Problem A
Cloud System

Time Limit: **3 seconds**

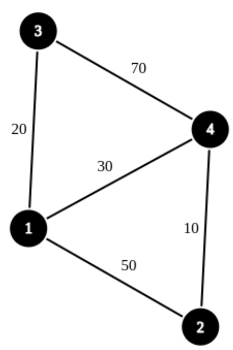
Mem limit: **512 Megabytes**

VNU-HCM University of Science has just built a new cloud computing system to run experiments of research projects related to deep-learning.

The system consists of N servers (numbered from 1 to N). Each pair of servers is connected by at most 1 cable (possibly 0).

The cable connecting server u and server v has the transmission capacity of $C_{u,v}$ megabits per nanosecond.

Let us define $F_{u,v}$ as the data transfer rate between server u and server v. To transfer data from server u to server v, data can be split into multi parts and transferred via multiple routes.



For example, to transfer data from server 1 to server 4, data can be split into 3 parts and transferred via 3 routes as follows:

- 1 - 4 (30 Mb per nanosecond)
- 1 - 3 - 4 (20 Mb per nanosecond)
- 1 - 2 - 4 (10 Mb per nanosecond)

so the transfer rate from 1 to 4 is 60 Mb per nanosecond or $F_{1,4} = 60$.

You are given the configuration of the network and the transmission capacity of cables, your task is to compute F.

Input

The input starts with T – the number of test cases. Then T test cases follow.

Each test case starts with an integer N, the number of servers in the network ($1 \leq N \leq 200$).

In the next N lines, each line contains N integers $C_{u,v}$ ($0 \leq C_{u,v} \leq 10000$).

It is guaranteed that $C_{u,v} = C_{v,u}$ and $C_{u,u} = 0$.

Note: The sum of N in the input does not exceeded 1000.

Output

For the t^{th} test case, print "`Case #t:`". The next N lines, print the $N \times N$ matrix F.

Sample input	Sample output
2 4 0 50 20 30 50 0 0 10 20 0 0 70 30 10 70 0 4 0 10 0 0 10 0 0 0 0 0 0 10 0 0 10 0	Case #1: 0 60 60 60 60 0 60 60 60 60 0 90 60 60 90 0 Case #2: 0 10 0 0 10 0 0 0 0 0 0 10 0 0 10 0

Problem B
Planning Tree

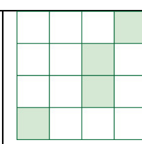
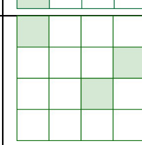
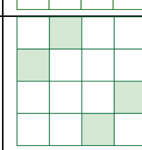
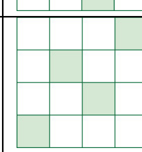
Time Limit: **1 second**
Mem limit: **512 Megabytes**

A new beautiful park has just been built in our city. Your team is now responsible for planting trees in the park.

The park can be considered as a grid with N rows and N columns. The rows are numbered from 1 to N from top to bottom. The columns are numbered from 1 to N from left to right. Let us denote the cell at i^{th} row and j^{th} column by (i, j).

Your team will plant exactly N trees, such that in each row and in each column, there is exactly one tree.

The entrance of the park is at cell $(1, 1)$ and the exit of the park is at cell (N, N); your team cannot plant trees at these two cells. Furthermore, there must exist a path from the entrance to the exit of the park such that a visitor only needs to move right or down. Obviously, visitors cannot walk through trees.

The following park is not valid, because column 3 has 2 trees	
The following park is not valid, because cell $(1, 1)$ is not empty, and there are only 3 trees	
The following park is not valid, because it is not possible to go from $(1, 1)$ to $(4, 4)$ moving only right or down	
The following park is valid	

Please determine the number of different valid parks. Two parks are considered different if there is at least one tree at different positions in the two parks.

Input

The first line of input contains a single integer T ($T \leq 10^5$) – the number of test cases.

