Problem A. Array As Palindrome

Time limit:	2 seconds
Memory limit:	512 megabytes

Let us call a zero-based array a of length n *palindromic*, if for any index i $(0 \le i < n)$ $a[i] = a[n-1-i]$.

Given an array, your task is to insert **exactly one** element in it to make an array palindromic, or determine that it is impossible.

Input

The first line of the input consists of one integer n $(1 \le n \le 100\,000)$, the length of array.

The second line contains n integers $a_0, a_1, \ldots, a_{n-1}$ $(-10^9 \le a_i \le 10^9)$.

Output

If it is impossible to insert exactly one integer into the array to make it palindromic, print -1.

Otherwise print two integers: index k $(0 \le k \le n)$ of inserted element in the new array and value of this element v $(-10^9 \le v \le 10^9)$.

For example, if you are inserting an element at the beginning, $k = 0$, if you are inserting an element between $a[i-1]$ and $a[i]$, $k = i$, and if you are inserting an element at the end, $k = n$.

If there is more than one way to obtain a palindromic array from the given one, you can choose any of them.

Examples

standard input	standard output
2 20 18	2 20
1 -2018	0 -2018
4 2 0 1 8	-1
2 8 8	1 2018

Note

- In the first sample, you add `20` to the end of array, obtaining an palindromic array `20 18 20`.
- In the second sample, you add `-2018` to the beginning of array, obtaining an palindromic array `-2018 -2018`.
- In the third sample, you cannot make an array palindromic by inserting one element, so the answer is -1.
- In the fourth sample you may insert any integer between two 8's, because `8 x 8` is a palindrome for any x. So let it be 2018.

Problem B. Balanced Advertising

Time limit:	4 seconds
Memory limit:	512 megabytes

— How do you sustain a business model in which users don't pay for your service?
— Senator, we run ads.

Mark Zuckerberg's testimony.

Many IT companies make most of their profit by selling ad slots every time the user enters their website, application, etc. So is Back Catalog Inc. In this problem we are going to consider ads on a specific topic which wished to remain anonymous. What is important is that there are n users looking for pages on this topic and there are m websites they are interested in. Moreover, we are given a list of k user-website pairs meaning that this user sometimes visits this website. There are no other visits to website except for mentioned in this list.

The topic we consider is not only specific, it is also very rare, so there are only two companies that would like to purchase some advertising. In particular, they can buy the exclusive rights to show ads to some particular user on some particular website. Obviously, none of the companies will buy rights for a non-existent user-website pair. Moreover, if a user-website pair remains unused (none of these two companies will buy it), this user will see no ads at this website at all.

The good old days of the internet are gone and nowadays there are tons of advertising rules to comply with. The first rule is that for each user, the number of websites that show him ads of the first company and the number of websites that show him ads of the second company should differ by no more than 1 (absolute difference). The second rule is that for each website, the number of users it shows ads of the first company and the number of users it shows ads of the second company should also differ by no more than 1 (absolute difference).

You are hired by the European Commission to verify the fairness of the above rules. None of the user-website pairs are acquired by advertising companies yet. Your goal is to find the maximum possible difference between the number of pairs bought by the first company and the number of pairs bought by the second company if all rules are obeyed.

Input

The first line of the input contains three integers n, m and k ($1 \le n, m, k \le 100\,000$) — the number of users, the number of websites and the number of user-website pairs for ad sales.

Then follow k lines. The i-th of these lines contains two integers u_i and v_i ($1 \le u_i \le n$, $1 \le v_i \le m$), meaning that user u_i visits website v_i. No pair appears in the input more than once.

Output

In the first line of the input print two integers a and b ($0 \le b \le a \le k$), the number of user-website pairs that the first company should buy and the number of user-website pairs that the second company should buy in the most unfair scenario, i.e. in the case that maximizes $a - b$.

The second line should contain a distinct integers from 1 to k — indices of the pairs that the first company should buy.

