The 2nd Universal Cup

Uni
Cup

Stage 10: Harbin

November 18-19, 2023

This problem set should contain 13 problems on 23 numbered pages.

Based on

China Collegiate Programming Contest (CCPC)

Hosted by

Prepared by

Problem A. Go go Baron Bunny!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Bad Frog is not willing to pay very much for programming competitions, and would provide only n brain cells to memory programming competition knowledge points.

In the morning, Bad Frog's teammates find that there are k knowledge points in Bad Frog's mind and they cost $a_1, a_2, \cdots, a_k$ brain cells respectively.

Since Bad Frog hardly practices, the number of brain cells to keep each knowledge point will decrease by 1 every day, and when the number becomes 0, Bad Frog will forget the knowledge point.

Every night, teammates find that Bad Frog does no practice and the brain cells to memory knowledge points decreases, so they will push Bad Frog to use all the remaining brain cells to learn a new knowledge point.

For example, if Bad Frog provides 22 brain cells and costs $1, 4, 7, 1, 5, 4$ brain cells to keep knowledge points respectively in the morning, then at night the brain cells cost will become $0, 3, 6, 0, 4, 3$ respectively, and then Bad Frog will forget the 2 knowledge points that are kept with zero brain cells. Finally, under the pressure from teammates, Bad Frog will learn a new knowledge point and use all the 6 remain brain cells to keep it so that the brain cells cost will become $3, 6, 4, 3, 6$ respectively.

Denote $A = \{a_1, a_2, \cdots, a_k\}$ as the multiset of the number of brain cells to keep knowledge points. Now there are t days to go before the competition. In order to have a stable performance, teammates hope that the **multiset** A on the competition day (t days from now) to be the same as it is today. Determine the number of **array**s $(a_1, a_2, \cdots, a_k)$ satisfying this condition, modulo 998244353.

Input

$n\ k\ t$

Constraints:

- $1 \le k \le n \le 10^{12}$
- $1 \le t \le 10^{12}$

Output

Print the answer modulo 998244353, as an integer.

Examples

standard input	standard output
2 1 2	1
8 4 2	6
29 7 154	0
50 10 10	77225400

Note

- For the first case, the only one array is $[2]$ ($[2] \to [1, 1] \to [2]$).
- For the second case, the 6 arrays are $[1, 1, 3, 3]$, $[1, 3, 1, 3]$, $[1, 3, 3, 1]$, $[3, 1, 1, 3]$, $[3, 1, 3, 1]$, $[3, 3, 1, 1]$.

Problem B. Memory

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Singing a song with cups of drink, That days are short is what I think. They are like dews in morning early, Therefore, I feel suffered for time gone surely.

Little G used to be a participant in programming contests and he had attended n contests in total, and for the i-th contest, Little G got a_i happiness. However, Little G would also be influenced by past contests since memory also plays an important role to influence one's mood. So we can use following formula to value Little G's mood after the i-th contest:

$$Mood(i) = \sum_{j=1}^{i} 2^{j-i} \times a_j$$

Now Little G is recalling the past and is curious about the moods after every contest, so he wants to tag the moods for every contest. Specifically, Little G will tag a positive sign ("+") for the i-th contest if $Mood(i) > 0$, or tag a negative sign ("-") if $Mood(i) < 0$, or tag a zero ("0") if $Mood(i) = 0$. But Little G is busy working and working, so he is now asking you, the best programming contestant, to help him tag the moods.

Input

The first line contains one integers n ($1 \le n \le 10^5$), denoting the number of contests Little G had attended.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$), denoting the happiness values after every contest.

Output

Output one line containing a string which is of length n and only contains "+", "-" or "0", denoting the mood tags after every contest.

Example

standard input	standard output
10 2 -1 4 -7 4 -8 3 -6 4 -7	+0+-+---+-

Note

- $Mood(1) = 2^0 \times 2 = 2 > 0$
- $Mood(2) = 2^{-1} \times 2 + 2^0 \times (-1) = 0$
- $Mood(3) = 2^{-2} \times 2 + 2^{-1} \times (-1) + 2^0 \times 4 = 4 > 0$
- $Mood(4) = 2^{-3} \times 2 + 2^{-2} \times (-1) + 2^{-1} \times 4 + 2^0 \times (-7) = -5 < 0$
- $Mood(5) = 2^{-4} \times 2 + 2^{-3} \times (-1) + 2^{-2} \times 4 + 2^{-1} \times (-7) + 2^0 \times 4 = \frac{3}{2} > 0$
- $\cdots$

Problem C. Karshilov's Matching Problem II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Karshilov, as always, likes the string matching problem. This time, he gives a string S of length n and assigns a value to each prefix of S. Specifically, the prefix of S with a length of $i(1 \le i \le n)$ is pre_i and its value is w_i.