T lines follow, each line contains a single integer N ($1 \leq N \leq 10^7$).

Output

Print exactly T lines, each line contains the number of ways to plant trees, satisfying all the given conditions. As the result can be very large, please prints the answer modulo $10^9 + 7$.

Sample Input	Sample Output
3 1 2 3 4	0 0 2 12

Problem C
Gold Miner

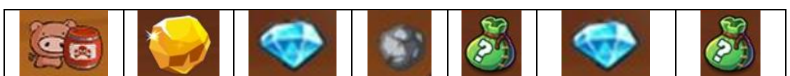
Time Limit: **3 seconds**

Mem limit: **512 Megabytes**

Mr. Phidang is a gold miner, and he is now at a gold mine. He has a powerful mining machine, but it seems heavy. Mr. Phidang is going to use the machine to mine gold or other precious minerals (e,g. diamonds, silver, etc.).

There are many types of minerals in the gold mine, and Mr. Phidang has his own preferences over them. The preference of each type is represented by a lowercase Latin letter, and the preference rank is considered by its alphabetical order. For example, Mr. Phidang's most preferable type of mineral will be '`a`', the '`b`' type has a little bit less preferred, '`c`' type is for defining even worse, and so on.

The gold mine can be described as a sequence of contiguous minable slots. The following figure describes the gold mine "`ebadcab`".



e	b	a	d	c	a	c

Mr. Phidang can start mining at any slot that he wants. However, after starting the mining process, he can only move to either of the two adjacent slots (left or right) to continue mining since the current slot needs a unit of time to be refilled, and his mining machine is too heavy so he cannot move far away. Given that the amount of mineral at each slot is unlimited, Mr. Phidang can move back to the previous slots to mine.

Mr. Phidang is very greedy so he always wants to **maximize his preference of the minerals that he can earns at each step.** This also means that Mr. Phidang wants to choose the string that has the smallest alphabetical order.

Please help Mr. Phidang find the best starting slot and the way to achieve the best preference after a certain number of steps. For example with the gold mine as "`ebadcac`", the minerals that Mr. Phidang earns after the 5 steps is "`ababa`", which is the best way for his preference.

Input

The first line contains the string s of lowercase Latin letters – the gold mine. The number of minable slots is not less than 2 and not greater than 100.

The second line contains one integer k $(1 \leq k \leq 10^4)$ – the number of mining steps.

Output

Print the sequence of letters without spaces – the values of minerals in the order that Mr. Phidang is going to mine.

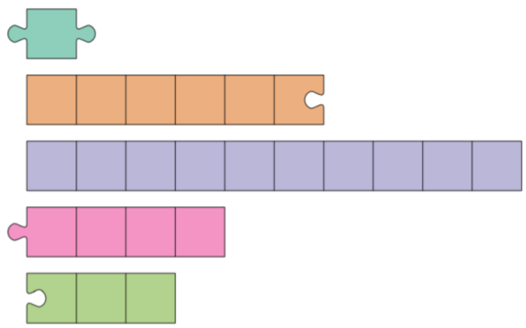
Sample Input / Sample Output

Sample Input	Sample Output
ebadcac 5	ababa
ccc 6	cccccc

Problem D
Puzzle Pieces

Time Limit: **2 seconds**
Mem limit: **512 Megabytes**

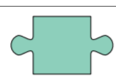
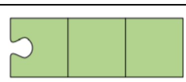
You are given n pieces of $1 \times a$ puzzle (one dimensional puzzle) as shown in the figure below.



Each piece has two heads which can be one of three types: **“in”**, **“out”**, or **“none”**.

- The “**in”** type means that the border of the head is **concave**.
- The **“out”** type describe a **convex** border.
- The “**none”** type denotes a **straight** border.

For example,

	This piece has two **“out”** heads, or two **convex** heads.
	This piece has an **“in”** head on the left and a **“none”** head on the other side.

