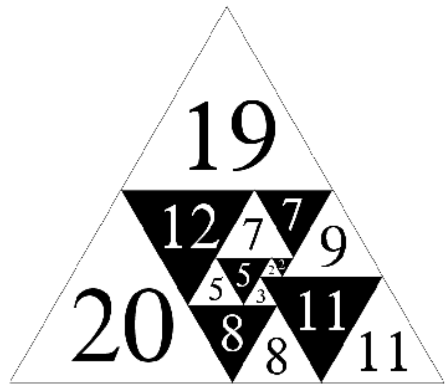
Problem A
Triangulate This!
Time Limit: 2 seconds

Given two positive integers A and B, where $B \leq A$ and B divides A, how many equilateral triangles of size B you need to completely cover the triangle of size A.



Input

The test file starts with an integer $T (T \leq 1000)$, the number of test cases.

Each test case consists of two integers A and B on a line. $(1 \leq B \leq A \leq 1,000,000, B|A)$

Output

For each test case, output the minimum number of equilateral triangles with side B that can completely cover the equilateral triangle with side A.

Sample Input	Sample Output
2 2 1 3 3	4 1

This page is intentionally left blank.

Problem B
Best Buddies
Time Limit: 6 seconds

You have signed up for the annual Calgary Collegiate Programming Contest. Unlike previous years, participants are not choosing their own team members, but are assigned to teams as follows. Consider we list all participants in alphabetic order. Then Team 1 consists of the first three participants in this order. Team 2 consists of the next three participants in this order. Assume that the number N of participants is divisible by 3 so that every team has exactly 3 team members.

Assume that every participant has exactly one first name and exactly one last name, the name is spelled over the alphabet $\{a, \ldots, z\}$, and that no two participants carry the exact same name. The alphabetic order we choose, is to first sort in lexicographically ascending order with respect to last names, and then, when necessary, with respect to first names. So john smith is alphabetized after carla smith, but before bob taylor.

Given the name of a participant, output the names of the participant's two team mates.

Input

The first line of the input contains an integer, N, the number of participants ($3 \leq N \leq 99,999$). Then follows N lines, each containing the first name and last name of the participant. Both first and last name consist of at least 2 and at most 16 lower case characters of English alphabet. The next line contains an integer, Q, the number of queries ($1 \leq Q \leq N$). Then follows Q lines, each containing the first name and last name of the participant. The queried names are guaranteed to exist in the original list.

Output

For each of the Q test cases, output the names of the two team mates in the described order.

Sample Input	Sample Output
12 john smith carla smith bob taylor alice watson sarah taylor ryan singh jessica li kevin wright charles lee megan davies ryan davies julia leung 3 ryan davies sarah taylor kevin wright	megan davies charles lee alice watson kevin wright sarah taylor alice watson

Problem C
Do Not Hex My Numbers!
Time Limit: 3 seconds

Given an integer N and list of D hexadecimal digits, what is the smallest positive integer X, whose representation in base 16 consists only of given digits, such that X is divisible by N?

Input

The first line of the input file starts with the integer T, the number of test cases $(1 \leq T \leq 100)$.

Each test case consists of two lines, in the following format:
$N\ D$
$d_1\ d_2 \ldots d_D$

$N(1 \leq N \leq 200,000)$ and $D(1 \leq D \leq 16)$ are as described in the problem statement (both given here in base 10) and $d_i(1 \leq i \leq D)$ are heximal digits allowed to be used in the result. You can assume that digits are sorted.

Output

For each test case, output the smallest positive number X in base 16 such that X is divisible by N and it contains only digits provided. If there is no such number, output ”no solution” instead.

Sample Input	Sample Output
4 1 3 a b c 2 8 1 3 5 7 9 b d f 1207 3 1 a f 33910 4 0 c e f	a no solution 1aa1aa c0ffee

This page is intentionally left blank.

Problem D
Serves Me Right
Time Limit: 2 seconds

You are in charge of maintaing a web server. There are up to N users that can connect to it and your company is paying for N concurrent licenses (meaning, all N of users can be logged in at the same time).

The economy is taking a hit and your company is looking to cut costs in all areas. Your supervisor would like to know if paying for N licenses is really necessary.

Server logs events in the following way:

- log time when a user connects to the server (user session starts)
- log time when a user disconnects from the server (user session is removed)
- log time when the server is restarted

Note that not all users necessarily log out (they may just close their browser instead) - that is why there is a session timeout value in minutes, at the end of which a user is disconnected regardless of his/her activity at the time. This automatic disconnect is not logged.

