Problem A. Arithmetic Rectangle (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	7 seconds
Memory limit:	256 mebibytes

We are given a grid consisting of $n \times m$ unit squares. Each of the unit squares contains a single integer. In this task we are interested in *arithmetic rectangles* lying on the grid, i.e., rectangles composed of unit squares such that numbers in every row and every column form arithmetic sequences. Recall that an arithmetic sequence is a sequence in which any two consecutive terms differ by the same amount.

In addition, we aim to find the largest arithmetic rectangle, i.e., the one covering the most unit squares. For example, the largest arithmetic rectangle on the grid below consists of nine unit squares:

5	3	5	7
2	4	4	4
3	5	3	1
6	3	2	4

Input

In the first line of the input there is a single integer t ($1 \leq t \leq 10\,000$) denoting the number of test cases that follow. The description of each test case begins with a line with two integers n and m ($1 \leq n, m \leq 3\,000$). In each of the following n lines there are m integers from the range $[0, 10^9]$. These numbers describe the grid. The size of each input file will not exceed 20 MB.

Output

Your program should output t lines with answers for the consecutive test cases. The answer for a single test case is one integer equal to the number of unit squares contained in the largest arithmetic rectangle that can be found on the grid described in the test case.

Examples

Standard input	Standard output
2 4 4 5 3 5 7 2 4 4 4 3 5 3 1 6 3 2 4 2 3 0 1 2 1 2 3	9 6

Problem B. Bytean Road Race (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	6 seconds
Memory limit:	256 mebibytes

The Bytean Road Race is to be held tomorrow in Bytetown city center. The streets of Bytetown form a regular grid: all of them go from the south to the north or from the west to the east. The participants of the race are only allowed to use some given parts of the roads.

Byteasar's task is to place the event's sponsors' banners on some of the crossings, and to do that he has to examine the route map of the race. The map depicts the segments of streets that the runners are allowed to use. There are n crossings and m vertical and horizontal road segments marked on it. Every segment starts and ends at some crossing and does not contain any other crossings. The road segments may only intersect at the crossings.

The crossings are numbered from 1 to n. The race will start at the crossing number 1 and finish at the crossing number n. The runners can pick their routes themselves, however they are required to run only south and east and only along the segments marked on the map. The road segments on the map are chosen in such a way that, by running in accordance with the rules, one can get to the finish from any place, and every place is reachable from the start crossing.

Byteasar wants to place the banners in a way that ensures that no runner will see a banner of the same sponsor twice. Therefore, for some pairs of crossings, Byteasar has to check whether it is possible for the route of some participant to go through both crossings. The race takes place tomorrow, so a program that will help him do the task is urgently needed.

Input

The first line of the input contains three integers n, m and k ($2 \le n \le 100\,000$, $1 \le m \le 200\,000$, $1 \le k \le 300\,000$). They denote the number of crossings on the race route, the number of marked segments on the map and the number of pairs of crossings to be checked, respectively.

The following n lines describe the locations of the crossings. The i-th of these lines contains the coordinates of the i-th crossing represented by two integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$). In addition, $x_1 \le x_n$ and $y_1 \ge y_n$. There will be at most one crossing at any given point. The axes of the coordinate system correspond to the real world cardinal directions in the natural way: the OX axis goes towards the east, and the OY axis — towards the north.

Each of the following m lines contains a description of a single segment on the map, consisting of a pair of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) denoting the indices of crossings connected by the segment. All of these segments are either vertical or horizontal and they may only intersect at common endpoints (the crossings).

