Problem A. String Transformations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

We are given two strings u and v composed of letters `a` and `b`. Our goal is to transform the string u into v using the following *swap* operation: choose two *disjoint* fragments `ab` and `ba` in the first string and swap them. Can u be transformed into v using a sequence of swap operations?

Input

The first line of input contains one integer n ($2 \le n \le 1\,000\,000$), the length of the strings. Two lines follow, each containing a string composed of n letters `a` and/or `b`. The first line describes the string u and the second line describes v. You can assume that the two strings differ.

Output

The first line of output should contain one word `TAK` (Polish for *yes*) or `NIE` (Polish for *no*) indicating if u can be transformed into v using the swap operations.

If the answer is positive **and** $n \le 4\,000$, your program should output an example sequence of swap operations. The first line should contain one integer m ($1 \le m \le 1\,000\,000$), the number of operations. Each of the following m lines should contain two integers a_i, b_i ($1 \le a_i, b_i \le n-1$), the positions of the starting letters of the fragments being swapped: a_i denotes the position of `ab` and b_i denotes the position of `ba`.

If there is more than one solution, your program should output any one of them. In particular, you do not need to minimize the number of operations, that is, the number m.

Examples

standard input	*standard output*
6 aabbaa baaaab	TAK 2 2 4 1 5
6 aaabbb ababab	NIE

Problem B. Shuffle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Byteasar gave a deck of n cards to his son Bytie as a gift. The cards in the deck are numbered 1 through n.

Bytie got very excited about the gift. For the whole evening he played with the cards and shuffled them. At some point he even acquired such a skill in shuffling that each time he obtained the same arrangement, i.e. the card from the k-th position (for $1 \leq k \leq n$) always went to the same a_k-th position (for $1 \leq a_k \leq n$).

Late in the evening Bytie's father went to the room and told him that it was time to sleep. Bytie asked his father to show Bytie before he goes to sleep how to shuffle the cards. Byteasar shuffled the cards so that a card from the k-th position went to the b_k-th position (for $1 \leq k, b_k \leq n$).

Bytie was really amazed by how Byteasar shuffled the cards. He would love to learn his father's shuffle! Unfortunately, Bytie is still very young and he is not as skilled in shuffling as his father. However, he got an idea that maybe he could repeat his shuffle a number of times to obtain the same arrangement as after his father's shuffle.

Now the boy has problems with falling asleep because he wonders if his idea is feasible. Help him find that out!

Input

The first line of input contains one integer n ($2 \leq n \leq 1\,000\,000$). The second and the third line contain descriptions of permutations $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, each of which is a sequence of n distinct integers between 1 and n. You can assume that the permutations are different.

Output

The first and only line of output should contain one word `TAK` (Polish for *yes*) or `NIE` (Polish for *no*) depending on whether there exists such $k > 1$ that after repeating Bytie's shuffle k times, one obtains Byteasar's shuffle.

Examples

standard input	*standard output*
4 2 4 3 1 1 2 3 4	TAK
4 1 2 3 4 2 4 3 1	NIE

Problem C. Matrix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

You are given an $n \times m$ matrix filled with zeroes and ones. We consider all the $m!$ possible reorderings of columns of the matrix. In particular, we are interested in those reorderings for which the ones in every row are placed in adjacent columns (i.e., they form a contiguous part of the row).

Your task is to compute the number of such reorderings of columns, modulo a given number x.

Input

The first line of the input contains three positive integers n, m, and x ($1 \le n, m \le 2\,000$, $2 \le x \le 1\,000\,007$); The following n lines describe the subsequent rows of the matrix. Each of these lines contains exactly m space-separated digits 0 or 1.

Output

Your program should output a single integer: the number of interesting reorderings of columns of the matrix, modulo x.

Examples

standard input	*standard output*
3 4 42 1 1 0 1 1 0 1 1 1 1 0 0	2

Problem D. Parcels

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Byteasar owns a garden in which he grows apple-trees bearing golden apples. The work in the garden is quite hard and Byteasar is not as strong and healthy as he used to be in the old days. Therefore he decided to divide his garden into parcels that he would give to his sons. Byteasar would like each of his sons to have a wealthy life, so he would like each of the parcels to contain at least one of the priceless apple-trees.