For any string t, He defines a value function $f(t) = \sum_{i=1}^{n} w_i \cdot occur(t, pre_i)$ based on the prefixes of given S, where $occur(t, pre_i)$ indicates the number of times pre_i occurs in the string t. For example: $occur(\texttt{heheh}, \texttt{heh}) = 2$ and $occur(\texttt{hhh}, \texttt{h}) = 3$.

Now, Karshilov has another string T of length n. He will give you m queries. And each query will contains two integers l, r, indicating to query the value of $f(T[l, r])$, where $T[l, r]$ represents a substring from the l-th character to the r-th character of the string T (that is, $T_l T_{l+1} \cdots T_r$).

Can you solve Karshilov's queries like you did two years ago?

Input

The first line contains two integers, $n, m (1 \le n, m \le 150,000)$, indicating the length of string S (string T) and the number of queries.

The second line contains a string S of length n.

The third line contains a string T with a length of n.

The fourth line contains n integers, $w_1, w_2, \cdots w_n$, where $w_i (0 \le w_i \le 10^8)$ is the value of pre_i .

For the next m lines, each line contains two integers $l, r (1 \le l \le r \le n)$, which means asking the value of $f(T[l, r])$.

String S and T are both composed of lowercase letters.

Output

The output contains m lines. The i-th line contains an integer, indicating the answer of the i-th query.

Examples

standard input	standard output
8 5 abbabaab aababbab 1 2 4 8 16 32 64 128 1 1 2 3 3 5 4 7 1 8	1 3 3 16 38
15 4 heheheheehhejie heheheheheheheh 3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 2 3 4 8 2 6 1 15	3 13 13 174

Problem D. A Simple MST Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

For the positive integer x, we define the number of its different prime factors as $\omega(x)$. For example, $\omega(1) = 0, \omega(8) = 1, \omega(12) = 2$.

In this problem, we treat each positive integer as a node. When we build an edge between node x and node y, we will cost $\omega(lcm(x, y))$, where $lcm(x, y)$ represents the least common multiple of x and y.

Next, you will be given T queries. For the i-th query we will give two integers l_i, r_i. What you need to answer is, when only considering nodes $l_i, l_i + 1, \cdots r_i$, what is the minimum cost if we build edges so that these $r_i - l_i + 1$ nodes can reach each other.

Note that all of the queries are distinct and in i-th query you can only build an edge between x, y when $l_i \leq x, y \leq r_i$.

Input

The first line contains an integer $T (T \leq 50000)$, indicating the number of queries.

For the next T lines, the i-th line contains two integers $l_i, r_i (1 \leq l_i \leq r_i \leq 10^6)$, indicating a query.

It is guaranteed that $\sum_{i=1}^{T} r_i \leq 10^6$.

Output

For each query, output an integer as your answer.

Examples

standard input	standard output
5 1 1 4 5 1 4 1 9 19 810	0 2 3 9 1812
2 27 30 183704 252609	8 223092

Problem E. Revenge on My Boss

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Bob is a businessman and Alice is his assistant. They planned to collect the raw materials in n cities, and then meet in one of the cities to process and sell the products.

Specifically, their plan is determined by the following steps:

- First, Alice arranges the n cities in any order to get the city sequence $p_1, p_2, \cdots p_n$(a permutation).
- Then, Bob selects a meeting city among the n cities. That is, Bob will select an integer $m(1 \le m \le n)$ and then use the city p_m as the meeting city.
- Next, they work from both ends of the city sequence. Alice goes to city $p_1, p_2 \cdots p_m$ for raw material collection, and Bob goes to city $p_n, p_{n-1}, \cdots p_m$for raw material collection.

Finally, Alice and Bob will meet at city p_m and process all the raw materials to produce products(one material can be used to produce exactly one product). After that they will sell all the products in the city p_m. Their final income can be calculated using the following information:

- If the city i is visited by Alice, then a_i raw materials will be collected by Alice.
- If the city i is visited by Bob, then b_i raw materials will be collected by Bob.
- Each product sold in the city i will earn c_i gold.
- All can know, the total income of the plan is $\left(\sum_{i=1}^{m} a_{p_i} + \sum_{i=m}^{n} b_{p_i}\right) \cdot c_{p_m}$ gold.

