Problem A. Quiz Game

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

You are playing a Quiz Game against your friend Nuradil. At the moment, you have X points, and Nuradil has Y points. Now it is time for the final question. Anyone who answers the final question correctly will get F points, independently on the choice of the other player. This means that either both of you, none of you, or one of you could get F points added to your current score.

The final question has n options, and they are indexed from 1 to n. **Only one** of these options is correct. You are also given two arrays of integers, A and B, each of size n.

For each i from 1 to n, the probability that the ith option is correct is equal to $A_i/10^5$. For example, if $A_1 = 24000$ then the probability that the ith option is correct is equal to $24000/100000$ which is 0.24.

Similarly, for each i from 1 to n, the probability that Nuradil will choose the ith option is equal to $B_i/10^5$.

Your task is to find an option that will maximize the probability that you will win. You will win only if your final score is **strictly greater** than the final score of Nuradil. If there are multiple options that give the highest winning probability, find the option with the smallest index among them.

Input

The first line of the input contains four integers $X, Y, F(0 \le X, Y, F \le 10^9)$, and $n(2 \le n \le 10^5)$.

The second line of the input contains n space separated integers $A_1, A_2, ..., A_n(0 \le A_i \le 10^5)$, such that the probability that option i is correct is equal to $A_i/10^5$.

The third line of the input contains n space separated integers $B_1, B_2, ..., B_n(0 \le B_i \le 10^5)$, such that the probability that option i will be chosen by Nuradil is equal to $B_i/10^5$.

It is guaranteed that $\sum_{i=1}^{n} A_i = \sum_{i=1}^{n} B_i = 10^5$.

Output

Print a single line with an integer which is the index of the option that maximizes your probability for winning. If there are multiple options, print the smallest index among them.

Example

standard input	standard output

`5 5 1 3`
`10000 30000 60000`
`70000 20000 10000`
	`3`

Problem B. Offline Spring!!!

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

As with the current pandemic, everything is going online. NU was no exception, however, most students are ready to take the risk and demanding to make the Spring semester offline again. But here comes the thing; the most powerful Katsu of all-time at NU, Shigeo, is about to make a final decision to make the Spring semester online with a single email...

Now, desperate students decided to ask the most powerful Zhaxybay of NU, ADJA, for help to hack Shigeo's mail to make a fake announcement about the offline Spring semester. ADJA brushing up his knowledge from the hardest courses taken at NU, he almost comes up with the password of Shigeo's mail. The things he found so far are:

*) De-hashing of a password requires lots of terms about prime properties (a prime number is a positive number that is divisible only by 1 and by itself, note that 1 is not considered as a prime number)

*) A *primee number* - is a number that consists only of prime digits

*) A *primeee number* - is a *primee* number that has prime sum of digits

*) A $mega - primeee$ *number* - it is a *primeee* number with prime length of the Longest **Strictly** Increasing Subsequence (LSIS) of its digits

*) A Shigeo *number* - it is a $mega - primeee$ number on the interval $[L, R]$

There is not so much time left, so ADJA needs your help as well. To hack the mail, ADJA needs to find the number of Shigeo numbers modulo 1000000007 ($10^9 + 7$). Can you help him and the students of NU to make the Spring semester offline again?

Input

In the first and only line you are given two positive integers L and R, ($1 \le L \le R < 10^{100}$).

Output

Print the number of Shigeo numbers modulo ($10^9 + 7$).

Examples

standard input	standard output
1 30	2
24 50	1

Note

The first few $mega - primeee$ numbers: 23, 25, 223, 227...

Problem C. Largest subarray with max median

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

Given an array A, you have to find its subarray that satisfies the following properties:

1. The median value of that subarray must be the largest possible among all subarrays of A.
2. Among all subarrays satisfying the condition 1, you have to pick the longest.
3. If there are still several options, you can select any subarray.

A median of an array is calculated in the following way:

1. Sort the array
2. If array has odd length, then the element on position $(size + 1)/2$ is the median (1-based indexing)
3. Otherwise the element on position $size/2$ is the median (1-based indexing)

For example for the array $[2, 4, 6]$, the median is 4, and for the array $[1, 3, 5, 7]$, the median is 3.

Input

The first line of the input contains a single integer N ($1 \leq N \leq 5{\cdot}10^5$) - the number of elements in the array. The second line contains N integers separated with spaces - elements of the array ($-10^9 \leq A[i] \leq 10^9$).

Output

Output two integers — starting and ending indices of the subarray (l and r). Both bounds must be inclusive.

