

A

Almost Permutation

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Recently Ivan noticed an array a while debugging his code. Now Ivan can't remember this array, but the bug he was trying to fix didn't go away, so Ivan thinks that the data from this array might help him to reproduce the bug.

Ivan clearly remembers that there were n elements in the array, and each element was not less than 1 and not greater than n . Also he remembers q facts about the array. There are two types of facts that Ivan remembers:

- 1 $l_i r_i v_i$ — for each x such that $l_i \leq x \leq r_i$ $a_x \geq v_i$;
- 2 $l_i r_i v_i$ — for each x such that $l_i \leq x \leq r_i$ $a_x \leq v_i$.

Also Ivan thinks that this array was a permutation, but he is not so sure about it. He wants to restore some array that corresponds to the q facts that he remembers and is very similar to permutation. Formally, Ivan has denoted the *cost* of array as follows:

$$\text{cost} = \sum_{i=1}^n (\text{cnt}(i))^2, \text{ where } \text{cnt}(i) \text{ is the number of occurrences of } i \text{ in the array.}$$

Help Ivan to determine minimum possible *cost* of the array that corresponds to the facts!

Input

The first line contains two integer numbers n and q ($1 \leq n \leq 50$, $0 \leq q \leq 100$).

Then q lines follow, each representing a fact about the array. i -th line contains the numbers t_i , l_i , r_i and v_i for i -th fact ($1 \leq t_i \leq 2$, $1 \leq l_i \leq r_i \leq n$, $1 \leq v_i \leq n$, t_i denotes the type of the fact).

Output

If the facts are controversial and there is no array that corresponds to them, print -1 . Otherwise, print minimum possible *cost* of the array.

Examples

input
3 0
output
3

input
3 1 1 1 3 2
output
5

input
3 2 1 1 3 2 2 1 3 2
output
9

input

3 2
1 1 3 2
2 1 3 1

output

-1

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Jul/13/2020 05:36:39 (i3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by



B

PolandBall and Hypothesis

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

PolandBall is a young, clever Ball. He is interested in prime numbers. He has stated a following hypothesis: "There exists such a positive integer n that for each positive integer m number $n \cdot m + 1$ is a prime number".

Unfortunately, PolandBall is not experienced yet and doesn't know that his hypothesis is incorrect. Could you prove it wrong? Write a program that finds a counterexample for any n .

Input

The only number in the input is n ($1 \leq n \leq 1000$) — number from the PolandBall's hypothesis.

Output

Output such m that $n \cdot m + 1$ is not a prime number. Your answer will be considered correct if you output any suitable m such that $1 \leq m \leq 10^3$. It is guaranteed the the answer exists.

Examples

input	
3	
output	
1	

input	
4	
output	
2	

Note

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

For the first sample testcase, $3 \cdot 1 + 1 = 4$. We can output 1.

In the second sample testcase, $4 \cdot 1 + 1 = 5$. We cannot output 1 because 5 is prime. However, $m = 2$ is okay since $4 \cdot 2 + 1 = 9$, which is not a prime number.



C

A Tide of Riverscape

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Walking along a riverside, Mino silently takes a note of something.

"Time," Mino thinks aloud.

"What?"

"Time and tide wait for no man," explains Mino. "My name, taken from the river, always reminds me of this."

"And what are you recording?"

"You see it, tide. Everything has its own period, and I think I've figured out this one," says Mino with confidence.

Doubtfully, Kanno peeks at Mino's records.

The records are expressed as a string s of characters '0', '1' and '.', where '0' denotes a low tide, '1' denotes a high tide, and '.' denotes an unknown one (either high or low).

You are to help Mino determine whether it's possible that after replacing each '.' independently with '0' or '1', a given integer p is **not** a period of the resulting string. In case the answer is yes, please also show such a replacement to Mino.

In this problem, a positive integer p is considered a period of string s , if for all $1 \leq i \leq |s| - p$, the i -th and $(i + p)$ -th characters of s are the same. Here $|s|$ is the length of s .

Input

The first line contains two space-separated integers n and p ($1 \leq p \leq n \leq 2000$) — the length of the given string and the supposed period, respectively.

The second line contains a string s of n characters — Mino's records. s only contains characters '0', '1' and '.', and contains at least one '.' character.

Output

Output one line — if it's possible that p is **not** a period of the resulting string, output any one of such strings; otherwise output "No" (without quotes, you can print letters in any case (upper or lower)).

