The 3rd Guangxi Collegiate Programming Contest

November 21

Problems

A Channel Assignment
B Function Composition
C Landlord
D Longest Road
E Low Power
F Next Number
G Package Delivery
H Photo of Dices
I Prime Partition
J Shift Fenwick Tree
K Subsequence
L Subset Query

Do not open before the contest has started.

Problem A. Channel Assignment

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

Little Q bought n new laptops and he wants to download the data copy from his old laptop at k new laptops within D seconds. For the i-th new laptop, it will take $\frac{a_i}{c_i} + b_i$ seconds to download the copy if c_i Wifi channels are assigned to this laptop. Please tell Little Q how many channels are needed in total.

Formally, Little Q should choose k new laptops and assign Wifi channels carefully such that $\max\{\frac{a_j}{c_j} + b_j\} \leq D$, where c_j are positive integers and j are the indices of selected laptops, and $\sum c_j$ is minimized.

Input

The first line contains a single integer T $(1 \leq T \leq 1000)$, the number of test cases. For each test case:

The first line contains three integers n, k $(1 \leq k \leq n \leq 100\,000)$ and D $(1 \leq D \leq 10^9)$, denoting the number of new laptops, the number of selected laptops and the deadline D.

Each of the following n lines contains two integers a_i and b_i $(1 \leq a_i, b_i \leq 10^9, b_i < D)$, describing each new laptop.

It is guaranteed that the sum of all n is at most $500\,000$.

Output

For each test case, output a single line containing an integer, denoting the number of channels needed in total.

Example

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

Problem B. Function Composition

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

There are n functions $f_1(x), f_2(x), \ldots, f_n(x)$, each of which is from one of the following three types of functions:

- "`1 k`" ($1 \le k \le 100$): $f_i(x) = x + k$.
- "`2 k`" ($1 \le k \le 100$): $f_i(x) = x \times k$.
- "`3 k`" ($1 \le k \le 100$): $f_i(x) = x^k$.

You will be given q queries. In each query, you will be given two integers r_i and x_i ($1 \le r_i \le n$, $1 \le x_i \le 100$). You need to calculate the value of:

$$f_1\left(f_2\left(\ldots\left(f_{r_i-2}\left(f_{r_i-1}\left(f_{r_i}\left(x_i\right)\right)\right)\right)\ldots\right)\right)$$

Input

The first line contains a single integer T ($1 \le T \le 500$), the number of test cases. For each test case:

The first line contains two integers n and q ($1 \le n, q \le 100\,000$), denoting the number of functions and the number of queries.

Each of the following n lines contains two integers t_i and k_i ($1 \le t_i \le 3$, $1 \le k_i \le 100$), describing each function.

Each of the following q lines contains two integers r_i and x_i ($1 \le r_i \le n$, $1 \le x_i \le 100$), describing a query.

It is guaranteed that the sum of all n and the sum of all q are at most $500\,000$.

Output

For each query, print a single line containing an integer denoting the answer. Since the answer can be very large, please output the answer modulo 101 instead.

Example

standard input	standard output
1 4 3 3 2 1 7 2 6 1 5 1 2 2 3 4 5	4 100 45

Note

- For the first query: $ans = 2^2 = 4$.
- For the second query: $ans = (3 + 7)^2 = 100$.
- For the third query: $ans = (((5 + 5) \times 6) + 7)^2 = 4489$, $4489 \bmod 101 = 45$.

Problem C. Landlord

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

There is a landlord selling off plots of land. He is selling it in a grid with $n \times m$ cells, the top-left cell is at $(1, 1)$ while the bottom-right cell is at (n, m). The price of the cell (i, j) which is at row i and column j is $p_{i,j}$ dollars.

Little Q is planning to buy a non-empty rectangle plot of land from the landlord. He will choose the values of x_1, x_2, y_1 and y_2, then buy all the cells (x, y) satisfying $1 \leq x_1 \leq x \leq x_2 \leq n$ and $1 \leq y_1 \leq y \leq y_2 \leq m$.