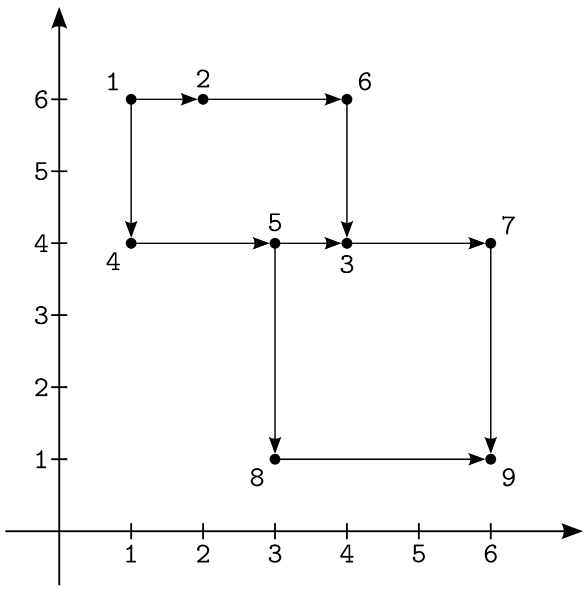
The next k lines contain the descriptions of pairs of crossings to be checked. The i-th of these lines contains two integers p_i, q_i ($1 \le p_i, q_i \le n$, $p_i \ne q_i$).

Output

Your program should output k lines. The i-th of these lines should contain the word `TAK` (i.e., *yes* in Polish) if it is possible for the route of some participant to go through both crossings p_i and q_i (in any order). Otherwise the output should be `NIE` (*no* in Polish).

Examples

Standard input	Standard output
9 10 4 1 6 2 6 4 4 1 4 3 4 4 6 6 4 3 1 6 1 1 2 4 1 2 6 3 6 5 4 5 3 5 8 3 7 7 9 9 8 4 8 2 5 8 7 7 6	TAK NIE NIE TAK



Problem C. Will It Stop?

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Byteasar was wandering around the library of the University of Warsaw and at one of its facades he noticed a piece of a program with an inscription “Will it stop?”. The question seemed interesting, so Byteasar tried to tackle it after returning home. Unfortunately, when he was writing down the piece of code he made a mistake and noted:

while $n > 1$ **do**
  if $n \bmod 2 = 0$ **then**
    $n := n/2$
  else
    $n := 3 \cdot n + 3$

Byteasar is now trying to figure out, for which initial values of the variable n the program he wrote down stops. We assume that the variable n has an unbounded size, i.e., it may attain arbitrarily large values.

Input

The first and only line of input contains one integer n ($2 \le n \le 10^{14}$).

Output

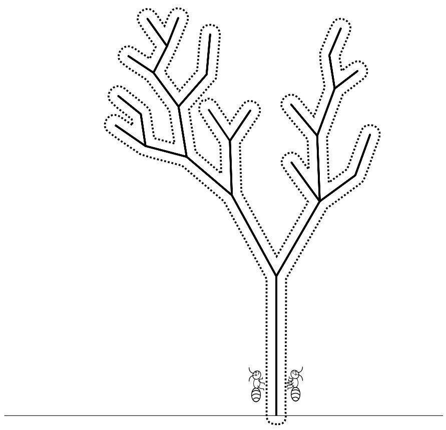
In the first and only line of output you should write a single word `TAK` (i.e., *yes* in Polish), if the program stops for the given value of n, or `NIE` (*no* in Polish) otherwise.

Examples

Standard input	Standard output
4	TAK

Problem D. Ants (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	30 seconds
Memory limit:	24 mebibytes



Computer geeks like trees. Ants also like trees. Therefore we are given a tree with two ants walking on it — the Left Ant and the Right Ant — in a way shown in the above figure (the ants walk along the path depicted with a dotted line). They start their journey at the lower end of the trunk, on its opposite sides. The Left Ant needs 2 seconds to walk along a single edge of the tree if walking from the root (upwards), and 1 second if walking towards the root (downwards). The Right Ant is two times faster. When the two ants meet, they both turn around and start walking in the opposite directions. If any of the ants steps from the tree to the ground, it immediately starts to climb on the opposite side of the trunk. Apart from that, the ants are so tiny that they would not be visible even under a microscope (they are depicted larger in the figure on purpose). Your task is to write a program that computes the moment at which the ants turn around for the second time.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 1000$) representing the number of test cases described in the input.

