



Задача А. Топки

Софтуерна компания е разработила играта "Топки". Играе се от един играч на правоъгълна дъска, разделена на $N \times M$ клетки, като всяка клетка е определена с координатите си (X, Y) – номера на реда и номера на колоната, съответно. Клетките на дъската могат да бъдат *свободни*, т.е. такива, в които може да се постави една топка, или *блокирани*. Играчът получава начална позиция с B топки върху дъската и крайната позиция с E топки, която трябва да достигне от началната, с някои от следните три операции:

- Първата операция може да се приложи към свободна клетка, ако в нея няма топка. В резултат в клетката се появява топка, а играчът получава A наказателни точки;
- Втората операция може да се приложи към свободна клетка с топка. В резултат топката изчезва, а играчът получава B наказателни точки;
- Третата операция може да се приложи към двойка съседни свободни клетки, ако в първата има топка, а във втората – няма. В резултат на операцията топката се премества от първата клетка във втората, а играчът получава C наказателни точки. Две клетки (X_1, Y_1) и (X_2, Y_2) са съседни, ако $|X_1 - X_2| + |Y_1 - Y_2| = 1$.

Напишете програма, която намира решение с най-малък брой наказателни точки.

Вход. На първия ред на стандартния вход ще бъде зададен броят на тестовете, които програмата трябва да реши при едно изпълнение. Първият ред за всеки тест ще съдържа, разделени с един интервал, двете цели числа N и M – броя на редовете и броя на колоните, съответно, а вторият ред – целите A , B и C , разделени с по един интервал. Всеки от следващите N реда съдържа по M знака $S_{i,j}$, с които се описва началната позиция на топките: Ако $S_{i,j} = \#$, тогава клетката с координати (i, j) е блокирана. Ако $S_{i,j} = \cdot$, тогава клетката с координати (i, j) е свободна и не съдържа топка, а ако $S_{i,j} = *$, тогава клетката с координати (i, j) е свободна и съдържа топка. Следва един празен ред, а след него N реда с по M знака, които по подобен начин описват крайната позиция.

Изход. За всеки тестов пример, в реда по който са зададени на входа, програмата трябва да изведе на стандартния изход ред с цялото число R такова, че най-евтиният начин за постигане на крайната позиция е с R наказателни точки

Ограничения. $3 \leq N, M \leq 60, 0 \leq A, B, C \leq 1000$

Пример

Вход	Изход
1 5 6 4 5 2 **#... ..*## .*.*.* ..#.# ..#.*. *.#...* ..*## **.... ..#.# **#*..	29

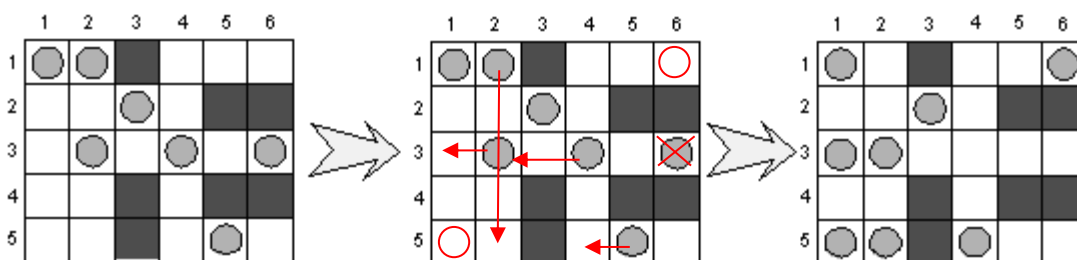
Обяснение на примера (виж Фигурата).

С третата операция преместваме топката от $(3,2)$ наляво в $(3,1)$ за 2 точки. След това с прилагане 4 пъти на третата операция преместваме топката от $(1,2)$ в $(5,2)$ за 8 точки. Сега, с прилагане 2



XXXV РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
13.05.2013, Университет „Асен Златаров“, Бургас

