The 43rd ACM International Collegiate Programming Contest
Asia Jiaozuo Regional Contest

November 25

Problems

A Xu Xiake in Henan Province
B Ultraman vs. Aodzilla and Bodzilla
C Supreme Command
D Keiichi Tsuchiya the Drift King
E Resistors in Parallel
F Honeycomb
G Shortest Paths on Random Forests
H Can You Solve the Harder Problem?
I Distance
J Carpets Removal
K Counting Failures on a Trie
L Connected Subgraphs

Do not open before the contest has started.

Problem A. Xu Xiake in Henan Province

Input file: `standard input`
Output file: `standard output`

Shaolin Monastery, also known as the Shaolin Temple, is a Chan ("Zen") Buddhist temple in Dengfeng County, Henan Province. Believed to have been founded in the 5-th century CE, the Shaolin Temple is the main temple of the Shaolin school of Buddhism to this day.

Longmen Grottoes, are some of the finest examples of Chinese Buddhist art. Housing tens of thousands of statues of Buddha and his disciples, they are located 12 kilometres (7.5 mi) south of present-day Luoyang in Henan province.

White Horse Temple is, according to tradition, the first Buddhist temple in China, established in 68 AD under the patronage of Emperor Ming in the Eastern Han dynasty capital Luoyang.

The Yuntai Mountain is situated in **Xiuwu County, Jiaozuo, Henan Province**. The Yuntai Geo Park scenic area is classified as an AAAAA scenic area by the China National Tourism Administration. Situated within Yuntai Geo Park, with a fall of 314 metres, Yuntai Waterfall is claimed as the tallest waterfall in China.

They are the most famous local attractions in Henan Province.

Now it's time to estimate the level of some travellers. All travellers can be classified based on the number of scenic spots that have been visited by each of them.

- A traveller that visited exactly 0 above-mentioned spot is a "Typically Otaku".
- A traveller that visited exactly 1 above-mentioned spot is a "Eye-opener".
- A traveller that visited exactly 2 above-mentioned spots is a "Young Traveller".
- A traveller that visited exactly 3 above-mentioned spots is a "Excellent Traveller".
- A traveller that visited all 4 above-mentioned spots is a "Contemporary Xu Xiake".

Please identify the level of a traveller.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 10^4.

For each test case, the only one line contains four integers A_1, A_2, A_3 and A_4, where A_i is the number of times that the traveller has visited the i-th scenic spot, and $0 \leq A_1, A_2, A_3, A_4 \leq 100$. If A_i is zero, it means that the traveller has never been visiting the i-th spot.

Output

For each test case, output a line containing one string denoting the classification of the traveller which should be one of the strings "`Typically Otaku`", "`Eye-opener`", "`Young Traveller`", "`Excellent Traveller`" and "`Contemporary Xu Xiake`" (without quotes).

Example

standard input	standard output
5 0 0 0 0 0 0 0 1 1 1 0 0 2 1 1 0 1 2 3 4	Typically Otaku Eye-opener Young Traveller Excellent Traveller Contemporary Xu Xiake

Problem B. Ultraman vs. Aodzilla and Bodzilla

Input file: `standard input`
Output file: `standard output`

Six months ago our hero, formerly known as Huriyyah, was beating all monsters in the land. Now he changed his name to Ultraman and left his beloved land. He is ready to take on a new challenge.

In a remote land, local citizens are suffering from the harassment of two powerful and horrible monsters: Aodzilla and Bodzilla. They eat small children who go out alone and even kill innocent persons. The apprehension of being attacked has overwhelmed people for several decades.

For the good of these unfortunate citizens, Ultraman sets out for the forest which is the main lair of Aodzilla and Bodzilla. In the forest, he faces these two fierce and cruel monsters and fights with them. The health points of Aodzilla and Bodzilla are HP_A and HP_B respectively, and their attack values are ATK_A and ATK_B respectively.

