Problem A. Arrays

Input file:	`arrays.in`
Output file:	`arrays.out`
Time limit:	6 seconds
Memory limit:	256 megabytes

You are given a sequence of $3 \cdot N$ integers $(X_1, X_2, \cdots, X_{3 \cdot N})$. Create three sequences $(A_1, A_2, \cdots, A_N)$, $(B_1, B_2, \cdots, B_N)$ and $(C_1, C_2, \cdots, C_N)$ such that:

- each of the integers from 1 to $3 \cdot N$ belongs to exactly one of the sequences A, B or C;
- the value of $S = \sum_{i=1}^{N} \left(X_{A_i} - X_{B_i}\right) \cdot X_{C_i}$ is the largest possible.

Input

The input file contains T test cases, all having the same value of N. The first line of the input file contains the integers T and N, constrained as shown in the adjacent table. Each of the following T lines describes one test case and contains $3 \cdot N$ integers, the members of the sequence X. All these values are in the range from 0 to 1000.

Constraints on N	Constraints on T
$1 \le N \le 10$	$1 \le T \le 1000$
$11 \le N \le 15$	$1 \le T \le 100$
$16 \le N \le 20$	$1 \le T \le 10$
$21 \le N \le 25$	$T = 1$

Output

The output file should consist of T lines. Each line should contain the largest possible value of S for the corresponding test case from the input file.

Example

arrays.in	arrays.out
1 2 4 1 8 2 0 5	46

Note. The maximal value is attained by taking $A = (1, 3)$, $B = (2, 5)$, $C = (4, 6)$.

Problem B. "Bulls and Cows"

Input file:	bulls.in
Output file:	bulls.out
Time limit:	1 second
Memory limit:	256 megabytes

You probably know the game "bulls and cows". Just in case, we explain the rules. The first player picks a four-digit number with all digits distinct (leading zero is allowed) and keeps it secret. The second player tries to guess the secret number. For each guess, the first player issues a response in the form "n bulls, m cows". A "bull" is a digit that is present in both the secret and the guess and occurs in the same position in both. A "cow" is a digit that is present in both numbers, but occurs in different positions.

For example, if the first player picked 5071, and the second guessed 6012, the response would be "one bull, one cow". Here the "bull" is the digit 0, as it is in the second position in both numbers, and the "cow" is the digit 1, as it is in the fourth position in the secret, but in the third position in the guess.

Write a program to count the number of cows and bulls for the given the secret and guess.

Input

The first line of the input file contains four digits, the number picked by the first player. The second line contains the number guessed by the second player in the same format.

Output

The first and only line of the output file should contain two integers separated by a space, the number of "bulls"and the number of "cows".

Example

bulls.in	bulls.out
5071 6012	1 1
4321 4321	4 0
1980 0879	0 3
1234 5678	0 0

Problem C. Courier's Route

Input file:	courier.in
Output file:	courier.out
Time limit:	1 second
Memory limit:	256 megabytes

There are N towns in a kingdom, numbered from 1 to N, with the capital having the number 1. Each town is enclosed in a city wall with 4 gates. The gates are also numbered, with the gates of the city i ($1 \le i \le N$) numbered $4 \cdot i - 3$, $4 \cdot i - 2$, $4 \cdot i - 1$, $4 \cdot i$. At each gate is exactly one road that leads to some gate of another town (note that there may be more than one road between two towns). All the roads can be travelled in both directions. Due to a system of bridges and tunnels, the roads do not intersect outside the towns.

A royal courier has to post copies of an Important Royal Decree on the outer sides of all gates of all towns. The courier can freely move from one gate to another in any town, but outside of towns, he's allowed to travel only along the roads. The courier starts from the capital and has to return to the capital after completing the assignment.

Is it possible for the courier to do this, passing each gate exactly once? Exiting a town through a gate only to post the decree on the outer side of the gate and immediately returning to the town counts as passing the gate once.

