Information

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Testing

Notice, that each subtask has a list of required subtasks. Subtask will be tested only if all tests of all required subtasks are passed. Be careful, some subtasks might not be tested, if your solution doesn't pass sample tests.

Scoring

We have two types of subtask scoring: "test" and "subtask".

"Test" means that points are awarded for each test in a subtask independently of other tests in this subtask.

"Subtask" means that points are awarded only if all tests in this subtask are passed.

For more information on subtask scoring read "Scoring" section of each problem.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

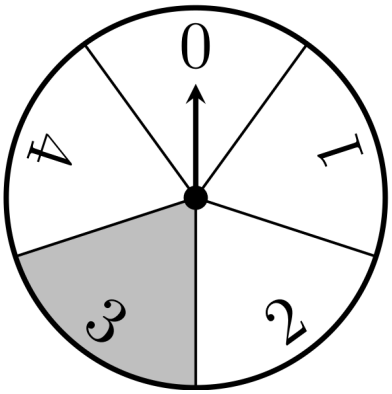
To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link.

For more information, on what feedback is given, read "Feedback" section of each problem.

Problem A. New Puzzle

Time limit: 1 second
Memory limit: 512 megabytes

Katya has a younger brother Denis, who loves puzzles. It seemed to him that he had already solved all the puzzles in the world. But, one day, returning from school with Katya, he noticed a vending machine with a puzzle that he has never seen before. He asked Katya to play with this machine and win a new puzzle.



Vending machine from the first example, before the start of the game.

The machine has a disk, divided into n sectors, rotating arrow and a button. When the button is pressed, the arrow starts to spin, and the player receives this prize, corresponding to the sector on which it stops. After observing the machine, Katya noticed that it is quite primitive, and after each click of the button the arrow moves k sectors clockwise. Also, Katya realized that the desired puzzle is at sector x, and it could be impossible to win. And then she asked you to help her.

Help Katya find out whether it is possible to win a puzzle, corresponding to sector x, if the arrow initially points to the sector s. And, if it is possible, calculate the minimal number of button clicks.

Input

A single line contains four integers n, k, s and x — the number of sectors in the machine, number of sectors the arrow moves after one click, number of a sector the arrow points initially to, and the number of a sector corresponding to desired puzzle, respectively ($2 \le n \le 20\,000$; $1 \le k < n$; $0 \le s, x < n$; $x \ne s$).

Output

Output one integer, the minimal number of clicks you need to make to get the desired puzzle, or «`-1`», if it is impossible.

Scoring

Subtask	Points	Additional Constraints	Scoring	Required subtasks
1	10	$k = 1$	subtask	—
2	15	$k = 2$	subtask	—
3	15	$s = 0$, $x = n - k$	subtask	—
4	30	It's possible to get desired puzzle	subtask	1 and 3
5	30	—	subtask	1, 2, 3, and 4

Feedback

You will be able to see the outcome of your solution on each test of each subtask.

Examples

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

Explanations

In the first example, the arrow will move through the sectors $0 \to 2 \to 4 \to 1 \to 3 \to \ldots$. Thus, after 4 button clicks you can get a puzzle from sector 3. The figure in the statement shows the machine from the first example before the game starts.

In the second example, the arrow will move only between the two sectors: 1 and 4. So it's impossible to get puzzle from sector 3.

Notice, that samples do not satisfy the additional constraints of the first, second, third, and fourth subtasks. Your solution will run on tests of the first four subtasks even if it does not pass the sample tests.

Problem B. Countdown

Time limit: 1 second
Memory limit: 512 megabytes

Zakhar and Oleg are classmates. They like to talk using voice messages in social networks. Most of the time, they talk about solving interesting problems. This time, Zakhar's eye met his old timer and he came up with an interesting problem for Oleg.

Zakhar sets his countdown timer to k seconds. Oleg doesn't know the value of k. At each moment in time the timer shows how many seconds are left until the end of the countdown (that is, at time 0 timer shows k, at time 1 it shows $k-1$ and so on). When the time is up, timer keeps showing number 0. Next, Zakhar looks at his timer n times and calculates the sum of numbers he saw at those moments. It is possible that Zakhar looks at his timer after it stops. In this case, Zakhar still adds the number on the timer to his sum — which is 0 in this case.

After that, Zakhar tells Oleg the sum s and n numbers — moments in time when he looked at the timer. Oleg wants to know the initial value of k.

While Oleg is thinking about this problem, Zakhar offers you to try and solve this problem yourself.

Input

First line contains two integers n and s — number of times, when Zakhar looked at his timer and the sum he got ($1 \le s \le 10^{18}$, $1 \le n \le 2 \cdot 10^5$).