However, Little Q has only k dollars, can you tell him how many rectangle plots of lands there are whose total price is not larger than k?

Input

The first line contains a single integer T $(1 \leq T \leq 100)$, the number of test cases. For each test case:

The first line contains three integers n, m and k $(1 \leq n \times m \leq 200\,000,\ 1 \leq k \leq 10^{18})$, denoting the size of the grid and the budget.

Each of the following n lines contains m integers, the i-th line contains $p_{i,1}, p_{i,2}, \ldots, p_{i,m}$ $(1 \leq p_{i,j} \leq 10^9)$, denoting the price of each cell.

It is guaranteed that the sum of all $(n \times m)$ is at most $200\,000$.

Output

For each test case, print a single line containing a single integer: the number of possible choices for Little Q.

Example

standard input	standard output
1 2 3 6 1 2 3 2 1 2	16

Problem D. Longest Road

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

There are n cities in Byteland, labeled by 1 to n. The Transport Construction Authority of Byteland is planning to construct $n-1$ bidirectional roads among these cities such that every pair of different cities are connected by these roads directly or indirectly.

There are two engineering companies in Byteland, named company 1 and company 2 for convenience. They have offered m possible candidate roads in total to construct. The i-th candidate road is offered by company c_i, and if it is finally constructed, there will be a road with length w_i connecting the u_i-th city and the v_i-th city directly.

To keep a balance between these two companies, the Transport Construction Authority of Byteland will select exactly k candidate roads from company 1 and select exactly $n-k-1$ roads from company 2. Moreover, it is difficult to construct long roads, so the Transport Construction Authority hopes the longest road to be as short as possible. Please write a program to help the Transport Construction Authority find the best solution.

Input

The first line contains a single integer T ($1 \leq T \leq 500$), the number of test cases. For each test case:

The first line contains three integers n, m and k ($2 \leq n \leq 100\,000$, $0 \leq k \leq n-1 \leq m \leq 200\,000$), denoting the number of cities, the number of candidate roads and the parameter k.

Each of the following m lines contains four integers u_i, v_i, w_i and c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq w_i \leq 10^9$, $1 \leq c_i \leq 2$), describing a candidate road.

It is guaranteed that the sum of all n is at most $100\,000$, and the sum of all m is at most $200\,000$.

Output

For each test case, output a single line containing an integer, denoting the length of the longest road in the best solution.

It is guaranteed that the answer always exists.

Example

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

Problem E. Low Power

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

There are n rectangle radar scanners on the ground. Their sides are all parallel to the coordinate axes. Each scanner covers some grid squares on the ground. The i-th scanner covers all the squares (x, y) satisfying $x_{i,1} \le x \le x_{i,2}$ and $y_{i,1} \le y \le y_{i,2}$.

Recently, the radar system is facing a critical low-power problem. You will be given q queries, in each query several radar scanners are temporarily unavailable, you need to report the number of squares covered by at least one available radar scanners. Note that these radar scanners are temporarily unavailable, they might be well in the next query.

Input

The first line contains a single integer T $(1 \le T \le 5)$, the number of test cases. For each test case:

The first line contains two integers n and q $(1 \le n, q \le 100\,000)$, denoting the number of radar scanners and the number of queries.

Each of the next n lines contains four integers, $x_{i,1}$, $x_{i,2}$, $y_{i,1}$, and $y_{i,2}$ $(1 \le x_{i,1} \le x_{i,2} \le 1000$, $1 \le y_{i,1} \le y_{i,2} \le 1000)$, describing the i-th radar scanner.

Each of the following q lines starts with an integer k_i $(1 \le k_i \le 5)$, followed by k_i distinct integers $p_{i,1}, p_{i,2}, \ldots, p_{i,k_i}$ $(1 \le p_{i,j} \le n)$, denoting the indexes of temporarily unavailable radar scanners in the i-th query.

Output

For each query, print a single line containing an integer, denoting the number of squares covered by at least one available radar scanners.

