International Collegiate Programming Contest

The 2025 ICPC

Asia Yokohama Regional Contest

Problem Set

Problem A

Tatami Renovation

Time Limit: 2 seconds

The city's art museum is known for its long straight corridor and the artistically decorated tiles laid along it. The corridor is a rectangular area 2 meters wide and is divided into one-meter-square cells. Each cell is either occupied by a single tile or is empty.

As a part of the renovation of the museum, all the empty cells will be covered with tatami mats, which are traditional Japanese floor mats. Each tatami mat is 1 meter by 2 meters in size, so each tatami mat covers exactly two adjacent cells. Tatami mats must not overlap each other, and they must not overlap any tile.

Depending on the initial placement of the tiles, it may be impossible to cover all the empty cells with tatami mats. To address this, the museum allows each tile to be moved from its original position to one of its adjacent cells that shares a side, but not farther. Tiles must not be moved out of the corridor. No more than one tile should be on a single cell after the moves.

Figure A.1. Before and after the renovation for Sample Input 1

The left figure shows the initial positions of the tiles for Sample Input 1, and the right figure shows one possible way to move one tile and arrange tatami mats to cover all the empty cells.

Determine whether it is possible to cover all the empty cells with tatami mats when you move some (possibly none) of the tiles appropriately. If it is possible, find the minimum number of tiles to be moved.

Input

The input consists of a single test case of the following format.

n l
r_1 c_1
r_2 c_2
$\vdots$
r_n c_n

The integer n ($1 \le n \le 10^5$) represents the number of the tiles in the corridor. The integer l ($3 \le l \le 10^{18}$) represents the length of the corridor in meters. Each pair of integers r_i and c_i ($i = 1, \ldots, n$) satisfies $1 \le r_i \le 2$ and $1 \le c_i \le l$, and indicates the location of the i-th tile. Specifically, if the corridor is viewed as a rectangle with its height of 2 cells and width of l cells, then the i-th tile is located at the r_i-th row from the top and the c_i-th column from the left. It is guaranteed that all tiles are initially on distinct cells.

Output

If you can cover all the empty cells in the corridor with zero or more tatami mats by moving some (possibly none) of the tiles appropriately, output the minimum number of tiles to be moved. Otherwise, output `no`.

Sample Input 1

```
4 5
2 3
1 4
1 5
2 1
```

Sample Output 1

```
1
```

Sample Input 2

```
1 3
1 1
```

Sample Output 2

```
no
```

Sample Input 3

```
6 1000000000000
1 2
1 3
1 4
2 1
2 2
2 3
```

Sample Output 3

```
3
```

Problem B
Minimizing Wildlife Damage
Time Limit: 3 seconds

The farmland you are cultivating consists of a certain number of plots arranged from west to east. Currently, each plot contains a certain amount of wheat; the amounts may differ across plots. All the wheat will be ready for harvest after a certain number of days.

A wild boar
www.irasutoya.com

One big problem you are facing is that a hungry wild boar comes from the west every night. If none of the plots have any wheat remaining, the boar just turns back. Otherwise, the boar goes to the westernmost plot that still has wheat and eats one unit of wheat there. The boar then continues by moving to the adjacent plot to the east and eating one unit of wheat, until it either encounters a plot with no wheat remaining or finishes eating at the easternmost plot, at which point it returns home.

To mitigate the damage, you plan to choose some plots (possibly none) and remove all the wheat from these plots today, so that the boar may return without eating too much wheat on subsequent days. After that, the boar continues to come every night, but there is nothing you can do to mitigate the damage further.

You are given one or more candidate days on which you may harvest. For each candidate harvest day, determine the maximum possible amount of wheat remaining for harvest on that day, assuming you remove all the wheat from optimally chosen plots. The optimal choice of plots may vary across candidate days.

Input

The input consists of a single test case of the following format.

$$
\begin{array}{l}
n\ m \\
a_1\ \cdots\ a_n \\
d_1 \\
\vdots \\
d_m
\end{array}
$$

The integer n is the number of plots ($2 \le n \le 2 \times 10^5$). Plots are numbered from 1 to n, from west to east. The integer m is the number of candidate harvest days ($1 \le m \le 2 \times 10^5$). For each $i = 1, \ldots, n$, the integer a_i is the number of units of wheat in plot i ($0 \le a_i \le 10^{12}$). For each $j = 1, \ldots, m$, the integer d_j is the number of days until the j-th candidate harvest day ($1 \le d_j \le 2 \times 10^{17}$), that is, the boar comes d_j times before that day.

Output

Output m lines. The j-th line should contain an integer representing the maximum possible number of units of wheat remaining on the j-th candidate harvest day.

