

Проблем 1. Најслабија карика

У популарном телевизијском квизу "Најслабија карика" неколико такмичара редом у круг одговарају на постављена питања. Први тачан одговор доноси један динар, док сваки следећи тачан одговор у низу доноси дупло више поена него претходни. Када неко одговори нетачно - ланац се прекида и следећи тачан одговор доноси 1 динар. Када такмичар каже "Банка", сав новац у тренутном ланцу тачних одговора се додаје на укупну загарантовану суму, а наредни тачан одговор поново вреди 1 динар. Ако такмичари у низу дају 15 тачних одговора, а између није било банкирања - награда износи невероватних 1000000000 (милијарду) динара, без обзира да ли је неко рекао "Банка" или су пре тога такмичари освојили нешто новца. Вама су на располагању подаци из претходне емисије. Треба одредити колико динара су такмичари зарадили.

Улаз. (Улазни подаци се налазе у датотеци `karika.in`) У првом реду улазне датотеке се налази низ знакова. Сваки од знакова је из скупа $\{'0', 'X', 'B'\}$, (сва три могућа знака су велика слова, што значи да први од три знака није цифра нула, већ слово O) и они представљају тачан одговор, нетачан одговор и банку. Дужина низа знакова је мања или једнака од 100000.

Излаз. (Излазне податке уписати у датотеку `karika.out`) У излазну датотеку треба уписати број динара који су такмичари зарадили.

Пример 1.

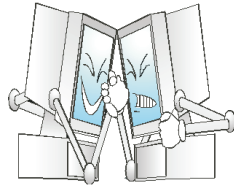
<code>karika.in</code>	<code>karika.out</code>
<code>000VX0XVX00V0X</code>	<code>10</code>

Објашњење. После банкеирања, такмичари редом добијају $7 + 0 + 3 = 10$ динара.

Пример 2.

<code>karika.in</code>	<code>karika.out</code>
<code>X00V0000000000000000X00V</code>	<code>1000000000</code>

Објашњење. Такмичари имају низ од 15 тачних одговора, па зарађују 1000000000 динара.



Проблем 2. Игра

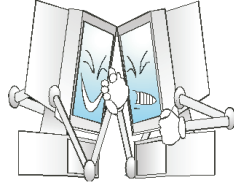
На папиру је написан број N ($1 \leq N \leq 10^7$). Два играча наизменично раде следеће: играч који је на потезу бира неки делилац броја који је написан на папиру (делилац не сме бити 1 или број са папира), рачуна количник броја са папира и изабраног делиоца, брише број који се налази на папиру, и пише добијени количник. Уколико играч не може да одигра свој потез (тј. једини делиоци су 1 и број са папира), он је победник. Написати програм који треба да учита низ бројева и за сваки од тих бројева одреди који играч побеђује у партији на чијем почетку се на папиру налази тај број. Сматрати да оба играча играју оптимално (повлаче најбоље потезе).

Улаз. (Улазни подаци се налазе у датотеци `igra.in`) У првом реду улазне датотеке се налази број M ($5 \leq M \leq 20$). У наредних M редова се налази по један број N_i ($1 \leq N_i \leq 10^7$).

Излаз. (Излазне податке уписати у датотеку `igra.out`) У излазну датотеку треба уписати M редова. У i -том реду исписати број 1, ако први играч побеђује кад је почетни број на папиру N_i , а 2, ако други играч побеђује.

Пример 1.

<code>igra.in</code>	<code>igra.out</code>
5	1
1	2
4	1
5	2
10	2
25	



Проблем 3. Саксије

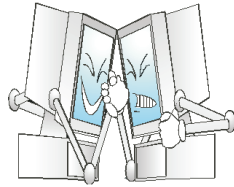
Баштован Тома је решио да улепша своје пространо двориште тако што ће га украсити цвећем. Он наине жели да најпре постави n саксија у ред ($1 \leq n \leq 1000000$) и да у свакој од ових n саксија буде k килограма земље ($0 \leq k \leq 1000$), а да потом у њима сади различито цвеће. Он је зато од фирме која се бави продајом саксија наручио n саксија таквих да се у свакој од тих n саксија налази k килограма земље. Та фирма је била толико љубазна да му не само донесе те саксије већ и да их постави у ред. Међутим, сутрадан је Тома доживео велики шок. Неке саксије имају више, а неке мање од k килограма земље. Тома се ипак мало смирио када је приметио да је укупна тежина земље у свим саксијама $k \cdot n$ килограма - па се ипак ова грешка да исправити. Тома жели да пребацивањем земље из једне у другу саксију учини да у свакој саксији буде k килограма земље, а да се при томе најмање умори. Тома зна да за пребацивање x килограма земље из саксије која је i -та по реду у саксију која је j -та по реду утроши $x \cdot |i - j|$ џула енергије. Одредити колико је минимално енергије коју Тома мора потрошити да би исправио грешку фирме која му је продала саксије.

