Problem A. The Motorway

Input file:	`standard input`
Output file:	`standard output`
Time limit:	9 seconds
Memory limit:	128 mebibytes

Autobyte Company is involved in the construction of one of Byteland motorways. Until recently the company collected toll charges only at the starting point of the motorway. However, Byteasar, the new chairman in charge of the company, noticed that in such a case the charged amount does not depend on the distance covered by customers in bytemiles. Therefore, the company plans to build toll-collecting points along the entire length of the motorway.

Byteasar, during his motorway trip, with the help of the odometer in his car, put down the location of all the n entry points (the position of an entry point is its distance from the start of the motorway). The company decided to locate $n+1$ toll-collecting points evenly along the motorway. That means the distance between each two subsequent toll-collecting points would be the same. At the same time between each two such points there should be a motorway entry point and there should be toll-collecting point between each two subsequent motorway entries. Luckily, it turned out that the existing location of entry points makes possible such an arrangement.

Your task would be to calculate the minimum and maximum distance between toll-collecting points. Formally speaking, we are seeking the lowest and highest value for l, for which there exists a position b_0 of first toll-collecting point, such that the consecutive points should be located in $b_0+l, b_0+2l, \ldots, b_0+nl$ positions. It may be so that the location of a given toll-collecting point, determined by the above procedure, falls in exactly the same position, as the location of an entry point. In this case the toll booth would be positioned in close vicinity of an entry point, either just before or just after it. In other words, the position of the j-th entry point should be included in the following interval $[b_0+(j-1)l, b_0+jl]$.

Input

The first line of input contains one integer n ($3 \leqslant n \leqslant 1\,000\,000$): the number of motorway entry points. The second line of input contains an increasing sequence of n integers $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$). The following sequence elements are the positions of subsequent motorway entry points.

Output

Your program should produce two real numbers presenting smallest and largest possible distance between two subsequent toll collecting points in bytemiles. You can assume that the difference between these values is not less than 10^{-9}.

Your result will be considered as being correct in case it is included in the interval $[x(1-\epsilon)-\epsilon, x(1+\epsilon)+\epsilon]$, where x is the correct answer, $\epsilon = 10^{-8}$. Therefore both relative error and absolute error of the answer equal to ϵ will be accepted.

Examples

standard input	standard output
6 2 3 4 5 6 7	0.833333333333 1.250000000000

Problem B. Bytehattan (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	128 mebibytes

Bytehattan is one of the islands in the capital of Byteland. The island is a witness to parades, outings and processions organized frequently. In fact these are so persistent, that street closing and serious traffic congestion result. Byteasar, employed at the town hall, has been appointed to monitor the city traffic.

The streets of Bytehattan form a regular $n \times n$ grid. Let us look at the map of Bytehattan as if it were grid coordinates: for each pair of integers x, y, such that $1 \leqslant x, y \leqslant n$, at point defined by its coordinates (x, y) there is an intersection. Each two street crossings 1 unit away are joined by a street measuring 1 unit.

Byteasar receives messages concerning street closures. Each message informs that a particular street will be closed from now on. After receiving such information concerning the closure of a given street, Byteasar should determine whether it would be still possible to commute between two intersections which are located at the ends of such a closed street, using roads which have not yet been closed. Help him and create a program helping him with his job.

Input

The first line of input contains two integers n and k ($2 \leqslant n \leqslant 1500$, $1 \leqslant k \leqslant 2n(n-1)$): the number of streets in Bytehattan and the number of messages concerning closed streets. Each of the following k lines contains information concerning the closure of one of the streets, information is provided in chronological order. Each of these lines consists of *two* streets described, one after the other. In practice exactly one of the streets becomes closed[1]. In case it is still possible to commute between two intersections which are located at the ends of such a closed street, described in the *previous line*, the first of these streets becomes closed. In case it is not possible, the second one is closed. The first closure, out of those k closures described in the input, applies to the first one out of the two streets listed. Each street can only be closed once.

The description of a given street is an integer pair a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) followed by a letter c_i ($c_i \in \{\mathtt{N}, \mathtt{E}\}$). Such a triple determines a street, with one of its ends is positioned at an intersection described by coordinates: (a_i, b_i). In case $c_i = \mathtt{N}$, the other end of the street is positioned at an intersection described by coordinates: $(a_i, b_i + 1)$. In case $c_i = \mathtt{E}$, the other end of the street is positioned at an intersection described by coordinates: $(a_i + 1, b_i)$. In case $c_i = \mathtt{N}$, then $b_i < n$, similarly if $c_i = \mathtt{E}$, then $a_i < n$.