The third line should contain b distinct integers from 1 to k — indices of the pairs that the second company should buy. Of course, none of these integers should appear in the second line that defines pairs for the first company.

If either of the subsets is empty just leave the corresponding line blank.

In each of the second and third lines it is allowed to print indices in arbitrary order. If there are several possible answers, you are allowed to print any of them.

Examples

standard input	standard output

3 4 4
1 1
2 2
3 3
3 4
	3 0
	1 2 3

3 3 9
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
	6 3
	1 8 9 3 4 5
	2 7 6

Problem C. Cutting the Strip

Time limit:	2 seconds
Memory limit:	512 megabytes

You work on a factory and operate a paper cutting machine. This machine can take a strip of paper of any length and cut it in either 2 or 3 equal parts.

As a part of your daily routine you receive a paper strip of length n and a list of desired pieces' lengths you should make. The list consists of m pairs of integers c_i and l_i meaning the result of your work should contain exactly c_i paper strips of length l_i. The times are tough and none of your clients tolerate paper losses so the total length of all pieces equals n, i.e. $\sum_{i=1}^{m} l_i \cdot c_i = n$.

Determine whether it is possible to fulfill the order and design the list of instructions to do so.

Input

The first line of the input contains single integer n ($1 \le n \le 2^{62}$) — the length of the paper strip you are given at the beginning. It is guaranteed that $n = 2^a \cdot 3^b$ for some integer a and b.

The second line contains a single integer m ($1 \le m \le 700$) — the number of pairs that describe the resulting lengths.

Next follow m lines, the i-th of these lines contains two integers l_i and c_i ($1 \le l_i \le n$, $1 \le c_i \le n$). It is guaranteed that all l_i are distinct and that the total length of all pieces equals to n.

Output

If there is no way to fulfill the order print "`NO`" (without quotes) in the only line of the output.

Otherwise, print "`YES`" (without quotes) in the first line of the output. In the second line print the number of operations blocks k ($0 \le k \le 500\,000$). The print k lines containing blocks descriptions. Each description consists of three integers l_i, c_i and o_i ($1 \le l_i, c_i \le n$, $2 \le o_i \le 3$) and means one should use the cutting machine to take c_i paper strips of length l_i and cut each of them in o_i parts of equal length.

It is guaranteed that if the solution exists, there exists a solution that uses no more than $500\,000$ blocks of operations.

If there are several possible solutions, you are allowed to print any of them.

Examples

standard input	standard output

1
1
1 1
	YES
	0

6
2
2 2
1 2
	YES
	2
	6 1 3
	2 1 2

6
3
2 1
3 1
1 1
	NO

6
2
5 1
1 1
	NO

Note

In the second sample one cannot swap the order of operations blocks.

Problem D. Distance and Union

Time limit:	8 seconds
Memory limit:	512 megabytes

You are given an undirected graph with n vertices. Initially it does not contain any edges.

You should perform two types of queries with this graph:

1. Add an edge between vertices u_i and v_i. It is guaranteed that there was no such edge in a graph before the query and that after performing the query graph is a forest.
2. Print the number of vertices w such that the distance between u_i and w equals to k_i where the distance is the number of edges in a simple path connecting u_i and w.

Input

The first line contains two integers n and q ($1 \le n \le 100\,000$, $1 \le q \le 200\,000$), the number of vertices and the number of queries respectively.

Next q lines contain three integers t_i, a_i, b_i ($1 \le t_i \le 2$, $0 \le a_i, b_i \le n-1$), the type of query and two numbers describing the query.

Let $last$ be the answer to the last query of second type or zero if there were no queries of second type yet. Then, if $t_i = 1$, you need to add an edge between vertices u_i and v_i where $u_i = ((a_i + last) \mod n) + 1$ and $v_i = ((b_i + last) \mod n) + 1$. If $t_i = 2$, you need to calculate the number of vertices w such that the distance between u_i and w is k_i where $u_i = ((a_i + last) \mod n) + 1$ and $k_i = (b_i + last) \mod n$.