When web server is restarted, all users are disconnected and their sessions are removed from the server.

Given a web server log for a day and a user session timeout value, what is the maximum number of users logged in at any given point in time?

Input

The first line of the input file starts with the integer T, the number of web logs to process $(1 \leq T \leq 100)$.

Each case starts with two integers on the first line - E and TO $(1 \leq E, TO \leq 1440)$ Then follow E log entries on each line. They can be in either of these two formats:
TIME SERVER RESTART
or
TIME USER ⟨USER_NAME⟩ ⟨USER_ACTION⟩

TIME is in the format HH:MM (24-hour clock)
USER_NAME is a non-empty string of up to 16 lower case english characters. There are no duplicate user names (i.e. if you see the same user name, it refers to the same user)
USER_ACTION is one of LOG_IN or LOG_OUT

The entries are ordered by time, all times are distinct. There will be no incosistencies in the server log. Not all N users have to be present in the log.

Output

For each test case, output the number of users that are using the web server and the maximum number of users logged in at the same time over the period covered by the log. Output both numbers on the single line, separated by a single space character.

Sample Input	Sample Output
2 4 30 08:30 USER alice LOG_IN 09:00 USER bob LOG_IN 09:30 USER alice LOG_IN 10:00 USER bob LOG_IN 4 45 08:30 USER alice LOG_IN 09:00 USER bob LOG_IN 09:30 USER alice LOG_IN 10:00 USER bob LOG_IN	2 1 2 2

Problem E
Train of Threes
Time Limit: 12 seconds

Threes! is a puzzle game available on iOS, Android, and elsewhere. Its premise is simple, yet provides a deep structure: Repeatedly match adjacent numbers to form larger numbers. Here we will play a simplified version of the game.

The game starts with an array of N numbers. Each number is either 1 or 2 or some number on the form 3×2^i for some integer $i \geq 0$.

We can combine two matching adjacent numbers to form larger numbers. We can combine a 1 adjacent to a 2 into a single 3. We can combine two adjacent 3s into a single 6, we can combine two adjacent 6s into a single 12, and we can in general combine two identical adjacent numbers higher than 3 into a single number that will be their sum. The 1s and 2s are special in that a 1 can only be matched with a 2, and a 2 can only be matched with a 1.

The goal of the game is to form the largest integer possible. The game ends when we can no longer combine two adjacent numbers. Consider we e.g. start with the array $\{12, 24, 6, 1, 2, 3, 12, 3, 3, 6, 1, 2, 2, 1, 24\}$ then the largest number we can form is 48.

Input

The first line of the input contains an integer T, the number of test cases ($1 \leq T \leq 1,000$). Then follows $2T$ lines. The first line of each test case contains an integer N, the size of the array ($1 \leq N \leq 10,000$). Then next line contains N integers, where each integer is 1, 2, or a number on the form 3×2^i for some integer $0 \leq i \leq 11$.

Output

For each of the T test cases, output the largest integer that can be formed.

Sample Input

```
3
5
1 2 2 2 2
6
24 12 6 3 2 1
4
3 3 3 3
```

Sample Output

```
3
48
12
```

This page is intentionally left blank.

Problem F
Roots! Really?
Time Limit: 2 seconds

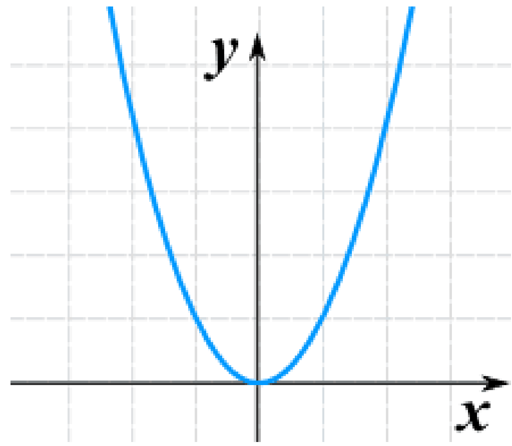
A quadratic equation

$$ax^2 + bx + c = 0$$

has two solutions x_+ and x_-, called roots, which are given by

$$x_\pm = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The two roots may be real or complex, and they may be identical or distinct. Given a quadratic equation and an interval $[s, t]$ (with $s \leq t$), we want to know if the equation has a real root in the interval $[s, t]$. That is, is it the case that $s \leq r \leq t$ where r is any of the roots x_- or x_+?



