Problem A. A Lot

Input file:	`alot.in`
Output file:	`alot.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes to help his friends a lot. Each of his friends has been assigned a homework of one task. Petya would love them to spend time playing with him rather than doing homework. That's why he decided to do all their homework by himself. Petya has Q friends, so he has to solve Q different tasks.

His friends' teacher has given a prime number P which is common for all tasks. Each specific task is about finding minimal integer Z in the interval $[0, P-1]$ inclusively which satisfies the equation:

$$X^Z \equiv Y \pmod{P}$$

Here X and Y are integers given individually to each student. Let's consider $0^0 = 1$.

Unfortunately Petya has a lot of friends, thus there are a lot of tasks to solve. You are to help Petya and find the answers to all of them.

Input

The first line of the input contains two integers P and Q ($2 \leqslant P \leqslant 10^8$, $1 \leqslant Q \leqslant 10^4$). P is guaranteed to be prime. Next Q lines contain two integers each — X and Y ($0 \leqslant X, Y < P$) which are values given in the corresponding tasks of Petya's friends.

Output

Output Q lines, one for each task given in the input. Each line should contain the minimal value of Z or -1 if it doesn't exist. Note that there could exist several solutions, but you should output only the minimal.

Examples

alot.in	alot.out
5 5 2 4 3 2 4 1 4 2 1 1	2 3 0 -1 0

Problem B. Almost Average

Input file:	`almost.in`
Output file:	`almost.out`
Time limit:	4 seconds *(6 seconds for Java)*
Memory limit:	256 mebibytes

Little Petya likes numbers a lot. He says that the number Q is the almost average value of the collection of P ($P > 1$) numbers with the sum S if Q is equal to $S/(P-1)$.

Recently he has received an array of N integers which are enumerated from 1 to N as a gift from his mother. Such arrays are not common in his town, so he decided to show it to each of his M friends. For this occasion they formed a queue and Petya is going to show his array to all friends strictly in the same order in which they are standing in the queue. He doesn't want to make a crowd in his flat, so he decided that only one friend can be in his flat at a time.

His friends know that it is impolite to come visiting someone without a present, so each of them decided to present Petya a question about his array. Each question is of the form: given two integers L and R ($1 \leqslant L < R \leqslant N$), find the two integers A and B, such that $1 \leqslant L \leqslant A < B \leqslant R \leqslant N$ and the almost average value of all integers from the array which have indexes between A and B, inclusive, is maximized. If there are several such pairs of A and B, he can pick any of them.

Help Petya to answer all friends' questions.

Input

The first line of the input contains two integers N and M ($1 \leqslant N \leqslant 10^5$, $1 \leqslant M \leqslant 3 \times 10^4$). The second line contains N integers — the array that Petya has received from his mother. All of them are not greater than 10^6 by an absolute value. Each of the next M lines contains two non-negative integers, not greater than 10^6 — numbers prepared by Petya's friends in order they are standing in the queue in encrypted format. In order to obtain the corresponding values of L and R for a particular question one should perform the following steps:

- Let A and B be the answer for the previous friend's question and P/Q be the almost average value of all numbers from the array with indexes between A and B, inclusive, in the form of an irreducible fraction. Let K be the number equal to $|P| \bmod |Q|$, where $|X|$ is an absolute value of the number X and "$|P| \bmod |Q|$" means the remainder of division. If this is the first friend's question, then we'll say that $K = 0$.
- Let L' and R' be the numbers read from the input.
- Calculate the values of L and R using the following formulas: $L = L' \oplus K$, $R = R' \oplus K$, where $\oplus$ is the bitwise exclusive or operation. It is guaranteed that if K was computed correctly, then L and R will satisfy the following inequality: $1 \leqslant L < R \leqslant N$.

Output

Output one number — the almost average value of all elements from the array with indexes between A and B, inclusive, where A and B are the answer for the last friend's question. You should print this number in the form of an irreducible fraction P/Q (see examples for further clarification).