Output

Print all answers to the queries of the second type in separate lines.

Example

standard input	standard output

6 7
1 0 1
1 2 3
1 2 4
2 2 1
1 0 3
1 5 0
2 5 0
	2
	3

Note

In the sample test the performed queries are:

1. Add an edge between 1 and 2.
2. Add an edge between 3 and 4.
3. Add an edge between 3 and 5.
4. Find the number of vertices w such that the distance between 3 and w equals to 1. There are two such vertices: 4 and 5; $last$ becomes equal to 2.
5. Add an edge between 3 and 6.
6. Add an edge between 2 and 3.

7. Find the number of vertices w such that the distance between 2 and w equals to 2. There are three such vertices: 4, 5 and 6; $last$ becomes equal to 3.

Problem E. Exhibitions

Time limit: 3 seconds
Memory limit: 512 megabytes

Camilla is taking a trip to Bytetown, Byteland. There are two main activities in Bytetown, swimming in a warm salty ocean and visiting the museum. Every day Camilla has to choose which one of the two activities to do.

There are two kinds of exhibitions in the Bytetownian Museum. The first kind occur on weekly basis: every exhibition holds every week at a certain day of week. The second kind occur on monthly basis and follow the same rule. All weekly and monthly exhibitions are different. Someday it may happen that both a weekly and a monthly exhibitions happen — what a lucky day for visitors, they can watch both of them during a single day!

Camilla wants to visit each exhibition at least once. She knows that Bytelandian week is p days and month is q days. When she arrives at Bytetown, it will be the first day of week and month (what a rare coincidence). She is going to stay in Bytetown for n days.

Besides exhibitions, Camilla likes swimming a lot, so she wants to spend the least possible number of days on going to the museum. Help Camilla to plan her vacation and find the minimum number of days required to visit all exhibitions.

Input

In the first line of input there are three numbers n, p, q ($1 \le n \le 10^{18}$, $1 \le p, q \le 10^7$). Two next lines contain strings of length $\lceil p/4 \rceil$ and $\lceil q/4 \rceil$ respectively containing digits and letters `a-f`, describing weekly and monthly exhibitions in hexadecimal format. The format description for weekly exhibitions is described below; monthly description follows the same format.

First, we write down a binary string of length p, where i-th ($1 \le i \le p$) character is `1` if there is an exhibition on i-th day of a week and `0` otherwise. Then we append zeroes to the end of the string until its length is divisible by 4. Then we split the string into blocks of length 4 and encode each block as a hexadecimal digit, least significant bits first.

For example, if $p = 6$ and exhibitions happen on days 2, 3, 4, and 6, we get the string `011101`. We append zeroes and get `01110100`. Finally, we encode this string as `e2` because $e_{16} = 14_{10} = 1110_2$ and $2_{16} = 2_{10} = 0010_2$.

Output

Print a single number, the minimum required number of days or "`-1`" if Camilla cannot visit all exhibitions.

Examples

standard input	standard output

6 4 3
4
3
	3

7 4 3
4
3
	2

2 4 3
4
3
	-1

100 48 33 596dda1c04c3 abc0abfe1	27

Note

In three first sample tests weeks are 4 days long and months are 3 days long. Weekly exhibitions are held only on the third day of week, monthly are held on the first and second days of month.

In the first sample Camilla has 6 days for her visits. She can visit the museum on the first three days, seeing one exhibition each day.

In the second sample she has 7 days. She can visit the museum on the day 2, seeing the second monthly exhibition, and on the day 7, seeing the weekly exhibition and the first monthly exhibition at the same time.

In the third sample Camilla does not have enough time to see the weekly exhibition at all.

Problem F. Folding

Time limit:	2 seconds
Memory limit:	256 megabytes

Little boy Alex likes origami very much. He spends days and nights folding weird figures from sheets of paper. When Alex starts to fold a new figure, he does the following:

- He takes a square sheet of paper of size $n \times n$ and puts it into a Cartesian plane so that the center of the square is located at the origin $(0, 0)$ and sheet edges are parallel to the axes.
- After that he makes several folds.