Examples

standard input	standard output
2 1 1	1 2
5 8 20 16 20 8	2 4

Note

For this problem we consider continuous subarrays, i.e. for a given array A its subarray $[l..r]$ includes all elements that have indices between l and r inclusive.

Problem D. Volleyball!!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Team “KarasuNU” is training for the local volleyball tournament. Hafa, the smartest team member, wants to know the probability of becoming a champion!

There are n teams that will participate in the upcoming tournament. Teams are numbered from 1 to n. By magic coincidence, n is a power of two.

The tournament consists of several stages. Initially, some ordering of teams $t_1, t_2, ...t_n$ is chosen uniformly at random among all possible orderings. Then, at each stage, the following happens:

1. Teams t_1 and t_2 play a match against each other, teams t_3 and t_4 play a match against each other and so on.
2. The losers of the matches are eliminated from the tournament and removed from t. Remaining teams in t are kept in the same order.
3. If only one team is left (size of t equals 1) then the team is called a *champion* and the tournament ends. Otherwise, a new stage is started.

For each pair of teams i and j, Hafa knows $p_{i,j}$ — the percentage probability of the i-th team winning the j-th team in a match. For each team i, he wants to know the probability of i becoming a *champion*.

After thinking for six microseconds, Hafa found the answers and went back to training his quick attack with Timka. But can you find the desired probabilities?

Let the answer for some team i be representable in the form of an irreducible fraction $\frac{P_i}{Q_i}$. You should find the value of $P_i \cdot Q_i^{-1} \mod 10^9 + 7$ for each team i.

Input

The first line contains an integer n ($2 \le n \le 16$, n is a power of two) — the number of teams in the tournament.

Each of the next n lines contains n integers. The j-th number in the i-th line is equal to $p_{i,j}$ — the percentage probability of the i-th team winning the j-th team in a match. It is guaranteed that $p_{i,j} + p_{j,i} = 100$ ($i \neq j$) and $p_{i,i} = 0$.

Output

Output n integers - the probabilities of becoming a *champion* for each team. The values should be taken modulo $10^9 + 7$.

Examples

standard input	standard output

2
0 100
0 0
	1
	0

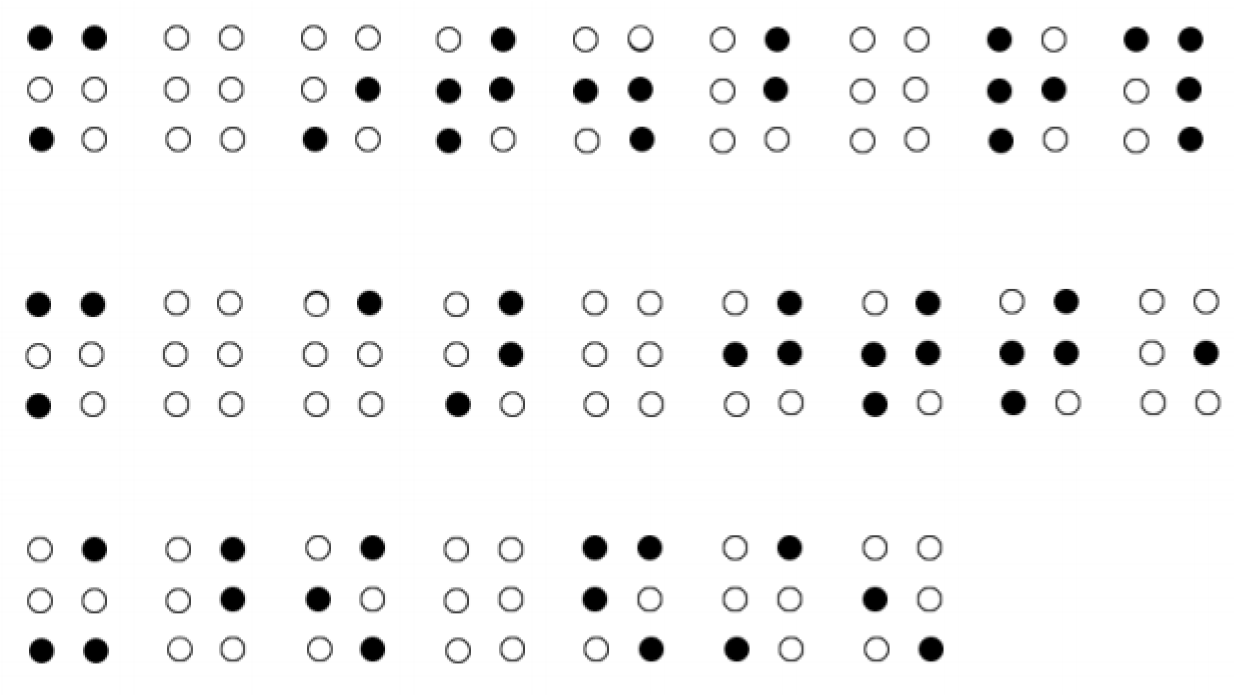
2
0 50
50 0
	500000004
	500000004

4
0 45 25 11
55 0 37 51
75 63 0 100
89 49 0 0
	751476672
	256882002
	337850003
	653791338

