Problem A: ASCII Addition

Time limit: 1 s
Memory limit: 512 MiB

Nowadays, there are smartphone applications that instantly translate text and even solve math problems if you just point your phone's camera at them. Your job is to implement a much simpler functionality reminiscent of the past – add two integers written down as ASCII art.

An *ASCII art* is a matrix of characters, exactly 7 rows high, with each individual character either a dot or the lowercase letter x.

An expression of the form $a + b$ is given, where both a and b are positive integers. The expression is converted into ASCII art by writing all the expression characters (the digits of a and b as well as the + sign) as 7×5 matrices, and concatenating the matrices together with a single column of dot characters between consecutive individual matrices. The exact matrices corresponding to the digits and the + sign are as folows:

```
xxxxx  ....x  xxxxx  xxxxx  x...x  xxxxx  xxxxx  xxxxx  xxxxx  xxxxx  .....
x...x  ....x  ....x  ....x  x...x  x....  x....  ....x  x...x  x...x  ..x..
x...x  ....x  ....x  ....x  x...x  x....  x....  ....x  x...x  x...x  ..x..
x...x  ....x  xxxxx  xxxxx  xxxxx  xxxxx  xxxxx  ....x  xxxxx  xxxxx  xxxxx
x...x  ....x  x....  ....x  ....x  ....x  x...x  ....x  x...x  ....x  ..x..
x...x  ....x  x....  ....x  ....x  ....x  x...x  ....x  x...x  ....x  ..x..
xxxxx  ....x  xxxxx  xxxxx  ....x  xxxxx  xxxxx  ....x  xxxxx  xxxxx  .....
```

Given an ASCII art for an expression of the form $a + b$, find the result of the addition and write it out in the ASCII art form.

Input

Input consists of exactly 7 lines and contains the ASCII art for an expression of the form $a + b$, where both a and b are positive integers consisting of at most 9 decimal digits and written without leading zeros.

Output

Output 7 lines containing ASCII art corresponding to the result of the addition, without leading zeros.

Example

input

```
....x.xxxxx.xxxxx.x...x.xxxxx.xxxxx.xxxxx.......xxxxx.xxxxx.xxxxx
....x.....x.....x.x...x.x.....x.........x...x...x...x.x...x.x...x
....x.....x.....x.x...x.x.....x.........x...x...x...x.x...x.x...x
....x.xxxxx.xxxxx.xxxxx.xxxxx.xxxxx.....x.xxxxx.xxxxx.xxxxx.x...x
....x.x.........x.....x.....x.x...x.....x...x...x...x.....x.x...x
....x.x.........x.....x.....x.x...x.....x...x...x...x.....x.x...x
....x.xxxxx.xxxxx.....x.xxxxx.xxxxx.....x.......xxxxx.xxxxx.xxxxx
```

output

```
....x.xxxxx.xxxxx.xxxxx.x...x.xxxxx.xxxxx
....x.....x.....x.x.....x...x.x.........x
....x.....x.....x.x.....x...x.x.........x
....x.xxxxx.xxxxx.xxxxx.xxxxx.xxxxx.....x
....x.x.........x.....x.....x.....x.....x
....x.x.........x.....x.....x.....x.....x
....x.xxxxx.xxxxx.xxxxx.....x.xxxxx.....x
```

Problem B: Book Borders

Time limit: 1 s
Memory limit: 512 MiB

A book is being typeset using a fixed width font and a simple greedy algorithm to fill each line. The book contents is just a sequence of words, where each word contains one or more characters.

Before typesetting, we choose a *maximum line length* and denote this value with m. Each line can be at most m characters long including the space characters between the words. The typesetting algorithm simply processes words one by one and prints each word with exactly one space character between two consecutive words on the same line. If printing the word on the current line would exceed the maximum line length m, a new line is started instead.

```
|its.a.long...|        |its.a.long.way|
|way.to.the...|        |to.the.top.if.|
|top.if.you...|        |you.wanna.rock|
|wanna.rock.n.|        |n.roll........|
|roll.........|
```

Text from the example input with maximum line lengths 13 and 14

You are given a text to be typeset and are experimenting with different values of the maximum line length m. For a fixed m, the *leading sentence* is a sentence (a sequence of words separated with a single space character) formed by the first words of lines top to bottom. In the example above, when the sample text is typeset with the maximum line length 14, the leading sentence is "`its to you n`".