Examples

input
10 7
1.0.1.0.1.
output
1000100010

input

10 6
1.0.1.1000

output

1001101000

input

10 9
1.....1

output

No

Note

In the first example, 7 is not a period of the resulting string because the 1-st and 8-th characters of it are different.

In the second example, 6 is not a period of the resulting string because the 4-th and 10-th characters of it are different.

In the third example, 9 is always a period because the only constraint that the first and last characters are the same is already satisfied.

Note that there are multiple acceptable answers for the first two examples, you can print any of them.

Supported by



D

Bad Luck Island

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Bad Luck Island is inhabited by three kinds of species: r rocks, s scissors and p papers. At some moments of time two random individuals meet (all pairs of individuals can meet equiprobably), and if they belong to different species, then one individual kills the other one: a rock kills scissors, scissors kill paper, and paper kills a rock. Your task is to determine for each species what is the probability that this species will be the only one to inhabit this island after a long enough period of time.

Input

The single line contains three integers r , s and p ($1 \leq r, s, p \leq 100$) — the original number of individuals in the species of rock, scissors and paper, respectively.

Output

Print three space-separated real numbers: the probabilities, at which the rocks, the scissors and the paper will be the only surviving species, respectively. The answer will be considered correct if the relative or absolute error of each number doesn't exceed 10^{-9} .

Examples

input	
2 2 2	
output	
0.333333333333 0.333333333333 0.333333333333	

input	
2 1 2	
output	
0.150000000000 0.300000000000 0.550000000000	

input	
1 1 3	
output	
0.057142857143 0.657142857143 0.285714285714	



E

Vanya And Cards

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vanya loves playing. He even has a special set of cards to play with. Each card has a single integer. The number on the card can be positive, negative and can even be equal to zero. The only limit is, the number on each card doesn't exceed x in the absolute value.

Natasha doesn't like when Vanya spends a long time playing, so she hid all of his cards. Vanya became sad and started looking for the cards but he only found n of them. Vanya loves the balance, so he wants the sum of all numbers on found cards equal to zero. On the other hand, he got very tired of looking for cards. Help the boy and say what is the minimum number of cards does he need to find to make the sum equal to zero?

You can assume that initially Vanya had infinitely many cards with each integer number from $-x$ to x .

Input

The first line contains two integers: n ($1 \leq n \leq 1000$) — the number of found cards and x ($1 \leq x \leq 1000$) — the maximum absolute value of the number on a card. The second line contains n space-separated integers — the numbers on found cards. It is guaranteed that the numbers do not exceed x in their absolute value.

Output

Print a single number — the answer to the problem.

Examples

input
3 2 -1 1 2
output
1

input
2 3 -2 -2
output
2

Note

In the first sample, Vanya needs to find a single card with number -2.

In the second sample, Vanya needs to find two cards with number 2. He can't find a single card with the required number as the numbers on the lost cards do not exceed 3 in their absolute value.

Supported by



F

Mashmokh and ACM

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Mashmokh's boss, Bimokh, didn't like Mashmokh. So he fired him. Mashmokh decided to go to university and participate in ACM instead of finding a new job. He wants to become a member of Bamokh's team. In order to join he was given some programming tasks and one week to solve them. Mashmokh is not a very experienced programmer. Actually he is not a programmer at all. So he wasn't able to solve them. That's why he asked you to help him with these tasks. One of these tasks is the following.

A sequence of l integers b_1, b_2, \dots, b_l ($1 \leq b_1 \leq b_2 \leq \dots \leq b_l \leq n$) is called *good* if each number divides (without a remainder) by the next number in the sequence. More formally $b_i \mid b_{i+1}$ for all i ($1 \leq i \leq l - 1$).

Given n and k find the number of good sequences of length k . As the answer can be rather large print it modulo 1000000007 ($10^9 + 7$).

Input

The first line of input contains two space-separated integers n, k ($1 \leq n, k \leq 2000$).

Output

Output a single integer — the number of good sequences of length k modulo 1000000007 ($10^9 + 7$).

Examples

input
3 2
output
5

input
6 4
output
39

input
2 1
output
2

Note

In the first sample the good sequences are: [1, 1], [2, 2], [3, 3], [1, 2], [1, 3].

Supported by



G

Drazil and Date

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Someday, Drazil wanted to go on date with Varda. Drazil and Varda live on Cartesian plane.