Byteasar's garden has the shape of an $n \times n$ square. For simplicity we introduce a rectangular coordinate system in the garden, in which the lower left corner of the garden has coordinates $(0, 0)$ and the upper right corner has coordinates (n, n). Some unit squares contain apple-trees. Each of the parcels should be a rectangle with sides parallel to the axes of the coordinate system and integer coordinates of corners. The sides of the parcels may touch, but the parcels cannot overlap. The parcels should cover the whole garden. Each parcel may have an arbitrary size and must contain at least one apple-tree.

You may assume that the desired division of the garden is possible to accomplish.

Input

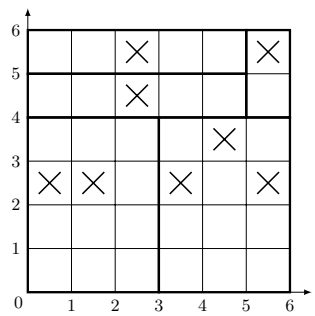
The first line of input contains two integers n and k ($1 \le n \le 1\,000$, $1 \le k \le n^2$) that represent the length of the side of the garden and the number of Byteasar's sons. The following n lines describe the unit squares of the garden. Each of those lines contains n characters `x` and/or `.` that represent a unit square with or without an apple-tree, respectively.

Output

Your program should output k lines that describe a correct division of the garden into parcels. Each of the lines should contain four integers x_1, y_1, x_2, y_2 that give the coordinates of the lower left (x_1, y_1) and the upper right (x_2, y_2) corner of the garden. The order of the parcels in the output does not matter.

Examples

standard input	*standard output*
6 5 ..x..x ..x... ....x. xx.x.x	0 0 3 4 0 4 5 5 5 4 6 6 3 0 6 4 0 5 5 6



Problem E. Alien Invasion

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

In the Byteland Department of Exobiology the Institute of Space Hazard has been established. All its employed researchers are to protect the citizens of Byteland from the disastrous effects of an alien invasion, which unfortunately undoubtedly is going to happen.

There are n cities in Byteland situated along the Byteroad. Cities are numbered 1 through n and the i-th city has a_i inhabitants.

It is a common knowledge that aliens always invade during the night and they can only invade a single city at once. Unfortunately all the inhabitants of an invaded city are instantaneously kidnapped and transported to a different galaxy.

Researchers of the institute try to find a way of securing the citizens. As aliens are not interested in rodents (such as rats), researchers decided to exploit this fact and use rats as a warning system. When aliens invade a city, two rats leave this city in opposite directions of the Byteroad, spreading the news of invasion. A rat needs almost one day to go between two neighboring cities, hence the news sent from an j-th city reaches a k-th city just before the dusk of the $|k-j|$-th day after the invasion. After hearing the news alarmed citizens hide themselves in shelters, staying out of reach of alien's tentacles. As Byteland's shelters are well-equipped, warned citizens stay in the shelters until the aliens stop invading Byteland and go back to their galaxy.

As you can see the described warning system may not save all the citizens. Researchers try to figure out the worst case scenario, that is, the maximum number of citizens that may be kidnapped.

Input

The first line of input contains a single integer n ($1 \le n \le 1\,000\,000$), the number of cities in Byteland. The second line of input contains a sequence of integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^9$), describing the number of inhabitants of subsequent cities along the Byteroad.

Output

The first and only line of output should contain a single integer: the number of kidnapped citizens in the worst case scenario.

Examples

standard input	*standard output*
6 5 9 1 3 7 2	16

Problem F. Spider 2

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

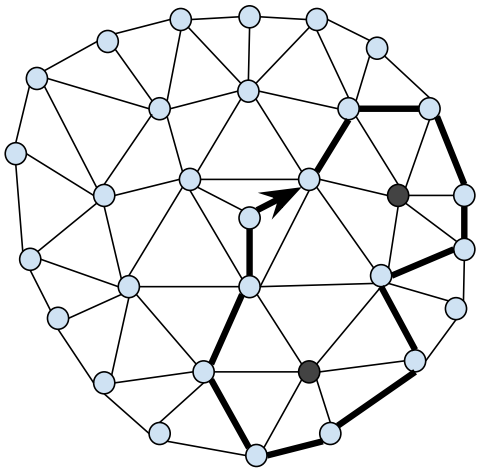
Spider Bytholomew has built himself a beautiful web. It consists of a central node and d rings, numbered from 1 to d going outwards. Every ring is a cycle consisting of nodes connected by strands of silk.