Output

Exactly k lines should be contained in the output. In case after i-th street closure it is still possible to commute between the intersections on a closed street from input, in the i-th line of output there should be the word "`TAK`" (Polish for *yes*). Otherwise the i-th line should contain word "`NIE`" (Polish for *no*).

Examples

standard input	standard output

`3 4`
`2 1 E 1 2 N`
`2 1 N 1 1 N`
`3 1 N 2 1 N`
`2 2 N 1 1 N`
	`TAK`
	`TAK`
	`NIE`
	`NIE`

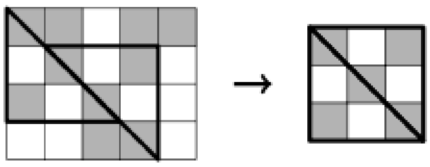
[1]The intention of the jury is that such an atypical input format would create the need for processing each street closure before starting the processing of subsequent street closures.

Problem C. The Carpenter (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 mebibytes

Byteasar would fancy a game of checkers, however his chessboard got lost somewhere. He only managed to find a wooden board, sized $n \times m$, divided into nm equal in size, square fields. Each field is painted either white or black, however the arrangement of the colours on this board not necessarily matches the proper chessboard pattern. In such a case Byteasar has decided to utilise his carpentry experience and with the help of a saw he plans to cut out a chessboard, which is a square consisting of certain number of fields, where two fields sharing sides have alternate colours.

It is not clear whether Byteasar manages to find out a properly sized square on the board. So, he decided to cut out two triangular pieces from the board in order to glue them together in such a way that a chessboard would be created. (The pieces must be separable, however they may be turned around in any way after cutting out). Help Byteasar and calculate the largest chessboard size that he is able to obtain by using this method. The figure below presents the board sized 4×5 and the two triangles, which could be glued together in order to form chessboard sized 3×3:



Input

The first input line contains two integers n and m ($1 \leqslant n, m \leqslant 1000$): the board size. The following n lines contains m integers each: j-th number from i-th line ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$) describes the colour positioned on the intersection of j-th column and i-th row of the board. Digit 0 describes white field and digit 1 — black field.

Output

The first and only output line should contain one integer, representing the largest chessboard size, which is obtainable by cutting two triangular pieces from the board and pasting them together.

Examples

standard input	standard output
4 5 1 1 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0	3
3 3 1 1 1 1 1 0 0 1 0	2

Problem D. Demonstrations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	128 mebibytes

This Sunday, The Byte Day will be celebrated in Bytetown, one of the most important annual Bytelandian celebrations. However, everything indicates that this year jubilation will not only be a rural family fete.

Well, Bytetown citizens are strongly divided concerning one crucial matter. Some believe that in line with tradition, the byte should always be equal to eight bits. However there are progress supporters who would rather go for more capacious, 16 bit bytes. Others see the whole matter much more rigidly and would eagerly like to declare that the byte should always have only four bits. Finally there are less significant subversive movements in Bytetown, whose members advocate that the count of bits in the byte should not be the power of two, or yet it must not necessarily be an even number! All of these societies plan to hold their own manifestation in order to convince Bytetown citizens to their cases.

Many Bytetown citizens are afraid that such a number of demonstrations might interfere with the The Byte Day celebrations. The Lord Major of Bytetown sensed a significant public support could be gained, by forbidding some of the demonstrations. Due to the fact that such decisions raise controversy, the Lord Mayor decided he would only cancel two demonstrations. Additionally he would like to be able to choose such demonstrations for cancellation, that would result in the the total time taken by any other possible demonstrations taking place in the city after the cancellation, to possibly be shortest. Help The Lord Mayor and give him a clue how much time in the city without a demonstration he can achieve.

Input

The first input line contains one integer n ($2 \leqslant n \leqslant 500\,000$): the number of planned demonstrations. Each of the subsequent n lines describes one demonstration: i-th of those lines contains two integers a_i and b_i ($0 \leqslant a_i < b_i \leqslant 10^9$), which mean that i-th demonstration begins a_i byteminutes after sunrise and ends b_i byteminutes after sunrise.

