The 2024 ICPC Vietnam
Northern Provincial Programming Contest

Hosted by: HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Date: October 27th, 2024

OVERVIEW

Note: All problems use standard input/output (STDIN/STDOUT)

Problem A
PILE
Time limit: 0.5 seconds

Given a sequence of n piles $\{1, 2, ..., n\}$ on a straight line. The pile i is located at the coordinate i and has height $h[i]$. We need to make a string connection between pile 1 and pile n using some intermediate piles $i[1], i[2], ..., i[k]$ $(1 < i[1] < i[2] < i[3] < ... < i[k] < n)$. (denote $i[0] = 1, i[k+1] = n$). Due to a technical constraint, two consecutive selected piles must have balanced heights and be close together. In particular, for 2 consecutive selected piles $i[j]$ and $i[j+1]$ $(j = 0, 1, ..., k)$, we have constraints:

- $i[j+1] - i[j] \leq D$
- $|h[i[j]] - h[i[j+1]]| \leq H$

Find the subset of intermediate piles satisfying the constraint above such that the sum of heights of the selected intermediates is minimal.

Input

- Line 1: contains 3 positive integers n, D, H $(1 \leq n \leq 10000, 1 \leq D \leq n, 1 \leq H \leq 10000)$
- Line 2: contains n positive integers $h[1], h[2], ..., h[n]$ $(1 \leq h[i] \leq 10000)$

Output

Write the sum of heights of piles selected in the solution, including piles 1 and n (write -1 if no solution is found)

Sample Input	Sample Output
13 4 2 4 1 7 2 6 3 5 1 7 5 3 4 3	13

Explanation

Piles used are 1, 4, 8, 11, 13 and the sum of heights is $4 + 2 + 1 + 3 + 3 = 13$

Problem B
HUNTING DOG
Time limit: 0.5 seconds

A young, poor farmer has n hunting dogs. Each hunting dog has three attributes: weight, intelligence, and market value. One day, the farmer needs to sell some of his hunting dogs to raise money for his wedding. The farmer has found a billionaire who has enough money to buy the dogs. However, the billionaire wants to select the hunting dogs that satisfy the following condition: two hunting dogs p and q can both be chosen if the weight of dog p is greater than or equal to that of dog q, then the intelligence of dog p must be at least equal to that of dog q and vice versa. Help the farmer choose the hunting dogs to sell in a way that will maximize the total money earned.

Input

- Line 1: one positive integer n ($1 \leq n \leq 10000$)
- Line 2 to $(n+1)$: each line consists of 3 integers separated by spaces, representing the weight, intelligence, and utility value of a hunting dog (the values of the integers are in the range $[1, 10000]$)

Output

A single positive integer is the maximum total market value of the selected hunting dogs

Sample Input	Sample Output
7 2300 7 10 2000 13 40 2800 13 40 2100 11 50 2500 6 20 2600 9 15 2000 17 50	90

Problem C
MAXIMUM SUM
Time limit: 6 seconds

Given a matrix A of size $N \times M$, consisting of **uniformly randomly generated integers** within the range $[0, k-1]$, where the rows are numbered from 1 to N, and the columns are numbered from 1 to M.

Let $S(r_1, c_1, r_2, c_2)$ represent the sum of elements within the subrectangle from row r_1 to row r_2 and from column c_1 to column c_2.

$$S(r_1, c_1, r_2, c_2) = \sum_{r_1 \le i \le r_2} \sum_{c_1 \le j \le c_2} A[i][j]$$

.

Find the maximum value of $S(r_1, c_1, r_2, c_2)$ that is divisible by k.

Input

- The first line contains three integers N, M, and k ($1 \le N \times M \le 10^6, 1 \le k \le 10^6$).
- The next n lines each contain m integers in the range $[0, k-1]$.

Output

- Output a single number, which is the largest sum found that is divisible by p. If no rectangle satisfies the condition, output 0.

Sample Input	Sample Output
6 7 5 1 2 0 0 3 3 1 3 0 3 1 0 1 2 1 1 0 2 0 3 2 2 4 1 4 4 0 3 0 2 3 0 2 2 1 0 0 3 0 1 0 4	65

Problem D
COMPLETE TASKS
Time limit: 1 second

Tue is assigned n tasks to complete, numbered from 1 to n. Each day, Tue will work on the tasks that remain unfinished from previous days, following these rules:

- On the first task of the day, Tue can choose any available task and start working on it.
- If task i is chosen, once Tue finishes task i, the next task j that can be selected must satisfy $j > a_i$.
- If no valid task can be selected after task i, Tue must wait until the next day to resume working.

