A: Alternative permutations

Memory Limit : **256 MB**

Johnny wrote a permutation to binary search trees (BST) converter: given a permutation $(\pi_1, \pi_2, \ldots, \pi_n)$ it assigns π_1 to the root of the BST, from numbers in $(\pi_2, \pi_3, \ldots, \pi_n)$ smaller than π_1 (in the same order!) it recursively creates a BST and attaches it as a left subtree of the root; symmetrically, from numbers in $(\pi_2, \pi_3, \ldots, \pi_n)$ larger than π_1 it also creates a BST and attaches it as a right subtree of th root.

To Johnny's surprise, it turns out that different permutations can result in the same BST – for instance the permutations $(2, 3, 1)$ and $(2, 1, 3)$ result in the same BST. He found this fact astonishing and immediately defined Johnny's Numbers J_k: the k-th Johnny's Number is the smallest n such that there is a BST on n nodes labelled with numbers $1, 2, \ldots, n$, that can be obtained from exactly k different permutations of the numbers $1, 2, \ldots, n$. The investigation of Johnny's Numbers is difficult and their popularity is decreasing. Help Johnny out–compute Johnny's Number J_k for the given k.

Input

The first and only line of the input contains a single natural number k $(1 \leqslant k \leqslant 10^{11})$.

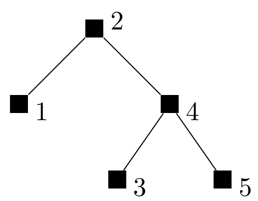
Output

You should write a single natural number in the first and only line of the output: k-th Johnny's Number J_k, assuming that it exists and it is at most 5 000. Otherwise, that is if J_k does not exists or it is larger than 5 000, you should write the word `NIE` (Polish for 'no').

Example

Input	Output
8	5 2 1 4 3 5

The tree having exactly eight generating permutations is shown below:



All permutations generating that tree are: $(2, 1, 4, 3, 5)$, $(2, 1, 4, 5, 3)$, $(2, 4, 1, 3, 5)$, $(2, 4, 1, 5, 3)$, $(2, 4, 3, 1, 5)$, $(2, 4, 3, 5, 1)$, $(2, 4, 5, 1, 3)$, $(2, 4, 5, 3, 1)$.

B: Cartoons

Memory Limit : **256 MB**

Sophie's parents made a DVD with episodes of her favourite cartoon. When she wants to watch it, they play her an interval of episodes, that is, a sequence of episodes that are consecutive on the DVD. Unfortunately, they were a little careless when making the DVD: some of the episodes repeat, to Sophie's dislike. An interval of episodes is interesting (to Sophie) if at least one episode in it is different than all other. Moreover, Sophie sometimes misses the episodes at the beginning of the interval (as she wants to play with toys a little more) and sometimes she does not watch the whole interval and misses some episodes at the end. Thus an interval of episodes is very interesting, if each of its subinterval of episodes is interesting.

Sophie's parents wonder, which of the intervals of the episodes on the DVD are very interesting? Help them out–given the full interval of episodes on the DVD compute, how many of its subintervals are very interesting?

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 500\,000$) which is the number of episodes on the DVD. The second and last line contains n natural numbers each separated by a single space, the i-th of them is the episode number a_i of the i-th episode ($1 \leqslant a_i \leqslant 10^9$).

Output

You should write a single natural number in the first and only line of the output: the number of very interesting intervals of the sequence of cartoons given in the input.

Example

Input	Output
5 1 6 1 6 6	10

The following sequences are interesting: all length-1 sequences, three sequences of length 2: only (6, 6) is not interesting, all three length-3 sequences, one length-4 sequence–(6, 1, 6, 6). Among them the sequences (1, 6, 6) and (6, 1, 6, 6) are not very interesting.

Input	Output
6 1 2 3 1 2 3	20

Only the full sequence $(1, 2, 3, 1, 2, 3)$ is not very interesting.

C: Identifier sequence

Memory Limit : **16 MB**

Johnny is a hacker and recently he came into possession of a database of a large bank. The most important task at hand is to extract the clients' IDs from it. They are sorted and concatenated into a single sequence of digits, without the marks where the concatenation took place; these marks are in a separate file, which Johnny does not have. Johnny wants to retrieve the IDs, that is, compute how many IDs can be obtained by cutting the sequence into pieces. Johnny believes in his luck: if the there are many ways to cut the sequence, surely the one that maximises the number of IDs is the correct one.

As the clients' IDs were assigned at different stages, they can be of different lengths and can have leading zeroes. But Johnny knows that each ID has a unique numerical value, when treated as number written in base-10 positional notation.

