SWERC 2021-2022 - Online Mirror (Unrated, ICPC Rules, Teams Preferred)

A. Organizing SWERC

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Gianni, SWERC's chief judge, received a huge amount of high quality problems from the judges and now he has to choose a problem set for SWERC.

He received n problems and he assigned a beauty score and a difficulty to each of them. The i-th problem has beauty score equal to b_i and difficulty equal to d_i. The beauty and the difficulty are integers between 1 and 10.

If there are no problems with a certain difficulty (the possible difficulties are $1, 2, \ldots, 10$) then Gianni will ask for more problems to the judges.

Otherwise, for each difficulty between 1 and 10, he will put in the problem set one of the most beautiful problems with such difficulty (so the problem set will contain exactly 10 problems with distinct difficulties). You shall compute the total beauty of the problem set, that is the sum of the beauty scores of the problems chosen by Gianni.

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains the integer n ($1 \le n \le 100$) — how many problems Gianni received from the judges.

The next n lines contain two integers each. The i-th of such lines contains b_i and d_i ($1 \le b_i, d_i \le 10$) — the beauty score and the difficulty of the i-th problem.

Output

For each test case, print the total beauty of the problem set chosen by Gianni. If Gianni cannot create a problem set (because there are no problems with a certain difficulty) print the string `MOREPROBLEMS` (all letters are uppercase, there are no spaces).

Example

input
2 3 8 4 9 3 6 7 12 3 10 10 1 10 2 10 3 10 4 3 10 10 5 10 6 10 7 10 8 10 9 1 10
output
MOREPROBLEMS 93

Note

In the **first test case**, Gianni has received only 3 problems, with difficulties $3, 4, 7$ which are not sufficient to create a problem set (for example because there is not a problem with difficulty 1).

In the **second test case**, Gianni will create a problem set by taking the problems 2, 3, 4, 5, 7, 8, 9, 10, 11 (which have beauty equal to 10 and all difficulties from 1 to 9) and one of the problems 1 and 6 (which have both beauty 3 and difficulty 10). The total beauty of the resulting problem set is $10 \cdot 9 + 3 = 93$.

B. Toys

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vittorio has three favorite toys: a teddy bear, an owl, and a raccoon. Each of them has a name.

Vittorio takes several sheets of paper and writes a letter on each side of every sheet so that it is possible to spell any of the three names by arranging some of the sheets in a row (sheets can be reordered and flipped as needed). The three names do not have to be spelled at the same time, it is sufficient that it is possible to spell each of them using all the available sheets (and the same sheet can be used to spell different names).

Find the minimum number of sheets required. In addition, produce a list of sheets with minimum cardinality which can be used to spell the three names (if there are multiple answers, print any).

Input
The first line contains a string t consisting of uppercase letters of the English alphabet ($1 \leq |t| \leq 1000$) — the name of the teddy bear.

The second line contains a string o consisting of uppercase letters of the English alphabet ($1 \leq |o| \leq 1000$) — the name of the owl.

The third line contains a string r consisting of uppercase letters of the English alphabet ($1 \leq |r| \leq 1000$) — the name of the raccoon.

The values $|t|$, $|o|$, $|r|$ denote the length of the three names t, o, r.

Output
The first line of the output contains a single integer m — the minimum number of sheets required.

Then m lines follow: the j-th of these lines contains a string of two uppercase letters of the English alphabet — the letters appearing on the two sides of the j-th sheet.

Note that you can print the sheets and the two letters of each sheet in any order.

Examples

input
AA GA MA
output
2 AG AM

input
TEDDY HEDWIG RACCOON
output
8 AT CH CY DG DO ER IN OW

input
BDC CAA CE
output
4 AD AE BB CC

Note
In the **first sample**, the solution uses two sheets: the first sheet has A on one side and G on the other side; the second sheet has A on one side and M on the other side.

The name AA can be spelled using the A side of both sheets. The name GA can be spelled using the G side of the first sheet and the A side of the second sheet. Finally, the name MA can be spelled using the M side of the second sheet and the A side of the first sheet.

C. European Trip

time limit per test: 2 seconds
memory limit per test: 256 megabytes

input: standard input
output: standard output

The map of Europe can be represented by a set of n cities, numbered from 1 through n, which are connected by m bidirectional roads, each of which connects two distinct cities. A *trip* of length k is a sequence of $k+1$ cities $v_1, v_2, \ldots, v_{k+1}$ such that there is a road connecting each consecutive pair v_i, v_{i+1} of cities, for all $1 \leq i \leq k$. A *special trip* is a trip that does not use the same road twice in a row, i.e., a sequence of $k+1$ cities $v_1, v_2, \ldots, v_{k+1}$ such that it forms a trip and $v_i \neq v_{i+2}$, for all $1 \leq i \leq k-1$.