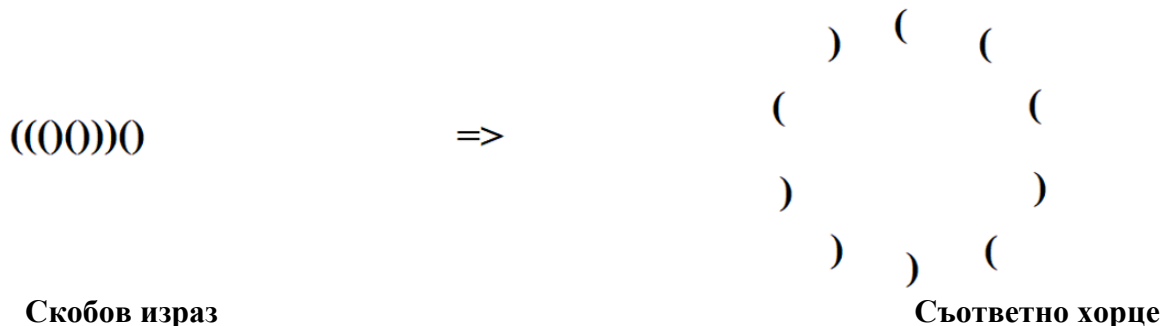
пъти на третата операция, топката от (3,4) отива в (3,2) за 4 точки, а с прилагане 1 път на третата операция топката от (5,5) отива в (5,4) за 2 точки. С първата операция поставяме нови топките в (5,1) и (1,6) за 8 точки, а с втората операция премахваме топката от (3,6) за 5 точки. Крайният резултат е $8.2 + 2.4 + 1.5 = 29$ точки.





Задача В. Хорце

Скобов израз е низ, съдържащ само отварящи и затварящи скоби. *Правилен скобов израз* е такъв скобов израз, който отговаря на условието за поставяне на скоби в аритметичен израз (в математиката и информатиката), т.е. броят на отварящите скоби е равен на броя на затварящите, а броят на отварящите скоби от началото на низа до всяка негова позиция е по-голям или равен на броя на затварящите. *Хорце (от скоби)* е скобов израз, чийто последен знак е съседен на първия и така се получава **кръгов низ**. Например, най-горната отваряща скоба от хорцето, показано вдясно на фигурата, е първата в низа, –показан вляво на фигурата, а тази в ляво от нея – последната.:



Напишете програма, която да пресметнете на колко места може да бъде разкъсано дадено хорце така, че полученият след такава операция скобов израз (съдържащ всички скоби от хорцето в зададения ред) да е правилен.

Вход. На първия ред на стандартния вход ще бъде зададен броя T на тестовите примери, които програмата трябва да реши при едно извикване. На всеки от следващите T реда ще бъде зададен по един низ от скоби, който представлява хорце от скоби.

Изход. За всяко едно от хорцата, зададени на входа, програмата трябва да изведе на отделен ред на стандартния изход броя на местата, където то може да бъде разкъсано така, че полученият след разкъсването скобов израз да е правилен.

Ограничения: За дължината L на всеки от зададените на входа низове е в сила $1 \leq L \leq 10^6$.

ПРИМЕР

Примерен вход	Изход за примера
2	0
(())	2
((())) ()	

Обяснение на изхода: Първото хорце не може да бъде разцепено коректно, тъй като броят на затварящите скоби е по-голям от броя на отварящите. Второто хорце може да бъде разцепено на две места: преди първия знак на входния низ и след осмия знак. Така ще се получат правилните скобови изрази ((())) () и () ((())) .



Задача С. Геноми

Известно е, че ДНК молекулите могат да бъдат представени като някаква последователност от знаците 'A', 'G', 'T' и 'C', наречена *линеен геном*. Учените установили, че геномите на някои микроорганизми не са линейни, а имат вид на пръстен. Ако $S = s_1s_2\dots s_n$ е една последователност от знаците в *пръстеновиден геном*, тогава линейните геноми $S_1 = S = s_1s_2\dots s_n$, $S_2 = s_2s_3\dots s_ns_1$, ..., $S_n = s_ns_1\dots s_{n-2}s_{n-1}$ наричаме *модификации* на този пръстеновиден геном. *Степен на сходство* на два линейни генома е броят еднакви знаци, стоящи на еднакви позиции в съответните последователности. *Степен на сходство* на два пръстеновидни генома A и B е максималната стойност на степента на сходство за всички техни модификации A_x и B_y . Напишете програма, която по зададени две модификации на пръстеновидните геноми A и B да определите степента им на сходство.