Drazil's home is located in point $(0, 0)$ and Varda's home is located in point (a, b) . In each step, he can move in a unit distance in horizontal or vertical direction. In other words, from position (x, y) he can go to positions $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ or $(x, y - 1)$.

Unfortunately, Drazil doesn't have sense of direction. So he randomly chooses the direction he will go to in each step. He may accidentally return back to his house during his travel. Drazil may even not notice that he has arrived to (a, b) and continue travelling.

Luckily, Drazil arrived to the position (a, b) successfully. Drazil said to Varda: "It took me exactly s steps to travel from my house to yours". But Varda is confused about his words, she is not sure that it is possible to get from $(0, 0)$ to (a, b) in exactly s steps. Can you find out if it is possible for Varda?

Input

You are given three integers a , b , and s ($-10^9 \leq a, b \leq 10^9$, $1 \leq s \leq 2 \cdot 10^9$) in a single line.

Output

If you think Drazil made a mistake and it is impossible to take exactly s steps and get from his home to Varda's home, print "No" (without quotes).

Otherwise, print "Yes".

Examples

input
5 5 11
output
No

Note

In fourth sample case one possible route is: $(0, 0) \rightarrow (0, 1) \rightarrow (0, 0)$.

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Jul/13/2020 05:33:55 (i3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by



ITMO UNIVERSITY

H

Weak Memory

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Zart PMP is qualified for ICPC World Finals in Harbin, China. After team excursion to Sun Island Park for snow sculpture art exposition, PMP should get back to buses before they leave. But the park is really big and he does not know how to find them.

The park has n intersections numbered 1 through n . There are m bidirectional roads that connect some pairs of these intersections. At k intersections, ICPC volunteers are helping the teams and showing them the way to their destinations. Locations of volunteers are fixed and distinct.

When PMP asks a volunteer the way to bus station, he/she can tell him the whole path. But the park is fully covered with ice and snow and everywhere looks almost the same. So PMP can only memorize at most q intersections after each question (excluding the intersection they are currently standing). He always tells volunteers about his weak memory and if there is no direct path of length (in number of roads) at most q that leads to bus station, the volunteer will guide PMP to another volunteer (who is at most q intersections away, of course). ICPC volunteers know the area very well and always tell PMP the best way. So if there exists a way to bus stations, PMP will definitely find it.

PMP's initial location is intersection s and the buses are at intersection t . There will always be a volunteer at intersection s . Your job is to find out the minimum q which guarantees that PMP can find the buses.

Input

The first line contains three space-separated integers n, m, k ($2 \leq n \leq 10^5$, $0 \leq m \leq 2 \cdot 10^5$, $1 \leq k \leq n$) — the number of intersections, roads and volunteers, respectively. Next line contains k distinct space-separated integers between 1 and n inclusive — the numbers of cities where volunteers are located.

Next m lines describe the roads. The i -th of these lines contains two space-separated integers u_i, v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — two intersections that i -th road connects. There will be at most one road between any two intersections.

Last line of input contains two space-separated integers s, t ($1 \leq s, t \leq n$, $s \neq t$) — the initial location of PMP and the location of the buses. It might not always be possible to reach t from s .

It is guaranteed that there is always a volunteer at intersection s .

Output

Print on the only line the answer to the problem — the minimum value of q which guarantees that PMP can find the buses. If PMP cannot reach the buses at all, output -1 instead.

Examples

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

3

input

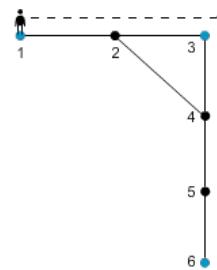
6 5 3
1 5 6
1 2
2 3
3 4
4 5
6 3
1 5

output

3

Note

The first sample is illustrated below. Blue intersections are where volunteers are located. If PMP goes in the path of dashed line, it can reach the buses with $q = 3$:



In the second sample, PMP uses intersection 6 as an intermediate intersection, thus the answer is 3.

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Jul/13/2020 05:36:20 (i3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by



I

Copying Data

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

We often have to copy large volumes of information. Such operation can take up many computer resources. Therefore, in this problem you are advised to come up with a way to copy some part of a number array into another one, quickly.