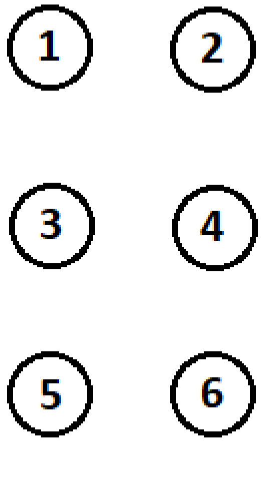
Problem E. Dulat's Braille

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

Dulat came up with his own Braille system for the English language. In his Braille system, every character in English can be represented as a **block of 6 dots** as in usual Braille systems. Every dot in a block can either be **raised or flat**. Next, Dulat has written down 3 sentences in English. After that, he transliterated the sentences into his Braille system. Here is what he got (each line represents a separate sentence):



Black represents raised dots, while white represents flat ones. The dots are numbered from 1 to 6 as shown below.



Given a sentence in English in the input, output its Braille representation using Dulat's system. The length of a string is less than 1000.

Input

Input consists of a single line containing a sentence in English where each character is either a letter from an English alphabet (lowercase or uppercase) or a space.

For example:

Hello people

Output

Print the transliterated sentence in Braille code where each Braille character is separated by space. Please, read the full output format explanation below.

After you have obtained a sentence in Braille code, convert each block of it to a string of length 6. If the *ith* dot in a block is raised, then the *ith* character of a string should be 1, otherwise it is 0.

For example, the first of the three sentences shown above in Braille code should be printed as below:

110010 000000 000110 011110 001101 010100 000000 101110 110101

Problem F. Among Us 2

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

Crewmates have eliminated all the impostors. Now, they are having a victory party in the cafeteria. But, it turns out that exactly one of them is infected by the new virus called ILOVENU. Who is it?

There are n crewmates at the party and **exactly** one of them is infected. Also, k of them are terrible liars who lie all the time, while the rest always tell the truth. All the crewmates know who are liars and who is "the infected" (they are just not telling you).

Each crewmate told you a statement of one of the following types:

- $+$ x — "Crewmate x is infected".
- - x — "Crewmate x is not infected".
- l x — "Crewmate x is a liar".
- t x — "Crewmate x is not a liar".

Given this information, you want to determine which of the crewmates can be "the infected".

Input

The first line contains two integers n and k ($1 \le n \le 5000$, $0 \le k \le n$) — the number of crewmates and the number of liars, respectively.

Each of the next n lines contains a character c_i ($c_i \in \{`+', `-', `l', `t'\}$) and an integer x_i ($1 \le x_i \le n$) — the statement of the i-th crewmate.

Output

If the given information is contradictory, output -1.

Otherwise, output an integer s — the number of crewmates who can be "the infected". Then, output s integers — indices of the crewmates in ascending order.

Examples

standard input	standard output
5 2 + 3 l 4 l 1 - 5 l 1	4 1 2 4 5
2 2 t 2 l 1	-1
3 0 - 2 - 3 - 1	-1
2 0 t 2 t 1	2 1 2

Note

In the second example, both of the crewmates are liars. Then, the statement of the second crewmate is contradictory.

In the third example, there are no liars. Then, according to the statements of the crewmates, no one is infected. But, it contradicts the fact that exactly one of the crewmates is infected.

The visual representation of the fourth example:

Problem G. nuope?

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

You are given a 1-indexed string s of size n. It consists of letters 'n', 'u', 'o', 'p' and 'e'. You should find the number of pairs i and j, such that:

- $j > 0$
- $1 \leq i - j$
- $i + j \leq n$
- $s_{i-j} = s_i = s_{i+j}$

Input

The first line contains an integer n ($1 \leq n \leq 200000$) — the size of s.

The second line contains n characters — the string s. It is guaranteed that s only consists of letters 'n', 'u', 'o', 'p' and 'e'.

Output

Output a single integer — the number of pairs described above.