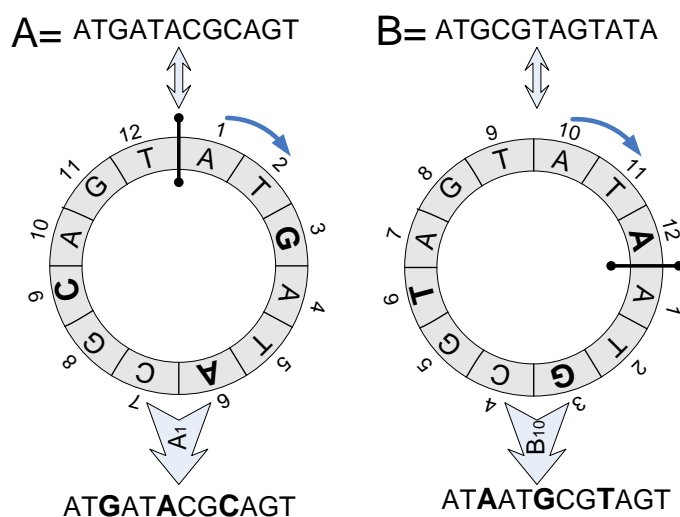
Вход. Програмата трябва да реши няколко тестови примера при едно изпълнение. За всеки тестов пример на един ред на стандартния вход ще бъдат зададени модификациите на A и B , разделени с интервал – низове с еднаква дължина, съставени от знаците 'A', 'G', 'T' и 'C'.

Изход. За всеки тестов пример на отделен ред на стандартния изход програмата трябва да изведе намерената степен на сходство на двата пръстеновидни генома.

Ограничение. Дължините на низовете няма да надхвърлят 100 знака.

Пример

Примерен вход	Изход за примера	Обяснение (несъвпадащите знаци са с удебелен шрифт)
AG TA	1	$A_1=AG$ $B_2=AT$
AGTC CTGA	2	$A_2=GTCA$ $B_3=CACT$
ATGATACGCAGT ATGCGTAGTATA	9	$A_1=ATGATACGCAGT$ $B_{10}=ATAATGCGTAGT$



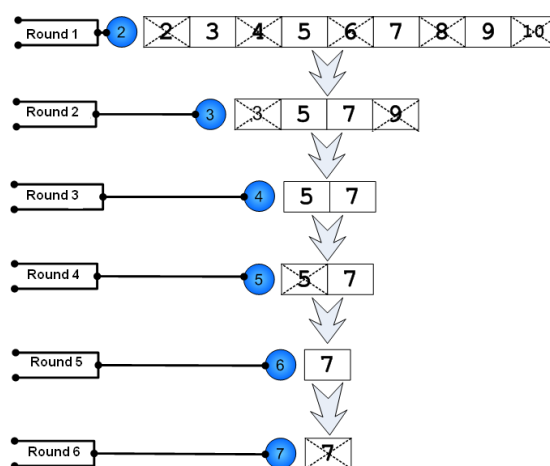
Илюстрация за третия тестов пример от Примерния вход. Несъвпадащите знаци са удебелени



Задача D. Бинго

В едно от многобройните казина в град Б. се предлага нов, интригуващ вариант на играта „Бинго“. За разлика от оригиналната игра, във всеки талон, който играчите купуват, са зададени целите положителни числа, принадлежащи на затворения интервал $[L, R]$. В примера по-долу е показан талон с $L = 2$ и $R = 10$, съдържащ числата 2, 3, 4, 5, 6, 7, 8, 9 и 10. Играта, както и оригиналното Бинго се провежда в няколко рунда по следния начин: в първия рунд крупиеето обявява за изтеглено числото 2, а във всеки следващ рунд – число, с едно по-голямо от числото в предишния рунд. Във всеки рунд, след като крупиеето обяви число N , играчите задраскват в талона си числата, които се делят на N без остатък. Играта печели този играч, който пръв задраска всички числа в талона си.

Например, при обявяване на числото 2, играчът с талона $[2, 10]$ зачертава числата 2, 4, 6, 8, 10, а числата 3, 5, 7, 9 остават незадраскани. След това крупиеето обявява номер 3, а играчът задрасква числата 3 и 9, като числата 5 и 7 остават незадраскани. При обявяване на номер 4 в третия рунд играчът не задрасква нищо, а при обявяване на номер 5 в четвъртия рунд задрасква 5. В петия рунд крупиеето обявява номер 6 и играчът не задрасква нищо, а в шестия рунд, след като крупиеето обяви номер 7, играчът задрасква и последното число 7. За него играта е приключила след 6 рунда и ако никой друг до този момент не е обявил „Бинго!“ той печели играта.



