Problem A. Aftermath

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Once upon a time, you had a nice positive integer n.

Since you like division, you quickly found all its positive integer divisors.

Not being a mean guy, you calculated a — the arithmetic mean of divisors of n. Surprisingly, this number turned out to be an integer.

Some time passed, and you calculated h — the harmonic mean of divisors of n. Even more surprisingly, this number turned out to be an integer, too!

Unfortunately, your memory let you down, and now you remember a and h but don't remember n. However, you remember that n didn't exceed 10^{15}.

Your muse suggested to bring good old times back and restore any value of n matching your records.

Input

The first line of the input contains a single positive integer a.

The second line of the input contains a single positive integer h.

It's guaranteed that there exists a positive integer $n \le 10^{15}$ such that the arithmetic mean of divisors of n is equal to a, which the harmonic mean of divisors of n is equal to h.

Output

Output any positive integer n not exceeding 10^{15} which doesn't contradict the given information.

Example

standard input	standard output
3 2	6

Note

The *arithmetic mean* is the sum of a collection of numbers divided by the number of numbers in the collection. For example, the arithmetic mean of 1, 2, 3 and 6 is equal to $\frac{1+2+3+6}{4} = 3$.

The *harmonic mean* is the reciprocal of the arithmetic mean of the reciprocals of numbers in the collection. For example, the harmonic mean of 1, 2, 3 and 6 is equal to $\left(\frac{1^{-1}+2^{-1}+3^{-1}+6^{-1}}{4}\right)^{-1} = 2$.

Thus, in the first example test case, $n = 6$ satisfies the requirements since its divisors are 1, 2, 3 and 6.

Problem B. Believer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Do you believe in dragons? Imagine that one of them wakes you up at night and asks the following:

Let's consider sequences of positive integers $a = \langle a_1, a_2, \ldots, a_k \rangle$.

Let $f(a, x)$ be the number of occurrences of x in a. For example, $f(\langle 1, 4, 1, 1 \rangle, 1) = 3$.

Let $c(y)$ be the number of ones in the binary expansion of y. For example, $c(13) = c(1101_2) = 3$.

Let $b(a) = \sum\limits_{i \in a} c(f(a, i))$. For example, $b(\langle 1, 4, 1, 1 \rangle) = c(3) + c(1) = 2 + 1 = 3$.

For the given value of n, find the maximum value of $b(a)$ over all sequences with $\sum\limits_{i=1}^{k} a_i = n$.

What would you answer?

Input

The first line of the input contains a single integer t ($1 \le t \le 10^3$) — the number of test cases.

Each of the next t lines contains a single integer n ($1 \le n \le 10^{18}$).

Output

For each test case in order of input, output a single integer — the answer to the problem.

Example

standard input	standard output
2 7 42	3 10

Note

In the first example test case, one possible sequence with $b(a) = 3$ is $a = \langle 1, 4, 1, 1 \rangle$.

Problem C. Chalk Outline (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Your friend Grace has an assignment. She has to draw a simple polygon. It must have exactly n vertices. It must not intersect or touch itself. No three consecutive vertices of the polygon must be collinear. Easy, right?

There is one more small restriction, though.

A *diagonal* of a polygon is a line segment connecting two non-neighboring vertices. We'll call a diagonal *internal* if every point lying on the diagonal (excluding vertices) lies **strictly inside** the polygon.

The number of internal diagonals of the polygon must be equal to k.

Grace has been trying to solve this problem for three days with no success. Now you wonder whether it's possible to fulfill the conditions at all.

Input

The only line of the input contains two integers n and k ($4 \le n \le 100$; $0 \le k \le \frac{n(n-3)}{2}$).

Output

If it's impossible to draw a polygon satisfying the requirements, output a single word "`No`". Otherwise, output "`Yes`".