The description of each test case consists of two lines. The first line contains an even integer n ($2 \leq n \leq 100\,000\,000$) denoting the number of edges in the tree. The second line holds a description of the tree. It is a string consisting of $\frac{n}{2}$ characters representing a $2n$-bit binary number written in a hexadecimal form (using digits and small letters from `a` to `f`). This number shows the Left Ant's path around the whole tree assuming that the Right Ant stands still. The consecutive bits of this number (starting from the left) mean if the Left Ant walks away from the root of the tree along the corresponding edge (bit 1) or if it walks towards the root along this edge (bit 0). The root has a trunk, that is, there is exactly one edge leading from the root of the tree.

The size of the input file does not exceed 50 MB, which is much more than the amount of memory available for your program.

Output

Your program should output t lines containing answers to the consecutive test cases. Each answer should represent the moment (in seconds) in which the ants turn around for the second time, given as an irreducible fraction `p/q` (without any white space around /), where p and q are positive integers. If the answer is integer then, obviously, $q = 1$.

Examples

Standard input	Standard output
1 28 fb1da30d1b7230	282/5

The sample data corresponds to the figure above, and transforms to the following sequence of bits:

```
1111 1011 0001 1101 1010 0011 0000 1101 0001 1011 0111 0010 0011 0000
```

Problem E. Gophers

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	13 seconds
Memory limit:	256 mebibytes

Dick Dastardly wants to bedevil poor Bytean gophers. These nice little creatures live in holes in the upper parts of High Bytemountains.

Dick has found a mountain ridge with n gopher holes located along a straight line (for simplicity, we index the holes from 1 to n, from west to east). Dick plans to torture gophers using rock & roll music. He has bought m CD players, put a different Bytels' album in each of them and arranged the CD players along the ridge. The music from a CD player disturbs gophers located in holes distant by at most l meters from it.

Feeling troubled, the gophers asked you to check in which holes they will not be able to sleep well during this winter. But now Dick Dastardly wants to make even more mess…

He will move the CD players from time to time. The gophers were able to steal Dick's secret plan and now they know precisely that on the morning of the i-th day Dick will take the CD player located p_i meters from the hole number 1 and will put it at a point located r_i meters from that hole. Help the gophers and count the number of holes in which they will not be able to fall asleep after each such operation.

Input

The first line of input contains four integers n, m, d and l ($2 \le n, m \le 500\,000$, $1 \le d \le 500\,000$, $1 \le l \le 10^9$) representing the number of gophers' holes, the number of Dick's CD players, the number of days and the range of a CD player, respectively.

The second line of input contains $n-1$ integers $x_2, x_3, \ldots, x_n$ ($0 < x_2 < x_3 < \ldots < x_n \le 10^9$) denoting the distances of the holes $2, 3, \ldots, n$ from the hole number 1.

The third line contains m integers $z_1, z_2, \ldots, z_m$ ($0 \le z_1 < z_2 < \ldots < z_m \le 10^9$) denoting the distances of the consecutive CD players from the hole number 1. All the CD players are located to the east of this hole.

Next, d lines follow. The i-th of these lines contains two integers p_i and r_i ($0 \le p_i, r_i \le 10^9$, $p_i \ne r_i$) meaning that in the beginning of the i-th day Dick is going to move the CD player located p_i meters from the hole number 1 to the point located r_i meters to the east from that hole. You may assume that before every such operation there is a CD player at the position p_i and there are no CD players at the position r_i.

Output

Your program should output $d+1$ lines. The line number i (for $i = 1, 2, \ldots, d$) should contain one integer representing the number of holes in which no gopher would be able to sleep well during the night *before* the i-th Dick's operation. The last line should contain this number after the last Dick's operation.