Sample Input 1	Sample Output 1
3 4 3 1 4 1 2 3 7	6 5 4 0

Sample Input 2	Sample Output 2
6 3 300 200 100 100 200 300 10 50 340	1140 1000 560

For Sample Input 1, if you do not remove any wheat, the amounts of wheat in the plots change as follows.

$$(3, 1, 4) \to (2, 0, 3) \to (1, 0, 3) \to (0, 0, 3) \to (0, 0, 2) \to (0, 0, 1) \to (0, 0, 0)$$

Instead, if you remove the wheat from plot 2, the amounts change as follows.

$$(3, 0, 4) \to (2, 0, 4) \to (1, 0, 4) \to (0, 0, 4) \to (0, 0, 3) \to (0, 0, 2) \to (0, 0, 1) \to (0, 0, 0)$$

This choice is optimal for all given candidate days.

For Sample Input 2, the optimal choices are as follows.

- For the first candidate day, removing nothing is optimal. The remaining amounts will be $(290, 190, 90, 90, 190, 290)$.
- For the second candidate day, removing the wheat from plot 3 is optimal. The remaining amounts will be $(250, 150, 0, 100, 200, 300)$.
- For the third candidate day, removing the wheat from plots 2 and 4 is optimal. The remaining amounts will be $(0, 0, 60, 0, 200, 300)$.

Problem C
Seagull Population
Time Limit: 2 seconds

An island on an extrasolar planet is famous as a good bird-watching spot, where you can see many seagull-lookalikes (simply called seagulls hereafter) throughout a year. The planet is quite similar to the Earth, but the number of days in a year is different.

Each seagull comes to the island exactly once a year, stays for a while, and leaves exactly once a year as well. Each seagull has its own schedule of coming and leaving the island, and quite punctually sticks to the schedule. That is, every year, it comes to the island on the same day of the year. Also, every year, it leaves on the same day of the year. Seagulls come to the island early in the morning and leave late in the afternoon. Seagulls that have come to the island may leave on the same day. On the other hand, seagulls may leave the island on one day and come again on the next day.

Members of the bird-watching club count the number of seagulls staying on the island every day around noon. Their counting is perfect, so that all seagulls present at that time are counted without any omission or duplication. However, the seagulls look so similar that identifying individuals is not possible.

Note that seagulls that leave the island on one evening and come again on the next morning are counted on all days in a year.

Given the daily counts of seagulls throughout a year, you want to know the total number of seagulls visiting the island. Since seagulls are indistinguishable, it is not possible to know the exact number. For example, if the counts are one on two consecutive days, the number of seagulls may be one or two. The minimum possible number is the only valuable information you can obtain.

Determine the minimum possible number of individual seagulls counted at least once in a year. If this minimum is small enough, also show a possible list of their stay periods that attains this minimum.

Input

The input consists of a single test case of the following format.

$$n$$
$$b_1\ b_2\ \cdots\ b_n$$

The integer n ($2 \le n \le 2 \times 10^5$) is the number of days in one year on that planet. Days are numbered from 1 to n throughout a year. The integer b_i ($0 \le b_i \le 2 \times 10^5$) is the number of seagulls counted on day i. At least one of b_i's is non-zero.

Output

Output the minimum possible number m of seagulls in the first line. If m is not greater than 2×10^5, then output m additional lines describing one possible list of their stay periods. The j-th of these m lines should contain two integers s_j and t_j ($1 \le s_j \le n$, $1 \le t_j \le n$) separated by a space, representing that the j-th seagull comes to the island on day s_j and leaves on day t_j. Note that s_j may be greater than t_j. In this case, the seagull stays on the island across years, from the last day of a year to the first day of the following year. When there are two or more possibilities, any of them is acceptable.

Sample Input 1	Sample Output 1
7 1 0 1 2 2 0 1	3 3 5 4 5 7 1

Sample Input 2	Sample Output 2
2 1 1	1 1 2

Sample Input 3	Sample Output 3
6 1 2 1 2 2 1	2 2 5 4 2

Sample Input 4	Sample Output 4
4 200000 0 200000 0	400000

The following figure depicts the visiting schedules of three seagulls in Sample Output 1. Note that the third seagull stays in the island across years.

