National Girls' Programming Contest 2022

https://toph.co/c/national-girls-programming-contest-2022

Schedule

The contest will run for **5h0m0s**.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C++20 GCC 12.1, C11 GCC 12.1, C11 GCC 9.2, Java 1.8, PyPy 7.1 (2.7), PyPy 7.1 (3.6), Python 2.7, and Python 3.7 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 10 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

A. Half Measures

The numbers $1, 2, 3, \cdots n$ (each integer from 1 to n once) are written on a board. In one operation, Jesse can choose any number x from the board and replace it by $\lfloor \frac{x}{2} \rfloor$(Half of x rounded down). Jesse wants to minimize the sum of the integers after performing exactly k operations. Help Jesse.

Input

The first line contains $T(1 \le T \le 10^5)$, the number of testcases.

Each of the following T lines contain two space separated integers, n and $k(1 \le n, k \le 10^9)$.

Output

For each testcase, print a single line containing a single integer, the minimum sum after performing exactly k operations.

Samples

Input	Output
3 4 1 4 3 3 10	8 5 0

For the first case, $1, 2, 3, 4$ are written. Jesse picks $x = 3$ and replaces it by $\lfloor \frac{3}{2} \rfloor = 1$. Now the board contains $1, 2, 1, 4$ with sum 8.

For the second case, initially the board contains $1, 2, 3, 4$.

- For the first operation, Jesse chooses $x = 1$, replaces it by 0. The board contains $0, 2, 3, 4$.
- For the second operation, Jesse chooses $x = 4$, replaces it by 2. The board contains $0, 2, 3, 2$.

Input	Output
• For the first operation, Jesse chooses $x = 3$, replaces it by 1. The board contains $0, 2, 1, 2$. For the third case, Jesse can make everything 0.	

B. Angels and Demons

You are walking at midnight along a road that has N checkpoints numbered from 1 to N from left to right. Each checkpoint may be empty, or it may have angels or demons stationed there. If a checkpoint has an angel or demon, they will cast a blessing or curse of power D upon you which will be active till checkpoint x. This blessing or curse will be in effect at all the checkpoints i for $s \leq i \leq x$. If you receive a blessing of power D, your health will increase by D at each checkpoint where it is active. Conversely, if you receive a curse of power D, your health will decrease by D at each checkpoint where it is active.

Formally, if an angel with power D is stationed at checkpoint s and has an endpoint at checkpoint x, then for each checkpoint i $(s \leq i \leq x)$ your health will increase by D. On the other hand if an demon with power D is stationed at checkpoint s and has an endpoint at checkpoint x, then for each checkpoint i $(s \leq i \leq x)$ your health will decrease by D.

If at any checkpoint (including checkpoint N) your health becomes **zero or negative**, you will die. What is the minimum health you need to start with, to survive the walk from checkpoint 1 to N? Please note here, the initial health must always be an **integer** value.

Note that there may be multiple angels and/or demons stationed at some checkpoints, and angels are slightly more powerful than demons. So if both angels and demons are active at a checkpoint, then all the blessings will take effect first, followed by the curses. Please refer to the explanation of the sample for better understanding.

Input

The first line of the input will contain an integer, $T \ (1 \leq T \leq 1000)$ which denotes the number of test cases.

Each of the test cases starts with two integers, $N \ (1 \leq N \leq 10^9)$ and $Q \ (1 \leq Q \leq 2 \times 10^5)$ describing the number of checkpoints and the total number of demons and angels. Then there will be Q lines. Each of the next Q lines will contain four integers separated by spaces in the following format describing the parameters for an angel or a demon:

- $type\ s\ x\ D\ (1 \leq type \leq 2; 1 \leq s \leq x \leq N; 1 \leq D \leq 10^4)$

If $type = 1$ then it is an angel, otherwise it is a demon. $D,\ s,\ x$ describes the power of the blessing/curse and the active region of the blessing/curse.

It is guaranteed that the sum of Q over all test cases $\leq 2 \times 10^5$.

Output

For each test case, output a single integer, the minimum amount of health needed to survive the walk.

Samples

Input	Output
1 10 3 2 1 3 2 1 1 2 2 1 2 3 1	1

Explanation of the sample:

You start with health 1. It can be shown that you cannot start with any integer health value lesser than this and be alive while completing the journey.

