Problem A. Simple Pattern

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a string encrypting a non-negative integer N: some characters of this string are equal to the corresponding digits of the number N, and the remaining characters are equal to the «*» character. It is allowed to replace such characters with any digits, but no leading zeros should appear in the record of the number N. One character «*» is replaced with one digit.

Your goal is to replace all «*» characters in the pattern with some digits in order to obtain a positive integer N with more than two different positive divisors.

Input

The input contains a single string of no more than 9 characters, the pattern of the number N. Each of the characters is either a digit or a «*» symbol.

It is guaranteed that it is possible to replace all characters «*» with some digits in such a way that the resulting number does not contain insignificant leading zeros.

Output

Print positive integer N that fits the pattern and has more than two different positive divisors. If there is no such N, output «-1».

Example

standard input	standard output
*00	300

Note

In the example, one can replace the asterisk with any digit except zero, because the resulting number N will be divisible by 100 in any case.

Following examples of source codes are the starting templates for most popular programming languages of the competition.

Note, a submission of these examples will get some points for the problem above.

Python

```
pattern = input()
result = pattern
print(result)
```

C++

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string pattern;
    cin >> pattern;

    string result = pattern;
```

```
    cout << result << endl;
    return 0;
}
```

C

```
#include <stdio.h>

int main() {
    char pattern[10];
    scanf("%s", pattern);

    char result[10];
    strcpy(result, pattern);

    printf("%s", result);
    return 0;
}
```

Java

```
import java.util.Scanner;

public class Main {
    public static void main(String[] arg) {
        Scanner in = new Scanner(System.in);
        String pattern = in.next();

        String result = pattern;