More formally, you've got two arrays of integers a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n of length n . Also, you've got m queries of two types:

1. Copy the subsegment of array a of length k , starting from position x , into array b , starting from position y , that is, execute $b_{y+q} = a_{x+q}$ for all integer q ($0 \leq q < k$). The given operation is correct — both subsegments do not touch nonexistent elements.
2. Determine the value in position x of array b , that is, find value b_x .

For each query of the second type print the result — the value of the corresponding element of array b .

Input

The first line contains two space-separated integers n and m ($1 \leq n, m \leq 10^5$) — the number of elements in the arrays and the number of queries, correspondingly. The second line contains an array of integers a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$). The third line contains an array of integers b_1, b_2, \dots, b_n ($|b_i| \leq 10^9$).

Next m lines contain the descriptions of the queries. The i -th line first contains integer t_i — the type of the i -th query ($1 \leq t_i \leq 2$). If $t_i = 1$, then the i -th query means the copying operation. If $t_i = 2$, then the i -th query means taking the value in array b . If $t_i = 1$, then the query type is followed by three integers x_i, y_i, k_i ($1 \leq x_i, y_i, k_i \leq n$) — the parameters of the copying query. If $t_i = 2$, then the query type is followed by integer x_i ($1 \leq x_i \leq n$) — the position in array b .

All numbers in the lines are separated with single spaces. It is guaranteed that all the queries are correct, that is, the copying borders fit into the borders of arrays a and b .

Output

For each second type query print the result on a single line.

Examples

input

```
5 10
1 2 0 -1 3
3 1 5 -2 0
2 5
1 3 3 3
2 5
2 4
2 1
1 2 1 4
2 1
2 4
1 4 2 1
2 2
```

output

```
0
3
-1
```

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jul/13/2020 05:36:02 (i3).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



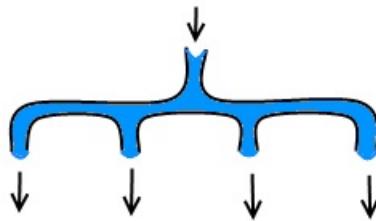
J

Pipeline

time limit per test: 0.4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vova, the Ultimate Thule new shaman, wants to build a pipeline. As there are exactly n houses in Ultimate Thule, Vova wants the city to have exactly n pipes, each such pipe should be connected to the water supply. A pipe can be connected to the water supply if there's water flowing out of it. Initially Vova has only one pipe with flowing water. Besides, Vova has several splitters.

A splitter is a construction that consists of one input (it can be connected to a water pipe) and x output pipes. When a splitter is connected to a water pipe, water flows from each output pipe. You can assume that the output pipes are ordinary pipes. For example, you can connect water supply to such pipe if there's water flowing out from it. At most one splitter can be connected to any water pipe.



The figure shows a 4-output splitter

Vova has one splitter of each kind: with 2, 3, 4, ..., k outputs. Help Vova use the minimum number of splitters to build the required pipeline or otherwise state that it's impossible.

Vova needs the pipeline to have exactly n pipes with flowing out water. Note that some of those pipes can be the output pipes of the splitters.

Input

The first line contains two space-separated integers n and k ($1 \leq n \leq 10^{18}$, $2 \leq k \leq 10^9$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print a single integer — the minimum number of splitters needed to build the pipeline. If it is impossible to build a pipeline with the given splitters, print -1.

Examples

input
4 3
output
2

8 4

output

-1

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Jul/13/2020 05:35:25 (i3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by



ITMO UNIVERSITY

K

Crazy Town

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Crazy Town is a plane on which there are n infinite line roads. Each road is defined by the equation $a_i x + b_i y + c_i = 0$, where a_i and b_i are not both equal to the zero. The roads divide the plane into connected regions, possibly of infinite space. Let's call each such region a block. We define an intersection as the point where at least two different roads intersect.

Your home is located in one of the blocks. Today you need to get to the University, also located in some block. In one step you can move from one block to another, if the length of their common border is nonzero (in particular, this means that if the blocks are adjacent to one intersection, but have no shared nonzero boundary segment, then it is not allowed to move from one to another one in one step).

Determine what is the minimum number of steps you have to perform to get to the block containing the university. It is guaranteed that neither your home nor the university is located on the road.

Input

The first line contains two space-separated integers x_1, y_1 ($-10^6 \leq x_1, y_1 \leq 10^6$) — the coordinates of your home.

The second line contains two integers separated by a space x_2, y_2 ($-10^6 \leq x_2, y_2 \leq 10^6$) — the coordinates of the university you are studying at.

