The 2019 ICPC Caribbean Local Contests

Real Contest Problem Set

Document composed by 13 pages (not including this cover)

Problem set developers

Marcelo Fornet (NEAR, Cuba)

Carlos Joa (INTEC, República Dominicana)

Reynaldo Gil (Datys, Cuba)

Carlos Toribio (Google, República Dominicana)

Elio Govea (UPR, Cuba)

Rubén Alcolea (UCI, Cuba)

Ernesto Peña (UO, Cuba)

José Carlos Gutierrez (Universität Bremen, Cuba)

Norge Vizcay (Cuba)

Abdel Rodríguez (Cuba)

Rafael Hernández (Cuba)

Aurora Gil (Cuba)

Alexander Bestard (UO, Cuba)

Ariel Cruz (UH, Cuba)

September 21st, 2019

Problem A. String in the tree I

A new family of trees has been found. These peculiar trees have exactly one fruit in each leaf and each bifurcation, similar to a tree data structure. The weirdest characteristic of these plants is that they are able to produce up to 62 fruits of different types (represented with an English lowercase character, uppercase character or numeric digit). In order to classify them it is needed a function that, given a tree (with all its fruits) and a pattern (a sequence of fruits), counts how many simple paths in the tree are equal to the pattern. A simple path is defined as a sequence of nodes where two adjacent nodes in the sequence are adjacent in the tree and it does not contain repeated nodes. We say a simple path is equal to a pattern if when we replace the number of the node in the path (the sequence) by the fruit it contains, we obtain a sequence equal to the pattern.

Input

The first line contains two integers N and L ($1 \leq N, L \leq 10^3$) indicating the number of nodes and the length of the pattern respectively. The second line has a string S ($|S| = N$) where S_i indicates the fruits contained in the node i. The third line contains the pattern P ($|P| = L$) to find. Then next $N - 1$ lines contains the structure of the tree with each line having two integers a and b ($0 \leq a, b < N$) indicating that these two nodes are adjacent.

Output

Print a single integer indicating the number of times the pattern appears in a simple path of the tree.

Example

standard input	standard output
4 3 baaa aba 0 1 0 2 0 3	6
15 4 ababbabaaabbabb abab 0 1 1 2 2 3 3 4 2 5 5 6 2 7 0 8 8 9 8 10 10 11 10 12 12 13 12 14	3

Problem B. String in the tree II

A new family of trees has been found. These peculiar trees have exactly one fruit in each leaf and each bifurcation, similar to a tree data structure. The weirdest characteristic of these plants is that they are able to produce up to 62 fruits of different types (represented with an English lowercase character, uppercase character or numeric digit). In order to classify them it is needed a function that, given a tree (with all its fruits) and a pattern (a sequence of fruits), counts how many simple paths in the tree are equal to the pattern. A simple path is defined as a sequence of nodes where two adjacent nodes in the sequence are adjacent in the tree and it does not contain repeated nodes. We say a simple path is equal to a pattern if when we replace the number of the node in the path (the sequence) by the fruit it contains, we obtain a sequence equal to the pattern.

Input

The first line contains two integers N and L ($1 \le N, L \le 10^6$) indicating the number of nodes and the length of the pattern respectively. The second line has a string S ($|S| = N$) where S_i indicates the fruits contained in the node i. The third line contains the pattern P ($|P| = L$) to find. Then next $N - 1$ lines contains the structure of the tree with each line having two integers a and b ($0 \le a, b < N$) indicating that these two nodes are adjacent.

Output

Print a single integer indicating the number of times the pattern appears in a simple path of the tree.

Example

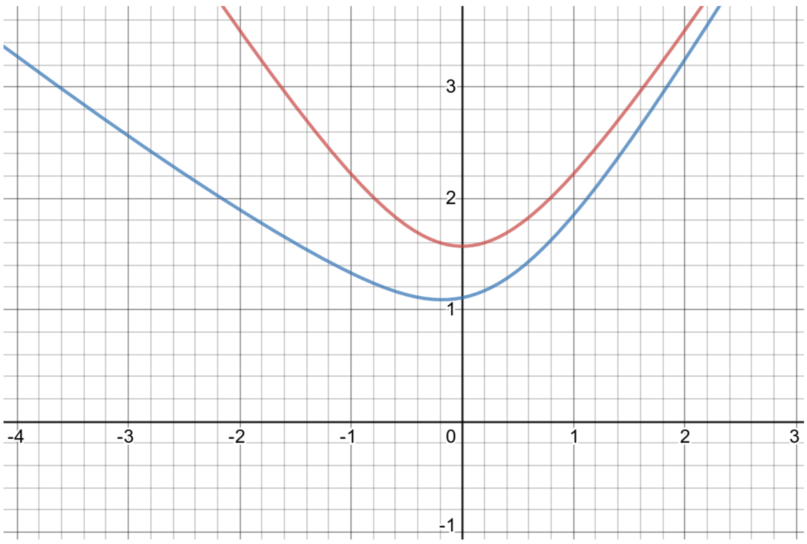
standard input	standard output
4 3 baaa aba 0 1 0 2 0 3	6
15 4 ababbabaaabbabb abab 0 1 1 2 2 3 3 4 2 5 5 6 2 7 0 8 8 9 8 10 10 11 10 12 12 13 12 14	3