There are a few rules for you:

1. You cannot reverse these pieces, in other words, you cannot swap left and right borders of any piece.
2. Any **“in”** head can be connected with any **“out”** head and vice versa.
3. You cannot connect pieces that have **“none”** heads.

You have to connect many pieces (possibly one), one after another, in order to achieve a single large piece of length L. Both heads of this combined piece must be **“none”** type. You wonder how many different sets of pieces that you can build up the large piece of length L, using **all** the pieces in the set. Because the number of different sets could be large, you have to calculate it modulo $10^9 + 7$.

Note: You should count the number of sets of pieces, not the number of ways of connecting them.

Input

The first line contains two integers n and L – the number of puzzle pieces and the desire length of the large piece $(1 \le n \le 300, 1 \le L \le 300)$.

The following n lines contain the description of the pieces. Each line contains an integer and two strings a_i, l_i and r_i — the length of the piece, type of its left head, and type of its right head, respectively $(1 \le a_i \le L; l_i, r_i \in \{$“`in`”, “`out`”, “`none`”$\})$. String “`in`” denotes concave border, “`out`” — convex, “`none`” — straight.

Output

Output a single integer — the number of sets of pieces, such that you can build the desired large piece using the given pieces, modulo $10^9 + 7$.

Sample Input / Sample Output

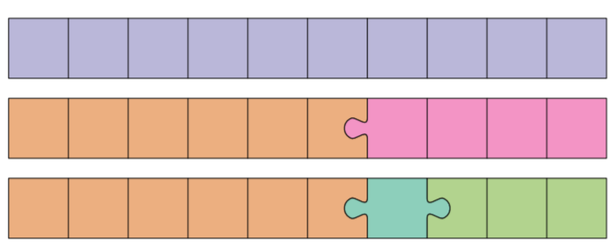
Sample Input	Sample Output

`5 10`
`1 out out`
`6 none in`
`10 none none`
`4 out none`
`3 in none`
	`3`

`4 5`
`1 none out`
`1 in out`
`2 in out`
`1 in none`
	`1`

Explain

The following figure explains the first sample test. There are three sets of pieces as illustrated in the figure.



Problem E
Martian Programmer

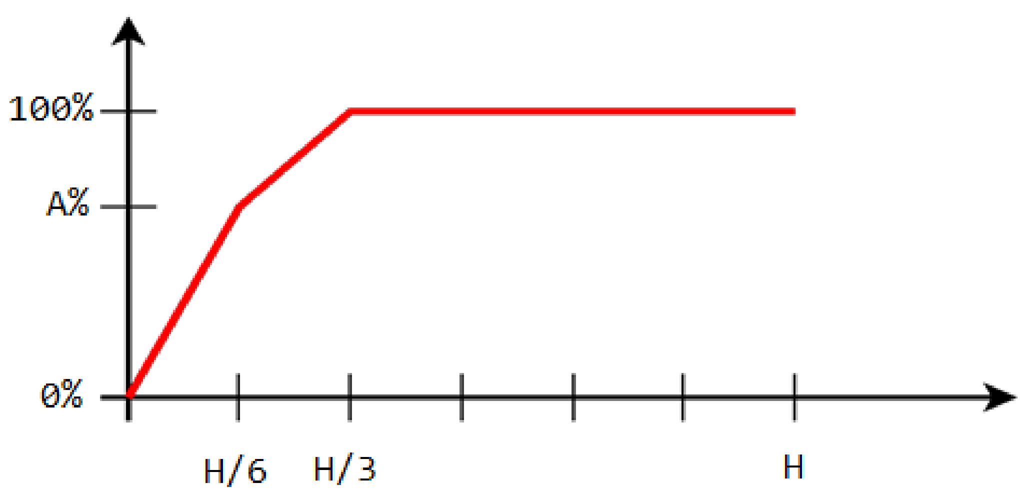
Time Limit: **1 second**

Mem limit: **512 Megabytes**

Mr. Phidang is an insane programmer coming from Mars, where there are exactly H hours a day. He has just read a scientific paper "Relationship between work and sleep". According to this research, all the time that you are not working, you are sleeping. Additionally, the research states and proves the following results:

