Problem A. Three Lighthouses

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

The «Victory» ship sailed away from the shore to participate in an international regatta. Having no positioning device onboard the ship went off the course and got lost in the seas. Luckily, the captain could see three lighthouses through his binoculars with their coordinates known. As everyone knows, binoculars make it is possible not only to observe the objects, but also measure their angular sizes, i.e angles between straight lines connecting objects' corner points with an observing eye. For example, if two lighthouses are located at points M_1 and M_2, and the ship is at point K, then the angular size of segment M_1M_2 is an angle M_1KM_2.

You need to find the ship's coordinates by given angular sizes of two segments connecting lighthouses M_1 and M_2, M_2 and M_3.

Input

The first line contains an integer n — number of possible locations of the lighthouses ($1 \le n \le 50\,000$). Each of the next n lines contains eight integers. The first six integers describe Cartesian coordinates of different points M_1, M_2, M_3 that do not exceed 10 000 by absolute value; the next two integers from the range [1; 179] are angular sizes (in degrees) of segments M_1M_2 and M_2M_3. It is guaranteed that the solution exists for each of the test cases.

Output

For each location of the lighthouses output coordinates of any possible ship position separated by space in a separate line. The answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Note that ship and lighthouse cannot share a position.

Examples

standard input	standard output
1 0 0 2 0 2 2 90 90	1.0000000000 1.0000000000

Problem B. High-Speed Pedestrian walkway 1.0

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

Construction of the High-Speed Pedestrian walkway that connects two capitals was ordered by the Department for Innovation. But nobody knew how to construct it. Fortunately, there is a unique plant in the city for pedestrian walkway components production.

The plant produces m types of tiles – rectangles of $1 \times k$ cm, available in c_k different colors,$1 \le k \le m$.

The pedestrian walkway consists of a rectangular segments of $2 \times n$ cm. Each segment in the construction is assembled with tiles of above mentioned kind. Since the completion of the pedestrian walkway will be accepted by the top Heads, so it was decided to construct it from unique segments, in other words, all segments should be different in way of tiling.

Now it is important to find out what the maximum possible number of unique segments is.

Input

The first line contains integers n and m ($1 \le n \le 10^{18}$, $1 \le m \le 9$) — length of a segment and number of tiles' kinds. The second line contains space separated m non-negative integers $c_1, c_2, \ldots, c_m$, at least one of which is different from zero. All integers do not exceed 10^9.

Output

Output the required number modulo $(10^9 + 9)$.

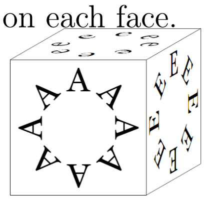
Examples

standard input	standard output
3 3 0 1 2	7
2 4 0 0 1 0	0

Problem C. Cubes

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

Gennady, who worked all his life at the Kazan Slot Machine Plant, went to Las Vegas and at the first casino discovered a slot machine of his own production. The slot machine contains N cubes with Latin letters written in a special way on each face.



At the beginning of the game, the slot machine puts the cubes on each other so that a player sees only letters on one side of the tower. Then the screen displays the word of N letters, which the player must draw on the visible side of the tower, by turning the cubes around the vertical axis. The word should be read from top to bottom.

Gennadiy has a huge advantage —- he remembers all cubes produced at his plant and can distinguish them from each other in the slot machine. However, he cannot know the exact orientation of the cube in the tower as he sees just one cube face. Help Gennady calculate the probability of that the task set by the slot machine is executable.

Input

The first two lines contain words with a length of N characters —- the initial and target words on the visible side of the cube tower ($1 \leq N \leq 10$). The next N lines contained six characters each describing the letters on the cube faces in the following order: front, top, right, bottom, left and rear ones. Cubes are listed in order from top to bottom. Uppercase and lowercase letters are considered to be different.