Напишете програма, която по зададен талон да определя в кой рунд играчът с този талон би спечелил играта.

Вход. На стандартния вход ще бъдат зададени няколко тестови примера, като за всеки пример на отделен ред ще бъдат зададени числата L и R , разделени с един интервал.

Изход. За всеки тестов пример, на отделен ред на стандартния изход, програмата трябва да изведе, намерения брой рундове за съответния талон.

Ограничения. $2 \leq L \leq R \leq 10^9$.

Пример

Примерен вход	Изход за примера
2 10	6
8 10	2
7 292	282



Задача Е. Приятели

На Крис му е скучно в часовете по математика, затова направил списък от N големи цели числа. В списъка има няколко двойки числа, които харесва, и някои двойки, които не харесва. Крис нарекъл двойките, които харесва, *приятели*. Две числа са приятели, ако имат поне една обща цифра (не непременно на една и съща позиция). Помогнете на Крис да преброи колко двойки числа в неговия списък са приятели.

Вход. Програмата трябва да обработва няколко тестови примера при едно извикване. За всеки тест на първия ред на стандартния вход е зададено едно цяло число N – броят на числата в списъка. Всеки от следващите N реда съдържа по едно цяло число a от списъка на Крис. В списъка няма равни числа.

Изход. За всеки тест, на отделен ред на стандартния изход, програмата трябва да изведе броя на двойките приятели в списъка.

Ограничения. $1 \leq N \leq 10^6$, $1 \leq a \leq 10^{18}$.

Пример

Примерен вход	Изход за примера
3	1
4	4
20	
44	
4	
32	
51	
123	
282	



Задача F. Брокер

Иво Икономиста решил, че е крайно време да използва уменията си, за да спечели милиони от търговия с притежавани от него еднотипни акции. По правилата за търговия с ценни книжа на избраната от него фондова борса, всеки ден се провежда по един търг. Брокер може да прави само една сделка на ден, като или купува една акция, или продава една акция в този ден. Цените за покупка и продажба на една акция през деня са равни. Иво „случайно“ научил (закъде си на борсата без информация), че в деня, в който започва да търгува цената на една от притежаваните от него акции е 1 евро, а във всеки следващ ден ще поскъпва с едно евро на ден.

Иво има намерение да участва в търга в продължение на K последователни дни, като всеки ден трябва или да купи една акция, или да продаде една акция. Стратегията му е много проста: ако има достатъчно пари, за да купи акция, той я купува, а в противен случай продава една от акциите си. В началото на търга той има N евро и не по-малко от K акции. Напишете програма, която да определи колко пари ще има Иво в края на K -тия ден от участието си в търга, спазвайки стратегията си.

Вход. На първия ред на стандартния вход ще бъде зададен броя на тестовете, които програмата трябва да обработи при едно изпълнение. За всеки тест на отделен ред на стандартния вход ще бъдат зададени, разделени с интервал, целите N и K .

Изход. За всеки тест на отделен ред на стандартния изход програмата трябва да изведе сумата, която Иво ще има в края на K -тия ден от участието си в търга.

Ограничения. $1 \leq N, K \leq 10^{15}$.

Пример

Примерен вход	Изход за примера
2	5
7 4	1 6 4 7
1000 2010	

Обяснение: За първия тест от примерния вход, в първия ден Иво купува акция за 1 евро и му остават 6 евро; във втория ден купува акция за 2 евро и му остават 4 евро; в третия ден отново купува една акция за 3 евро и му остава 1 евро, затова в четвъртия ден продава една от акциите си за 4 евро и завършва с 5 евро.