When making a fold, he chooses an **oriented** line l, takes the part of the figure located to the left of l and puts it above the part of the figure located to the right of l by folding it over l. For more details refer to the input format.

After making k folds, Alex wonders what percentage the resulting figure area is of the initial area of the square sheet. Help him find that out.

Input

The first line of input contains two integers n and k ($1 \le n \le 100$, $1 \le k \le 5$), the side length of the original square sheet of paper and the number of folds made by Alex.

The following k lines describe the folds. Each description consists of four integers x_1, y_1, x_2, y_2 ($-100 \le x_1, x_2, y_1, y_2 \le 100$) that define an **oriented** fold line l. It is guaranteed that the points (x_1, y_1) and (x_2, y_2) are distinct.

The left part of the figure is put onto the right part of the figure. To decide which side is left and which side is right, one must imagine himself standing at the point (x_1, y_1) looking in the direction of the point (x_2, y_2), the left hand side would define the left part and the right-hand side would define the right part. Note that by this definition the order of points (x_1, y_1) and (x_2, y_2) matters.

It is possible that the fold line l does not intersect the figure at all. This case is handled naturally: if the figure is located to the left of l, it is reflected (mirrored) over the folding line, and if it is located to the right of l, nothing happens.

Output

The first line of output must contain a single integer between 0 and 100 defining which percentage the resulting figure area is of the intitial sheet of paper area.

Your answer will be considered correct if it differs from the actual fraction (not rounded to the integer number of percents) by no more than 1%.

Examples

stdin	stdout

10 3
0 0 5 5
0 0 0 5
-5 5 -5 -5
	38

10 1 -5 -5 1 4	67

Problem G. Good Old Football

Time limit:	1 second
Memory limit:	512 megabytes

FIFA World Cup 2018 held in Russia gave a huge impact on football popularity in Moscow. Fascinated by the great matches, lots of foreign football fans and overall impression of non-stop holiday in the city, many young people decided to drop computer games and went to streets in order to play football and to become the next generation of first class football players.

Unfortunately, there are too many players and too little places suitable for playing football at the streets. There is a company of n friends that want to play as a team in a match tomorrow evening on a local street field. Street matches do not usually follow the football rules as strictly as the official matches. For example, they may lack the referee, the field may be significantly smaller or the game may last for a different time rather than regular 45 minutes per half. This time, the team should have m people and the game is going to last for t minutes.

The number m happened to be smaller than n, meaning that not everyone will be able to spend all the time at the field. In order to give everybody a chance to play for a long enough time, friends came up with a following replacement algorithm: after each 5 minutes of a game, player of a team that has spend the most time at the field among current players leaves the field and currently waiting player who spent the least time at the field enters the game instead. If there are several choices for leaving and entering players, each of them is chosen arbitrarily among people having the most or least playing time respectively. Consider the replacement itself to be instant (i.e. taking 0 minutes to perform).

Game has not started yet but the friends are curious whether their algorithm is indeed fair. Calculate the least and the most time that is going to be spent by any of the friends at the field.

Input

The only line of input contains three integers ($2 \le n \le 10^{12}$, $1 \le m < n$, $1 \le t \le 10^{12}$), the number of friends, the number of players at field and the duration of a game in minutes.

Output

Print two integers a and b, the least among the times spend by all friends at the field in minutes and the most.

Examples

standard input	standard output
`6 4 13`	`3 13`
`12 11 60`	`55 55`

Note

Consider the first sample case. Suppose we have players numbered from 1 to 6 and players from 1 to 4 are initially at the field. There will be two replacements; they may go as follows. First, by the end of minute 5 the player 1 leaves the field (as he already spent 5 minutes in game as well as the players 2, 3 and 4) and becomes replaced with a player 5 (as he spent 0 minutes at the field as well as the player 6). Then, by the end of minute 10 the player 2 leaves the field (as he already spent 10 minutes at the field as well as the players 3 and 4, and the player 5 spent only 5 minutes at the field) and becomes replaced with a player 6 (as he spent 0 minutes at the field, while the player 1 who is currently out of the field already spent 5 minutes in game).