Examples

standard input	standard output
6 nuopen	0
3 nnn	1
11 ononononono	10

Problem H. Anuar and Ogres

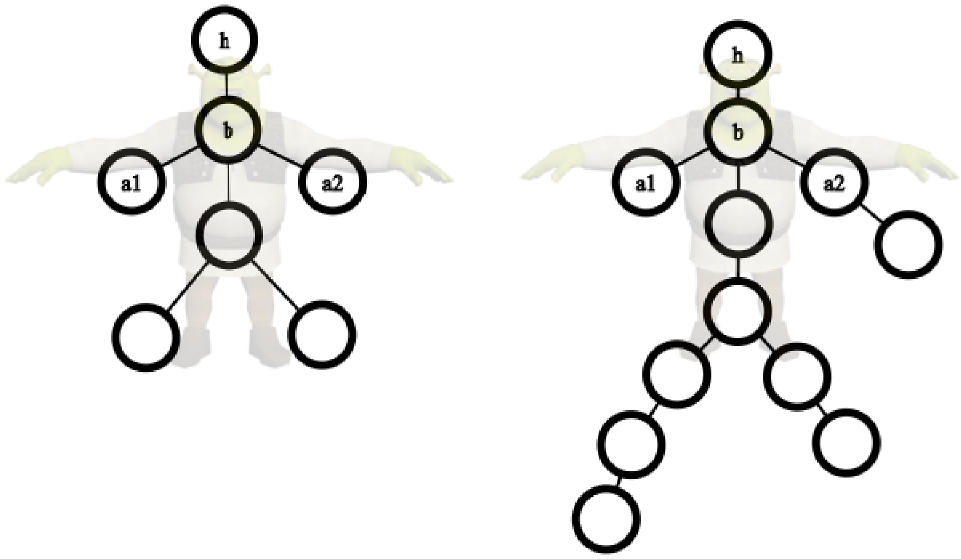
Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Taldykorgan Reverse Engineering Ogre University (TREO University) is organizing TREO University Open — an annual competition in finding ogres in a tree. Anuar, the best student of TREO University, wants to find all the ogres and win the main prize — a year's supply of bread from Free Flow! Can you help him?

A rooted tree is called a *rooted path* if each non-leaf vertex of the tree has exactly one child.

A rooted tree is called an *ogre* if the following holds:

- The root of the tree h has exactly one child b.
- b has exactly 3 children, two of which can be selected as vertices a_1 and a_2.
- The subtrees of a_1 and a_2 are rooted paths.
- The subtree of the remaining child of b is constructed in the following way: Initially, it is a rooted path p. Then, another two rooted paths are connected to p's only leaf as subtrees.



Some examples of ogres.

You are given a rooted tree with n vertices. Let $F(v)$ be the number of ways to delete vertices in the subtree of v, such that the subtree becomes an ogre (and the tree is still connected). Two ways are considered different if they end up with different subtrees.

You should find the value of $\sum_{i=1}^{n} F(i)$ modulo $10^9 + 7$.

Input

The first line contains an integer n ($2 \le n \le 500000$) — the number of vertices in a tree.

The second line contains integers $p_2, p_3, ...p_n$. Here, the vertex p_i ($2 \le i \le n$, $1 \le p_i < i$) is the parent of the vertex i.

Vertex 1 is the root of the tree.

Output

Output a single integer — the value of $\sum_{i=1}^{n} F(i)$ modulo $10^9 + 7$.

Examples

standard input	standard output
7 1 2 2 2 5 5	1
13 1 1 2 2 2 6 6 6 6 10 10 12	23

Problem I. Permutation Game

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

N people are playing a game, consisting of multiple rounds. For each round, a participant gets some integer number of points between 0 and $N-1$, and all participants get different amount of points within one round. Game ends when at least one of the players will have total points $\geq X$. It is possible that several players can score $\geq X$ points at the same time.

What is the minimum and maximum number of possible rounds in this game?

Input

One and only line contains two integers N and X ($2 \leq N, X \leq 10^9$).

Output

Print the minimum and maximum number.

Examples

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

Problem J. Ternary array

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

Given an array, you need to find the minimum number of swaps of two **adjacent** elements in order to make the array “ternary“.

An array is considered to be “ternary“ if it is non-decreasing until some point, then it is non-increasing until the end, e.g. [1, 2, 3, 2, 2, 1, 0], [1, 2, 2, 2], [3, 2, 1].