They fight in a cave through turn-based battles. During each second, the Ultraman will be attacked by monsters at first, and the damage is the sum of attack values of all alive monsters. Then he will select **exactly one** monster which is still alive and attack it. The selected monster will suffer a damage of value i (i.e. its health point will be decreased by i) where i represents that Ultraman has launched i attacks in total, from the beginning to the present, to these two monsters (and the current attack is the i-th one). That is to say, during the 1-st second, one of these two monsters will be under an attack of damage 1, during the 2-nd second, one of them, if alive, will be under an attack of damage 2, during the 3-rd second, one of them, if alive, will be under an attack of damage 3, and so on. If at some time, the health point of a monster is less than or equal to zero, it will die immediately. The Ultraman will win if both monsters have been killed.

Now, you are asked to develop a strategy to minimize the total damage Ultraman should suffer before he wins the battle. A strategy can be described as a string whose length is the total time that the battle will last. The i-th character in the string is '`A`', if the Ultraman chooses to attack Aodzilla during the i-th second; otherwise, the i-th character is '`B`', which means Bodzilla will be the target during that second. You are also asked to find the optimal strategy whose string description is the smallest in lexicographical order among all possible optimal strategies.

For two distinct strings s and t, if one string is a prefix of the other, then the one with a shorter length is smaller in lexicographical order. In other cases, s is smaller than t in lexicographical order if the first character of s is smaller than the first character of t, or in case they are equivalent, the second character of s is smaller than the second character of t, etc. The case t is smaller than s in lexicographical order is defined similarly as the former case.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 10^5.

For each test case, the only one line contains four integers HP_A, HP_B, ATK_A and ATK_B, where $1 \le HP_A, HP_B, ATK_A, ATK_B \le 10^9$.

We guarantee that there are at most 100 test cases with $\max\{HP_A, HP_B\} > 10^3$.

Output

For each test case, output a line containing an integer indicating the minimal total damage Ultraman should suffer, and a string describing the optimal strategy such that the string description is the smallest in lexicographical order among all possible optimal strategies. You should output exactly one whitespace between the number and the string.

Example

standard input	standard output
2 5 15 5 25 5 15 25 5	155 BBBBBA 105 AAABBB

Problem C. Supreme Command

Input file: `standard input`
Output file: `standard output`

Lewis likes playing chess. Now he has n rooks on the chessboard with n rows and n columns. All rows of the chessboard are labelled with 1 through n from top to bottom. All columns of the chessboard are labelled with 1 through n from left to right. All rooks are labelled with 1 through n as well. At the very beginning, **each row or column contains exactly one rook**. However, Lewis allows a square with two or more rooks during the game.

Now he starts to play a game named Supreme Command. He will provide several supreme commands to all rooks. All possible commands are in the following four different formats.

- `L k`: Every rook moves k squares to the left;
- `R k`: Every rook moves k squares to the right;
- `U k`: Every rook moves k squares upward;
- `D k`: Every rook moves k squares downward.

For a Supreme Command with given number k, if a rook, after moving less than k squares, had arrived at a boundary (which locates in the left-most columns, right-most column, top row or bottom row) such that the rook cannot move further, it would stay there and not move outside the chessboard.

He will also have several queries about rooks. The only two possible formats about queries are listed as follows.

- `?  k`: Ask the current position of the k-th rook;
- `!`: Ask how many pairs of rooks there are currently located in the same square.

Your task in this problem is to answer these queries correctly.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 1000.

For each test case, the first line contains two integers n which is described as above, and m indicating the total number of supreme commands and queries, where $1 \le n, m \le 3 \times 10^5$.

Each of the following n lines contains two integers x and y, describing a rook located at the intersection of the x-th row and the y-th column, where $1 \le x, y \le n$.

Then the following m lines describe all Supreme Commands and queries in chronological order, where all given parameters k are integers ranged from 1 to n.

We guarantee that the sum of n and the sum of m in all test cases are up to 10^6 respectively.

Output

For each test case, output several lines to answer all queries.

For each query of the first type ("? k"), output a line containing two integers x and y, which indicate the current position of the k-th rook is the intersection of the x-th row and the y-th column. You should output exactly one whitespace between these two numbers.