Bob really wants to make money, so when he selects the meeting city p_m, he must try to make the total income as large as possible.

Alice knows Bob is greedy, so she knows Bob's strategy for choosing the meeting city. However, Bob often mistreats Alice, which makes Alice unhappy. So in the first step, Alice wants that the total income of the plan is as small as possible finally.

Can you help Alice plan the city sequence? Please output the city sequence $p_1, p_2, \cdots p_n$ to make the finally total income of the plan as small as possible.

Input

Each test data of this problem has multiple cases.

The first line contains an positive integer T, indicating the number of cases.

For each case:

- The first line contains an integer $n(1 \le n \le 10^5)$, indicating the number of cities.
- For the next n lines, the i-th line contains 3 integers indicating the information of the i-th city, $a_i, b_i, c_i (1 \le a_i, b_i, c_i \le 10^6)$, whose meanings are described above.

It is guaranteed that sum of n over all cases is not greater than 10^5. That is, $\sum n \le 10^5$.

Output

For each case, output a line, which contains n integers, $p_1, p_2, \cdots p_n$, indicating the order planned by Alice to make the total income as small as possible. If there are multiple solutions, you can print any of them.

Example

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

Problem F. Palindrome Path

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given an $n \times m$ grid maze, each cell (i, j) is either blank or blocked. Now, George is in the maze and the start cell is (sr, sc), he needs to visit all the blank cells in the maze and go to the exit cell (er, ec) finally. George can do 4 kinds of moves to travel the maze, assuming George is on cell (i, j) currently:

1. Try to move left(denoted as "L"). If $j > 1$ and cell $(i, j - 1)$ is blank, George will move to cell $(i, j - 1)$, or George will stay at (i, j).
2. Try to move right(denoted as "R"). If $j < m$ and cell $(i, j + 1)$ is blank, George will move to cell $(i, j + 1)$, or George will stay at (i, j).
3. Try to move up(denoted as "U"). If $i > 1$ and cell $(i-1, j)$ is blank, George will move to cell $(i-1, j)$, or George will stay at (i, j).
4. Try to move down(denoted as "D"). If $i < n$ and cell $(i + 1, j)$ is blank, George will move to cell $(i + 1, j)$, or George will stay at (i, j).

So his task is to do several moves to **visit all blank cells** and **go to the exit cell** finally, where the start cell is considered to be visited in the beginning. To make it more fun, he wants to make his move sequence palindrome. Formally, denote his move sequence is $M_1, M_2, \cdots, M_k\,(M_i \in \{L, R, U, D\})$, then for all integers $i\,(1 \le i \le k)$, M_i and M_{k-i+1} should be the same.

Please help him find a palindrome move sequence to **visit all blank cells** and **go to the exit cell** finally. If multiple solutions exist, print any one of them. If no solution, report it.

Input

The first line contains two integers $n, m\,(1 \le n, m \le 30)$, denoting the size of the maze.

The following n lines each contains a binary string, denoting the given maze. The j-th character in the string in the i-th line will be "1" if cell (i, j) blank, while it will be "0" if cell (i, j) is blocked.

The next line contains four integers $sr, sc, er, ec\,(1 \le sr, er \le n, 1 \le sc, ec \le m)$, denoting the positions of the start cell and the exit cell.

It is guaranteed that the start cell (sr, sc) and exit cell (er, ec) are both blank.

Output

If solution exists, print the move string $S\,(0 \le |S| \le 10^6)$ in one line, or print "-1" in one line.

If the goal could be achived by doing no operations in a test case, outputing an empty string is ok. But you should output an empty line in this situation.

DO NOT add extra space at the end of your output, or you will get "Wrong Answer" verdict.