Second line contains n integers $a_1, a_2 \ldots, a_n$ — moments in time, when Zakhar looked at his timer and added the numbers on the timer ($0 \le a_i \le 10^9$, $a_i < a_{i+1}$).

It is guaranteed that in all tests such k always exists.

Output

Print integer k — number of seconds, which Zakhar set his timer initially to.

Scoring

Subtask	Points	Constraints			Scoring	Required subtasks
		n	s	Additional		
1	30	$n \le 100$	$s \le 10^6$	$a_i < k \le 1000$	subtask	—
2	23	$n \le 1000$	$s \le 10^6$	$a_i = i - 1$	subtask	—
3	18	$n \le 1000$	$s \le 10^6$	$a_i \le 10^4$	subtask	1, 2
4	29	$n \le 2 \cdot 10^5$	$s \le 10^{18}$	—	subtask	1, 2, 3

Feedback

You will be able to see the outcome of your solution on each test of each subtask.

Examples

standard input	standard output
2 6 1 3	5
4 4 1 3 4 7	4

Explanations

In the first sample Zakhar set his timer to 5 seconds. After one second he saw number 4 and after three seconds he saw number 2. The sum of these two observations is 6.

In the second sample Zakhar sets his timer to 4 seconds. After 1, 3, 4 and 7 seconds from the timer's start, he sees 3, 1, 0 and 0, respectively. The sum of these values is 4.

Notice, that samples do not satisfy the additional constraints of the first and second subtasks. Your solution will be tested on tests of the first and second subtasks even if it does not pass the sample tests.

Problem C. Railroad

Time limit: 2 seconds
Memory limit: 512 megabytes

Ann got a great toy railroad for her birthday. It consists of stations and railways. Each railway connects exactly two stations. Two stations that are connected by a railway are called *adjacent*.

Due to strange circumstances some of the stations were initially colored and the others are not. Plus every uncolored station has **at most two** adjacent uncolored stations. Ann decided to paint all uncolored stations to make her toy more vivid. She has k different colors to fulfill her wish. Also Ann likes bright toys so she wants to color stations in such way that any two adjacent stations have different colors.

Help Ann color her toy as described above or find out that it's impossible. Remember that she doesn't want to repaint stations that have already been colored, so you can paint only initially uncolored stations!

Input

The first line contains three integers n, m and k — number of stations, railways and colors, respectively ($1 \le n \le 2 \cdot 10^5$; $0 \le m \le 4 \cdot 10^5$; $1 \le k \le n$).

The next line contains n integers that describe initial colors of stations. Colors are numbered from 1 to k. If a station doesn't have a color, the corresponding number is -1.

The next m lines contain railways' descriptions. Every line contains v_i and u_i — the stations, which are connected by the i-th railway ($1 \le v_i, u_i \le n$, $v_i \ne u_i$).

Each uncolored station has at most two uncolored neighbors.

Output

On the first line print "`Yes`", if described coloring exists or «`No`», otherwise.

If such coloring exists, print n integers, which describe colors of stations in the final coloring, on the next line. Remember, that you can not repaint stations which were colored initially. If multiple colorings exist, print any of them.

Scoring

Subtask	Points	Constraints	Score	Required subtasks
1	15	$n \le 9$; $m \le 50$; $k \le 5$	subtask	—
2	11	All stations are uncolored and there is at most one path between any two stations, $k \le 5$	subtask	—
3	11	All stations are uncolored, $k \le 5$	subtask	2
4	12	Each uncolored station has at most one uncolored neighbor	subtask	—
5	23	$k \le 5$	subtask	1, 2 and 3
6	28	—	subtask	1, 2, 3, 4 and 5

Path is a sequence of distinct railways, which connects two stations. Adjacent railways have a common station.

Feedback

You will be able to see the outcome of your solution on each test of each subtask.

Examples

standard input	standard output
2 1 2 -1 2 1 2	Yes 1 2
3 3 2 -1 -1 -1 1 2 2 3 3 1	No

Explanations

There are two stations connected by one railway in the first example. The first station isn't colored and the second one has color number 2. So painting the first station by color 1 results in correct coloring.

There is no coloring which fits the conditions in the second example, since it's impossible to paint cycle of length three with two colors.

Pay attention that samples don't satisfy constraints of the second, third and fourth subtasks. Your solution will be tested on this subtasks even if it fails sample tests.

Problem D. Math Candies

Time limit: 3 seconds
Memory limit: 512 megabytes

Little Max's most favorite activities are eating sweets and solving difficult mathematical riddles. The boy is happy when he finds a puzzle on a candy wrapper, which is a set of non-negative integers that he needs to sum up. Unfortunately, all the numbers on the wrapper are stuck together so that Max can't understand where one number ends, and where the next one starts.