Given a text and two integers a and b, find the length of the leading sentence for every candidate maximum line length between a and b inclusive. The length of a sentence is the total number of characters it contains including the space characters.

Input

The first line contains the text to be typeset – a sequence of words separated by exactly one space character. Each word is a string consisting of one or more lowercase letters from the English alphabet.

The second line contains two integers a and b – the edges of the interval we are interested in, as described above.

It is guaranteed that $1 \le w \le a \le b \le z \le 500\,000$, where w is the length of the longest word in the text and z is the total number of characters in the text including the space characters.

Output

Output $b - a + 1$ lines – the k-th of those lines should contain a single integer – the total length of the leading sentence when the maximum line length is equal to $a - 1 + k$.

Example

input	output

`its a long way to the top if you wanna rock n roll`
`13 16`
	`22`
	`12`
	`12`
	`15`

Problem C: Counting Cities

Time limit: 1 s
Memory limit: 512 MiB

Carl travels a lot for his work. Each time he travels, she visits a single city before returning home.

Now his new boss want to know, how many different cities had Carl visited for work. He have a log for all Carl's trips and asks you to count number of cities Carl has visited at least once.

Input

The first line of input contains a single positive integer $T \leq 50$ indicating the number of test cases. The first line of each test case also contains a single positive integer n indicating the number of work trips Carl has taken so far. The following n lines describe these trips. The ith such line simply contains the name of the city Carl visited on his ith trip.

Note that city names in list used by boss only contain lowercase letters, have at least one letter, and do not contain spaces.

The number of trips is at most 100 and no city name contains more than 20 characters.

Output

For each test case, simply output a single line containing a single integer that is the number of distinct cities that Carl has visited on his work trips.

Example

input

```
2
7
zagreb
krakow
warsaw
krakow
prague
zagreb
krakow
3
barnaul
barnaul
barnaul
```

output

```
4
1
```

Problem D: Digit Division

Time limit: 1 s
Memory limit: 512 MiB

We are given a sequence of n decimal digits. The sequence needs to be partitioned into one or more contiguous subsequences such that each subsequence, when interpreted as a decimal number, is divisible by a given integer m.

Find the number of different such partitions modulo $10^9 + 7$. When determining if two partitions are different, we only consider the locations of subsequence boundaries rather than the digits themselves, e.g. partitions 2|22 and 22|2 are considered different.

Input

The first line contains two integers n and m ($1 \le n \le 300\,000$, $1 \le m \le 1\,000\,000$) – the length of the sequence and the divisor respectively. The second line contains a string consisting of exactly n digits.

Output

Output a single integer – the number of different partitions modulo $10^9 + 7$.

Example

input
```
4 2
1246
```
output
```
4
```

input
```
4 7
2015
```
output
```
0
```

input
```
4 3
3003
```
output
```
8
```

Problem E: Electoral Estimations

Time limit: 1 s
Memory limit: 512 MiB

In the Byteland first round of president elections is started. It is possible that that the winner (the candidate receiving the most votes) receives less than the majority of the votes. Then second round will be called.

To evaluate electoral estimations, Byteland scientists modelled the vote. They asked you to write a program, which, given the results of the first round, determines the leader (in case there's no tie), and says, if it is final victory.

Input

The first line of input contains a single positive integer $T \leq 500$ indicating the number of test cases. The first line of each test case also contains a single positive integer n indicating the number of candidates in the election. This is followed by n lines, with the ith line containing a single nonnegative integer indicating the number of votes candidate i received.

There are at least 2 and no more than 10 candidates in each case, and each candidate will not receive more than 50 000 votes. There will be at least one vote cast in each election.

Output

Provide a line of output for each test case. If the winner receives more than half of the votes, print the phrase "`Victory`" followed by the candidate number of the winner. If the winner does not receive more than half of the votes, print the phrase "`Leader` followed by the candidate number of the winner. If a winner cannot be determined because no single candidate has more vote than others, print the phrase "`Tie`". The candidates are numbered by sequential integers from 1 to n.