Examples

Standard input	Standard output
5 3 4 1 2 5 6 11 2 4 8 2 1 4 10 8 6 1 8	2 3 3 5 3

Problem F. Laundry

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	8 seconds
Memory limit:	256 mebibytes

A few friends have decided to do laundry together. They are all very neat, so each day each of the friends wears one clean pair of socks and one clean shirt. The friends have put all dirty socks and shirts to their washing machine. Now they have started to plan how they will dry their laundry. To put this in order, they have decided that:

- every sock will be fastened to the string with a single clothespin,
- each shirt will be fastened with three clothespins,
- one person's socks should all be fastened with clothespins of the same color,
- one person's shirts should all be fastened with clothespins of the same color,
- clothes belonging to different persons may not be fastened with clothespins of the same color,
- apart from that, they wish to use the smallest possible number of clothespin's colors.

Now they have scattered all their clothespins on the floor and counted the number of clothespins of each color. Unfortunately, they were not able to figure out which colors should each of them use. Write a program that will help them with this problem.

Input

The first line of input contains two integers n and k ($2 \le n, k \le 1\,000\,000$) denoting the number of friends and the number of clothespins' colors available. The second line contains n numbers $d_1, d_2, \ldots, d_n$ representing the number of days each friend was collecting laundry ($1 \le d_i \le 1\,000\,000$). The third line contains k numbers $l_1, l_2, \ldots, l_k$ representing the numbers of clothespins of respective colors ($1 \le l_i \le 4\,000\,000$).

Output

Your program should output the minimal number of clothespins' colors, needed to dry all the laundry.

If it is not possible to dry all the laundry in the requested manner, your program should output a single word `NIE` (i.e., *no* in Polish).

Examples

Standard input	Standard output

`2 4`
`3 4`
`20 10 8 10`
	`3`

`3 8`
`5 4 3`
`14 14 14 14 14 14 14 14`
	`NIE`

Explanation for the first example: The first person needs 6 clothespins for her socks and 9 clothespins for her shirts. The second person needs 8 clothespins for her socks and 12 clothespins for her shirts. The second person should use the clothespins of the first color both for her socks and her shirts. The first person may then use, e.g., the clothespins of the second and the fourth color.

Problem G. Bits Generator (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	5 seconds
Memory limit:	256 mebibytes

Byteasar likes to play with his random (well, actually pseudorandom) bits generator, which he has found on his computer. This generator works in a very simple way. The moment the computer is turned on, an integer in the range between 0 and $m-1$ is chosen automagically. This integer is called the *seed* of the generator; we will use variable z to represent it. Then, in order to generate a random bit, the following function is called. It computes a new value of the seed which is then used to generate a single bit:

$z := \lfloor (z \cdot a + c)/k \rfloor \bmod m$
if $z < \lfloor m/2 \rfloor$ **then**
  return 0
else
  return 1

The numbers a, c, k are some constants. Byteasar has called this function n times and has thus obtained a sequence of bits $b_1, b_2, \ldots, b_n$. Now he is wondering what is the number of different possible values of the initial seed.

Input

The first line of the input contains five integers a, c, k, m and n ($0 \le a, c < m$, $1 \le k < m$, $2 \le m \le 1\,000\,000$, $1 \le n \le 100\,000$). The second line contains an n-character string consisting of digits `0` and `1`; the i-th digit of the string represents the bit b_i.

Output

You should output one integer representing the number of integers from the range between 0 and $m-1$ which could have been the initial seed of the generator.

Examples

Standard input	Standard output
3 6 2 9 2 10	4

Explanation of the example: The initial seed of the generator could have been equal to 1, 2, 7 or 8.

Problem H. Afternoon Tea

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

During his visit at Bytic Islands Byteasar really enjoyed the national beverage of Byteans, that is, tea with milk. This drink is always prepared in a strictly determined manner, which is as follows. Firstly the teacup is filled with tea mixed half and half with milk. Then, an n-letter *ceremonial word* consisting of letters `H` and `M` is chosen. Now, for $i = 1, 2, \ldots, n$, the following action is performed: if the i-th letter of the ceremonial word is `H`, one should drink half of the teacup, add tea until the teacup is full and stir. On the other hand, if the i-th letter of the word is `M`, one should perform a similar action, however milk should be added instead of tea. After such action is performed for each letter of the ceremonial word, the remaining liquid is disposed of.

