

Проблем 1. Голф

Ђурајгер Паунудс је професионални играч голфа. Баш се данас вратио с турнира на ком је извојевао веома тешку, али утолико слађу победу, и сада због посттурнирског стреса има проблема с ружним сновима. Ђурајгер сања да се налази на огромном терену за голф, на ком га, са жељом да га што даље лансира, јури огромна палица за голф! Срећом, на овом огромном терену за голф има много огромних рупа (као што је очекивано), и Ђурајгер је своју шансу видео у томе да се сакрије у једну од њих. Он треба да одабере једну рупу и потрчи к њој праволинијски. Палица одмах закључује ка којој рупи Ђурајгер трчи, и потрчаће ка истој рупи двапут већом брзином. Треба пронаћи рупу ка којој Ђурајгер треба да потрчи како би се успешно спасао, ако таква постоји.

Улаз. (Улазни подаци се налазе у датотеци `golf.in`) У првом реду улазне датотеке налазе се два реална броја, који представљају x -координату и y -координату Ђурајгерове почетне позиције. У другом реду улазне датотеке налазе се још два реална броја, који представљају x -координату и y -координату почетне позиције палице. У трећем реду налази се природан број n ($1 \leq n \leq 1.000.000$), који представља број рупа на игралишту. У наредних n редова налазе се по два реална броја, при чему се у i -том од тих редова налазе координате i -те рупе.

Издаз. (Издазне податке уписати у датотеку `golf.out`) У први и једини ред издазне датотеке уписати редни број рупе у коју Ђурајгер треба да се сакрије (уколико постоји више таквих, наћи било коју), односно `Nadrljao je!` уколико таква рупа не постоји.

Пример 1.

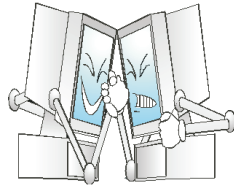
<code>golf.in</code>	<code>golf.out</code>
0.0 0.0	1
5.0 5.0	
3	
1.0 1.0	
9.0 9.0	
4.0 5.0	

Пример 2.

<code>golf.in</code>	<code>golf.out</code>
0.0 0.0	Nadrljao je!
5.0 5.0	
2	
9.0 9.0	
4.0 5.0	

Објашњење. Слика испод приказује распоред рупа, као и позиције на којима се налазе Ђурајгер и палица за први пример. Такође је стрелицом показано која је то рупа спасоносна за Ђурајгера.





Проблем 2. Рафаело куглице

Мирко и Славко су добили на поклон неколико кутија рафело куглица. Да се не би посађали око расподеле, Мирко је предложио следеће: наизменично ће узимати (и јести) по једну рафаело куглицу из произвољне кутије, и онај ко узме последњу куглицу из неке кутије добија као награду све преостале куглице. Пошто је Мирко предложио начин расподеле, Славко има предност да бира да ли ће први почети да узима, или ће то задовољство препустити Мирку. Наравно, Славко жели да поједе што више куглица, па је на вама да му дошапнете шта да ради (да ли да узима први или други). Претпоставља се да и Мирко и Славко узимају тако да поједу што је више могуће куглица.

Улаз. (Улазни подаци се налазе у датотеци `rafaelo.in`) Улазна датотека садржи тачно три реда, а у сваком реду је један тест пример. Сваки ред улазне датотеке садржи следеће податке: број кутија K ($2 \leq K \leq 50$), а затим K бројева из опсега $[1, 100]$ (они представљају количине куглица у кутијама).

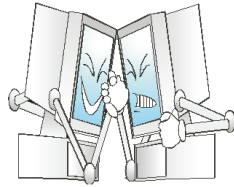
Изаз. (Излазне податке уписати у датотеку `rafaelo.out`) За сваки од три тест примера из улазне датотеке, у посебан ред излазне датотеке исписати 1 ако Славко треба да узима први, односно 2 ако треба да узима други.

Пример 1.

<code>rafaelo.in</code>	<code>rafaelo.out</code>
2 2 2	2
3 3 2 1	1
4 3 2 3 3	1

Објашњење. Важи следеће:

- 1. комбинација - из које год кутије први да узме, у тој кутији ће остати 1 куглица, коју онда узима други и тиме добија и све остале куглице
- 2. комбинација први може одмах да узме куглицу из последње кутије, и тиме добија све остале куглице
- 3. комбинација - уколико неко узме куглицу из кутије која има 2 куглице, тада други добија све остале. Наизменичним узимањем из кутије која има 3 куглице, добијамо да први може да поједе више.



Проблем 3. Бројање

Након што су се фино најели рафаело куглица, Мирко и Славко су одлучили да помоћу теста утврде колико је могуће држати концентрацију са пуним стомаком. Тест се састоји у томе да Мирко говори бројеве Славку (изговарајући сваки пут један од 400 омиљених бројева), и у произвољном моменту тражи од њега да му каже који је K -ти број по величини од свих бројева које је рекао до тада. На вама је да помогнете Славку у одговарању на задата питања. Разлог зашто желите да помогнете Славку није битан.