Input

The first and only line of the input contains a sequence on n ($1 \leqslant n \leqslant 100\,000$) base-10 digits. There are no other characters between those digits, in particular, no spaces. This is the sequence of IDs, which Johnny has.

Output

You should write one natural number in the first and only line of the output: the maximal number of clients' IDs into which the sequence can be cut.

Example

Input	Output
10540910	4

The sequence of clients' IDs corresponding to the answer is : 1, 05, 40, 910. Note the ID 05 has a numerical value 5.

D: Daycare children

Memory Limit : **64 MB**

Sophie works at a daycare. There are n children in her group, each one should be given a set of at least k toys of different kinds–the children do not have preferences of toy kinds nor of toys of the same kind. Toys come in $\left\lfloor \frac{3k^2}{2} \right\rfloor$ different kinds and Sophie has access to unlimited supply of toys of each kind. Children like to play in pairs and for a pair of children to be able to play together, there has to be exactly one kind of toys such that both of them have a toy of this kind; otherwise either they have different kinds of toys and it is hard for them to play or they have a choice and they feel confused. Moreover, each child wants to be special and so no two children can have the same set of toy kinds. Help Sophie in her work: write a program which computes for each child its set of toy kinds, so that each pair of children can play together.

Input

The first and only line of the input contains two natural numbers n and k ($1 \leqslant n \leqslant \binom{k}{2}$, $2 \leqslant k \leqslant 50$), separated by a single space.

Output

You should write n lines to the output. The i-th line should begin with a natural number k_i: the number of toys that the i-th child gets, a single space and then a sequence of k_i pairwise different toy kinds–that is, natural numbers from the set $\{1, 2, \ldots, \lfloor \frac{3k^2}{2} \rfloor\}$, each separated by a single space.

Example

Input	Output
3 3	4 10 1 2 13 3 1 3 4 6 1 5 6 7 8 9

In this example there are three children, each should be given at least three toys out of total $\left\lfloor \frac{3 \cdot 3^2}{2} \right\rfloor = 13$ toy kinds: $1, 2, \ldots, 13$. In the given solution each pair of children has a toy of kind 1 in common (and no other).

Input	Output
5 4	4 1 2 3 13 4 1 4 7 10 4 4 5 6 13 4 7 8 9 13 4 10 11 12 13

In this example there are five children and each is to be given at least four different toy kinds, the toys come in $\left\lfloor \frac{3 \cdot 4^2}{2} \right\rfloor = 24$ kinds: $1, 2, \ldots, 24$. In the given solution the pairs of children that do not include the second child have the toy 13 in common, the pairs of the second child and, respectively, first, third, fourth, and fifth child have a common toy of the kind: $1, 4, 7, 10$, respectively.

E: English

Memory Limit : **256 MB**

Sophie wants to learn the English letters and she asked Johnny for help. He wants to prepare a set of words so that each letter appears in one of them and as Johnny does not like to repeat himself, each letter should appear in exactly one word (exactly once). Sophie does not fully trust Johnny–he pulled her leg too many times before–so she wants to verify that the words are in her English dictionary. Unfortunately, she spilled some tea over it and all one- and two-letter words are now unreadable and all other words are unreadable with a probability 1/2. Help Johnny–write a program that reads the list of readable words from the dictionary and computes the set of words according to the requirements. Luckily, Johnny has the same dictionary as Sophie (also with no one- nor two-letter words, but with all other words readable), he can share it with you so that you can prepare a bit. Furthermore Sophie personally guarantees that you can choose the requested set of words from her dictionary.

Input

The first line of the input contains a single natural number n of words in the dictionary ($1 \leqslant n \leqslant 20\,000$). The following n lines contain lexicographically-sorted words of length at least 3, one word per line. Each test file was randomly generated from a dictionary that contains 20 000 words (each of length at least 3), i.e. each word was put in the file independently with a probability 1/2.

Output

You should write a single natural number k ($1 \leqslant k \leqslant 8$) in the first line of the output. In the following k lines you should write k words, one per line. Each of those words should appear in the input dictionary and each letter of the English alphabet, which contains 26 letters in total, should appear exactly once in exactly one of those words. It is guaranteed that for each input file there is a solution.

Example

Input	Output
20 biz bur doughty faq fwd hex jane job kings kpx lexmark nyt plz pvc quiz rfc sgh sql toy wmd	7 nyt lexmark sgh quiz pvc job fwd

Due to space constraints, the dictionary in the example was not created randomly and it is presented only to demonstrate the constraints on the solution. In particular, it will not be used for testing the correctness of your solution. Nevertheless, the seven words in the output contain all 26 letters of the English alphabet, each exactly once.