Each time Byteasar performs the ceremony, he wonders which of the ingredients he has drunk more: tea or milk. Help Byteasar answer this question.

Input

The first line of input holds an integer n ($1 \le n \le 100\,000$). The second line contains an n-letter word consisting of letters `H` and `M`; this is the ceremonial word used by Byteasar.

Output

Your program should output a single letter `H` if Byteasar has drunk more tea than milk; a single letter `M` if he has drunk more milk than tea; or the word `HM` if he has drunk equal amounts of tea and milk.

Examples

Standard input	Standard output
5 HMHHM	H

Explanation of the example: Byteasar has drunk $1\frac{37}{64}$ teacups of tea and $\frac{59}{64}$ teacups of milk in total.

Problem I. Intelligence Quotient (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	10 seconds
Memory limit:	256 mebibytes

At the University of Byteland one can only study maths and computer science. Currently there are n maths students and m computer science students. These majors are so hard to study that nobody studies both of them at the same time.

Byteasar is the rector of the university. He would like to form a team of students which will solve all the hardest problems of mankind. Since he knows the IQ of each student, he has decided to form a team with the largest possible sum of IQs of its members.

However, IQ is not everything. That is why the rector would like all members of the team to know each other. It is known that all maths students know each other. And similarly, each computer science student knows every other student majoring in computer science.

Help the rector by writing a program that will help him form a team of students with the largest possible sum of IQs in which all the members know each other.

Input

The first line of the input contains three integers n, m and k ($1 \le n, m \le 400$, $0 \le k \le n \cdot m$) which denote the number of maths students, the number of computer science students and the number of pairs of students from different majors that know each other, respectively.

Each of the following k lines describes one pair of acquaintances: the i-th of these lines contains two integers a_i and b_i ($1 \le a_i \le n$, $1 \le b_i \le m$) denoting an index of a maths student and an index of an computer science student from the i-th pair. The maths students are indexed with integers starting from 1 and so are the computer science students.

The following line contains n integers in the range $[1, 10^9]$, which represent the IQs of the subsequent maths students. The next line contains m integers representing the IQs of the computer science students, in a similar format.

Output

The first line of the output should contain one integer equal to the maximum sum of IQs in a team satisfying Byteasar's requirements.

The second line should contain one integer — the number of maths students that Byteasar should choose. The third line should contain the indices of these students, listed in any order. If there are no maths students in the team, an empty line should be printed.

The following two lines should describe the indices of computer science students assigned to the team, in a similar format.

If there are multiple optimal solutions, your program should output any one of them.

Examples

Standard input	Standard output
3 2 3 1 1 2 1 2 2 1 3 1 1 2	6 1 2 2 1 2

Problem J. Cave (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	6 seconds
Memory limit:	256 mebibytes

Byteasar has discovered a cave. It appears that the cave contains n chambers connected with passages in such a way that there exists a single way of getting from any chamber to any other chamber.

The cave should now be examined more thoroughly, so Byteasar has asked his friends for help. They have all arrived at the cave and they are willing to divide themselves into groups. Each group should examine the same number of chambers, and each chamber should be examined by exactly one group. Additionally, for the groups not to interfere with each others' work, the members of each group should be able to move between the assigned chambers without passing through chambers assigned to other groups.

How many groups can the explorers be divided into?

Input

The first line of the input contains one integer n ($2 \le n \le 3\,000\,000$) denoting the number of chambers in the cave. The chambers are numbered 1 through n.

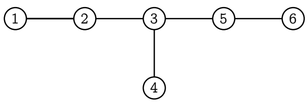
The following $n-1$ lines describe connections between the chambers. The i-th of these lines contains an integer a_i ($1 \le a_i \le i$) which represents a passage connecting chambers number $i+1$ and a_i.

