15-295 Fall 2018 #4 Selection Round 1

A. The Sverdlovsk incident

2.0 s, 64 MB

During the years of the Soviet Union, the name of the city of Yekaterinburg was Sverdlovsk, in honor to the Bolshevik Iakov Sverdlov, son of a Jewish artisan, that was an excellent orator and was one of the main protagonists beside Lenin in the revolution in October 1905. He was considered honest, energetic and hard-working, and was respected by all groups of the party. He died at 34. The city got back its original name in 1991 by the initiative of Boris Yeltsin, the first Russian president, born in that city.

In April 2nd, 1979, while the city was still called Sverdlovsk, there was an anthrax leaking from a military factory in the city. This incident is often called "Biological Chernobyl", and caused approximately 100 deaths, despite the exact number of victims and contaminated are still unknown. The Soviet Union denied for years the real causes of the incident and every victim record disappeared, since they could uncover serious violations of the Biological Weapons Convention.

The Soviet authorities had to use highly sophisticated decontamination procedures, specially in rural areas. Each rectangular area of dimensions N by M was divided in $N \times M$ square sectors of one squared meter. These sectors were identified by the coordinates of their centers, numbered from west to ease and from south to north from $(1, 1)$.

Each sector was considered decontaminated if it was covered by at least K health agents. Each agent was capable of covering a circular area. The radius of this area depended on the equipment used and the experience of the agent. Your task is to determine how many of those sectors are considered decontaminated, that is, covered by at least K agents. A sector is considered to be covered if its center is in an area covered by some agent.

Input

On the first line, a single integer T, the number of test cases.

The first line of each test case contains two integers, N and M, the dimension of the rectangular area mentioned in the statement. The second line contains the number of agents, C, and the number K.

The c-th of the next C lines contains the description of the c-th agent, that is, three integers X_c, Y_c and R_c, meaning (X_c, Y_c) is the center of the circular area covered by that agent, and the radius is R_c.

Limits

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^3$
- $1 \leq M \leq 10^5$
- $1 \leq K \leq C \leq 10^3$
- $1 \leq X_c \leq N$
- $1 \leq Y_c \leq M$
- $0 \leq R_c \leq 10^8$

Output

For each test case, print the number of decontaminated sectors.

input
2 10 10 2 1 3 3 2 8 8 2 15 15 6 2 4 4 2 5 5 1 6 6 3 7 7 2 10 10 0 11 10 1
output
26 20

More precisely, a sector with center (x_s, y_s) is covered by an agent positioned at (x_a, y_a) capable of decontaminating an area with radius r if

$$(x_s - x_a)^2 + (y_s - y_a)^2 \leq r^2.$$

B. Knights (30 second time limit)

Magnus is the youngest chess grandmaster ever. He loves chess so much that he has decided to decorate his home with chess pieces. To decorate his long corridor, he has decided to use knights. His corridor is covered by beautiful marble squares of M rows and N columns. He wants to put the knights pieces into some (or possibly none) of these cells. Each cell will contain at most one knight.

The special thing in his arrangement is no pair of knights can attack each other. (Two knights can attack each other if they are placed in opposite corner cells of a 2 by 3 rectangle).

Given the dimension of the long corridor, your task is to calculate how many ways Magnus can arrange his knight pieces. Two arrangements are considered different if there exists a cell which contains a knight in the first arrangement but not in the other arrangement.

Input

The first line of the input is the number of test cases T ($T \leq 10$). Then T test cases follow. Each test case consists of 2 numbers: M, the number of rows, and N, the number of columns. ($1 \leq M \leq 4, 1 \leq N \leq 10^9$).

Output

For each test case, print the number of possible ways modulo 1,000,000,009 ($10^9 + 9$)

Sample Input	Sample Output
4 1 2 2 2 3 2 4 31415926	4 16 36 413011760

C. Please, go first

You are currently on a skiing trip with a group of friends. In general, it is going well: you enjoy the skiing during the day and, of course, the après-skiing during the night. However, there is one nuisance: the skiing lift. As always, it is too small, and can only serve one person every 5 seconds. To make matters worse, you and your friends generally don't arrive simultaneously at the lift, which means that you spend time waiting at the bottom of the mountain for the lift and at the top again for your friends.

The waiting at the top is especially inefficient. In fact, you realize that if your friends haven't arrived yet, you might as well let other people pass you in the queue. For you, it makes no difference, since otherwise you'd be waiting at the top. On the other hand, your actions might save them time if their friends have already arrived and are currently waiting for them at the top.

You are wondering how much time would be saved if everybody adopts this nice attitude. You have carefully observed the queue for a while and noticed which persons form groups of friends. Suppose someone lets another pass if doing this doesn't change his own total waiting time, but saves time for the other person. Do this over and over again until it can't be done anymore. How much time will this save, in total?