        System.out.println(result);
    }
}
```

Scoring

There are 50 tests in this task, each of them is scored with 2 points.

It is guaranteed that at least for 20% of tests the pattern of the number N ends with «*» character, at least for 20% of tests there is no «*» character in the input number N, and in 20% of tests the pattern N consists of two characters. The sample from the statement is included in these 50 tests.

Problem B. Going Home

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After new pandemic restrictions were imposed Alexey's University turned back to distance learning, so Alexey decided to go home and see his parents. Unfortunately, this trip takes a while as he first takes a direct flight to Moscow and then takes direct flight to his home city.

The country has n cities and m bidirectional flights. There may be an arbitrary number of bidirectional flights between any pair of cities. Moreover, it is allowed to have a bidirectional flight between a city and itself (as some people just like to flight and read books).

While seating on board Alexey thinks how great it would be to change flights. In particular, take a bidirectional flight between city A and city B, and a bidirectional flight between city B and city C and turn them into one bidirectional flight between city A and city C, while flights AB and BC just disappear. Here we use AB as a way to denote a bidirectional flight between city A and city B.

Alexey is really fond of solving competitive programming tasks and a long flight is a pretty boring thing. So he wonders, whether he can use an operation of replacing flights AB and BC with a single flight AC (deleting AB and BC) multiple times in order to achieve a situation that there is only one bidirectional flight in the whole country. While performing these operations in his head Alexey is allowed to pick any two different flights, in particular two different bidirectional flights between the same pair of cities or a flight from city to itself. If some operation removes flight AB and there are multiple such flights, flight AB is being removed as many times as it appears in this operation (i.e. once or twice).

Given the description of all flights, help Alexey decide whether he can apply his operation some number of times in order to keep only one bidirectional flight in the whole country.

Input

The first line of the input contains three integers n, m and p ($1 \le n, m \le 10^5$, $p \in \{0, 1\}$) — the number of cities in the country, the number of bidirectional flights and an indicator of whether you should print the sequence of operations or not.

The following m lines contain a pair of cities a_i and b_i each ($1 \le a_i, b_i \le n$) — the indices of cities that are connected by the i-th bidirectional flight.

Output

In the first line print «NO» if there is no way to get only one bidirectional flight in the whole country using Alexey's operation. Otherwise, print «YES».

If $p = 0$, you do not need to print the sequence of operations. If $p = 1$, you should print $m - 1$ lines. Each line should contain three space-separated integers a_i, b_i and c_i ($1 \le a_i, b_i, c_i \le n$), denoting indices of the cities. That means that for i-th operation Alexey replaces two bidirectional flights a_ib_i and b_ic_i with one bidirectional flight a_ic_i.

Examples

standard input	standard output
3 2 1 1 2 2 3	YES 3 2 1
3 3 1 1 2 2 3 1 3	YES 1 2 3 1 3 1
3 3 0 1 2 2 3 1 3	YES
4 6 1 1 2 2 3 3 4 4 1 1 3 2 4	NO

Scoring

Tests for this problem are divided into five groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **required** groups. Note that for some groups it is not required to pass sample tests.

Group	Points	Additional constraints		Req. groups	Comment
		n, m	p		
0	0	–	–	–	Sample tests.
1	10	$n, m \le 5$	–	0	
2	20	$n, m \le 1000$	–	0, 1	
3	15	$n, m \le 100\,000$	–	–	Each city initially has no more than 2 bidirectional flights connected to it (a flight between a city and itself is counted for this city 2 times)
4	20	$n, m \le 100\,000$	$p = 0$	–	You only need to determine «NO»/«YES», you should not print the sequence of operations in case of a positive answer
5	35	–	–	0 – 4	

Problem C. Strange Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Alex is a student, it's December, so he must get ready to his exams. Now he is planning to review all course materals.

Today Alex is planning to prepare to his physics exam. He is going to learn by heart an array a of n integers. Unfortunately Alex is not good in physics, so he always gets distracted with some strange activity. Now he chooses some segment of the array $a_l, a_{l+1}, \ldots, a_r$, and calculates the following strange sum for it:

1. Alex considers all pairs x, y such that $l \le x \le y \le r$.
2. For each such pair x, y Alex adds Hamming distance between segments $a_x, a_{x+1}, \ldots, a_y$ and $a_l, a_{l+1}, \ldots a_{l+(y-x)}$ to the strange sum.

Hamming distance between arrays $p_1, p_2, \ldots, p_k$ and $q_1, q_2, \ldots, q_k$ is the number of such indices i that $p_i \neq q_i$.

Your task is to help Alex to calculate the strange sum for several segments $[l, r]$.

Input

The first line of input contains two integers n and q ($1 \le n, q \le 2 \cdot 10^5$) — the length of the array and the number of segments that the strange sum must be calculated for.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the given array.

After that q queries follow, one on a line. Each line contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), that means that Alex must calculate the strange sum for the segment from l_i to r_i.

Output

The i-th line of output must contain one integer — the strange sum for the i-th query.

Examples

standard input	standard output
4 4 1 2 1 3 1 1 2 3 2 4 1 4	0 1 4 8
7 5 2 1 5 6 6 2 3 1 4 4 7 4 4 1 3 3 5	10 7 0 4 3

Note

Let us consider the last query in the first sample test. The Alex's array is $[1, 2, 1, 3]$, the segment requested is from 1 do 4.

Denote Hamming distance as h, them the answer is the sum of the following values:

- $h([1,2,1,3],[1,2,1,3]) = 0$
- $h([1,2,1],[1,2,1]) = 0$
- $h([2,1,3],[1,2,1]) = 3$
- $h([1,2],[1,2]) = 0$
- $h([2,1],[1,2]) = 2$
- $h([1,3],[1,2]) = 1$
- $h([1],[1]) = 0$
- $h([2],[1]) = 1$
- $h([1],[1]) = 0$
- $h([3],[1]) = 1$

So the strange sum is 8.

Scoring

Tests for this problem are divided into seven groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the required groups. Note that for some groups it is not required to pass sample tests.

Group	Points	Additional Constraints		Req. Groups	Comment
		n, q	a_i		
0	0			Sample tests	
1	10	$n, q \le 100$	–	0	
2	15	$n, q \le 1000$	–	0, 1	
3	15	$n, q \le 10\,000$	–	0, 1, 2	
4	15	$n, q \le 100\,000$	$a_i \le 2$	–	
5	15	$n, q \le 100\,000$	$a_i \le 50$	0, 4	
6	15	$n, q \le 100\,000$	–	0–5	
7	15	–	–	0–6	

Problem D. A third grade problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3.5 seconds
Memory limit:	256 megabytes

Let the number n be generated from ones using a additions and b multiplications, that is, there are exists a correct arithmetic expression with operations (,), $+$, $\cdot$, 1, which equals to n and has exactly a operations $+$ and b operations $\cdot$.

Examples of **correct** expressions:

- 1
- $1+1$
- $(1+1)\cdot(1+1)+1$
- $(1+(1+1)+(1))$

Examples of **incorrect** expressions:

- $(1$
- $((()$
- 111
- $(1+1)^{1+1+1+1}$

You are given q queries with two numbers n_i, a_i. For each query, you have to output the minimal number b_i, such that n_i can be generated using at most a_i additions and at most b_i multiplications. If there is no such b_i, output -1.

Input

The first line contains 2 numbers q, g ($1 \le q \le 10^6, 0 \le g \le 1$) — the number of queries and the flag – it indicates, whether this test belongs to the first group of tests (1 — test belongs to the first group or 0 — test does not belong).

The next q lines contain pairs of integers n_i, a_i ($1 \le n_i \le 5 \cdot 10^5, 0 \le a_i \le 5 \cdot 10^5$) — a number to be generated, and the maximum number of additions, allowed for this query.

Output

For each query, output in a separate line the minimal possible b_i, or -1 if it does not exist.

Example

standard input	standard output
10 0 10 1 12 4 10 5 17 7 9 4 12 5 2 1 7 4 12 5 18 7	-1 2 1 1 1 1 0 1 1 1

Note

Possible expressions for the task statement test:

1. It is impossible to generate 10 using 1 + and any number of $\cdot$.
2. $((1+1)\cdot(1+1))\cdot(1+1+1)=12$
3. $(1+1+1+1+1)\cdot(1+1)=10$
4. $(1+1+1+1)\cdot(1+1+1+1)+1=17$
5. $(1+1+1)\cdot(1+1+1)=9$
6. $(1+1+1+1)\cdot(1+1+1)=12$
7. $1+1=2$
8. $(1+1+1)\cdot(1+1)+1=7$
9. $(1+1+1+1)\cdot(1+1+1)=12$
10. $(1+1+1+1+1+1)\cdot(1+1+1)=18$

Scoring

The tests for this problem consist of eight groups. Points for each group are awarded only for passing all tests in the group and all tests of required previous groups. Please note, that passing the tests from the task statement is not required for some groups.

Group	Points	Add. constraints		Required groups	Comments
		n_i, a_i	b_i		
0	0	–	–	–	Tests from the task statement
1	6	–	$b_i = 1$ or answer -1	–	
2	12	$n_i \le 400$, $a \le 20$	–	0	
3	15	$a_i \le 20$	–	0, 2	
4	13	$a_i \le 30$	–	0, 2 – 3	
5	12	$a_i \le 40$	–	0, 2 – 4	
6	11	$a_i \le 60$	–	0, 2 – 5	
7	31	–	–	0 – 6	