

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача А. МОДУЛНО СОРТИРАНЕ

Дадена е редица a_1, a_2, \dots, a_N от N цели положителни числа и цяло положително число $m > 1$. Напишете програма, която да сортира редицата в нарастващ ред спрямо остатъка получен при делене на елементите на m . Ако две числа имат еднакъв остатък, тогава по-малкото число трябва да бъде първо.

Вход: На първия ред на стандартния вход ще бъде зададен броят на тестовете. За всеки тестов пример стандартният вход съдържа два реда. На първия ред са числата m и N , а на втория – редицата от цели положителни числа, разделени с интервал.

Изход: За всеки тестов пример на един ред на стандартния изход програмата трябва да изведе елементите на сортираната редица, разделени с по един интервал. След последното изведено число не трябва да има интервал.

Ограничения: $N \leq 1000000$, всички останали числа са от тип `int`.

ПРИМЕР

Вход	Изход
2	12 2 6 10 3
4 5	3 1 4 10 5 11
10 3 2 6 12	
3 6	
3 4 5 10 11 1	

Given a sequence a_1, a_2, \dots, a_N of N positive integers and a positive integer $m > 1$. Write a program that sorts the sequence in ascending order by the remainder obtained by dividing the elements by m . If two numbers have the same remainder, then the smaller number must be the first.

Input: The first line of the standard input will specify the number of tests. For each test case, the standard input has two lines. On the first line are the numbers m and N , and on the second – the sequence of positive integers, separated by an interval.

Output: For each test case, the program must output on a row of the standard output the elements of the sorted sequence, separated by one interval. No interval after the last number.

Constraints: $N \leq 1000000$, all other numbers are of type `int`.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача В. ОТРЕЗИ

На метална лента са отбелязани N позиции, номерирани последователно с числата от 1 до N , от ляво надясно. Известни са разстоянията между M от позициите. Напишете програма, която да намери разстоянията между всеки две съседни позиции или да установи че това е невъзможно (т.е. данните са недостатъчно или са противоречиви).

Вход. На първия ред на стандартния вход ще бъде зададен броят на тестовите примери. Всеки пример започва с ред, на който са зададени броят N на позициите и броят M на двойките позиции, разстоянията между които са известни. Следват M реда с по три числа – първите две са номерата B и E на две позиции, а третото – разстоянието R между тези позиции.

Изход. Ако данните в някой тест позволяват да се намерят всички разстояния, тогава програмата трябва да изведе на първия ред на стандартния изход за този тест числото 1, а на втория ($N - 1$) дробни числа – разстоянията между всяка двойка съседни позиции, отляво надясно, разделени с точно един интервал. След последното разстояние не трябва да има интервал. Разстоянията трябва да се изведат с точно три знака след десетичната точка. Ако данните в тестовия пример са недостатъчни или противоречиви, тогава програмата трябва да изведе за този тест само ред с числото 2.

Ограничения. $2 \leq N \leq 200$, $0 \leq M \leq 5000$, $1 \leq B \leq E \leq N$, $0 \leq R < 2^{15}$. Всички разстояния са дробни неотрицателни числа с не повече от 3 знака след десетичната точка.

Пример

Вход	Изход
2	1
3 2	1.250 1.750
1 2 1.250	2
3 1 3	
3 2	
1 2 1.000	
1 3 0.005	

On a metal strip N positions are labeled consecutively from 1 to N , left to right. The distances between some of the positions are known. Write a program to find the distances between any two adjacent positions or to find that this is impossible (i.e. the data is insufficient or inconsistent).

Input. The first line of the standard input will contain the number of the test cases. Each test case begins with the number N of the positions and the number M of the pairs of positions, the distances between which are known. Each of the following E lines contains three numbers – the first two are the positions B and E , and the third is the distance R between these positions.

Output. If the data in a test allows all distances to be found, then the program must output on the first line of the standard output, for this test, the number 1, and on the next – the $N - 1$ distances between the consecutive positions from left to right separated by a single interval and without an interval after the last distance. The distances must be printed with exactly 3 digits after the decimal point. If the data in the test case is insufficient or contradictory, then the program should only output the number 2 for this test.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Constraints. $2 \leq N \leq 200$, $0 \leq E \leq 5000$, $1 \leq B \leq E \leq N$, $0 \leq R < 2^{15}$. All distances are fractional non-negative integers with no more than 3 digits after the decimal point.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача С. ТРОИЧНА СИМЕТРИЧНА