Given an integer k, compute the number of distinct special trips of length k which begin and end in the same city. Since the answer might be large, give the answer modulo $998\,244\,353$.

Input

The first line contains three integers n, m and k ($3 \leq n \leq 100$, $1 \leq m \leq n(n-1)/2$, $1 \leq k \leq 10^4$) — the number of cities, the number of roads and the length of trips to consider.

Each of the following m lines contains a pair of distinct integers a and b ($1 \leq a, b \leq n$) — each pair represents a road connecting cities a and b. It is guaranteed that the roads are distinct (i.e., each pair of cities is connected by at most one road).

Output

Print the number of special trips of length k which begin and end in the same city, modulo $998\,244\,353$.

Examples

input
4 5 2 4 1 2 3 3 1 4 3 2 4
output
0

input
4 5 3 1 3 4 2 4 1 2 1 3 4
output
12

input
8 20 12 4 3 6 7 5 7 8 2 8 3 3 1 4 7 8 5 5 4 3 5 7 1 5 1 7 8 3 2 4 2 5 2 1 4 4 8 3 6 4 6
output
35551130

Note

In the **first sample**, we are looking for special trips of length 2, but since we cannot use the same road twice once we step away from a city we cannot go back, so the answer is 0.

In the **second sample**, we have the following 12 special trips of length 3 which begin and end in the same city: $(1, 2, 4, 1)$, $(1, 3, 4, 1)$, $(1, 4, 2, 1)$, $(1, 4, 3, 1)$, $(2, 1, 4, 2)$, $(2, 4, 1, 2)$, $(3, 1, 4, 3)$, $(3, 4, 1, 3)$, $(4, 1, 3, 4)$, $(4, 3, 1, 4)$, $(4, 1, 2, 4)$, and $(4, 2, 1, 4)$.

D. Evolution of Weasels

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A wild basilisk just appeared at your doorstep. You are not entirely sure what a basilisk is and you wonder whether it evolved from your favorite animal, the weasel.

How can you find out whether basilisks evolved from weasels? Certainly, a good first step is to sequence both of their DNAs. Then you can try to check whether there is a sequence of possible mutations from the DNA of the weasel to the DNA of the basilisk.

Your friend Ron is a talented alchemist and has studied DNA sequences in many of his experiments. He has found out that DNA strings consist of the letters A, B and C and that single mutations can only remove or add substrings at any position in the string (a substring is a contiguous sequence of characters). The substrings that can be removed or added by a mutation are AA, BB, CC, ABAB or BCBC. During a sequence of mutations a DNA string may even become empty.

Ron has agreed to sequence the DNA of the weasel and the basilisk for you, but finding out whether there is a sequence of possible mutations that leads from one to the other is too difficult for him, so you have to do it on your own.

Input
Each test contains multiple test cases. The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains a string u ($1 \leq |u| \leq 200$) — the DNA of the weasel.

The second line of each test case contains a string v ($1 \leq |v| \leq 200$) — the DNA of the basilisk.

The values $|u|$, $|v|$ denote the lengths of the strings u and v. It is guaranteed that both strings u and v consist of the letters A, B and C.

Output
For each test case, print YES if there is a sequence of mutations to get from u to v and NO otherwise.

Example

input
8 A B B C C A AA BB BB CC CC AA ABAB BCBC ABC CBA
output
NO NO NO YES YES YES YES NO

E. Round Table

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n people, numbered from 1 to n, sitting at a round table. Person $i+1$ is sitting to the right of person i (with person 1 sitting to the right of person n).

You have come up with a better seating arrangement, which is given as a permutation $p_1, p_2, \ldots, p_n$. More specifically, you want to change the seats of the people so that at the end person p_{i+1} is sitting to the right of person p_i (with person p_1 sitting to the right of person p_n). Notice that for each seating arrangement there are n permutations that describe it (which can be obtained by rotations).

In order to achieve that, you can swap two people sitting at adjacent places; but there is a catch: for all $1 \leq x \leq n-1$ you cannot swap person x and person $x+1$ (notice that you **can** swap person n and person 1). What is the minimum number of swaps necessary? It can be proven that any arrangement can be achieved.

Input
Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 10\,000$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains a single integer n ($3 \le n \le 200\,000$) — the number of people sitting at the table.

The second line contains n distinct integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, $p_i \ne p_j$ for $i \ne j$) — the desired final order of the people around the table.

The sum of the values of n over all test cases does not exceed $200\,000$.