Input

The first line of the input file contains N ($2 \le N \le 1000$). Each of the following $2 \cdot N$ lines describes one road and contains two integers, separated by a space: the numbers of the two gates connected by the road.

Output

The first line of the output file should contain the word "Yes" or "No" (without the quotes) depending on whether the courier's assignment can be completed while passing each gate exactly once. In case it is possible, the second line of the file should contain $4 \cdot N$ integers, separated by spaces: the numbers of the gates in the order the courier passes them. If there are several solutions, output any one of them.

Example

courier.in	courier.out
4 1 9 2 10 3 13 4 8 5 11 6 14 7 16 15 12	Yes 1 3 13 14 16 15 12 9 10 11 5 6 7 8 4 2

Problem D. Dales and Hills

Input file:	dales.in
Output file:	dales.out
Time limit:	2 seconds
Memory limit:	256 megabytes

Let's consider a number sequence $a_1, \cdots, a_N$. We call the continuous subsequence $a_i, \cdots, a_j, \cdots, a_k$ $(1 \le i < j < k \le N)$ of the sequence a *hill* if $a_t < a_{t+1}$ for any $i \le t < j$ and $a_t > a_{t+1}$ for any $j \le t < k$. In this case we call $\min\{j - i, k - j\}$ the *height* of the hill. Similarly, we call the continuous subsequence a *dale* if $a_t > a_{t+1}$ for any $i \le t < j$ and $a_t < a_{t+1}$ for any $j \le t < k$. In this case we call $\min\{j - i, k - j\}$ the *depth* of the dale.

Compute the height of the highest hill and the depth of the deepest dale in the given sequence.

Input

The first line of the input file contains T $(1 \le T \le 100\ 000)$, the number of test cases. The test cases follow, occupying two lines each. The first of the two lines contains N $(1 \le N \le 1\ 000\ 000)$, the second the members of the sequence, separated by spaces. The sum of values of N over all test cases in the file does not exceed 1 000 000. The absolute values of the members of the sequences do not exceed 100 000.

Output

The output file should consist of T lines and each line should contain two integers, the height of the highest hill and the depth of the deepest dale. If there are no hills or no dales, output 0 in the corresponding position.

Example

dales.in	dales.out
2 10 4 4 1 6 3 2 1 2 5 7 10 2 3 4 5 6 7 8 9 10 9	1 3 1 0

Problem E. Extremal Permutations

Input file:	`extremal.in`
Output file:	`extremal.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

A member a_i of the sequence $a_1, a_2, \cdots, a_n$ is called a *local extreme* if either $a_i > a_{i-1}$ and $a_i > a_{i+1}$ (local maximum) or $a_i < a_{i-1}$ and $a_i < a_{i+1}$ (local minimum). A sequence $p_1, p_2, \cdots, p_n$ is called a *permutation* of the integers from 1 to n if each of the integers appears in the sequence exactly once. A permutation is called *extremal* if each member (except the first and the last) is a local extreme.

Compute the total number of extremal permutations of the integers from 1 to n and output the result modulo m.

Input

The first and only line of the input file contains the integers n ($1 \le n \le 10\,000$) and m ($1 \le m \le 10^9$).

Output

The output file should contain a single integer, the remainder from division of the total number of extremal permutations of integers from 1 to n by the given integer m.

Example

extremal.in	extremal.out
3 10	4
3 3	1

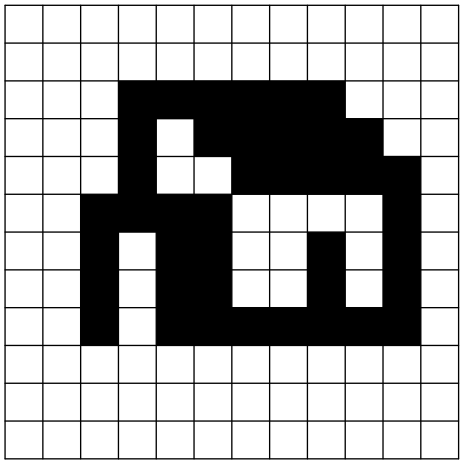
Note. The extremal permutations of $1 \cdots 3$ are (1, 3, 2), (2, 1, 3), (2, 3, 1) and (3, 1, 2).