At checkpoint 1 first you get a blessing of power 2, your health becomes 3, then you get a curse of power 2 and your health drops to 1.

At checkpoint 2 first you get two blessings of power 1 and 2, your health becomes 1 + 1 + 2 = 4, then you get a curse of power 2, your health becomes 2.

At checkpoint 3, first you get a blessing of power 1, your health becomes 3, then you get a curse of power 2, your health becomes 1.

Then you continue with health 1 to checkpoint 10 to finish your journey. You didn’t die and you made it to the end! Congrats!

C. Strong Passwords

Mitu had been using the password `mitu1997qt` on all her online accounts. And sure enough, her accounts got hacked. She has come crying to you. But you are merely a privacy enthusiast, not a hacker yourself to hack her accounts back. All you can do is suggest her a strong password, so that her new accounts do not get hacked.

By your definition, a *strong* password should be no lesser than 12 characters. And, for ease-of-use, passwords should not be longer than 24 characters. In a strong password, there should be at least 1 lowercase ASCII letter (a,b,..,z), at least 1 uppercase ASCII letter (A,B,..,Z), at least 2 ASCII digits (0,1,..,9) and at least 2 special characters from these six `&!@._%` ASCII special characters. There should be no other characters in the password, other than lowercase and uppercase ASCII letters, ASCII digits and the six mentioned ASCII special characters. There should definitely not be any spaces in the password.

For example, `Mitu@1997` is not a strong password because the length is only 9 and there is only one special character. `&strongpassword!` is also not a strong password because there is no uppercase letter in there, nor any digits.

Will you please suggest Mitu a *strong* password?

Input

There is no input for this problem.

Output

Output a single line containing the password.

The password should be no lesser than 12 characters and no more than 24 characters. And the password must be *strong* as per the definition stated above.

D. Interesting Parenthesis

Parentheses sequence of length $2n$ is the sequence containing **exactly** n opening parentheses «$($» and **exactly** n closing parentheses «$)$»

A parentheses sequence is called balanced if one can turn it into a valid math expression by adding characters «$+$» and «1». For example, sequences «$(())()$», «$()$» and «$(()(()))$» are balanced, while «$)($», «$(()$» and «$(()))($» are not.

A parentheses sequence, s of length $2n$ is called interesting if one can turn it balanced by doing the following operation **at most once**.

- Select two indices $i, j (1 \leq i, j \leq 2n)$ and swap $s[i]$ and $s[j]$.

You have to calculate the number of different interesting parentheses sequence of length $2n$, and print it modulo 998244353.

Input

First line will contain $T (1 \leq T \leq 10^3)$, the number of test cases. Each of the next T lines will contain one integer $n (1 \leq n \leq 10^3)$, half the length of the interesting parentheses sequence.

Output

For each testcase, print one line containing an integer $ans (0 \leq ans < 998244353)$, which is the number of different interesting parentheses sequence modulo 998244353.

Samples

Input	Output
3 1 2 100	2 6 829549001

In the first test case, there are 2 interesting parentheses sequence, these are as follows.

1. $()$
2. $)($

In the second test case, there 6 interesting parentheses sequence, these are as follows.

1. $(())$
2. $()()$
3. $())($
4. $)(()$
5. $)()($
6. $))(($

E. Shortest Path, Yet Again?

Mr O is teaching Algorithms 101 to his students. The class just finished learning BFS and now he wants to test their understanding. So he came up with the idea of telling them to solve the shortest path problem in a grid. In particular, A $n \times m$ grid has n rows and m columns and Mr O has marked some grid cells as "obstacles". His students have to find the shortest distance from a "start" cell to an "end" cell. From a cell one can go left, right, up or down provided one does not go outside the grid and the cell they go to is not an obstacle. Mr O will assign two distinct non-obstacle cells as start and end.

Quite happy with himself, he realized that now he has to make testing data to test his students. In other words, he needs to come up with some grids with a known shortest distance between the start and end cell (The length of a path through the grid is the number of cells on the path). Being a bit lazy, Mr O has hired you to construct such grids. He will tell you the grid dimensions and the required shortest distance, you just need to construct the grid. **Note** that the shortest distance means the count of cells in the shortest path, including the start and end cells.

Input

The first line will contain a positive integer T, meaning there will be T independent testcases.