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with an integer n ($1 \leq n \leq 25\,000$): the number of people in the line for the lift.
- one line with n alphanumeric characters (uppercase and lowercase letters and numbers): the queue. The first person in this line corresponds to the person at the head of the queue. Equal characters correspond to persons from the same group of friends.

Output

Per test case:

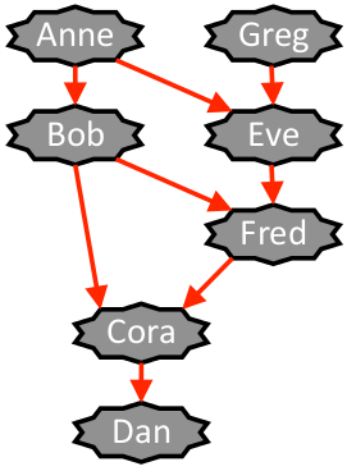
- one line with an integer: the time saved, in seconds.

Sample in- and output

Input	Output
2 6 AABABB 10 Ab9AAb2bC2	15 45

D. Promotions

The Fair Inc. administration decided to promote the best employees and limited the number of promotions to a fixed interval $[A, B]$. The directors compared the employees' performance and their evaluations resulted in a consistent precedence relation among employees, which has to be respected by promotions. This means that, for every pair of employees x and y, if x outperformed y, then y may be promoted only if x is promoted.



In order to understand whether the data collected so far is enough for ensuring fairness, the executive chairman wants to know:

- How many employees will certainly be promoted in the interval endpoints (i.e., if the number of promotions is A and if the number of promotions is B).
- How many employees have no possibility of being promoted (even if the number of promotions is B).

Consider the example depicted in the figure. There are seven employees and eight precedence rules. An arrow from an employee x to an employee y means that x outperformed y. The number of promotions is limited to the interval $[3, 4]$. Therefore:

- If there are only three promotions, the promoted employees must be:
  - either Anne, Bob and Greg,
  - or Anne, Eve and Greg.

  In this case, two employees (Anne and Greg) will certainly be promoted. Notice that, with the current information, Bob and Eve may or may not win a promotion.
- If there are four promotions, the promoted employees have to be:
  - Anne, Bob, Eve and Greg.

  So, with four promotions, four employees (Anne, Bob, Eve and Greg) will certainly be promoted and three employees (Cora, Dan and Fred) have no possibility of being promoted.

Task

Write a program that, given the interval of the number of promotions, the set of employees and the precedence relation among them, computes, for each of the interval endpoints, the number of employees that will certainly be promoted, and the number of employees that have no possibility of being promoted.

The precedence relation is consistent in the sense that, if an employee x outperformed an employee y, y did not outperform (directly or indirectly) x.

Input

The first line of the input has four space separated integers: A, B, E, and P. A and B are the interval endpoints, E is the number of employees and P is the number of precedence rules. Employees are identified by integers, ranging from 0 to $E - 1$.

Each of the following P lines contains two distinct space separated integers, x and y, which indicate that employee x outperformed employee y.

Constraints

$1 \leq A < B < E$	Interval endpoints.
$2 \leq E \leq 5\,000$	Number of employees.
$1 \leq P \leq 20\,000$	Number of precedence rules.

Output

The output consists of three lines. The first line contains the number of employees that will certainly be promoted if there are A promotions. The second line contains the number of employees that will certainly be promoted if there are B promotions. The third line contains the number of employees that have no possibility of being promoted (even if there are B promotions).

Sample Input

```
3 4 7 8
0 4
1 2
1 5
5 2
6 4
0 1
2 3
4 5
```

Sample Output

```
2
4
3
```

Problem **E.** – Complete Naebbirac's sequence

Naebbirac is a young and easy-to-get-bored sailor. He likes sequences of integers and to come up with ways to classify them. Naebbirac says that a sequence is *complete* for a chosen integer K, if the sequence only contains integers between 1 and K, and each integer between 1 and K appears the same number of times.

Based on that, Naebbirac created a game to entertain himself and his peers, when the waters calm down and there's not much they can do to spend their time in the middle of the ocean.

First he chooses a positive integer K and then he uses chalk to draw on the deck a sequence S having N integers between 1 and K. After that he challenges one of his peers. The goal of the challenged peer is to turn the sequence S into a *complete* sequence by performing exactly one of the following three possible operations:

- "`-x`": remove one occurrence of integer x from S;
- "`+x`": add a new integer with value x in S; or
- "`-x +y`": replace one occurrence of integer x from S by an integer with value y.

Naebbirac is quite smart. He never writes a sequence that is already *complete* and often the written integers don't follow a pattern, making it quite hard to find an operation that solves the puzzle. One of your friends, that usually sails with Naebbirac, is tired of always losing the game. Are you able to help your friend and create a computer program that can find a solution to Naebbirac's game before they go on their next trip?