To earn her wage, Tue must complete all n tasks. Therefore, he wants to finish all tasks in the fewest possible number of days. Write a program to determine the minimum number of days required for Tue to complete all tasks, and for each day, provide the order of tasks that need to be completed.

Input

- The first line contains an integer n ($1 \le n \le 3 \times 10^5$).
- The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$).

Output

- The first line outputs the integer d, which is the minimum number of days needed to complete all n tasks.
- The next d lines: on the t-th line, the first number is k_t, followed by k_t numbers representing the tasks completed on day t in order. If there are multiple ways to complete the tasks in the minimum number of days, print any valid solution.

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

Problem E
DIVISIBILITY FACTOR
Time limit: 0.5 seconds

Let the numeric string Σ be a mysterious cipher. Within this sequence, there exists an enigmatic quality, the so-called *divisibility factor*, tethered to a prime number $\mathbb{P}$. This factor is defined as the count of distinct pairs of positions $\{i, j\}$ $(1 \le i \le j \le |\Sigma|)$, where the number formed by the digits in the string Σ between positions i and j, inclusive, is divisible by the prime $\mathbb{P}$.

For example, the divisibility factor of the string $\Sigma =$ `101010` with respect to $\mathbb{P} = 5$ is 12. The pairs $\{i, j\}$ and corresponding substrings are: $\{1,2\} : 10$, $\{1,4\} : 1010$, $\{1,6\} : 101010$, $\{2,2\} : 0$, $\{2,4\} : 010 \equiv 10$, $\{2,6\} : 01010 \equiv 1010$, $\{3,4\} : 10$, $\{3,6\} : 1010 \equiv 10$, $\{4,4\} : 0$, $\{4,6\} : 010 \equiv 10$, $\{5,6\} : 10$ and $\{6,6\} : 0$. Note that a number with leading zeros is considered equal to the corresponding number without leading zeros.

Task: Given a numeric string S and a prime number $\mathbb{P}$, you are required to answer Q queries of the form: find the divisibility factor for the substring of S from position l to position r inclusive.

Input

The first line contains a single prime number $\mathbb{P}$ $(\mathbb{P} \le 10^9 + 7)$.

The second line contains a non-empty numeric string S $(|S| \le 10^5)$.

The third line contains a single positive integer Q, which is the number of queries $(Q \le 10^5)$.

Each of the next Q lines specifies one query and contains two integers l and r, which are the left and right boundaries of the substring whose divisibility factor is of interest $(1 \le l \le r \le |S|)$.

Output

For each query, print a single integer on a separate line that is the divisibility factor of the corresponding substring.

Sample Input	Sample Output
5 101010 5 1 1 1 6 2 2 2 4 3 3	0 12 1 4 0

Problem F
INDEPENDENT LINE
Time limit: 3 seconds

Given a set A consisting of n points on a plane, where the $i-th$ point has coordinates (x_i, y_i). Draw $(n(n-1))/2$ segments between all pairs of points. A line is called independent with respect to (w.r.t) the set A if it does not intersect any of these segments at any point.

Task: Given m lines, determine how many of them are independent w.r.t the set A. The lines are provided in the form of three parameters (a, b, c) representing the equation $ax + by + c = 0$.

Input

The first line contains a positive integer T $(T \leq 15)$, the number of test cases. Each test case is given with the following structure:

- The first line contains a positive integer n $(1 \leq n \leq 10^5)$.
- The next n lines, where the i-th $(1 \leq i \leq n)$ line contains two integers $(x_i, y_i)(|x_i|, |y_i| \leq 10^8)$ specifying the coordinates of the i-th point.
- The next line contains a positive integer m $(1 \leq m \leq 10^5)$.
- The last m lines, where the i-th $(1 \leq i \leq m)$ line contains three integers $a_i, b_i, c_i(|a_i|, |b_i|, |c_i| \leq 10^8)$ representing the equation of the i-th line.

Output

The output consists of T lines. The i-th line contains a single integer, which is the number of lines that are independent w.r.t the set A corresponding to the i-th test case.

Sample Input	Sample Output
1 3 0 1 1 0 -1 0 4 1 -1 0 1 0 3 0 1 0 1 0 -1	1

Explaination: Only the second line $(x+3=0)$ satisfies the conditions to be considered independent $w.r.t$ the set A.

Problem G
NICESTR
Time limit: 0.5 seconds

A binary string is a string that contains only the characters 0 or 1. A binary string is called *nice* if for each character 1 in the string, the number of consecutive 0s from it to the nearest 1 character on the left or to the beginning of the string is equal to the number of consecutive 0s characters from it to the nearest 1 on the right or to the end of the string. That is, for each character 1, the number of consecutive 0s immediately to its left is equal to the number of consecutive 0s immediately to its right. For example, the string 0001000 is a nice string, while the string 001010 is not nice because to the left of the first character 1 there are two 0s and to the right of it there is only one character 0.