На далечната планета Трития живеят странни същества. Те имат по три очи, три уши, три крака, три ръце. И логиката им е различна – на въпроси за верността на съждение, освен с *Да* и *Не*, те често отговарят и *Знам ли*. Затова и компютрите им не са двоични, както при нас, а троични. Целите числа на тази планета се представят с три цифри, които според нашите традиции можем да означим със знаците -, 0 и +. Когато цяло число е записано в такава, *троично-симетрична*, позиционна бройна система като $a_n a_{n-1} a_{n-2} \dots a_1 a_0$, тогава ако цифрата a_i е -, тя „внася“ в стойността на числото -3^i , ако е + внася 3^i , и внася 0, ако цифрата е 0. Например, троично симетричното число $++-0-+$ е десетичното число $3^5 + 3^4 - 3^3 - 3^1 + 3^0 = 243 + 81 - 27 - 3 + 1 = 295$, а пък числото $--+0-+$ е десетичното число $-3^5 + 3^4 - 3^3 - 3^1 + 3^0 = -243 + 81 - 27 - 3 + 1 = -191$. Иначе, цивилизацията на тази планета е малко по-развита от нашата, като най-често използваните компютри са 64 тритови (*тритът*, за разлика от бита, има три различни стойности, съответни на трите цифри на бройната система). Представете си, че трябва да реализирате обичайните аритметични операции събиране и изваждане в тази система. Напишете програма, която по зададени две числа τ_1 и τ_2 в троично-симетрична система намира тяхната сума $\tau_1 + \tau_2$ и разлика $\tau_1 - \tau_2$.

Вход. На първия ред на стандартния вход ще бъде зададен броят T на тестовите примери, които програмата трябва да обработи. За всеки тестов пример, на два отделни реда, ще бъдат зададени две троични симетрични числа τ_1 и τ_2 .

Изход. За всеки тестов пример програмата трябва да изведе по два реда. На първия от тях – сумата $\tau_1 + \tau_2$, а на втория – разлика $\tau_1 - \tau_2$.

Ограничения. Броят на цифрите на всяко от двете числа не надхвърля 63.

ПРИМЕР

Вход	Изход
2	++0--
++-0-+	+00000
--+0-+	++00-0+00
++0-++-0-	++0-0+0-+
+0-0+	

Strange creatures live on the distant planet Thritia. They have three eyes, three ears, three legs, three hands. And their logic is different – questions about the truth of a judgment, except *Yes* and *No*, they often answer with *Who knows*. Therefore, their computers are not binary, as in our case, but ternary. The integers on this planet are represented by three digits, which according to our traditions, can be denoted by the signs -, 0 and +. When an integer is written in such a *ternary symmetric* positional numeric system such as $a_n a_{n-1} a_{n-2} \dots a_1 a_0$, then if the digit a_i is -, it "imports" into the value of the number -3^i , if + then it imports 3^i , and imports 0 if the digit is 0. For example, the ternary symmetric number $++-0-+$ is the decimal number $3^5 + 3^4 - 3^3 - 3^1 + 3^0 = 243 + 81 - 27 - 3 + 1 = 295$, and the number $--+0-+$ is the decimal number $-3^5 + 3^4 - 3^3 - 3^1 + 3^0 = -243 + 81 - 27 - 3 + 1 = -191$. Otherwise, the civilization of this planet is a little more advanced than ours, with the most commonly used computers being 64-trits (trit, unlike bit, have three different values corresponding to the three digits of the number system). Imagine having to perform the usual arithmetic operations of addition

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

and subtraction in this system. Write a program that, given two numbers τ_1 and τ_2 in a ternary-symmetric system, finds their sum $\tau_1 + \tau_2$ and difference $\tau_1 - \tau_2$.

Input. The first line of standard input will contain the number of the test cases that the program must process. For each test case, two ternary-symmetric numbers τ_1 and τ_2 will be given on two separate rows. The number of digits of each of the two numbers does not exceed 63.

Output. For each test case, the program should output two rows. On the first one – the sum $\tau_1 + \tau_2$ and on the second one the difference $\tau_1 - \tau_2$.

Constraints. The numbers have no more than 63 digits.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача D. БУТИЛКИ

В съд има W литра вода. Разполагаме с N бутилки с дадени вместимости. Напишете програма, която намира колко най-много бутилки може да напълним догоре, така че съдът да бъде изпразнен изцяло.

Вход. На първия ред на стандартния вход е зададен броят на тестовете. Данните за всеки тест са зададени на два реда. На първия от тези редове са зададени вместимостта W на съда и броят N на бутилките. На следващия ред са записани N -те вместимости на бутилките в литри. Възможно е да има бутилки с еднаква вместимост. Всички числа във входа са цели и положителни и отделени едно от друго с интервали.