1. If a person sleeps 0 hour per day, then his performance will be 0%
2. If a person sleeps $H/3$ hours per day, then his performance will be 100%
3. If a person sleeps $H/6$ hours per day, then his performance will be A%
4. In case a person sleeps from H/6 to H/3 hours per day, his performance increases linearly from A% to 100%
5. In case a person sleeps from H/6 to 0 hours per day, his performance decreases linearly from A% to 0%

Mr. Phidang visualizes the information by drawing the following graph.



Mr. Phidang believes that the amount of programming work he does each day is equal to the product of his working time and his performance. Thus, the problem is that if he sleeps more, he works less, and if he sleeps less, he has less performance.

Mr. Phidang wants to get back to programming as soon as possible. Please help him determine the maximum daily amount of work that he can do with the optimal choice of sleep time.

Input

The input contains two integers A and H – the performance as a percentage if the sleep time is H/6, and the number of hours per day on Mars ($0 \le A \le 100, 1 \le H \le 10^5$).

Output

The output should contains a single real number – the maximum daily amount of work that can be done by Mr. Phidang. The relative error does not exceed 10^{-6}.

Sample Input	Sample Output
75 24	1600.00000000
100 24	2000.00000000
77 123	8214.26086957

Problem F
Expected Value

Time Limit: **1 second**
Mem limit: **256 Megabytes**

Having a permutation $p = (p_1, p_2, \ldots p_N)$ of the first N positive integers, let us define:

$\boldsymbol{g_i(p)}$ is the greatest common divisor of the first i elements of p $(1 \leq i \leq N)$.

$\boldsymbol{f(p)}$ is the number of distinct integers in the array g.

For example, if $p = (2, 4, 6, 3, 1, 5)$ then

- $g_1 = GCD(2) = 2$
- $g_2 = GCD(2, 4) = 2$
- $g_3 = GCD(2, 4, 6) = 2$
- $g_4 = GCD(2, 4, 6, 3) = 1$
- $g_5 = GCD(2, 4, 6, 3, 1) = 1$
- $g_6 = GCD(2, 4, 6, 3, 1, 5) = 1$

Thus, $\boldsymbol{f(p)}$ is equal to 2.

Given an integer N, we generate a random permutation p of size N (uniformly random), your task is to calculate the expected value of $\boldsymbol{f(p)}$.

Input

The input contains only one integer $(1 \leq N \leq 200{,}000)$.

Output

You should print the expected value of $f(p)$ modulo $10^9 + 7$.

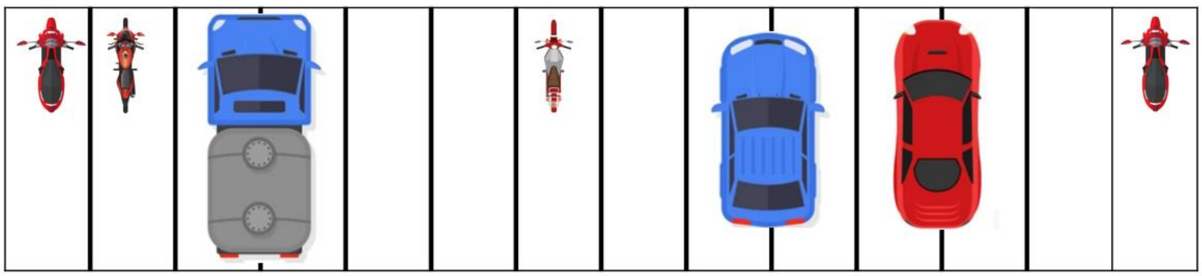
Formally, let $M = 10^9 + 7$, it can be shown that the answer can be expressed as an irreducible fraction u / v where u and v are integers and $v \mathrel{!}= 0 \ (mod\ M)$. You should output the integer equal to $u * v - 1\ (mod\ M)$. In other words, output such an integer x that $0 \leq x < M$ and $x * v = u\ (mod\ M)$.