Задача G. Тестване

Последните години Джени напредна много в състезателната информатика и вече не само че може да решава много задачи, но и започна да предлага свои. Но целият процес около написване на генератор, със средно 200-300 реда код, и генериране на тестовете ѝ е омръзнал. Новата задача, по която работи Джени, е със следното условие: „Дадено е претеглено дърво с N върха и константата K . Задачата е да се намери такова множество от ребра E , с минимална сума на теглата на ребрата в него, че като се премахнат ребрата от E от дървото, да се получи дърво (или поддърво) с K върха.“

За да си спести усилия, Джени решила да копира тестове от друга своя задача, в която също е зададено претеглено дърво, като за всеки тест само добави произволно K . Тъкмо привършила с промените в тестовете и започнала да пише пример в условието и ... изненада! Нейното решение давало грешен отговор на примера, който измислила. Съответно, не е много ясно доколко верни са отговорите и за останалите тестове, които е използвала. Затова се обръща с голяма молба към Вас, да сте неин тестер, като напишете програма, която по зададени тестове, намира отговорите им.

Вход. На първия ред на стандартния вход ще бъде зададен броят на тестовете T , които програмата трябва да реши при едно извикване, а на следващите редове ще бъдат данните за тестовете. За всеки тестов пример на първия ред на стандартния вход ще бъдат зададени целите N и K – броят върхове в дървото и броят на върховете в дървото/поддървото, което трябва да се отдели. На всеки от следващите $N - 1$ реда на тестовия пример ще бъдат зададени по три цели числа x, y и t , които показват, че има ребро в зададеното дърво между върховете с номера x и y , а тегло на това ребро е t (върховете са номерирани с числата от 1 до N).

Изход. За всеки тест, по реда им на въвеждане, програмата трябва да изведе на отделен ред на стандартния изход минималната сума от теглата на ребрата, които да бъдат премахнати така, че да получим поне едно дърво със зададения брой върхове.

Ограничения. $1 \leq N \leq 10^3$; $1 \leq K \leq \min(N, 2 \cdot 10^2)$; $1 \leq x, y \leq N$; $1 \leq t \leq 10^6$.

Пример

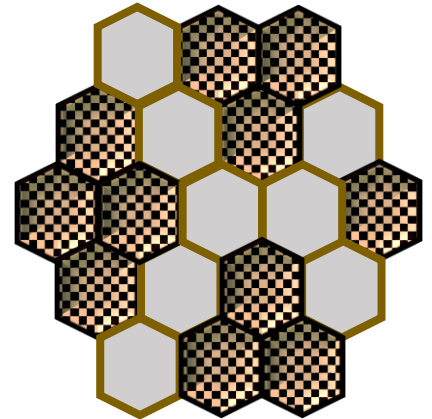
Примерен Вход	Изход за примера	Обяснение
2 6 3 1 2 3 1 3 1 2 4 1 2 5 2 2 6 1 5 2 1 2 3 1 3 1 2 4 1 2 5 2	3 3	<p>В червено са показани оптимални решения за двата теста:</p> <p>Забележете, че във втория пример, след изтриване на реброто между върхове 1 и 2, се получава гора от две поддървета. Едното от тях е дървото с върхове 1 и 3, което е съставено от два върха, каквото се търси в този тестов пример.</p>



Задача Н. Бактериални изследвания

Шестоъгълникът, показан на фигурата, е съставен от по-малки еднакви правилни шестоъгълни клетки, подредени в редове по следния начин:

- на първия ред поставяме 3 шестоъгълни клетки, допрени една до друга;
- на следващия ред поставяме 4 клетки, допрени една до друга и допрени до горния ред.
- Във всеки от следващите редове увеличаваме с едно броя на клетките, докато получим ред с максимален брой клетки.
- След това, във всеки следващ ред намаляваме с едно броя на клетките и завършваме с ред от 3 клетки.



За две клетки казваме, че са *съседни* ако имат обща страна. Част от клетките, оцветени на фигурата в кафяво, са заразени. Няколко заразени съседни клетки, оградени със здрави клетки и/или граничещи с контура на хексагона, образуват *остров*. Напишете програмата, която намира броя на островите, образувани от заразени клетки на даден шестоъгълник.

Вход. На стандартния вход ще бъдат зададени няколко тестови примера. За всеки тестов пример, на първия ред ще бъде зададен номерът K на най-дългия ред на шестоъгълника (броенето на редовете започва от 1), последван от толкова редове, колкото са редовете на шестоъгълника. Във всеки от тези редове е зададено, от ляво надясно, състоянието на поредната клетка в реда: 1 за заразена клетка и 0 за здрава, разделени с по един интервал.