Example

standard input	standard output
1 3 7 1 4 1 4 3 6 3 6 4 5 2 3 1 1 1 2 1 3 2 1 2 2 1 3 2 2 3 3 1 2 3	18 18 28 4 16 16 0

Problem F. Next Number

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

You are playing a game with Little Q. In this game, two players take turns reporting the smallest positive integer y such that:

- y is not a multiple of 7.
- Digit 7 never occurs in the decimal representation of y.
- $y > x$, where x is the last integer reported by the opponent.

The player with the wrong answer will be punished. Now Little Q reports x correctly, try to write a program to find the answer!

Input

The first line contains a single integer T $(1 \leq T \leq 100)$, the number of test cases. For each test case:

The only line contains an integers x $(1 \leq x \leq 100)$, denoting the last integer reported by Little Q. It is guaranteed that x is not a multiple of 7, and digit 7 never occurs in the decimal representation of x.

Output

For each test case, output a single line containing an integer y, the answer from you.

Example

standard input	standard output
5 2 6 12 13 69	3 8 13 15 80

Problem G. Package Delivery

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

Little Q likes online shopping very much. In the next n days, there will be m packages delivered to the post office in total. Let's label the next n days as day 1, day 2, …, day n respectively. For the i-th package, it will arrive at the post office at day l_i, and the deadline to take it back home is day r_i, which means Little Q can take it back home at day x if and only if $l_i \le x \le r_i$. The package later arrived at the post office will always have a later deadline.

Every time Little Q comes to the post office, he can take at most k packages together back home at the same time. Note that Little Q can go to the post office multiple times during a single day. However, the weather will not be so good on some days. Formally, every time Little Q comes to the post office at day i, he will get c_i unhappiness. Please help Little Q determine how to take these m packages back home such that the total unhappiness is minimized.

Input

The first line contains a single integer T $(1 \le T \le 10)$, the number of test cases. For each test case:

The first line contains three integers n, m and k $(1 \le n \le 4000, 1 \le k \le m \le 4000)$, denoting the number of coming days, the number of packages and the number of packages Little Q can carry at the same time.

The second line contains n integers $c_1, c_2, \ldots, c_n$ $(1 \le c_i \le 100\,000)$, denoting the unhappiness each time for every coming day.

Each of the following m lines contains two integers l_i and r_i $(1 \le l_i \le r_i \le n)$, describing a package. It is guaranteed that the package later arrived at the post office will always have a later deadline.

Output

For each test case, output a single line containing an integer, denoting the minimum possible total unhappiness.

Example

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

Problem H. Photo of Dices

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

Little Q likes unpredictable things like dice. He has thrown the dice for n times. Every time he throws the dice, he will take a 7×7 photo of the dice, so he got n photos in the end and he put these photos together from left to right.

You will be given these photos, please write a program to recognize the numbers shown on the dices and report the sum of them. Note that the number shown on the dice can only be from $\{1, 2, 3, 4, 5, 6\}$.

Input

The first line contains an integer T ($1 \le T \le 50$), the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10$), denoting the number of photos.

Each of the next 7 lines contains a string consists of $7n$ characters, denoting the photos. See the sample input for details, each number from $\{1, 2, 3, 4, 5, 6\}$ has its unique pattern, and the photo will never be rotated or mirrored.

Output

For each test case, print a single line containing an integer, denoting the sum of numbers shown on the dices.

Example

standard input

```
3
1
+-----+
|.....|
|..#..|
|.###.|
|..#..|
|.....|
+-----+
2
+-----++-----+
|....#||....#|
|...#.||.....|
|.....||..#..|
|.#...||.....|
|#....||#....|
+-----++-----+
3
+-----++-----++-----+
|##.##||#...#||##.##|
|##.##||.....||.....|
|.....||..#..||##.##|
|##.##||.....||.....|
|##.##||#...#||##.##|
+-----++-----++-----+
```

standard output

```
1
5
15
```

Problem I. Prime Partition

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

You are given two positive integers n and m, your task is to find the number of ways to split n into several prime numbers $p_1, p_2, \ldots, p_k$ such that $2 \le p_1 < p_2 < \cdots < p_k \le m$ and $p_1 + p_2 + \cdots + p_k = n$. For example $10 = 2 + 3 + 5 = 3 + 7$. Note that when n is prime, $n = p_1$ is also valid. Here, a number x is prime if and only if $x > 1$ and x has no divisors other than 1 and x.

