Problem A. Cookies

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 mebibytes

You are to hold a cookie party! You have prepared N cookies numbered from 1 through N. The **sweetness** of the cookie i is A_i. You expect that M children numbered from 1 to M will attend the party. All of them will bring their homemade cookies, and the child i will bring a cookie of sweetness B_i. Besides, you know the taste preference of each child. The child i loves sweet cookies if $S_i =$`S` and loves bitter cookies if $S_i =$`B`.

The party will proceed in the following manner:

- First, you are given an integer k and put cookies $1, 2, \cdots, k$ on the table.
- Then, children $1, 2, \cdots, M$, in this order, come to the table. When the child i comes to the table, the child first put his/her homemade cookie on the table. Then, if the child loves sweet cookies, he/she eats the sweetest cookie (a cookie with the largest sweetness) on the table. If the child loves bitter cookies, he/she eats the bitterest cookie (a cookie with the smallest sweetness) on the table. Note that each child eats exactly one cookie, and a child may eat his/her cookie.
- Finally, you eat all the cookies left on the table.

You have not yet decided the value of k. For each integer $k = 1, 2, \cdots, N$, find the sum of the sweetness of cookies you will eat.

Note that only after you get the answer for $k = i$ can you know the value of A_{i+1}. See the input section for more details.

Input

Input is given from Standard Input in the following format:

N

$A'_1\ A'_2\ \cdots\ A'_N$

M

$B_1\ B_2\ \cdots\ B_M$

S

Here A'_i is the encrypted value of A_i, and the real value can be calculated as $A_i = (A'_i + lastans \mod 10^9)$, where $lastans$ denotes the answer for $k = i - 1$ if $i > 1$ and 0 if $i = 1$.

Constraints

- $1 \le N \le 2 \times 10^5$
- $0 \le A_i \le 10^9 - 1$
- $0 \le A'_i \le 10^9 - 1$ (see the input section for the definition of this variable)
- $1 \le M \le 2 \times 10^5$
- $0 \le B_i \le 10^9 - 1$
- $|S| = M$
- S_i is either '`S`' or '`B`'.

- All values in input are integers.

Output

Print N integers in one line. The i-th integer should be the sum of the sweetness of cookies you will eat when $k = i$.

Examples

standard input
3 3 999999999 0 2 4 2 BS

standard output
2 5 9

standard input
10 810737462 262894941 12979345 90139935 834123271 768745833 928886601 144082546 35547099 840309069 10 854737038 93768450 848842263 62613614 800833082 316988396 203584286 283415773 762732633 756024517 SBSSSBSSBS

standard output
756024517 959608803 1243024576 1560012972 1893177483 2287313726 2503514053 3151110652 3337768403 3515845875

Note

In first sample, $A = (3, 1, 5)$.

When $k = 2$, the party proceeds as follows:

- You put 2 cookies of sweetness 3 and 1.
- The child 1 puts the cookie of sweetness 4 and eats the cookie of sweetness 1.
- The child 2 puts the cookie of sweetness 2 and eats the cookie of sweetness 4.
- You eat cookies of sweetness 2 and 3.

Problem C. Sum Modulo

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

Snuke found a random number generator. It generates an integer between 1 and N (inclusive). An integer sequence $A_1, A_2, \cdots, A_N$ represents the probability that each of these integers is generated. The integer i ($1 \le i \le N$) is generated with probability A_i/S, where $S = \sum_{i=1}^{N} A_i$. The process of generating an integer is done independently each time the generator is executed.

Snuke has an integer X, which is now 0. He can perform the following operation any number of times:

- Generate an integer v with the generator and replace X with $X + v \mod M$.

Find the expected number of operations until X becomes K, and print it modulo 998244353. More formally, represent the expected number of operations as an irreducible fraction P/Q. Then, there exists a unique integer R such that $R \times Q \equiv P \mod 998244353,\ 0 \le R < 998244353$, so print this R.

We can prove that the expected number of operations until X becomes K is a finite rational number. However, we did **not** prove its integer representation modulo 998244353 can be defined. Our sincerest apologies. Nonetheless, you don't have to worry about division by 0. More precisely, We can model this problem as an absorbing markov chain (`https://en.wikipedia.org/wiki/Absorbing_Markov_chain`), and we guarantee that in all tests, the corresponding fundamental matrices can be defined modulo 998 244 353.