The central node is connected to all a_0 nodes of ring 1. For all i between 1 and $d-1$:

- Each node of ring i is connected to exactly a_i consecutive nodes of ring $i+1$.
- In addition, every two consecutive nodes of ring i share one common neighbor in ring $i+1$.

Therefore, ring $i+1$ has $a_i - 1$ times as many nodes as ring i.

Bytholomew takes a walk around the web. He starts from the central node and then moves along the silk, not visiting any node more than once and finishing where he started. The spider catches a fly in every node that is strictly contained inside the figure bounded by the walk.



Your task is to compute the total number of flies caught.

Input

The first line of input contains the the sequence $a_0, \ldots, a_{d-1}$ without any separating whitespace ($3 \le a_i \le 5, 1 \le d \le 10^6$). The second line of the input contains a similarly formatted sequence $t_1, \ldots, t_n (1 \le t_i \le 8, 1 \le n \le 10^6)$. It describes the subsequent turns taken by the spider during the walk. The spider exited the i-th node on the path using the t_i-th strand of silk in clockwise order, where strand 0 is the one the spider used to enter the node.

The value t_1 describes the turn taken in the first node encountered after leaving the central node, and t_n describes the turn the spider would need to take in the central node if he wanted to go along the path again.

Output

Your program should output a single integer: the number of the web's nodes strictly inside the figure bounded by the spider's walk.

Examples

standard input	*standard output*
343 342122212432	2
343 6561	0

The first example corresponds to the picture in the problem statement.

Problem G. Tax

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

The king of Byteland decided to follow a worldwide trend and introduce taxes everywhere he can. His new invention is the so-called *travel tax* that is paid by everyone travelling through the country.

Each Bytean road is assigned a tax rate. When passing through a town one needs to pay a tax that equals the *maximum* of the tax rate on the road that was used to enter the town and of the tax rate on the road that was used to exit the town. One also pays the tax in the first and the last town on the trip: there the tax amount equals the rate of the only corresponding road of the trip.

Your friend Byteasar is going for a trip from Bytetown to Bitcity. Help him plan his trip so that the amount of tax he pays is minimal.

Input

The first line of input contains two integers n and m ($2 \leq n \leq 100\,000, 1 \leq m \leq 200\,000$), the number of towns and the number of roads in Byteland. The towns are numbered 1 through n.

The following m lines contain descriptions of roads. The i-th of those lines contains three integers a_i, b_i, c_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq c_i \leq 1\,000\,000$). This means that the towns a_i and b_i are connected by a bidirectional road with the tax rate equal to c_i bythalers. Each pair of towns is connected by at most one road.

Output

The first and only line of output should contain one integer: the minimal tax amount (in bythalers) on a trip from Bytetown (i.e. the town number 1) to Bitcity (i.e. the town number n). You can assume that in each input data there is a sequence of roads connecting these two towns.

Examples

standard input	*standard output*
4 5 1 2 5 1 3 2 2 3 1 2 4 4 3 4 8	12

Explanation of the example: The optimal trip leads through the towns 1, 3, 2 and 4. The tax amount paid in the respective towns is: 2, $\max(2, 1) = 2$, $\max(1, 4) = 4$ and 4. This yields 12 bythalers of tax in total.

Problem H. Circles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	13 seconds
Memory limit:	512 mebibytes

The farmers of Byteland will remember the last summer for a long time. Usually a summer burns into one's memory due to plentiful harvest, severe drought or hail. However the last summer was so extraordinary because of strange shapes, which appeared at several wheat fields. As a Bytean expert in all unusual problems, Byteasar decided to explain this phenomenon on the ground of science. In order to do this, he thoroughly inspected each wheat field and noticed that each shape was made by crumbling all wheat belonging to a circular region. Every two circles touch in at most one point (in particular no circle can be contained in a different circle).