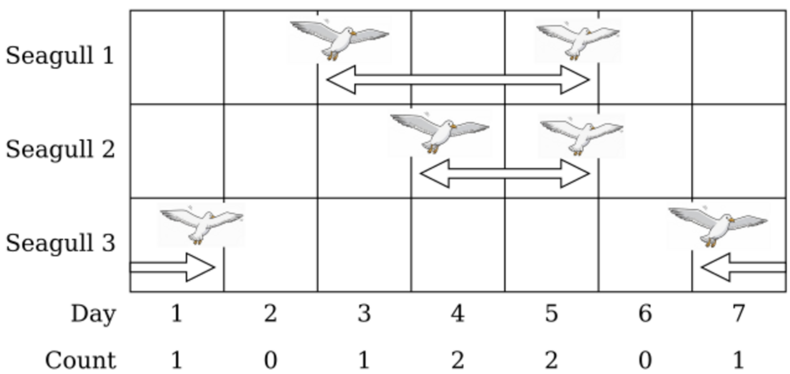


Figure C.1. Visiting schedules of the seagulls of Sample Output 1

Problem D

Decompose and Concatenate

Time Limit: 2 seconds

You are given an integer greater than or equal to 2. When this number is decomposed into the sum of two *positive* integers, a new integer number can be formed as the concatenation of the decimal representations of the two integers. Find the maximum possible number formed in this way.

For example, 102 can be decomposed and concatenated as follows.

$$\begin{array}{lcr} 1+101 & \to & 1101 \\ 2+100 & \to & 2100 \\ 3+99 & \to & 399 \\ 4+98 & \to & 498 \\ & \vdots & \\ 101+1 & \to & 1011 \end{array}$$

Among them, $92 + 10 \to 9210$ is the largest.

Input

The input consists of a single test case in a single line. The line contains an integer between 2 and 10^{17}, inclusive, which is the integer to be decomposed and concatenated.

Output

Output the maximum possible number in a line.

Sample Input 1	Sample Output 1
8	71

Sample Input 2	Sample Output 2
2025	10251000

Sample Input 3	Sample Output 3
102	9210

Sample Input 4	Sample Output 4
99999999999999999	8999999999999999910000000000000000

Problem E
Cutting Tofu
Time Limit: 2 seconds

You are preparing miso soup, a favorite among Japanese people, with one of the most popular ingredients, *tofu*. Tofu is a white cuboid-shaped food that is usually cut into smaller pieces and then put into the soup.

You plan to chop a single block of tofu to make at least a required number of cubes of the same size. You cut the tofu along planes parallel to the faces of the tofu block. Each cut goes all the way through the block, dividing all the pieces it passes through. You should not move the tofu block nor its fragments until tofu cubes are completely cut out.

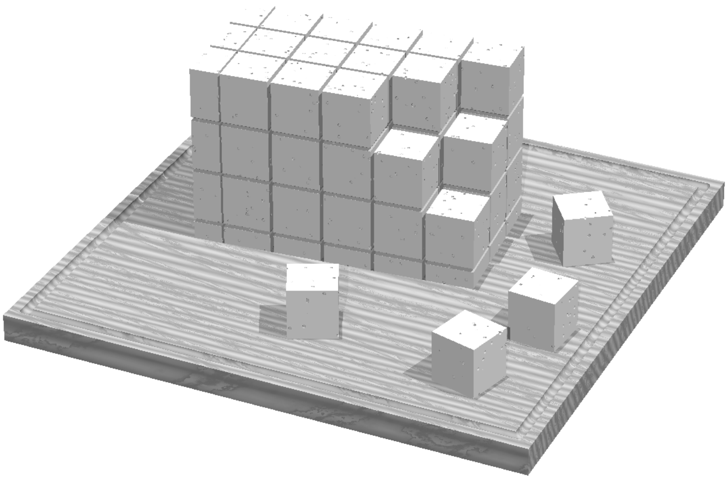


Figure E.1. An example of cutting the tofu

As long as the required number of tofu cubes can be obtained, you want to make cubes as large as possible. You don't mind leaving excess tofu cubes or leftover fragments, as they can be used for other dishes.

Given the dimensions of the block of tofu (its length, width, and height) in integer multiples of unit length, and the required number of tofu cubes, your task is to find the maximum possible side length of the tofu cubes. Since it can be proven that the answer is a rational number under the given constraints, it should be represented as a reduced fraction.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 10^4$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

> a b c
> k

The three integers a, b, and c represent the length, width, and height of the block of tofu, respectively. They are between 1 and 10^9, inclusive. The integer k ($1 \le k \le 10^9$) represents the required number of tofu cubes.

Output

For each test case, output two positive integers p and q in a line, separated by a single space. Here, p and q are mutually prime integers, meaning that p/q is the maximum possible side length of the tofu cubes.

Sample Input 1	Sample Output 1
3 1 1 1 2 2 2 3 7 2 3 5 240	1 2 1 1 1 2

Sample Input 2	Sample Output 2
3 1000000000 999999998 999999999 1000000000 1 1 999999999 1000000000 314 1000000000 1000000000 271828	499999999 500 999999999 1000000000 314 1

Problem F

Astral Geometry

Time Limit: 2 seconds

You, a young astronomer, are interested in the spatial arrangement of a set of stars with distinctive features. Knowing this might contribute to the understanding of early-universe cosmology. You can perform measurements using a specialized instrument for this purpose.