The boy remembers that there are always exactly k numbers on a wrapper, none of them containing leading zeros. He decided to split this string of digits into numbers on his own. He just needs to insert $k - 1$ delimiters into the string. Max wants to get the most interesting puzzle, the one that has the maximum possible result.

Your task is to find the maximum total sum of k numbers that Max can get by splitting given string.

Input

The first line contains two integers n and k — the number of digits in the string on the wrapper and the number of integers, in which string must be split ($1 \le k \le n \le 5 \cdot 10^5$).

The second line contains a string of n decimal digits.

It's possible to split the string into integers without any leading zero.

Output

Output single integer — maximum possible result.

Scoring

Subtask	Points	Constraints	Scoring	Required subtasks
1	17	$n \le 9$	subtask	—
2	24	$n \le 100$	subtask	1
3	33	$n \le 1\,000$	subtask	1, 2
4	26	$n \le 5 \cdot 10^5$	subtask	1, 2, 3

Feedback

In subtasks 1, 2 and 3 the outcome for every test is shown.

In subtask 4 the outcome for the first failed test of the subtask is shown.

Examples

standard input	standard output
3 2 528	60
4 3 9050	95
5 3 07800	807

Explanations

In the first sample input splitting with the maximum sum is 52, 8;

In the second sample input splitting with the maximum sum is 90, 5, 0;

Consider all splitting for the third sample input:

1) 0, 7, 800 — valid splitting, sum is 807;

2) 0, 78, 00 — invalid splitting, there is a leading zero in 00;

3) 0, 780, 0 — valid splitting, sum is 780;

4) 07, 8, 00 — invalid splitting, there are leading zeros in 00 and 07;

5) 07, 80, 0 — invalid splitting, there is a leading zero in 07;

6) 078, 0, 0 — invalid splitting, there is a leading zero in 078.

Thus a splitting with the maximum sum is 0, 7, 800.

Problem E. K-th order statistic

Time limit: 2 seconds
Memory limit: 512 megabytes

Perhaps you are familiar with the problem of finding k-th order statistic in an array. Problem is to find out what number is at position k, if all numbers are sorted in ascending order. For example, the 2-nd order statistic from the numbers $[10, 2, 5]$ is 5, and n-th order statistic is always equal to the maximum of n numbers.

In this task, you need to come up with an expression that computes the k-th order statistic of n different positive integers using only the arithmetic operations "`+`", "`-`", "`*`", "`/`", the function returning the absolute value of the number `abs`, the variables named with the first n lowercase letters of the Latin alphabet, starting with '`a`', and also any integers not exceeding 10^9 by absolute value. You can use the unary minus, but you can not use the unary plus. Also, the expression can not contain any whitespace characters.

Verification of your expression will be conducted as follows: for each test, a certain number of sets of values of n variables are generated. These sets are determined before the start of the competition and are the same for all solutions of all participants. All values of variables are positive, different and do not exceed 1000.

The solution will be considered correct if the evaluation of the expression printed by your program gives the correct result for all selected sets of variables. All calculations are performed from left to right, but taking into account the priorities of the operators. The unary minus has the biggest priority. The operators "`*`" and "`/`" have the same priority, less than the unary minus, and operators "`+`" and "`-`" have the same priority, less than all of the above. For all generated sets of variables, your expression can not divide by zero, and all intermediate calculations must be integers from -10^9 to 10^9.

Input

The only line contains two integers n and k ($1 \le k \le n \le 26$).

Output

Output one line containing an expression that computes the k-th order statistics from n given variables. The length of the expression should not exceed 10^5. If there are several different correct expressions, you can print any of those.

Scoring

Subtask	Points	Constraints	Score	Required subtasks
1	17	$n \le 3$	subtask	—
2	32	$n \le 5$	subtask	1
3	27	$n \le 12$	subtask	1, 2
4	24	$n \le 26$	subtask	1, 2, 3

Feedback

In the first subtask you will be able to see the outcome of your solution on each test.

In subtasks 2, 3, and 4 you will be able to see the total score for the subtask and the outcome for the first failed test.

Example

standard input	standard output
2 1	abs(-a+b+(a*a-b*b)/abs(b-a))/2

Explanation

In the first example you need to create an expression which evaluates the minimum of a and b. Let's show the result of evaluation for different values of a and b:

- $a = 3$, $b = 5$. Result is $\frac{1}{2}|-3+5+\frac{9-25}{|3-5|}| = \frac{|5-3-8|}{2} = \frac{|-6|}{2} = 3 = a = \min(a, b)$.
- $a = 5$, $b = 2$. Result is $\frac{1}{2}|-5+2+\frac{25-4}{|5-2|}| = \frac{|-5+2+7|}{2} = \frac{|4|}{2} = 2$, which equals also to $\min(a, b)$.