

Проблем 2. Огласна табла

Тојин Вомић је веома савестан студент. Како би што боље испратио своје (и туђе) напредовање током студија, много времена проводи пред огласном таблом анализирајући тамо приказане бодове. Нажалост, Тојин је овог семестра мало попустио с учењем, па му резултати на огласној табли баш и не иду у корист, те је размишљао како не би било лоше да се неки од њих сакрију од очију јавности. И баш у том моменту, др Груповић је, носећи цедуљу на којој су резултати с његовог последњег колоквијума, замолио Тојина да дотичну окачи на огласну таблу. (Све цедуље на огласној табли, као и цедуља професора Груповића, округлог су облика.)

Тојин је брзо размишљао: окачиће цедуљу тако да покрије што више већ објављених резултата - а што, будући да је излазност на колоквијуме проф. Груповића увек била веома висока, неће бити тешко. Но, како је Тојин ипак студент математике а не информатике, замолио је вас да му помогнете.

Улаз. (Улазни подаци се налазе у датотеци `tabla.in`.) У првом реду улазне датотеке налази се природан број n ($1 \leq n \leq 100$), који представља број резултата објављених на огласној табли. У сваком од наредних n редова налазе се по три реална броја, при чему се у i -том од тих редова налазе координате центра i -те цедуље, као и њен полупречник. Најзад, у последњем реду налази се још један реалан број, који представља полупречник Тојинове/Груповићеве цедуље. Координате су реални бројеви из интервала $[-1000, 1000]$. Полупречници су реални бројеви из интервала $[1, 1000]$. Сви реални бројеви су дати на највише 2 децимале.

Издаз. (Издазне податке уписати у датотеку `tabla.out`.) У први и једини ред издазне датотеке исписати један цео број: највећи могући број објављених резултата које Тојин може да покрије загадањем цедуље. Цедуља се сматра покривеном само ако је у потпуности скривена од очију јавности, тј. уколико ниједан делић не извирује.

Пример 1.

<code>tabla.in</code>	<code>tabla.out</code>
3	2
0 1 1	
1 0 1.41	
0 -1 1	
1.92	

Проблем 6. Низ

Дат је низ a целих бројева дужине n . Под трансформацијом низа подразумевамо повећавање или смањивање једног елемената низа за 1. Одредити минимални број трансформација низа потребних да се он преведе у строго растући низ b . За сваки елемент низа приказати разлику: $b[i] - a[i]$.

Улаз. (Улазни подаци се налазе у датотеци `niz.in`.) У првом реду се налази природни број n ($1 \leq n \leq 5.000$) који означава број елемената низа. У наредном реду се налази n целих бројева који представљају елементе низа. Апсолутна вредност елемената није већа од 1.000.000.000.

Излаз. (Излазне податке уписати у датотеку `niz.out`.) У првом реду треба исписати минимални број трансформација потребних за добијање строго растућег низа. У наредном реду штампати n бројева одвојених једним размаком који представљају промене бројева почетног низа. Уколико промене нису јединствене штампати било коју.

Пример 1.

<code>niz.in</code>	<code>niz.out</code>
4	4
1 1 1 1	-1 0 1 2

Пример 2.

<code>niz.in</code>	<code>niz.out</code>
5	5
1 1 1 6 4	-1 0 1 0 3

Објашњење. У примеру 1 описаним трансформацијама би добили растући низ $b = 0, 1, 2, 3$. У другом примеру решење није јединствено, нпр. промене су могле бити и: $-1, 0, 1, -1, 2$.

Напомена. У 40% тест примера апсолутна вредност елемената низа неће бити већа од 2.000.

Проблем 3. Балони

Мали Ђура је добио следећи задатак. Њему се доносе празни балони и балони са водом. Кад му се донесе празан балон, поставља се питање из колико највише донетих балона са водом до сада може да истресе сву воду у управо донети празан балон. Да би описали задатак који је постављен малом Ђури, дефинисаћемо две операције

- **N** v - донет је нов балон у коме се налази v литара воде
- **Q** v - донет је нов балон запремине v литара, али празан. У том моменту, мали Ђура треба да одговори из колико **највише** до сада донетих балона може да се преспе вода у тренутно донешени

Напомена. Када се донесе празан балон, балони из којих може да се преспе вода се не празне, само се даје одговор колико највише може да се испразни.

Улаз. (Улазни подаци се налазе у датотеци `baloni.in`.) У првом реду се налази природан број M ($1 \leq M \leq 100.000$) који представља број операција. У сваком од наредних M редова се налазе операције облика $C v$, где $C \in \{\mathbf{N}, \mathbf{Q}\}$. Ако $C = \mathbf{N}$, тада $1 \leq v \leq 200.000$. Ако $C = \mathbf{Q}$, тада $1 \leq v \leq 100.000.000$.