The instrument uses its own three-dimensional Cartesian coordinate system, in which the origin $(0, 0, 0)$ is set on Earth, and the positions of the stars are modeled as lattice points (points whose coordinates are all integers). You already know the distances to all the stars of interest from Earth, but their directions are unknown.

In a single measurement, you specify two distinct stars, and the instrument reports the distance between them. Note that the instrument does not report the absolute or relative directions of the stars.

Determine the distances between all pairs of the stars within the limited number of measurements.

Interaction

The first line of input contains an integer n, the number of stars of interest ($2 \le n \le 100$). The stars are numbered from 1 to n. The second line contains n integers. The i-th of them is the **squared** distance from the origin to star i. It is guaranteed that all stars have integer coordinates, each between -4000 and 4000, inclusive. No two stars are at the same position. No star is at the origin.

After reading in these two lines, you may start measurements. To measure the distance between stars i and j, write a line of the form "`measure` i j", where i and j are distinct integers between 1 and n, inclusive. In response, an input line containing an integer denoting the squared distance between stars i and j becomes available. You can perform up to **300 measurements**.

When you have determined the distances between all pairs of the stars, write a line containing only `answer`, followed by $n - 1$ lines of the following format.

$$
\begin{array}{l}
d_{1,2}\ d_{1,3}\ \cdots\ d_{1,n} \\
d_{2,3}\ \cdots\ d_{2,n} \\
\vdots \\
d_{n-1,n}
\end{array}
$$

Here, $d_{i,j}$ ($1 \le i < j \le n$) is an integer denoting the squared distance between stars i and j. Note that you do not need to determine the stars' coordinates. After writing these lines, the interaction stops, and your program must terminate without any extra output.

If your output does not conform to the specifications above, or if the number of measurements exceeds 300, your submission will be judged as a wrong answer.

The coordinates of the stars are fixed before the interaction starts; they do not change during the interaction.

You are provided with a command-line tool for local testing. For more details, refer to the clarifications in the contest system.

Sample Interaction 1

Read	Write
`3`	
`1 11 4`	
	`measure 1 2`
`14`	
	`measure 2 3`
`11`	
	`measure 3 1`
`9`	
	`answer`
	`14 9`
	`11`

In Sample Interaction 1, the stars' coordinates can be as follows:

- star 1 is at $(1, 0, 0)$,
- star 2 is at $(-1, -1, 3)$, and
- star 3 is at $(-2, 0, 0)$.

The distance between stars 1 and 2 is $\sqrt{14}$. In the first measurement, the squared distance, 14, is returned.

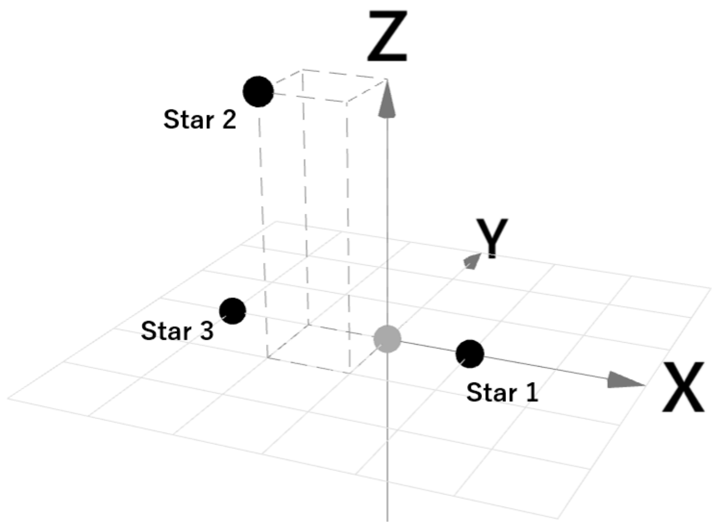


Figure F.1. Illustration of Sample Interaction 1

Read	Sample Interaction 2	Write
`4`		
`47944017 47920034 47960009 47968006`		
		`answer`
		`191728099 2 191824043`
		`191760077 12`
		`191856029`

Read	Sample Interaction 3	Write
`5`		
`1 4 9 50 149`		
		`answer`
		`5 10 45 162`
		`13 38 181`
		`29 206`
		`371`

In Sample Interaction 2, the stars' coordinates can be as follows:

- star 1 is at $(-3998, -3998, -3997)$,
- star 2 is at $(3997, 3997, 3996)$,
- star 3 is at $(-3999, -3998, -3998)$, and
- star 4 is at $(3999, 3999, 3998)$.