Problem F. Figure ans Spots

Input file:	`figure.in`
Output file:	`figure.out`
Time limit:	1 second
Memory limit:	256 megabytes

Let's consider an infinite sheet of grid paper. Initially all the cells are white and you can paint some of them black.

Two cells are called 8-neigbours if they share a side or a corner. An 8-path between black cells A and B is a sequence of cells $X_0 = A, X_1, \cdots, X_{L-1}, X_L = B$ such that all cells in the sequence are black and for all $0 \le i < L$ the cells X_i and X_{i+1} are 8-neighbours. A set of black cells is called a figure if there is an 8-path from each of them into each other.



Two cells are called 4-neighbours if they share a side. A 4-path between white cells A and B is a sequence of cells $X_0 = A, X_1, \cdots, X_{L-1}, X_L = B$ such that all cells in the sequence are white and for all $0 \le i < L$ the cells X_i and X_{i+1} are 4-neigbours. A finite set of white cells is called a *spot* if:

- There is a 4-path from each of them into each other.
- The previous condition is broken when any other white cell is added to the set.

We say that a figure has the height H and the width W if it fits in a rectangle H rows high and W columns wide, but does not fit in a rectangle $H - 1$ rows high and W columns wide nor in a rectangle $H - 1$ rows high and W columns wide.

The image above shows a figure with height 7 and width 9 and containing two spots.

Given the numbers H, W and N, construct a figure with height exactly H and width exactly W and containing exactly N spots.

Input

The input file contains several test cases. The first line of the file contains T ($1 \le T \le 100$), the number of test cases. Each of the following T lines describes one test case and contains three integers H, W and N ($1 \le H, W \le 20$, $1 \le N \le 200$), separated by spaces.

Output

The output file should contain the following data for each test case:

- If it is possible to construct a figure with the given parameters, output any of the possible figures as H rows consisting of W characters each; output the character "`.`" for a black cell and the character "`#`" for a white cell.
- If it is impossible to construct the required figure, output a single line containing the word `Impossible`.

The output data for two different test cases should be separated by an empty line.

Example

figure.in	figure.out
3 7 9 2 20 20 22 5 5 10	#.....## #.#.....# #.##..... ....####. .#..##.#. .#..##.#. .#....... .#####.######.#####. ..###...####...###.. ...#.....##.....#... ....##..##..#...#... ...#.#.#..#.##.##... ...###.#....#.#.#... ...#.#.#..#.#...#... ...#.#..##..#...#... ....###..##..###...##.. ..#..#..#.#..#.#..#. ..#..#....###..#.... ..#..#..#.#....#..#. .###..##..#.....##.. ....#.....##.....#... ..###...####...###.. .#####.######.#####. Impossible

Problem G. Game for Little Johnny

Input file:	`game.in`
Output file:	`game.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Most of all in his life John likes his little son Johnny and his favorite integer number N. Well, actually John also likes math, so he wants his son to learn math from the early childhood.

To achieve this goal John plays the following game with Johnny. Each morning he tells the following tale to the son:

"In some magic kingdom there is a very tasty sweet cake and this cake costs exactly N bananas (a monetary unit of this kingdom). However, there are only two types of banknotes in this kingdom, one with the value of A bananas and the other with the value of B bananas ($A < B$).

According to the kingdom's laws, when buying a product, one has to use the set of banknotes with the total value equal to the product's cost so that no change is required and this set must contain at least one banknote of each of two types. Unfortunately, people in the kingdom are very bad at math, so nobody can find a way to buy the cake. Would you be able to buy it, Johnny?"

In other words, Johnny is given three integers N, A and B, $1 \le A < B \le N$, and is required to find integers $x, y \ge 1$ such that $A \cdot x + B \cdot y = N$. If Johnny succeeds at this task, John gives him a real tasty sweet cake as a prize.