Output
For each test case, print the minimum number of swaps necessary to achieve the desired order.

Example

input
3 4 2 3 1 4 5 5 4 3 2 1 7 4 1 6 5 3 7 2
output
1 10 22

Note
In the **first test case**, we can swap person 4 and person 1 (who are adjacent) in the initial configuration and get the order $[4, 2, 3, 1]$ which is equivalent to the desired one. Hence in this case a single swap is sufficient.

F. Antennas

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n equidistant antennas on a line, numbered from 1 to n. Each antenna has a power rating, the power of the i-th antenna is p_i.

The i-th and the j-th antenna can communicate directly if and only if their distance is at most the minimum of their powers, i.e., $|i - j| \le \min(p_i, p_j)$. Sending a message directly between two such antennas takes 1 second.

What is the minimum amount of time necessary to send a message from antenna a to antenna b, possibly using other antennas as relays?

Input
Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 100\,000$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains three integers n, a, b ($1 \le a, b \le n \le 200\,000$) — the number of antennas, and the origin and target antenna.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$) — the powers of the antennas.

The sum of the values of n over all test cases does not exceed $200\,000$.

Output
For each test case, print the number of seconds needed to trasmit a message from a to b. It can be shown that under the problem constraints, it is always possible to send such a message.

Example

input
3 10 2 9 4 1 1 1 5 1 1 1 1 5 1 1 1 1 3 1 3 3 3 1
output
4 0 2

Note
In the **first test case**, we must send a message from antenna 2 to antenna 9. A sequence of communications requiring 4 seconds, which is the minimum possible amount of time, is the following:

- In 1 second we send the message from antenna 2 to antenna 1. This is possible since $|2-1| \leq \min(1,4) = \min(p_2, p_1)$.
- In 1 second we send the message from antenna 1 to antenna 5. This is possible since $|1-5| \leq \min(4,5) = \min(p_1, p_5)$.
- In 1 second we send the message from antenna 5 to antenna 10. This is possible since $|5-10| \leq \min(5,5) = \min(p_5, p_{10})$.
- In 1 second we send the message from antenna 10 to antenna 9. This is possible since $|10-9| \leq \min(5,1) = \min(p_{10}, p_9)$.

G. Gastronomic Event

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

SWERC organizers want to hold a gastronomic event.

The location of the event is a building with n rooms connected by $n-1$ corridors (each corridor connects two rooms) so that it is possible to go from any room to any other room.

In each room you have to set up the tasting of a typical Italian dish. You can choose from n typical Italian dishes rated from 1 to n depending on how good they are (n is the best possible rating). The n dishes have distinct ratings.

You want to assign the n dishes to the n rooms so that the number of pleasing tours is maximal. A *pleasing tour* is a nonempty sequence of rooms so that:

- Each room in the sequence is connected to the next one in the sequence by a corridor.
- The ratings of the dishes in the rooms (in the order given by the sequence) are increasing.

If you assign the n dishes optimally, what is the maximum number of pleasing tours?

Input
The first line contains an integer n ($2 \leq n \leq 1\,000\,000$) — the number of rooms.

The second line contains $n-1$ integers $p_2, p_3, \cdots, p_n$ ($1 \leq p_i < i$). Each p_i indicates that there is a corridor between room i and room p_i. It is guaranteed that the building has the property that it is possible to go from any room to any other room.

Output
Print the maximum number of pleasing tours.

Examples

input
5 1 2 2 2
output
13

input
10 1 2 3 4 3 2 7 8 7
output
47

Note
In the **first sample**, it is optimal to place the dish with rating 1 in room 1, the dish with rating 2 in room 3, the dish with rating 3 in room 2, the dish with rating 4 in room 5 and the dish with rating 5 in room 4.

All the 13 possible pleasing tours are: (1), (2), (3), (4), (5), $(1,2)$, $(3,2)$, $(2,4)$, $(2,5)$, $(1,2,4)$, $(1,2,5)$, $(3,2,4)$, $(3,2,5)$.

There are also other ways to assign the dishes to the rooms so that there are 13 pleasing tours.

H. Boundary

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bethany would like to tile her bathroom. The bathroom has width w centimeters and length l centimeters. If Bethany simply used the basic tiles of size 1×1 centimeters, she would use $w \cdot l$ of them.

However, she has something different in mind.

- On the interior of the floor she wants to use the 1×1 tiles. She needs exactly $(w-2) \cdot (l-2)$ of these.

- On the floor boundary she wants to use tiles of size $1 \times a$ for some positive integer a. The tiles can also be rotated by 90 degrees.

For which values of a can Bethany tile the bathroom floor as described? Note that a can also be 1.

Input
Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The descriptions of the t test cases follow.