Output

Output the probability. The answer is considered to be correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

standard input	standard output
HALLW HELLO XABCDH XAECDe AbcdeL AbcdeL ABOWCD	0.25

Note

We see the face with the ‘A’ letter on the second cube, and we need to rotate the cube so that we see the face with the ‘E’ letter. Depending on the rotation of the face with the ‘A’ letter, the face with the 'E' letter may be "on the side" or "on the top/bottom" towards Gennadiy. In the first case, we can rotate the cube as required, but in the second and third cases — we cannot. The probability of finding 'E' "on the side" is equal to the probability of finding it "on the top/bottom" and amounts to $1/2$. Therefore, the probability to rotate the cube the right way is $1/2$. The 'W' and 'O' letters are in a similar manner, so the overall probability is $1/4$.

Problem D. Camelogistics

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

Logistics is a very difficult thing. Logistics in a desert is an incredible art.

In a vast desert there is a warehouse with N apples. There is a plant which produces apple juice L kilometers away from the apple warehouse. The only way to transport apples is a freight camel parked near the warehouse. This camel has a phenomenal load capacity, but it cannot carry more than K apples. Moreover the camel needs eat one apple each kilometer otherwise there will be no way to carry the next batch. After each kilometer on the way from the warehouse there is an awning where it is possible to leave any amount of apples.

Find out the maximum possible amount of apples that can be delivered to the plant.

Input

A single line contains space-separated integers L, N, K ($1 \le L, N, K \le 10^9$)

Output

Output a single number — the highest possible number of apples delivered to the plant.

Examples

standard input	standard output
5 30 10	11
1000 2999 1001	533
5 4 10	0

Problem E. The Carpet

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

A perfectly round crater was found on a distant planet. Two stations arrived to study the crater. In order to start exploration, it is necessary to lay a special carpet roll on the surface, so that the rovers will use it to move around. It is possible to unroll the carpet all over the surface except for the crater area. The rovers must be able to move on the carpet roll from one station to another and to the crater's edge. Carpets are very expensive in Space, that's why it is necessary to make the carpet roll's length as short as possible.

Input

The first line contains two pairs of Cartesian coordinates of the stations separated by space. The second line contains coordinates of the center and the crater's radius, respectively. All numbers are integers and do not exceed $4 \cdot 10^4$ by absolute value. It is guaranteed that the stations are placed outside the crater.

Output

Output the total length of the carpet roll with absolute or relative error that does not exceed 10^{-6}.

Examples

standard input	standard output
3 4 4 4 0 0 5	1.000000

Problem F. GCD and LCM

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

It is not difficult to find out the greatest common divisor and the least common multiple of several integers. No doubt you have solved such problems... But have you ever tried to find the integers by their greatest common divisor (GCD) and least common multiple (LCM)? Or, at least, have you determined how many of such sets do GCD and LCM have? Probably, you haven't ...

Find out how many ordered sets of k positive integers which greatest common divisor and the least common multiple are equal to d and m respectively. For example, for $k = 2$, $d = 2$, $m = 12$, there are four described sets: $(2, 12)$, $(12, 2)$, $(4, 6)$ and $(6, 4)$.

Input

The first line contains one integer k — number of integers in the sets ($2 \le k \le 10^{18}$). The second line contains integers d and m — GCD and LCM of integers in the set($1 \le d \le m \le 10^9$).

Output

Output the number of ordered sets by modulo $(10^9 + 9)$.

Examples

standard input	standard output
2 2 12	4
3 4 5	0

Problem G. Pots

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

In a cooking pot shop sales decreased dramatically. Marketing managers of the shop did a research and found out that the reason was frying pans. People stopped buying pots as pans are both cheaper and more compact during storage. The Board of Directors made a decision to extent assortment and start selling also pans. The first batch is already ordered.

Warehouse logistics department was given a task to find a place for the new goods. Now there are N pots in the warehouse. Every pot has a diameter D_i. There is the only way to save space — it is possible to embed into any pot another one of a smaller diameter, into which in turn other can be embedded.

Help the logistics specialist to find a minimal number of pots in the warehouse in which all other pots can be embedded.

Input

The first line contains a single number N ($1 \leq N \leq 1000$). The second line contains N integers D_i separated by spaces ($1 \leq D_i \leq 10\,000$).