Example

input

```
4
3
10
21
10
3
20
10
10
3
10
10
10
4
15
15
15
45
```

output

```
Victory 2
Leader 1
Tie
Leader 4
```

Problem F: Funny Function

Time limit: 1 s
Memory limit: 512 MiB

Frank calls a number *funny* if its K-based notation doesnt contain two successive zeros.

Today he created his own function F and called it *funny function*. $F(N, K)$ is equal to amount of funny numbers of base K, containing exactly N digits (leading zeroes are not allowed). Given N and K, find $F(N, K)$.

Input

Input contains two integers N and K — number of digits and base of the system ($2 \le K \le 10$, $2 \le N$, $N + K \le 18$)

Output

Print $F(N, K)$.

Example

input

```
3
6
```

output

```
175
```

Problem G: Gregor's Game

Time limit: 2 s
Memory limit: 512 MiB

Gregor invented new card game for kids. The rules are quite simple, each player is dealt a hand of cards. Each card has one picture on each side. They take turns playing cards and the first one to run out of cards is the winner.

A player's turn consists of picking a subset of cards from their hand and laying them on the table. The only rule is that the cards must be placed on the table such that no two cards are showing the same picture.

Help Gregor to determine if he can play his entire hand on his first round.

Input

The first line of the input contains a single positive integer T ($T \leq 10$) indicating the number of test cases.

Each test case begins with a single integer n denoting the number of cards in Gregor's hand. Here $1 \leq n \leq 50\,000$. Following this are n lines, each describing a card in Gregor's hand.

The pictures on the cards are represented by integers. The ith card is given by two integers p_i, q_i where $1 \leq p_i, q_i \leq 2n$.

Output

For each test case you should output a single line with the word "Yes" if it is possible for Gregor to play his entire hand in one turn, or "No" if Gregor cannot play his entire hand in one turn.

Example

input

```
3
3
1 2
1 3
2 3
3
1 2
1 2
1 2
1
1 1
```

output

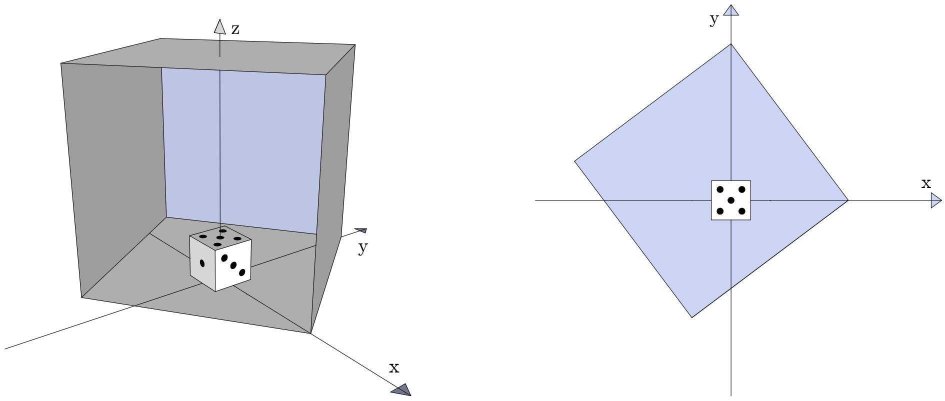
```
Yes
No
Yes
```

Problem H: Hovering Hornet

Time limit: 1 s
Memory limit: 512 MiB

You have managed to trap a hornet inside a box lying on the top of your dining table. Unfortunately, your playing dice is also trapped inside – you cannot retrieve it and continue your game of Monopoly without risking the hornet's wrath. Instead, you pass your time calculating the expected number of spots on the dice visible to the hornet.

The hornet, the dice and the box are located in the standard three-dimensional coordinate system with the x coordinate growing eastwards, the y coordinate growing northwards and the z coordinate growing upwards. The surface of the table corresponds to the x-y plane.



Perspective and the birds-eye view of the second example input

The dice is a $1 \times 1 \times 1$ cube, resting on the table with the center of the bottom side exactly in the origin. Hence, the coordinates of its two opposite corners are $(-0.5, -0.5, 0)$ and $(0.5, 0.5, 1)$. The top side of the dice has 5 spots, the south side 1 spot, the east side 3 spots, the north side 6 spots, the west side 4 spots and the (invisible and irrelevant) bottom side 2 spots.