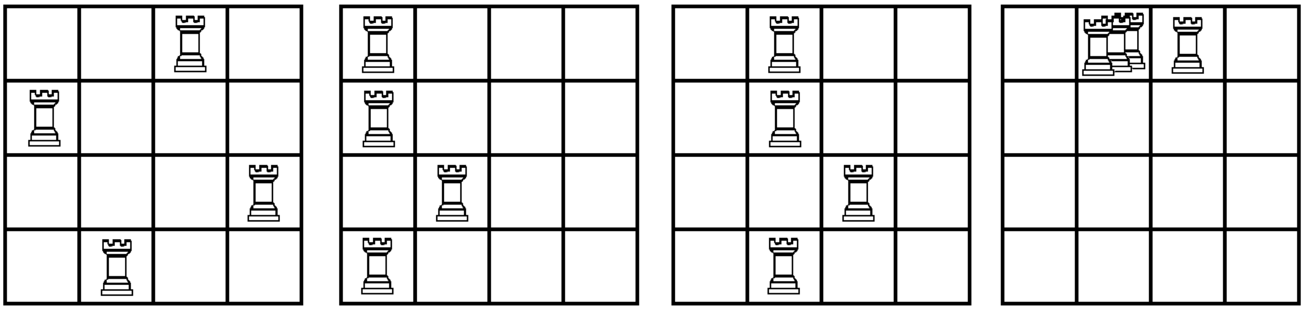
For each query of the second type ("!"), output a line containing an integer which indicates the number of pairs of rocks that are currently located in the same square.

Example

standard input	standard output
1	3 2
4 9	2 1
3 4	3 3
2 1	4 2
4 2	0
1 3	3
L 2	
? 1	
? 2	
R 1	
? 1	
? 3	
!	
U 3	
!	

Note

The following figures illustrate the chessboard at the beginning and after each Supreme Commands in the sample case.



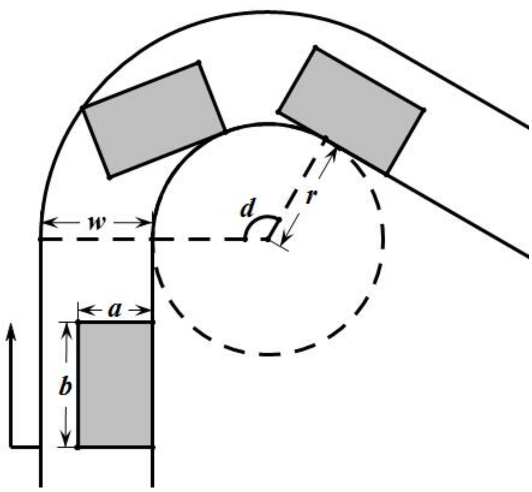
Problem D. Keiichi Tsuchiya the Drift King

Input file: `standard input`
Output file: `standard output`

Drifting is a driving style in which the driver uses the throttle, brakes, clutch, gear shifting and steering input to keep the car in a state of oversteer while manoeuvring from turn to turn. As a sport, the drifting was first practised in Japan in the late 80s before gaining worldwide popularity in the decade that followed.

Keiichi Tsuchiya is a Japanese driver who is better known as the Drift King. He played an important role in popularizing the art of drifting. This pioneer inspired many successful drivers. He appeared in the movie **The Fast and the Furious: Tokyo Drift** and he is often employed on various movie sets as both driver and stunt coordinator. Keiichi Tsuchiya's talent in the drifting is especially true of his big stunt, the ultimate drifting.

Here is what he could do. The drift car he drives is shaped like a rectangular box of width a inches and of length b inches. He makes a right turn of a curve whose internal boundary is an arc with d degrees in a circle with a radius of r inches. As a super-skilled driver, he maintains his car to keep the contact and tangency at the internal boundary. That is, the right front corner of the car should always run along the internal boundary and the direction of the car body should always be tangential to the internal boundary.