Examples

almost.in	almost.out

`7 1`
`-1 -3 -1 -500 -2 -1 -2`
`1 5`
	`-5/2`

`6 2`
`10 -1 10 -1000000 3 3`
`1 6`
`2 7`
	`6/1`

Note

In the first example the optimal segment is either $[1,3]$ or $[5,7]$.

In the second example the optimal segment for the first query is $[1,3]$ which has the almost average value equal to $19/2$. For the second query, $K = 19 \bmod 2 = 1$, which means that $L = 3$ and $R = 6$. The optimal segment for this query is $[5,6]$.

Problem C. Amoeba

Input file:	`amoeba.in`
Output file:	`amoeba.out`
Time limit:	3 seconds
Memory limit:	256 mebibytes

Little Petya likes cosmic amoebas a lot. Recently he has received a tree of N vertices as a gift from his mother. He wants to place one of his amoebas on the tree.

In order to place a cosmic amoeba of age d years and which has t tentacles on the tree one should do the following. First, a vertex should be chosen. The body of the cosmic amoeba will be placed into this vertex. Second, one should choose t simple paths of length d edges which will contain the amoeba's tentacles. A path is considered to be simple if all vertices in it are distinct. Each path should contain the central vertex as one of its ends. Every pair of paths should have exactly one common vertex, which is the central one.

Petya has a collection of cosmic amoebas of all ages between A and B years, inclusive. For each integer d which belongs to the interval $[A, B]$ he wants to find out what is the maximal number of tentacles a d-year-old amoeba can have so that it will still be possible to place it on the tree. The bodies of amoebas of different ages can be placed in different vertices.

Input

The first line of input contains three integer numbers N, A and B ($1 \leqslant N \leqslant 3 \times 10^5$, $1 \leqslant A \leqslant B \leqslant N-1$). Each of the next $N-1$ lines contain two numbers which represent an edge of the tree. Vertices are numerated from 1 to N. It is guaranteed that the graph in the input forms a tree.

Output

Output $B-A+1$ numbers, each on a separate line. The i'th number of output should contain the maximal number of tentacles which an amoeba of age $A+i-1$ years can have.

Examples

amoeba.in	amoeba.out
6 1 5 1 2 2 3 1 4 4 5 1 6	3 2 1 1 0

Problem D. Automaton

Input file:	`automaton.in`
Output file:	`automaton.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes deterministic finite state automatons a lot. He knows everything about them, though he is not very good at languages and he only knows the first two letters of English alphabet: '`A`' and '`B`'. Considering this, we'll only take into account a binary alphabet in this problem. Recently Petya has received a book called «Deterministic finite state automatons for kids of ages 2–9» from his mother. Let's introduce a few definitions from this book (they are simplified for kids, so **use them as formal rules**):

A deterministic finite state automaton (DFA or just an automaton) is a directed graph in which each vertex has exactly two outgoing edges: one of them is labeled with the letter '`A`' and another one is labeled with the letter '`B`'. In addition to this, there is exactly one starting vertex and at least one terminal vertex in the graph.

DFA processes the string S in the following way. At the very beginning the marker is in the starting vertex and the automaton has the whole string S on the input. On each turn, if the string is not empty, it takes the first character of the string, moves the marker from the current vertex along the edge with the label corresponding to this letter to adjacent vertex. After that it removes the first letter from the string. The process continues until the string on the input is empty. If the marker is in the terminal vertex at this point, then the automaton accepts the string S, otherwise it rejects the string.

We'll say that a DFA describes a set of strings if it accepts every string from the set and rejects every string which is not in the set.

A DFA is considered to be minimal if there is no DFA with smaller number of vertices which describes the same set of strings.

Of course, Petya knew all those definitions and he found the book boring very soon. There remained only one page which Petya didn't consider boring. It was a catalogue of automatons, where he could choose and buy any DFA present in the catalogue.