Sample input	Sample output
2	500000005

Problem G
Parking

Time Limit: **1 second**
Mem limit: **256 Megabytes**

Today, Mr. Tipu has a thesis defense at VNU-HCM University of Science, so he has driven his car to the University campus. The University parking lot is a line of ***n* parking zone**. Each zone has one meter in width, which fits one motorcycle. A car occupies exactly two adjacent zones of the total width of two meters. The figure below illustrates the university parking lot.



Mr. Tipu wants to park his car in the parking lot, but some of the parking zones are occupied by other vehicles. Additionally, there are **a queue of *k* vehicles** trying to enter the parking lot in front of Mr. Tipu. The vehicles in the queue when entering the parking lot will occupy any empty zones: **one for a motorcycle and two adjacent zones for a single car**. In case there is no space left, all the current vehicles in the queue leave the University and not occupying any zone.

Because the thesis defense time is approaching, Mr. Tipu is in a hurry. Fortunately, the people in the queue are very nice, and they offer Mr. Tipu to enter the parking lot before them. However, Mr. Tipu does not want to take advantage of that, so he wonders how many vehicles in front of him that he would pass to enter the parking lot on time, given that he will be late definitely in case there are no parking zones left that fit his car in the parking lot.

Note that the people in the queue may choose their parking zones for their vehicles imperfectly that leave no place for Mr. Tipu's car.

Given a parking lot of n parking zones, some of which are occupied, and a queue of k vehicles (could be motorcycles or cars) apart from Mr. Tipu's car, please help Mr. Tipu to identify, for each i from 0 to k, whether Mr. Tipu can park his car or not, in case he enters the parking lot after the first i vehicles in the queue.

Input

There are several tests in each test case.

The first line contains a single integer t – the number of tests in the input ($1 \leq t \leq 50\,000$).

Each test consists of two lines.

The first of these two lines is a string of n characters ($1 \leq n \leq 10^5$), which can be '`.`' or '`X`'. The i^{th} character denotes the status of the i^{th} zone: '`.`' for empty and '`X`' for occupied.

The second of the two lines consist of k characters ($1 \leq k \leq n$), which can be '`M`' or '`C`', denoting a motorcycle or a car, respectively. The first character represents the first vehicle at the head of the queue and the last character is at the tail.

Output

The output contains t lines. Each line contains $k + 1$ characters, which can be '`Y`' or '`N`'. For each i from 0 to k print '`Y`', if Mr. Tipu can find a parking zone for his car, entering after the first i vehicles in the queue, no matter which zones they occupy, and print '`N`' otherwise.

Sample input	Sample output
1 XXXX..X.XXXX.X MM	YNN
3 ...... MMMC .X..X. MMM ...... CCCM	YYYNN YNNN YYNNN

Explanation

In the first sample, Mr. Tipu can park his car in the two empty parking zones in case he enter the parking lot first. Otherwise, the first motorcycle in the queue occupies one of the empty zones, so he will not be able to park his car.

In the test of the second sample, the first three motorcycles could park in the following way: "`M.M.M.`", then Mr. Tipu will not be able to fine a two adjacent parking zones to park his car.

Problem H
Tom and Jery

Time limit: **1 second**

Mem limit: **256 Megabytes**

Tom and Jerry are chasing each other in a maze that is formed as a tree. This tree has n nodes.

Jerry is currently at node x and there is a bomb set up by Tom here. The bomb has its range t, so when it explodes, all the nodes that have distance to node x less than or equal to $t + 1$ will be burnt. If Jerry gets burnt, he will be fainted and caught by Tom.