We have measured that the straightaways before and after the curve are long enough, and the width of the lane is invariable. As what we meet in real life, if a lane has a fixed width, for each point of its one side, the distance to the nearest point of the other side is exactly its width. Now you are asked to calculate the minimal width w of the lane so that the Drift King could drive throughout the curve without drifting out of the lane.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 10^4.

For each test case, the only one line contains four integers a, b, r and d, where $0 < a, b, r < 100$ and $0 < d < 180$.

Output

For each test case, output a line containing the minimal width (in inches) of the lane with an absolute or relative error of at most 10^{-6}. Precisely speaking, assuming that your answer is a and the jury's answer is b, your answer will be considered correct if $\frac{|a-b|}{\max\{1,|b|\}} \leq 10^{-6}$.

Example

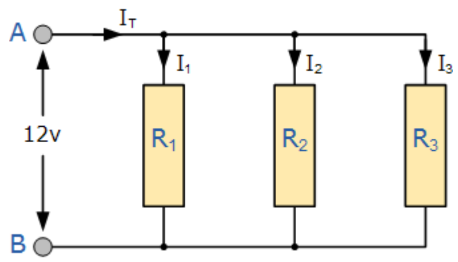
standard input	standard output
4 1 2 2 120 1 2 2 60 1 2 2 30 1 2 2 15	1.605551275464 1.605551275464 1.598076211353 1.415415569072

Problem E. Resistors in Parallel

Input file:	`standard input`
Output file:	`standard output`

In this physics problem, what we are concerned about are only resistors. If you are poor at physics, do not worry, since solving this problem does not require you to have advanced abilities in physics.

Resistors are said to be connected together in parallel when both of their terminals are respectively connected to each terminal of the other resistors.



We have the following parallel resistor equation for k resistors with resistances $R_1, R_2, \cdots, R_k$ in parallel and their combined resistance R:

$$R = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_k}}.$$

Now you have n resistors, the i-th of which has a resistance of r_i ohms with the equation

$$r_i = \begin{cases} \infty & \texttt{if } i \texttt{ can be divided by } d^2 \texttt{ for some integers } d \geq 2, \\ i & \texttt{otherwise.} \end{cases}$$

You also have n selections, the i-th of which is a set of resistors S_i such that

$$S_i = \{\texttt{the } j\texttt{-th resistor} \mid j \texttt{ is a divisor of } i\}.$$

Please find a selection in which the resistors form a parallel resistor with the minimum resistance and output the reduced fraction $\frac{p}{q}$ of its resistance.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 100.

For each test case, the only one line contains an integer n, where $1 \leq n \leq 10^{100}$.

Output

For each test case, output a line containing a reduced fraction of the form `p/q` indicating the minimum possible resistance, where `p` and `q` should be positive numbers that are coprime.

Example

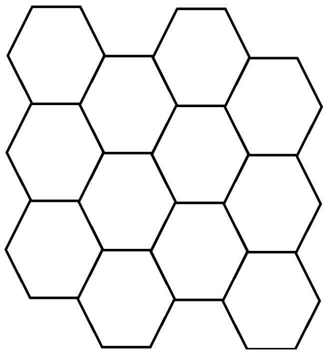
standard input	standard output
3 10 100 1000	1/2 5/12 35/96

This page is intentionally left blank

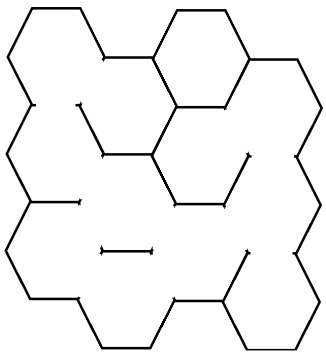
Problem F. Honeycomb

Input file: `standard input`
Output file: `standard output`

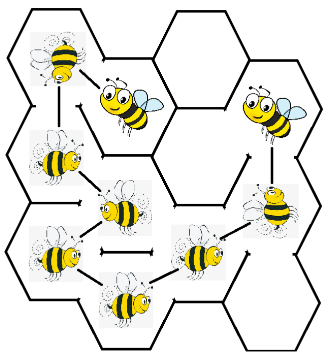
A honeycomb is a mass wax cells built by honey bees, which can be described as a regular tiling of the Euclidean plane, in which three hexagons meet at each internal vertex. The internal angle of a hexagon is 120 degrees, so three hexagons at a point make a full 360 degrees. The following figure shows a complete honeycomb with 3 rows and 4 columns.



