Problem A. Ling

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 1024 megabytes

You have n piles of stones arranged in a row. The i-th pile contains a_i stones (where a_i can be 0). Among all the stones, exactly one stone is special, and you want to find it. Each time, you can choose a number of stones from a pile to be tested, and the test will tell you whether the special stone is among the chosen ones.The stones you choose must come from one pile.

You want to minimize the number of tests needed to identify the special stone in the worst-case scenario.

There are q queries, each giving l and r. For each query, you need to answer the minimum number of tests required to determine the special stone, given that it is within the piles from l to r.

Since the value of a_i can be large, it will be represented as a binary string of length m in the format of a 01 string. The first character of the string is the most significant bit, and the last character is the least significant bit.

Input

The first line contains three positive integers n, m, and q, representing the length of the sequence, the maximum bit length of a_i, and the number of queries, respectively.

The next n lines each contain a binary string of length m representing a_i.

The next q lines each contain two integers l and r, representing the range for the query.

Output

For each query, output the minimum number of tests required to determine the special stone.

Examples

standard input	standard output
5 3 5 100 100 101 001 111 1 3 1 3 3 3 4 4 5 5	4 4 3 0 3
15 8 10 01100110 10101011 01011110 10011100 10100101 01101010 11001010 01000110 11001101 01100011 01100101 10110010 01011101 01000101 11101000 1 14 14 14 13 15 6 11 11 12 15 15 8 15 10 11 7 11 9 14	20 7 9 12 9 8 14 8 11 12

Note

$1 \le n, q \le 5 \times 10^4, 1 \le m \le 1000, 1 \le l \le r \le n.$

Problem B. Long

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You need to cover an $n \times m$ rectangle using several 1×2 or 2×1 dominoes. Each position must be covered exactly once, and the dominoes must not extend outside the rectangle.

Additionally, there may be two types of constraints:

1. The short sides of the dominoes cannot be adjacent, meaning no two dominoes can share a side of length 1.

2. The long sides of the dominoes cannot be adjacent, meaning no two dominoes can share a side of length 2 (even if they only share one edge).

There are T queries, each giving n, m, a, b, representing the size of the rectangle and whether the two constraints exist. When a is 0, the first constraint exists; when a is 1, the first constraint does not exist. When b is 0, the second constraint exists; when b is 1, the second constraint does not exist. For each query, you need to determine if there is a way to cover the entire rectangle.

Input

The first line contains a positive integer T.

The next T lines each contain four integers n, m, a, b.

Output

For each query, output "Yes"or "No"indicating whether there is a way to cover the entire rectangle.

Example

standard input	standard output
5 3 3 1 1 2 4 0 1 1 5 1 0 5 5 0 0 1 2 0 0	No Yes No No Yes

Note

$1 \leq T \leq 10^5, 1 \leq n, m \leq 10^9, 0 \leq a, b \leq 1.$

Problem C. Tou

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

wzj is working on a cloze test. For those who are not familiar with cloze tests, here is an introduction:

A cloze test consists of n questions, each with exactly m options. A sequence of n positive integers between 1 and m is called an answer set.

The standard answer to the cloze test is a specific answer set. The score of an answer set is defined as the number of positions where it matches the standard answer. An answer set is called balanced if each integer from 1 to m appears exactly the same number of times (in this problem, n is guaranteed to be a multiple of m). The standard answer is always balanced.

Here is some information about wzj:

wzj has a clear understanding of his own level in cloze tests. He knows that he has a probability p_i of scoring exactly i. This means he has a probability p_0 of getting all wrong, a probability p_1 of getting exactly 1 correct, ..., and a probability p_n of getting all correct. wzj selects among answers with the same score uniformly at random. That is, wzj's process of completing the cloze test is as follows: he first randomly determines his score according to the probabilities p_i, and then randomly selects one answer set among those that achieve this score.

Given this background:

One day, wzj completed a cloze test and was pleasantly surprised to find that his answer was balanced. Calculate the expected value of his score. Output the result modulo 998244353.

Input

The first line contains two positive integers n and m, indicating the number of questions and the number of options for each question. It is guaranteed that n is a multiple of m.