In Sample Interaction 3, the stars' coordinates can be as follows:

- star 1 is at $(1, 0, 0)$,
- star 2 is at $(0, -2, 0)$,
- star 3 is at $(0, 0, 3)$,
- star 4 is at $(3, -4, 5)$, and
- star 5 is at $(-6, 7, -8)$.

Problem G
Charity Raffle
Time Limit: 3 seconds

You are joining a charity raffle in which several types of prizes can be won. The raffle is held in a somewhat unusual way: with one draw using a single draw ticket, two prizes of different types are randomly picked up. However, you don't win both of them; you should choose one out of the two.

You have a certain number of draw tickets. You will use all of them and win the same number of prizes. As you don't want too many prizes of the same type, when choosing out of the two prizes, you choose the one that you have won fewer so far. You may have won the same number of the two prize types, including cases where both are zero. The prize types are sequentially numbered, and you choose the one with the lower type number in that case.

Despite the above strategy, it is not certain that you can avoid winning too many prizes of the same type. You feel *unhappy* if there is a prize type for which the number of prizes you win exceeds a certain limit. You want to know how many possible combinations of prizes you may get that do *not* make you unhappy after using up all of your draw tickets. Two combinations of prizes are considered different when they differ in the count of at least one type of prize. You may assume that there is an unlimited supply of any type of prize.

Input

The input consists of a single test case of the following format.

n k m

The first integer n ($2 \leq n \leq 10^6$) is the number of different types of prizes. The second integer k ($1 \leq k \leq 10^6$) is the number of draw tickets you have. The third integer m ($1 \leq m \leq k$) is the maximum number of prizes of a single type that does not make you unhappy.

Output

Output the number of possible combinations of prizes that do not make you unhappy, modulo $998\,244\,353$, which is a prime number.

Sample Input 1	Sample Output 1
`3 1 1`	`2`

Sample Input 2	Sample Output 2
`3 3 2`	`4`

Sample Input 3	Sample Output 3
`3 3 1`	`1`

Sample Input 4	Sample Output 4
`2025 1207 64`	`660312977`

In Sample Input 1, you will win either the first or the second type of prize, but not the third.

In Sample Input 2, the following four combinations of prizes are possible. In any of these, the number of prizes of any single type is no more than $m = 2$, and thus, you are not unhappy.

- Two prizes of the first type and one prize of the second type
- Two prizes of the first type and one prize of the third type
- Two prizes of the second type and one prize of the third type
- One prize of each of the three types

In Sample Input 3, only the last one of the four combinations above does not contain more than one prize of any single type.

Problem H
U-Shaped Panels
Time Limit: 2 seconds

A rectangular pond is in the backyard of your house. As the length and width of the pond are integer multiples of one meter, the pond can be considered to consist of one-meter-square sections.

You always feel the pond is too large, and you want to cover some of its sections using the panels kept in the barn. All of these panels have the same size and the same shape, consisting of one-meter-square boards arranged in a U shape. The overall size of the panels is k meters by k meters, and $3k - 2$ boards are on the three edges of that square. The rest, the $(k-2) \times (k-1)$ rectangular area, is void. Figure H.1 illustrates a panel of size $k = 5$.

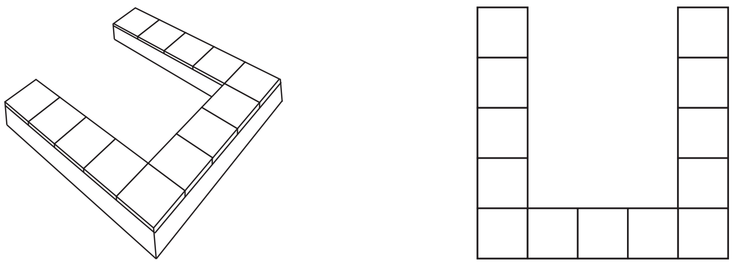


Figure H.1. A panel of size $k = 5$ (left) and its top view (right)

You plan to cover certain one-meter-square sections of the pond with the panels and leave the rest uncovered. Panels should be placed so that each of their boards fits exactly one section. As long as this is satisfied, orientations of the panels can be arbitrarily chosen. Panels should not overlap one another nor stick out of the pond.

Determine whether your plan is feasible or not.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 10$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

$$\begin{array}{l} n\ m\ k \\ s_1 \\ \vdots \\ s_n \end{array}$$

The first line contains three integers n, m, and k. The integers n and m denote the length and the width of the pond, respectively ($5 \le n \le 1000$, $5 \le m \le 1000$). The integer k denotes the size of the U-shaped panels ($\mathbf{5} \le k \le 1000$). The following n lines represent your plan. The i-th of them contains a string s_i of length m consisting of the characters '`#`' and '`.`'. Let us say a one-meter-square section is at (r, c) if it lies between $r - 1$ meters and r meters from the front edge, and between $c - 1$ meters and c meters from the left edge. For each i and j, if the j-th character of s_i is '`#`', the section at (i, j) should be fully covered with a board of a panel. Otherwise, the section should remain fully uncovered.