The box is a $5 \times 5 \times 5$ cube also resting on the table with the dice in its interior. The box is specified by giving the coordinates of its bottom side – a 5×5 square.

Assume the hornet is hovering at a uniformly random point in the (continuous) space inside the box not occupied by the dice. Calculate the expected number of spots visible by the hornet. The dice is opaque and, hence, the hornet sees a spot only if the segment connecting the center of the spot and the location of the hornet does not intersect the interior of the dice.

Input

Input consists of 4 lines. The k-th line contains two floating-point numbers x_k and y_k $(-5 \le x_k, y_k \le 5)$ – coordinates of the k-th corner of the bottom side of the box in the x-y plane. The coordinates are given in the counterclockwise direction and they describe a square with the side length of exactly 5.

The box fully contains the dice. The surfaces of the box and the dice do not intersect or touch except along the bottom sides.

Output

Output a single floating point number – the expected number of spots visible. The solution will be accepted if the absolute or the relative difference from the judges solution is less than 10^{-6}.

Example

input	input

`-2.5 -1.5`
`2.5 -1.5`
`2.5 3.5`
`-2.5 3.5`
	`3 0`
	`0 4`
	`-4 1`
	`-1 -3`

output	**output**
`10.6854838710`	`10.1226478495`

Problem I: Ice Igloos

Time limit: 10 s
Memory limit: 512 MiB

A fishing village built on the surface of a frozen lake far north in the arctic is endangered by global warming – fractures are starting to form on the lake surface. The village consists of n igloos of spherical shape, each occupying a circular area of the surface.

An igloo can be represented as a circle in the coordinate plane: the center of the circle is a point with integer coordinates, while the radius is a positive floating-point number less than 1 with exactly one fractional digit.

Given the locations of possible ice fractures, the villagers would like to know how many igloos are affected by each. Formally, given q queries where each query is a straight line segment defined by the two endpoints, find the number of igloos each segment intersects. A segment intersects an igloo if it has at least one point in common with the interior of the circle.

Input

The first line contains an integer n ($1 \leq n \leq 100\,000$) - the number of igloos. Each of the following n lines contains three numbers x, y and r – the coordinates of the center and the radius of one igloo. The coordinates x and y are integers such that $1 \leq x, y \leq 500$, while r is a floating-point number with exactly one fractional digit such that $0 < r < 1$. No two igloos will intersect or touch.

The following line contains an integer q ($1 \leq q \leq 100\,000$) - the number of queries. Each of the following q lines contains four integers x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 500$) - the coordinates of the two endpoints of the segment. The two endpoints will be different. Endpoints may be inside igloos.

You may assume that, for every igloo i and the segment s, the square of the distance between s and the center of i is either less than $r^2 - 10^{-5}$ or greater than $r^2 + 10^{-5}$ where r is the radius of the igloo i.

Output

Output should consist of q lines. The k-th line should contain a single integer – the number of igloos that are intersected by the k-th segment.

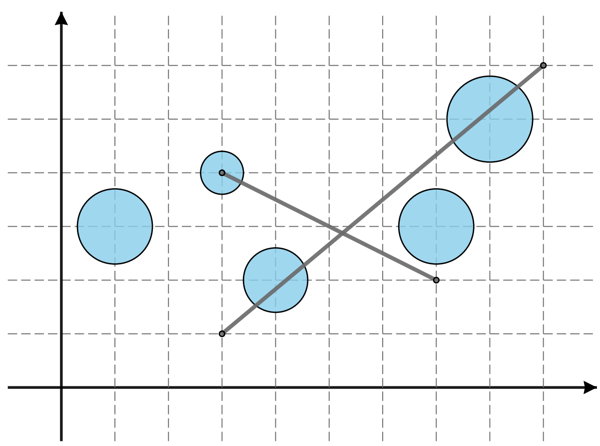
Example

input

```
5
4 2 0.6
7 3 0.7
8 5 0.8
1 3 0.7
3 4 0.4
2
3 1 9 6
3 4 7 2
```

output

```
2
1
```



Problem J: Juice Junctions

Time limit: 4 s
Memory limit: 512 MiB

You have been hired to upgrade an orange juice transport system in an old fruit processing plant. The system consists of pipes and junctions. All the pipes are bidirectional and have the same flow capacity of 1 liter per second. Pipes may join at junctions and each junction joins *at most three pipes*. The flow capacity of the junction itself is unlimited. Junctions are denoted with integers from 1 to n.