The second line contains $n+1$ non-negative integers $p_0, p_1, \cdots, p_n$, representing the probability of wzj scoring each possible score. The sum of p_i is guaranteed to be 1 modulo 998244353.

Output

A single integer representing the expected value of wzj's score modulo 998244353.

Examples

standard input	standard output
4 2 0 0 499122177 0 499122177	598946615
6 2 752799823 916109267 344699104 500885563 291014794 599627729 587841133 (In the data, p is in the same line.)	333821851
9 9 926490943 986291687 543042765 643699715 240607358 250532305 505125659 594335210 257420759 43675365 (In the data, p is in the same line.)	635318919

Note

$1 \le n, m \le 2000, 0 \le p_i < 998244353$.

Ensure that the denominator is not a multiple of 998244353.

Problem D. Zi

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Initially, you have a string S and an empty string T. The length of S is n. There are n rounds of operations where, in each round, you remove one character from S and insert it into any position in T.

For example, $S = \text{abc}, T = \text{de}$ can become $S = \text{ac}, T = \text{dbe}$.

Including the initial state, there are $n + 1$ states in total. The cost of each round is the sum of the edit distances between S and T for these $n + 1$ states.

The edit distance between strings S and T is defined as the minimum number of operations required to transform S into T using the following operations:

1. Insert a character into any position in S.

2. Delete a character from any position in S.

3. Replace any character in S with any character.

For a given S, you want to minimize this cost, denoted as $f(S)$. For all strings S of length n from an alphabet of size m, you want to compute the sum of $f(S)$ for all S where $f(S) \leq \text{limit}$, modulo $10^9 + 7$.

Input

The first line contains three positive integers n, m, limit.

The output is a single integer, which is the answer modulo $10^9 + 7$.

Output

A single integer, which is the answer modulo $10^9 + 7$.

Examples

standard input	standard output
`6 3 24`	`4392`
`10000000 10000000 1000000000000000000`	`477911217`

Note

$1 \leq n, m \leq 10^7, 1 \leq \text{limit} \leq 10^{18}$

Problem E. An

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the dream, May and Ray returned to the side of the little jellyfish. They continued to play their unique game just as they used to.

May and Ray are playing the roles of princesses of two countries, each with n knights. The i-th knight of May has a health value of a_i, and the i-th knight of Ray has a health value of b_i. The i-th knight of May stands opposite the i-th knight of Ray.

When a knight's health becomes 0, the knight dies. The game ends when no knights are standing opposite each other.

The two take turns commanding their knights to attack, with May commanding first. During each command, the current player can choose a pair of knights that are still standing opposite each other and order their knight to attack, reducing the opposing knight's health by 1.

May and Ray both care about their knights, so they both aim to maximize the number of their surviving knights. Both princesses are very clever and will always make the optimal decision.

You, the little jellyfish, as the witness of this duel, want to know in advance how many knights May will have surviving in the end.

Input

This problem has multiple test cases. The first line contains a positive integer T representing the number of test cases.

For each test case, the first line contains a positive integer n representing the number of knights on each side.

The second line contains n positive integers $a_1, a_2, \ldots, a_n$ representing the health values of May's knights.

The third line contains n positive integers $b_1, b_2, \ldots, b_n$ representing the health values of Ray's knights.

Output

For each test case, output a single positive integer representing the number of knights May will have surviving in the end.

Example

standard input	standard output
2 2 2 2 2 2 5 19 11 11 12 16 17 20 18 14 13	1 2

Note

$1 \leq T \leq 2000, 1 \leq n \leq 10^5, \sum n \leq 10^5, 1 \leq a_i, b_i \leq 10^9$.

Problem F. Hong

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Construct a string S of length n with the character set $\{0, 1, 2\}$ such that there are exactly k square substrings.

A square substring is a string of the form TT, where T is any non-empty substring. For example, the string aaa has two square substrings.

Input

The first line of input contains two integers n and k.

Output

If there is a solution, output Yes on the first line. On the second line, output n integers representing each character of the string S.

Otherwise, output No on a single line.