Изход. За всеки тестов пример, според реда им във входа, програмата трябва да изведе на нов ред на стандартния изход търсения максимален брой. Когато не е възможно да бъдат напълнени бутилки според условието на задачата, програмата трябва да изведе числото 0.

Ограничения: $0 < W < 100$, $0 < N < 300$, вместимостта на всяка от бутилките е по-малко от 300 литра.

ПРИМЕР

Вход	Изход
2	0
6 3	3
2 5 7	
12 5	
3 6 5 3 6	

There are W liters of water in a container. We have N bottles, each with a given capacity. Write a program that finds how many bottles we can fill completely, so that the container will be completely emptied.

Input. The first line of the standard input contains the number of test cases. The data for each test are written on two lines. The first of these lines contains the capacity W of the container and the number N of the bottles. The next line contains the N capacities of the bottles in liters. There may be bottles of the same capacity. All numbers in the input are positive integers and are separated each other by spaces.

Output. For each test case, according to their order in the input, the program must write on a new line of the standard output the found maximal number of bottles. When it is not possible to fill the bottles according to the task statement, the program must output the number 0.

Constraints: $0 < W < 100$, $0 < N < 300$, the capacity of each bottle is less than 300.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача Е. ИЗОБРАЖЕНИЕ

Дадено е многозначно изображение $E: N \rightarrow 2^N$, дефинирано в множеството N на естествените числа, със стойности – множества от естествени числа. За всяко $x \in N$, $E(x)$ е множеството от прости числа, които се получават, като се добави една цифра (по-голяма от 0) отляво на числото x . Например, $E(0)$ и $E(2)$ са празни множества, а $E(1) = \{11, 31, 41, 61, 71\}$. Ако X е множество от естествени числа $\{x_1, x_2, \dots, x_m\}$, тогава нека с $E(X)$ да означим множеството $E(x_1) \cup E(x_2) \cup \dots \cup E(x_m)$ – обединение на m -те множества. Сега можем да дефинираме $E^{(1)}(X) = E(X)$, $E^{(2)}(X) = E(E^{(1)}(X))$, а $E^{(k)}(X) = E(E^{(k-1)}(X))$, за $k = 3, 4, \dots$. Напишете програма, която по зададени x и k намира елементите на множеството $E^{(k)}(\{x\})$.

Вход. Програмата трябва да може да обработи няколко тестови примера. За всеки тестов пример, на един ред на стандартния вход, са зададени числата x и k .

Изход. За всеки тестов пример, на един ред на стандартния изход, програмата трябва да изведе елементите на търсеното множество, наредени по големина в нарастващ ред. Ако множеството е празно, програмата трябва да изведе 0.

Ограничения. $2 \leq x \leq 1000$, $2 \leq k \leq 12$

ПРИМЕР

Вход	Изход
13 2	1613 2113 3313 3613 5113 6113 9613
22 10	0

A multivalued function $E: N \rightarrow 2^N$ is given, defined in the set N of the natural numbers, with values – sets of natural numbers. For each $x \in N$, $E(x)$ is the set of prime numbers obtained by adding one digit (greater than 0) to the left of the number x . For example, $E(0)$ and $E(2)$ are empty sets, and $E(1) = \{11, 31, 41, 61, 71\}$. If X is a set of natural numbers $\{x_1, x_2, \dots, x_m\}$, then $E(X)$ denote the union of the m sets $E(x_1) \cup E(x_2) \cup \dots \cup E(x_m)$. Now we can define $E^{(1)}(X) = E(X)$, $E^{(2)}(X) = E(E^{(1)}(X))$, and $E^{(k)}(X) = E(E^{(k-1)}(X))$, for $k = 3, 4, \dots$. Write a program that, by given x and k , finds the elements of the set $E^{(k)}(\{x\})$.

Input. The program must be able to handle several test cases. For each test case, the numbers x and k are assigned to one line of the standard input.

Output. For each test case the program must display , on a separate line of the standard output, the elements of the searched set, ordered in ascending order. If the set is empty, the program should output 0.

Constraints. $2 \leq x \leq 1000$, $2 \leq k \leq 12$

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача F. ФЕРИБОТНИ ЛИНИИ

Архипелагът Островландия се състои от няколко острова. Всеки остров е свързан с поне една фериботна линия с някой от другите острови, като всички фериботни линии са двупосочни. Между някои два острова може да има и повече от една фериботна линия. Амбициозен владетел на Островландия решил да отвори още няколко фериботни линии така, че да може да се стигне от всеки остров до всеки друг остров – с един ферибот или с няколко ферибота, т.е. с прекачвания от един ферибот на друг в общо за двете линии пристанище. Напишете програма, която да намери минималният брой нови линии, които трябва да бъдат открити.