The sum of n's over all the test cases does not exceed 1000. The same applies to m.

Output

For each test case, output in a line `yes` if your plan is feasible; `no` otherwise.

Sample Input 1

```
1
10 10 5
..........
..........
....#####.
..#.#.#.#.
..#.#.#.#.
..#.#.#.#.
..#.#.#.#.
..#####...
..........
..........
```

Sample Output 1

```
yes
```

Sample Input 2

```
2
5 21 5
.#...##...###....##..
.#..#..#..#..#..#..#.
.#..#.....###...#....
.#..#..#..#.....#..#.
.#...##...#......##..
6 14 6
.############.
.#..........#.
.#..........#.
.#..........#.
.#..........#.
.############.
```

Sample Output 2

```
no
yes
```

Problem I
Game of Names
Time Limit: 2 seconds

Alice and Bob are playing a game on a board with a certain number of cells in a single row. Some (possibly none) of the cells have a player's name written in them, either "Alice" or "Bob". Other cells are initially blank.

Starting with Alice, the two players alternately take moves. In one move, the player in turn chooses a blank cell that does not have an adjacent cell with the player's own name, and then writes the player's name in the chosen blank cell. Note that the opponent's name in an adjacent cell does not matter.

The player who cannot make a move loses the game. Given the initial state of the board, determine which of Alice and Bob will win when they play their best.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 2 \times 10^5$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

n
s

The integer n represents the number of cells on the board ($1 \le n \le 2 \times 10^5$). The initial state of the board is given as a string s of length n.

For each i ($1 \le i \le n$), the i-th character s_i of s is either '`a`', '`b`', or '`.`', and represents the initial state of the i-th cell from the left. Here, s_i is '`a`' if the i-th cell contains the name Alice, '`b`' if it contains the name Bob, and '`.`' if it is blank.

It is guaranteed that the initial board does not contain two adjacent cells with the same name.

The sum of n's over all the test cases does not exceed 2×10^5.

Output

For each test case, output `alice` if Alice wins and `bob` if Bob wins, in one line.

Sample Input 1	Sample Output 1
3 2 .. 3 .a. 4 ab..	bob bob alice

Problem J
ICPC Board
Time Limit: 2 seconds

As an archaeologist, you have discovered a rectangular wooden board in the ruins of an ancient city. The board is divided into a grid, and each grid cell appears to have been engraved originally with one of the letters '`C`', '`I`', and '`P`'. However, due to decay over time, some of the letters are now indistinguishable.

During your investigation, you made the following hypothesis: any square of 2×2 cells on the board originally had two '`C`'s, one '`I`', and one '`P`'.

You now want to check whether this hypothesis is consistent with the discovered board. If it is, show one possibility of the original arrangement of the letters that aligns with the hypothesis.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 500$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

$$
\begin{array}{l}
n \ m \\
c_{1,1} \, c_{1,2} \cdots c_{1,m} \\
c_{2,1} \, c_{2,2} \cdots c_{2,m} \\
\vdots \\
c_{n,1} \, c_{n,2} \cdots c_{n,m}
\end{array}
$$

The first line of a test case contains two integers n and m ($2 \le n \le 1000$, $2 \le m \le 1000$). They represent the number of rows and columns of the board, respectively. The next n lines, each containing m characters, describe the discovered board. The j-th character of the i-th line, $c_{i,j}$, is one of '`C`', '`I`', '`P`', and '`?`'. If $c_{i,j}$ is '`C`', '`I`', or '`P`', the cell in row i and column j is identifiable as having that letter. If $c_{i,j}$ is '`?`', the letter in that cell is indistinguishable.

The sum of n's over all the test cases does not exceed 1000. The same applies to m.

Output

For each test case, if the hypothesis is not consistent with the discovered board, output `no` in a single line. Otherwise, output `yes` in the first line, followed by n lines representing one possibility of the original arrangement of the letters that aligns with the hypothesis. Each of these n lines should contain m characters. The j-th character of the i-th line should be the letter in the cell in row i and column j. If there are multiple possible arrangements, you may output any of them.

Sample Input 1	Sample Output 1
3 5 7 I?I?I?I ?P?P?P? I?I?I?I ?P?P?P? I?I?I?I 4 4 ICPC CPCI ICPC CPCI 2 2 ?? ??	yes ICICICI CPCPCPC ICICICI CPCPCPC ICICICI no yes IC PC

Problem K
Membership Structure of a Secret Society
Time Limit: 2 seconds

A secret society with an undisclosed name was established in the year 1899 by a single founder, whose name is also kept secret. Subsequent members have joined the society through recommendations of existing members.