Input

Input is given from Standard Input in the following format:

$N\ M\ K$

$A_1\ A_2\ \cdots\ A_N$

Constraints

- $1 \le N \le \min(500, M - 1)$
- $2 \le M \le 10^{18}$
- $1 \le K \le M - 1$
- $1 \le A_i \le 100$
- All values in input are integers.

Output

Output the expected number of operations until X becomes K, modulo 998 244 353.

Example

standard input	standard output
2 3 1 1 1	2
10 100 50 1 2 3 4 5 6 7 8 9 10	439915532

Problem D. Xor Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Determine whether there exists a sequence of N nonnegative integers $a_1, a_2, \cdots, a_N$ that satisfies all of the following conditions, and if it exists, find the minimum possible value of the maximum of the array.

- $a_1 + a_2 + \cdots + a_N = S$
- $a_1 \oplus a_2 \oplus \cdots \oplus a_N = X$ (Here $\oplus$ denotes bitwise xor operation)

Note that there are T tests in one input file.

Input

Input is given from Standard Input in the following format:

T

$N_1\ S_1\ X_1$

$N_2\ S_2\ X_2$

$\vdots$

$N_T\ S_T\ X_T$

Here, N_i, S_i, X_i represent values of N, S, X for the i-th test, respectively.

Constraints:

- $1 \le T \le 500$
- $1 \le N \le 2^{60} - 1$
- $0 \le S \le 2^{60} - 1$
- $0 \le X \le 2^{60} - 1$
- All values in input are integers.

Output

Print T lines. In the i-th line, print -1 if there doesn't exist an array with the mentioned property in the i-th test, and print the minimum possible value of the maximum of the array if it exists.

Example

standard input	standard output
6 3 9 3 4 8 0 6 19 1 1 15 15 2 6 5 5 4 3	3 2 4 15 -1 -1

Note

The following is a solution for each test:

- (3,3,3)
- (2,2,2,2)
- (2,3,3,3,4,4)
- (15)
- Impossible
- Impossible

Problem E. Count Modulo 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3.5 seconds
Memory limit:	1024 mebibytes

You are given K distinct nonnegative integers $A_1, A_2, \cdots, A_K$. Count the number of sequences of N nonnegative integers $a_1, a_2, \cdots, a_N$ that satisfies all of the following conditions, modulo **2**.

- $a_1 + a_2 + \cdots + a_N = S$
- For each i $(1 \leq i \leq N)$, there exists an integer j such that $a_i = A_j$.

Note that there are T tests in one input file.

Input

Input is given from Standard Input in the following format:

T

Description of the 1-st test

Description of the 2-nd test

$\vdots$

Description of the T-th test

The description of each test is in the following format:

N S K

A_1 A_2 $\cdots$ A_K

Constraints:

- $1 \leq T \leq 5$
- $1 \leq N \leq 10^{18}$
- $0 \leq S \leq 10^{18}$
- $1 \leq K \leq 200$
- $0 \leq A_1 < A_2 < \cdots < A_K \leq 10^5$
- All values in input are integers.

Output

For each test, print the count modulo 2.

Example

standard input
2 5 10 3 1 2 3 1000000000000000000 2545332177123938 1 10 0 1683 21728 31623 35054 37834 39329 56842 68603 74742

standard output
1 0

Note

In the first test, there are a total of 51 sequences that satisfy conditions.

Problem F. Robots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

There are N robots numbered from 1 through N and N antennas numbered from 1 through N in a straight line. The coordinate of the robot i is a_i and the coordinate of the antenna i is b_i. All coordinates are distinct.

Currently, all antennas are inactive. You are going to activate them one by one. When you activate an antenna, the nearest robot (if two robots are closest to it, only the left one) moves to the antenna and explodes along with it.

Find an order to activate antennas so that the total distance of robots' moves is minimum possible.

Input

Input is given from Standard Input in the following format:

N

$a_1\ a_2\ \dots\ a_N$

$b_1\ b_2\ \dots\ b_N$

Constraints:

- $1 \le N \le 2 \times 10^5$
- $0 \le a_1 < a_2 < \cdots < a_N \le 10^9$
- $0 \le b_1 < b_2 < \cdots < b_N \le 10^9$
- $a_1, a_2, \dots, a_N, b_1, b_2, \dots, b_N$ are all distinct.
- All values in input are integers.