Then there will be T lines, The i-th line will contain 3 integers n, m, s meaning you have to create a $n \times m$ grid where the shortest distance between start and end cell is exactly s.

- $1 \leq T \leq 10^5$
- $1 \leq n, m \leq 10^5$
- $4 \leq nm \leq 10^5$
- $2 \leq s \leq \frac{nm}{2}$
- The sum of nm over all testscases $\leq 10^6$

Under these constraints, it is always possible to create a required grid.

Output

For each testcase n, m, s, output n lines. The i-th line will have m contiguous characters. The j-th character on the i-th line should be

- `#` if cell (i, j) is an obstacle
- `s` if cell (i, j) is a start cell
- `e` if cell (i, j) is an end cell
- `.` otherwise

Note your output must have exactly one start cell and one end cell. There are no constraints on the number of obstacles or free cells. The shortest distance between the start and end cell must be exactly s. If there are multiple valid solutions, you can output any.

Samples

Input	Output
2 4 4 7 3 5 3	s... .##. .##. ...e .s.e. ....

F. Predict The Frequency

Luis was sleeping. He suddenly woke up with three positive integers N, M and X. He told his younger brother Hansi these three integers. Hansi is too young and weird. He wrote all possible distinct arrays of length N such that only integers from 1 to M occurs in it and none of their frequency is more than X. Note that, any element in an array will be in $[1, M]$, but not all integers from $[1, M]$ may be present in an array.

Luis defines score of an array as the maximum frequency of its elements in it. Can you tell the sum of score of all arrays that Hansi wrote? Print it modulo P.

Frequency of an element in an array is defined as the number of times the element occurs in that array.

Input

First line of the input will have two space-separated integers T $(1 \le T \le 10^5)$, number of testcases and integer $P(10^8 \le P \le 10^9)$. P will be **prime**.

Each of the next T lines will have three space-separated integers N, M and X $(1 \le N, M, X \le 200)$ describing the testcase.

Output

For each testcase, output an integer ans $(0 \le ans < P)$, which is the sum of score of all arrays that Hansi wrote, modulo P.

Samples

<u>Input</u>	<u>Output</u>
3 998244353 2 2 1 2 2 2 2 1 1	2 6 0

In first test, Hansi will write $[1, 2]$ and $[2, 1]$. Score of both arrays is 1. So, the output will be $1 + 1 = 2$.

In second test, Hansi will write $[1, 1]$, $[1, 2]$, $[2, 1]$ and $[2, 2]$. Score of the arrays are 2, 1, 1, 2 respectively. So, the sum will be 6.

In third test, Hansi will not be able to write any array. So, the output will be 0.

G. Alligator Sky

There are N balls in a box. You have two robots Alice and Bob. You will set favorite number for each of them. Both of their favorite numbers are **positive** integer, not exceeding N. Say, Alice's favorite number is set as A and Bob's favorite number is set as B. Now, they move alternatingly. Alice moves first.

On Alice's move, it takes **exactly** A balls from the box. On Bob's move, it takes **exactly** B balls from the box. If someone cannot complete its move, the game stops.

You know your favorite integer is K $(1 \le K \le N)$. That's why, you will set the values of A and B in such a way that $A + B \ge K$ and **the box will not contain any ball in the end**. Remember both of A and B are positive integers, not exceeding N.

What is the maximum number of total moves the robots can perform, if you set the values of A and B satisfying the given conditions? It can be shown that there is at least one way of setting the values of A and B exists satisfying the given conditions.

Input

First line will have one integer T $(1 \le T \le 10^5)$, denoting the number of testcases.

Each of the next T lines will have two integers N $(1 \le N \le 10^9)$ and K $(1 \le K \le N)$, describing the testcase.

Output

Output one line for each testcase, consisting maximum possible number of total moves.

Samples

Input	Output
2 10 3 7 2	7 7

For first testcase, you can set $A = 1$ and $B = 2$. There will be 7 moves in total.

Note that, if you set $A = 2$ and $B = 1$, after 6 moves, there will be 1 ball in the box. Alice cannot perform his move from here and the game ends. So, you cannot set $A = 2$ and $B = 1$ in this case.

You could set $A = 10$ and $B = 10$. That would result in 1 total move. But it is not optimal here.