One unique rule for joining the society has been strictly enforced: Recommendations can be made by one or more existing members, but the same member group can recommend only one new member. For example, if a member was allowed to join upon the recommendation by a group of existing members $\{A, B, C\}$, no other persons can be allowed by the same recommender group. It is perfectly acceptable, however, for a group $\{A, B\}$ to recommend another new member. Although the set $\{A, B\}$ is a subset of $\{A, B, C\}$, they are distinct sets. For consistency, the group of recommenders of the founder is considered to be the empty set.

Through investigation of this secret society, you have obtained several information fragments representing some part of the membership structure of the society. Each information fragment takes one of the following forms of statements, in which the symbols a, b and c are integers designating certain members of the society.

`recommend` a b
: meaning that the member a belongs to the group that recommended the member b to join.

`not-recommend` a b
: meaning that the member a does *not* belong to the group that recommended the member b to join.

`intersection` a b c
: meaning that the group of the recommenders of the member a is the set intersection of recommender group of b and that of c. In other words, all of those who recommended a also recommended both b and c, and all of those who recommended both b and c also recommended a.

Two or more occurrences of the same integer mean the same member, even in different statements. On the other hand, two or more different integers may be aliases of the same person, even in a single statement.

The obtained information may be partial, that is, the recommendations of some members may be missing, and, moreover, there may be some members not mentioned in any of the statements.

As the information sources are not necessarily reliable, some false information might have crept in. You would like to know whether these statements are consistent, that is, whether there can be a recommendation relationship consistent with all of these statements.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 3000$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

$$\begin{array}{l} n \\ s_1 \\ \vdots \\ s_n \end{array}$$

The first line contains a single integer n, the number of statements ($1 \le n \le 3000$). Each of the following n lines is in either of the formats "`recommend` a b", "`not-recommend` a b", or "`intersection` a b c", with all of a, b, and c being integers between 1 and $3n$, inclusive.

The sum of n's over all the test cases does not exceed 3000.

Output

For each test case, output `yes` in one line if the situation described in the statements is possible, and output `no`, otherwise.

Sample Input 1

```
3
2
recommend 1 2
not-recommend 1 2
2
recommend 1 2
recommend 2 1
3
intersection 1 2 2
recommend 1 3
not-recommend 2 3
```

Sample Output 1

```
no
no
no
```

Sample Input 2	Sample Output 2
4 3 intersection 3 2 1 recommend 3 2 not-recommend 3 1 4 intersection 1 2 3 recommend 4 2 recommend 4 3 not-recommend 4 1 3 intersection 3 2 1 recommend 2 5 intersection 3 4 5 5 recommend 1 3 not-recommend 2 3 not-recommend 3 2 not-recommend 1 2 not-recommend 2 1	yes no yes yes

In Sample Input 1, all the test cases describe impossible situations.

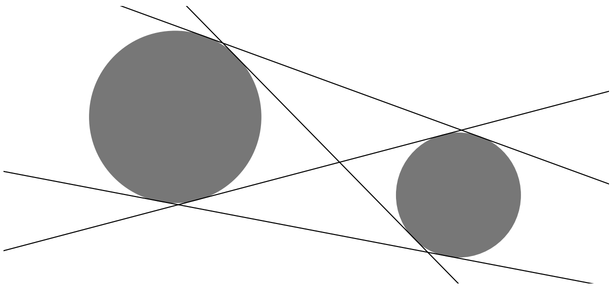
- In the first test case, the two statements are clearly contradicting each other.
- In the second test case, the first statement implies that the member 1 joined the society first, and later, the member 2 joined, because otherwise the member 1 could not have recommended the member 2. The second statement, however, implies the reverse order. Satisfying both is impossible.
- In the third test case, the first statement essentially says that the recommender set of the member 1 is identical to that of the member 2. As no two members have the same recommender set, 1 and 2 must designate the same member. Then, the second and the third statements are contradictory.

The first test case of Sample Input 2 is possible. There are many possible scenarios. One example is as follows: the member 3 is actually the founder, the member 2 joined by the recommendation of $\{3\}$, and the member 1 joined by the recommendation of $\{2\}$.

The last test case of Sample Input 2 is also possible. Note that it is not necessary that all the members of the society are mentioned in the information. There must be at least 4 members in the society.

Problem L
Common Tangent Lines
Time Limit: 2 seconds

It is well known that two disks on a plane have four distinct common tangent lines when they are disjoint (neither overlapping nor touching).