Examples

standard input	standard output
2 2 11 11 1 1 2 2	RDLUULDR
2 2 10 01 1 1 2 2	-1

Note

- For the first case, the path based on the move sequence is as follows:
  1. Try to move right from $(1,1)$, since $j = 1 < m$ and $(1,2)$ is blank, George will move to $(1,2)$
  2. Try to move down from $(1,2)$, since $i = 1 < n$ and $(2,2)$ is blank, George will move to $(2,2)$
  3. Try to move left from $(2,2)$, since $j = 2 > 1$ and $(2,1)$ is blank, George will move to $(2,1)$
  4. Try to move up from $(2,1)$, since $i = 2 > 1$ and $(1,1)$ is blank, George will move to $(1,1)$
  5. Try to move up from $(1,1)$, since $i = 1$, George will stay at $(1,1)$
  6. Try to move left from $(1,1)$, since $j = 1$, George will stay at $(1,1)$
  7. Try to move down from $(1,1)$, since $i = 1 < n$ and $(2,1)$ is blank, George will move to $(2,1)$
  8. Try to move right from $(2,1)$, since $j = 1 < m$ and $(2,2)$ is blank, George will move to $(2,2)$

  It can be seen that all blank cells are visited and the ending cell is exactly the exit cell $(2,2)$.
- For the second case, George from $(1,1)$ can never reach $(2,2)$, so print "-1" in one line.

Problem G. The Only Way to the Destination

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

There is a maze that can be represented as $n \times m$ grids. For convenience, we define (x, y) as the grid in the x-th row and y-th column.

Initially, every grid is empty. Alice wants to design this maze. So she will place k walls in the grids. Each wall is represented as x_1, x_2, y, where it means that $\forall x_1 \le i \le x_2$, (i, y) will becomes a part of this wall and cannot be passed by. Except wall grids, all of the other grids are empty. Alice ensures that after placing k walls, all empty grids remain connected and the maze has at least one empty grid. And it is guaranteed that different walls have no common grids.

Now, Alice want to know, does any pair of empty grids have only one simple path connecting them?

If your are not familar with the definition of "simple path", here is it:

A simple path connecting empty grids (x_s, y_s) and (x_d, y_d) is defined as a sequence of grid positions $S = \{(x_1, y_1), (x_2, y_2) \cdots (x_{len}, y_{len})\}, (len \ge 2)$ that satisfies the following conditions:

- $(x_1, y_1) = (x_s, y_s), (x_{len}, y_{len}) = (x_d, y_d)$
- $\forall 1 \le i \le len,\ 1 \le x_i \le n, 1 \le y_i \le m$
- $\forall 1 \le i \le len,\ (x_i, y_i)$ is an empty grid
- $\forall 1 \le i \le len - 1,\ |x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$
- $\forall 1 \le i < j \le len, (x_i, y_i) \ne (x_j, y_j)$

If any two different empty grids $(x_s, y_s), (x_d, y_d), \{(x_s, y_s) \ne (x_d, y_d)\}$ have exactly one simple path connecting them, output "YES". Otherwise, output "NO".

Input

The first line contains three integers $n, m, k (1 \le n, m \le 10^9, 1 \le k \le 10^5)$, indicating the number of rows, the number of columns, and the number of walls.

For the next k lines, each line contains three integers $x_1, x_2, y (1 \le x_1 \le x_2 \le n, 1 \le y \le m)$ indicating a wall placed in the maze.

It is guaranteed that each pair of empty grids has at least one simple path connecting them.

Output

Output a string to represent the answer, "YES" or "NO".

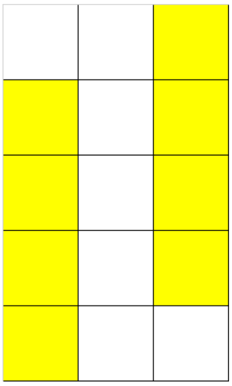
Examples

standard input	standard output
5 3 2 2 5 1 1 4 3	YES
5 3 1 2 4 2	NO
2 4 2 2 2 1 1 1 4	NO

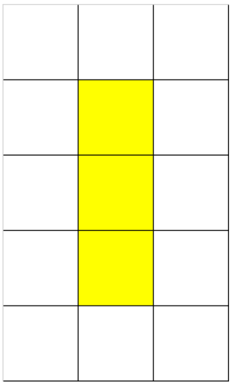
Note

Sample explanations contains Alice's maze, with yellow squares representing walls and white squares representing empty grids.

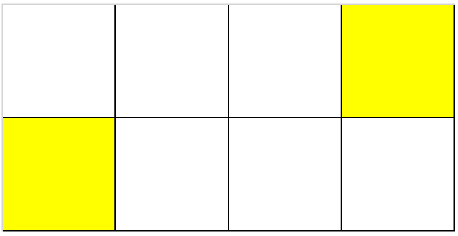
1. In the first example, we can observe that regardless of the choice of starting and ending grid, there is only one simple path.