The third line contains an integer n ($1 \leq n \leq 300$) — the number of roads in the city. The following n lines contain 3 space-separated integers ($-10^6 \leq a_i, b_i, c_i \leq 10^6$; $|a_i| + |b_i| > 0$) — the coefficients of the line $a_i x + b_i y + c_i = 0$, defining the i -th road. It is guaranteed that no two roads are the same. In addition, neither your home nor the university lie on the road (i.e. they do not belong to any one of the lines).

Output

Output the answer to the problem.

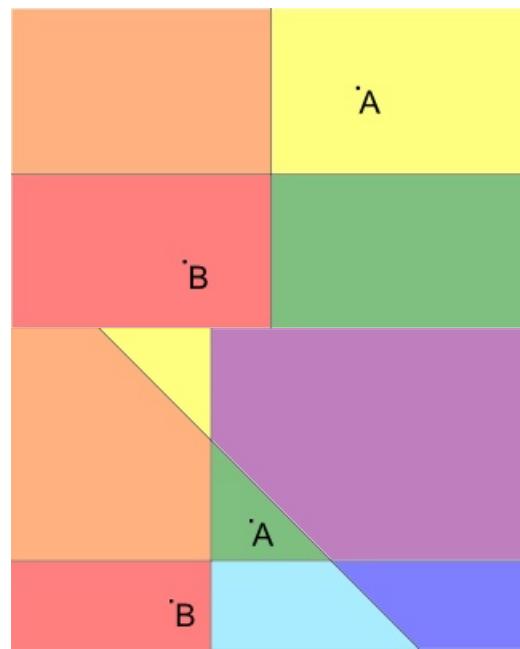
Examples

input
1 1 -1 -1 2 0 1 0 1 0 0
output
2

input
1 1 -1 -1 3 1 0 0 0 1 0 1 1 -3
output
2

Note

Pictures to the samples are presented below (A is the point representing the house; B is the point representing the university, different blocks are filled with different colors):



[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jul/13/2020 05:35:07 (i3).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



L

Vasya and Chess

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vasya decided to learn to play chess. Classic chess doesn't seem interesting to him, so he plays his own sort of chess.

The queen is the piece that captures all squares on its vertical, horizontal and diagonal lines. If the cell is located on the same vertical, horizontal or diagonal line with queen, and the cell contains a piece of the enemy color, the queen is able to move to this square. After that the enemy's piece is removed from the board. The queen cannot move to a cell containing an enemy piece if there is some other piece between it and the queen.

There is an $n \times n$ chessboard. We'll denote a cell on the intersection of the r -th row and c -th column as (r, c) . The square $(1, 1)$ contains the white queen and the square $(1, n)$ contains the black queen. All other squares contain green pawns that don't belong to anyone.

The players move in turns. The player that moves first plays for the white queen, his opponent plays for the black queen.

On each move the player has to capture some piece with his queen (that is, move to a square that contains either a green pawn or the enemy queen). The player loses if either he cannot capture any piece during his move or the opponent took his queen during the previous move.

Help Vasya determine who wins if both players play with an optimal strategy on the board $n \times n$.

Input

The input contains a single number n ($2 \leq n \leq 10^9$) — the size of the board.

Output

On the first line print the answer to problem — string "white" or string "black", depending on who wins if the both players play optimally.

If the answer is "white", then you should also print two integers r and c representing the cell (r, c) , where the first player should make his first move to win. If there are multiple such cells, print the one with the minimum r . If there are still multiple squares, print the one with the minimum c .

Examples

input
2
output
white 1 2

Note

In the first sample test the white queen can capture the black queen at the first move, so the white player wins.

In the second test from the statement if the white queen captures the green pawn located on the central vertical line, then it will be captured by the black queen during the next move. So the only move for the white player is to capture the green pawn located at (2, 1).

Similarly, the black queen doesn't have any other options but to capture the green pawn located at (2, 3), otherwise if it goes to the middle vertical line, it will be captured by the white queen.

During the next move the same thing happens — neither the white, nor the black queen has other options rather than to capture green pawns situated above them. Thus, the white queen ends up on square (3, 1), and the black queen ends up on square (3, 3).

In this situation the white queen has to capture any of the green pawns located on the middle vertical line, after that it will be captured by the black queen. Thus, the player who plays for the black queen wins.

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Jul/13/2020 05:34:49 (i3).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by