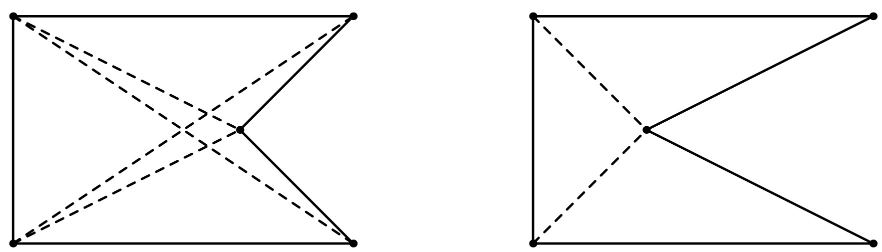
Examples

standard input	standard output
`5 4`	`Yes`
`5 2`	`Yes`
`4 0`	`No`

Note

Below are possible polygons for the first and the second example test cases, respectively. Solid line segments represent polygon edges, and dashed line segments represent internal diagonals.

In the third example test case, any polygon with four vertices has at least one internal diagonal, so the answer is "`No`".



Problem D. Do I Wanna Know? (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You are in charge of organizing the new edition of Arctic Competition for Monkeys (ACM). There are three monkeys taking part in this competition, numbered from 1 to 3. Every two monkeys participate in a separate contest with one problem against each other. There are no ties.

You know the results of all three contests. These results are described by three integers a, b and c. a denotes the result of the contest between monkeys 1 and 2: if $a = 1$, then monkey 1 defeated monkey 2, otherwise $a = 2$. Similarly, b is the number of the monkey who won the contest between monkeys 1 and 3, and c is the number of the monkey who won the contest between monkeys 2 and 3.

Find out if there is a monkey who defeated the other two monkeys.

Input

The only line of the input contains three integers a, b and c ($a = 1$ or $a = 2$; $b = 1$ or $b = 3$; $c = 2$ or $c = 3$).

Output

If there is a monkey who defeated the other two monkeys, output "`Yes`". Otherwise, output "`No`".

Examples

standard input	standard output
`2 1 2`	`Yes`
`1 3 2`	`No`

Note

In the first example test case, monkey 2 defeated both monkeys 1 and 3, and the answer is "`Yes`".

In the second example test case, monkey 1 defeated monkey 2, monkey 2 defeated monkey 3, and monkey 3 defeated monkey 1. There is no monkey who defeated both opponents.

Problem E. Exit Song (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Your favorite singer is giving a farewell concert soon, and you just can't miss this.

The concert will be held in a hall which has n rows, numbered from 0 to $n-1$, with m seats in each row, consecutively numbered from 0 to $m-1$.

Unfortunately, k seats are already unavailable for reservation. These seats are given by pairs (r_1, s_1), (r_2, s_2), ..., (r_k, s_k). For every i from 1 to k, the ticket for seat s_i in row r_i is gone.

You are definitely coming to the concert, but you have no idea if any of your friends would like to join. You are considering all options to buy tickets for several (at least one) consecutive seats in the same row. How many such options do you have?

Input

The first line of the input contains three integers n, m and k ($1 \le n, m \le 10^5$; $1 \le k \le \min(n \cdot m, 10^5)$) — the dimensions of the concert hall and the number of reserved seats, respectively.

The second line of the input contains three integers r_1, a_r and b_r ($0 \le r_1, a_r, b_r < n$).

The third line of the input contains three integers s_1, a_s and b_s ($0 \le s_1, a_s, b_s < m$).

As the input could be quite large, it's encoded in the following way: the values of r_1 and s_1 are given, and for every i from 2 to k the values of r_i and s_i can be found using the following formulae:

$$r_i = (r_{i-1} \cdot a_r + b_r) \bmod n;$$
$$s_i = (s_{i-1} \cdot a_s + b_s) \bmod m.$$

All pairs (r_i, s_i) are distinct.

Output

Output a single integer — the number of options to buy tickets for several consecutive seats in the same row.

Examples

standard input	standard output
3 4 3 1 2 0 2 1 1	18
22 13 41 7 12 14 5 8 1	1195

Note

In the first example test case, seats $(1, 2)$, $(2, 3)$ and $(1, 0)$ are occupied. There are 10 options to buy tickets in row 0, 2 options in row 1 and 6 options in row 2. The sum is $10 + 2 + 6 = 18$.

Problem F. Forever and Always

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Consider an abstract voting procedure. For example, this might be voting for the best song of "Bullet for My Valentine" in 2019.