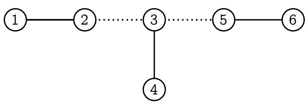
Output

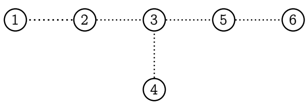
Your program should output a single line containing all integers k, such that the chambers can be divided into k disjoint sets of equal size, and one can move between any two chambers belonging to the same set passing only through chambers from this set. The numbers should be written in an ascending order and separated with single spaces.

Examples

Standard input	Standard output
6 1 2 3 3 5	1 3 6







Problem K. Cross Spider

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

The Bytean cross spider (*Araneida baitoida*) is known to have an amazing ability. Namely, it can instantly build an arbitrarily large spiderweb as long as it is contained in a single plane. This ability gives the spider an opportunity to use a fancy hunting strategy. It does not need to wait until a fly is caught in an already built spiderweb; if only the spider knows the current position of a fly, it can instantly build a spiderweb to catch the fly.

A cross spider has just spotted n flies in Byteasar's garden. Each fly is flying still in some point of a 3D space. The spider is wondering if it can catch all the flies with a single spiderweb. Write a program that answers the spider's question.

Input

The first line of the input contains an integer n ($1 \le n \le 100\,000$). The following n lines contain a description of the flies in a 3D space: the i-th line contains three integers x_i, y_i, z_i ($-1\,000\,000 \le x_i, y_i, z_i \le 1\,000\,000$) giving the coordinates of the i-th fly (a point in a 3-dimensional Euclidean space). No two flies are located in the same point.

Output

Your program should output a single word `TAK` (i.e., *yes* in Polish) if the spider can catch all the flies with a single spiderweb. Otherwise your program should output the word `NIE` (*no* in Polish).

Examples

Standard input	Standard output
4 0 0 0 -1 0 -100 100 0 231 5 0 15	TAK
4 0 1 0 -1 0 -100 100 0 231 5 0 15	NIE

Problem L. Choose The Best Computer (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Famous hardware tester Tom performs some analysis and determines, that the most preferred personal computer is one that has the largest value of the formula

$$2 \cdot R + 3 \cdot S + D,$$

where

- RAM (in gigabytes), denoted as R;
- CPU speed (in megahertz), denoted as S;
- disk drive space (in gigabytes), denoted as D.

Your task is to read a given list of computers and output the top two computers in order of preference, from highest preference to lowest preference.

Input

The first line of input will be an integer n ($1 \le n \le 10^4$). Each of the remaining n lines of input will contain a computer specification. A computer specification is of the form: computer name (a string of less than 20 uppercase English letters), the RAM available (an integer R with $1 \le R \le 128$), the CPU speed (an integer S with $1 \le S \le 4000$), the disk drive space (an integer D with $1 \le D \le 3000$).

There is one space between the name, RAM, CPU speed and disk drive space on each line.

Output

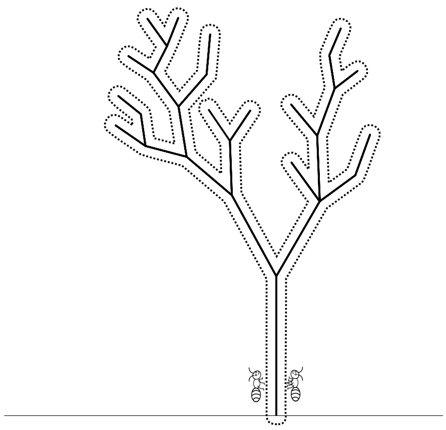
The output is the name of the top two preferred computers, one name per line, sorted in decreasing order of preference. If there is a tie in the rankings, pick the computer(s) whose name(s) are lexicographically smallest (i.e., "(IBM)" is smaller than "`LENOVO`"). If there is only one computer, output that computer on one line (i.e., do not print it twice).