Изход. За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход броя на островите от заразени клетки.

Ограничение. $2 < K < 2400$.

Пример (първият тест е за шестоъгълника от Фигурата):

Вход	Изход
3	4
0 1 1	2
1 0 1 0	
1 1 0 0 1	
1 0 1 0	
0 1 1	
2	
1 0 0	
0 1 0 1	
0 1 0	



Задача I. Търговец

Имало едно време един търговец, който решил да тръгне на пътешествие из различни градове и да продава стоки с цел печалба. За превозване на стоките търговецът използва малко ремарке с определена товароносимост. За всяка от стоките се знае наличното количество в килограми в склада на търговеца и цената за един килограм.

Напишете програма, която определя от коя стока какво количество да се вземе, така че общата стойност да е възможно най-голяма, като общото тегло не трябва да надхвърля товароносимостта на ремаркетото.

Вход. Програмата трябва при едно изпълнение да въведе от стандартния вход и да обработи няколко теста. Първият ред за всеки тест съдържа цяло число m – максималното тегло, което може да вози ремаркетото. От втория ред се въвежда броят n на различните стоки в склада на търговеца. Всеки от следващите n реда съдържа по две числа с дробна част a_i и b_i , където a_i е наличното количество в кг от i -тата стока, а b_i е цената в лева на 1 кг от i -тата стока. Числата a_i имат три цифри, а числата b_i – две цифри след десетичната запетая. Между данните за отделните тестове има по един празен ред.

Изход. За всеки тестов пример, в реда по който са зададени на входа, програмата трябва да изведе на стандартния изход ред с число s – максималната възможна стойност в лева на натоварените в ремаркетото стоки. На следващите n реда за всяка стока трябва да се изведе количеството c_i в кг, което трябва да се вземе от нея. След извеждането на данните за всеки тест да се извежда по един празен ред.

Числото s трябва да се изведе с две цифри след десетичната точка, а числата c_i , $i = 1, 2, \dots, n$ с три цифри след десетичната точка.

Упътване. В C++ за извеждане на стойност x от тип `double` с две цифри след десетичната точка използвайте `cout << fixed << setprecision(2) << x << endl;` и съответно `cout << fixed << setprecision(3) << x << endl;` за извеждане с три цифри след десетичната точка.

Ограничения. $1 \leq m \leq 1000$, $1 \leq n \leq 100$, $1 \leq a_i \leq 100$, $1 \leq b_i \leq 100$

Пример

Вход	Изход
50	240.00
3	10.000
10.000 6.00	20.000
30.000 4.00	20.000
20.000 5.00	
	67.45
100	10.250
1	
10.250 6.58	



Задача J. Монети

Имало едно време малко селце, в което хората използвали различни видове монети за ежедневните си трансакции. Всяка монета имала различна стойност и хората използвали различни комбинации от монети, за да плащат за покупките си. Имало обаче един проблем – жителите на селото често се затруднявали да образуват дадена сума, особено когато нямали достатъчно монети от същия номинал.

Помогнете на жителите от селцето, като напишете програма, която намира минималния брой монети, необходими за образуването на дадена сума.

Вход

Програмата трябва при едно изпълнение да въведе от стандартния вход и да обработи няколко теста. Първият ред за всеки тест съдържа две цели числа: s и n , разделени с интервал, s представлява сумата, която трябва да се образува, а n представлява броят на различните номинали на наличните монети. Вторият ред за всеки тест съдържа списък от n цели числа, разделени с интервал, които представляват стойността на всеки номинал на монетите.

Изход

За всеки тестов пример на един ред на стандартният изход се извежда едно цяло число, представляващо минималния брой монети, необходими за образуването на дадената сума.

Ограничения

$$1 \leq s, n \leq 1000$$

Пример

Вход	Изход
50 3	2
25 10 5	5
100 5	
1 2 5 10 20	



Задача К. Къде е топчето?

За да убият времето си, двамата приятели Димо и Иво играят една игра, която са видели на градския пазар. Димо поставя три обърнати чаши на масата и слага малко топче под едната от тях. След това Димо разменя местата на двойки чаши, докато Иво се опитва да познае местоположението на топчето.