Given a binary string, your task is to delete some characters from it to make it a nice string. Print the length of the longest nice string that can be generated by deleting characters from the given string.

Input

- The first line contains an integer n, which is the length of the binary string ($1 \le n \le 500\,000$).
- The second line is a sequence of n characters, either 0 or 1. Make sure this line contains at least one 1 character.

Output

Output a line containing a single integer, the length of the longest possible string.

Sample Input	Sample Output
10 0000010100	7
4 1111	4
7 0101001	5

Note

From the string 0000010100, we can obtain the longest possible string 0001000 by removing the last character 1 and the first two characters 0s. The string 1111 is a nice string itself, so there is no need to delete any characters. From the string 0101001 we can delete the last two characters to get the nice string 01010.

Problem H
PRINTING
Time limit: 0.1 seconds

While printing documents, Dr. Lecter realized that the pages he printed were completely out of order. There are N document pages, numbered from 1 to N. A total of M sheets of paper have been printed, where the i-th sheet contains the contents of two pages a_i and b_i, each page on one side of the paper (the paper can be rotated freely so that either page a_i or b_i can come first). Since printing is very time-consuming and the remaining blank sheets for printing are limited, Dr. Lecter wants to determine the minimum number of additional sheets of paper required to print the necessary pages on both sides of the sheets, so that a subset of these sheets can be arranged into a **usable document**. As this document will only be used by Dr. Lecter, it is sufficient that there exists a sequence of numbers $i_1, i_2, ..., i_N$ such that $i_1 < i_2 < \ldots < i_N$ and the i_j-th page in the document corresponds to page number j $(1 \leq j \leq N)$, making the document usable.

For example, let $N = 4, M = 2$, and there are two printed sheets. The first sheet contains two pages $(1, 4)$, and the second sheet contains two pages $(3, 3)$. In this case, Dr. Lecter only needs one more sheet to print the pages $(1, 2)$. Then, by arranging the sheets in the order $(1, 2), (3, 3), (4, 1)$, he will have a usable document (since the pages $1, 2, 3, 5$ in this document form a complete sequence of pages $1, 2, 3, 4$).

Task: Help Dr. Lecter determine the minimum number of additional sheets of paper needed to create a usable document.

Input

- The first line contains two positive integers N and M $(1 \leq N \leq 400, 0 \leq M \leq 10^5)$.
- The i-th of the next M lines contains two integers a_i and b_i $(1 \leq a_i, b_i \leq N)$.

Output

Print a single integer, which is the minimum number of additional sheets of paper required to create a usable document.

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

Explanation

In the second example, we don't need to use the sheet with the two pages $(2, 2)$. We only need to rotate the sheet with the two pages $(3, 2)$ to rearrange it as $(2, 3)$, and then use one more sheet to print two pages $(1, 1)$ (or $(1, 2)$, $(1, 3)$ are also acceptable) and place this sheet before the one with pages $(2, 3)$ to have a usable document.

In the third example, no additional printing is required. We can simply arrange the existing sheets in the order $(1, 4), (2, 3), (4, 5)$.

Problem I
REPLACEMENT
Time limit: 0.5 seconds

Alice has just developed a language operating on characters that are natural numbers less than Z. She has a set of n transformation rules, each in the form: $x \text{ --> } y_1, y_2, \ldots, y_k$ (with $2 \le x < Z$; $0 \le y_i < Z$). Each rule can be applied multiple times, and the process stops only when a binary sequence is obtained. For example, with the rules:

3 –> 2,3;
3 –> 2,4,6;
2 –> 0,0,1;
4 –> 1;
6 –> 4,0,4

She can transform the sequence 1,2,3,4 as follows:

1,2,3,4 –> 1, 2, **2, 3**, 4 –> 1, 2, 2, 3, **1** –> 1, 2, 2, **2, 4, 6**, 1 –> 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1

Bob has a set of m binary sequences. Help Alice, starting from a single number, find the shortest binary sequence that does not have any subsequence (contiguous) that appears in Bob's set.

Input

- The first line contains three integers: Z, n, m; $(0 \le Z - 2 \le n \le 100;\ 0 \le m \le 50)$.
- Each of the following n lines contains a rule in the format: $x\ k\ y_1\ y_2\ \ldots\ y_k$; The sum of k over all rules is at most 100. For each integer x $(2 \le x < Z)$, there exists at least one rule that starts from x.
- Each of the following m lines contains one of Bob's binary sequences in the format: $k\ b_1\ b_2\ \ldots\ b_k$. The sum of k over all sequences is at most 50.