Output

Your program should produce exactly one non-negative integer, describing by how much time demonstrations taking place could possibly be shortened, in case The Lord Mayor of Bytetown cancels maximum two demonstrations.

Examples

standard input	standard output
5 0 9 1 4 2 5 7 9 6 7	4

Problem E. The Exam

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 mebibytes

Professor Byteoni is preparing *Bit & Byte Theory* exam. He has already prepared n questions. Each of these questions has been ranked with an expected difficulty coefficient by the professor. This coefficient is a natural number ranging from 1 to n. Each of the questions holds a different coefficient.

Now the professor is considering the exam questions sequence. Professor wishes to determine whether his students are able to judge the question difficulty by themselves. For this purpose he plans to line up his questions in such a way, that coefficients of subsequent questions differ at least by k. Help the professor to find such a sequence.

Input

The first and only input line contains two integers n and k ($2 \leqslant n \leqslant 1\,000\,000$, $1 \leqslant k \leqslant n$): the number of questions prepared by professor and the lower limit of the difficulty difference of subsequent exam questions.

Output

Your program should output one line containing sought question difficulty coefficients sequence, in other words a sequence of n pairwise distinct natural numbers ranging from 1 to n, where each two subsequent numbers differ at least by k. If there are numerous correct answers, your program should write any one of these. In case the sought sequence does not exist, your program should write only one word: "`NIE`" (Polish for *no*).

Examples

standard input	standard output
`5 2`	`1 4 2 5 3`
`5 4`	`NIE`

Problem F. Speed Cameras

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 mebibytes

The Lord Mayor of Bytetown plans to locate a number of radar speed cameras in the city. There are n intersections in Bytetown numbered from 1 to n, and $n-1$ two way street segments. Each of these street segments stretches between two intersections. The street network allows getting from each intersection to any other.

The speed cameras are to be located at the intersections (maximum one per intersection), wherein The Lord Mayor wants to maximise the number of speed cameras. However, in order not to aggravate Byteland motorists too much, he decided that on every route running across Bytetown roads that does not pass through any intersection twice there can be maximum k speed cameras (including those on endpoints of the route). Your task is to write a program which will determine where the speed cameras should be located.

Input

The first line of input contains two integers n and k ($1 \leqslant n, k \leqslant 1\,000\,000$): the number of intersections in Bytetown and maximum number of speed cameras which can be set up on an individual route. The lines that follow describe Bytetown street network: the i-th line contains two integers a_i and b_i ($1 \leqslant a_i, b_i \leqslant n$), meaning that there is a two-way street segment which joins two intersections numbered a_i and b_i.

Output

The first output line should produce m: the number describing the maximum number of speed cameras, that can be set up in Byteland. The second line should produce a sequence of m numbers describing the intersections where the speed cameras should be constructed. Should there be many solutions, your program may output any one of them.

Examples

standard input	standard output
5 2 1 3 2 3 3 4 4 5	3 1 2 5

Problem G. Game (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 mebibytes

Byteted and Bited decided to play marbles. There is an ever number of marbles in the urn. Each marble has been marked by exactly one digit. The rules of the game are very simple: the players take on random one marble each from the urn in turns. The game ends when the urn is empty. The player who has accumulated a set of marbles with a larger product of digits, wins.

The boys very much got to like this game. They are both very ambitious and they really like to win, so a draw makes nobody happy. Byteted and Bited are determined to avoid such an ending situation at all costs. Write a program which will check if for a given initial set of marbles in the urn, the game can end up drawn.

Input

The first line of input contains one integer t ($1 \leqslant t \leqslant 1000$), indicating the number of test cases to be considered.

Each of the following t lines contains ten non-negative integers $k_0, \ldots, k_9$ ($0 \leqslant k_i \leqslant 10^{15}$), where k_i indicates the number of marbles marked with i digit. The sum of the numbers k_i is even and positive in each test case.

Output

Your program should produce t lines containing answers to respective test cases. The result for each test case that can end with a draw is word "`TAK`" (Polish for *yes*). In the opposite case the result should be "`NIE`" (Polish for *no*).