There are n persons taking part in voting, and there are m options to vote for. Every person has formed their own *preference list* which includes some of the options, in order from most preferred to least preferred. Note that some options might be missing from the preference list — such options are not just little preferred, but unacceptable.

Voting is conducted in *iterations*.

In the first iteration, every person votes for the first option on their preference list. The number of votes for every option is counted and announced to everyone.

In every subsequent iteration, every person intends to vote for the option on their preference list which has received the most number of votes in the previous iteration. If there are several such options, the one that comes earlier on the preference list is chosen.

Before every iteration, it's asked whether anyone is going to vote differently from the previous iteration. If this is not the case, the iteration is not conducted, and the results of the last iteration are declared to be the final vote results. Otherwise, voting takes place, and similarly to the first iteration, the number of votes for every option is counted and announced to everyone again. Note that votes of the previous iterations become ignored.

This sort of voting procedure looks very cumbersome to you, and, most importantly, it looks like it may take forever to find out the results! To prove your point, propose values of n, m and preference lists such that at least 100 iterations of voting will be conducted.

Input

The only line of the input contains a single integer p — the required number of iterations.

There are two test cases. In test case 1, $p = 2$. In test case 2, $p = 100$.

Output

Output two integers n and m ($1 \le n \le 10^5$; $1 \le m \le 2 \cdot 10^5$) — the number of persons and the number of options, respectively, followed by n preference lists.

Each preference list must be described by k_i ($1 \le k_i \le m$) — the number of options on the list, followed by k_i distinct integers $a_{i,j}$ ($1 \le a_{i,j} \le m$) — option identifiers on the list, in order from most preferred to least preferred.

The sum of all values of k_i must not exceed $2 \cdot 10^5$.

Example

standard input	standard output

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

Note

Consider the example test case.

In the first iteration everyone votes for the first option on their list. Thus, the first person votes for option 1, the second and the fourth persons vote for option 2, and the third person votes for option 5.

In the second iteration, seeing that option 2 is now prevailing, the first person will change their vote from option 1 to option 2. Everyone else will keep their vote as is. In particular, the third person will keep his vote for option 5 since both option 5 and option 1 have had one vote in the first iteration, but option 5 is earlier on their list.

Finally, the third iteration is not conducted since nobody is willing to change their vote anymore. Two iterations have been conducted, which satisfies $p = 2$.

Problem G. Gate 21 (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You are participating in a ski race. It's rumored that an autograph of Serj Tankian is the grand prize.

Every racer must pass through n gates numbered from 1 to n. The i-th gate consists of several equivalent checkpoints, which can be considered as points on a plane having coordinates (i, j) for all integers j between l_i and r_i, inclusive. It's required to pass through exactly one checkpoint at every gate in increasing order of gate numbers.

Unfortunately, you are very bad at turning on skis. Thus, you would like to prepare a route for yourself which is a straight line passing through a single checkpoint at every gate. How many route options do you have?

Input

The first line of the input contains a single integer n ($2 \le n \le 100$).

Each of the next n lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le 100$).

Output

Output a single integer — the number of valid straight routes you can take.

Example

standard input	standard output
3 1 3 2 3 1 5	6

Note

In the example test case, all possible routes are:

- $(1,1) \to (2,2) \to (3,3)$;
- $(1,2) \to (2,2) \to (3,2)$;
- $(1,3) \to (2,3) \to (3,3)$;
- $(1,2) \to (2,3) \to (3,4)$;
- $(1,3) \to (2,2) \to (3,1)$;
- $(1,1) \to (2,3) \to (3,5)$.

Problem H. Hamilton (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Lin-Manuel is moving along a strip consisting of n cells consecutively numbered from 1 to n.

He starts at cell a and want to finish at cell b. In the process, he wants to visit every cell exactly once.

From any cell x, Lin-Manuel can walk to the neighboring cell on the left, cell $x - 1$ (if it exists), or on the right, cell $x + 1$ (if it exists).

He can also call for his friend, witch Miranda, who will grant him a magic power. With this power, he will be able to fly exactly once from his current cell x to any cell y such that the greatest common divisor of x and y is 1.

Lin-Manuel doesn't want to burden Miranda too much. Thus, he would like to achieve his goal flying as few times as possible.