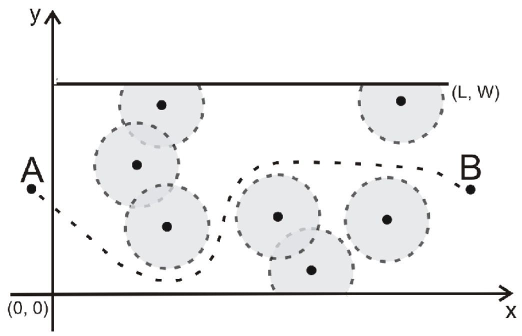
Output

Print $Z - 1$ lines, where the i^{th} line contains the length of the shortest binary sequence that Alice can create starting from the number $a = i + 1$, such that no contiguous subsequence appears in Bob's set. If no finite sequence satisfies the conditions, print 0. The input guarantees that if a finite sequence exists, there exists a sequence with the length not exceeding $2^{64} - 1$.

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

Problem J
SENSORS
Time limit: 1 second

An attacker is trying to enter military zone B from village A. However, village A and military zone B are separated by a valley, which is guarded by a set of N turnable camera sensors. The range of sensing of each sensor is a circle represented by a pair (P, R) where P is the location of the sensor, R is the sensing radius of the sensor. R is set to 100 meters. Thus, depending on the locations of the sensors, it may be possible to pass the valley safely, keeping the distance to the closest sensor strictly larger than 100 meters at any moment.



The attacker asks you to write a program that, given the width W and the length L of the valley and the coordinates of every sensor, first determines whether the attacker can pass the valley unnoticed. If this is impossible, then the attacker wants to know the minimum number of sensors that must be destroyed to pass the valley safely.

Input

The first line contains three integers L, W, and N ($1 \leq W \leq 50000$, $1 \leq L \leq 50000$, $1 \leq N \leq 250$). Each of the following N lines contains a pair of integers X_i and Y_i – the coordinates of i-th sensor ($0 \leq X_i \leq L$, $0 \leq Y_i \leq W$). The coordinates are given in meters relative to the valley: the southwestern corner of the valley has coordinates (0, 0), and the northeastern corner of the valley has coordinates (L, W), as seen in the picture. Note that passing the valley may start at coordinate $(0,\ y_s)$ for any $0 \leq y_s \leq W$ and end at coordinate (L, y_e) for any $0 \leq y_e \leq W$. Neither y_s nor y_e need to be integer.

Output

Print the minimum number of sensors that must be destroyed for the attacker to pass the valley safely. If the attacker can pass without any destruction, the program should print 0.

Sample Input	Sample Output
200 350 5 0 50 200 350 100 200 39 93 123 321	1

Problem K
PALINDROMIC SEQUENCE
Time limit: 1.5 seconds

In a final exam, candidates were asked to solve an Interesting Challenge in Palindromic Counting (ICPC). This challenge involves counting K-*embedded palindromic sequence*, which not only exhibit symmetry but also conform to an important rule: their elements must sum up to a given value, N.

A sequence is *palindromic sequence* if it reads the same forward and backward. Formally, a sequence $S = (s_1, s_2, \ldots, s_n)$ of length N is called a *palindromic sequence* if it satisfies $s_i = s_{n-i+1}$ for all $1 \le i \le n$. For example $(1, 2, 1)$ and $(4, 4)$ are *palindromic sequences* while $(1, 2)$ and $(3, 4, 5)$ are not. A *palindromic sequence* is called K-*embedded palindromic sequence* if it does not contain K at any position.

Given two integers N and K, find the number of K-*embedded palindromic sequences* of positive integers whose elements sum to N. Two sequences are considered different if they have different lengths or if they differ in at least one position. As the result may be large, it should be returned under modulo 998 244 353.

Input

The first line contains an integer T ($1 \le T \le 100$), the number of test cases. Each of the following T lines contains two numbers N and K ($1 \le N \le 10^{18}$, $1 \le K \le 20$) describing a test case.

Output

For each test case, output the residue of the result after dividing 998 244 353 on a single line.

Sample Input	Sample Output
3 6 2 10 3 40 1	4 8 1037630

Problem L
EQPAIR
Time limit: 0.5 seconds

Given a sequence of n integers $a_1, a_2, ..., a_n$. Count the number Q of pairs of 2 indices (i, j) such that $1 \leq i < j \leq n$ and $a_i = a_j$.

Input

- Line 1: contains a positive integer n $(1 \leq n \leq 100000)$
- Line 2: contains n integers $a_1, a_2, ..., a_n$ $(1 \leq a_i \leq 1000000)$

Output

Write the value $Q \bmod 10^9 + 7$

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

Explanation

There are 4 pairs: (1, 4), (1, 6), (2, 3), (4, 6)