H. Plantik

It is Argentina vs France in the football world cup final of 2022 in a parallel universe.

You are a supporter of Argentina. 120 minutes of play has ended in a draw. So, penalty shootout will begin. You know the scoring probabilities for each of the 22 players on the field. You also know that both teams would send shooters in an optimal order. And the order must be submitted to the referee **before** the shootout begins. Argentina start first. The rules of penalty shootout are as follows:

1. Teams take shots alternately.
2. For any consecutive 11 shots of a team, a player can shoot exactly once.
3. If both teams have taken equal number of shots and each team has taken at least 5 shots, then the leading team wins. If scores are still tied, they continue shooting until a team wins.

We denote a goal with $\checkmark$ and a miss with $\times$.

The shootout begins. The scorecard after 5 shots per team is:
ARG: $\checkmark\checkmark\checkmark\times\checkmark$
FRA: $\times\checkmark\checkmark\checkmark\checkmark$
This means Argentina missed its 4th shot and France missed its 1st shot. The score is tied at 4-4.

You are losing your sanity from the suspense. You know the first 5 shooters went in optimal order which was submitted before the shootout began. Now you calculate the probability of Argentina winning the world cup. This means you calculate it **after** 5 shots per team was taken and before Argentina's 6th shot. Print the number modulo 998244353.

Formally, let the probability be an irreducible fraction $\frac{x}{y}$. Print the value $x \cdot y^{-1}$mod 998244353. Where y^{-1} is an integer such that $y \cdot y^{-1} \equiv 1$ mod 998244353. Input data for this problem was made such that y^{-1} exists.

A player might need to take multiple shots in the shootout. The outcome of any shot **does not change** his scoring probability.

Examples:
Some possible outcomes of the shootout continuing from the current situation:

ARG: ✓✓✓×✓×
FRA: ×✓✓✓✓✓
After 6 shots per team, France lead. So France wins.

ARG: ✓✓✓×✓ ✓ ✓ ×✓
FRA: ×✓✓✓✓ ✓ ✓ × ×
After 9 shots per team, Argentina lead by 7-6. So Argentina wins.

ARG: ✓✓✓×✓× ×× ✓ ✓✓××✓×× ✓ ✓✓××✓✓ ✓
FRA: ×✓✓✓✓ × × ×✓✓ ✓××✓×× ✓ ✓✓××✓✓×
After 24 shots per team, Argentina lead by 14-13. So Argentina wins.

Input

The input consists of 2 lines, each containing 11 space separated integers.
The i-th integer $A_i (1 \leq A_i < 100)$ on the first line indicates that scoring probability for the i-th player of Argentina is $\frac{A_i}{100}$.
The i-th integer $F_i (1 \leq F_i < 100)$ on the second line indicates that scoring probability for the i-th player of France is $\frac{F_i}{100}$.
Input is **not** necessarily given in the optimal shooting order.

Output

Print a single integer, the probability of Argentina winning modulo 998244353.

Samples

Input	Output
`99 99 99 99 99 99 99 99 99 99 99` `1 1 1 1 1 1 1 1 1 1 1`	`403188064`
The actual probability is $\frac{9801}{9802}$.	

Input	Output
	`471266917`

Input	Output
`41 49 94 58 69 65 98 57 39 7 71` `85 5 4 90 8 46 21 75 84 41 93`	
The actual probability is $\frac{8055795054746322203650663244113381}{12123531088532242843736676673393538}$. Which is roughly equal to 0.6644759679270646.	

Input	Output
`50 50 50 50 50 50 50 50 50 50 50` `50 50 50 50 50 50 50 50 50 50 50`	`499122177`
The probability is $\frac{1}{2}$.	

I. Farewell Gift for Messi

Lionel Messi, the greatest footballer of all time, is going to retire from international football! On Messi's retirement announcement ceremony, his teammates Emiliano Martínez and Rodrigo De Paul decided to gift Messi an array of n **non-negative integers** where each element of the array is **strictly less than** 2^p. Martínez will be happy if the bitwise OR of all elements of the array is equal to x. On the other hand, De Paul will be happy if the bitwise AND of all elements of the array is equal to y.

You have to find the number of possible arrays for which both Martínez and De Paul will be happy to gift it to Messi. Since the number may be very large, print it modulo Messi's favorite number 998244353.