More formally, a 1-indexed array is called “ternary“ if and only if there exists an integer i, such that $1 \leq i \leq n$ and the following holds:

- for all $1 < j \leq i$ we must have $a_{j-1} \leq a_j$
- for all $i \leq k < n$ we must have $a_{k+1} \leq a_k$

Input

The first line has an integer N $(1 \leq N \leq 3 \cdot 10^5)$.

The second line has N integers, $(1 \leq A_i \leq N)$.

Output

Print the minimum number of swaps needed to make the initial array “ternary“.

Examples

standard input	standard output
6 1 3 2 3 2 4	2
5 1 2 3 2 2	0
5 3 1 2 2 2	1

Note

In the first testcase it is enough to swap elements at positions $(2, 3)$ and $(5, 6)$.

In the second testcase the array is already ternary.

In the third testcase it is enough to swap the first and second elements.

Problem K. Vaccines

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

It is the year 3030, we are still at our homes, sadly, in quarantine. Google engineers still Working From Home although most companies work in the office with safety measures. Kazakhstani medical workers and researchers are the only ones left who have a chance to save people on this planet from COVID-19. To develop the best vaccine with 100% certainty, vaccine developers worked very hard for a century. After a century of researches, Kazakhstani researchers found perfect requirements for the best vaccines.

They found out n types of infections and identified the constants x, y, z. Each infection appears to have 2 properties a_i and b_i. Research Assistants at Toqayev University (imaginary university founded in 3010) have been working hard to find a perfect vaccine.

Finally, in the year 3030, they found that each vaccine can be **uniquely** characterized by an integer number between x and y. It turns out that a vaccine characterized by the integer V works effectively against the infection i if and only if $V \ mod \ z^{a_i} = b_i$. A vaccine is considered to be the "best"if it is effective against **all** n infection types.

Imaginary researchers of imaginary Toqayev University are asking you, the real student, for help. Can you help them to find the number of the "best"vaccines (the number of the "best"characteristics)?

Input

In the first line, you are given three non-negative integers x, y, z ($1 \le x \le y \le 10^{18}$, $2 \le z \le 50$).

The second line contains one single integer n ($1 \le n \le 10^5$).

The third line contains n integers, elements of the array a ($1 \le a_i \le 100$).

In the fourth line, you are given n elements of the array b ($0 \le b_i \le 10^{18}$, $b_i < z^{a_i}$).

Output

Print your answer to the given problem

Example

standard input	standard output

10 41 3
3
1 3 2
1 13 4
	2

Note

Integers 13 and 40 are the "best"vaccine characteristics in the sample testcase

Problem L. NU Market

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

A brand new market just opened near NU in Nur-Sultan which offers all sorts of groceries for a cheap price. Since students at NU live by the famous Steve Jobs quote "Stay hungry stay foolish they are all pretty hungry now. So they all decided to immediately go and buy some groceries in the new market.

The market can be represented as a two-dimensional grid infinite in all four directions (because the market is really big this is a proper approximation).

Initially, there are n students shopping around the market. Each student's initial position can be described by two integers:

x - student's initial position's horizontal coordinate in meters.

y - student's initial position's vertical coordinate in meters.

It is **guaranteed** that no two student have the same initial position.

In a moment, each student will start moving with a speed of **1 meters per second** in the provided direction d. d is one of $'n',' s',' e',' w'$. Here:

$'n'$ (north) means that the student will move in the positive y-axis direction.

$'s'$ (south) means that the student will move in the negative y-axis direction.

$'e'$ (east) means that the student will move in the positive x-axis direction.

$'w'$ (west) means that the student will move in the negative x-axis direction.

If two or more students meet at the same point at the same time they collide with each other and **all of them** fall down. **Fallen students can no longer collide with any other student.** Given the initial positions, and the moving direction for each student, find out how many students will remain unfallen after all collisions have occured.

Input

The first line of input contains a single integer $n(1 \leq n \leq 1000)$ denoting the number of students. n lines follow, the ith of which contains two space-separated integers $x, y(-10^9 \leq x, y \leq 10^9)$, followed by a single character $d(d \in \{'n',' s',' e',' w'\})$ **(without quotes)** which describe the ith student.

Output

Print a single integer equal to the number of students who remain unfallen after all collisions have occured, i.e. students who will not collide with any other student.

Example

standard input	standard output

`6`
`2 2 n`
`2 7 s`
`4 4 w`
`4 9 e`
`8 9 w`
`6 7 n`
	`1`

Note

Explanation for the sample test case: The 1st and the 3rd student collide and fall down. The 4th, 5th, 6th students all collide at the same point and all fall down. Hence, the only student left unfallen is the second one. So the answer is 1.