The catalogue contains only minimal automatons, because all others are considered too dangerous for kids under 9 years old. In addition to this, each automaton in the catalogue can accept only the strings of length L. All automatons which satisfy these two conditions are present in the catalogue. The price of an automaton is equal to the number of strings in the set it describes.

Petya wants to buy an automaton with the largest possible number of vertices. If there are several such automatons, he will choose the cheapest one. Help him to find such automaton. Output the number of vertices in it and its price.

Input

Input contains the only number L ($1 \leqslant L \leqslant 1000$), which means that all automatons in the catalogue can accept only strings of length L.

Output

Output should contain two numbers, separated by a single space. The first one is the largest possible number of vertices in an automaton which can be found in the catalogue. The second one is the lowest price of an automaton with the largest number of vertices.

Examples

automaton.in	automaton.out
3	8 4

Note

The set which is described by one of the possible automatons which Petya could buy is: “AAA”, “ABB”, “BAA”, “BAB”.

Problem E. Average Palindromes

Input file:	`palindromes.in`
Output file:	`palindromes.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes palindromes a lot. Recently he has received a string S, which consists of lowercase letters of English alphabet and question marks, as a gift from his mother. Each question mark can be replaced with any lowercase English letter with equal probability.

For each position in the string Petya wants to find out an expected value of the length of the longest odd palindrome centered at that position. A palindrome is considered to be odd if its length is an odd integer. Output an average value of expected length of the longest odd palindrome for all positions.

Input

The only line of the input contains a string S ($1 \leqslant |S| \leqslant 10^5$). This string can contain only lowercase letters of English alphabet and question marks.

Output

Output should contain one number — the average value of expected length of the longest odd palindrome for all positions. The answer will be considered correct if an absolute or relative error is not greater than 10^{-6}.

Examples

palindromes.in	palindromes.out
`ababa`	`2.600000`
`ab?bab`	`2.038462`

Note

For the first sample, expected lengths of the longest odd palindromes for each position are: 1, 3, 5, 3, 1.

For the second sample, expected lengths of the longest odd palindromes for each position are: 1, $\frac{14}{13}$, 5, $\frac{15}{13}$, 3, 1.

Problem F. Continued Fraction

Input file:	`continued.in`
Output file:	`continued.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes continued fractions a lot. A continued fraction is a fraction written in the form

$$a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_k}}}}$$

where each a_i is a non-negative integer number and k is the height of the fraction. For example, $2 + \frac{1}{1+\frac{1}{3}}$ has height 2 and is equal to $\frac{11}{4}$. The number 2 is also a continued fraction with height 0 and value 2. Moreove, $a_i > 0$ for $i > 0$ and $a_k > 1$ if $k > 0$.

With these restrictions, each rational number can be uniquely represented in the form of a continued fraction. Moreover, all irrational numbers can also be represented as continued fractions with infinite height.

Now Petya is learning how to convert a rational number into a continued fraction, though he still can't do it for numbers of the form $\frac{10^N}{10^M-1}$. Help him to find the representation of a fraction of the given form as a continued fraction.

Input

Input contains two integer numbers N and M ($1 \leqslant N \leqslant 300\,000$, $1 \leqslant M \leqslant 300\,000$).

Output

The first line of output should contain a single integer k — the height of the found continued fraction. The second line should contain $k+1$ integer numbers, separated by a single space — a_0, a_1, …, a_k.

Examples

continued.in	continued.out
1 1	1 1 9

Problem G. K-plets

Input file:	`k-plets.in`
Output file:	`k-plets.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes counting things a lot. Recently he has received a set of all integers from 1 to N, inclusive, as a gift from his mother. Now he is wondering what is the number of ways to partition this set into K-plets of numbers. A K-plet is a set of K distinct numbers in which their order is not relevant. The order of K-plets in the partition is not relevant either. However, not all partitions will make Petya happy. Some of them might scare him. The partition makes him scared if it contains at least one scary K-plet. Petya kindly wrote down all K-plets which he considers scary and it appeared that there are exactly M of them.