Улаз. (Улазни подаци се налазе у датотеци `brojanje.in`) У првом реду улазно датотеке налази се број N ($5 \leq N \leq 100.000$). Сваки од наредних N редова има један од два формата:

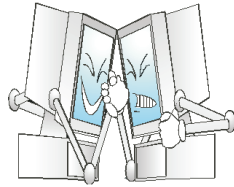
- 1 a - означава да је Мирко изговорио број a ($0 \leq a \leq 65535$).
- 2 k - означава да је Мирко тражио од Славка да му каже који је k -ти број по величини (гарантује се да ће k бити мање или једнако тренутном броју изговорених бројева)

Напоменимо још једном да ће број различитих изговорених бројева бити не већи од 400 (поједини бројеви се могу понављати).

Излаз. (Излазне податке уписати у датотеку `brojanje.out`) За сваки ред из улазне датотеке који је облика ' $2 k$ ', исписати у нов ред излазне датотеке одговор на Мирково питање. Постојаће бар један такав ред.

Пример 1.

<code>brojanje.in</code>	<code>brojanje.out</code>
7	0
1 0	5
1 1	2
1 5	
2 1	
2 3	
1 2	
2 3	



Проблем 4. Бурићи

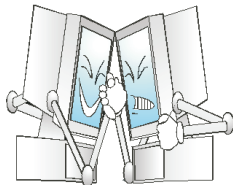
Дато је n бурића. У сваком бурету се налази одређена количина воде. Перица може да пробуши укупно m рупа на днима бурића ($m \geq n$). Кроз сваку пробушену рупу излази 1 литар воде у секунди. Перица све рупе буши истовремено и жели да их пробуши тако да што пре ни у једном бурету не остани ни мало воде (тј. да сва вода истече што пре). Одредити колико је минимално време после бушење након кога ни у једном бурету неће више бити воде.

Улаз. (Улазни подаци се налазе у датотеци `burici.in`) У првом реду текстуалне датотеке налазе се редом природни бројеви n (број бурића, $n \leq 50.000$) и m (број рупа, $m \leq 400.000$). У другом реду налази се n природних бројева (сваки је мањи или једнак 2.000.000.000) тако да i -ти ($1 \leq i \leq n$) број означава број литара у i -том бурету.

Излаз. (Излазне податке уписати у датотеку `burici.out`) У првом реду текстуалне датотеке исписати један реалан број а то је минимално време које се тражи заокружено на две децимале (признаје се свако решење које се од званичног решења разликује по апсолутној вредности за не више од 0.01).

Пример 1.

<code>burici.in</code>	<code>burici.out</code>
3 9	2.00
6 10 2	



Проблем 5. Војници

Перица игра једну игру на свом рачунару. Он има n својих војника од којих сваки има неку јачину. Дато је и n противничких војника од којих сваки такође има неку јачину. Јачине тих $2n$ војника су различите (тј. не постоје два војника са једнаким јачинама). Перица треба да уради следећу ствар: треба да састави n парова војника тако да се сваки пар састоји од једног његовог и једног противничког војника и да се сваки од $2n$ војника појављује у тачно једном пару. И тада креће битка. У сваком од n двобоја (у i -том двобоју ($1 \leq i \leq n$) учествују војници i -тог пара) побеђује војник који је јачи. За сваког од n противничких војника дата су по два броја: један који говори колико Перица добија поена уколико његов (Преичин) војник победи тог војника и други који говори колико Перица губи поена уколико његов војник изгуби од тог војника. Перица на почетку има 0 поена. Одредити колики је максималан број поена који Перица може скупити (тај број може бити и негативан).

Улаз. (Улазни подаци се налазе у датотеци `vojnici.in`) У првом реду текстуалне датотеке налази се природан број n ($n \leq 2.000$). У другом реду налази се n природних бројева: i -ти од тих бројева ($1 \leq i \leq n$) представља јачину i -тог Перичиног војника (сваки од бројева је мањи или једнак 2.000.000.000). У трећем реду налази се n природних бројева: i -ти број у том реду ($1 \leq i \leq n$) представља јачину i -тог противничког војника (сваки од бројева је мањи или једнак 2.000.000.000). У четвртном реду налази се n природних бројева: i -ти број ($1 \leq i \leq n$) представља број поена који Перица добија уколико је тај противнички војник поражен (сваки од бројева је мањи или једнак 1.000). У петом реду налази се n природних бројева: i -ти број ($1 \leq i \leq n$) представља број поена који Перица губи уколико је тај противнички војник у двобоју у коме је учествовао изашао као победник (сваки од бројева је мањи или једнак 1.000).

Излаз. (Излазне податке уписати у датотеку `vojnici.out`) У првом реду текстуалне датотеке исписати један цео број а то је максималан број поена који Перица може скупити.

Пример 1.

<code>vojnici.in</code>	<code>vojnici.out</code>
3	14
9 12 3	
4 5 6	
10 2 7	
5 3 1	