For each next day John is going to use different numbers A and B in his tale, but the number N will always be the same (it's his favorite integer after all!). John likes his son, so he wants to always choose A and B in such way that Johnny's task has a solution. And of course John doesn't want to use the same pair (A, B) for two or more different days.

John is worried about the following question: for how long is he able to tell this tale, that is, how many different pairs (A, B) exist such that Johnny's task has a solution? However, John is not so good at math himself (it's a secret, don't tell it to Johnny!), so you have to help him to answer this question.

Input

The input file contains one integer N ($3 \le N \le 100\,000$).

Output

The output file must contain one integer, the amount of different pairs (A, B) such that Johnny's task has a solution.

Example

game.in	game.out
10	15

Note. The pairs from the example test case are (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 3), (2, 4), (2, 6), (2, 8), (3, 4), (3, 7), (4, 6).

Problem H. Hotel in Ves Lagos

Input file:	`hotel.in`
Output file:	`hotel.out`
Time limit:	1 second
Memory limit:	256 megabytes

A new hotel is being built in the city of Ves Lagos. The hotel will have an infinite number of rooms (it is out of fashion to build hotels with finite numbers of rooms). The new hotel also tries to cater for superstitious guests.

The most common superstition in Ves Lagos is that the number 13 brings bad luck. Accordingly, only numbers whose decimal forms do not contain the substring "13" will be used to label the rooms in the new hotel. For example, the hotel will have rooms numbered 1, 3, 14, 31, 123, but will not have the rooms 13, 132, 913, 1308, 1313.

Let's consider the list of all room numbers, ordered increasingly. Find the N-th number in this list (members of the list are indexed from 1).

Input

The input file contains several test cases. The first line of the file contains T ($1 \le T \le 100$), the number of test cases. Each of the following T lines describes one test case and contains the integer N ($1 \le N \le 10^{18}$).

Output

The output file should contain exactly T lines, with the i-th line containing exactly one integer, the answer for the i-th test case from the input file.

Example

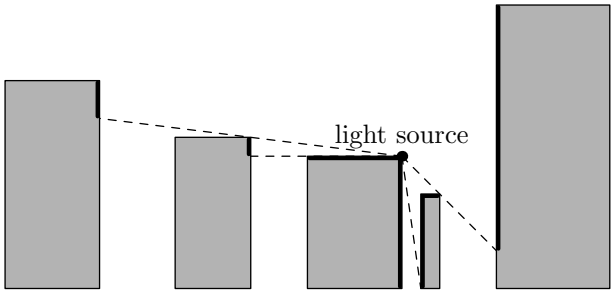
hotel.in	hotel.out
3 20 150 1	21 162 1

Problem I. Illumination of Buildings

Input file:	`illumination.in`
Output file:	`illumination.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

The city of Harbin is famous for the nighttime illumination of its buildings. Unfortunately, the economic crisis of the world has not left the welfare of the city undisturbed. An audit performed by the city council has revealed that lighting is the single largest expense in the budget of the city. Accordingly, it was decided to cut the costs for lightning as much as possible, but without sacrificing the quality of the illumination, as the council has no desire to damage the world fame of the city.

Let's consider a 2-dimensional model of the city where each building is described by three integers L, R, H and modeled as a rectangle with edges parallel to the coordinate axes and two opposing corners at the points $(L, 0)$ and (R, H). It may be assumed that the rectangles in the model do not intersect and even do not touch each other. Line segments $[(L, 0); (L, H)]$ and $[(R, 0); (R, H)]$ are called the side edges and line segment $[(L, H); (R, H)]$ the top edge.



The city council plans to install a number of light sources to illuminate the buildings. Each light source is to be installed on a top edge of a building (possibly on an endpoint of the top edge). There may be any number of light sources on one building. It is known that a light source installed at (x_1, y_1) will illuminate all points (x_2, y_2) where the line segment $[(x_1, y_1); (x_2, y_2)]$ does not contain any internal points of any buildings. It is allowed for the segment to contain any (even infinite) number of edge and corner points of buildings.