Of course, Jerry wants to get rid of the exploding zone by running through the edges in the maze. On the other hand, Tom wants to catch Jerry, so he plans to block a few edges on the maze so that Jerry cannot move through these edges.

Task: Given a set of m queries, each query has two numbers x and t, indicating the initial node that Jerry and the bomb are currently at, and the range of the bomb. Please help Tom to find the minimum number of edges to block in order to catch Jerry.

Input

The first line contains two integers n, m $(1 \leq n, m \leq 10^5)$ – the number of nodes on the maze and the number of queries.

In the next $n - 1$ lines, each line contains two integers u, v $(1 \leq u, v \leq n; u \neq v)$ denoting that there is an edge between u and v.

In the next m lines, each line contains two integers x, t $(1 \leq x \leq n; 0 \leq t \leq n)$ representing a single query.

Output

The minimum number of edges that Tom has to block.

Sample input

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

Sample output

```
1
0
2
```

Problem I
Inversion Number

Time limit: **1 second**
Mem limit: **256 Megabytes**

You are given an array of n integers $a_1, a_2, a_3, \ldots, a_n$, which is a permutation of n numbers from 1 to n. An inversion number of an array is the number of inversions, i.e. pairs (i, j) that satisfy the condition $1 \le i < j \le n$ and $a_i > a_j$.

You are allowed to perform a transformation on the array as follows:

- You pick an integer x that has value in range 1 to n.
- All the elements that are less than x will be moved to the left of x and all the elements that are greater than x will be moved to the right of x.
- The order of the elements that are less than x and the order of elements that are greater than x have to be the same after the transformation.

For example, given an array of [6,2,3,5,1,4] and you pick $x = 4$. After the transformation, the array will be [2,2,1,4,6,5] and it has three inversions. However, if you pick $x = 5$, then the array will be [2,3,1,4,5,6] and it has only two inversions.

Task: Find the minimum inversion number of the array after your transformation.

Input

The first line contains an integer n $(1 \le n \le 10^6)$ – the number of elements in the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ – the array itself.

Output

The minimum inversion number.

Sample input	Sample output
6 6 2 3 5 1 4	2

Problem J
The Biggest Plus Sign

Time limit: **1 second**
Mem limit: **256 Megabytes**

Dr. Memphis is creating a program that can recognize a plus sign (`'+'`) in a grayscale image in bitmap format. His program can detect a plus sign in the following forms:

- A plus sign is formed by adjacent black pixels.
- A plus sign has the size of $4 \times k + 1$, where $k \geq 0$ and its center can be at any pixel then expands to 4 directions (up, down, left, right) with the length of k.

Because the image is not very clear and contains noise, Dr. Memphis wants to identify the biggest plus sign in the image.

Task: You are given an image of size $m \times n$ $(1 \leq m, n \leq 5000)$ consisting of black and white pixels. Please find the biggest plus sign in the image.

Input

The first line contains two integer m, n – the size of the image.

Each of the next m line contains a string of n characters (`0` or `1`), `0` represents white pixel and `1` is for black.

The rows are indexed from 1 to m. The indices of the columns are from 1 to n from left to right.

Output

The first line contains the size of the biggest plus sign. If the image does not contain any plus sign, print "`-1`" (without the quotation marks).

If there is such the biggest plus sign, the second line contains two integers that indicate the row index and column index of the biggest plus sign's center. If there are multiple biggest plus sign, output the one that has the smallest row index, then the smallest column index (in case their row indices are the same).

Sample input	Sample output
8 8 00001000 00001000 00111110 00001000 00001000 00100000 01110000 00100000	9 3 5

Problem K
Squid Game

Time limit: **2 seconds**
Mem limit: **256 Megabytes**

There are n players attending the "Squid Game" version in a secret facility.

Each player is placed in a one-dimension space. There is one assassin player in the game. The players and the assassin can move at a speed of 1 meter per second, but only one player can move at a time. If the assassin and any players are at the same point at the same time, those players will be eliminated.

