Codeforces Beta Round #90

A. Epic Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Simon and Antisimon play a game. Initially each player receives one fixed positive integer that doesn't change throughout the game. Simon receives number a and Antisimon receives number b. They also have a heap of n stones. The players take turns to make a move and Simon starts. During a move a player should take from the heap the number of stones equal to the greatest common divisor of the fixed number he has received and the number of stones left in the heap. A player loses when he cannot take the required number of stones (i. e. the heap has *strictly* less stones left than one needs to take).

Your task is to determine by the given a, b and n who wins the game.

Input

The only string contains space-separated integers a, b and n $(1 \leq a, b, n \leq 100)$ — the fixed numbers Simon and Antisimon have received correspondingly and the initial number of stones in the pile.

Output

If Simon wins, print "0" (without the quotes), otherwise print "1" (without the quotes).

Sample test(s)

input
3 5 9
output
0

input
1 1 100
output
1

Note

The greatest common divisor of two non-negative integers a and b is such maximum positive integer k, that a is divisible by k without remainder and similarly, b is divisible by k without remainder. Let $gcd(a, b)$ represent the operation of calculating the greatest common divisor of numbers a and b. Specifically, $gcd(x, 0) = gcd(0, x) = x$.

In the first sample the game will go like that:

- Simon should take $gcd(3, 9) = 3$ stones from the heap. After his move the heap has 6 stones left.
- Antisimon should take $gcd(5, 6) = 1$ stone from the heap. After his move the heap has 5 stones left.
- Simon should take $gcd(3, 5) = 1$ stone from the heap. After his move the heap has 4 stones left.
- Antisimon should take $gcd(5, 4) = 1$ stone from the heap. After his move the heap has 3 stones left.
- Simon should take $gcd(3, 3) = 3$ stones from the heap. After his move the heap has 0 stones left.
- Antisimon should take $gcd(5, 0) = 5$ stones from the heap. As $0 < 5$, it is impossible and Antisimon loses.

In the second sample each player during each move takes one stone from the heap. As n is even, Antisimon takes the last stone and Simon can't make a move after that.

B. Before Exam

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya is about to take his first university exam in about several minutes. And it's not just some ordinary exam, it's on mathematical analysis. Of course, right now Vasya can only think of one thing: what the result of his talk with the examiner will be...

To prepare for the exam, one has to study proofs of n theorems. It is known that there will be k examination cards on the exam and each card contains $\lfloor \frac{n}{k} \rfloor$ distinct theorems. Besides, no theorem is mentioned in more than one card (that is, $n - k \cdot \lfloor \frac{n}{k} \rfloor$ theorems won't be mentioned in any card). During the exam several students may get the same card.

We do not know the exact way theorems are distributed by cards, however the students that took the exam before Vasya told him what theorems their cards contained. Vasya evaluates his *level of proficiency in the i-th theorem* by some number a_i. *The level of proficiency in some card* is the average of the levels of proficiency in the theorems that are included in the card. Now Vasya wants to know the minimally and maximally possible levels of his proficiency in the card he gets on the exam. Vasya wants to determine it by the data he has collected from other students. Unfortunately, Vasya has no time left to do the math and he asked you to help him.

Input

The first line contains two integers n and k $(1 \le k \le n \le 100)$ — the number of theorems and the number of cards correspondingly. The second line contains n integers a_i $(0 \le a_i \le 100)$, the i-th number $(1 \le i \le n)$ corresponds to Vasya's proficiency in the i-th theorem.

The third line contains number q $(0 \le q \le 100)$ — the number of people that have taken the exam before Vasya. Each of the following q lines contains the description of a student's card: $\lfloor \frac{n}{k} \rfloor$ integers from 1 to n inclusive. They are the numbers of theorems included in the card in the order in which they are enumerated in the input data. The numbers are given in an arbitrary order. It is guaranteed that the given cards are valid (that is, that all theorems in one card are different and that different people get cards that either don't contain the same theorems or coincide up to the theorems' permutation).

Output

Print two real numbers, representing Vasya's minimum and maximum proficiency in the card he will get on the exam. The absolute or relative error should not exceed 10^{-6}.

Sample test(s)

input
7 3 7 15 0 19 10 5 12 2 1 6 7 4
output
5.0000000000 15.5000000000

input
4 2 10 8 1 17 2 2 3 3 2
output
4.5000000000 13.5000000000

Note

Let's analyze the first sample. Vasya's proficiency in the cards whose content he already knows equals 6 and 15.5 correspondingly. The three theorems that are left are only enough to make one exam card. If we consider all possible variants of theorems included in the card we can see that in the best case scenario Vasya gets the card that contains theorems 4 and 7 (his proficiency would equal 15.5) and in the worst case scenario he gets theorems 3 and 5 (his proficiency would equal 5).

The $\lfloor x \rfloor$ operation denotes taking integer part of real number x (rounding down).

C. Education Reform

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yet another education system reform has been carried out in Berland recently. The innovations are as follows:

An academic year now consists of n days. Each day pupils study exactly one of m subjects, besides, each subject is studied for no more than one day. After the lessons of the i-th subject pupils get the home task that contains no less than a_i and no more than b_i exercises. Besides, each subject has a special attribute, the complexity (c_i). A school can make its own timetable, considering the following conditions are satisfied:

- the timetable should contain the subjects in the order of the complexity's strict increasing;
- each day, except for the first one, the task should contain either k times more exercises, or more by k compared to the previous day (more formally: let's call the number of home task exercises in the i-th day as x_i, then for each i ($1 < i \le n$): either $x_i = k + x_{i-1}$ or $x_i = k \cdot x_{i-1}$ must be true);
- the total number of exercises in all home tasks should be maximal possible.