Вход. Програмата трябва да може да обработи по няколко тестови примера. Всеки тестов пример започва с ред на стандартния вход, на който е зададен броят N на фериботните линии. Следват N реда, на всеки от които са зададени, разделени с интервал, двата острова, които съответната линия свързва. Островите са означени с трибуквени низове от малки латински букви, в които буквата z не се използва. Различните острови са означени с различни низове. Краят на тестовите примери е означен с 0.

Изход. За всеки тестов пример, на ред на стандартния вход, програмата трябва да изведе броя на новите фериботни линии, които трябва да се открият, за да се изпълни желанието на владетеля.

Ограничения. $1 < N < 100000$.

ПРИМЕР

Вход	Изход
2 gen fen jar far 0	1

Archipelago Island consists of several islands. Each island is connected with at least one ferry line to one of the other islands, all ferry lines being bidirectional. There may be more than one ferry line between some two islands. The aspiring ruler of the Island has decided to open a few more ferry lines so that he can reach from any island to any other island, with one ferry or with several ferries, i.e. with transfers from one ferry to another at a port common to both lines. Write a program to find the minimum number of new lines to be opened.

Input. The program must be able to handle several test cases. Each test case begins with the line of the standard input that contains the number N of the ferry lines. On each of the following N lines is assigned, separated by an interval, a pair of two islands that the corresponding line connects. The islands are indicated by three-letter strings of small Latin letters, in which the letter z is not used. Different islands are marked with different strings. The end of the test examples is denoted by 0.

Output. For each test case, in the order of the standard input, the program must output the number of new ferry lines that need to be opened in order to fulfill the ruler's wish.

Constraints. $1 < N < 100000$.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача G. ПЪТИЩА

Разглеждаме точки с целочислени координати (x,y) в равнината. Някои от точките са свързани с хоризонтална или вертикална отсечка до някои от четирите им най-близки точки измежду дадените, намиращи се отгоре, отдолу, отляво или отдясно. Път, свързващ две точки наричаме последователност от описаните отсечки, чрез които от едната точка може да отидем до другата. Два пътя са различни, ако съответните им последователности от отсечки са различни. В един път може да има повторения на едни и същи отсечки. Дължина на пътя е броят на отсечките в последователността, определяща пътя. Напишете програма, която намира колко различни пътя с дадена дължина свързват две дадени точки.

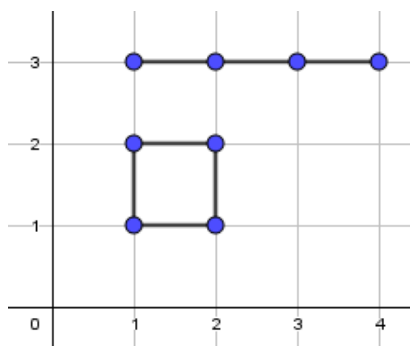
Вход. На първия ред на стандартния вход е зададен броят на тестовете. Данните за всеки тест започват с ред, съдържащ броя k на отсечките. Следват k реда, всеки описващ една отсечка: две двойки координати, задаващи двете крайни точка на отсечката. Следва ред, съдържащ дължината L на търсените пътища и броя Q на въпросите към програмата. Всеки въпрос е записан на отделен ред и се състои от координатите на две точки. Числата във входа са разделени с интервали. Дадените отсечки са различни. Различни са и двете крайни точки на всяка отсечка.

Изход. Програмата трябва да изведе за всеки тест на отделен ред на стандартния изход по едно цяло неотрицателно число, съответстващо на търсения брой (пресметнат по модул 100000007) пътища с дължина L по реда на въпросите.

Ограничения. $1 < k < 190$; $0 < L < 100$; $0 < Q < 7$. Координатите на точките във входа са цели положителни числа, по-малки от 15.