Output

Print the answer in the following format:

X

$p_1\ p_2\ \dots\ p_N$

Here, X must be a minimum total distance, and p_i is the index of the antenna that you activate in the i-th.

If there are multiple solutions, you can print any of them.

Example

standard input	standard output
3 1 2 3 11 12 13	30 3 2 1

Problem H. Construct Points

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Construct four points a, b, c and d on the two-dimensional plane that satisfy the following conditions:

- x and y coordinates of all points are integers, and their absolute values don't exceed 10^9
- a and b are different, and c and d are different.
- Let l be the line passing through a and b, and let m be the line passing through c and d.

Then,

- l and m are not parallel.
- the absolute values of x and y coordinates of the cross point of l and m are not less than 10^{27}.

Input

There is no input.

Output

Print the answer in the following format:

a_x a_y

b_x b_y

c_x c_y

d_x d_y

Problem I. Amidakuji

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a positive integer N. Construct a sequence of permutations of $(1, 2, \cdots, N)$, $p_1, p_2, \ldots, p_K$, that satisfy following conditions, or report that it's impossible.

- $0 \leq K \leq \lceil \log_2 N \rceil + 1$, where K is the length of the sequence.
- $p_1, p_2, \ldots, p_K$ are permutations of $(1, 2, \ldots, N)$.

In other words, they are bijections from
$1, 2, \ldots, N$ to
$1, 2, \ldots, N$.

- For all x and y $(1 \leq x, y \leq N)$, there is a sequence of bijections $q_1, q_2, \ldots, q_K$ such that $(q_K \circ q_{K-1} \circ \cdots \circ q_1)(x) = y$ and $q_i = p_i$ or p_i^{-1} for all i.

Here, $\circ$ denotes function composition, and when $K = 0$, $q_K \circ q_{K-1} \circ \cdots \circ q_1$ is defined as an identity function.

Input

Input is given from Standard Input in the following format:

N

Constraints:

$1 \leq N \leq 1000$

Output

If there is no solution, print -1. Otherwise, print the answer in the following format:

K

$p_{1,1}$ $p_{1,2}$ $\cdots$ $p_{1,N}$

$\vdots$

$p_{K,1}$ $p_{K,2}$ $\cdots$ $p_{K,N}$

Here, $p_{i,j}$ must be a value of $p_i(j)$.

If there are multiple solutions, you can print any of them.

Example

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

Note

In Sample 1 for $x = 2, y = 1$, we can set $q_1 = p_1, q_2 = p_2^{-1}, q_3 = p_3$ and get $q_3(q_2(q_1(2))) = 1$.

Problem M. Zig Zag Nametag

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

When ninjas go to conferences they wear fake nametags. One ninja in particular wants to impress his Sensei. His Sensei chooses a new favorite number every day. The pupil wants to put a name on his nametag that encodes his Sensei's favorite number! This name will consist of only lower case letters. He assigns a value to each letter, based on its position in the alphabet (e.g. $a = 1$, $b = 2$, ..., $z = 26$). Then, he encodes the Sensei's number by adding up the absolute values of the differences of every consecutive pair of letters. For example, the string "`azxb`" has the value of:

$$|a - z| + |z - x| + |x - b| = |1 - 26| + |26 - 24| + |24 - 2| = 49$$

The name that the ninja will write on his nametag is the shortest string that encodes to his Sensei's favorite number. If there's more than one string of the shortest length, he'll choose the one that comes first alphabetically. Given the Senseis favorite number, k, find the string that the ninja should put on his nametag.

Input

Input consists of a single line with a single integer k ($1 \leq k \leq 10^6$), which is the Sensei's favorite number. There will always be a name that encodes to the Sensei's number.

Output

Output a single line with a string of lower case letters, which is the name that the ninja should put on the nametag to impress the Sensei.

Examples

standard input	standard output
1	ab
19	at
77	aoazb

Problem N. Knight Jumps

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

In Chess, a knight can move two squares in one direction and then one in a perpendicular direction. It can jump, meaning that it only requires that the destination square be open — the path between can be occupied.

Given a grid, a starting point and destination point, determine the least number of moves the knight must make to get from the start to the destination. Some squares of the grid may be occupied, so that the knight cannot move there.