Input

The first line of the input contains an integer, N, the number of test cases ($1 \leq N \leq 1,000$). Then follows N lines, each containing five integers, a, b, c, s, and t, with $-10^7 \leq a, b, c, s, t \leq 10^7$, $a \neq 0$, and $s \leq t$.

Output

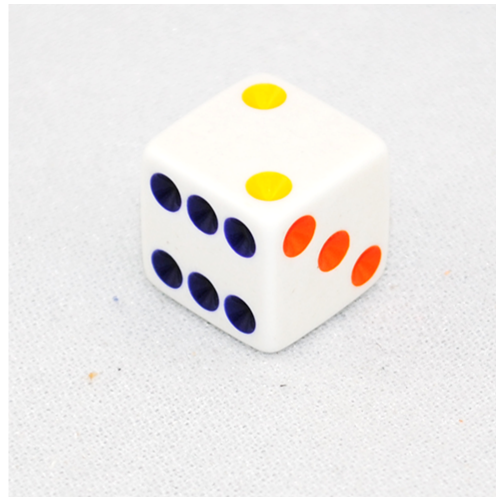
For each of the N test cases, output "Yes" if the equation $ax^2 + bx + c = 0$ has a real root in the interval $[s, t]$. Output "No" otherwise.

Sample Input	Sample Output
3 1 0 0 -1 0 -1 5 -4 2 3 4 4 1 0 100	Yes No No

This page is intentionally left blank.

Problem G
Dyeing Dice
Time Limit: 2 seconds

You have just opened your new die colouring factory, which takes a plain die made out of natural wood and colour its six faces. The factory is of course custom-made and consists of a board of size $N \times N$. Some of the N^2 squares are filled with paint, while other squares contain nothing. The square in the SW corner of the board is numbered $(1, 1)$, the square in the NW corner $(1, N)$, the square in the NE corner (N, N), and the square in the SE corner $(N, 1)$.



We colour the die by rolling it through a pre-programmed path of squares. When the die lands on a square with paint, the face of the die that faces the paint takes the colour of the paint. When the die lands on a square with no paint, nothing happens. The die is coloured once all six faces are coloured, at which point the die is removed from the factory (the remaining squares on the path, if any, are disregarded). The square $(1, 1)$ never contains any paint.

We initially put the die on entry $(1, 1)$ with the number "1" facing up, and the number "2" facing the square $(2, 1)$. This implies that the number "6" is at the bottom of the die, the number "3" facing the square $(1, 2)$, the number "4" adjacent to the south side of the board, and the number "5" adjacent to the west side of the board. (Note that the two numbers of opposite faces of a die always add up to 7.)

We can roll the die in four directions: N, S, E, and W to an adjacent square, and as we do so, the faces tip over. If we e.g. roll the die north from the initial position, the die is now on square $(1, 2)$ with the number "4" facing upwards and the number "6" facing southwards. If we roll the die east from the initial position, the die is then on square $(2, 1)$ with the number "5" facing upwards and the number "4" facing southwards.

As we colour a die by rolling it, three things can go wrong. Firstly, the die can be rolled off the board. Secondly, not all six faces are being coloured. Thirdly, we may attempt to re-colour a face of the die that has already been coloured. Re-colouring a face of the die with the same colour as it already is, is ok, but re-coloring it with a different colour would yield a mixing of paint that is not permitted.

Input

The first line of the input contains three integers, N, M, and P. The size of the board N ($2 \leq N \leq 200$), the number of squares M ($1 \leq M < N^2$) containing paint, and the length P ($1 \leq P \leq 100,000$) of the path.

Then follows M lines, each containing three integers, x, y, and t, where square (x, y) contains colour t. Here $1 \leq x, y \leq N$ and $1 \leq t \leq 100$.

Finally follows one line with a string over the alphabet $\{N, S, E, W\}$ of length P. This is the path by which we roll the die. N indicates we roll the die north, and so on.

Output

Six integers on a single line which are the colours of the six faces, in order, starting from the colour of the face with the number "1" to the colour of the face with the number "6". If the die is not coloured correctly, output 0 0 0 0 0 0.

Sample Input	Sample Output
3 7 15 1 2 11 1 3 19 2 1 12 2 3 13 3 3 14 3 2 11 3 1 15 NSENNESSNWWNNNN	14 12 11 11 13 15