Help him and find the smallest number of flights required along with the optimal sequence of visiting cells.

Input

The only line of the input contains three integers n, a and b ($2 \le n \le 9$; $1 \le a, b \le n$; $a \ne b$) — the number of cells on the strip, the starting cell, and the finishing cell, respectively.

Output

If it's impossible to achieve the goal with any number of flights, output a single integer -1.

Otherwise, output the smallest number of flights required to travel from cell a to cell b visiting all cells exactly once, followed by n distinct integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le n$) — cell numbers in order of visiting, describing any valid path which needs the smallest possible number of flights. In particular, it must be true that $c_1 = a$ and $c_n = b$.

Examples

standard input	standard output
`5 1 5`	`0` `1 2 3 4 5`
`6 4 5`	`1` `4 3 2 1 6 5`
`7 5 3`	`2` `5 4 7 6 1 2 3`
`4 1 3`	`-1`

Problem I. I've Got Friends (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

British scientists have found out that friendship is very predictable. They claim that two people can become friends if and only if they have at least one common favorite type of food among two favorite types of food of both.

A famous orchestra from Manchester was chosen for a scientific experiment. Each musician was asked to choose exactly two distinct types of food they like the most.

In the initial report, the scientists have published the list of two favorite types of food of each musician.

You wonder how many pairs of musicians can become friends according to the scientific theory.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 100$) — the number of musicians in the experimental group. The musicians are numbered from 1 to n.

Each of the next n lines contains two integers $f_{i,0}$ and $f_{i,1}$ ($1 \leq f_{i,j} \leq 10^4$; $f_{i,0} \neq f_{i,1}$) — identifiers of favorite types of food of the i-th musician. Different integers correspond to different types of food.

Output

Output a single integer — the number of pairs of musicians who can potentially be friends according to the scientific theory.

Example

standard input	standard output
7 58 42 101 202 42 58 303 202 78 7788 202 404 404 101	6

Note

In the first example test case, pairs of musicians who can become friends are $(1,3)$, $(2,4)$, $(2,6)$, $(2,7)$, $(4,6)$ and $(6,7)$.

Problem J. Joke

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

A card game, often called "Fool's Game", is quite popular in Russia. We will describe a modification of this game for two players and a deck of 54 cards (52 standard cards along with two jokers, black and red).

All spades and clubs are assumed to have black color. All hearts and diamonds are assumed to have red color. Jokers have no rank or suit, just color. One suit is declared to be a *trump*.

Initially, both players have six cards. The other 42 cards, in some order, constitute a *deck*.

A game consists of rounds. Before the round each player has several cards, one of the players is *starting*, the other one is *covering*. The starting player starts by laying one or several cards of the same rank down on the table. Jokers can never be laid down by the starting player. The number of cards must not exceed the number of cards the covering player has. The covering player in turn *covers* all the cards with some of her cards, laying them on the table above the uncovered cards. A card can cover another if at least one of the following is true:

- it has the same suit and higher rank (the order of ranks is 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A, where 2 is the lowest rank and A is the highest rank);
- it is a trump and the card to cover is not a trump;
- it is a joker and its color matches the color of the card to cover;
- it is a joker and its color matches the color of the trump suit.

After the cards on the table are all covered, the starting player can *toss* some more cards to be covered. Similarly, jokers can never be tossed by the starting player. The rank of each card tossed must be among the ranks of the cards already on the table at the moment. Now the newly added cards must be covered by the covering player, after that the starting player can toss more cards, and so on. The starting player cannot toss more cards than the covering player has at the moment.

The round ends when either the covering player cannot or does not want to cover all uncovered cards on the table, or when the starting player cannot or does not want to toss more cards. In the first case, when the covering player declares that she does not want to cover all uncovered cards on the table, the starting player is given a chance to toss in more cards which are not jokers. The ranks of the cards tossed must be among the ranks of the cards already on the table. The number of uncovered cards on the table cannot exceed the number of cards that the covering player has at the moment. After that, the covering player loses the round and takes all the cards from the table, adding them to her cards. Starting player keeps her starting role and moves again in the next round.