Help Petya to determine the number of partitions that won't scare him. Since this number can be very large, output it modulo $10^9 + 7$, otherwise it might scare Petya. Note that if N is not divisible by K then the answer is 0.

Input

The first line of input contains three integers N, M and K ($1 \leqslant N \leqslant 10^9$, $0 \leqslant M \leqslant 54$, $1 \leqslant K \leqslant 100$) — the number of elements in Petya's set, the number of scary K-plets and the amount of numbers in each K-plet. Each of the next M lines contains K numbers which represent a scary K-plet. All elements of scary K-plets are integers from 1 to N, inclusive. A scary K-plet can not contain equal elements.

Output

Output one integer — the number of partitions that won't scare Petya, modulo $10^9 + 7$.

Examples

k-plets.in	k-plets.out
9 4 3 1 2 3 1 2 4 4 5 6 7 8 9	243

Problem H. NIMG

Input file:	`nimg.in`
Output file:	`nimg.out`
Time limit:	3 seconds
Memory limit:	256 mebibytes

Little Petya likes to play with his friend Masha a lot. Recently they have learned a new game called NIMG. NIMG is played with several piles of stones between two players who alternate moves. On each turn a player picks a pile with X stones (X should not be less than the given number F) and breaks it into M piles (number M is chosen by the player from the interval $2..X$ inclusive). Note that number M is not fixed and selected by the player on each single move.

The player should split a pile in the way that the difference between produced pile with the maximum number of stones and the minimum number of stones is as small as possible. For example if the selected pile contains 8 stones and the number M picked by the player is 3, then the resulting split would be $2, 3, 3$. If the player chose M to be 4, then the split would be $2, 2, 2, 2$.

The player who can't make his or her turn loses. This happens when every pile contains less than F stones.

Petya is a gentleman and yields the right to make the first turn to Masha. Both Petya and Masha are very experienced in the game and always make their turns in the optimal way. Kids found out that the winner can be determined by looking on the initial configuration of stones. They have planned to play R games with different initial piles configurations. You have to foreseen the future and find the winner for each of those games.

Input

The first line of the input contains two integers R and F ($1 \leqslant R \leqslant 100$, $2 \leqslant F \leqslant 10^5$). Next R lines contain the description of games. Each description starts with the integer G ($1 \leqslant G \leqslant 100$) which corresponds to the number of piles at the start of the game. The rest G integers of the description are the numbers of stones in piles. Those numbers are positive and not greater than 10^5.

Output

Output one line with R numbers separated with single space. Each integer corresponds to the outcome of the respective game from the input. It should be 1 if Masha wins the game, and 0 if the winner is Petya.

Examples

nimg.in	nimg.out
4 3 1 1 1 2 1 3 1 4	0 0 1 1

Note

In the first and second game Masha loses as she can't make the first move.

In the third game Masha wins when choosing either $M = 2$ or $M = 3$.

In the last game Masha also wins regardless of the value of M.

Problem I. Semi-cool Points

Input file:	`semi-cool.in`
Output file:	`semi-cool.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes points a lot. He says that the point (x, y) is semi-cool if both x and y can be represented as $a + 0.5$, where a is an integer (possibly negative). Note that the value of a can be different for x and y.

Recently Petya has received a convex polygon with N vertices as a gift from his mother. Now he is wondering what is the number of semi-cool points that lie inside the polygon or on its border.

Input

The first line of input contains one integer N ($3 \leqslant N \leqslant 10^5$) — the number of vertices in the Petya's polygon. Next N lines contain two integer numbers each — coordinates of vertices of the polygon, given in either clockwise or counterclockwise order. All coordinates don't exceed 10^9 by absolute value. It is guaranteed that the polygon is convex and no three of its vertices lie on the same line.