Examples

standard input
5
0 1 0 1 1 4 1 0 5 1
0 1 1 0 3 0 0 0 0 3
1 1 0 4 0 0 2 0 0 2
1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000
0 999999 999999 1000000 1000000 1000000 1000000 1000000 1000000 1000000

standard output
TAK
NIE
NIE
TAK
NIE

Problem H. The Hero (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 mebibytes

Byteotheus, most famous Byteotian hero, once again emerged victorious from the battle. While his crew are loading the ship up with the acquired valuables, in his cabin, Byteotheus plans his way back to his homeland island — The Bitaca. It is not an easy task. Many gods envy Byteotheus popularity among the people and gladly would take him down a peg or two.

Fortunately, some of them look favourably on him, especially goddess Bythena. It was none other but her that sent Byteotheus a dream last night, warning him of the dangers that he could encounter.

There are n islands on the Byteonian Sea. It will be convenient to number those from 1 to n. Presently Byteotheus's ship is at island 1, and its destination is The Bitaca — island n. In some cases two islands are joined by *one-way* sea routes, additionally each of those islands is a start point for maximum of 10 sea routes. We are numbering the sea routes from 1 to m; i-th route leads from island a_i, to island b_i, and it takes exactly d_i days to cover it. In case the ship set sail on i-th route, starting from island a_i at dawn on j day, it will reach its destination island b_i at dawn, at day $j + d_i$. The ship can stop at any island for an indefinite period before moving on again. However, before reaching a successive island, it cannot deviate off the set path, and sail no longer that is required to cover the particular route. Byteotheus can start his voyage from island 1 at dawn on the first day, at the earliest.

The goddess Bythena warning has been very precise. She provided Byteotheus an exact list of p traps, prepared by the gods. Every trap is situated on a certain island and is active for a certain time period. To be more precise, the i-th trap is located on the w_i island and is active from the day s_i until and including the day k_i. The traps are really dangerous — in case Byteotheus's ship finds itself on an island with an active trap, no one will survive. Luckily his homeland Bitaca is free from traps, and no traps on the island 1 are active on the first day.

Obviously Byteotheus wants to plan his way home, to avoid all traps. He wonders, however, how much longer he would need for his voyage because of them. Help him and indicate the minimum number of days necessary to safely return to Bitaca.

Input

The first line of input contains two integers n and m ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 1\,000\,000$): the number of islands and the number of sea routes. Subsequent m lines describe the sea routes: included in the i-th there are three integers a_i, b_i, d_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$, $1 \leqslant d_i \leqslant 10^9$), indicating that the i-th route leads from island a_i to island b_i and it takes d_i days. All routes are one way. Every island is a start point for maximum of 10 sea routes.

The next line contains integer p ($0 \leqslant p \leqslant 100\,000$), describing the number of the traps. Next p lines hold the description of the traps: in the i-th line there are three integers w_i, s_i, k_i ($1 \leqslant w_i < n$, $1 \leqslant s_i \leqslant k_i \leqslant 10^9$), indicating that the i-th trap is located on the island w_i and is active from the day s_i until and including the day k_i. If $w_i = 1$, then $s_i > 1$.

Output

In case it is not possible to plan the route avoiding all the traps, the one and only line should output word "`NIE`" (Polish for *no*). In the opposite case, an integer d should be output describing the minimum number of days required to finalise the voyage (the ship reaches Bitaca on the day $d + 1$ at sunrise).

Examples

standard input	standard output
5 6 1 2 3 1 4 13 2 3 1 2 4 2 3 2 2 4 5 1 5 1 2 4 1 8 8 2 6 7 2 10 11 4 6 7	10

Note

Byteotheus set sail from island 1 on the first day, at sunrise. He arrives on island 2 on the fourth day. There he waits one day and starts off for island 3. After getting there on the sixth day, he immediately turns back to island 2, where from he travels in the direction of island 4 on the eighth day. He arrives there on the tenth day and finally reaches Bitaca on the eleventh day.

Problem I. Genetic Engineering

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 mebibytes

Byteotian paleoarchaeologists recently unearthed a few ambers, which had trapped ancient mosquitoes inside. After analysing the samples of insects it turned out that they come from the Jurassic period, and therefore likely to have been in contact with large reptiles that dominated the Byteotian lands. This gave geneticists a quaint idea: to try to recover byteoraptor genetic material from the blood of mosquitoes.

Byteoraptor genome, as in all Bytean organisms, is a chain consisting of a number of byteo-aminoacids. For simplicity we denote the types of byteo-aminoacids by natural numbers. Redundancy occurs in a genome — every type of byteo-aminoacid is repeated k times (specifically, the length of each correct genome is a multiple of k). In other words, if we divide the genome into blocks consisting of subsequent k byteo-aminoacids, each block will contain byteo-aminoacids of the same kind.