In the second case, when all cards on the table are covered and the starting player cannot or does not want to toss more cards, the covering player wins the round and the cards on the table are removed from the game. The players' roles for the next round are swapped: the covering player becomes the starting one and vice versa.

Between rounds, if the starting player of the previous round has less than six cards, she draws additional cards from the top of the deck one by one until she has exactly six cards or the deck becomes empty. After that, similarly, if the covering player of the previous round has less than six cards, she draws additional cards from the top of the deck one by one until she has exactly six cards or the deck becomes empty.

If, before the start of a round, one of the players has no cards, and the other one has one or more cards, then the player with no cards wins the game. If both players have no cards, then the game ends in a draw. If both players have at least one card, but all cards of the starting player of the upcoming round are jokers, then the starting player cannot make a move and lay down any cards, the game ends and the covering player of the upcoming round wins the game.

Two players, Johann and Sebastian, are going to play a game by the rules described above. Johann is the starting player of the first round.

Given the trump suit, the cards the players initially have, and the order of the remaining cards in the deck, find out who wins the game if both play optimally. Both players have full information about cards in the game and the order of cards in the deck.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case is described in four lines. The first line contains six card descriptions — the cards of Johann. The second line contains six card descriptions — the cards of Sebastian. The third line contains 42 card descriptions — the cards in the deck, from top to bottom. The fourth line contains a single character — the trump suit.

Each card except jokers is specified by its rank ('2'...'9', 'T' for 10, 'J' for Jack, 'Q' for Queen, 'K' for King, 'A' for Ace) followed by its suit ('S' for spades, 'C' for clubs, 'D' for diamonds, 'H' for hearts). Red joker is specified by two characters, "RJ". Black joker is specified by two characters, "BJ".

All 54 cards in every test case are different.

Output

For each test case, output a single line containing the name of the winner of the game, or "Bach" if the game ends in a draw.

Example

standard input	standard output
2 TC QD 2S TH 4S 3C AS RJ AC 7D 6C BJ 3D 4C 8C AD TD TS 7H JS KD 4H QC 6H 9D 7C 9H JC AH 5H 6S QH KS 5S 5D 3H JD JH 8H QS 2H 4D 5C 9S KH 6D 9C 8D 8S KC 7S 3S 2D 2C S TC 8S JS JD 5C 9C QS 8C 3H 4D 4H 2D QH 7S 7H 3C 2H 7C TD 9H 8D AH 7D QC JH 5D AS 5H 3D JC 2S 6D AC 9D 4C 6S KD 8H 6C 4S RJ KH 3S TS KC KS 5S QD 9S BJ 6H TH AD 2C D	Johann Sebastian

Note

The third line of both example test cases is displayed as several lines. In the official test data, all 42 cards of the deck are described using one line.

Problem K. Kids Aren't Alright (Div. 2 Edition)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

As unlikely as it may seem, a crazy guy on the phone claims to have kidnapped your precious child. You don't really believe him, as all your children (possibly none) are playing in front of you right now, safe and sound. Anyway, you're fairly curious about the situation, so you ask the criminal what he wants for releasing his hostage.

As boring as it may seem, the kidnapper asks for money. Just money. You are about to hang up the phone in disappointment when something peculiar attracts your attention. Your interlocutor is not telling you the exact amount he wants. Instead, he proposes you a riddle.

As ridiculous as it may seem, the riddle is:

"*How many non-empty sets of positive integers exist such that their greatest common divisor is 1, while their least common multiple is m?*".

Then, the abductor tells you that the answer to this riddle, taken modulo 998244353, is the exact amount of money he wants for returning your imaginary offspring.

You're now wondering about the rates at the kidnapping market, since you've been away from this kind of affairs for quite some time. Not that you're going to pay the snatcher a single penny, though.

Input

The only line of the input contains a single integer m ($1 \le m \le 10^5$).

Output

Output a single integer — the amount of money you've been asked for.

Examples

standard input	standard output
6	7
100	322

Note

In the first example test case, all suitable sets are $\{1, 6\}$, $\{2, 3\}$, $\{1, 2, 3\}$, $\{1, 2, 6\}$, $\{1, 3, 6\}$, $\{2, 3, 6\}$, and $\{1, 2, 3, 6\}$.