Examples

standard input	standard output
6 2	Yes 0 2 0 0 2 0
8 5	Yes 0 0 1 1 0 0 1 1
6 13	No

Note

$0 \leq n, k \leq 10^5$.

Problem G. Dou

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

You are given a rooted tree with the root at node 1. Each edge in the tree can either be a solid edge or a virtual edge. A valid chain decomposition is defined as one where each node has at most one solid edge leading to its children.

Next, we define the operation $access(u)$:

1. For each node x on the chain from u to the root, set all edges leading towards its children to virtual edges.

2. Set all edges on the chain from u to the root to solid edges.

Initially, the entire tree T_0 consists of virtual edges. After n operations, T_i results from performing $access(P_i)$ on T_{i-1}, where P is a permutation of integers from 1 to n.

Now, you are given the status of each edge on T_n. You want to determine how many permutations P exist such that performing operations in order according to P results in T_n.

Additionally, there are q modifications to T_n. Initially, T_n consists entirely of virtual edges. Each modification specifies a node u, and performs the $access(u)$ operation on T_n. The modifications are not independent. After each modification, you need to output how many permutations P satisfy the sequence of operations to eventually obtain T_n. Answers should be given modulo 998244353.

Input

The first line contains two positive integers n, q.

The second line contains $n-1$ integers $fa_2, fa_3, \cdots, fa_n$, where fa_i denotes the parent node of node i.

The next q lines each contain a single integer u, indicating an $access(u)$ operation on T_n.

Output

There are q lines of output, each containing a single integer representing the answer after each modification.

Example

standard input	standard output
8 9 1 2 1 2 5 5 2 6 2 6 8 3 1 1 5 8	5040 1680 5040 1680 1680 280 280 5040 1680

Note

$1 \le n, q \le 2 \times 10^5, 1 \le fa_i, u \le n$.

Problem H. Ru

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There is an undirected graph where each vertex has a unique weight a_i.

Initially, a chess piece is placed on a vertex. In each move, the piece moves to the adjacent vertex with the smallest weight. If all adjacent vertices have weights greater than the current vertex, the movement stops immediately.

You can freely assign weights a_i to the vertices (ensuring they are all unique) and choose the initial position of the chess piece. Your goal is to maximize the number of vertices visited by the chess piece.

Input

The first line contains two positive integers n, m.

The next m lines each contain two positive integers u, v, indicating an edge in the graph.

Output

Output a single integer, which is the maximum number of vertices visited by the chess piece.

Examples

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

Note

$1 \le n \le 40, 1 \le m \le \binom{n}{2}, 1 \le u, v \le n$.

Problem I. Gu

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Given a weighted connected undirected graph with n vertices and m edges, where the edge weights range from 1 to k. The weight of its minimum spanning tree (MST) is defined as the mode of the edge weights. If there are multiple numbers with the same frequency, the mode is considered to be the smallest one. Find the weight of the MST with the minimum possible mode.

Note that there may be duplicate edges or self-loops.

Input

This problem has multiple sets of test data. The first line contains a positive integer T, indicating the number of sets of data.

For each set of data:

The first line contains three positive integers n, m, k.

The next m lines each contain three positive integers u, v, w, indicating an edge from u to v with weight w.

Output

Output one line for each set of data, which is the answer.

Example

standard input	standard output
3 5 5 3 2 1 1 3 1 2 4 3 3 5 2 2 3 4 3 5 7 3 2 1 3 3 2 2 4 1 1 5 1 2 4 5 3 1 5 3 2 5 1 5 4 3 2 1 3 3 2 3 4 1 3 5 2 2	2 1 3

Note

$1 \le n, m \le 5 \times 10^4, 1 \le u, v \le n, 1 \le w \le k, \sum n, \sum m \le 5 \times 10^4, 1 \le k \le 10, T \le 5 \times 10^3$.

Problem J. Xiang

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Find the number of labeled trees that satisfy the following conditions:

1. n vertices.

2. Each vertex has a degree less than or equal to d.

3. There exists a path that includes nodes $1, 2, 3, \cdots, k$. Note that this path may also include other nodes.

Output the result modulo 998244353.