Улаз. (Улазни подаци се налазе у датотеци `saksije.in`) У првом реду текстуалне датотеке налазе се редом цели бројеви n и k . У другом реду ове датотеке налази се n целих бројева који су већи или једнаки 0. Наине i -ти број ($1 \leq i \leq n$) у другом реду означава колико се килограма земље налази у i -тој саксије када се гледа са прозора Томине спаваће собе са лева на десно.

Излаз. (Излазне податке уписати у датотеку `saksije.out`) У излазну датотеку треба уписати један број који је једнак $W \bmod 1000000000$, где је W једнак минималном броју џулова које Тома мора потрошити да би исправио грешку фирме од које је купио саксије.

Пример 1.

<code>saksije.in</code>	<code>saksije.out</code>
6 4	8
5 6 2 1 7 3	



Проблем 4. Камен

Мали Ђурица је нашао N каменчића. На сваком каменчићу је написано једно велико слово енглеске абецеде (слова од 'A' до 'Z'). Он је измерио сваки каменчић и утврдио његову масу у грамима. Након тога је написао реч дужине M која се састоји само од великих слова енглеске абецеде. Малог Ђурицу занима која је најмања маса каменчића, које треба да изабере, тако да кад их некако сложи у један ред добије претходно написану реч.

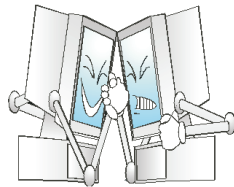
Пошто мали Ђурица жели да се игра, овај проблем је препустио Вама.

Улаз. (Улазни подаци се налазе у датотеци `kamen.in`) У првом реду улазне датотеке се налазе цели бројеви N ($1 \leq N \leq 50000$) и M ($1 \leq M \leq 50000$). У наредних N редова се налазе по једно слово s_i и један цео број t_i , где s_i представља слово на i -том каменчићу (s_i ће увек бити велико слово енглеске абецеде), а t_i ($1 \leq t_i \leq 50000$) масу i -тог каменчића. Потом се учитава M редова који садрже по једно слово и то су редом слова речи коју је написао мали Ђурица.

Изаз. (Излазне податке уписати у датотеку `kamen.out`) У првом реду датотеке исписати најмању укупну масу каменчића које је потребно користити да би се њиховим слагањем добила написана реч. Ако је немогуће добити написану реч исписати -1 .

Пример 1.

<code>kamen.in</code>	<code>kamen.out</code>
5 3	100
A 10	
B 40	
C 30	
B 20	
E 50	
C	
E	
B	



Пример 2.

камен.in

14 11

A 1

I 2

K 3

L 4

O 5

P 6

N 7

F 8

R 9

O 10

M 11

T 12

I 13

Z 14

I

N

F

O

R

M

A

T

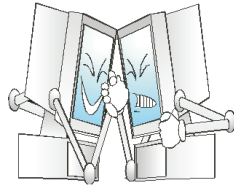
I

K

A

камен.out

-1



Проблем 5. Кртице

Две групе кртица су се нашле на суседним страницама правоугаоне ливаде оивичене оградом. Свака кртица је пронашла по једну рупу на огради. У тренутку кад је почела да пада киша кртице су почеле да копају тунеле у правцу нормалном на ограду. Кртица копа тунел док не дође на већ ископан тунел (сви тунели су на истој дубини), а онда престаје и почиње да се креће већ ископаним тунелом. Ако је дата брзина којом свака кртица копа тунел, израчунати колико дуго ће свака кртица копати тунел пре него што наиђе на већ ископан тунел. Ако нека кртица никада неће наићи на ископан тунел, исписати -1 .

Улаз. (Улазни подаци се налазе у датотеци `krlice.in`) У првом реду датотеке налазе се два цела броја: n_1 - број кртица у првој групи (оне које се налазе на једној страни ливаде) и n_2 - број кртица у другој групи (кртице које се налазе на другој страни ливаде) ($0 < n_1, n_2 \leq 5000$). У следећих n_1 редова се налазе подаци о кртицама из прве групе (у једном реду су подаци о једној кртици). У сваком реду се налазе по два реална броја који означавају удаљеност кртице од угла ливаде (на почетку, пре него што започне копање канала) и брзину којом та кртица копа тунел. У следећих n_2 редова се налазе подаци о кртицама из друге групе. Удаљености и брзине су између 0 и 1000000.

Изаз. (Изразне податке уписати у датотеку `krlice.out`) У сваком од $n_1 + n_2$ редова треба исписати по један број који означава време колико одговарајућа кртица (из улазне датотеке) сама копа тунел. Времена одговарају кртицама по редоследу којим су наведене у улазу, заокружено на две децимале. Дозвољена грешка 0.01.

Пример 1.

<code>krlice.in</code>	<code>krlice.out</code>
2 3	4.50
1 4	0.88
2 5.1	-1.00
4.5 3.6	1.00
1 1	-1.00
5 0.5	

Напомена: није могуће да две кртице стигну у истом тренутку копајући тунеле.