Bayan 2012-2013 Elimination Round (ACM ICPC Rules, English statements)

A. Old Peykan

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

There are n cities in the country where the Old Peykan lives. These cities are located on a straight line, we'll denote them from left to right as $c_1, c_2, \ldots, c_n$. The Old Peykan wants to travel from city c_1 to c_n using roads. There are $(n - 1)$ **one way** roads, the i-th road goes from city c_i to city c_{i+1} and is d_i kilometers long.

The Old Peykan travels 1 kilometer in 1 hour and consumes 1 liter of fuel during this time.

Each city c_i (except for the last city c_n) has a supply of s_i liters of fuel which immediately transfers to the Old Peykan if it passes the city or stays in it. This supply refreshes instantly k hours after it transfers. The Old Peykan can stay in a city for a while and fill its fuel tank many times.

Initially (at time zero) the Old Peykan is at city c_1 and s_1 liters of fuel is transferred to it's empty tank from c_1's supply. The Old Peykan's fuel tank capacity is unlimited. Old Peykan can not continue its travel if its tank is emptied strictly between two cities.

Find the minimum time the Old Peykan needs to reach city c_n.

Input

The first line of the input contains two space-separated integers m and k $(1 \le m, k \le 1000)$. The value m specifies the number of roads between cities which is equal to $n - 1$.

The next line contains m space-separated integers $d_1, d_2, \ldots, d_m$ $(1 \le d_i \le 1000)$ and the following line contains m space-separated integers $s_1, s_2, \ldots, s_m$ $(1 \le s_i \le 1000)$.

Output

In the only line of the output print a single integer — the minimum time required for The Old Peykan to reach city c_n from city c_1.

Examples

input
4 6 1 2 5 2 2 3 3 4
output
10

input
2 3 5 6 5 5
output
14

Note

In the second sample above, the Old Peykan stays in c_1 for 3 hours.

B. Friends

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have n friends and you want to take m pictures of them. Exactly two of your friends should appear in each picture and no two pictures should contain the same pair of your friends. So if you have $n = 3$ friends you can take 3 different pictures, each containing a pair of your friends.

Each of your friends has an attractiveness level which is specified by the integer number a_i for the i-th friend. You know that the attractiveness of a picture containing the i-th and the j-th friends is equal to the exclusive-or (xor operation) of integers a_i and a_j.

You want to take pictures in a way that the total sum of attractiveness of your pictures is maximized. You have to calculate this value. Since the result may not fit in a 32-bit integer number, print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line of input contains two integers n and m $(1 \le n \le 5 \cdot 10^4;\ 0 \le m \le \frac{n \cdot (n-1)}{2})$ — the number of friends and the number of pictures that you want to take.

Next line contains n space-separated integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$ — the values of attractiveness of the friends.

Output

The only line of output should contain an integer — the optimal total sum of attractiveness of your pictures.

Examples

input
3 1 1 2 3
output
3

input
3 2 1 2 3
output
5

input
3 3 1 2 3
output
6

C. Mirror Box

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

Mirror Box is a name of a popular game in the Iranian National Amusement Park (INAP). There is a wooden box, 10^5 cm long and 100 cm high in this game. Some parts of the box's ceiling and floor are covered by mirrors. There are two negligibly small holes in the opposite sides of the box at heights h_l and h_r centimeters above the floor. The picture below shows what the box looks like.

In the game, you will be given a laser gun to shoot once. The laser beam must enter from one hole and exit from the other one. Each mirror has a preset number v_i, which shows the number of points players gain if their laser beam hits that mirror. Also — to make things even funnier — the beam must **not** hit any mirror more than once.

Given the information about the box, your task is to find the maximum score a player may gain. Please note that the reflection obeys the law "the angle of incidence equals the angle of reflection".

Input

The first line of the input contains three space-separated integers h_l, h_r, n ($0 < h_l, h_r < 100$, $0 \le n \le 100$) — the heights of the holes and the number of the mirrors.

Next n lines contain the descriptions of the mirrors. The i-th line contains space-separated v_i, c_i, a_i, b_i; the integer v_i ($1 \le v_i \le 1000$) is the score for the i-th mirror; the character c_i denotes i-th mirror's position — the mirror is on the ceiling if c_i equals "T" and on the floor if c_i equals "F"; integers a_i and b_i ($0 \le a_i < b_i \le 10^5$) represent the x-coordinates of the beginning and the end of the mirror.