Input

The first line contains three positive integers n, d, k.

Output

Output one integer, which is the answer modulo 998244353.

Examples

standard input	standard output
4 2 2	12
10 3 4	33957360

Note

$1 \leq d < n \leq 500, 1 \leq k \leq n \leq 500$.

Problem K. Si

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

Alice and Bob are playing a number guessing game.

Alice has secretly chosen an integer x. Bob can ask Alice any integer y, and Alice will respond with either Yes or No, indicating whether $x \geq y$. Additionally, Alice keeps track of the cost of each query, which is $|x - y|$. Note that Bob is unaware of the specific cost associated with each query.

Bob knows that x must be one of the integers $a_1, a_2, \cdots, a_n$ that are given in ascending order. To determine x, Bob wants to minimize the worst-case scenario of the sum of query costs.

Please determine the minimum possible sum of query costs in the worst-case scenario.

Input

The first line contains a positive integer n.

The second line contains n positive integers a_i, guaranteed to be sorted in ascending order.

Output

Output one integer, which is the minimum sum of query costs in the worst-case scenario.

Examples

standard input	standard output
4 1 5 6 7	4
10 8 18 20 24 50 53 77 92 100 121	64

Note

$1 \leq n \leq 1000, 1 \leq a_i \leq 10^9$.

Problem L. Zhi

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

The game consists of n levels, and you know in advance the probability of passing each level, given as $\frac{a_1}{100}, \frac{a_2}{100}, \cdots, \frac{a_n}{100}$, where a_i are non-negative integers.

You can perform any number of operations. Each operation allows you to choose an index $i\,(1 \le i < n, a_{i+1} > 0, a_i < 100)$ and decrement a_{i+1} by one while incrementing a_i by one.

After performing operations, you start the game, aiming to maximize the probability of passing all levels.

Output the maximum probability multiplied by 100^n modulo 998244353.

Input

The first line contains a positive integer T, indicating the number of test cases.For each test case:

The first line contains a positive integer n.

The second line contains n non-negative integers a_i.

Output

For each test case, output one non-negative integer, representing the result of the maximum probability multiplied by 100^n modulo 998244353.

Example

standard input	standard output
5 4 1 2 3 4 4 4 3 2 1 2 2 2 1 1 2 2 3	36 24 4 1 6

Note

$1 \le T, n, a_i \le 100$.

Problem M. BuZhi

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

Little Jellyfish had a dream.

She dreamed that there were two identical boxes in front of her, one containing a gem worth c gold coins and the other empty. If she chose the correct box, she would receive the gem; otherwise, she would receive nothing.

Little Jellyfish didn't know which box contained the gem. Suddenly, her friend May appeared before her.

"I know which box the gem is in, but I won't answer your question for free. You need to pay me 1 gold coin to ask. I will tell you the correct box with probability p, and with probability $1-p$, I will tell you the wrong box."

Little Jellyfish hesitated but missed this opportunity after all. Now she misses May a lot and wants to know the maximum expected number of gold coins she can earn if she asks questions optimally.

The dream keeps replaying in Little Jellyfish's mind, but she still hasn't found the answer.

Input

The first line contains a positive integer T indicating the number of test cases.

For each test case, there is a line with three non-negative integers c, a, b. The value of p can be calculated as $p = \frac{a}{b}$.

Output

For each test case, output x/y where $\frac{x}{y}$ represents the answer. Ensure that $\gcd(x, y) = 1$ and y is a positive integer.

Example

standard input	standard output
4 9 1 2 5 4 5 20 9 10 200 4 9	9/2 3/1 710/41 530455/4149

Note

$1 \le T \le 100, 1 \le c \le 1000, 1 \le a \le b \le 10$.

In the first example case, May's answer holds no value, so the optimal strategy is to randomly choose a box.

In the second example case, the optimal strategy involves asking May once and then choosing the box she indicates. The expected number of gold coins earned is $5 \times \frac{4}{5} - 1 = 3$. It can be proven that this is the best strategy, as asking more than once would cost at least 2 gold coins, thereby not exceeding an expected earning of 3 gold coins.