Each test case consist of a single line, which contains two integers w, l ($3 \le w, l \le 10^9$) — the dimensions of the bathroom.

Output
For each test case, print an integer k ($0 \le k$) — the number of valid values of a for the given test case — followed by k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i$) — the valid values of a. The values $a_1, a_2, \ldots, a_k$ have to be sorted from smallest to largest.

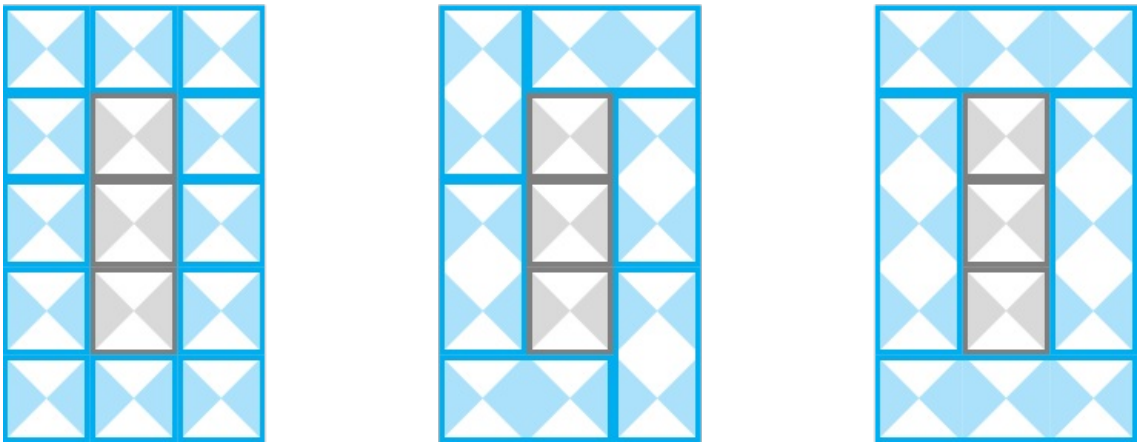
It is guaranteed that under the problem constraints, the output contains at most $200\,000$ integers.

Example

input
3 3 5 12 12 314159265 358979323
output
3 1 2 3 3 1 2 11 2 1 2

Note
In the **first test case**, the bathroom is 3 centimeters wide and 5 centimeters long. There are three values of a such that Bethany can tile the floor as described in the statement, namely $a = 1$, $a = 2$ and $a = 3$. The three tilings are represented in the following pictures.

$a = 1$ $a = 2$ $a = 3$



I. Ice Cream Shop

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

On a beach there are n huts in a perfect line, hut 1 being at the left and hut $i + 1$ being 100 meters to the right of hut i, for all $1 \le i \le n - 1$. In hut i there are p_i people.

There are m ice cream sellers, also aligned in a perfect line with all the huts. The i-th ice cream seller has their shop x_i meters to the right of the first hut. All ice cream shops are at distinct locations, but they may be at the same location as a hut.

You want to open a new ice cream shop and you wonder what the best location for your shop is. You can place your ice cream shop anywhere on the beach (not necessarily at an integer distance from the first hut) as long as it is aligned with the huts and the other ice cream shops, even if there is already another ice cream shop or a hut at that location. You know that people would come to your shop only if it is strictly closer to their hut than any other ice cream shop.

If every person living in the huts wants to buy exactly one ice cream, what is the maximum number of ice creams that you can sell if you place the shop optimally?

Input
The first line contains two integers n and m ($2 \le n \le 200\,000$, $1 \le m \le 200\,000$) — the number of huts and the number of ice cream sellers.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^9$) — the number of people in each hut.

The third line contains m integers $x_1, x_2, \ldots, x_m$ ($0 \le x_i \le 10^9$, $x_i \ne x_j$ for $i \ne j$) — the location of each ice cream shop.

Output
Print the maximum number of ice creams that can be sold by choosing optimally the location of the new shop.

Examples

input
3 1 2 5 6 169
output
7

input
4 2 1 2 7 8 35 157
output
15

input
4 1 272203905 348354708 848256926 939404176 20
output
2136015810

input
3 2 1 1 1 300 99
output
2

Note
In the **first sample**, you can place the shop (coloured orange in the picture below) 150 meters to the right of the first hut (for example) so that it is the closest shop to the first two huts, which have 2 and 5 people, for a total of 7 sold ice creams.

In the **second sample**, you can place the shop 170 meters to the right of the first hut (for example) so that it is the closest shop to the last two huts, which have 7 and 8 people, for a total of 15 sold ice creams.