F: Encryption function

Memory Limit : **64 MB**

After her computer class Sophie invented her own encrypting function, which takes a number as an input. Given a number it treats it as a sequence of base-10 digits (with no leading zeroes), masks out every possible subset of positions in this sequence, interprets the new sequence as a base-10 number (possibly with leading zeroes) and adds all numbers obtained in such a way. So far Sophie failed to devise a decryption algorithm. Help her–write a program that decrypts the encrypted number.

Input

The first and only line of the input contains a single natural number n ($1 \leqslant n \leqslant 10^{18}$), this is the output of Sophie's encryption function.

Output

In the first and only line of the output you should write a single natural number m, for which the encrypted value is n, or `NIE` (Polish for 'no') if no such a number exists.

If there are several correct answers, you can output any of them.

Example

Input	Output
177	123

Computing the value of the encryption function on 123 gives $1 + 2 + 3 + 12 + 13 + 23 + 123 = 177$.

Input	Output
42	NIE

There is no sequence whose encrypted value is 42.

G: Printer's head

Memory Limit : **64 MB**

Johnny bought a 3D printer. He wants to test it on a simple task: print n cuboids with equal square bases and heights $1, 2, \ldots, n$ in the given order. The printer works in left-to-right and right-to-left sweeps, the sweeps can be mixed arbitrarily, i.e. two left-to-right sweeps can be executed one after another, similarly right-to-left. In one sweep the printer can stop over an arbitrary number of fields and print a cuboid on each of them; the first printed cuboid is of a given height and each next one is 1 lower (the head of the printer cools down). The printer cannot print on fields on which something was printed already.

Sweeps cost money. Help Johnny minimise the number of sweeps utilised for the task.

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 1\,000\,000$), this is the number of cuboids to be printed. The second and last line contains a sequence of n pairwise distinct natural numbers from the set a_i ($1 \leqslant a_i \leqslant n$), each separated by a single space. These are the heights of the consecutive cuboids to be printed.

Output

You should write in the first and only line of the output a single natural number: the minimal number of sweeps needed to print the given sequence of cuboids.

Example

Input	Output
6 3 2 4 1 5 6	2

Johnny can print cuboids of heights 6, 5, 4, 3 in one right-to-left sweep, and the 2, 1 in a left-to-right sweep.

Input	Output
8 8 7 4 1 5 2 3 6	3

Johnny can print cuboids of heights 8, 7, 6 in a left-to-right sweep, then 5, 4 in right-to-left sweep and finally 3, 2, 1 also in a right-to-left sweep.

H: Passwords

Memory Limit : **64 MB**

Johnny is obsessed with computer security: he has a different password for each website, he destroys the printouts, etc. And this is his demise: he realised that he accidentally put the sheet with his passwords to the paper shredder! But what are the odds, this sheet of paper was shredded so that each piece of paper corresponds to one column of text. Moreover, Johnny knows for sure that all passwords consist only of capital letters of English alphabet, they are pairwise different, they all have the same length and they were written down in the lexicographic order. Johnny numbered the columns and put them side by side but he is not sure whether the order he came up with is correct. Help Johnny–write a program that computes how to permute the columns of the text so that words written in rows are lexicographically ordered. If this is possible for many different permutations, choose the one which is lexicographically smallest.

Input

The first line of the input contains two natural numbers n, m ($1 \leqslant n \cdot m \leqslant 1\,000\,000$), separated by a single space. The following n lines contain n words, one per line. Each one of them consists of m capital letters of the English alphabet.

Output

You should write m natural numbers in the first and only line of the output: the permutation of the columns after which the words in rows are sorted lexicographically. If there are many such permutations, you should write the one that is lexicographically smallest among them. If there is no such a permutation, you should write `NIE` (Polish for 'no') instead.

Example

Input	Output
2 5 TOMEK KASIA	3 1 2 4 5

After permuting the columns in the described way we obtain words: `MTOEK`, `SKAIA`, which are lexicographically sorted.

Input	Output
3 3 CAB CBA BAC	NIE

In this example there is no way to permute the columns so that the words obtained in the rows are lexicographically sorted.