2. In the second example, choose $(1, 1)$ as the starting grid and $(5, 1)$ as the destination grid. $S_1 = \{(1, 1), (2, 1), (3, 1), (4, 1), (5, 1)\}$ and $S_2 = \{(1, 1), (1, 2), (1, 3), (2, 3), (3, 3), (4, 3), (5, 3), (5, 2), (5, 1)\}$ are two different simple paths.



3. In the third example, choose $(1, 2)$ as the starting grid and $(2, 3)$ as the destination grid. $S_1 = \{(1, 2), (1, 3), (2, 3)\}$ and $S_2 = \{(1, 2), (2, 2), (2, 3)\}$ are two different simple paths.



Problem H. Energy Distribution

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n planets in the galaxy. Some undirected tunnels connect planets. There exists at most one tunnel connecting each pair of planets. So these tunnels can be described as an $n \times n$ matrix $W_{n \times n}$. Specifically, the tunnel connecting planet i and j has a width of $w_{i,j}$(If there is no tunnel between planet i and j, then $w_{i,j} = 0$).

Now, you want to distribute exactly 1.0 unit of energy among the n planets. Suppose that planet i is distributed e_i(a real number) unit of energy ($e_i \geq 0, \sum_{i=1}^{n} e_i = 1$), these planets will bring E magical value, where $E = \sum_{i=1}^{n} \sum_{j=i+1}^{n} e_i e_j w_{i,j}$.

Please distribute the energy and maximize the magical value.

Input

The first line contains an interger $n(1 \leq n \leq 10)$.

For the next n lines, each line contains n intergers. The j-th integer in the i-th line is $w_{i,j}(0 \leq w_{i,j} \leq 1000)$. Indicating the matrix $W_{n \times n}$.

Output

Output a real number as the answer. If your answer is A while the standard answer is B, your answer will be accepted if and only if $\frac{|A-B|}{\max(|A|,1)} \leq 10^{-6}$.

Examples

standard input	standard output
2 0 1 1 0	0.250000
3 0 2 1 2 0 2 1 2 0	0.571429

Problem I. Rolling For Days

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

In the game TFT, players need to buy units in the store to build up a powerful lineup against other players. Only one-star units will be refreshed in the store, three identical one-star units can be synthesized into a corresponding two-star unit, and three identical two-star units can be synthesized into a corresponding three-star unit. Higher star units will obviously be much more powerful than lower star units. The units refreshed in the store are random, and refreshing requires coins. So it's possible that you can't get the units you want although having a lot of coins.

An annoying example from the game

As a disciplined player, you're curious about just how many refreshes are needed to complete a particular deck. To that end, you're going to start by solving the following simplified problem:

There are a total of n cards in the store's card pool, which are categorized into m kinds, where the i-th kind of card has a total of a_i cards. You have decided on your target deck, and your target deck requires b_i cards of the i-th kind. Each time you refresh, one card will be randomly drawn from the remaining cards in the card pool with equal probability. If the drawn card is not needed (i.e., the number of cards in this kind already purchased is equal to the number that your target deck requires), you'll put the card right back into the card pool; otherwise, you'll purchase the card and remove it from the card pool. How many refreshes do you expect to need to complete your target deck? Since the answer may be large and floating point errors cannot be ignored, please output the answer modulo 998244353.

Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Input

The first line contains two integers $n, m (1 \le n \le 1000; 1 \le m \le 12)$, denoting the total number of cards and the number of kinds.

The second line contains m integers $a_1, a_2, ..., a_m (1 \le a_i \le n, \sum_{i=1}^{m} a_i = n)$, denoting the numbers of cards of each type.

The third line contains m integers $b_1, b_2, ..., b_m (0 \le b_i \le a_i)$, denoting the numbers of cards required for your target deck.

Output

Output a single integer, denoting the expect number of refreshes modulo 998244353 .

Examples

standard input	standard output
2 2 1 1 1 1	2
4 2 2 2 2 1	582309210
1000 12 101 43 34 281 23 24 12 25 66 222 145 24 37 43 27 257 5 11 12 19 62 41 87 13	265294941

Problem J. Game on a Forest

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Plover and Georgia are playing a game on a forest graph. Given is an **undirected acyclic graph**(a forest) with n nodes and m edges. Now, Plover and Georgia take turns operating on the graph, with Georgia starting first.