Example

Standard input	Standard output
4 ABC 13 22 1 DEF 10 20 30 GHI 11 2 2 JKL 20 20 20	JKL DEF

Problem M. Ants (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	10 seconds
Memory limit:	256 mebibytes



Computer geeks like trees. Ants also like trees. Therefore we are given a tree with two ants walking on it — the Left Ant and the Right Ant — in a way shown in the above figure (the ants walk along the path depicted with a dotted line). They start their journey at the lower end of the trunk, on its opposite sides. The Left Ant needs 2 seconds to walk along a single edge of the tree if walking from the root (upwards), and 1 second if walking towards the root (downwards). The Right Ant is two times faster. When the two ants meet, they both turn around and start walking in the opposite directions. If any of the ants steps from the tree to the ground, it immediately starts to climb on the opposite side of the trunk. Apart from that, the ants are so tiny that they would not be visible even under a microscope (they are depicted larger in the figure on purpose). Your task is to write a program that computes the moment at which the ants turn around for the second time.

Input

The first line of the input contains a single integer t ($1 \le t \le 1000$) representing the number of test cases described in the input.

The description of each test case consists of two lines. The first line contains an even integer n ($2 \le n \le 100\,000\,000$) denoting the number of edges in the tree. The second line holds a description of the tree. It is a string consisting of $\frac{n}{2}$ characters representing a $2n$-bit binary number written in a hexadecimal form (using digits and small letters from `a` to `f`). This number shows the Left Ant's path around the whole tree assuming that the Right Ant stands still. The consecutive bits of this number (starting from the left) mean if the Left Ant walks away from the root of the tree along the corresponding edge (bit 1) or if it walks towards the root along this edge (bit 0). The root has a trunk, that is, there is exactly one edge leading from the root of the tree.

Output

Your program should output t lines containing answers to the consecutive test cases. Each answer should represent the moment (in seconds) in which the ants turn around for the second time, given as an irreducible fraction `p/q` (without any white space around `/`), where p and q are positive integers. If the answer is integer then, obviously, $q = 1$.

Examples

Standard input	Standard output
1 28 fb1da30d1b7230	282/5

The sample data corresponds to the figure above, and transforms to the following sequence of bits:

```
1111 1011 0001 1101 1010 0011 0000 1101 0001 1011 0111 0010 0011 0000
```

Problem N. Huffman encoding (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

The basic idea of Huffman encoding is that each character is associated with a binary sequence (i.e., a sequence of 0s and 1s). These binary sequences satisfy the prefix-free property: a binary sequence for one character is never a prefix of another characterвЂ™s binary sequence.

It is easy to see that to construct a prefix-free binary sequence, simply put the characters as the leaves of a binary tree, and label the left edge as 0 and the right edge as 1. The path from the root to a leaf node forms the code for the character at that leaf node.

For example, the binary tree constructs a prefix-free binary sequence for the characters A, B, C, D, E by next way: A is encoded as 00, B is encoded as 01, C is encoded as 10, D is encoded as 110 and E is encoded as 111.

The benefit of a set of codes having the prefix-free property is that any sequence of these codes can be uniquely decoded into the original characters.

Your task is to read a Huffman code (i.e., a set of characters and associated binary sequences) along with a binary sequence, and decode the binary sequence to its character representation.

Input

The first line of input will be an integer k ($1 \le k \le 20$), representing the number of characters and associated codes. The next k lines each contain a single character, followed by a space, followed by the binary sequence (of length at most 10) representing the associated code of that character.

You may assume that the character is an alphabet character (i.e. uppercase or lowercase English letter). You may assume that the sequence of binary codes has the prefix-free property. On the $k+2$-nd line is the binary sequence which is to be decoded. You may assume the binary sequence contains codes associated with the given characters, and that the $k+2$nd line contains no more than 250 binary digits.

Output

On one line, output the characters that correspond to the given binary sequence.