Пример:

Вход	Изход	Пояснение
2	3	Фигурата илюстрира първия тест във входа. Между точките $(1,3)$ и $(3,3)$ има три различни пътя с дължина 4 и те са: $(1,3)-(2,3)-(1,3)-(2,3)-(3,3)$; $(1,3)-(2,3)-(3,3)-(2,3)-(3,3)$; $(1,3)-(2,3)-(3,3)-(4,3)-(3,3)$.
7	8	
1 3 2 3	0	
2 3 3 3	3	
3 3 4 3		
1 2 2 2		
2 2 2 1		
2 1 1 1		
1 1 1 2		
4 2		
1 3 3 3		
2 2 1 1		
3		
1 1 1 2		
1 2 1 3		
1 3 1 4		
5 2		
1 1 1 3		
1 1 1 4		



We consider set of points with integer coordinates (x,y) in the plane. Some of the points are connected by a horizontal or vertical straight line segments to some others of their four closest points, located up,

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

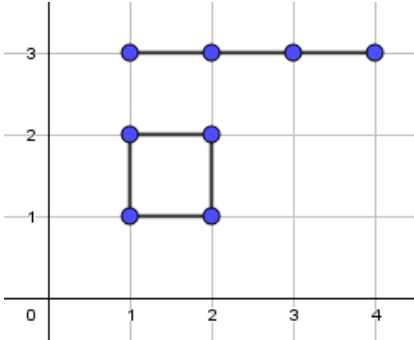
down, left or right. We call a path connecting two points a sequence of the given segments by which we can go from one of the point to the another. Two paths are different if their respective sequences of segments are different. There may be repetitions of same segments in the path. The length of the path is the number of segments in the sequence that defines the path. Write program that finds how many different paths of a given length connect two given points.

Input. The first line of the input contains the number of the test cases. The data for each test begins with a line of the standard input containing the number k of the segments, followed by k lines, each describing one segment: two pairs of coordinates, specifying the two endpoints of the segment. The following line contains the given length L and the number Q of questions. Each question is written in a separate line and consists of the coordinates of two points. The numbers in the input are separated by spaces. All the given segments are different. Both end points of each segment are also different.

Output. The program must output on separate lines of the standard output, one non-negative integer per line, equals to the found number of paths (reduced by modulo 100000007) with length L , corresponding to the order of the questions.

Constraints. $1 < k < 190$; $0 < L < 100$; $0 < Q < 7$. The coordinates of all points in the input are positive integers less than 15.

Example:

Input	Output	Explanation
2 7 1 3 2 3 2 3 3 3 3 3 4 3 1 2 2 2 2 2 2 1 2 1 1 1 1 1 1 2 4 2 1 3 3 3 2 2 1 1 3 1 1 1 2 1 2 1 3 1 3 1 4 5 2 1 1 1 3 1 1 1 4	3 8 0 3	<p>The figure illustrates the first test in the input. Between points (1,3) and (3,3) there are three different paths of length 4 and they are:</p> <p>(1,3)-(2,3)-(1,3)-(2,3)-(3,3); (1,3)-(2,3)-(3,3)-(2,3)-(3,3); (1,3)-(2,3)-(3,3)-(4,3)-(3,3).</p> 

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача Н. ЖАБАТА-ПРИНЦЕСА

Всеки ден Крис тича около езерото с N водни лилии. В езерото живее жабата Квак, която подскача по листата на лилиите. По спомени от приказките, Крис вярва, че ако целуне Квак, тя ще се превърне в красива принцеса. Но как да я хване?

След продължителни наблюдения Крис установява, че Квак може да подскача само в четири посоки, които той отбелязва с буквите A – североизток, B – югоизток, C – северозапад и D – югозапад. Всяка разходка е предварително планирана и се извършва по списък от K последователни посоки за подскок. Квак скача винаги до най-близкото листо в избраната посока. Ако няма листо в избраната посока, Квак остава на листото на което се намира, след което пробва подскок в следващата посока в списъка. След като Квак скочи, листото, от което е скочила потъва и изчезва. По време на разходката всички листа са неподвижни. Крис използва това обстоятелство и изготвя списък с целочислените координати на листата. Положителната посока на абсцисата на координатната система, която използва, сочи изток. Крис иска да определи координатите на листото, на което Квак ще се окаже накрая. Той ще я изчака там и ще я целуне. Напишете програма, която решава проблема на Крис и му помага да превърне Квак в красива принцеса.

Вход. От първия ред на стандартния вход се въвежда броя на тестовите случаи. Всеки тест започва с ред, на който са зададени N и K . На третия ред ще е зададен низ с K букви, всяка от които е A , B , C или D – редицата от посоките, в които Квак се опитва да скочи. Всеки от следващите N реда съдържа две цели числа X и Y – координатите на едно от растенията. Квак първоначално се намира на растението, чиито координати са посочени на първия от тези редове.

Изход. За всеки тестов случай програмата трябва да изведе на отделен ред на стандартния изход координатите на растението, до което ще стигне Квак за съответния тестов случай, разделени с един интервал.