Geneticists were able to isolate a suspected chain consisting of byteo-aminoacids, from the blood of a mosquito, being n in length. Unfortunately, the chain may not be a valid genome — scientists suspect that it may have been contaminated by foreign byteo-aminoacids. Presently they want to test their hypothesis and remove the least byteo-aminoacids from that chain, such that a normal genome emerges. In case of many equally good possibilities, the researchers are interested in the genome that is the earliest in lexicographical order [2]. Your task is to help them to make a breakthrough discovery.

Input

The first line contains two integers n and k ($1 \leqslant n \leqslant 1\,000\,000$, $2 \leqslant k \leqslant 1\,000\,000$): the length of extracted chain of byteo-aminoacids and redundancy degree of a correct genome. The second line contains a sequence n of integers $g_1, \ldots, g_n$ ($1 \leqslant g_i \leqslant 1\,000\,000$): the types of subsequent byteo-aminoacids in the chain.

Output

The output should contain two lines. The first one should contain the number m ($0 \leqslant m \leqslant n$) denoting length of the longest proper genome, which may arise by removing some byteo-aminoacids from the specified chain.

The second line should contain a chain of m numbers describing the types of subsequent byteo-aminoacids in the correct genome. In case there are multiple solutions, your program should output the smallest lexicographically. If $m = 0$ (i.e. geneticists have failed to isolate any non-empty correct genome), the second line of output should be empty.

Examples

standard input	standard output
16 3 3 2 3 1 3 1 1 2 4 2 1 1 2 2 2 2	9 1 1 1 2 2 2 2 2 2

[2] Let l_1 and l_2 be two different chains of the same length, consisting of byteo-aminoacids. To determine which one is earlier in lexicographical order it is necessary to find the first position where the chains differ. The chain earlier in the lexicographical order is the one which has byteo-aminoacid marked with a lower number in this position.

Problem J. Robin Hood

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	128 mebibytes

Robin Hood takes the rich to distribute to the poor. Together with his gang they robbed a convoy carrying gold to the counts' castle and n caskets fell prey to robbers. After transporting their loot to the cave it turned out that i-th (for $i = 1, 2, \ldots, n$) casket contains exactly i money-bags full of gold.

In case a poor man comes to Robin Hood asking for a few gold ducats, Robin Hood utilises the following procedure. Firstly he chooses a non-empty casket that contains the smallest number of money-bags containing gold. In case the casket contains exactly only one money-bag, Robin Hood hands it to the man in need, and sees him go away happily. Otherwise, if the casket contains an odd number of money-bags, Robin Hood puts one of the money-bags in his pocket, and starts the whole process again. However, in case there is an even number of money-bags, Robin Hood takes exactly half of them out and puts them in an empty casket (luckily the empty caskets are plentiful in the cave) and begins the whole procedure anew. Therefore if a penniless man comes to Robin Hood, and in case he still will be in a possession of at least one non-empty casket, as a result of (possibly multiple) employment of Robin Hood's procedure, the poor man is sure to get the money-bag full of gold. The poor would come to the Robin Hood's cave until all the caskets are empty.

Fellow robbers from Robin Hood's gang wonder if their leader does not ruin the good name of thugs with his behaviour. They want to know how many looted money-bags remain in Robin Hood's pocket when all the caskets are empty.

Input

The first and only line of the input contains one integer n ($1 \leqslant n \leqslant 10^9$), which indicates the number of caskets robbed by Robin Hood's gang.

Output

The first and only line of output should contain an integer representing the number of money-bags with gold, which will remain in Robin Hood's pocket after emptying all the caskets.

Examples

standard input	standard output
5	2

Problem K. Blanket

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	128 mebibytes

This summer, Byteburg citizens are turning out in droves at the city beach down by the Byteotian Lake to experience the joy of sunbathing. Every Byteburg citizen arrives at the beach equipped with the blanket manufactured by *Byteasar & Son*, the trendiest this season. All blankets are of equal size $a \times b$ (although different patterns), and each sunbather sets out his blanket in such a way that her blanket longer side is always perpendicular to the lake.