Problem C. Chinese curves

You are in a world with curves of the form $f(x) = \arctan(e^x + a)\sqrt{b \cdot x^2 + c}$, where a, b and c are integers. In this world, there are two types of queries:

1. Given three integers a, b and c, create a new curve with parameters a, b and c.
2. Given an integer p, determine the minimum value of $f(p)$ among all curves.

The following image corresponds to the sample test case, where the blue curve is the last curve added. Here is an explanation of the sample input:



1. From the first line of input, mask m is intially set to 0.
2. Query 1 creates a curve with parameter $a = 1000$, $b = 1$, $c = 1$ (since mask $m = 0$).
3. Query 2 asks for the minimum value of $f(1)$ among all curves. Since there is only one curve, answer is simply $\arctan(e^1 + 1000)\sqrt{1^2 + 1}$, which is about 2.22.
4. After answering query 2, mask m becomes $\lfloor 2.22 \rfloor \oplus 0 = 2$.
5. Query 3 creates a curve with parameters a, b, and c equal to $3 \oplus 2 = 1$.
6. As $p = 3 \oplus 2 = 1$, query 4 asks for the minimum value of $f(1)$ among the two existing curves. From the image, we can see that the last added curve yields a minimum value of about 1.85.
7. After answering query 4, mask m is updated to $\lfloor 1.85 \rfloor \oplus 2 = 3$.
8. The last query asks to find the minimum value of $f(41 \oplus 3) = f(42)$ among the two curves.

Input

In the first line, there are two integers q and m, where q is the number of queries to follow ($1 \le q \le 10^5$) and m is a mask ($0 \le m \le 10^5$). Each of the following q lines contains a query using one of these two formats:

1. $a\ b\ c$: if the query is of the first type, where $0 \le b \le 10^6$, $0 \le a, c \le 10^{18}$.
2. p: if the query is of the second type, where $0 \le p \le 10^6$.

In the input, the values of a, b, c and p are not given directly. Instead, you should restore them from the inputs a', b', c' and p' by computing $a = a' \oplus m$, $b = b' \oplus m$, $c = c' \oplus m$, and $p = p' \oplus m$, where m is the current value of mask m and $\oplus$ is the bitwise xor operator (available as the ˆ symbol in many computer programming languages). After each query of type 2, the mask m changes to $\lfloor s \rfloor \oplus m$, where s is the answer to the query, $\oplus$ is the bitwise xor operator, and $\lfloor s \rfloor$ is the floor function that yields the integer part of number s. It is guaranteed that the first query is of type 1.

Output

For each query of type 2, output a line with the required answer, rounding to two digits of precision after the decimal point.

Example

standard input	standard output
5 0 1 1000 1 1 2 1 1 3 3 3 2 3 2 41	2.22 1.85 65.99

Problem D. Set

Find a set R such that:

- All its elements are positive integers.
- It has exactly N elements.
- None of its elements is greater than L.
- Sum of its elements is S.
- The total amount of elements $X \in R$, such that $X - 1 \notin R$ is less than 100.

Note: On the second test case the chosen set was: $3, 4, 5, 7, 9, 10$

Input

Three integers N ($1 \leq N \leq 10^9$), L ($1 \leq L \leq 10^9$), S ($1 \leq S \leq 10^{18}$).

Output

One line with two integers A and B that represent the number of elements in the set such that the predecessor/successor doesn't belong to the set, respectively.
One line with A integers sorted increasingly: these are all the elements in the set whose predecessor does not belong to the set.
One line with B integers sorted increasingly: these are all the elements in the set whose successor does not belong to the set.
If there is no solution, print `"-1"` without quotes.

Example

standard input	standard output
1 1 1	1 1 1 1
6 15 38	3 3 3 7 9 5 7 10

Problem E. Rotate circles

There is a board of circles with n rows and m columns. Each circle is divided in four equal regions. Each of the four regions is colored with distinct colors. You can select a circle and rotate it 90 degrees clockwise. A state of the board is beautiful if the two adjacent regions of every pair of neighboring circles have the same color. Given the initial state of the board, how many different beautiful states you can reach by performing the action as many times as you want. Two states are considered different if there is at least one circle that has different rotations in both states.