J. Training Camp

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are organizing a training camp to teach algorithms to young kids. There are n^2 kids, organized in an n by n grid. Each kid is between 1 and n years old (inclusive) and any two kids who are in the same row or in the same column have different ages.

You want to select exactly n kids for a programming competition, with exactly one kid from each row and one kid from each column. Moreover, kids who are not selected must be either older than both kids selected in their row and column, or younger than both kids

selected in their row and column (otherwise they will complain). Notice that it is always possible to select n kids satisfying these requirements (for example by selecting n kids who have the same age).

During the training camp, you observed that some kids are good at programming, and the others are not. What is the maximum number of kids good at programming that you can select while satisfying all the requirements?

Input
The first line contains n ($1 \le n \le 128$) — the size of the grid.

The following n lines describe the ages of the kids. Specifically, the i-th line contains n integers $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($1 \le a_{i,j} \le n$) — where $a_{i,j}$ is the age of the kid in the i-th row and j-th column. It is guaranteed that two kids on the same row or column have different ages, i.e., $a_{i,j} \neq a_{i,j'}$ for any $1 \le i \le n$, $1 \le j < j' \le n$, and $a_{i,j} \neq a_{i',j}$ for any $1 \le i < i' \le n$, $1 \le j \le n$.

The following n lines describe the programming skills of the kids. Specifically, the i-th line contains n integers $c_{i,1}, c_{i,2}, \ldots, c_{i,n}$ ($c_{i,j} \in \{0, 1\}$) — where $c_{i,j} = 1$ if the kid in the i-th row and j-th column is good at programming and $c_{i,j} = 0$ otherwise.

Output
Print the maximum number of kids good at programming that you can select while satisfying all the requirements.

Examples

input
3 1 2 3 3 1 2 2 3 1 1 0 0 0 0 1 0 0 0
output
1

input
4 1 2 3 4 2 1 4 3 3 4 1 2 4 3 2 1 1 1 1 0 0 0 1 0 1 1 0 1 0 0 0 1
output
2

Note
In the **first sample**, it is not possible to select the two kids good at programming (in row 1 and column 1, and in row 2 and column 3), because then you would have to select the kid in row 3 and column 2, and in that case two kids would complain (the one in row 1 and column 2, and the one in row 3 and column 1).

A valid selection which contains 1 kid good at programming is achieved by choosing the 3 kids who are 1 year old.

In the **second sample**, there are 10 valid choices of the n kids that satisfy the requirements, and each of them selects exactly 2 kids good at programming.

K. Pandemic Restrictions

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After a long time living abroad, you have decided to move back to Italy and have to find a place to live, but things are not so easy due to the ongoing global pandemic.

Your three friends Fabio, Flavio and Francesco live at the points with coordinates (x_1, y_1), (x_2, y_2) and (x_3, y_3), respectively. Due to the mobility restrictions in response to the pandemic, meetings are limited to 3 persons, so you will only be able to meet 2 of your friends at a time. Moreover, in order to contain the spread of the infection, the authorities have imposed the following additional measure: for each meeting, the sum of the lengths travelled by each of the attendees from their residence place to the place of the meeting must not exceed r.

What is the minimum value of r (which can be any nonnegative real number) for which there exists a place of residence that allows you to hold the three possible meetings involving you and two of your friends? Note that the chosen place of residence need not have integer coordinates.

Input
The first line contains the two integers x_1, y_1 ($-10^4 \le x_1, y_1 \le 10^4$) — the coordinates of the house of your friend Fabio.

The second line contains the two integers x_2, y_2 ($-10^4 \le x_2, y_2 \le 10^4$) — the coordinates of the house of your friend Flavio.

The third line contains the two integers x_3, y_3 ($-10^4 \le x_3, y_3 \le 10^4$) — the coordinates of the house of your friend Francesco.

It is guaranteed that your three friends live in different places (i.e., the three points (x_1, y_1), (x_2, y_2), (x_3, y_3) are guaranteed to be distinct).

Output

Print the minimum value of r which allows you to find a residence place satisfying the above conditions. Your answer is considered correct if its absolute or relative error does not exceed 10^{-4}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-4}$.