Input

The first line contains two integers K ($3 \leq K \leq 1000$) and N ($1 \leq N \leq 10^4$), indicating respectively the integer that Naebbirac chooses at the beginning of the game, and the length of the sequence written on the deck. The second line contains N integers $S_1, S_2, \ldots, S_N$ ($1 \leq S_i \leq K$ for $i = 1, 2, \ldots, N$) representing the written sequence; you can safely assume that the sequence is not *complete*.

Output

Output a single line with the description of the operation that allows your friend to win the game or an "`*`" (asterisk) if there is no way to win. The description of the operation must follow the format shown on the statement, i.e. "`-x`", "`+x`" or "`-x +y`".

Sample input 1	**Sample output 1**
`3 5` `1 3 2 3 1`	`+2`

Sample input 2	**Sample output 2**
`3 7` `1 2 3 3 3 2 1`	`-3`

Sample input 3	**Sample output 3**
`3 6` `3 1 2 1 3 1`	`-1 +2`

Sample input 4	**Sample output 4**
`3 6` `2 3 2 2 2 1`	`*`

Problem F.

Graphics Design

Time limit: 4 seconds

Source: Pexels

You are in a graphics design course. You and the rest of your class need to finish your final projects. The instructor has several items (cameras, camcorders and super computers) that the students will use to finish their projects. Each project consists of several subprojects that must be completed in order. These subprojects may or may not be different for each student. Each subproject may need a camera, a camcorder and/or a super computer (8 possibilities). The instructor has given each subproject a number, its priority.

A subproject is eligible to be started if all items that are needed for that subproject are available and the student has fully completed all of their subprojects before this one. Out of all eligible subprojects, the one with the highest priority will be started first. This student will borrow the items needed to complete the subproject and then return them after the subproject is finished.

If, after a subproject has started and the items have been borrowed, there still remain eligible subprojects, the eligible subproject with the highest priority is started. This means that it is possible that several subprojects are started at the same time (see Sample Input 1). It is also possible that several students are waiting for an item to be returned, and so there are no eligible subprojects at a specific moment in time (see Sample Input 2). Note that the order of each student's subprojects is fixed. That is, they must complete their first subproject first, then their second subproject, and so on, even if the priorities of their subprojects are not in decreasing order (see Sample Input 3).

If a student starts a subproject at time x and it takes t time units to complete, then the subproject is considered finished at time $x + t$ and the student must return the borrowed items. This means that the items borrowed for that subproject are available to be borrowed again at time $x + t$ and that the student is able to start their next subproject at time $x + t$ (subject to availability of the items and the subproject's priority, as specified above). Each camera, camcorder and super computer can only be used by one student at a time. Students must do the subprojects in order, even if they have the items needed for a later subproject but not the current one.

For each student, you are to compute the time that they finish their last subproject.

Input

The first line of input contains a single integer n ($1 \leq n \leq 1\,000$), which is the number of students in the class. The second line contains 3 integers a ($1 \leq a \leq 1\,000$), which is the number of cameras available, b ($1 \leq b \leq 1\,000$), which is the number of camcorders available, and c ($1 \leq c \leq 1\,000$), which is the number of super computers available. The third line contains n integers $d_1, \ldots, d_n$ ($1 \leq d_i \leq 250$), which are the number of subprojects that each student needs to complete for their project.

This is followed by d_i lines for each student i in the class, one line for each subproject in the order they must be completed by that student. Each line for a subproject consists of an integer t ($1 \leq t \leq 1\,000\,000$), which is the amount of time it takes to complete the subproject, an integer p ($1 \leq p \leq 1\,000\,000$), which is the priority of the subproject, and between zero and three distinct strings. Each of these strings will be one of `Camera`, `Camcorder` or `Computer`.

All priorities will be distinct.

Output

For each student, display the time that they finish their last subproject, in the same order that the students are given in the input.

Sample Input 1

```
3
1 1 1
1 1 1
4 1 Camera
4 2 Camcorder
4 3 Computer
```

Sample Output 1

```
4 4 4
```

Sample Input 2

```
3
1 1 1
1 1 1
3 3 Computer
4 2 Computer
5 1 Camera Computer
```

Sample Output 2

```
3 7 12
```

Sample Input 3

```
2
1 1 1
2 1
1 1 Computer
1 3 Computer
1 2 Computer
```

Sample Output 3

```
3 1
```

Sample Input 4

```
3
2 2 2
2 1 3
2 3 Camera
5 1 Camera Camcorder
3 2 Camcorder Computer
1 6 Camera Camcorder Computer
1 5 Camera Camcorder Computer
1 4 Camera Camcorder Computer
```

Sample Output 4

```
8 3 3
```