Problem A. Alice's Nim

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

Alice: "Hi, Bob! Let's play Nim!"

Bob: "Are you serious? I don't want to play it. I know how to win the game."

Alice: "Right, there is an algorithm to calculate the optimal move using XOR. How about changing the rule so that a player loses a game if he or she makes the XOR to 0?"

Bob: "It sounds much better now, but I suspect you know the surefire way to win."

Alice: "Do you wanna test me?"

This game is defined as follows.

1. The game starts with N heaps where the i-th of them consists of a_i stones.
2. A player takes any positive number of stones from any single one of the heaps in one move.
3. Alice moves first. The two players alternately move.
4. If the XOR sum, $a_1 \oplus a_2 \oplus \ldots \oplus a_N$, of the numbers of remaining stones of these heaps becomes 0 as a result of a player's move, the player loses.

Your task is to find which player will win if they do the best move.

Input

The first line contains an integer N which is the number of the heaps ($1 \leq N \leq 10^5$). Each of the following N lines gives the number of stones in each heap ($1 \leq a_i \leq 10^9$).

Output

Output the name of winner, "`Alice`" or "`Bob`", when they do the best move.

Examples

standard input	standard output
2 1 1	Alice
5 1 2 3 4 5	Bob
10 1 2 3 4 5 6 7 8 9 10	Alice

Problem B. Bag with Gifts

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

Santa is going to pack gifts into a bag for a family. There are N kinds of gifts. The size and the price of the i-th gift ($1 \le i \le N$) are s_i and p_i, respectively. The size of the bag is C, thus Santa can pack gifts so that the total size of the gifts does not exceed C. Children are unhappy if they are given multiple items of the same kind gift, so Santa has to choose at most one gift of the same kind per child.

In addition, if a child did not receive a gift that the other children in the same family receive, he/she will complain about that. Hence Santa must distribute gifts fairly to all the children of a family, by giving the same set of gifts to each child. In other words, for a family with k children, Santa must pack zero or k items for each kind of gifts. Santa gives one bag to one family, therefore, the total size of the gifts for each family does not exceed C.

Santa wants to maximize the total price of packed items for a family but does not know the number of children in the family he is going to visit yet. The number seems at most M. To prepare all the possible cases, calculate the maximum total price of items for a family with k children for each $1 \le k \le M$.

Input

The first line contains three integers C, N and M, where C ($1 \le C \le 10^4$) is the size of the bag, N ($1 \le N \le 10^4$) is the number of kinds of the gifts, and M ($1 \le M \le 10^4$) is the maximum number of children in the family. The i-th line of the following N lines contains two integers s_i and p_i ($1 \le s_i, p_i \le 10^4$), where s_i and p_i are the size and the price of the i-th gift, respectively.

Output

The output should consist of M lines. In the k-th line, print the maximum total price of gifts for a family with k children.

Examples

standard input	standard output
6 3 2 1 2 2 10 3 5	17 24
200 5 5 31 41 59 26 53 58 97 93 23 84	235 284 375 336 420
1 1 2 1 1	1 0
2 2 2 1 1 2 100	100 2

Problem C. Colorful Drink

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

In the Jambo Amusement Garden (JAG), you sell colorful drinks consisting of multiple color layers. This colorful drink can be made by pouring multiple colored liquids of different density from the bottom in order.

You have already prepared several colored liquids with various colors and densities. You will receive a drink request with specified color layers. The colorful drink that you will serve must satisfy the following conditions.

- You cannot use a mixed colored liquid as a layer. Thus, for instance, you cannot create a new liquid with a new color by mixing two or more different colored liquids, nor create a liquid with a density between two or more liquids with the same color by mixing them.
- Only a colored liquid with strictly less density can be an upper layer of a denser colored liquid in a drink. That is, you can put a layer of a colored liquid with density x directly above the layer of a colored liquid with density y if x<y holds.