Output

Output the obtained number.

Examples

standard input	standard output
5 7 5 2 5 2	2

Problem H. Messenger

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

New technologies appear every day. A famous message delivery service called «LCI» was launched a few years ago. The messenger became popular by its technical error — all communication centers distort messages. A numeric key X_i was assigned to each communication center, and transition of each message inverted its prefix of X_i characters length and the suffix of $L - X_i$ characters, where L is a length of the message. The text becomes unrecognizable, but people like it.

Version 16.0, which fixes the annoying bug, has recently been released. Messages began to reach the destination without changes, but it turned out that «LCI» became as boring as hundreds of other messengers. People lost all their interest and deleted their accounts.

The messenger developers have tried to return the well-loved pecularity, but it turned out that it was impossible to rollback the communication centers to the previous firmware. Now, it is necessary to create an «LCI» Network Emulator, which can calculate the final message by sent message and passed centers keys.

Input

The first line contains the sent message — an L character string ($1 \le L \le 200\,000$) consisting of lowercase Latin letters, numbers, spaces and punctuation marks. The second line contains the single integer N ($1 \le N \le 200\,000$) — a number of communication centers passed by the message. The next N lines contain communication center keys, X_i ($0 \le X_i \le L$), in accordance with their passing order.

Output

A single line should contain the final message.

Examples

standard input	standard output
abcde 5 2 3 5 3 1	baedc

Problem I. Manhattan Project

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

In a seven-dimensional space there is a six-dimensional institution where five-dimensional workers run a four-dimensional data base. The database contains information about the points defined by four coordinated. The institution receives three kinds of requests :

1. Add a point with the given coordinates
2. Delete a point with the given coordinates.
3. Provide a distance between the given point and the most distant point.

In this space the distance between the points (x_1, x_2, x_3, x_4) and (y_1, y_2, y_3, y_4) is calculated as $|x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3| + |x_4 - y_4|$.

There going to be some staff reduction in the institution, so it is necessary to have the above mentioned work automated.

Input

The first line contains an integer N ($1 \le N \le 10^5$) — a number of requests. Each of the next N lines contains five integers: t, x_1, x_2, x_3, x_4 — a kind of request and four coordinates of the point respectively ($1 \le t \le 3, -10^8 \le x_1, x_2, x_3, x_4 \le 10^8$).

It is guaranteed that at the moment of request of type 1 (add) the point (x_1, x_2, x_3, x_4) does not exist, at the moment of request of type 2 (delete) the point (x_1, x_2, x_3, x_4) exists, at the moment of request of type 3 (distance request) at least one point exists. There is no points in the database until the first request arrives.

Output

For each request of 3 kind output the request's result in a separate line.

Examples

standard input	standard output
5 1 0 0 0 0 1 -10 2 6 -9 3 -8 0 9 -5 2 0 0 0 0 3 -8 0 9 -5	22 11
9 1 2 0 0 0 1 4 3 0 0 1 1 5 0 0 3 2 3 0 0 3 -1 2 0 0 2 4 3 0 0 3 -1 2 0 0 2 2 0 0 0 3 1 5 0 0	3 6 5 0

Problem J. Liquid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

In 2316 after successful colonization of XMars people started to colonize another planet – XVenus. First of all, they have sent a space probe to the planet in order to get the place prepared for living modules installation. A small spaceship with special liquid on board leaves the orbital station, heads to the planet and pours the liquid down to the surface, and the places treated like that become liveable.

The whole process is organized in the following way: the planet's surface is divided into many segments. Every segment is given a number starting from zero. The program installed at the Station has n stages of two different types: surface treatment and surface cleaning. If the spaceship treats the surface on i-stage, it carries p_i tons of the liquid onboard. One ton of the liquid is enough to treat one segment. The spaceship scans the surface, detects the first untreated segment that has the number greater than x_i and from here starts to treat the segments, moving in an increasing order. If the spaceship runs out of the liquid or encounters the segment that has already been treated, it returns to the station, bringing the liquid back to the storage tank. After that, the spaceship loads again the liquid and departs to fulfil the next stage.