Here we guarantee that the first cell in the second column always locates in the bottom right side of the first cell in the first column, as shown above. A general honeycomb may, on the basis of a complete honeycomb, lose some walls between adjacent cells, but the honeycomb is still in a closed form. A possible case looks like the figure below.



Hamilton is a brave bee living in a general honeycomb. Now he wants to move from a starting point to a specified destination. The image below gives a feasible path in a 3×4 honeycomb from the 1-st cell in the 2-nd column to the 1-st cell in the 4-th column.



Please help him find the minimum number of cells that a feasible path has to pass through (including the starting point and the destination) from the specified starting point to the destination.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 10^4.

For each test case, the first line contains two integers r and c indicating the number of rows and the number of columns of the honeycomb, where $2 \le r, c \le 10^3$.

The following $(4r+3)$ lines describe the whole given honeycomb, where each line contains at most $(6c+3)$ characters. Odd lines contain grid vertices represented as plus signs ("+") and zero or more horizontal edges, while even lines contain two or more diagonal edges. Specifically, a cell is described as 6 vertices and at most 6 edges. Its upper boundary or lower boundary is represented as three consecutive minus signs ("-"). Each one of its diagonal edges, if exists, is a single forward slash ("/") or a single backslash ("\") character. All edge characters will be placed exactly between the corresponding vertices. At the center of the starting cell (resp. the destination), a capital "S" (resp. a capital "T") as a special character is used to indicate the special cell. All other characters will be space characters. Note that if any input line could contain trailing whitespace, that whitespace will be omitted.

We guarantee that all outermost wall exist so that the given honeycomb is closed, and exactly one "S" and one "T" appear in the given honeycomb. Besides, the sum of $r \cdot c$ in all test cases is up to 2×10^6.

Output

For each test case, output a line containing the minimum number of cells that Hamilton has to visit moving from the starting cell ("S") to the destination ("T"), including the starting cell and the destination. If no feasible path exists, output `-1` instead.

Example

standard input	standard output

<pre>1
3 4
  +---+   +---+
 /   \   /   \
+   +---+   +---+
 \   \   /   \
  +   +   S   +---+   T   +
 /   \   /   /
+   +---+   +   +
 \   \   /   \
  +---+   +---+   +
 /   /
+   +---+   +   +
 \   /   \
  +---+   +---+   +
       \   /   \   /
        +---+   +---+</pre>
	7

Problem G. Shortest Paths on Random Forests

Input file: `standard input`
Output file: `standard output`

Here is a problem related to forest, which is a special type of graph. Before introducing this problem to you, we intend to show some definitions used in this problem. A labelled forest with n vertices is an acyclic undirected simple graph in which vertices are labelled by $1, 2, \cdots, n$. Two labelled forests are regarded as different if their numbers of vertices are different or, if they have the same number of vertices, for some integers i and for vertices labelled by i in these two forests, their neighbours have different labels (which means that the sets of labels corresponding to all neighbours of vertices labelled by i in these two forests are different).

Tree-like structures are constructed in computer programming constantly, which is the most fascinating part Bob has ever seen. Today, Bob wants to randomly choose a labelled forest G from all possible labelled forests having n vertices with equal probability. Then, he will set $\delta(i, j)$ to the number of edges on the shortest path from the vertex labelled i to the vertex labelled j if the shortest path exists, or set $\delta(i, j)$ to m otherwise. Bob is curious about the expected value of

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \delta^2(i, j),$$

but it's hard for him. Can you help Bob find out the expected value modulo 998244353?

More precisely, if the reduced fraction of the expected value is $\frac{p}{q}$, what you should provide is the minimum non-negative integer r such that $qr \equiv p \pmod{998244353}$.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 2×10^5.

For each test case, the only one line contains two integers n and m where $1 \le n \le 2 \times 10^5$ and $n \le m \le 998244352$.