There is a shelter that can accommodate any number of players, but the assassin will not pass. If any player gets to the shelter and there is no assassin there, then he is saved.

You are the one that knows the map and all players' positions as well as the position of the assassin. You need to find the maximum number of players that you can be guaranteed to save. Every second, you can give a movement order to only one player.

Input

The first line contains a number n $(1 \leq n \leq 10^5)$ – the number of players.

The second line contains n integers $x_1, x_2, \dots x_n$ $(|x_i| \leq 10^9)$ – the coordinates of players.

The third line contains a single integer y $(|y| \leq 10^9)$ – the coordinate of the assassin in meters.

The last line contains a single integer z $(|z| \leq 10^9)$ – the coordinate of the shelter.

Output

Print a single number - the maximum number of players that you are guaranteed to save.

Sample input	Sample output
3 -1 1 3 2 0	1
3 0 0 9 10 1	2

Problem L
Oil Factory

Time limit: **2 seconds**
Mem limit: **256 Megabytes**

In an oil factory in southern Vietnam, there are n containers in a row. A specialized robot can do 2 types of operations.

1. In the first operation, the robot will pour out half of the oil in all containers that currently have even liters of oil.
2. In the second operation, the robot will pour out 1 liter out of all containers that currently have odd liters of oil.

Given a sequence of operations, display the total liters of oils in all containers after each time the robot finishes an operation.

Input

The first line contains an integer n - the number of containers $(1 \leq n \leq 10^5)$.

The second line contains n integers a_i separated by spaces $(1 \leq a_i \leq 2^{30} - 1)$ - the current liters of oil in each container.

The third line contains a sequence of operations in the form of characters "`0`" (first operation) and "`1`" (second operation)

The number of operations ranges from 1 to 10^5.

Output

After each command, print the sum of oil (in liters) in all containers on a separate line.

Sample input	Sample output
5 1 2 3 4 5 0110	12 8 8 4
3 1 1 1 01	3 0

Problem M
Solitaire

Time limit: **2 seconds**
Mem limit: **256 Megabytes**

Mr. Bikrone invented the following solitaire:

Let n cards be laid out in a row, each of which contains one number $a_1, a_2, \dots, a_n$.

Mr. Bikrone looks through these cards from left to right and puts some of them to the right end. The condition for moving a card is the presence to the right of it at least one card with a larger value than its own. More strictly: the card at position i with the value a_i will be moved to the right if there is a position j ($j > i$) such that $a_j > a_i$.

Solitaire is completed when Mr. Bikrone cannot move any card to the right end.

You need to help Mr. Bikrone figure out how many transfers will have to be done to complete the solitaire.

Input

The first line contains one integer n ($1 \le n \le 10^6$) – the number of cards in the solitaire.

The second line contains n space-separated integers a_i - a description of the original location cards on the table from left to right ($1 \le a_i \le 10^9$).

Output

Print one number - the number of transfers need to be done to finish the solitaire.

Sample input	Sample output
10 3 7 6 8 5 8 2 1 7 6	14

Explanation

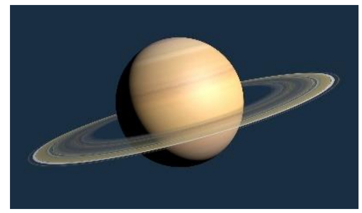
- Originally, the cards having values `3,7,6,5,2,1` are first moved to the right end, which take Mr. Bikrone **6 transfers**. After that, the new setup is `8,8,7,6,3,7,6,5,2,1.`
- Next, the cards having values `6,3` are moved to the right end, which leads to the sequence `8,8,7,7,6,5,2,1,6,3`. Mr. Bikrone needs to transfer **2 times.**
- The cards valued `5,2,1` are then transferred to the right end with **3 transfers** and leads to the sequence `8,8,7,7,6,6,3,5,2,1`.
- After that, only one card of value `3` is transferred to the right end, which takes **1 transfer** and the sequence is now `8,8,7,7,6,6,5,2,1,3`.
- Finally, the two cards `2,1` are transferred to the right end and Mr. Bikrone has the final version of the card sequence `8,8,7,7,6,6,5,3,2,1`, with the **2 last transfers**.