Излаз. (Излазне податке уписати у датотеку `baloni.out`.) За сваку операцију облика **Q** v , редом, у један ред исписати колико највише до сада донешених балона са водом може да се испразни у дати балон.

Пример 1.

<code>baloni.in</code>	<code>baloni.out</code>
9	0
Q 10	1
N 3	2
Q 10	2
N 7	3
Q 10	
N 2	
Q 10	
N 3	
Q 10	

Објашњење. Након доношења првог празног балона, не постоји ни један донет балон са водом, те није могуће ни један пресути у исти. Након доношења последњег празног балона у њега је могуће пресути воду из балона запремина 3, 2 и 3, што је и највише у овом случају.

Пример 2.

baloni.in
10
N 5
N 8
N 1
Q 20
N 11
Q 8
N 1
N 2
Q 31
N 12

baloni.out
3
2
6

Проблем 5. Чокоадолат

Ноћица много воли чоколаду. На срећу, одмах на улазу фирме у којој ради, „Мегахард“, налази се аутомат за чоколаду, чокоадолат. Њена фирма се бави производњом оперативног система „Дорс“ и, логично, софтвер на чокоадолату заснован је управо на том оперативном систему. Нажалост, многи тврде да је овај оперативни систем далеко инфериоран у односу на конкурентски, чија је маскота поларни медвед (а који много воли Ноћичин пријатељ Тугомир, што је и предмет њихових честих препирки, али то није битно за ову причу). Управо да пропагатори конкуренције не би добили за право, оно што се јутрос десило с чокоадолатом у „Мегахарду“ мора остати тајна — а пошто знамо да сте ви савесна деца, која неће около ширити трачеве (а и будући да Ноћица тражи помоћ од вас, па и нема баш неки избор), испричаћемо вам.

Софтвер у чокоадолату начисто је побрљавио. Уместо да износе убачених апоена сабере (како би било природно), он на све њих примени операцију XOR. Запослени у фирми то су брзо схватили, али проћи ће доста времена док не стигну резервни делови, што је Ноћицу бацило у очај.

Срећом, управа фирме донела је одлуку да се чокоадолат може користити и док овако погрешно обрачунава убачен износ. То је мало олакшало муке Ноћици, али и даље је у недоумици: уколико дође пред чокоадолат с одређеном количином апоена, шта од тога треба да убади како би добила што више чоколаде, тј. како би чокоадолат зарачунао што је већи могући износ? Постоји још једна зачкољица: паузе за узимање чоколаде у „Мегахарду“ не трају баш дуго, па Ноћица нема времена да полагаано пребира по новчанику. Једино може стићи да одабере две новчанице, и све што је у новчанику између њих (укључујући њих) стави у чокоадолат. Јасно, Ноћица је врхунска програмерка, те јој у нормалним околностима ово не би представљало никакав проблем, али како не може да функционише без чоколаде, преклиње вас да јој помогнете.

Улаз. (Улазни подаци се налазе у датотеци `sokolada.in.`) У првом реду улазне датотеке налази се природан број n ($1 \leq n \leq 250.000$), који представља број апоена које Ноћица има код себе (неки од њих могу бити и једнаки). У наредних n редова налази се по један цео број из интервала $[0, 60.000]$, при чему сваки од њих представља вредност по једног Ноћичиног апоена.

Излаз. (Излазне податке уписати у датотеку `sokolada.out.`) У први ред и једини ред излазне датотеке треба уписати један цео број, који представља максималну могућу своту коју Ноћици може зарачунати чокоадолат.

Пример 1.

<code>sokolada.in</code>	<code>sokolada.out</code>
5	30
13	
3	
11	
18	
12	

Проблем 1. Игрица

У градићу у ком живе Кићи и Ћики постоји једна наградна игрица. У игрици има m учесника. Сваки учесник изабере неко од првих n слова енглеске абецеде. Онда се од тих слова направи једна реч - слово које је изабрао први учесник је прво слово те речи, итд. На крају се преброји колико пута се свако од слова појављује у тој речи. Сви они који су изабрали слово које се појављује непаран број пута добијају бродих.

Учесници наравно не знају која ће слова остали бирати, али пошто су Кићи и Ћики јако симпатични они су успели да од сваког учесника сазнају по неколико слова која он сигурно неће изабрати.

И сада они имају задатак за вас: за свако од првих n слова они ће вам рећи да ли желе да се оно појави паран број пута, непаран број пута или им је свеједно. Ви треба да им кажете колико од речи које се могу добити у овој игрици задовољавају те њихове услове (претпостављамо да нико није слагао Кићија и Ћикија). Пошто број речи може бити велик, довољно је да им кажете остатак при дељењу тог броја са 10.007.