All limitations are separately set for each school.

It turned out that in many cases a_i and b_i reach 10^{16} (however, as the Berland Minister of Education is famous for his love to half-measures, the value of $b_i - a_i$ doesn't exceed 100). That also happened in the Berland School №256. Nevertheless, you as the school's principal still have to work out the timetable for the next academic year...

Input

The first line contains three integers n, m, k ($1 \le n \le m \le 50$, $1 \le k \le 100$) which represent the number of days in an academic year, the number of subjects and the k parameter correspondingly. Each of the following m lines contains the description of a subject as three integers a_i, b_i, c_i ($1 \le a_i \le b_i \le 10^{16}$, $b_i - a_i \le 100$, $1 \le c_i \le 100$) — two limitations to the number of exercises on the i-th subject and the complexity of the i-th subject, correspondingly. Distinct subjects can have the same complexity. The subjects are numbered with integers from 1 to m.

Please do not use the `%lld` specificator to read or write 64-bit numbers in C++. It is preferred to use the `cin` stream or the `%I64d` specificator.

Output

If no valid solution exists, print the single word "`NO`" (without the quotes). Otherwise, the first line should contain the word "`YES`" (without the quotes) and the next n lines should contain any timetable that satisfies all the conditions. The $i + 1$-th line should contain two positive integers: the number of the subject to study on the i-th day and the number of home task exercises given for this subject. The timetable should contain exactly n subjects.

Sample test(s)

input
4 5 2 1 10 1 1 10 2 1 10 3 1 20 4 1 100 5

output
YES 2 8 3 10 4 20 5 40

input
3 4 3 1 3 1 2 4 4 2 3 3 2 2 2

output
NO

D. String Transformation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let s be a string whose length equals n. Its characters are numbered from 0 to n - 1, i and j are integers, $0 \le i < j < n$. Let's define function f as follows:

$$f(s, i, j) = s[i + 1 \ldots j - 1] + r(s[j \ldots n - 1]) + r(s[0 \ldots i]).$$

Here $s[p \ldots q]$ is a substring of string s, that starts in position p and ends in position q (inclusive); "+" is the string concatenation operator; $r(x)$ is a string resulting from writing the characters of the x string in the reverse order. If $j = i + 1$, then the substring $s[i + 1 \ldots j - 1]$ is considered empty.

You are given two strings a and b. Find such values of i and j, that $f(a, i, j) = b$. Number i should be maximally possible. If for this i there exists several valid values of j, choose the minimal j.

Input

The first two input lines are non-empty strings a and b correspondingly. Each string's length does not exceed 10^6 characters. The strings can contain any characters with ASCII codes from 32 to 126 inclusive.

Output

Print two integers i, j — the answer to the problem. If no solution exists, print "`-1 -1`" (without the quotes).

Sample test(s)

input
Die Polizei untersucht eine Straftat im IT-Bereich. untersucht eine Straftat.hciereB-TI mi ieziloP eiD
output
11 36

input
cbaaaa aaaabc
output
4 5

input
123342 3324212
output
-1 -1

E. Alternative Reality

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the year of 3000 travelling around parallel realities became a routine thing. However one has to take into consideration that travelling like that is highly dangerous as you never know beforehand where you're gonna get...

Little Vasya, for instance, found himself in a gaming reality and now he has to successfully complete all levels of a very weird game to get back. The gaming reality is a three-dimensional space where n points are given. The game has m levels and at the beginning of the i-th level the player is positioned at some plane Q_i that passes through the origin. On each level Vasya has to use special robots to construct and activate n powerful energy spheres of the equal radius with centers at the given points. The player chooses the radius of the spheres himself. The player has to spend R units of money to construct spheres whose radius equals R (consequently, one can construct spheres whose radius equals zero for free). Besides, once for each level a player can choose any point in space and release a laser ray from there, perpendicular to plane Q_i (this action costs nothing). The ray can either be directed towards the plane or from the plane. The spheres that share at least one point with the ray will be immediately activated. The level is considered completed if the player has managed to activate all spheres. Note that the centers of the spheres are the same for all m levels but the spheres do not remain: the player should construct them anew on each new level.

Help Vasya find out what minimum sum of money will be enough to complete each level.

Input

The first line contains two integers n and m ($1 \le n \le 900$, $1 \le m \le 100$) — the number of energetic spheres and the number of levels in the game correspondingly.

Each of the following n lines contains three integers x_i, y_i, z_i ($0 \le x_i, y_i, z_i \le 10^4$) — the coordinates of the center of the i-th sphere. Assume that these points do not change their positions throughout the game.

Then follow m lines, each containing three integers a_i, b_i, c_i ($0 \le a_i, b_i, c_i \le 100$, $a_i^2 + b_i^2 + c_i^2 > 0$). These numbers are the coefficients in the equation of plane Q_i ($a_i x + b_i y + c_i z = 0$), where the player is positioned at the beginning of the i-th level.

Output

Print m numbers, one per line: the i-th line should contain the minimum sum of money needed to complete the i-th level. The absolute or relative error should not exceed 10^{-6}.

Sample test(s)

input

```
4 1
0 0 0
0 1 0
1 0 0
1 1 0
0 0 1
```

output

```
0.7071067812
```

input

```
5 3
0 1 0
1 0 1
1 2 1
2 0 1
1 3 0
1 1 1
1 2 3
3 0 3
```

output

```
1.6329931619
1.6366341768
1.5411035007
```

input

```
2 1
0 20 0
0 0 0
0 10 0
```

output

```
0.0000000000
```

The only programming contests Web 2.0 platform