One of this year's sunbathers is professor Byteoni. After a few days of sunbathing professor noticed, that all the people who come to the beach always set out their blankets in their own favourite individual places, always the same. Although people come to the beach and leave it at different times, the professor never heard that any sunbather had taken over somebody's else's favourite place by putting the blanket there. This observation made the professor so curious, he decided to study this phenomenon.

For that purpose he set a coordinate system on the beach, and for every of the n Byteburgians noted down the coordinates of each of the spots where individual citizens always put their blankets. The system is devised in such a way that the OX-axis is parallel to a sides, and the OY-axis to b sides of all of the blankets. The professor initially wanted to calculate the area of intersection of the areas occupied by the blankets for each pair of them. But then he realized that it is enough for further research that he has only the average of these values. In other words, he is interested in the expected value of the area of intersection of the fields occupied by blankets belonging to two different random people of Byteburg. Using the data provided by the professor, help him do the calculation.

Input

The first line of input contains three integers n, a and b ($2 \leqslant n \leqslant 200\,000$, $1 \leqslant a, b \leqslant 1\,000\,000$) indicating respectively the number of Byteburg inhabitants and the sizes of the blankets. Each of the subsequent n lines contains two integers x_i and y_i ($0 \leqslant x_i, y_i \leqslant 1\,000\,000$) indicating the coordinates of the point where i-th Byteburg citizen always puts the lower left corner of his blanket.

Output

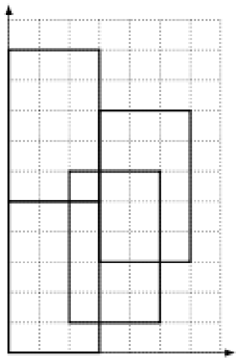
Your program should print one real number that represents the average area of intersection of the areas occupied by the blankets belonging to pairs of Byteburg inhabitants. Your result will be deemed valid if it is in the following range $[x - \epsilon, x + \epsilon]$, where x is the correct answer, and $\epsilon = 10^{-2}$.

Examples

standard input	standard output
4 3 5 0 0 2 1 3 3 0 5	1.833333333

Note

The exact result is: $\frac{4+0+0+1+6+0}{6} = 1\frac{5}{6}$.



Problem L. Wipe! (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

John is the computer expert at his company and has now been tasked to find some software for erasing data properly. It is very important that the data should not be recoverable afterwards, so it should be overwritten on the hard drive several times. Unable to find any free program up to the task, John decides to write such a program himself. The user interface is simple, it only asks for the file to be destroyed and n, the number of times it should be overwritten. This number can range from 1 (quick deletion) to 20 (maximum security). John processes the file bit by bit and does not consider writing a zero where there was already a zero as really overwriting. So for each of the n sweeps, he overwrites each zero with a one and each one with a zero.

John knows that independent testing is important, so he has asked you to write the verification routine. He will not listen to your objections to the algorithm so eventually you give in.

Input

The first line of the input contains a single integer $1 \leqslant N \leqslant 20$. The two following lines each contain a string containing only the characters '`0`' and '`1`'. The first of these lines represent the bits of the file before deletion and the second the bits on the same position on the hard drive after the file has been deleted. The length of these strings are the same and between 1 and 1 000 characters.

Output

Output a single line containing either the words "`Deletion succeeded`" if each bit is switched N times or "`Deletion failed`" if this is not the case.

Examples

standard input	standard output

`1`
`10001110101000001111010100001110`
`01110001010111110000101011110001`
	`Deletion succeeded`

`20`
`0001100011001010`
`0001000011000100`
	`Deletion failed`

Problem M. Timebomb (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

You and your teammates from the anti-bomb squad of the local police have been called to defuse a bomb found in the only pub in town. Fearing the tragic consequences this might produce, you go to the scene as quickly as possible. After some research, you learn that the bad guys have created a tricky way to allow them to defuse the bomb at will. You find a remote control with a button that you can take to a safe place. You also find that it is possible to connect to the bomb through a wireless connection and retrieve an ASCII representation of a code every 2 seconds. The bomb then gets defused if the button is pressed when the code is a number divisible by 6. But you have to be careful. If you press the button when the ASCII representation of the code is not a number divisible by 6 or has an invalid representation for any digit, the bomb will explode instead. You have to rely on your programming skills to write a program able to tell you if it is safe to press the button, before it blows out the pub (and the beer).