We guarantee that the modular multiplicative inverse of q in each test case always exists, in other words, the condition $q \not\equiv 0 \pmod{998244353}$ is guaranteed to be true in all test cases.

Output

For each test case, output a line containing the answer modulo 998244353.

Example

standard input	standard output
4 1 1 2 3 3 7 4 16	0 5 66 576

This page is intentionally left blank

Problem H. Can You Solve the Harder Problem?

Input file: `standard input`
Output file: `standard output`

You are given a sequence **a** denoted by $(a_1, a_2, \cdots, a_n)$. A query with $1 \le l \le r \le n$ is defined as

$$Q_{max}(l, r) = \max\{a_l, a_{l+1}, \cdots, a_r\}.$$

An easy problem may ask you to calculate the sum of answers for all queries with integers $1 \le l \le r \le n$, but we would like to show you a harder one.

Define a classifier as a container that stores unique elements. Each element in the classifier has two values, named the key value and the mapped value. The **key value** of each element is a consecutive subsequence of **a** and the **mapped value** of that element is an integer indicating the maximum value in the subsequence. The classifier only stores elements with distinct key values, which means extra duplicated elements (with the same key value) would be removed from the classifier.

Denote $S(l, r)$ as $(a_l, a_{l+1}, \cdots, a_r)$, a consecutive subsequence of **a** meeting the condition that l and r are integers with $1 \le l \le r \le n$. Now we intend to use a classifier **CA** to store all the consecutive subsequences $S(l, r)$ of **a** with their $Q_{max}(l, r)$. You are asked to determine the **sum of mapped values** in **CA**.

Actually, what we defined above is a map of the form `map<vector<int>, int>` in C++ or `Map<ArrayList<Integer>, Integer>` in Java, so if you are seasoned using these data structures, you may realize that what we intend to do is to insert all possible elements $(S(l, r), Q_{max}(l, r))$ with integers $1 \le l \le r \le n$ into the classifier **CA** and then ask you to calculate the sum of mapped values in it.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 1000.

For each test case, the first line contains an integer n indicating the number length of the given sequence **a**, where $1 \le n \le 2 \times 10^5$.

The second line contains n positive integers $a_1, a_2, \cdots, a_n$ describing the sequence **a**, where $1 \le a_i \le 10^6$.

We guarantee that there are at most 10 test cases with $n > 1000$.

Output

For each test case, output a line containing the sum of mapped values in **CA**.

Example

standard input	standard output
2 3 1 2 3 3 2 3 3	14 14

Note

In the first sample case, the sum of mapped values in **CA** is equal to $1 + 2 + 3 + 2 + 3 + 3$.

In the second sample case, the sum of mapped values in **CA** is equal to $2 + 3 + 3 + 3 + 3$.

This page is intentionally left blank

Problem I. Distance

Input file: `standard input`
Output file: `standard output`

There are n points on a horizontal line, labelled with 1 through n from left to right.

The distance between the i-th point and the $(i+1)$-th point is a_i.

For each integer k ranged from 1 to n, you are asked to select exactly k different given points on the line to maximize the sum of distances between all pairs of selected points.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 1000.

For each test case, the first line contains an integer n indicating the number of points, where $2 \le n \le 10^5$.

The second line contains $(n-1)$ positive integers $a_1, a_2, \cdots, a_{n-1}$, where $1 \le a_i \le 10^4$.

We guarantee that the sum of n in all test cases is up to 10^6.

Output

For each test case, output a line containing n integers, the i-th of which is the maximum sum of distances in case $k = i$. You should output exactly one whitespace between every two adjacent numbers and avoid any trailing whitespace in this line.

Example

standard input	standard output
1 5 2 3 1 4	0 10 20 34 48

Note

The figure below describes the sample test case.