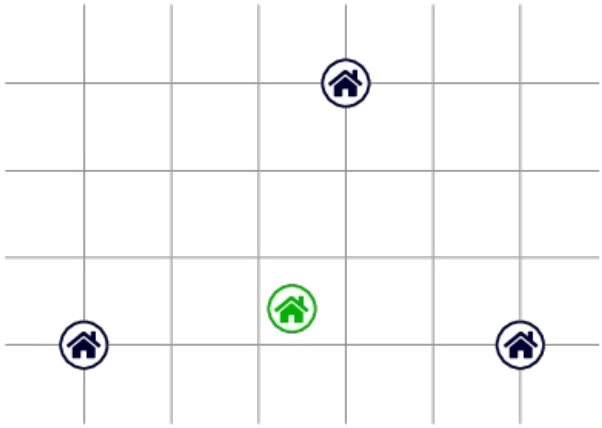
Examples

input
0 0 5 0 3 3
output
5.0686143166

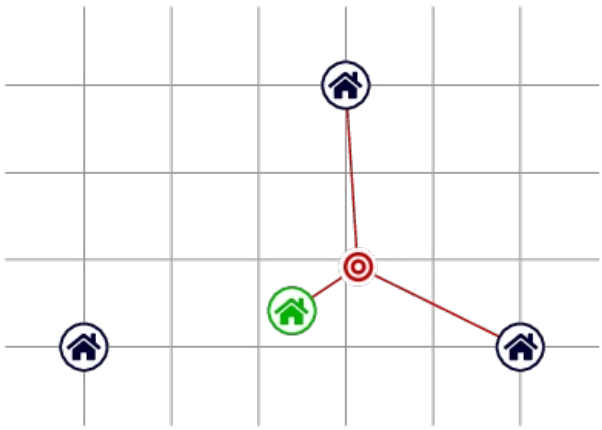
input
-1 0 0 0 1 0
output
2.0000000000

Note

In the **first sample**, Fabio, Flavio and Francesco live at the points with coordinates $(0, 0)$, $(5, 0)$ and $(3, 3)$ respectively. The optimal place of residence, represented by a green house in the picture below, is at the point with coordinates $(2.3842..., 0.4151...)$.



For instance, it is possible for you to meet Flavio and Francesco at the point depicted below, so that the sum of the lengths travelled by the three attendees is at most (and in fact equal to) $r = 5.0686...$.



In the **second sample**, any point on the segment $\{(x, 0) : -1 \le x \le 1\}$ is an optimal place of residence.

L. Il Derby della Madonnina

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input

output: standard output

The derby between Milan and Inter is happening soon, and you have been chosen as the assistant referee for the match, also known as linesman. Your task is to move along the touch-line, namely the side of the field, always looking very carefully at the match to check for offside positions and other offences.

Football is an extremely serious matter in Italy, and thus it is fundamental that you keep very close track of the ball for as much time as possible. This means that you want to maximise the number of kicks which you monitor closely. You are able to monitor closely a kick if, when it happens, you are in the position along the touch-line with minimum distance from the place where the kick happens.

Fortunately, expert analysts have been able to accurately predict all the kicks which will occur during the game. That is, you have been given two lists of integers, $t_1, \ldots, t_n$ and $a_1, \ldots, a_n$, indicating that t_i seconds after the beginning of the match the ball will be kicked and you can monitor closely such kick if you are at the position a_i along the touch-line.

At the beginning of the game you start at position 0 and the maximum speed at which you can walk along the touch-line is v units per second (i.e., you can change your position by at most v each second). What is the maximum number of kicks that you can monitor closely?

Input

The first line contains two integers n and v ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq v \leq 10^6$) — the number of kicks that will take place and your maximum speed.

The second line contains n integers $t_1, \ldots, t_n$ ($1 \leq t_i \leq 10^9$) — the times of the kicks in the match. The sequence of times is guaranteed to be strictly increasing, i.e., $t_1 < t_2 < \cdots < t_n$.

The third line contains n integers $a_1, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$) — the positions along the touch-line where you have to be to monitor closely each kick.

Output

Print the maximum number of kicks that you can monitor closely.

Examples

input
3 2 5 10 15 7 17 29
output
2

input
5 1 5 7 8 11 13 3 3 -2 -2 4
output
3

input
1 2 3 7
output
0

Note

In the **first sample**, it is possible to move to the right at maximum speed for the first 3.5 seconds and stay at position 7 until the first kick happens, and then immediately move right also at maximum speed to watch the second kick at position 17. There is no way to monitor closely the third kick after the second kick, so at most 2 kicks can be seen.

M. Bottle Arrangements

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Gabriella has been instructed to organize a renowned wine tasting event which will be attended by m critics. On display, there will be n different varieties of wine, each of which can either be a red wine or a white wine.

The wines will come in n bottles arranged in a line on the table, and, for convenience, each critic will sip from a contiguous interval of bottles: that is, he/she will taste exactly the bottles at position a, $a + 1$, $\ldots$, b for some $1 \leq a \leq b \leq n$. The interval depends on the critic, who will select it on the spot according to their preferences. In fact, the i-th critic ($1 \leq i \leq m$) has requested that he/she wants to taste exactly r_i red wines and w_i white wines.