Стандартната версия на играта, която Димо и Иво са видели да се играе на градския пазар „Тука има, тука нема“, позволява на играча да види началното местоположение на топчето, след което трябва да познае неговото крайно местоположение след размените.

Все пак, Димо и Иво обичат да играят на версия, в която Иво не знае началното местоположение на топчето и може да познае местоположението му след всяка размяна. Димо, който знае правилния отговор, дава на Иво оценка на края, която е равна на броя правилни познавания.

Даден е броят на размените и познаванията, но не и началното местоположение на кръгчето. Моля, определете най-голямото възможно количество точки, които Иво може да е спечелил.

Вход. На първия ред на стандартния вход ще е зададен броят T на тестовете. Следват T теста. Всеки от тестовете започва с ред, който съдържа цяло число N , което е броят на размените. Всеки от следващите N реда описва стъпка от играта и съдържа три цели числа **A**, **B** и **C**, които указват, че Димо е разменил местата на чашите **A** и **B**, след което Иво е предположил, че топчето е под чаша **C** след размяната. Всички тези цели числа са или 1, или 2, или 3, и $A \neq B$.

Изход. За всеки тест да се изведе едно единствено число - максималния брой точки, които Иво може да е спечелил.

Ограничения: $1 \leq T \leq 100$, $1 \leq N \leq 100$

Пример

Вход	Изход
1	2
3	
1 2 1	
3 2 1	
1 3 1	

Пояснения за примера: Отговорът е 2. Ако топчето е под чаша номер 1, Иво ще познае мястото на топчето точно веднъж (при последното си предположение). Ако топчето е под чаша номер 2, той ще познае мястото на топчето два пъти (при първите две предположения). Ако топчето е под чаша номер 3, той няма да направи нито едно правилно предположение.



Задача L. Изследователи

Група от млади изследователи решили да направят експедиция в дълбоката джунгла на Южна Америка. Целта им е да изследват местните растителни и животински видове, да съберат проби от почвата и минералите и да изучат климатичните условия в района. Екипът се състои от петима членове, включително един ботаник, един зоолог, двама метеоролози и един географ. Те ще се нуждаят от всички свои умения и знания, за да оцелеят в джунглата и да извършат успешно мисията си. Пътуването им започва с малък самолет, който ще ги отведе до малко селище в близост до границата на джунглата. От там трябва да продължат пътя с коли и след това пеша. Те трябва да преодолеят голямо разстояние и да преборят много предизвикателства, но всичко това ще им помогне да открият неизвестни светове и да се запознаят с тайните на дивата природа.

След като пристигат в селището, младите изследователи се запознават с местните жители и се обзавеждат с необходимото оборудване за пътуването в джунглата. Подготовката включва закупуване на специална екипировка, като за тяхно щастие, местните майстори имат голям опит в изработването на инструменти и средства за оцеляване в условията на джунглата. След като екипът е готов за изпращането пред предизвикателствата на експедицията, започва изследването на джунглата.

Макар че са добре подготвени, на изследователите предстои да се сблъскат с множество препятствия – неравни терени, гъста растителност, опасни животни и други неща, които могат да ги забавят и да усложнят работата им. Младите изследователи разполагат с карта на джунглата, на която са отбелязани N места, номерирани от 1 до N , които са от интерес за експедицията, и M пътеки, които свързват някои от местата. Те се намират в мястото с номер 1, а трябва да се доберат безопасно до мястото с номер N , където ще разположат базовия си лагер. Пътеките са такива, че от място 1 може да се достигне до всяко друго място.

При избора на маршрута е необходимо да се има предвид, че изследователите нямат право да се движат два пъти подред от място с по-малък номер към място с по-голям номер или от място с по-голям към място с по-малък номер. Т.е. трябва да се движат на зиг-заг, като започнат от мястото с номер 1, след това отидат до място с по-голям номер, след това към място с по-малък номер от предното и така нататък. В общността на Шуарите, за която е картата, вярват, че зигзагообразното движение е по-трудно за следене и затова отблъсква потенциалните преследващи изследователите зли духове. На гърба на картата пише, че ако не спазват това правило, ще ги сполети проклятието на местния шаман Аяуска и ще умрат. Това е част от вярванията на Шуарите, които възприемат пътуването като духовен опит и внимателно спазват своите традиции и обичаи, за да се предпазят от опасностите по пътя.