You are given four lines on the xy-plane. Your objective is to apply parallel translations to these lines so that they become four distinct common tangent lines of some pair of disjoint disks with positive radii. You want to do this with a *cost* as low as possible, where the cost is defined as the sum of the translation distances. The translation distance for a line is the distance between the lines before and after the translation.

Each line is specified by two parameters, a and d, and is defined by

$$x \cos\left(\frac{\pi a}{180}\right) + y \sin\left(\frac{\pi a}{180}\right) = d.$$

For example, the line with $(a, d) = (60, 1)$ represents $x/2 + \sqrt{3}y/2 = 1$, since $\cos(\pi/3) = 1/2$ and $\sin(\pi/3) = \sqrt{3}/2$.

First, determine whether the objective is achievable. If it is, determine the *minimum required cost,* defined as follows: the minimum value c such that the objective is achievable with a cost at most $c + \varepsilon$ for any positive real ε. The objective does not have to be achievable with the cost exactly equal to c.

Input

The input contains one or more test cases. The first line of the input contains an integer t ($1 \le t \le 1000$), which is the number of test cases. The descriptions of the t test cases follow, each in the following format.

$$\begin{array}{l} a_1\ d_1 \\ \vdots \\ a_4\ d_4 \end{array}$$

For $i = 1, \ldots, 4$, two integers a_i and d_i are the parameters that specify the i-th line ($0 \le a_i < 180$, $-1000 \le d_i \le 1000$). Two or more lines may be identical before translations.

Output

For each test case, if the objective is not achievable, output `no` in a line. Otherwise, output the minimum required cost described above in a line. The absolute or relative error of the output must be less than or equal to 10^{-7}.

Sample Input 1

```
4
90 0
90 0
45 0
135 0
0 -200
0 100
30 0
150 0
120 100
120 75
30 50
30 -100
178 -132
144 -83
165 199
19 31
```

Sample Output 1

```
0
86.602540378444
no
173.814220263386
```

In the first test case of Sample Input 1, you have to move at least one of the first two identical lines (Figure L.1 (a)). For $\varepsilon > 0$, translating one line by $\varepsilon/2$ in the positive y-direction and the other by $\varepsilon/2$ in the negative y-direction achieves the objective with cost ε. This results in four tangent lines for disks with radii $\varepsilon/2$ (Figure L.1 (b)). Since ε can be arbitrarily small, the minimum required cost is 0. The remaining cases are illustrated in Figures L.1 (c) to (e).

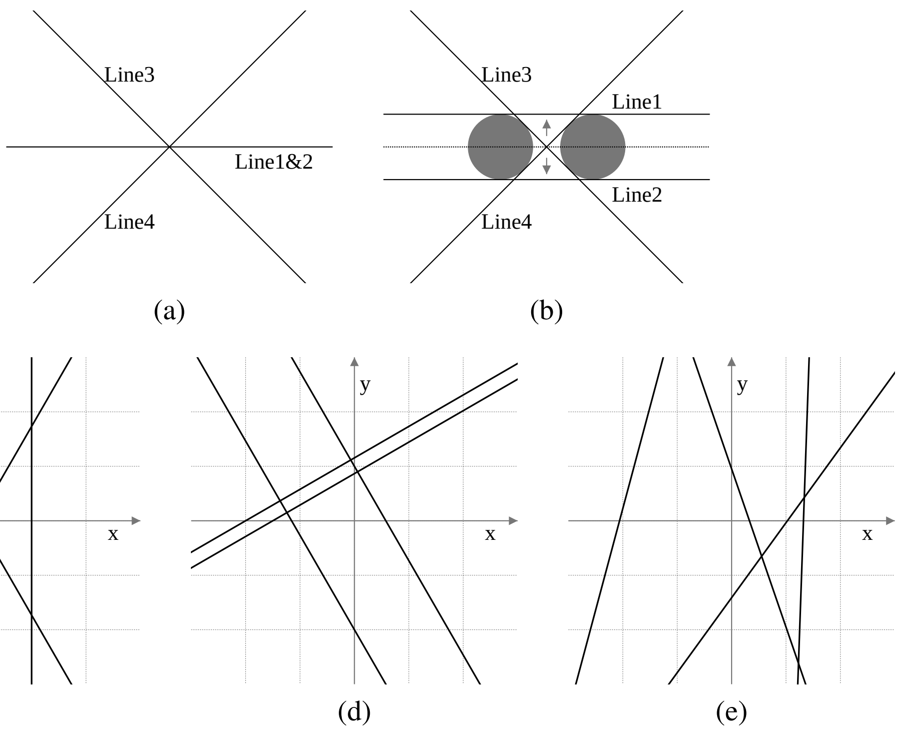


Figure L.1. Illustration of Sample Input 1