Your task is to create a program to determine whether a given request can be fulfilled with the prepared colored liquids under the above conditions or not.

Input

The first line of the input consists of an integer N ($1 \leq N \leq 10^5$), which represents the number of the prepared colored liquids. The following N lines consists of C_i and D_i ($1 \leq i \leq N$). C_i is a string consisting of lowercase alphabets and denotes the color of the i-th prepared colored liquid. The length of C_i is between 1 and 20 inclusive. D_i is an integer and represents the density of the i-th prepared colored liquid. The value of D_i is between 1 and 10^5 inclusive. The $(N+2)$-nd line consists of an integer M ($1 \leq M \leq 10^5$), which represents the number of color layers of a drink request. The following M lines consists of O_i ($1 \leq i \leq M$). O_i is a string consisting of lowercase alphabets and denotes the color of the i-th layer from the top of the drink request. The length of O_i is between 1 and 20 inclusive.

Output

If the requested colorful drink can be served by using some of the prepared colored liquids, print "`Yes`". Otherwise, print "`No`".

Examples

standard input	standard output
2 white 20 black 10 2 black white	Yes
2 white 10 black 10 2 black white	No
2 white 20 black 10 2 black orange	No
3 white 10 red 20 white 30 3 white red white	Yes

Problem D. Drawing Magic Triangles

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

Fallen angel Yohane plans to draw a magic symbol composed of triangles on the earth. By casting some magic spell on the symbol, she will obtain magic power; this is the purpose for which she will draw a magic symbol. The magic power yielded from the magic symbol is determined only by the common area of all the triangles. Suppose the earth is a two-dimensional plane and the vertices of the triangles are points on the plane. Yohane has already had a design of the magic symbol, i.e. the positions, sizes, shapes of the triangles. However, she does not know how much magic power will be obtained from the symbol. Your task as a familiar of the fallen angel is to write a program calculating the common area of given triangles on a two-dimensional plane.

Input

The first line of the input contains an integer N, which is the number of triangles ($1 \leq N \leq 10^5$). The i-th line of the following N lines contains six integers $x_{i,1}$, $y_{i,1}$, $x_{i,2}$, $y_{i,2}$, $x_{i,3}$, and $y_{i,3}$, where $(x_{i,j}, y_{i,j})$ is the coordinate of the j-th vertex of the i-th triangle ($-1,000 \leq x_{i,j}, y_{i,j} \leq 1,000$).

You can assume that every triangle has a positive area and that the vertices of every triangle are in the counter-clockwise order.

Output

Output the common area of given triangles in a line. The output can contain an absolute or relative error no more than 10^{-6}.

Examples

standard input	standard output
2 0 0 2 0 0 2 0 1 2 1 0 3	0.5
2 0 0 100 0 50 100 50 -50 100 50 0 50	3125
5 0 0 1 0 0 1 0 0 2 0 0 2 0 0 3 0 0 3 0 0 4 0 0 4 0 0 5 0 0 5	0.5

Problem E. Enlarge Circles

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

You are given N distinct points on the 2D plane. For each point, you are going to make a single circle whose center is located at the point. Your task is to maximize the sum of perimeters of these N circles so that circles do not overlap each other. Here, "overlap" means that two circles have a common point which is not on the circumference of at least either of them. Therefore, the circumferences can be touched. Note that you are allowed to make a circle with radius 0.

Input

The first line contains an integer N, which is the number of points ($2 \le N \le 200$). Each of the following N lines gives the coordinates of a point. Integers x_i and y_i ($-100 \le x_i, y_i \le 100$) in the i-th line of them give the x- and y-coordinates, respectively, of the i-th point. These points are distinct, in other words, $(x_i, y_i) \ne (x_j, y_j)$ is satisfied if i and j are different.

Output

Output the maximized sum of perimeters. The output can contain an absolute or a relative error no more than 10^{-6}.