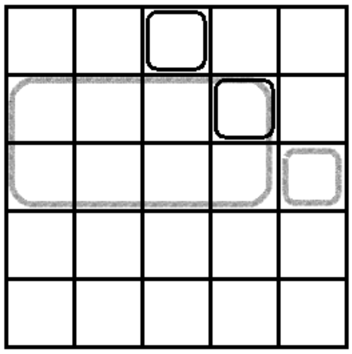
The only best selection for $k = 2$ should choose the leftmost and the rightmost points, while a possible best selection for $k = 3$ could contain any extra point in the middle.

This page is intentionally left blank

Problem J. Carpets Removal

Input file: `standard input`
Output file: `standard output`

Mike's living room is covered by m^2 square tiles. These tiles form a $m \times m$ grid in which rows are numbered with 1 through m from top to bottom and columns are numbered with 1 through m from left to right.



Above these tiles on the floor are laying n rectangular carpets whose sides are parallel to sides of the grid. Each carpet covers the intersection of several consecutive rows and several consecutive columns, forming a rectangle. Precisely speaking, the i-th carpet of them is described by four integers x_l, x_r, y_l, y_r with $1 \le x_l \le x_r \le m$ and $1 \le y_l \le y_r \le m$, indicating that the carpet covers all tiles which are both ranged from the x_l-th row to the x_r-th row and ranged from the y_l-th column to the y_r-th column.

Now Mike asks you to take away **exactly two** carpets of them, in order to minimize the number of tiles that would still be covered by at least one remaining carpet.

The figure provided above describes the sample case, in which those two rectangular regions with imaginary boundaries indicate an optimal removal of two carpets.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 1000.

For each test case, the first line contains two integers n and m indicating the number of carpets and the number of tiles in each row or column, where $3 \le n \le 3 \times 10^5$ and $1 \le m \le 1500$.

Each of the following n lines contains four integers x_l, x_r, y_l, y_r describing a carpet laying on the floor and its postion, where $1 \le x_l \le x_r \le m$ and $1 \le y_l \le y_r \le m$.

We guarantee that the sum of n in all test cases is up to 2×10^6, while the sum of m^2 in all test cases is up to 5×10^7.

Output

For each test case, output a line containing the minimum number of tiles that would still be covered by at least one remaining carpet after removal of two carpets.

Example

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

This page is intentionally left blank

Problem K. Counting Failures on a Trie

Input file: `standard input`
Output file: `standard output`

In computer science, a trie, also called prefix tree, is a kind of search tree, an ordered tree data structure, where the keys are usually strings.

Here we have a trie with $(n+1)$ nodes labelled 0 through n, and its root is the node labelled by 0. All edges in the trie are directed from nodes with lower heights to nodes with higher heights. Each of them contains only a lowercase letter, and any two edges with a common starting node have different characters.

We would like to introduce a special matching on the trie for an arbitrary string $T[1..k]$.

The matching will start at the root, consider all characters $T[1], T[2], \cdots, T[k]$ in order, and move along some edge which starts from the current node and contains the currently considered character. If the matching arrives at some node but fails to move along some edge, then:

- it will record a failure at the current character, skip the character and move back to the root; and
- it will consider the next character in the next match step because **failed characters should be skipped**.

Finally, after considering all characters in T, the matching will stay at a node indicating the destination of the matching on the trie. We denote the label of the destination as $Dest(T)$, and the total number of failures during the matching process as $CFail(T)$.

Now we give you a string with lowercase letters of length m denoted by $S[1..m]$, and you need to answer q queries. A query is defined as

$$Query(l, r) = (CFail(S[l..r]), Dest(S[l..r])),$$

which asks the total number of failures and the destination of the matching for $S[l..r]$, a substring of S.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 1000.

For each test case, the first line contains three integers n, m and q indicating the number of nodes in the trie, the length of given string S and the total number of queries respectively, where $1 \le n, m, q \le 10^5$.

The following n lines describe the trie. The i-th line of them contains an integer f_i, satisfying $0 \le f_i < i$, and a lowercase letter c_i describing the parent node of the i-th node and the character of the edge between them respectively.

The next line contains a string in all lowercase letters indicating the given string S of length m.

Each of the following q lines contains two integers l and r indicating a query $Query(l, r)$, where $1 \le l \le r \le m$.