Input

First line of input is a single integer T ($1 \leq T \leq 1500$) denoting number of test cases.

Each of the next T lines describes the input for a single test case which consists of four integers: n ($1 \leq n \leq 50$), p ($1 \leq p \leq 30$), x ($0 \leq x < 2^p$) and y ($0 \leq y < 2^p$).

Output

For each test case, output the number of arrays which satisfies all the conditions stated above. Since the number may be very large, print it modulo 998244353.

Samples

Input	Output
3 3 1 1 0 2 4 11 8 21 12 4044 3980	6 4 2097150

Possible arrays for test 1: $[1, 1, 0], [1, 0, 1], [0, 1, 1], [0, 0, 1], [0, 1, 0], [1, 0, 0]$

Possible arrays for test 2: $[8, 11], [9, 10], [10, 9], [11, 8]$

J. Wizard Duel

Wizard Duel is approaching soon in Hogwarts and the students of the Dueling Club are absolutely thrilled! One more time, they can show off their skills in front of the whole school in a fun, yet sincere competition against each other. It is a competition where in each round, a wizard/witch faces off against another wizard/witch in a friendly battle of witchcraft and wizardry.

Hermione, being the clever girl she is, wants to pick her battles carefully. Each wizard and witch has a fixed power in her eyes. The value of the power of any wizard, she couldn't determine. But what she does know, is the ratio of power between a few pairs of wizards and/or witches. She is wondering, given the information she has, if it is possible to know that in a battle of one wizard vs another, who will win?

Formally, there are N wizards and witches (including Hermione). Each of them has P_i **fixed** powers, but that value is unknown to Hermione. She does know, however, that the powers are **positive** and the ratio of P_i and P_j (i.e. $\frac{p_i}{p_j}$) for M pairs of wizards and witches (i, j).

Hermione has Q queries, each of them asks that in a battle of wizard i and wizard j, is it possible to know who will win given the information she has? If yes, then who wins? Is it a draw instead?

Note that, a wizard i will **win** against another wizard j if the former has **more** power i.e. $P_i > P_j$. And they will draw if they have equal power i.e. $P_i = P_j$.

Input

The first line of input contains an integer $T (1 \leq T \leq 100)$ denoting the number of test cases.

For each test case, the first line contains three integers N, M, Q.

- $2 \leq N \leq 35$.
- $1 \leq M, Q \leq \frac{N(N-1)}{2}$.

M lines follow, each containing four integers i, j, x, y meaning that the ratio of power between wizards i and j is $\frac{x}{y}$ (i.e. $\frac{P_i}{P_j} = \frac{x}{y}$). You can safely assume that $i \neq j$. Furthermore, the values x and y will be co-prime to each other i.e. $gcd(x, y) = 1$. You can also assume that each of the information regarding the ratio of powers will be consistent throughout a test case. No two unordered (i, j) pair will be the same in a test case.

- $1 \leq i, j \leq N$ and $i \neq j$.
- $1 \leq x, y \leq 3$ and $gcd(x, y) = 1$.

Another Q lines follow, each containing two integers a, b describing the queries. For each of these queries, Hermione wants to know who will win in the battle between wizards a and b.

- $1 \leq a, b \leq N$ and $a \neq b$.

Output

For each query, output a string on a line. For a query (a, b), you should output —

- "**win**" if wizard a will win against b.
- "**lose**" if wizard a will lose against b.
- "**draw**" if the wizards have equal power and will draw.
- "**unknown**" if it cannot be decided who will win or draw, based on the information Hermione has.

Please note that, you should print these strings without the quotes. Refer to the sample I/O for clarity.

Samples

Input	Output
2 4 2 3 1 2 2 3 3 2 2 1 1 2 3 1 2 4 2 1 1 1 2 1 1 2 1	lose win unknown draw

Explanation for the Sample I/O

The first three lines in output correspond to the queries of test case 1. The last line corresponds to the query of test case 2.

In test case 1,

- $\frac{P_1}{P_2} = \frac{2}{3}$, thus wizard 1will lose against wizard 2.
- $\frac{P_3}{P_1} = 3$, thus wizard 3 will win against wizard 1.
- there is no information about wizard 4, thus it is unknown who will win between wizards 2 and 4.

In test case 2,

- Wizard 1 and 2 have equal power, thus it will be a draw.