The figure above shows a light source and all points illuminated by it.

You are asked to install the minimal number of light sources to ensure that both sides of each building are completely illuminated. A side of a building is completely illuminated if for every point P on the side (including endpoints) there exists at least one light source L that illuminates the point P.

Input

The input file contains several test cases. The first line of the file contains T ($1 \le T \le 10\ 000$), the number of test cases. The line is followed by T blocks, each describing a test case.

The first line of a block contains N ($1 \le N \le 1000$), the number of buildings in the city. Each of the following N lines describes one building and contains three integers L, R and H ($1 \le L < R \le 10\ 000$, $1 \le H \le 10\ 000$).

It may be assumed that the sum of squares of values of N over all test cases in an input file does not exceed 1 000 000.

Output

The output file should contain one line for each test case given in the input file. Each line should contain a single integer, the minimal number of light sources required to illuminate both side edges of all the buildings in the city.

Example

illumination.in	illumination.out
2 4 3 4 1 5 6 1 7 8 1 1 2 10 6 3 4 1 5 6 1 7 8 1 1 2 10 11 12 10 9 10 1	5 4

Problem J. Journal

Input file:	`journal.in`
Output file:	`journal.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

There's an article of N words, with the i-th word consisting of L_i letters. The article has to be printed in a journal on a page W characters wide (for simplicity, let's assume that the page can contain as many lines as necessary and that all characters have the same width). The algorithm of laying out the text on the page of the journal is as follows. The words are placed in order: 1-st, 2-nd, $\cdots$, N-th. The location for the i-th word is determined by the following rules:

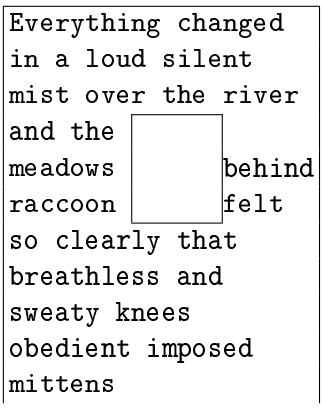
1. The word is placed in a block of L_i consecutive unoccupied character positions in the same row.
2. The first character of the i-th word and the last character of the $(i-1)$-th word can't be in adjacent positions of the same row. (Obviously, this applies only for $i > 1$.)
3. The block for the i-th word must not stand earlier than the block for the $(i-1)$-th word. A block A is considered to stand earlier than a block B if A is in a higher row than B or if they are in the same row and A is located to the left from B. (This also applies only for $i > 1$.)
4. Out of all blocks satisfying to the previous conditions, the earliest one is chosen.

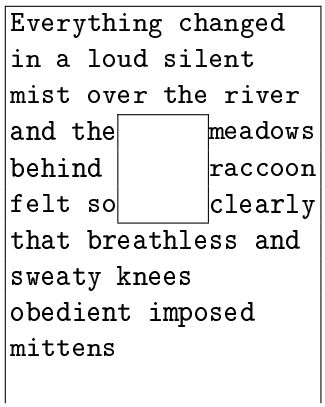
The layout of a sample text on a page 20 characters wide is shown below:

```
Everything changed
in a loud silent
mist over the river
and the meadows
behind raccoon felt
so clearly that
breathless and
sweaty knees
obedient imposed
mittens
```

In addition to the article, an illustration has to be placed on the page. The illustration takes up a rectangular area R rows high and C columns wide. The illustration is placed on the page before the text is laid out and it may be placed anywhere on the page. After the illustration is placed, all character positions covered by it are considered occupied and then the text is laid out according to the algorithm described above.

A row on the page is considered used if, after the illustration is placed and the text laid out, at least one character of the row is occupied. The total number of rows used may depend on the position of the illustration. For example, two possible final layouts of the illustration and the text are shown below, with one using 11 and the other 10 rows:





Given the dimensions of the illustration and a description of the text, place the illustration in a way that minimizes the number of rows occupied by the final layout.

