB**	Igen
FZ***	S B	FZ***	
*K***		*K***	
*S***		*S***	
*****		*****	
*****		*****	
BPZ**		SPZ**	
**FKS		**FKB	
*****		*****	
*****		*****	

Értékelés: a megoldás lényegét leíró dokumentáció 1 pontot ér. További 9 pont kapható arra a programra, amely a megfelelő bemenetekre helyes kimenetet ad 1 másodperc futásidő alatt.

Beküldendő egy `is21.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

S. 120. Egy különös autóbusz külön utasok elszállításán dolgozik. A megállóban utasok várakoznak, mindegyikben legalább egy. Néhány megálló között közvetlen buszközlekedés van, tehát a busz más megállók érintése nélkül halad közöttük. Bármely két megálló között pontosan egy útvonal van. Az utasok azért különösek, mert az odaérkező buszra a várakozók közül mindig pontosan egy száll föl. A busz is különös, mert útja során nem halad át olyan megállón, amelyben már nincs várakozó. Nincs menetrend, a buszvezető feladata, hogy a megállót olyan sorrendben érintse, hogy a lehető legtöbb utast tudja fölvenni.

A megállót 1-től kiindulva pozitív egészekkel azonosítjuk, az utolsó megálló sorszáma M . Az autóbusz kezdetben az 1-es megállóban tartózkodik, és induláskor fölvesz egy embert. Minden megállóról tudjuk, hogy milyen más megállókkal van közvetlen buszkapcsolatban. Kezdetben minden megállóban legalább egy, legföljebb U utas várakozik, akik türelmesen várják a buszt, nem hagyják el a megállót. Az autóbusz az utasok összegyűjtése után az 1-es megállóba tér vissza.

A program standard bemenete M és U , majd a következő M sor mindegyikében az adott megállóban várakozó utasok száma, utána az adott sorszámú megállóból közvetlen buszjáráttal elérhető megállók sorszáma szerepel. A program standard kimenete legyen a legtöbb fölvehető utas száma.

Példa bemenet (a sortöréseket / jellel helyettesítettük)	Példa kimenet
12 10 / 3 2 3 6 7 / 6 1 / 2 1 4 / 5 3 / 4 6 / 4 5 1 9 10 11 4 1 8 / 6 7 12 / 6 6 / 5 6 / 2 6 / 3 8	26

Korlátok: $2 \leq M \leq 100\,000$, $1 \leq U \leq 30$.

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. Részpontoszám kapható arra a programra, amely csak kisebb M és U érték esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `s120.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: 2017. december 10.