No two mirrors will share a common point. Consider that the x coordinate increases in the direction from left to right, so the border with the hole at height h_l has the x coordinate equal to 0 and the border with the hole at height h_r has the x coordinate equal to 10^5.

Output

The only line of output should contain a single integer — the maximum possible score a player could gain.

Examples

input
50 50 7 10 F 1 80000 20 T 1 80000 30 T 81000 82000 40 T 83000 84000 50 T 85000 86000 60 T 87000 88000 70 F 81000 89000
output
100

input
80 72 9 15 T 8210 15679 10 F 11940 22399 50 T 30600 44789 50 F 32090 36579 5 F 45520 48519 120 F 49250 55229 8 F 59700 80609 35 T 61940 64939 2 T 92540 97769
output
120

Note

The second sample is depicted above. The red beam gets $10 + 50 + 5 + 35 + 8 + 2 = 110$ points and the blue one gets 120.

The red beam on the picture given in the statement shows how the laser beam can go approximately, this is just illustration how the laser beam can gain score. So for the second sample there is no such beam that gain score 110.

D. Numbers

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a sequence of n distinct integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$. You want to remove some integers in such a way that the resulting sequence of integers satisfies the following three conditions:

1. the resulting sequence is not empty;
2. the exclusive or (xor operation) of all the integers in the resulting sequence equals 0;
3. if you write all the integers of the resulting sequence (from beginning to the end) in a row in the decimal numeral system and without any spaces, the written number is divisible by p.

You are given the sequence of n integers a and a prime number p, find a way to satisfy the described conditions.

Input

The first line of the input contains two integers n and p $(1 \le n, p \le 50000)$. Next line contains n space-separated distinct integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$.

It is guaranteed that p is a prime number.

Output

If there is no solution for the given input, print "No" (without quotes) in the only line of the output.

Otherwise print "Yes" in the first line of output. The second line should contain an integer k $(k > 0)$ specifying the number of remaining elements and the third line should contain k distinct integers $x_1, x_2, \ldots, x_k$ $(1 \le x_i \le n)$. These integers mean that you should remove all integers from the sequence except integers $a_{x_1}, a_{x_2}, \ldots, a_{x_k}$ to satisfy the described conditions.

If there are multiple solutions, any of them will be accepted.

Examples

input
3 3 1 2 3

output
Yes 3 1 2 3

input
3 5 1 2 3

output
No

E. Flights

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

LiLand is a country, consisting of n cities. The cities are numbered from 1 to n. The country is well known because it has a very strange transportation system. There are many one-way flights that make it possible to travel between the cities, but the flights are arranged in a way that once you leave a city you will never be able to return to that city again.

Previously each flight took exactly one hour, but recently Lily has become the new manager of transportation system and she wants to change the duration of some flights. Specifically, she wants to change the duration of some flights to exactly 2 hours in such a way that all trips from city 1 to city n take the same time regardless of their path.

Your task is to help Lily to change the duration of flights.

Input

First line of the input contains two integer numbers n and m $(2 \leq n \leq 1000; 1 \leq m \leq 5000)$ specifying the number of cities and the number of flights.

Each of the next m lines contains two integers a_i and b_i $(1 \leq a_i < b_i \leq n)$ specifying a one-directional flight from city a_i to city b_i. It is guaranteed that there exists a way to travel from city number 1 to city number n using the given flights. It is guaranteed that there is no sequence of flights that forms a cyclical path and no two flights are between the same pair of cities.

Output

If it is impossible for Lily to do her task, print "No" (without quotes) on the only line of the output.

Otherwise print "Yes" (without quotes) on the first line of output, then print an integer ans_i $(1 \leq ans_i \leq 2)$ to each of the next m lines being the duration of flights in new transportation system. You should print these numbers in the order that flights are given in the input.

If there are multiple solutions for the input, output any of them.

Examples

input
3 3 1 2 2 3 1 3
output
Yes 1 1 2

input
4 4 1 2 2 3 3 4 1 4
output
No

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

F. Race

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

The Old City is a rectangular city represented as an $m \times n$ grid of blocks. This city contains many buildings, straight two-way streets and junctions. Each junction and each building is exactly one block. All the streets have width of one block and are either vertical or horizontal. There is a junction on both sides of each street. We call two blocks adjacent if and only if they share a common side. No two blocks of different streets are adjacent and no two junctions are adjacent.