Example

Standard input	Standard output
5 A 00 B 01 C 10 D 110 E 111 00000101111	AABBE

Problem O. Cave (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	20 seconds
Memory limit:	256 mebibytes

Byteasar has discovered a cave. It appears that the cave contains n chambers connected with passages in such a way that there exists a single way of getting from any chamber to any other chamber.

The cave should now be examined more thoroughly, so Byteasar has asked his friends for help. They have all arrived at the cave and they are willing to divide themselves into groups. Each group should examine the same number of chambers, and each chamber should be examined by exactly one group. Additionally, for the groups not to interfere with each others' work, the members of each group should be able to move between the assigned chambers without passing through chambers assigned to other groups.

How many groups can the explorers be divided into?

Input

The first line of the input contains one integer n ($2 \leq n \leq 3\,000\,000$) denoting the number of chambers in the cave. The chambers are numbered 1 through n.

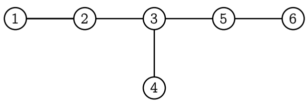
The following $n-1$ lines describe connections between the chambers. The i-th of these lines contains an integer a_i ($1 \leq a_i \leq i$) which represents a passage connecting chambers number $i+1$ and a_i.

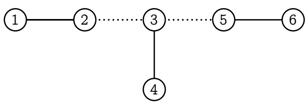
Output

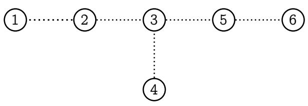
Your program should output a single line containing all integers k, such that the chambers can be divided into k disjoint sets of equal size, and one can move between any two chambers belonging to the same set passing only through chambers from this set. The numbers should be written in an ascending order and separated with single spaces.

Examples

Standard input	Standard output
6 1 2 3 3 5	1 3 6







Problem P. Sequences (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	7 seconds
Memory limit:	256 Mebibytes

Given a sequence of positive integers of length n, we define a *prime* subsequence as a consecutive subsequence of length at least two that sums to a prime number greater than or equal to two. For example, given the sequence:

$$35638$$

There are two prime subsequences of length 2 ($5 + 6 = 11$ and $3 + 8 = 11$), one prime subsequence of length 3 ($6 + 3 + 8 = 17$), and one prime subsequence of length 4 ($3 + 5 + 6 + 3 = 17$).

For a given sequence, find out shortest prime subsequence.

Input

The first line consists of an integer t ($1 \le t \le 20$) — the number of test cases. Each test case consists of one line. The line begins with the integer n, $0 < n \le 10^4$, followed by n non-negative integers less than 10^4 comprising the sequence.

Output

For each sequence, print the "`Shortest prime subsequence length is x:`", where x is the length of the shortest prime subsequence, followed by the shortest prime subsequence, separated by spaces. If there are multiple such sequences, print the one that occurs first. If there are no such sequences, print "`No prime subsequences.`".

Example

Standard input

```
3
5 3 5 6 3 8
5 6 4 5 4 12
21 15 17 16 32 28 22 26 30 34 29 31 20 24 18 33 35 25 27 23 19 21
```

Standard output

```
Shortest prime subsequence length is 2: 5 6
Shortest prime subsequence length is 3: 4 5 4
No prime subsequences.
```

Problem Q. Switch (Division 2 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are walking by a row of K ($4 \le K \le 25$) lights, some of which are on and some of which are off. In this initial configuration, there is no consecutive sequence of four lights that are on.

Whenever four or more consecutive lights are on, the lights in that consecutive block will turn off.

You can only turn on lights that are off.

What is the minimum number of lights you need to turn on in order to end up with all K lights off?

Input

The first line of input will consist of the integer K, indicating the number of lights. Each of the next K lines will have either the integer 0 (to represent a light that is off) or the integer 1 (to represent a light that is on).

Output

Your program should output the minimum number of lights that must be turned on in order to have all K lights be off.

Example

Standard input	Standard output
5 1 1 0 1 1	1