Thus, by the end of the game player 1 will spend 5 minutes in the game, player 2 will spend 10 minutes in the game, players 3 and 4 will play for all 13 minutes, player 5 will spend 8 minutes in the game and player 6 will spend only 3 minutes in the game.

In the second sample case each time the replacement happens, the only remaining player enters the game and each player will spend exactly 5 minutes out of the game, thus all players will spend exactly 55 minutes in game.

Problem H. Hard Learning

Time limit:	1 second
Memory limit:	512 megabytes

Young Nick has a literature lesson tomorrow. His homework is to learn a poem by heart. Nick is a well known procrastinator, so he is going to start learning the poem just before the lesson, as late as possible.

The poem consists of n stanzas. It takes exactly one minute to learn any stanza by heart and exactly one minute to declaim it afterwards. As you may know, the more times you learn a stanza, the longer you remember it. Nick knows that if he learns some stanza for the k-th time (not necessary in a row) then he will be able to declaim it during next k^2 minutes. For example, if he learns a stanza for the first time, he must declaim it immediately. If he learns it once again later, he will have three minutes to start declaiming before he forgets it. After next learning he will have eight minutes, and so on. It does not matter what happens between successive attempts to learn a stanza, only their number matters.

Nick knows that he will start declaiming the poem as soon as the lesson starts. He must declaim all n stanzas one by one. How much time before the lesson does Nick need to learn all stanzas and declaim the poem successfully?

Input

A single integer n ($1 \le n \le 10\,000$): the number of stanzas in the poem.

Output

In the first line print an integer k: the minimum number of minutes required to learn the whole poem to be able to declaim it. In the next line print k integers from 1 to n: the order in which Nick should learn stanzas to be able to declaim it from 1-st to n-th.

Examples

standard input	standard output
1	1 1
4	8 1 2 3 4 1 2 3 4

Problem I. Intellectual Shopping

Time limit:	3 seconds
Memory limit:	512 megabytes

Though Helen is a well-payed engineer in a well-known company she enjoys saving money a lot! This time she wants to find out how to use promotions in the neighboring grocery to optimize her purchases.

Every time Helen visits the shop she buys exactly n items. The i-th item costs c_i burles. During the j-th day, shop offers the following promotion. For every x_j items you buy you pay only y_j part of the full price for the least valuable item in this group. That means Helen can save some money by splitting her n items in groups of size x_j (and, possibly, one group of a smaller size) and paying less for one item in each group except the smaller one if it is present.

Shop has already announced promotions for m next days. Help Helen compute the minimum amount of money she needs to purchase all n items on each of these days.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 300\,000$), the number of items Helen purchases during every visit to the shop and the number of promotions to consider.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^9$), costs of individual items.

Then follow m. The j-th of these lines contains integer x_j and real value y_j ($2 \le x_j \le 300\,000$, $0 \le y_j \le 1$) — parameters of the j-th day promotion. Real values y_j are given with no more than six digits after decimal point.

Output

Print m real values. The i-th of them should be equal to the minimum possible amount of money required to purchase all n items on the i-th day. Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}.

Examples

standard input	standard output

4 2
1 1 1 1
2 0.75
3 0
	3.5000000000
	3.0000000000

3 3
10 9 2
2 0.5
3 0.2
4 0
	16.5000000000
	19.4000000000
	21.0000000000

Problem J. Judge Problem

Time limit:	2 seconds
Memory limit:	512 megabytes

The day of Nsk subregional competition is approaching and it is time to prepare a problem set. There are m members of jury team and n problems selected for the contest. Each of the problems is proposed by one of these jury members. For each problem chief judge identified the difficulty of its development d_i.