Ограничения. $1 \leq N, K \leq 10^5, 0 \leq X \leq 10^9, 0 \leq Y \leq 10^9$.

ПРИМЕР

Вход	Изход
2	7 4
7 5	5 3
ACDBB	
5 6	
8 9	
4 13	
1 10	
7 4	
10 9	
3 7	
6 12	
AAAAABCCDD	
1 1	
2 2	
3 3	
4 4	
5 3	
6 2	

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Chris runs around the lake with N water lilies every day. A Qwak frog lives in the lake, bouncing on the leaves of the lilies. According to memoirs, Chris believes that if he kisses Quack, she will become a beautiful princess. But how do he get it? After long observations, Chris finds that Qwak can only jump in the four directions he marks with the letters A – northeast, B – southeast, C – northwest and D – southwest. Each walk of jumps is pre-planned in a list of K consecutive jump directions. Qwak always jumps to the nearest leaf in the current direction. If there is no leaf in the selected direction, she remains on the leaf on which it is located, then tries a jump in the next direction in the list. After Qwak has jumped, the leaf from which it has jumped dies and disappears. During the walk, all the leaves are motionless. Chris uses this circumstances and draws up a list of integer coordinates of the leaves. The positive direction of the abscissa of the coordinate system he uses points east. Chris wants to determine the coordinates of the leaf that Qwak will end up with. He will wait for her there and kiss her. Write a program that solves Chris's problem and helps him turn Qwak into a beautiful princess.

Input. On the first line of the standard input the number of test cases will be given. Each test starts with a line, in which N and K are given. The third row will contain a string of K letters, each of which is A, B, C, or D – the series of directions in which Qwak tries to jump. Each of the following N lines contains two integers X and Y – the coordinates of one of the plants. Qwak is initially found on the plant whose coordinates are indicated on the first of these rows.

Output. For each test case, the program must output, on a separate line of the standard output, the coordinates of the final plant for the respective test case, separated by one interval.

Constraints. $1 \leq N, K \leq 10^5, 0 \leq X \leq 10^9, 0 \leq Y \leq 10^9$.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача I. ТРОЙКИ

Дадени са целите положителни числа X , Y и Z . Напишете програма, която намира броя на тройките (x, y, z) , за които $x * z > y^2$, където $0 < x \leq X$, $0 < y \leq Y$ и $0 < z \leq Z$.

Вход: Програмата трябва да може да обработва няколко тестови примера при едно извикване. За всеки тестов пример на ред на стандартния вход ще бъдат зададени X , Y и Z , разделени с интервал.

Изход: За всеки тестов пример, на един ред на стандартния изход, програмата трябва да изведе броя на търсените тройки (x, y, z) , за които $x * z > y^2$.

Ограничения: $1 \leq X, Y, Z \leq 1000$.

ПРИМЕР

Вход	Изход
3 2 2	6
3 3 3	11

Given are positive integers X , Y , and Z . Write a program that finds the number of triples (x, y, z) for which $x * z > y^2$, where $0 < x \leq X$, $0 < y \leq Y$, and $0 < z \leq Z$.

Input: The program must be able to handle several test cases in a single call. For each test case, a row will be given on the standard input with X , Y and Z , separated by an interval.

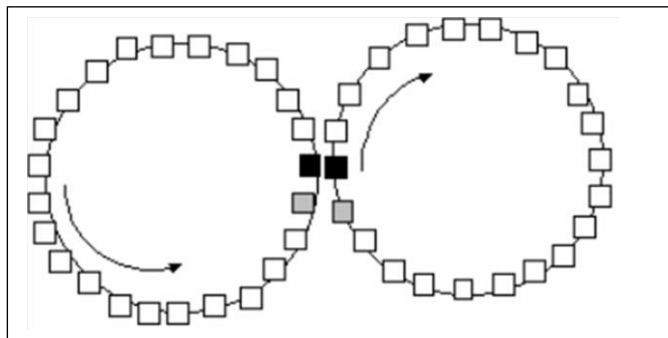
Output: For each test case the program must output on a line of the standard output the number of searched triples (x, y, z) for which $x * z > y^2$.

Constraints: $1 \leq X, Y, Z \leq 1000$.

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача J. ВИЕНСКИ КОЛЕЛА

Две виенски колела A и B имат a и b кабинки, съответно, и се въртят в противоположни посоки (виж Фигурата). Абсолютно невъзможно е по време на движение човек да прескочи от една кабинка в друга кабинка на колело в което се вози. Но двете колела са поставени едно до друго по такъв начин, че когато кабинка от едното колело се доближи максимално до кабинка от другото колело, човек може да прескочи от едната в другата.