The total number of transfers are $6 + 2 + 3 + 1 + 2 = \mathbf{14}$ **transfers**.

Problem N
From Mars to Saturn

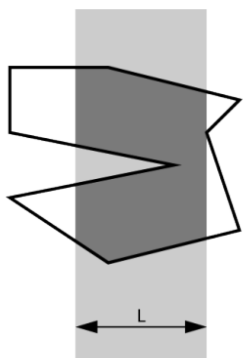
Time Limit: **2 seconds**

Mem limit: **512 Megabytes**



After reading the research paper about work and sleep, the insane Martian programmer Mr. Phidang wants to explore and looking for life on the new planet, which is the Saturn. Thus, he is going to send a spacecraft from Mars to Saturn. His spacecraft contains a huge scanner in order to scan the Saturn's surface.

Assuming that the Saturn's surface is a 2D polygon, and the spacecraft flies over it. The scanner on the spacecraft projects a laser on the surface. In particular, the spacecraft flies by a straight line parallel to the y-axis and the scanner width is L.



Mr. Phidang wants to maximize the explored area. Please help him to find the maximum area.

Input

The first line contains two integers n and L $(1 \le n, L \le 20{,}000)$ – the number of vertices of the polygon, describing the Saturn's surface, and the width of the scanner.

Each of the next n lines contains two integers x_i and y_i – coordinates of the polygon vertices. Vertices are given in counterclockwise direction. Polygon has no self-intersections and self-touches.

All coordinates are integers, not exceeding 10^4 by their absolute value.

Output

Output the maximum area that can be scanned by the scanner.

The absolute or relative error should not exceed 10^{-4}.

Sample Input **Sample output**

Sample Input	Sample output
3 3 0 0 5 1 1 7	14.28
3 40 60 10 50 50 0 0	1111.1111111111113

Problem O
Watchmen

Time Limit: **2 seconds**
Mem limit: **256 Megabytes**

Watchmen are in a danger and Doctor Manhattan together with his friend Daniel Dreiberg should warn them as soon as possible. There are n watchmen on a plane, the i^{th} watchman is located at point $(x_i;\ y_i)$.

They need to arrange a plan, but there are some difficulties on their way. As you know, Doctor Manhattan considers the distance between watchmen i and j to be $|x_i - x_j| + |y_i - y_j|$. Daniel, as an ordinary person, calculates the distance using the formula:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

The success of the operation relies on the number of pairs $(i;\ j)$ $(1 \le\ i\ <\ j \le\ n)$, such that the distance between watchman i and watchmen j calculated by Doctor Manhattan is equal to the distance between them calculated by Daniel. You were asked to compute the number of such pairs.

Input

The first line of the input contains an integer n $(1 \le\ n \le\ 200\ 000)$, the number of watchmen.

Each of the following n lines contains two integers x_i and y_i $(|x_i|, |y_i| \le 10^9)$.

Output

Print the number of pairs of watchmen such that the distance between them calculated by Doctor Manhattan is equal to the distance calculated by Daniel.

Sample Input / Sample output

Sample Input	Sample output
3 1 1 7 5 1 5	2
6 0 0 0 1 0 2 -1 1 0 1 1 1	11

Explanation

In the first sample, the distance between watchman 1 and watchman 2 is equal to $|1-7|+|1-5|=10$ for Doctor Manhattan and $\sqrt{(1-7)^2+(1-5)^2}=2\cdot\sqrt{13}$ for Daniel. For pairs $(1;\ 1),(1;\ 5)$ and $(7;\ 5),(1;\ 5)$ Doctor Manhattan and Daniel will calculate the same distances.