I: Laser intensification

Memory Limit : **64 MB**

Sophie works at a laser factory, which is now introducing a new laser intensifier. This device is a grid, each node of this grid receives and emits photons: for each photon received from the left or down it emits one photon up and one to the right, those photons will be received by nodes in appropriate directions (or are lost if there is no node in this direction). Unfortunately, the industrial scale production is far from being perfect and some of the nodes in the grid are faulty: they neither receive nor emit photons. To be more precise: we know that some nodes are for sure faulty and each other is faulty with the probability $1 - p$. Luckily, the exact value of p can be altered by appropriate changes of the pressure during the production. Sophie's task is to compute the value of p such that a single photon emitted to the lower-left node of the grid will, in expectation, yield k photons that are received at the upper-right corner of the grid or establishing that this is not possible. Help her in this task.

Note: if the node in the upper-right corner is faulty then it does not receive any photons.

Input

The first line of the input contains four natural numbers w, h, n, k, $(1 \leqslant w, h \leqslant 5\,000,\ 0 \leqslant n \leqslant 50,\ 1 \leqslant k \leqslant 10^{10000})$, each separated by a single space, denoting: grids dimensions (width and height), the number of faulty nodes and the number of photons that should be received at the node $(w-1, h-1)$. The following n lines contain the descriptions of faulty nodes, one per each line. In each of those lines there are two natural numbers x, y $(0 \leqslant x < w,\ 0 \leqslant y < h)$, separated by a single space; these are the coordinates of the faulty node. All those nodes are pairwise distinct.

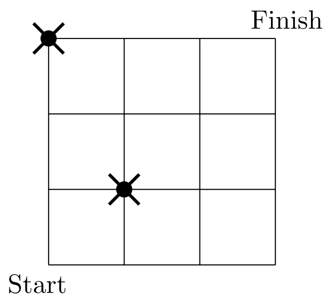
Output

You should write one real number in the first and only line of the output: the requested probability p . The answer is accepted if such a probability exists and the absolute or relative error is at most 10^{-6}. If such a probability does not exist, you should write a word `NIE` (Polish for 'no'). The answer `NIE` is accepted if indeed such a probability does not exist.

Example

Input	Output
4 4 2 5 0 3 1 1	0.953069489

The situation looks like on picture below: photon starts in grid node marked Start while photon sensor is in grid node marked Finish.



Input	Output
3 4 1 10 0 1	NIE

Even for $p = 1$ only 4 photons reach the upper-right corner.

J: Autumn cleaning

Memory Limit : **16 MB**

Autumn is coming and Sophie wants to prepare for it by emptying her grandparents' basement. She wants to sell unused items and she already put a price tag (non-negotiable!) on all of them and put the offer on the web; note that some items may have the same price. She was quickly contacted by a junk dealer, who wants to buy exactly k items, no matter which (junk is junk). Unfortunately, he visited the ATM only a moment ago and he has only a lot of r-złoty bills ('złoty' is the Polish currency). In how many different ways they can make a trade?

Input

The first line of the input contains three natural numbers n, k and r ($1 \leqslant n \leqslant 1\,000\,000$, $1 \leqslant k \leqslant 3\,000$, $1 \leqslant r \leqslant 10$), each separated by a single space, denoting: the number of items Sophie wants to sell, the number of items that the junk dealer wants to buy and the face value of the bills he has. The second and last line of the input contains n natural numbers a_i ($1 \leqslant a_i \leqslant 1\,000\,000$), each separated by a single space. These are the prices of the items that Sophie wants to sell.

Output

You should write a single natural number in the first and only line of the output: remainder modulo $10^6 + 3$ of the number of sets of k items whose total price is divisible by r.

Example

Input	Output
5 3 4 8 1 2 2 3	2

The junk dealer can buy first, second and fifth item, with a total price $8 + 1 + 3 = 12$ or first, third and fourth, paying $8 + 2 + 2 = 12$.

K: Chip cards

Memory Limit : **16 MB**

Johnny designs chips for chipcards. There are n ports at the upper and bottom boundary of such a chip. The ports at the upper boundary are labelled from left-to-right by consecutive natural numbers $1, 2, \ldots, n$; the ports at the bottom boundary are also labelled from left-to-right with the same numbers, but in a possibly other order. A pair of ports labelled with the same number is connected by a straight-line path; the paths are grouped into layers and the paths in one layer cannot cross. Furthermore, ports at both boundaries are grouped into sockets: each socket consists of consecutive ports and each port belongs to exactly one socket. Note that the sockets at the bottom boundary may be of different lengths than those at the upper boundary. Lastly, a socket can be switched, in which case the order of ports in it is reversed.

Johnny learned through the grapevine that he is to be fired. He decided to have a little revenge–he is going to design the next chip so that it is as expensive in production, as possible. The key production cost is the number of layers, so he will switch each socket in a way to maximise the number of needed layers. This turned out to be much more difficult than his usual tasks and Johnny seeks your help.