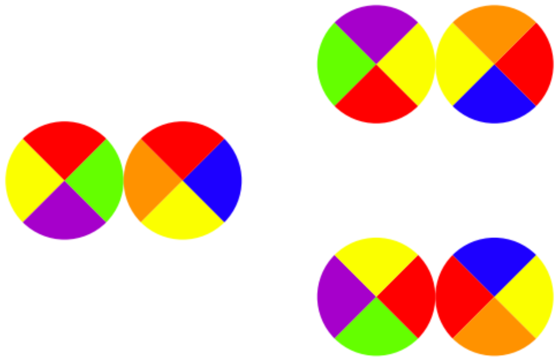


Image for second test case. In the left of the image is the state of the circles as described in the input. In the right are the two beautiful states. Note that colors are [r]ed, [b]lue, [p]urple, [y]ellow and [o]range.

Input

First line contains two integers n, m ($1 \leq n, m \leq 10$) number of rows and columns of the board respectively. Next n lines contains description of each row of the board. Each line contains $4 \cdot m$ characters, every group of 4 characters is the description of a circle. Each character c_i is lowercase English letter and represents a color of one of the four regions of the circle. Regions are described in clockwise order starting from upper region. See image for more details.

Output

Print one integer, the number of different beautiful states.

Example

standard input	standard output
2 2 r b k g r b k g r b k g k g r b	4
1 2 r g p y r b y o	2

Problem F. Sigma

Fito and María were playing with a map of the kingdom of Sigma. The map consists of N cities and roads between pairs of cities (roads can be used in any direction). At the start of the game, the kids selected two cities: Fito chose city F while María picked city M. The kids were very careful in their choices: the cities were distinct and were not connected by a road. They decided to solve differet challenges. Given K, Fito wants to find at most K cities (different from F and M) such that if they are removed from the map along with the roads incident to the removed cities, then there isn't any route between city F and city M. On the other hand, María wants to find at least $K + 1$ routes such that:

1. All the routes start at city F.
2. All the routes end at city M.
3. There are no routes with repeated cities.
4. For every pair of routes the only cities in common are F and M (there are no repeated internal cities among the routes).

Help any of the children find the answer to his/her question. Note: There may exist multiples roads between the same pair of cities, and roads between a city and itself.

Input

One line with five integers N, T, F, M, K ($1 \leq N, T, K \leq 10^5$, $1 \leq F, M \leq N$). The number of cities, the number of roads, the city selected by Fito, the city selected by María and the picked number respectively. Then T lines with two integers u, v ($1 \leq u, v \leq N$) denoting an undirected road between cities u y v.

Output

If you find a solution for Fito, print in the first line the word "FITO" without quotes. In the second line print a number C ($0 \leq C \leq K$), the amount of cities that Fito has to remove to solve his challenge. Then print a line with C positive integers u ($1 \leq u \leq N$), the cities he will remove. Any valid answer will be accepted. If you find a solution for María's challenge, print in the first line the word "MARIA" without quotes. In the next line print number C ($K + 1 \leq C \leq N$), the amount of routes María will pick. In the following C lines print each route using the following format. First print the number X of cities in the route and next X integers u ($1 \leq u \leq N$), the cities. Each route must obey restrictions stated above. Read second test case for more details. Any valid answer will be accepted. You just have to find an answer for at most one of the two children. In case none of the challenges has any solution, print "NONE" without quotes.

Example

standard input	standard output
4 4 1 4 2 1 2 1 3 2 4 3 4	FITO 2 2 3
4 4 1 4 1 1 2 1 3 2 4 3 4	MARIA 2 3 1 2 4 3 1 3 4

Problem G. Three numbers

You are given three digits A, B, C (not necessarily different). Find three digits X, Y, Z satisfying the following restrictions:

1. X, Y, Z, are mutually different,
2. $X \neq A$
3. $Y \neq B$
4. $Z \neq C$
5. $1 \leq X, Y, Z \leq 9$

Input

Only one line with three digits $1 \leq A, B, C \leq 9$.

Output

Three digits X, Y, Z satisfying the restrictions of the problem. (Any triple satisfying the restrictions will be acepted).

Example

standard input	standard output
9 8 7	1 2 3
1 1 1	9 6 3

Problem H. Queries with recurrences

Problems involving queries and recurrences are very common in ICPC. In this problem, we are going to mix a little both topics. Given the recurrence $F_n = a \times F_{n-1} + b \times F_{n-2} + k$ and an array A with N values, write a program to perform Q queries with the following format:

- $1\ a\ b\ k$: Increase each value within the range $[a, b]$ with the value F_n, where $n = a \times b$ ($1 \leq a \leq b \leq N$, $1 \leq k \leq 2N$).
- $2\ a\ b$: Return the sum of values $A_a + A_{a+1} + ... + A_b$. This value can be very big, so print the answer modulo 1000000007. ($1 \leq a \leq b \leq N$).