Output

Output one integer — the number of semi-cool points which lie inside the polygon or on its border.

Examples

semi-cool.in	semi-cool.out
3 0 0 0 1 1 0	1

Problem J. Stairs

Input file:	`stairs.in`
Output file:	`stairs.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes stairs a lot. He has very large stairs in his garden and he likes to go up and down stairs all the time. Each morning he goes up stairs, spends there a day and comes back home for a dinner. He is getting bored of such lifestyle, so Petya decided to diversify it a bit. In order to do this, he decided to climb up stairs in the following way. At the beginning he is standing on the ground right before the first stair. Each time when he is standing on the stair number i he comes back for a few stairs (he can go down for any number of stairs between 0 and i, inclusive) and then takes off from that place and jumps to a stair number j ($j > i$). There is a restriction that $j - i$ must be from a good set. Petya wrote down his good set on the wall of his house in case he forgets it. He stops climbing after reaching the stair number N. Note that Petya can't jump higher than the stair N.

After a while he's got bored even with this new plan of going up stairs. He decided to create a plan, according to which he will choose a different way of climbing each day. The number of different ways is finite, so the plan has a finite number of items. Moreover, Petya selected one fixed ordering of ways of climbing and called it the Plan. The only thing the Plan doesn't cover is what Petya should wear on each day. On programming competitions Petya has won M T-shirts of different colors and he wants to wear one of them on each day he climbs the stairs according to the Plan.

Now Petya is wondering, how many ways are there to extend the Plan with T-shirt choices for each day. He thinks that two ways of extending the Plan are different if there exists such a way of climbing stairs that Petya would wear different T-shirts on respective day according to these two Plans. Petya doesn't care about fashion much, thus he might not wear some of T-shirts at all or even wear the same T-shirt on all days. Since the number of ways to extend the Plan with T-shirts choices can be pretty large, output it modulo 1 234 567 891. Note that if Petya can't reach the N-th stair, then the answer is 1.

Input

The first line of input contains three integer numbers N, M and K ($1 \leqslant N \leqslant 10^{18}$, $1 \leqslant M \leqslant 10^9$, $1 \leqslant K \leqslant 10$) — the number of stairs in the stairway, the number of different T-shirts and the number of elements in the good set. Next line contains elements of the good set. All of them are positive integers, not greater than 10. All elements of the good set are distinct.

Output

Output one integer — the number of ways to extend the Plan with T-shirts choices that Petya can make modulo 1 234 567 891.

Examples

stairs.in	stairs.out
3 2 1 1	64
10 2 2 1 3	1216288599
26 13 3 1 5 7	346505065

Note

In the first sample from the stair i Petya can move only to the stair $i + 1$. Here is the list of all of his possible movements from each stair:

- From the ground he can only jump to the stair 1.
- From the stair 1 he can either directly jump to the stair 2 or come back to the ground and then jump to the stair 2.
- From the second stair he can either return to ground and then jump to the stair 3 or come back to the stair 1 and then jump to the stair 3 or even jump there directly from the stair 2.

Thus the total number of items in the Plan is equal to 6. He has 2 T-shirts, which means that there are $2^6 = 64$ different ways of extending the Plan with T-shirt choices.

Problem K. Number of Zeroes

Input file:	`zeroes.in`
Output file:	`zeroes.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Little Petya likes to draw on the walls a lot. He spent a lot of time, writing down all the integers from 1 to 10^k, inclusive, in a row. After he had finished, he decided to show this masterpiece to his friend Masha. The only thing she wondered was the number of zeroes written on the wall. Help Petya to answer her question. Since this number can be very large, output it modulo p, otherwise the children will be scared.

Input

The first line of input contains two integers k and p ($1 \leqslant k \leqslant 10^{18}$, $1 \leqslant p \leqslant 10^9$).

Output

Output one integer — the number of zeroes written on the wall.

Examples

zeroes.in	zeroes.out
1 1000	1
3 1000	192