Before proposing the upgrades, you need to analyze the existing system. For two different junctions s and t, the s-t flow is defined as the maximum amount of juice (in liters per second) that could flow through the system if the source was installed at junction s and the sink at junction t. For example, in the system from the first example input below, the 1-6 flow is 3 while the 1-2 flow is 2.

Find the sum of all a-b flows for every pair of junctions a and b such that $a < b$.

Input

The first line contains two integers n and m ($2 \le n \le 3\,000$, $0 \le m \le 4\,500$) – the number of junctions and the number of pipes. Each of the following m lines contains two different integers a and b ($1 \le a, b \le n$) which describe a pipe connecting junctions a and b.

Every junction will be connected with at most three other junctions. Every pair of junctions will be connected with at most one pipe.

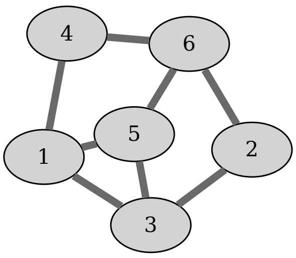
Output

Output a single integer – the sum of all a-b flows for every pair of junctions a and b such that $a < b$.

Example

input

```
6 8
1 3
2 3
4 1
5 6
2 6
5 1
6 4
5 3
```

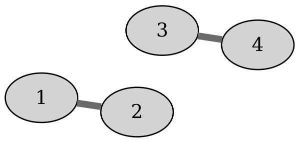


output

```
36
```

input

```
4 2
1 2
3 4
```



output

```
2
```

Problem K: Kernel Knights

Time limit: 2 s
Memory limit: 512 MiB

Jousting is a medieval contest that involves people on horseback trying to strike each other with wooden lances while riding at high speed. A total of $2n$ knights have entered a jousting tournament – n knights from each of the two great rival houses. Upon arrival, each knight has challenged a single knight from the other house to a duel.

A *kernel* is defined as some subset S of knights with the following two properties:

- No knight in S was challenged by another knight in S.
- Every knight not in S was challenged by some knight in S.

Given the set of the challenges issued, find one kernel. It is guaranteed that a kernel always exists.

Input

The first line contains an integer n ($1 \leq n \leq 100\,000$) – the number of knights of each house. The knights from the first house are denoted with integers 1 through n, knights from the second house with integers $n+1$ through $2n$.

The following line contains integers $f_1, f_2, \ldots, f_n$ – the k-th integer f_k is the index of the knight challenged by knight k ($n+1 \leq f_k \leq 2n$).

The following line contains integers $s_1, s_2, \ldots, s_n$ – the k-th integer s_k is the index of the knight challenged by knight $n+k$ ($1 \leq s_k \leq n$).

Output

Output the indices of the knights in the kernel on a single line. If there is more than one solution, you may output any one.

Example

input

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

output

```
1 2 4 8
```

Problem L: Larry's Lemma

Time limit: 1 s
Memory limit: 512 MiB

In his essay about 3 as the magic number, young mage Larry Lotter said about next lemma: that almost all integers can be represented as a number that ends in 3 in some numeric base, sometimes in more than one way. Consider the number 11, which is represented as 13 in base 8 and 23 in base 4.

In order to evaluate the essay, Professof of Numerology asked you to write a program which will find the smallest base for a given number so that the number's representation in that base ends in 3.

Input

Each line of the input contains one nonnegative integer n. The value $n = 0$ represents the end of the input and should not be processed. All input integers are less than 2^{31}. There are no more than 1 000 nonzero values of n.

Output

For each nonzero value of n in the input, print on a single line the smallest base for which the number has a representation that ends in 3. If there is no such base, print instead "`Fail`".

Example

input

```
11
123
104
2
3
0
```

output

```
4
4
101
Fail
4
```