Examples

standard input	standard output
3 0 0 3 0 5 0	31.415926535
3 0 0 5 0 0 5	53.630341225
9 91 -18 13 93 73 -34 15 2 -46 0 69 -42 -23 -13 -87 41 38 68	1049.191683488

Problem F. Formula

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

You are given an integer N and a string consisting of '+' and digits. You are asked to transform the string into a valid formula whose calculation result is smaller than or equal to N by modifying some characters. Here, you replace one character with another character any number of times, and the converted string should still consist of '+' and digits. Note that leading zeros and unary plus are prohibited.

For instance, "0123+456" is assumed as invalid because leading zero is prohibited. Similarly, "+1+2" and '2++3" are also invalid as they each contain a unary expression. On the other hand, "12345", "0+1+2" and "1234+0+0" are all valid.

Your task is to find the minimum number of the replaced characters. If there is no way to make a valid formula smaller than or equal to N, output -1 instead of the number of the replaced characters.

Input

The first line contains an integer N, which is the upper limit of the formula ($1 \leq N \leq 10^9$). The second line contains a string S, which consists of '+' and digits and whose length is between 1 and 1,000, inclusive. Note that it is **not** guaranteed that initially S is a valid formula.

Output

Output the minimized number of the replaced characters. If there is no way to replace, output -1 instead.

Examples

standard input	standard output
100 +123	2
10 +123	4
1 +123	-1
10 ++1+	2
2000 1234++7890	2

Problem G. GuruGuru

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

You are playing a game called Guru Guru Gururin. In this game, you can move with the vehicle called Gururin. There are two commands you can give to Gururin: 'R' and 'L'. When 'R' is sent, Gururin rotates clockwise by 90 degrees. Otherwise, when 'L' is sent, Gururin rotates counterclockwise by 90 degrees.

During the game, you noticed that Gururin obtains magical power by performing special commands. In short, Gururin obtains magical power every time when it performs one round in the clockwise direction from north to north. In more detail, the conditions under which magical power can be obtained is as follows.

- At the beginning of the special commands, Gururin faces north.
- At the end of special commands, Gururin faces north.
- Except for the beginning and the end of special commands, Gururin does not face north.
- During the special commands, Gururin faces north, east, south, and west one or more times, respectively, after the command of 'R'.

At the beginning of the game, Gururin faces north. For example, if the sequence of commands Gururin received in order is RRRR or RRLRRLRR, Gururin can obtain magical power. Otherwise, if the sequence of commands is "LLLL" or "RLLR", Gururin cannot obtain magical power.

Your task is to calculate how many times Gururin obtained magical power throughout the game. In other words, given the sequence of the commands Gururin received, calculate how many special commands exists in it.

Input

The first line consists of a string S, which represents the sequence of commands Gururin received. S consists of 'L' and 'R'. The length of S is between 1 and 10^3 inclusive.

Output

Print the number of times Gururin obtained magical power throughout the game in one line.

Examples

standard input
RRRRLLLLRRRR
standard output
2

standard input
RLLRLLLLRRRLLLRRR
standard output
0

standard input
LR
standard output
0

standard input
`RRLRRLRRRRLRRLRRRRLRRLRRRRLRRLRR`
standard output
`4`

Problem H. Hand Patterns

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

You have a deck of $N \times M$ cards. Each card in the deck has a rank. The range of ranks is 1 through M, and the deck includes N cards of each rank.

We denote a card with rank m by m here.

You can draw a hand of L cards at random from the deck. If the hand matches the given pattern, some bonus will be rewarded. A pattern is described as follows.

```
hand_pattern = card_pattern1 ' ' card_pattern2 ' ' ... ' ' card_patternL
card_pattern = '*' | var_plus
var_plus = variable | var_plus '+'
variable = 'a' | 'b' | 'c'
```

A hand matches the *hand pattern* if each *card pattern* in the hand pattern matches with a distinct card in the hand.