Gabriella has yet to choose how many bottles of red wine and white wine there will be, and in what order they will appear. Help her find an arrangement (that is, a sequence of n bottles of either red or white wine) that satisfies the requests of all the critics, or state that no such arrangement exists.

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains two integers n and m ($1 \le n \le 100$, $1 \le m \le 100$) — the number of bottles of wine and the number of critics.

Each of the next m lines contains two integers r_i and w_i ($0 \le r_i, w_i \le 100$, $r_i + w_i \ge 1$) — the number of red and white wines that the i-th critic wants to taste.

Output

For each test case, if at least one solution exists, print a string of length n made up of the characters `R` and `W`, where the j-th character ($1 \le j \le n$) denotes the type of the wine in the j-th bottle of the arrangement (`R` for *red* and `W` for *white*). If there are multiple solutions, print any.

If no solution exists, print the string `IMPOSSIBLE`.

Example

input
3 5 3 1 0 3 2 2 2 4 3 2 1 1 1 0 3 3 2 0 2 0 3
output
RWRRW IMPOSSIBLE WWW

Note

In the **first test case**, there are $n = 5$ bottles of wine to be arranged and $m = 3$ critics. The arrangement `RWRRW` satisfies the requests of all three critics. Indeed:

- the first critic can choose the interval $[3, 3]$, which contains exactly one bottle of red wine (note that $[1, 1]$ and $[4, 4]$ are other valid choices);
- the second critic can choose the interval $[1, 5]$, which contains 3 bottles of red wine and 2 bottles of white wine;
- the third critic can choose the interval $[2, 5]$, which contains 2 bottles of red wine and 2 bottles of white wine.

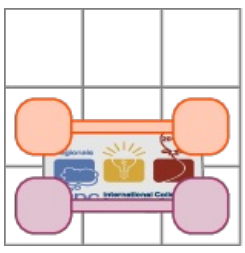
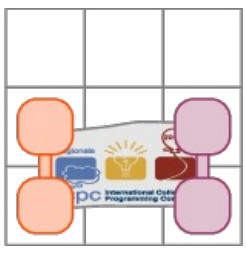
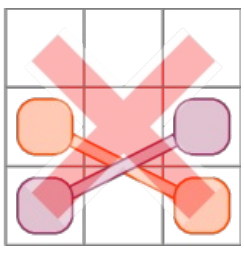
N. Drone Photo

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Today, like every year at SWERC, the n^2 contestants have gathered outside the venue to take a drone photo. Jennifer, the social media manager for the event, has arranged them into an $n \times n$ square. Being very good at her job, she knows that the contestant standing on the intersection of the i-th row with the j-th column is $a_{i,j}$ years old. Coincidentally, she notices that no two contestants have the same age, and that everyone is between 1 and n^2 years old.

Jennifer is planning to have some contestants hold a banner with the ICPC logo parallel to the ground, so that it is clearly visible in the aerial picture. Here are the steps that she is going to follow in order to take the perfect SWERC drone photo.

- First of all, Jennifer is going to select four contestants standing on the vertices of an axis-aligned rectangle.
- Then, she will have the two younger contestants hold one of the poles, while the two older contestants will hold the other pole.
- Finally, she will unfold the banner, using the poles to support its two ends. Obviously, this can only be done if the two poles are parallel and **do not cross**, as shown in the pictures below.

Being very indecisive, Jennifer would like to try out all possible arrangements for the banner, but she is worried that this may cause the contestants to be late for the competition. How many different ways are there to choose the four contestants holding the poles in order to take a perfect photo? Two choices are considered different if at least one contestant is included in one but not the other.

Input

The first line contains a single integer n ($2 \leq n \leq 1500$).

The next n lines describe the ages of the contestants. Specifically, the i-th line contains the integers $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($1 \leq a_{i,j} \leq n^2$).

It is guaranteed that $a_{i,j} \neq a_{k,l}$ if $i \neq k$ or $j \neq l$.

Output

Print the number of ways for Jennifer to choose the four contestants holding the poles.

Examples

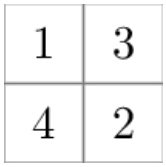
input
2 1 3 4 2
output
0

input
2 3 2 4 1
output
1

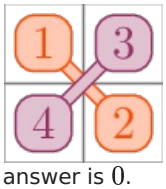
input
3 9 2 4 1 5 3 7 8 6
output
6

Note

In the **first sample**, there are 4 contestants, arranged as follows.

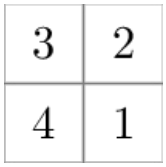


There is only one way to choose four contestants, with one pole held by contestants aged 1 and 2 and the other one by contestants aged 3 and 4. But then, as we can see in the picture, the poles cross.