Input

The input file contains several test cases. The first line of the file contains T ($1 \le T \le 1\,000$), the number of test cases. Descriptions of the T test cases follow.

The description of each test case starts with the integers N, W, R, C ($1 \le N \le 10\,000$, $1 \le W, R \le 1\,000$, $1 \le C \le W$), followed by the description of the text of the article as N integers $L_1, L_2, \cdots, L_N$ ($1 \le L_i \le W$). The numbers may be separated by spaces and/or newlines.

It may be assumed that the sum of values of N over all test cases in the file does not exceed 10 000.

Output

For each test case, output a single line containing the minimal number of rows needed to lay out the article.

Example

journal.in	journal.out
1 26 20 3 6 10 7 2 1 4 6 4 4 3 5 3 3 7 6 7 4 2 7 4 10 3 6 5 8 7 7	10

Problem K. Kids and Prizes

Input file:	`kids.in`
Output file:	`kids.out`
Time limit:	1 second
Memory limit:	256 megabytes

ICPC (International Cardboard Producing Company) is in the business of producing cardboard boxes. Recently the company organized a contest for kids for the best design of a cardboard box and selected M winners. There are N prizes for the winners, each one carefully packed in a cardboard box (made by the ICPC, of course). The awarding process will be as follows:

- All the boxes with prizes will be stored in a separate room.
- The winners will enter the room, one at a time.
- Each winner selects one of the boxes.
- The selected box is opened by a representative of the organizing committee.
- If the box contains a prize, the winner takes it.
- If the box is empty (because the same box has already been selected by one or more previous winners), the winner will instead get a certificate printed on a sheet of excellent cardboard (made by ICPC, of course).
- Whether there is a prize or not, the box is re-sealed and returned to the room.

The management of the company would like to know how many prizes will be given by the above process. It is assumed that each winner picks a box at random and that all boxes are equally likely to be picked. Compute the mathematical expectation of the number of prizes given (the certificates are not counted as prizes, of course).

Input

The first and only line of the input file contains the values of N and M ($1 \le N, M \le 100\,000$).

Output

The first and only line of the output file should contain a single real number: the expected number of prizes given out. The answer is accepted as correct if either the absolute or the relative error is less than or equal to 10^{-9}.

Example

kids.in	kids.out
5 7	3.951424
4 3	2.3125

Problem L. L-Shapes

Input file:	`lshape.in`
Output file:	`lshape.out`
Time limit:	2 seconds
Memory limit:	256 Megabytes

Let's say that two line segments of non-zero length form an *L-shape* if they are at a 90° angle to each other and one endpoint of one segment coincides with one endpoint of the other.

There are N line segments on a plane. The segments are numbered from 1 to N. Count the number of distinct pairs of segments that form L-shapes. Two pairs are considered distinct if they contain segments with different numbers.

Input

The first line of the input line contains the integer N ($1 \le N \le 5000$). Each of the following N lines describes one segment and contains four integers x_1, y_1, x_2, y_2 ($-10\,000 \le x_1, y_1, x_2, y_2 \le 10\,000$), where (x_1, y_1) and (x_2, y_2) are endpoints of the segment. It may be assumed that for each segment $x_1 \ne x_2$ or $y_1 \ne y_2$.

Output

The output file should contain a single integer, the total number of distinct pairs of line segments forming L-shapes.

Example

lshape.in	lshape.out
7 0 4 0 7 4 4 1 6 1 6 -3 0 4 4 0 4 0 0 0 4 0 0 0 2 0 4 4 4	5

Note. In the example the L-shapes are formed by the following pairs of segments: (1, 4), (1, 7), (2, 3), (4, 5), (5, 7). Note that the segments 4 and 7 coincide, but the pairs (1, 4) and (1, 7), for example, are still considered distinct.