Input

Each input will begin with two integers n and m ($2 \le n, m \le 100$), indicating the height and width of the grid. Each of the next n lines will hold m characters, representing the grid. The grid will consist only of '`.`' (open square), '`#`' (occupied square), '`K`' (the knights starting position) or '`X`' (the knights destination). There will be exactly one '`K`' and exactly one '`X`' in each input.

Output

Print a single line with a single integer indicating the minimum number of moves the knight needs to get to the destination, or -1 if the knight cannot make it.

Examples

standard input	standard output
5 4 K... .... .#.. .#.. ...X	5
3 3 K.. .X. ...	-1

Problem O. Lost Digit

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

You are helping an archaeologist decipher some runical digits. He knows that this ancient society used a Base 10 system, and that they never start a number with a leading zero unless it is exactly zero (single digit). He's figured out most of the digits, as well as a few operators, but he needs your help to figure out the rest.

The professor will give you a simple math expression. He has converted all of the runes he knows into digits. The only operators he knows are addition (+), subtraction (-), and multiplication (*), so those are the only ones that will appear. Each number will be in the range from -999999 to 999999, and will consist of only the digits '0'-'9', possibly a leading '-', and possibly a few ?'s. The ?'s represent a digit that the professor doesn't know (never an operator, an '=', or a leading '-'). All of the ?'s in an expression will represent the same digit, and it won't be one of the other given digits in the expression.

Given an expression, figure out the value of the rune represented by the question mark. If more than one digit works, give the lowest one. If no digit works, well, that's bad news for the professor — it means that he's gotten some of his runes wrong. Output -1 in that case.

Input

Input will consist of a single line, of the form: $<$ *number* $><$ *op* $><$ *number* $>=<$ *number* $>$. Each $<$ *number* $>$ will consist of only the digits '0'-'9' , with possibly a single leading minus ('-'), and possibly some ?'s. No number will begin with a leading 0 unless it is 0, no number will begin with -0, and no number will have more than 6 places (digits or ?'s). The $<$ *op* $>$ will separate the first and second $<$ *number* $>$'s, and will be one of: '+', '-' or '*'. The '=' will always be present between the second and third $<$ *number* $>$'s. There will be no spaces, tabs, or other characters. There is guaranteed to be at least one ? in every equation.

Output

Print a single line with the lowest digit that will make the equation work when substitued for the ?'s, or print -1 if no digit will work.

Examples

standard input	standard output
1+1=?	2
123*45?=5?088	6
-5?*-1=5?	0
19--45=5?	-1
??*??=302?	5

Problem P. Multiple Top N

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

In College Football, many different sources create a list of the Top 25 (or, Top n) teams in the country. Since it's subjective, these lists often differ, but they're usually very similar. Your job is to compare two of these lists, and determine where they are similar.

In particular, you are to partition them into sets, where each set represents the same contiguous positions in both lists, and has the same teams, and is as small as possible. If the lists agree completely, you'll have n sets, where n is the number of teams in each list.

Input

Input will begin with an integer n ($1 \le n \le 10^6$), indicating the number of teams ranked. The next n lines will hold the first list, in order. The team names will appear one per line, and consist of between 1 and 8 capital letters only. After this will be n lines, in the same format, indicating the second list. Both lists will contain the same team names, and all n team names will be unique.

Output

Output the size of each set, in order, one per line.

Examples

standard input	standard output
5 A B C D E A C D B E	1 3 1
3 FIRST SECOND THIRD FIRST SECOND THIRD	1 1 1
3 UNITED CITY CHELSEA CHELSEA UNITED CITY	3

Problem Q. New Contest Director

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

Elections of the new Contest Director are hold! Help the election commission by counting votes and telling them the winner. If more than one candidate ties with the most votes, print out all of their names in alphabetical order.

Input

Input will begin with an integer n ($1 \leq n \leq 1000$), indicating the number of votes. The next n lines will hold the votes. The candidate's names will appear one per line, and consist of between 1 and 20 capital letters only.

Output

Print the name of the candidate with the most votes. If there is a tie, print out all of the names of candidates with the most votes, one per line, in alphabetical order.

Examples

standard input	standard output
5 BILL MIKE BILL BILL MIKE	BILL
5 BILL MARSHA BILL MARSHA ANONYMOUS	BILL MARSHA