Surface cleaning as a tool engineered to fix the mistakes in the surface treatment system. If on i-stage the spaceship cleans the surface, it lands on sector x_i and if it is flooded by liquid, eliminates it, restoring the surface's initial state.

Find out the segments that will be treated at each stage of the first type.

Input

The first line contains positive integer n that doesn't exceed 10^5. The following n lines describe stages. Two positive integers x_i and p_i in each, separated by spaces ($0 < x_i \le 5 \cdot 10^8, 0 < p_i \le 1000$), describe surface treatment stage. A single negative integer y_i ($-5 \cdot 10^8 \le y_i < 0$) describes the cleaning stage, i.e it is necessary to clean the segment $x_i = -y_i$.

Output

In a separate line print two space-separated numbers: the first and the last sector numbers, treated on this stage. Output the results as per order in the input file.

Examples

standard input	standard output
6 5 2 6 3 4 7 1 1 -4 2 4	5 6 7 9 4 4 1 1 2 4

Problem K. Partial Reverse

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

Consider strings of length N consisting of zeros and ones. Lets define a *partial reverse*:

- Choose a substring of N, containing *even* number of ones.
- Replace this substring by one with reversed order of characters (i.e. `001010` will be replaced with `010100`).

Two words are *equivalent*, if we can create one word from the another using no more than $16 \cdot N$ partial reverse operations.

Given two words of same length, determine if they are equivalent, and, if so, find atleast one sequence of partial reverse operations, transforming first word into the second.

Input

First line of the input file contains first word, second one contain second word. Both words have the same length N ($2 \leq N \leq 10^4$) and consist only of ones and zeroes.

Output

If those words are not equivalent, print "`NO`" in a single line.

Otherwise, print "`YES`" in the first line. Second line must contain one integer K — number of operations used to transform first word into the second. i-th of next K lines must contain description of i-th operation — two integers $a_i < b_i$ — 1-based indices of the first and last character of substring, used for i-th operation.

If here are more than one correct sequence, print any of them.

Examples

standard input	standard output
110001111 111100011	YES 1 3 7
110 010	NO

Problem L. Three machines

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

Three machines read integers from the inserted cards and print new cards with pairs of positive integers. The first machine reads the card $(a; b)$ and outputs a new card $(a + 1; b + 1)$.The second machine reads the card $(a; b)$, and if both numbers are even integers, outputs the card $(a/2; b/2)$. When two cards $(a; b)$ and $(b; c)$ are inserted into the third machine, it outputs the card $(a; c)$. All machines return the inserted cards.

There is a single card $(a; b)$. It is necessary to obtain the card $(c; d)$ in no more than 15000 operations or find that it's impossible.

Input

The first line contains integers a and b. The second line contains c and d $(1 \leq a, b, c, d \leq 2000)$. The initial and the final cards are different.

Output

If it is not possible to obtain the required card, output 0. Otherwise print k in the first line — number of times the machines have been used. In the following k lines output usage description in the following format: « <number of the machine> <integers on the first card> [<integers on the second card>] ». If the third machine is used, the second integer on the first card must be equal to the first integer on the second card. k must not exceed 15000.

Examples

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

Problem M. The smallest fraction

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

Currently the world knows a great number of sensational archaeological discoveries. Data storage devices relating to the period of programming of the XX century were discovered during recent excavations. Decryption of files allowed scientists to prove the hypothesis that ancient programmers were able to make simple arithmetic operations with fractions. Many texts have been deciphered, many mysteries have been solved. However one problem has remained unsolved: a calculation of *the smallest* positive fraction that, if divided by each of n given fractions, results in an integer number.

May be you will manage to solve it...

Input

The first line contains an integer n ($1 \leq n \leq 6$). Each of the following n lines contain two integers a_i, b_i that are numerator and denominator of irreducible fraction ($1 \leq a_i \leq 10^3$, $1 \leq b_i \leq 10^9$).

Output

Print two positive integers separated by space that are numerator and denominator of the smallest irreducible fraction satisfying the condition of the problem.

Examples

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