If the card pattern is an asterisk ('`*`'), it matches any card. Characters '`a`', '`b`', and '`c`' denote variables and all the occurrences of the same variable match cards of the same rank.

A card pattern with a variable followed by plus ('`+`') characters matches a card whose rank is the sum of the rank corresponding to the variable and the number of plus characters.

You can assume that, when a hand pattern includes a card pattern with a variable followed by some number of plus characters, it also includes card patterns with that variable and all smaller numbers (including zero) of plus characters. For example, if "`a+++`" appears in a hand pattern, card patterns '`a`', '`a+`', and '`a++`' also appear in the hand pattern.

There is no restriction on which ranks different variables mean. For example, '`a`' and '`b`' may or may not match cards of the same rank.

We show some example hand patterns. The pattern

```
a * b a b
```

matches the hand:

```
3 3 10 10 9
```

with 'a'-s and 'b'-s meaning 3 and 10 (or 10 and 3), respectively. This pattern also matches the following hand.

```
3 3 3 3 9
```

In this case, both 'a's and 'b's mean 3. The pattern

```
a a+ a++ a+++ a++++
```

matches the following hand.

```
4 5 6 7 8
```

In this case, 'a' should mean 4.

Your mission is to write a program that computes the probability that a hand randomly drawn from the deck matches the given hand pattern.

Input

The input is a sequence of datasets. Each dataset is formatted as follows.

```
N M L
card_pattern1 card_pattern2 ... card_patternL
```

The first line consists of three positive integers N, M, and L. N indicates the number of cards in each rank, M indicates the number of ranks, and L indicates the number of cards in a hand. N, M, and L are constrained as follows: $(1 \leq N \leq 7, 1 \leq M \leq 60, 1 \leq L \leq 7, L \leq N \times M)$

The second line describes a hand pattern.

The end of the input is indicated by a line containing three zeros separated by a single space.

Output

For each dataset, output a line containing a decimal fraction which means the probability of a hand matching the hand pattern.

The output should not contain an absolute error greater than 10^{-8}.

No other characters should be contained in the output.

Example

standard input	standard output
1 1 1 a 3 3 4 a+ * a * 2 2 3 a a b 2 2 3 * * * 2 2 3 * b b 2 2 2 a a 2 3 3 a a+ a++ 2 6 6 a a+ a++ b b+ b++ 4 13 5 a a * * * 4 13 5 a a b b * 4 13 5 a a a * * 4 13 5 a a+ a++ a+++ a++++ 4 13 5 * * * * * 4 13 5 a a a b b 4 13 5 a a a a * 7 60 7 a b a b c c * 7 60 7 * * * * * * * 7 60 7 a a+ a++ a+++ a++++ a+++++ a++++++ 1 14 4 b a+ a a 0 0 0	1.0000000000 0.8809523810 1.0000000000 1.0000000000 1.0000000000 0.3333333333 0.4000000000 0.1212121212 0.4929171669 0.0492196879 0.0228091236 0.0035460338 1.0000000000 0.0014405762 0.0002400960 0.0002967709 1.0000000000 0.0000001022 0.0000000000

Problem I. Infinite Multiplication Table

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

You received a card with an integer S and a multiplication table of infinite size. All the elements in the table are integers, and an integer at the i-th row from the top and the j-th column from the left is $A_{i,j} = i \times j$ $(i, j \geq 1)$. The table has infinite size, i.e., the number of the rows and the number of the columns are infinite.

You want to seek for rectangular regions of the table in which the sum of numbers is S. Your task is to count the number of integer tuples (a, b, c, d) that satisfies $1 \leq a \leq b, 1 \leq c \leq d$ and $\sum_{i=a}^{b} \sum_{j=c}^{d} A_{i,j} = S$.

Input

The first line consists of one integer S $(1 \leq S \leq 10^5)$, representing the summation of rectangular regions you have to find.

Output

Print the number of rectangular regions whose summation is S in one line.