Input

The first line of the input contains a natural number n ($1 \leqslant n \leqslant 5\,000$) that denotes the number of ports on each boundary of the chip. The second line contains n different natural numbers from the set $\{1, 2, \ldots, n\}$, each separated by a single space; these are the labels of ports on the bottom boundary (as a reminder: the labels on the consecutive ports on the upper boundary are $1, 2, \ldots, n$). The third row contains a natural number k ($1 \leqslant k \leqslant n$) denoting the number of sockets on the upper boundary, a single space, and then k numbers denoting the amount of ports in the consecutive sockets of the upper boundary. Each socket has at least one port, the sum of sockets' numbers of ports on the upper boundary is equal to n. The fourth row contains a similar description of sockets on the bottom boundary.

Output

You should write a single natural number in the first and only line of the output: the maximal possible number of needed layers.

Example

Input	Output
8 3 5 1 2 8 4 7 6 3 3 1 4 4 2 2 2 2	4

On the upper boundary the first socket contains ports $1, 2$ and 3, the second: port 4, and the third: $5, 6, 7$ and 8. By switching the sockets we can obtain the following sequences of ports on the upper boundary:

- $(1, 2, 3)$ (4) $(5, 6, 7, 8)$;
- $(1, 2, 3)$ (4) $(8, 7, 6, 5)$;
- $(3, 2, 1)$ (4) $(5, 6, 7, 8)$;
- $(3, 2, 1)$ (4) $(8, 7, 6, 5)$.

L: Camping in the woods

Memory Limit : **64 MB**

Johnny went camping together with his friends. They want to use the huts at the camping ground, which are all located on a circle, not necessarily at the same distances (for simplicity, we will measure the distance along the circle). Each member of the group, including Johnny, may choose any hut–there are at least as many huts as friends in the group. However, everybody wants some privacy and it is desirable that the distances between occupied huts are large; clearly each hut is meant for a single occupancy. Help them in planning their holidays: write a program that will determine the set of huts to be occupied, so that the minimal distance (measured along the circle) between them is maximal possible.

Input

The first line of the input contains two natural numbers n and k ($2 \leqslant k \leqslant n \leqslant 500\,000$), separated by a single space, denoting: the number of huts and friends (including Johnny). The second and last line contains a sequence of n natural numbers a_i ($1 \leqslant a_i \leqslant 10^9$), denoting the distance between the hut number i from the hut number $i+1$ (for $i < n$) or the distance between hut n and 1 (for $i = n$). All distances are measured along the circle.

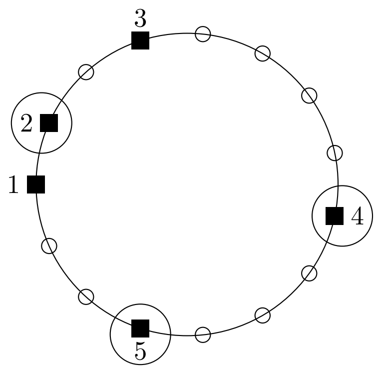
Output

You should write a single natural number in the first and only line of the output: the maximal, among all sets of k occupied huts, minimal distance between two occupied huts. Again, the distance is measured along the circle.

Example

Input	Output
5 3 1 2 5 4 3	4

Situation looks like on picture below:



Friends can choose huts numbered 2, 4 and 5. Then distances between neighbouring selected huts are: 7, 4 and 4.

M: Maximal Sandbox

Memory Limit : **64 MB**

Johnny wants to build a rectangular sandbox in his yard. He went to the the sawmill to buy some planks for its sides, one plank should go for each side, as Johnny has no means of connecting or shortening the planks. Naturally, the bigger the sandbox is, the better. The sawmill offers a large variety of planks of various lengths–help Johnny in choosing four of them that can be used to make a sandbox with maximal area.

Input

The first line of the input contains a single natural number n ($1 \leqslant n \leqslant 1\,000\,000$) that is the number of planks in the sawmill. The second and last line contains a sequence of n natural numbers a_i ($1 \leqslant a_i \leqslant 100$), each separated by a single space. These are the lengths of planks offered by the sawmill.

Output

In the first and only line of the output you should write a single natural number–the largest possible area of the sandbox that can be built from the available planks according to the specifications or the word `NIE` (Polish for 'no') if this is not possible.

Example

Input	Output
9 6 1 5 5 3 8 6 7 6	30

Johnny should take two planks of length 5 and two of length 6.