Since there is no valid way to choose four contestants, the answer is 0.

In the **second sample**, the 4 contestants are arranged as follows.



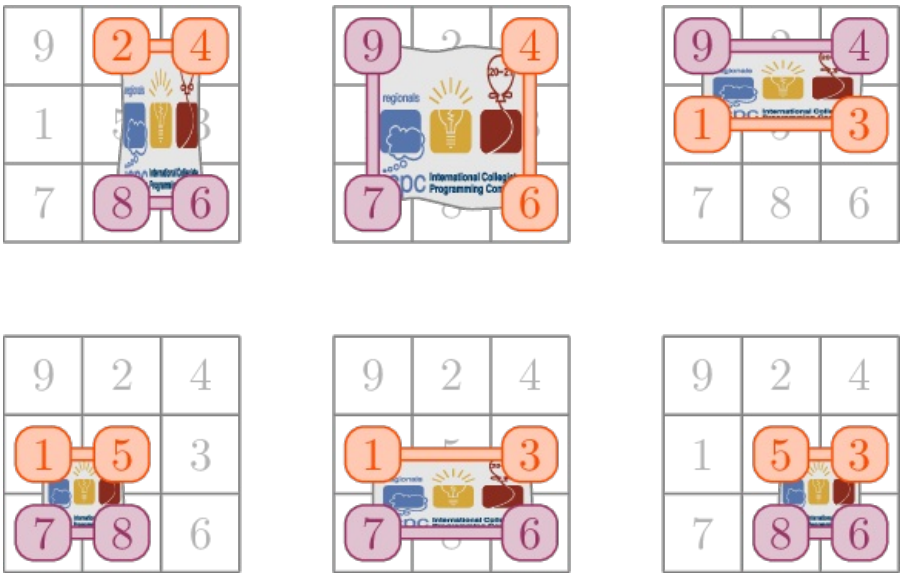
Once again, there is only one way to choose four contestants, but this time the poles don't cross.

Therefore, the answer is 1.

In the **third sample**, the 9 contestants are arranged as follows.

9	2	4
1	5	3
7	8	6

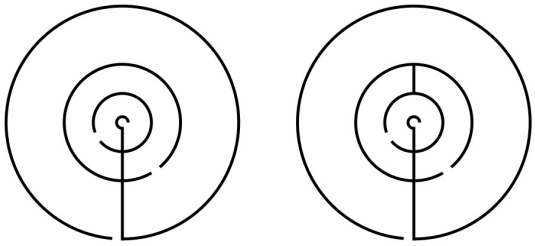
There are 6 ways of choosing four contestants so that the poles don't cross, as shown in the following pictures.



O. Circular Maze

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a circular maze such as the ones shown in the figures.



Determine if it can be solved, i.e., if there is a path which goes from the center to the outside of the maze which does not touch any wall. The maze is described by n walls. Each wall can be either circular or straight.

- Circular walls are described by a radius r, the distance from the center, and two angles θ_1, θ_2 describing the beginning and the end of the wall in the clockwise direction. Notice that swapping the two angles changes the wall.
- Straight walls are described by an angle θ, the direction of the wall, and two radii $r_1 < r_2$ describing the beginning and the end of the wall.

Angles are measured in degrees; the angle 0 corresponds to the upward pointing direction; and angles increase clockwise (hence the east direction corresponds to the angle 90).

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \le t \le 20$) — the number of test cases. The descriptions of the t test cases follow.

The first line of each test case contains an integer n ($1 \le n \le 5000$) — the number of walls.

Each of the following n lines each contains a character (C for circular, and S for straight) and three integers:

- either r, θ_1, θ_2 ($1 \le r \le 20$ and $0 \le \theta_1, \theta_2 < 360$ with $\theta_1 \ne \theta_2$) if the wall is circular,
- or r_1, r_2 and θ ($1 \le r_1 < r_2 \le 20$ and $0 \le \theta < 360$) if the wall is straight.

It is guaranteed that circular walls do not overlap (but two circular walls may intersect at one or two points), and that straight walls do not overlap (but two straight walls may intersect at one point). However, circular and straight walls can intersect arbitrarily.

Output

For each test case, print YES if the maze can be solved and NO otherwise.

Example

input
2 5 C 1 180 90 C 5 250 230 C 10 150 140 C 20 185 180 S 1 20 180 6 C 1 180 90 C 5 250 230 C 10 150 140 C 20 185 180 S 1 20 180 S 5 10 0
output
YES NO

Note

The two sample test cases correspond to the two mazes in the picture.