We guarantee that the sum of n, the sum of m and the sum of q in all test cases are up to 10^6 respectively.

Output

For each test case, output several lines to answer all queries.

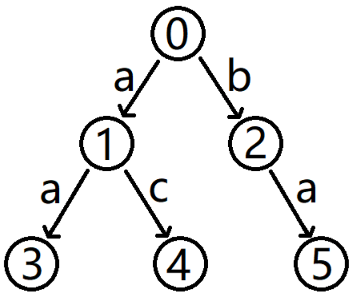
For each query output a line containing two integers in a line indicating $CFail(S[l..r])$ and $Dest(S[l..r])$ respectively. You should output exactly one whitespace between these two numbers.

Example

standard input	standard output
1 5 10 5 0 a 0 b 1 a 1 c 2 a aaacbaacab 1 5 1 6 1 7 3 9 4 10	2 2 2 5 3 0 2 1 4 0

Note

The figure below shows the trie given in the sample test case.



Paths for all queries are described as following, where each $\Longrightarrow$ indicates a failure.

- $Path(1,5) = Path($“aaacb”$) = 0 \longrightarrow 1 \longrightarrow 3 \Longrightarrow 0 \Longrightarrow 0 \longrightarrow 2$;
- $Path(1,6) = Path($“aaacba”$) = 0 \longrightarrow 1 \longrightarrow 3 \Longrightarrow 0 \Longrightarrow 0 \longrightarrow 2 \longrightarrow 5$;
- $Path(1,7) = Path($“aaacbaa”$) = 0 \longrightarrow 1 \longrightarrow 3 \Longrightarrow 0 \Longrightarrow 0 \longrightarrow 2 \longrightarrow 5 \Longrightarrow 0$;
- $Path(3,9) = Path($“acbaaca”$) = 0 \longrightarrow 1 \longrightarrow 4 \Longrightarrow 0 \longrightarrow 1 \longrightarrow 3 \Longrightarrow 0 \longrightarrow 1$;
- $Path(4,10) = Path($“cbaacab”$) = 0 \Longrightarrow 0 \longrightarrow 2 \longrightarrow 5 \Longrightarrow 0 \Longrightarrow 0 \longrightarrow 1 \Longrightarrow 0$.

Problem L. Connected Subgraphs

Input file:	`standard input`
Output file:	`standard output`

An algorithm master in graph theory would never endure any disconnected subgraph.

An esthetician would only consider edge-induced subgraphs as necessary subgraphs.

An OCD patient would always choose a subgraph from a given simple undirected graph randomly.

Those are why Picard asks you to calculate, for choosing four different edges from a given simple undirected graph with equal probability among all possible ways, the probability that the edge-induced subgraph formed by chosen edges is connected. Here we say a subset of edges in the graph together with all vertices that are endpoints of edges in the subset form an edge-induced subgraph.

To avoid any precision issue, Picard denotes the probability as p and the number of edges as m, and you should report the value $\left(p \cdot \binom{m}{4}\right) \bmod (10^9 + 7)$. It is easy to show that $p \cdot \binom{m}{4}$ is an integer.

Input

The input contains several test cases, and the first line contains a positive integer T indicating the number of test cases which is up to 10.

For each test case, the first line contains two integers n and m indicating the numbers of vertices and edges in the given simple undirected graph respectively, where $4 \le n \le 10^5$ and $4 \le m \le 2 \times 10^5$.

The following m lines describe all edges of the graph, the i-th line of which contains two integers u and v which represent an edge between the u-th vertex and the v-th vertex, where $1 \le u, v \le n$ and $u \ne v$.

We guarantee that the given graph contains no loops or multiple edges.

Output

For each test case, output a line containing an integer corresponding to the value $\left(p \cdot \binom{m}{4}\right) \bmod (10^9 + 7)$, where p indicates the probability which you are asked to calculate.

Example

standard input	standard output
2 4 4 1 2 2 3 3 4 4 1 4 6 1 2 1 3 1 4 2 3 2 4 3 4	1 15

This page is intentionally left blank