Във всеки един момент точно една кабинка от колело A се сближава с точно една кабинка от колело B . Колелетата се въртят с такава скорост, че в следващия момент се сближават кабинките, съседни на кабинките, които са били близки в предишния момент. На Фигурата, двете кабинки, оцветени в черно са близко една до друга. В следващият момент, двете сиви кабинки ще се доближат една към друга. Ако някой се вози в колело A , може да изчака неговата кабинка да се доближи до кабинка от колело B и да скочи в нея. След това, може да изчака кабинката, в която се намира, да се доближи до кабинка от колело A и да се върне в него.

Задача 1: В началния момент, палав тийнейджър се намира в една от кабинките на колело A . Като се има предвид броя на кабинките на двете колела (a и b), да се намери броя на различните кабинки на колело A , които той може да посети с помощта на колело B .

Задача 2: Известен е броя на кабинките на колело A . Да се намери минималния брой кабинки на колело B , така че тийнейджърът, в която и кабинка на виенско колело A да се намира, да може да се качи във всяка една от останалите кабинки на колело A с помощта на колело B .

Напишете програма, която решава поставените задачи.

Вход: Първият ред на стандартния вход съдържа броя на тестовите примери. За всеки тестов пример на един ред са зададени броят a на кабинките на колело A и броят b на кабинките на колело B , разделени с интервал. Данната b се използва само за решаване на Задача 1.

Изход: За всеки тест, на един ред на стандартния изход програмата трябва да изведе две естествени числа, разделени с интервал. Първото трябва да е решение на Задача 1, а второто – на Задача 2.

Ограничения: $20 \leq a, b \leq 10^9$.

ПРИМЕР

Вход	Изход
1 24 22	12 23

Two ferris wheels A and B have a and b cabins, respectively, and rotate in opposite directions (see Figure). It is absolutely impossible during a movement for a person to jump from one cabin to another in a same wheel. But the two wheels are placed side by side in such a way that when a cabin from one wheel approaches as close as possible from a cabin from the other wheel, one can jump from first to the other. At any given moment, exactly one cabin from wheel A approaches exactly one cabin from wheel B . The wheels rotate at such a speed that, in the next moment, the cabins adjacent to the cabins that were close the previous moment converge. In the Figure, the two cabins painted black are close to each other. In the next moment, the two gray cabins will approach each other. If one rides in wheel A ,

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

he/she can wait for their cabin to approach a cabin from wheel B and jump into it. Then, it can wait for the cabin in which it is located, to approach a cabin of wheel A and return to it.
Task 1: At the start, a naughty teenager is in one of the cabins of wheel A . Given the number of cabins on both wheels (a and b), find the number of different cabins on wheel A that he can visit with the help of wheel B .

Task 2: Given the number of cabins of wheel A . Find the minimum number of cabins per wheel B so that the teenager, doesn't matter in which cabin of the wheel A is located, could visit each of the other cabins of A using B .

Write a program that solves the tasks.

Input: The first line of the standard input contains the number of test cases. For each test case, on one line is given the number a of cabins of A and the number b of cabins of B , separated by an interval. Data b is only used to solve Task 1.

Output: For each test, the program must output two natural numbers, separated by an interval, on one line of the standard output. The first must be a solution to Task 1 and the second to Task 2.

Constraints: $20 \leq a, b \leq 10^9$

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача К. ТУРНИР

Турнир по стрелба с лък се провежда съгласно следните правила. N мишени са подредени в редица и номерирани от 1 до N , включително, според мястото им в редицата (най-лявата мишена е номер 1, а най-дясната – номер N). В турнира участват $2N$ стрелци. Във всеки кръг на турнира има по двама стрелци на всяка мишена. Всички стрелци са класирани по умение с число от 1 до $2N$, като по-малък ранг съответства на по-доброто умение. Няма двама стрелци с еднакъв ранг. Всеки път, когато се състезават двама стрелци, печели този с по-малък ранг.

Всеки кръг на турнира протича по следната процедура: Двамата стрелци на всяка мишена се състезават помежду си и определят победител и губещ, а след това: победителите по мишени от 2 до N , включително, се преместват на мишената отляво (т.е. на мишени от 1 до $N - 1$ съответно; губещите по мишени от 2 до N , включително, както и победителят по мишена 1, остават на същата мишена; губещият по мишена 1 се премества към мишена N .