Byteasar suspects that the circles describe messages sent by aliens. Unfortunately, understanding their language is very hard. At this point Byteasar collected all the information about the shapes he found and he is going to use some tools of statistical analysis. The more interesting data he collects, the better. Byteasar asked you to write a program, which given the description of all the circles computes the number of pairs of circles having a common point.

Input

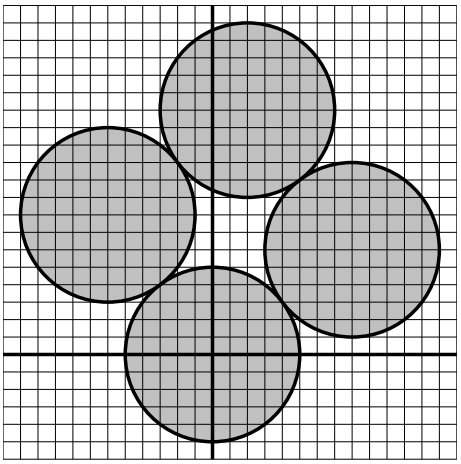
The first line of input contains a single integer n ($1 \le n \le 500\,000$), the number of circles. Each of the following n lines describes a circle. In the i-th of those lines there are three integers x_i, y_i, r_i ($-10^9 \le x_i, y_i \le 10^9$, $1 \le r_i \le 10^9$), meaning that the center of the i-th circle has coordinates (x_i, y_i) while its radius equals r_i.

Output

Your program should output the number of pairs of circles having a common point.

Examples

standard input	*standard output*
4 0 0 5 8 6 5 -6 8 5 2 14 5	4



Problem I. Knapsack

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Johnny and Margaret are leaving for a trip. The boy wanted to impress his friend and decided to pack all their belongings to a single knapsack. Moreover, the heavier the knapsack the better, as it will impress Margaret even more. Obviously Johnny cannot pack all the things he has at home, as it could cause the knapsack to tear apart, and such a disaster would make Johnny so ashamed that he could never look into Margaret's eyes again.

Furthermore some objects are useless without other objects. Johnny has to make sure that, for each object, all objects that it depends on are also taken. For example, if Johnny takes a radio, he has to take batteries as well, while taking a tripod enforces taking a camera. For each object i Johnny determined a different object j, such that the object i is useless without the object j, or he figured out that the object i is self-contained and does not require any other object to be taken.

Your task is to help Johnny and compute the maximum total weight of all the objects he can take without exceeding the capacity of the knapsack and at the same time without taking any useless object. In addition, you should find any choice of objects that fulfills these criteria.

Input

The first line of input contains two integers n and p ($1 \le n \le 200$, $1 \le p \le 10^6$), describing the number of objects that are under Johnny's consideration and the capacity of the knapsack (in kilograms). If one packs objects of total weight exceeding p, then the knapsack is going to tear apart.

The objects are numbered 1 through n. Each of the following n lines contains a description of an object — the i-th of those lines contains two integers j_i and m_i ($0 \le j_i < i$, $1 \le m_i \le p$), that represent the number of the object the object i depends on (if $j_i = 0$, then the object i is self-contained) and the weight of the object i in kilograms.

Output

Your program should output two lines. The first line should contain a single integer: the maximum total weight (in kilograms) of objects that Johnny can take. The second line should describe any valid selection of objects of this total weight. The description should consist of the count of the objects followed by a strictly increasing sequence of their indices.

Examples

standard input	*standard output*
7 11 0 3 0 1 2 3 2 2 4 4 5 3 5 2	10 4 2 4 5 6

Problem J. Tapestries

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

An exhibition of tapestries is opening in Byteotian Museum of Fine Arts. The main exhibition room, viewed from top, is a polygon (not necessarily convex). A tapestry is hanged on each wall of the room, each tapestry taking all the area of its wall.

A lamp has been installed in the room in order to illuminate the exhibition. The lamp is glowing uniformly in all directions. However, while some of the tapestries have to be flooded with light, others cannot be exposed to strong light.

Byteasar, the curator, has been moving the lamp around the room, but so far he is not satisfied with the results. Now he is terrified by the prospect of moving the tapestries around instead — this would require much effort, and the exhibition is to open soon. Perhaps you will be able to tell him if his attempts are doomed or not?