Input

The first line contains a single integer T $(1 \le T \le 300\,000)$, the number of test cases. For each test case:

The only line contains two integers n and m $(2 \le n, m \le 300\,000)$.

Output

For each test case, output a single line containing an integer, denoting the number of ways. Since the answer can be very large, please output the answer modulo 2 instead.

Example

standard input	standard output
3 5 3 5 5 10 5	1 0 1

Problem J. Shift Fenwick Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

A Fenwick Tree is a data structure that can efficiently update elements and calculate prefix sums in a table of numbers. When compared with a flat array of numbers, the Fenwick Tree achieves a much better balance between two operations: element update and prefix sum calculation. In a flat array of n numbers, you can either store the elements, or the prefix sums. In the first case, computing prefix sums requires linear time; in the second case, updating the array elements requires linear time (in both cases, the other operation can be performed in constant time). Fenwick Trees allow both operations to be performed in $O(\log n)$ time. This is achieved by representing the numbers as a tree, where the value of each node is the sum of the numbers in that subtree. The tree structure allows operations to be performed using only $O(\log n)$ node accesses.

Little Q learned Fenwick Tree recently, and now he is trying to solve the following problem using Fenwick Tree:

In that problem you will be given an array $a_1, a_2, \ldots, a_n$, and you need to perform q operations, each of which is from one of the following two types:

- "`1 x`" ($1 \leq x \leq n$): Change the value of a_x into $a_x + 1$.
- "`2 x`" ($1 \leq x \leq n$): Report the value of $a_1 + a_2 + \cdots + a_x$.

Little Q implemented the first type of operation like this:

```
void add(int x, int n) { for (int i = x; i <= n; i += i & -i) cnt ++; }
```

And he implemented the second type of operation like this:

```
void ask(int x, int n) { for (int i = x; i; i -= i & -i) cnt ++; }
```

Initially the value of variable *cnt* is zero, it is easy to show that the larger value of *cnt* is, the slower Little Q's program will be. After several TLE (Time Limit Exceeded) submissions, Little Q decides to make his program faster by shifting the array for k units.

So now Little Q implements the first type of operation like this:

```
void add(int x, int n) { for (int i = x + k; i <= n + k; i += i & -i) cnt ++; }
```

And now he implements the second type of operation like this:

```
void ask(int x, int n) { for (int i = x + k; i; i -= i & -i) cnt ++; }
```

Different values of k will get different values of *cnt*. Please help Little Q calculate the final value of *cnt* after doing all the q operations for each value of $k = 0, 1, 2, \ldots, n$.

Input

The first line contains a single integer T ($1 \leq T \leq 10$), the number of test cases. For each test case:

The first line contains two integers n and q ($1 \leq n, q \leq 2000$), denoting the length of the array and the number of operations.

Each of the following q lines contains two integers op_i and x_i ($1 \leq op_i \leq 2$, $1 \leq x_i \leq n$), describing an operation.

Output

For each test case print $n+1$ lines, where the i-th ($1 \le i \le n+1$) line contains a single integer, the final value of cnt when $k = i - 1$. Note that $cnt = 0$ initially for each k.

Example

standard input	standard output
1 5 3 1 2 2 3 1 1	7 5 5 7 8 5

Problem K. Subsequence

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

In mathematics, a subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. For example, the sequence "`abd`" is a subsequence of "`abcdef`" obtained after removal of elements "`c`", "`e`" and "`f`". The relation of one sequence being the subsequence of another is a preorder.