Examples

standard input	standard output
25	10
1	1
5	4
83160	5120

Problem J. Jiro's Job

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

Fox Jiro is one of the staffs of the ACM-ICPC 2018 Asia Yokohama Regional Contest and is responsible for designing the network for the venue of the contest. His network consists of N computers, which are connected by M cables. The i-th cable connects the a_i-th computer and the b_i-th computer, and it carries data in both directions. Your team will use the S-th computer in the contest, and a judge server is the T-th computer.

He decided to adjust the routing algorithm of the network to maximize the performance of the contestants through the magical power of prime numbers. In this algorithm, a packet (a unit of data carried by the network) is sent from your computer to the judge server passing through the cables a prime number of times if possible. If it is impossible, the contestants cannot benefit by the magical power of prime numbers. To accomplish this target, a packet is allowed to pass through the same cable multiple times.

You decided to write a program to calculate the minimum number of times a packet from S to T needed to pass through the cables. If the number of times a packet passes through the cables cannot be a prime number, print -1.

Input

The first line of the input consists of four integers N, M, S, and T ($2 \le N \le 10^5$, $1 \le M \le 10^5$, $1 \le S, T \le N$, $S \ne T$). The i-th line of the following M lines consists of two integers a_i and b_i ($1 \le a_i < b_i \le N$), which means the i-th cables connects the a_i-th computer and the b_i-th computer in the network. You can assume that the network satisfies the following conditions.

- The network has no multi-edge, i.e., $(a_i, b_i) \ne (a_j, b_j)$ for all i, j ($1 \le i < j \le M$).
- The packets from N computers are reachable to T by passing through some number of cables. The number is not necessarily a prime.

Output

If there are ways such that the number of times a packet sent from S to T passes through the cables is a prime number, print the minimum prime number of times in one line. Otherwise, print -1.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>5 5 1 5
1 2
1 4
2 3
3 4
3 5</td><td>3</td></tr>
<tr><td>5 4 1 5
1 2
2 3
3 4
4 5</td><td>-1</td></tr>
<tr><td>2 1 1 2
1 2</td><td>3</td></tr>
<tr><td>3 3 1 2
1 2
1 3
2 3</td><td>2</td></tr>
</table>

Problem K. K-rough Sorting

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

For skilled programmers, it is very easy to implement a sorting function. Moreover, they often avoid full sorting to reduce computation time if it is not necessary. Here, we consider "rough sorting" which sorts an array except for some pairs of elements. More formally, we define an array is "K-roughly sorted" if an array is sorted except that at most K pairs are in reversed order. For example, 1 3 2 4 is 1-roughly sorted because (3,2) is only the reversed pair. In the same way, 1 4 2 3 is 2-roughly sorted because (4,2) and (4,3) are reversed.

Considering rough sorting by exchanging adjacent elements repeatedly, you need less number of swaps than full sorting. For example, 4 1 2 3 needs three exchanges for full sorting, but you only need to exchange once for 2-rough sorting.

Given an array and an integer K, your task is to find the result of the K-rough sorting with a minimum number of exchanges. If there are several possible results, you should output the lexicographically minimum result. Here, the lexicographical order is defined by the order of the first different elements.

Input

The first line contains two integers N and K. The integer N is the number of the elements of the array ($1 \leq N \leq 10^5$). The integer K gives how many reversed pairs are allowed ($1 \leq K \leq 10^9$). Each of the following N lines gives the element of the array. The array consists of the permutation of 1 to N, therefore $1 \leq x_i \leq N$ and $x_i \neq x_j$ $(i \neq j)$ are satisfied.

Output

The output should contain N lines. The i-th line should be the i-th element of the result of the K-rough sorting. If there are several possible results, you should output the minimum result with the lexicographical order.

Examples

standard input	standard output
3 1 3 2 1	1 3 2
3 100 3 2 1	3 2 1
5 3 5 3 2 1 4	1 3 5 2 4
5 3 1 2 3 4 5	1 2 3 4 5