Each operation must be one of the following types:

1. Select an edge in the graph. Then remove it.
2. Select a node in the graph. Then remove both the node and the edges attached to it.

When one of them is unable to operate on his/her turn(which means the graph is empty), the game is lost and the other one wins. Now, assuming that Plover and Georgia are both extremely smart, Georgia wants to know how many different first moves can make her win the game.

Input

The first line contains two integers $n, m (1 \le m < n \le 10^5)$, indicating the number of nodes and the number of edges, respectively.

For the next m lines, the i-th line contains two integers $u_i, v_i (1 \le u_i < v_i \le n)$, indicating an edge connected the u_i-th node and the v_i-th node.

It is guaranteed that for all $i, j (1 \le i < j \le m)$, $u_i \neq u_j$ or $v_i \neq v_j$ is satisfied.

Output

Output contains an integer $W (0 \le W \le n + m)$, indicating the number of different first moves that can make Georgia win.

Examples

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

Note

For the first sample, if Georgia selects node 1 or node 2 for the first move, she can win the game.

For the second sample, if Georgia selects any one of the 3 edges for the first move, she can win the game.

Problem K. Omniscia Spares None

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Our fortune is clouded with uncertainty.

The omniscia sees through you...

All things in this world have their laws...

Yet stratagems, constellations... are human creations!

The Disciples of Sanctus Glue-Catfish are cheating and oppressing the civilians. They have plundered and damaged many valuable works, treading on people's effort.

However, the Matrix of Prescience has stopped operating, the symbols have dimmed, and the Disciples of Sanctus Glue-Catfish have not been destroyed. Fu Xuan wants to restore the Matrix of Prescience, but she has no forces available. She needs your help to restart the Matrix of Prescience's base terminals and eliminate the abominations of Sanctus Glue-Catfish along the way.

Each base terminal can be seen as a chessboard. The n-th base terminal contains n layouts. To active the base terminal, you should move the layouts on the ground and connect some pairs of them to make the pattern satisfy:

- The coordinates of each layout are integers, and no two layouts are in the same position;
- The graph of layouts and connections is a **simple graph**;
- Each connecting line is a segment. And no two connecting segments intersect internally. That is, every two connecting segments can only intersect at endpoints(layouts);
- No more than 4 layouts have connections less than 6.

Now, given n, please find a solution for the n-th base terminal. (If there are multiple solutions, print any one of them. If no solution, report it.)

A **simple graph** is an undirected graph which is:

- Not a multigraph, that is, there is no more than one edge between each pair of vertices;
- Not a loop-graph, that is, there are no loops, that is, edges which start and end at the same vertex.

Input

n

Constraints:

- $1 \le n \le 100$

Output

If no solution, print "No" in one line. Else print "Yes" in one line and then print your arrangement in following format:

$$\begin{array}{|ll|} x_1 & y_1 \\ x_2 & y_2 \\ & \vdots \\ x_n & y_n \\ m & \\ u_1 & v_1 \\ u_2 & v_2 \\ & \vdots \\ u_m & v_m \end{array}$$

where $x_i, y_i\,(|x_i|, |y_i| \le 10^9)$ denotes the coordinates of the i-th layout, m denotes the number of pairs of the layouts you connect. $u_i, v_i\,(1 \le u_i, v_i \le n)$ denotes there is a connection between the u_i-th layout and the the v_i-th layout.

All numbers you output should be integers.

Examples

standard input	standard output
3	Yes -1 0 0 1 1 0 2 1 2 2 3
4	Yes -998244353 -998244353 0 998244353 6 6 1 1 2 1 2 4 3

Note

For more example, the following arrangements are all legal:

But this one is illegal because there are connections intersect internally:

Problem L. Palm Island

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Toph is playing a card game. She has n cards and each card has a unique number of $1, 2 \cdots n$. In this game, Toph can operate the deck of the cards. We may wish to assume that the cards from the top to the bottom of the deck are $p_1, p_2, \cdots p_n$ (a permutation), then each operation must be one of the following two cases:

1. Place the top card at the bottom of the deck, that is, change the order of the deck into $p_2, p_3 \cdots p_n, p_1$.
2. Place the second card from the top at the bottom of the deck, that is, change the order of the deck into $p_1, p_3 \cdots p_n, p_2$

Now, you know that the initial order(from top to bottom) of Toph's deck is $a_1, a_2, \cdots a_n$, and Toph wants to change the order of the deck into $b_1, b_2, \cdots b_n$ after some operations. Please construct the operation sequence to help Toph implement the change.