Input

The input contains in the first line two integer numbers: N ($1 \leq N \leq 10^5$) and Q ($1 \leq Q \leq 10^5$). The following Q lines contain the information of the queries, as it was explained above. The recurrence always has two base cases: $F[0] = 0$ and $F[1] = k$. In addition, at the beginning each value of the array $A[i] = 0$.

Output

For each query of kind 2, print the corresponding result.

Example

standard input	standard output
10 5 1 1 2 1 2 1 6 1 1 3 2 2 1 4 2 2 6	4 40 26
10 5 1 3 5 15 2 1 10 1 4 6 5 2 1 4 2 6 10	854873249 469363035 566114207

Problem I. Build me a fence

Given a list with n positive integers $(a_1, a_2, \cdots, a_n)$, find n points $(p_1, p_2, \cdots, p_n)$ that form a closed polygon such that:

1. Distance between points p_i and p_{i+1} is equal to a_i if $1 \leq i < n$.
2. Distance between points p_n and p_1 is equal to a_n.
3. Polygon has positive area (ie, it is a non degenerate polygon).
4. Non consecutive sides should not intersect.

Input

A line with one integer n $(3 \leq n \leq 100)$, number of sides of the polygon. Then a line with n positive integers $(1 \leq a_i \leq 100)$.

Output

If there is no valid polygon satisfying the constraints print "No" without quotes. Otherwise print "Si" without quotes, and next print n lines with vertex coordinates x, y $(-10^6 \leq x, y \leq 10^6)$ in consecutive order. Any polygon that satisfies problem constraints will be considered valid. Real distance between printed vertex and expected distance should be less than 10^{-6}.

Example

standard input	standard output
4 1 1 1 1	Si 0 0 0 1 1 1 1 0
4 1 1 3 1	No

Problem J. Elevator

In a building with n floors some packages have arrived for each department and it is necessary to carry them up to their respective recipients. There are m packages at floor 0 and each of them has annotated the floor to which it must be sent. For security issues, the elevator can carry at most k packages at the same time. A delivery consists in the following process: select the maximum floor of the delivery, move the elevator to that floor and finally deliver the packages to the current floor and the ones below it (while the elevator goes down to floor 0). In each delivery of the packages, the elevator must end up at floor 0. Your task is to find out the optimal strategy to carry all the packages with the minimum number of deliveries.

Input

A line with three integers n, m, k, the amount of floors, the amount of packages to be delivered and the maximum amount of packages that can be carried in the elevator in each moment respectively. Then a line with m integers. The integer a_i is the floor to which the i-th package must be sent. Limits: $1 \leq n, m, k \leq 100000$, $1 \leq a_i \leq n$.

Output

An integer that denotes the minimum number of deliveries to carry up all the packages as it was explained above.

Example

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

Problem K. Polygon

As you all must know, ICPC is one of the most prestigious competitions in the field of computer science. This year, team Limscape is one of the strongest teams in the Caribbean Region, and they are training harder in order to advance to the 2020 ICPC World Finals. They recently took part in the Caribbean Local Contest, where they solved all but one problem and ended up tied in number of solved problems with teams UZ**, Frickyfox and VLCU.h. Now they want to solve it off-contest, and they ask for your help, due to your knowledge in Competitive Programming field. The problem was the following: You are given a simple polygon (it doesn't intersect or touch with itself) consisting of n vertices. A triangulation of this polygon is a subdivision in $n-2$ triangles, defined by a set of non-intersecting diagonals, in which the vertices of the triangles are also vertices of the polygon. Each vertex i has a special value v_i associated. We define the cost of a triangle as the bitwise XOR operation of the special values of vertices that form the triangle, and the cost of a triangulation as the sum of the costs of the triangles that compose it. We are interested in calculating the minimum possible value of a triangulation for such polygon.

Input

The input consist of a line containing the integer n, the number of vertices that compose the given polygon ($3 \le n \le 300$). The following n lines contains 2 space separated integers x_i and y_i, describing the coordinates of the vertices in counterclockwise order ($-1000 \le x_i, y_i \le 1000$). Finally follows a line with n space separated integers v_i, the special value for each vertex, in the same order as they are given in the input ($0 \le v_i < 2^{10}$).

Output

You should print a single line with the minimum cost of a triangulation for the given polygon.

Example

standard input	standard output
4 0 0 1 0 1 1 0 1 1 2 3 2	0