In this problem, you will be given two extremely long strings A and B, you need to check whether A is a subsequence of B.

To decrease the size of input, A and B will be given in compressed form. For each string, you will be given a list of pairs $(c_1, p_1), (c_2, p_2), \ldots, (c_n, p_n)$, where c_i is always a single lower-case English letter and $1 \le p_i \le 10^9$, denoting the string $c_1^{p_1} + c_2^{p_2} + \cdots + c_n^{p_n}$. For example $(\mathtt{a}, 2), (\mathtt{a}, 3), (\mathtt{b}, 1), (\mathtt{c}, 2)$ denotes "`aaaaabcc`".

Input

The first line contains a single integer T $(1 \le T \le 500)$, the number of test cases. For each test case:

The first line contains an integers n $(1 \le n \le 100\,000)$, denoting the length of the compressed string A.

Each of the following n lines contains a single lower-case English letter c_i and an integer p_i $(1 \le p_i \le 10^9)$, describing the list of compressed string A.

Then in the following line, there is an integer n' $(1 \le n' \le 100\,000)$, denoting the length of the compressed string B.

Each of the following n' lines contains a single lower-case English letter c'_i and an integer p'_i $(1 \le p'_i \le 10^9)$, describing the list of compressed string B.

It is guaranteed that the sum of all n and the sum of all n' are at most $500\,000$.

Output

For each test case, print a single line. If A is a subsequence of B, print `Yes`, otherwise print `No`.

Example

standard input	standard output
2 3 a 1 b 2 c 1 5 a 1 b 1 x 1 b 1 c 2 1 a 9 1 a 8	Yes No

Problem L. Subset Query

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Little Q and Little T are playing a guessing game. In each round, Little T comes up with n positive integers $x_1, x_2, \dots, x_n$ and writes down them on the paper. He should ensure these integers are all in the range $[1, m]$, and he is not allowed to change them during the game. Then Little Q can ask as many questions as he likes. There are two types of questions Little Q can ask:

- "What's the value of the minimum number in the subset S"?
- "What's the value of the maximum number in the subset S"?

For each question, Little T will always answer it honestly in the form "It's in the range $[l, r]$".

Now Little Q has asked q questions, he still can't figure out the value of $x_1, x_2, \dots, x_n$, can you please tell him the minimum possible value of $x_1+x_2+\cdots+x_n$ and the maximum possible value of $x_1+x_2+\cdots+x_n$?

Input

The first line contains a single integer T ($1 \le T \le 5$), the number of test cases. For each test case:

The first line contains three integers n, m and q ($1 \le n \le 8$, $1 \le m \le 1\,000\,000$, $1 \le q \le 500$), denoting the number of numbers, the upper bound of numbers and the number of questions.

Each of the following q lines starts with two integers op_i and k_i ($1 \le op_i \le 2, 1 \le k_i \le n$), followed by k_i distinct integers $p_{i,1}, p_{i,2}, \dots, p_{i,k_i}$ ($1 \le p_{i,j} \le n$), ending with two integers l_i and r_i ($1 \le l_i \le r_i \le m$), describing each question and the answer:

- $op_i = 1$ means the value of the minimum number among the set $\{x_{p_{i,1}}, x_{p_{i,2}}, \dots, x_{p_{i,k_i}}\}$ is in the range $[l_i, r_i]$.
- $op_i = 2$ means the value of the maximum number among the set $\{x_{p_{i,1}}, x_{p_{i,2}}, \dots, x_{p_{i,k_i}}\}$ is in the range $[l_i, r_i]$.

Output

For each test case, output a single line containing two integers $\min\{sum\}$ and $\max\{sum\}$, denoting the minimum possible value of $x_1 + x_2 + \cdots + x_n$ and the maximum possible value of $x_1 + x_2 + \cdots + x_n$.

It is guaranteed that the answer always exists.

Example

standard input	standard output
2 3 10 1 1 1 1 5 5 5 100 2 1 3 2 3 4 7 9 2 3 2 3 4 7 9	7 25 23 227