Toph has no patience. So the number of operations should not exceed n^2.

Input

The first line contains an integer T , indicating the number of test cases.

For each test case:

- The first line contains an integer, $n (3 \le n \le 1000)$, indicating the number of Toph's cards.
- The second line contains n integers $a_1, a_2, \cdots a_n$, a permutation indicating the order of the deck initially.
- The third line contains n integers $b_1, b_2, \cdots b_n$, a permutation indicating the order of the deck want to make.

It is guaranteed that the sum of n in T test cases is not exceed 1000.

Output

For each test case:

- Output a line, which contains a string $s_1 s_2 \ldots s_k (s_i \in \{1, 2\},\ 1 \le i \le k)$ as your operation sequence. The length of the string should not exceed n^2, or you will get "Wrong Answer".
- If there are multiple solutions, output any of them.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>2
3
1 2 3
2 3 1
4
1 2 3 4
2 1 3 4</td><td>1
112212</td></tr>
</table>

Note

If $a_1, a_2 \cdots a_n$ and $b_1, b_2 \cdots b_n$ are the same in a test case, outputing an empty string is ok. But you should output an empty line in this situation.

DO NOT add extra space at the end of lines, or you will get "Wrong Answer" verdict.

Problem M. Painter

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Little G is a painter and is painting on a 2D plane. Each integral point has a color character and the initial color characters for all integral points are "."(ASCII = 46). Now Little G is planning to do some operations one by one, where each operation is in one of the following three types:

1. "Circle $x\,y\,r\,col$", which means to draw a circle. Formally, change the color characters to col for these points (u, v) that $(u - x)^2 + (v - y)^2 \le r^2$.
2. "Rectangle $x_1\,y_1\,x_2\,y_2\,col$", which means to draw a rectangle. Formally, change the color characters to col for these points (u, v) that $x_1 \le u \le x_2, y_1 \le v \le y_2$.
3. "Render $x_1\,y_1\,x_2\,y_2$", which means to render the image of given region. Formally, print the color characters for these points (u, v) that $x_1 \le u \le x_2, y_1 \le v \le y_2$.

But now, Little G is busy replying clarifications, so could you help him and be the painter?

Input

The first line contains one integers n $(1 \le n \le 2000)$, denoting the number of operations.

Following n lines each contains one operation, which is in one of the following three types:

1. "Circle $x\,y\,r\,col\,(0 \le |x|, |y|, r \le 10^9)$", which means to draw a circle. Formally, change the color characters to col for these points (u, v) that $(u - x)^2 + (v - y)^2 \le r^2$.
2. "Rectangle $x_1\,y_1\,x_2\,y_2\,col\,(-10^9 \le x_1 \le x_2 \le 10^9, -10^9 \le y_1 \le y_2 \le 10^9)$", which means to draw a rectangle. Formally, change the color characters to col for these points (u, v) that $x_1 \le u \le x_2, y_1 \le v \le y_2$.
3. "Render $x_1\,y_1\,x_2\,y_2\,(-10^9 \le x_1 \le x_2 \le 10^9, -10^9 \le y_1 \le y_2 \le 10^9)$", which means to render the image of given region. Formally, print the color characters for these points (u, v) that $x_1 \le u \le x_2, y_1 \le v \le y_2$.

It is guaranteed that all of the $x, y, r, x_1, y_1, x_2, y_2$ above are integers.

It is guaranteed that the sum of the rendering region areas(which equal $(x_2 - x_1 + 1) \times (y_2 - y_1 + 1)$) doesn't exceed 10^4, and that col denotes visible characters, whose ASCII codes are between 33 and 126.

Output

For each rendering operation "Render $x_1\,y_1\,x_2\,y_2$", print $y_2 - y_1 + 1$ lines each containing one string of length $x_2 - x_1 + 1$, denoting the region image(from row y_2 to row y_1).

Example

standard input	standard output
7 Circle 0 0 5 * Circle -2 2 1 @ Circle 2 2 1 @ Rectangle 0 -1 0 0 ^ Rectangle -2 -2 2 -2 _ Render -5 -5 5 5 Render -1 0 1 2	.....*..... ..*******.. .**@***@**. .*@@@*@@@*. .**@***@**. *****^***** .****^****. .**_____**. .*********. ..*******.. .....*..... @*@ *** *^*