Though jury members used to be good participants like you, they took an arrow in the knee a long time ago. Each of them tells the chief judge his limit a_j meaning this jury member is ready to develop only problems of difficulty $d_i \leq a_j$. However, it is always more interesting to work on your own problems, so they also name threshold b_j meaning they are ready to develop their own problem if $d_i \leq b_j$. Moreover, all jury members are extremely busy so they would like to reduce their part in contest preparation as much as possible.

Chief judge knows that every time he tries to convince some jury member to prepare one more task, this member's discomfort grows exponentially. So he decided to minimize the maximum number of problems one jury member will work on.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 200\,000$), the number of problems selected for the upcoming competition and the number of jury members.

The second line contains n integers d_i ($1 \leq d_i \leq 200\,000$), the i-th of them denotes the complexity of development of the i-th problem.

The third line contains n integers e_i ($1 \leq e_i \leq m$), the i-th of them stands for the index of the jury member who suggested this problem idea for the competition.

The following m lines describe jury members capabilities. The i-th line contains two integers a_i and b_i ($1 \leq a_i \leq b_i \leq 200\,000$), the maximum difficulty of a problem that the i-th jury member agrees to work on if he is not this problem's idea author and the same value for his own problems.

Output

Print the minimum possible value of maximum number of problems that some jury member will work on. If there is no way to prepare the given problem set obeying all the constraints, print -1.

Examples

standard input	standard output

4 2
5 10 9 8
1 1 2 1
5 10
3 10
	3

3 1
3 5 10
1 1 1
4 8
	-1

4 3
10 6 8 4
1 2 3 1
7 20
4 5
3 8
	2

Note

In the first sample test, the second jury member is only capable to work on his own problem, so the first member takes responsibility for the remaining three problems.

In the second sample test the only jury member is not able to prepare the third problem (sometimes that happens, true story!)

In the third sample test, jury member 1 works on problems 1 and 2, while member 2 works on problem 4 and member 3 works on problem 3.

Problem K. Kingdom and Reforms

Time limit:	1 second
Memory limit:	512 megabytes

In the kingdom of Berland there are n cities and m bidirectional roads connecting some pairs of cities. The government did not care for the road system for the long time, hence there was no rhyme or reason to how the roads were constructed. It is possible, for instance, for multiple roads to connect the same pair of cities, or even for a city to have an adjacent road leading to the same city (or multiple such roads even). Furthermore, it may be possible that the country is not connected at all, that is, for a pair of cities there may not be a way to reach one from the other using the roads. Needless to say, all roads are in a very sorry state and desperately need repairs.

A new transportation minister was assigned recently, tasked with renovation of the road system. The minister does not really want to create new roads nor to destroy any of the old ones, since the citizens grew accustomed to the existing road system, however poor it may be. The renovation the minister has in mind will consist of converting each existing road into either a modern automobile highway or a high-speed train track. To avoid anti-monopoly investigations, the minister has to ensure that each city has at least one adjacent highway and at least one adjacent train track. Help the minister choose the new type for each road so that this condition is satisfied, or determine that doing this is impossible.

Input

The first line contains two integers n and m ($1 \le n \le 300\,000$, $0 \le m \le 300\,000$), the number of cities and roads respectively.

The following m lines describe the roads. The i-th of these lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$), indices of the cities connected by the i-th road.

Note that multiple roads and roads connecting a city to itself are allowed, and it is not guaranteed that all the cities are connected by the road network.

Output

If there is a suitable assignment of road types, print a string of m characters. The i-th of these characters should be "`H`" if the i-th road should be converted into a highway, or "`T`" if it should be made into a train track. If there are many possible assignments, print any of them.

If there is no suitable assignment, print the only string "`Impossible`".

Examples

standard input	standard output

3 4
1 2
1 3
2 3
2 3
	HTHT

3 1 1 2	Impossible

3 4
1 1
1 1
2 3
3 2
	HTHT