There is an annual festival and as a part of it, The Old Peykan follows a special path in the city. This path starts from a block in a street, continues with many junctions and ends in a block of some street. For each street block, we know how much time it takes for the Old Peykan to go from this block to an adjacent block. Also the Old Peykan can go from each junction to its adjacent street blocks in one minute. Of course Old Peykan can't go to building blocks.

We know the initial position of the Old Peykan and the sequence of junctions that it passes to reach its destination. After passing all the junctions and reaching the destination, it will stay there forever. Your task is to find out where will the Old Peykan be k minutes after it starts moving. Consider that The Old Peykan always follows the shortest path that passes through the given sequence of junctions and reaches the destination.

Note that the Old Peykan may visit some blocks more than once.

Input

The first line of input contains three integers m, n and k ($3 \le m, n \le 100, 1 \le k \le 100000$). Next m lines are representing the city's map. Each of them containts n characters, each character is a block:

- Character "#" represents a building.
- Digits "1", "2", ..., "9" represent a block of an street and this digit means the number of minutes it takes for the Old Peykan to pass this block.
- Characters "a", "b", ..., "z" means that this block is a junction and this character is it's name. All the junction names are unique.

Consider that all blocks have the coordinates: the j-th in the i-th line have coordinates (i, j) $(1 \le i \le m, 1 \le j \le n)$.

The $(m + 2)$th line contains two integers r_s and c_s $(1 \le r_s \le m, 1 \le c_s \le n)$, string s and another two integers r_e and c_e $(1 \le r_e \le m, 1 \le c_e \le n)$. The path starts from block (r_s, c_s), continues through junctions in the order that is specified by s and will end in block (r_e, c_e). Length of s is between 1 and 1000.

It's guaranteed that string s denotes a correct path from the start position to the end position and string s doesn't contain two consecutive equal letters. Also start position (r_s, c_s) and the end position (r_e, c_e) are street blocks.

Output

In a single line print two integers r_f and c_f — (r_f, c_f) being the position of the Old Peykan after exactly k minutes.

Examples

input
3 10 12 ########## #z1a1111b# ########## 2 3 ab 2 8
output
2 8

input
10 3 5 ### #w# #1# #a# #1# #1# #1# #1# #b# ### 3 2 abababababababab 6 2
output
8 2

input

```
3 10 6
##########
#z1a1311b#
##########
2 3 ab 2 8
```

output

```
2 7
```

G. Challenging Balloons

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

Martha — as a professional problemsetter — proposed a problem for a world-class contest. This is the problem statement:

Tomorrow is Nadia's birthday, and Bardia (her brother) is assigned to make the balloons ready!

There are n balloons (initially empty) that are tied to a straight line on certain positions $x_1, x_2, \ldots, x_n$. Bardia inflates the balloons from left to right. As a result, i-th balloon gets bigger and bigger until its radius reaches the pressure endurance p_i or it touches another previously-inflated balloon.

While Bardia was busy with the balloons, he wondered "What will be the sum of radius of balloons after all of the balloons are inflated?". Being a nerdy type of guy, he is now thinking about the problem instead of preparing his sister's birthday. Calculate the answer to Bardia's problem so that Nadia's birthday won't be balloon-less.

Artha — Martha's student — claimed his solution got accepted. Martha (being his teacher for a long time!) knew he couldn't have solved the problem for **real** and thus thinks there is something wrong with the testcases. Artha isn't anyhow logical, which means there is no way for Martha to explain the wrong point in his algorithm. So, the only way is to find a testcase to prove him wrong!

Artha's pseudo-code is shown below:

You should output a small testcase for the problem such that Artha's algorithm is incorrect. The algorithm's output is considered correct if it differs from the correct value by no more than 1.

Input

Please pay attention! No input will be given to your program for this problem. So you do not have to read from the input anything.

Output

You should output the generated small testcase (which Artha's solution doesn't get it right). It should be in the following format:

- First line must contain the only number n ($1 \le n \le 500$).
- The i-th of the next n lines should contain the description of the i-th balloon — two space-separated integers x_i, p_i ($1 \le p_i \le 10^6, 0 \le x_1 < x_2 < \ldots < x_n \le 10^6$).

Examples

Note

The testcase depicted in the figure above (just showing how output should be formatted):

```
4
0 9
6 3
12 7
17 1
```