Турнирът продължава R кръга. Когато пристигнете за турнира, другите $2N - 1$ стрелци са наредени в редица и вие трябва да се вмъкнете някъде сред тях. След като заемете позицията си, двамата най-леви стрелци в редицата ще започнат турнира от мишена 1, следващите два – от мишена 2 и т.н., а двамата най-десни стрелци – от мишена N . Искате да се вмъкнете по такъв начин, че да завършите турнира по мишена с възможно най-малък номер. Ако има няколко начина да направите това, предпочитате този, който започвате от мишена с възможно най-голям номер. Напишете програма, да ви помогне да изберете началната си позиция.

Вход. Първият ред на стандартния вход съдържа броя на тестовете. Всеки тест започва с ред, който съдържа целите числа N и R , разделени с интервал. Следващият ред съдържа ранговете на стрелците. Първи в списъка е вашият ранг. Следват ранговете на останалите стрелци, в реда, по който са подредени в началото (отляво надясно).

Изход. За всеки тестов случай програмата трябва да изведе на стандартния изход ред, съдържащ цяло число между 1 и N , включително, което е номерът на мишената, на която трябва да започнете турнира.

Ограничения. $1 \leq N \leq 200\,000$ (броят на мишените N е половината от броя на стрелците), $2N \leq R \leq 1\,000\,000\,000$, $1 \leq S_k \leq 2N$ (S_k е рангът на k -тия стрелец).

ПРИМЕР

Вход	Изход
1 4 8 7 4 2 6 5 8 1 3	3

An archery tournament is held according to the following rules. There are targets arranged in a line and numbered from 1 to N inclusive according to their place on the line (the leftmost target being target 1,

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

and the rightmost target being target N). There are also $2N$ archers. At any round during the tournament, there are two archers on each target. All archers are ranked by skill with integers from 1 to $2N$, where a smaller rank corresponds to better skill. No two archers have the same rank. Whenever two archers compete, the one with the smaller rank will always win.

Every round of the tournament goes according to the following procedure: The two archers on each target compete with each other and determine a winner and a loser between them and then: the winners on targets 2 to N inclusive move to the target on their left (i.e., targets 1 to $N - 1$ respectively); the losers on targets 2 to N inclusive, as well as the winner on target 1, remain on the same target; the loser on target 1 moves to target N .

The tournament continues for R rounds. When you arrive for the tournament, other archers are standing in a line, and you have to insert yourself somewhere into the line amongst them. After you take your position, the two leftmost archers in the line will start the tournament on target 1, the next two will start on target 2 and so on, with the two rightmost archers starting on target N . You want to insert yourself in such a way as to ensure that you will finish the tournament on a target with as small a number as possible. If there are multiple ways to do this, you prefer the one that starts at a target with as large a number as possible. Write a program to choose your initial position.

Input. The first line of the standard input contains the number of the test cases. Each test case starts with line that contains the integers N and R , separated by a space. The next line contains the ranks of the archers. The first in the list is your rank. The rest are the ranks of the other archers, in the order in which they have arranged in the beginning (from left to right).

Output. For each test case the program must write to standard output a single line containing a single integer between 1 and N inclusive – the number of the target on which you will start the tournament.

Constraints. $1 \leq N \leq 200\,000$ (the number of targets N is equal to half the number of archers), $2N \leq R \leq 1\,000\,000\,000$, $1 \leq S_k \leq 2N$ (S_k is the rank of the k -th archer).

XXXII РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
АУБГ, Благоевград, 09.05.2020

Задача L. ПОВТОРЕН НИЗ

Даден е низ T с дължина N , съставен от големи латински букви. Напишете програма която да намира в него най-дългия подниз, който се среща поне два пъти.

Вход. Първият ред на стандартния вход съдържа броя на тестовите, които програмата трябва да обработи. За всеки тест на един ред ще бъде зададен низът T .

Изход. За всеки тестов случай програмата трябва да изведе, на отделен ред на стандартния изход, най-дългия намерен подниз. Ако съществува повече от един подниз, който се среща поне два пъти в T , програмата трябва да изведе най-малкия лексикографски такъв низ.

Ограничения. $9 \leq N \leq 100000$.

ПРИМЕР

Вход	Изход
1 АВААААВВАВ	ААА

A string T of length N is given, consisting of uppercase Latin letters. Write a program that finds in T the longest substrings that occur at least twice.

Input. The first line of standard input contains the number of tests cases that the program must complete. Each test starts consists of a line with string T .

Output. For each test case, the program should output, on separate line of the standard output, the longest repeated substring in T . If there is more than one string that occurs at least twice, the program must output the smallest lexicographically string.

Constraints. $9 \leq N \leq 100000$.