Your task is to determine if there is such a spot that placing the lamp in it satisfies the following:

- each wall is to be either completely illuminated or completely shaded, as required by the tapestry hanging on it; there can be no wall that is partly illuminated and partly shaded;
- if the lamp is located exactly on the wall or its extension, this wall is not illuminated;
- the lamp can neither be switched off nor taken away from the room; it has to be on while located (strictly) inside the room (i.e., it cannot be placed in a corner or on any wall).

Input

There is a single integer t ($1 \leq t \leq 20$) in the first line of the standard input, denoting the number of data sets. The following lines describe these data sets.

The first line of a single description holds a single integer n ($3 \leq n \leq 1\,000$), denoting the number of walls in the main exhibition room. Then the following n lines specify the room's shape. Each of those lines contains a pair of integers x_i and y_i ($-30\,000 \leq x_i, y_i \leq 30\,000$ for $i = 1, 2, \ldots, n$), separated by a single space, denoting the coordinates of the room's corner or, in other words, the vertex of corresponding polygon. The vertices are given clockwise.

The next n lines specify the tapestries' requirements. Each of those lines contains a single letter, `S` or `C`, denoting that the wall has to be illuminated or shaded, respectively. The letter in the i-th of these lines (for $1 \leq i \leq n-1$) regards the wall between the i-th and the $(i+1)$-th vertex. The letter in the last of these lines regards the wall between the last and the first vertex.

The polygon depicting the room's shape has no self-crossings, i.e., with the exception of successive sides, which share a common vertex, no two sides of the polygon share a common point. Furthermore, no three vertices of the polygon are collinear.

Output

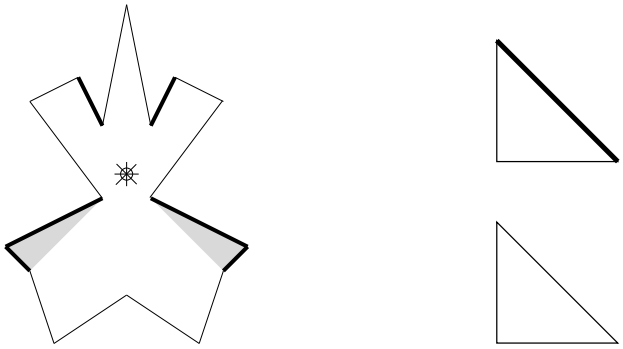
For each data set your program should print to the standard output a single line containing a single word: `TAK` (Polish for *yes*) if the lamp can be placed so as to satisfy all aforementioned requirements, or `NIE` (Polish for *no*) otherwise.

Examples

standard input	*standard output*
1 16 5 -3 4 -4 3 -7 0 -5 -3 -7 -4 -4 -5 -3 -1 -1 -4 3 -2 4 -1 2 0 7 1 2 2 4 4 3 1 -1 C S S S S C C S S C S S C S S C	TAK
2 3 0 0 0 1 1 0 S C S 3 0 0 0 1 1 0 S S S	NIE TAK

In the figures below, the thick sides denote the walls that have to be shaded while the remaining sides — the walls that have to be illuminated. The figure for the first example shows a correct placement of

the lamp.



Problem K. Two Cakes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Two urgent cake orders came to Byteasar's confectionery. As we all know, cakes have layers. The confectionery offers n different kinds of layers and each cake contains exactly one layer of each kind. A cake order specifies a sequence in which the layers are to be placed.

Byteasar hires n confectioners; the i-th confectioner (for $1 \leq i \leq n$) can prepare only a layer of the i-th kind. Each confectioner places his layer in a single minute (during that time he or she can work on a single cake only). Layers of a cake are to be placed one by one, however two cakes can be processed in parallel. How much time will it take to fulfill the two given cake orders, assuming that the cakes are produced in an optimal manner?

Input

The first line of input contains a single integer n ($1 \leq n \leq 1\,000\,000$). Two lines follow containing a description of the first and the second cake order respectively. Each cake order is a sequence of n pairwise distinct integers of value between 1 and n, describing the subsequent layers of the ordered cake starting from the topmost layer.

Output

The first and only line of output should contain a single integer: the number of minutes needed to produce the two ordered cakes.

Examples

standard input	*standard output*
3 1 2 3 3 2 1	4