Изследователите са добре подготвени, обсъждат различни варианти и правят изчисления, за да намерят оптималното решение – най-късия зигзагообразен маршрут. Напишете програма за да разберете кой е този маршрут.

Вход. На първия ред на стандартния вход ще е зададен броят T на тестовете, които програмата трябва да реши при едно извикване. Следват данните за T теста. Всеки от тестовете започва с ред, който съдържа двете цели числа N и M – броя на местата и броя на пътеките. На всеки от следващите M реда ще бъде зададено описанието на една от пътеките с тройка цели числа указващи: номер на мястото от което тръгва пътеката, номер на мястото в което свършва и дължина на пътеката в километри.

Изход. За всеки тест, на отделен ред на стандартния изход, програмата трябва да изведе едно цяло число – дължината най-краткия зигзагообразен път в джунглата от мястото с номер 1 до мястото с номер N , в километри.

Ограничения: $0 < N \leq 100$, $0 < M \leq 1000$.



XXXV РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
13.05.2013, Университет „Асен Златаров“, Бургас

ПРИМЕР

Примерен вход	Изход за примера
1 5 7 1 2 4 1 3 2 1 4 7 2 3 3 2 4 5 3 4 1 3 5 3	11

Обяснение на примера: Най-късият зигзагообразен път е 1-4-3-5 с дължина 11 км ($7+1+3=11$).



Задача М. Колонизиране на Марс

Жълтите патета колонизират Марс! Сега те трябва да построят изследователска база. Базата трябва да има n модула, които представляват еднакви правоъгълници с размери $a \times b$. Всички модули ще се разположат в рамките на правоъгълно поле с размери $w \times h$. При това, модулите трябва да са във вид на мрежа – страните им трябва да са успоредни на страните на полето и трябва да са ориентирани по един и същ начин.

За защита, патешките инженери могат да добавят на модулите защитен слой. Дебелината на защитния слой трябва да бъде цяло число. Модул, на който е добавено защитно поле с дебелина d ще има размери $(a+2d) \times (b+2d)$. Определете максималната дебелина d на защитен слой, който може да бъде добавен на модулите, така че да могат n модула, да се разположат в полето.

Вход

Програмата трябва да обработва няколко тестови примера.

На първия ред на стандартния вход е дадено едно цяло число T – броя на тестовите примери.

Следват T реда с по пет цели числа n, a, b, w и h – броя модули, размерите на модулите и размерите на полето, съответно.

Изход

За всеки тест, на отделен ред, изведете на стандартния изход едно цяло число – търсената максимална дебелина d на защитния слой, който може да бъде добавен на модулите.

Ограничения

$$1 \leq n, a, b, w, h \leq 10^{18}$$

Всички числа на входа са цели. Гарантирано е, че без допълнителна защита всички модули ще могат да се поберат в полето.

Пример

Вход

1
11 2 3 21 25

Изход

2



Задача N. Най-близък общ предшественик

Дадено е кореново дърво с n ($1 \leq n \leq 100000$) върха, номерирани с целите числа от 0 до $n-1$. Трябва да се отговори на m заявки за намиране на най-близкия общ предшественик на два върха.

Заявките се генерират по следния начин. Дадени са целите числа a_1, a_2 и числата x, y, z . Числата a_3, a_4, \dots, a_m се генерират по следния начин:

$a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Първата заявка е за върховете (a_1, a_2) . Ако отговора на $(i-1)$ -вия въпрос е v , то i -тия въпрос е от вида $((a_{2i-1} + v) \bmod n, a_{2i})$.

Вход

Входът се състои от няколко набора входни данни. Всеки набор от входни данни има следния вид. Първо са дадени целите числа n и m . Корен на дървото е връх 0 . Следва един ред с $n-1$ числа, като i -тото от тях е номера на предшественика на върха i . Следва един ред с целите числа a_1 и a_2 ($0 \leq a_1, a_2 \leq n-1$). Следващият ред съдържа три цели числа x, y, z ($0 \leq x, y, z \leq 10^9$).

Изход

За всеки набор от входни данни изведете на отделен ред едно цяло число – сумата от отговорите на всички заявки за съответния набор от данни.

Пример

Вход	Изход
3 2	2
0 1	
2 1	
1 1 0	