Улаз. (Улазни подаци се налазе у датотеци `igrica.in.`) У првом реду улазне датотеке налазе се два броја, n и m (број слова из алфабета која могу бити коришћена и број учесника у игри). Следи m редова са по n бројева у сваком од њих - у i -том од тих редова j -ти број је 0 ако i -ти учесник сигурно неће изабрати j -то слово, односно 1 ако можда хоће. У последњем реду датотеке налази се још n бројева - j -ти од њих је 0 ако Кићи и Ћики желе да се j -то слово појави паран број пута, 1 ако желе да се појави непаран број пута, односно -1 ако им је свеједно.

Излаз. (Излазне податке уписати у датотеку `igrica.out.`) У излазној датотеци треба да се налази један број - број речи које задовољавају Кићијеве и Ћикијеве услове по модулу 10.007.

Ограничења. Нека је k_i број јединица у реду датотеке који одговара i -том учеснику.

- у 20% тест примера: $n \leq 10, m \leq 6, 1 \leq k_i \leq n$
- у 20% тест примера: $n \leq 15, m \leq 12, 1 \leq k_i \leq 3$
- у осталим примерима: $n \leq 15, m \leq 50, 1 \leq k_i \leq n$

Пример 1.

<code>igrica.in</code>	<code>igrica.out</code>
5 4	3
1 0 0 0 1	
1 1 1 0 0	
1 0 1 0 0	
0 0 1 0 0	
1 -1 0 -1 -1	

Објашњење. Први учесник ће изабрати неко од слова $\{A, E\}$, други $\{A, B, C\}$, трећи $\{A, C\}$ и четврти $\{C\}$. Тражи нам се број речи у којима се A појављује непаран а C паран број пута. Могуће су следеће речи:

AAAC, AACC, ABAC, ABCC, ACAC, ACCC, EAAC, EACC, EBAC, EBCC, ECAC, ECCC.

Међу њима речи које задовољавају услов су *ABCC, EACC, ECAC* - има их 3.

Проблем 4. Торта

Млађани Зи ће ускоро да дипломира. Његови другари су одлучили да му за поклон направе бесконачну велику тарту. Након што су тарту испекли, нафиловали и украсили, другари су одлучили да тарту исеку у аутоматском секачу. Аутоматски секач је направа коју су Зијеви другари, електроничари направили током студија. Секач ради на следећем принципу: торта се убаци у секач, затим се унесе жељени број резова n . Секач ће затим исећи тарту тачно n пута. Уколико посматрамо тарту као бесконачну праву, сваки рез представља праву у тој равни.

Аутоматски секач ће одабрати положај n резова насумично, и након завршеног сечења ће одштампати извештај о положају сваког од n резова. Секач је испрограмиран тако да се никад неће десити да три различита реза пролазе кроз исту тачку на тарту. У извештају секача сваки рез је описан двома тачкама у равни торте кроз које пролази дати рез.

Након што су исекли тарту, Зијеви другари су гледали секачев извештај не би ли утврдили на колико парчића је торта исечена. Обично је млађани Зи тај који би помогао другарима у решавању сличних проблема. Пошто је торта изненађење, другари су одлучили да вас приупитају за помоћ.

Помозите Зијевим другарима да, користећи извештај секача одреде на колико парчића је торта исечена.

Улаз. (Улазни подаци се налазе у датотеци `torta.in`.) У првом реду улазне датотеке налази се природан број n ($1 \leq n \leq 10^6$), који представља број резова које је секач направио. У сваком од наредних n редова налазе се четири цела броја x_1, y_1, x_2, y_2 раздвојених размаком, који представљају координате двеју тачака кроз које дати рез пролази. Уколико тарту замислимо као бесконачну равну, дати рез представља праву која пролази кроз тачке (x_1, y_1) и (x_2, y_2) . Свака од координата је у опсегу од -10^9 до 10^9 . Никоја три реза неће пролазити кроз исту тачку на тарту.

Издаз. (Издазне податке уписати у датотеку `torta.out`.) У први ред издазне датотеке треба уписати природан број k , који представља број парчића на које је торта исечена по модулу 1.000.003.

Пример 1.

<code>torta.in</code>	<code>torta.out</code>
4	5
0 0 0 1	
1 0 1 1	
2 0 2 1	
3 0 3 1	

Пример 2.

<code>torta.in</code>	<code>torta.out</code>
3	7
0 0 1 2	
1 2 2 0	
0 0 2 0	

Пример 3.

<code>torta.in</code>	<code>torta.out</code>
3	6
0 0 0 1	
1 0 1 1	
0 0 1 0	