Input

The input consists of an ASCII representation of a code. This code has between 2 and 8 digits. Each digit is represented by 5 rows and 3 columns of characters, which can be either a space or a star character '`*`'. No other type of character (except for the new line character) will ever appear in the input. There is also one column of spaces (and only spaces) to separate each digit. After the last digit you will find a column of new line characters. Note that although every digit will always be of size 5×3, there is no guarantee it will represent a valid digit between 0 and 9 inclusive. The valid 5×3 representations for each digit are given below.

```
### ### ### ### ### ### ### ### ### ###
***   * *** *** * * *** *** *** *** ***
* *   *   *   * * * *   *       * * * * *
* *   * *** *** *** *** ***   * *** ***
* *   * *     *   *   * * *   * * *   *
***   * *** ***   * *** ***   * *** ***
```

The hash '`#`' characters on the top are there only to mark the 3 columns used for a digit and are not part of the digits' representation.

The code can have leading zeros, hence an ASCII representation of, for example, 00000076 represents the number 76. You may also safely assume that every valid code will correspond to a strictly positive number.

Output

Print one line with "`BEER!!`" if it is safe to press the button and defuse the bomb, and "`BOOM!!`" otherwise.

Examples

standard input:

```
***   * * * *** *** *** ***
* *   * * *   *   *   * *
* *   * *** *** *** *** ***
* *   *   * *     * *   * *
***   *   * *** *** *** ***
```

standard output:

```
BEER!!
```

standard input:

```
  *   * *** *** *** * *
  *   * **    * * * * *
  *   * *** *** *** ***
  *   * *   *   * *   *
  *   * *** *** ***   *
```

standard output:

```
BOOM!!
```

standard input:

```
*** ***   * *** ***   *
*   * *   * * *   *   *
*** * *   * *** ***   *
  * * *   *   * *     *
*** ***   * *** ***   *
```

standard output:

```
BOOM!!
```

standard input:

```
*** *** *** * * ***
  *   *   * * * * *
***   * *** *** ***
*     *   *   * * *
***   * ***   * ***
```

standard output:

```
BEER!!
```

Problem N. Planting Trees (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

Farmer John has recently bought n tree seedlings that he wants to plant in his yard. It takes 1 day for John to plant a seedling (John isn't particularly hardworking), and for each tree John knows exactly in how many days after planting it grows to full maturity. John would also like to throw a party for his farmer friends, but in order to impress them he would like to organize the party only after all the trees have grown. More precisely, the party can be organized at earliest on the next day after the last tree has grown up.

Help John to find out when is the earliest day when the party can take place. John can choose the order of planting the trees as he likes, so he wants to plant the trees in such a way that the party will be as soon as possible.

Input

The input consists of two lines. The first line contains a single integer N ($1 \leqslant N \leqslant 100\,000$) denoting the number of seedlings. Then a line with N integers t_i follows ($1 \leqslant t_i \leqslant 1\,000\,000$), where t_i denotes the number of days it takes for the ith tree to grow.

Output

You program should output exactly one line containing one integer, denoting the earliest day when the party can be organized. The days are numbered $1, 2, 3, \ldots$ beginning from the current moment.

Examples

standard input	standard output
4 2 3 4 3	7
6 39 38 9 35 39 20	42

Problem O. Number Trick (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

Jakub is to hold a presentation on useful mathematical tricks. E.g., to take the square root of a number you just need to remove the first half of the number. To convince his audience he uses the well tested method of proof by example: $\sqrt{25} = 5$ and $\sqrt{5776} = 76$ so the method obviously works. To multiply a number by $X = 2.6$ all you have to do is move the first digit to the end of the number, $135 \times 2.6 = 351$ and $270270 \times 2.6 = 702702$.

Jakub wants to demonstrate that this last method works for any X. To do this he will ask his audience for values of X and then show them example multiplications for which the method works. Jakub has noticed that he can not just pick arbitrary numbers for his examples, so now he wants your help. Can you write a program that given X gives a list of integers for which multiplying by X is equivalent to moving the first digit to the end of the number? Jakub does not like very large numbers so do not list any numbers with more than 8 digits.

Input

The input is a single decimal number X ($1 \leqslant X < 1000$) with at most 4 digits after the decimal point.

Output

Output a list of all positive integers less than 10^8 for which Jakub' second trick works. Write the numbers in ascending order, one number per line. If the list is empty, output instead "`No solution`".

Examples

standard input	standard output

`2.6`	`135`
	`270`
	`135135`
	`270270`

`3.1416`	`No solution`