TOP SECRET

APS
HSPC

HIGH SCHOOL PROBLEM SHEET

Open at 12:00pm

A: Car Chase

A rogue agent is escaping in a car through a city grid, and it's your mission to catch them before they vanish. Luckily, your car is faster than the agent's.

The city is represented as a 2D grid. The rogue agent starts at a specific coordinate and moves 1 block per minute in one direction (either horizontally or vertically). On the other hand, you can move 1 block per minute in both directions, allowing you to cover up to 2 blocks of Manhattan distance in a single minute. You can assume that the agent will always move away from you. You can also assume that they will move in either the positive x or positive y direction.

Movement Rules:

- **Spy's Movement:** The spy moves 1 block per minute, either horizontally or vertically.
- **Your Movement:** You can move to 2 blocks in one direction per minute or 1 block in both directions per minute.
  - For example: Starting at (1, 1), you can move to (2, 2) in one minute, or you can move to either (1, 3), or (3, 1).

Input

The first line consists of an integer, T, representing the number of test cases.

For each test case:

The first line contains two integers, X__agent and Y__agent, the starting coordinates of the agent.

The second line contains two integers, X__you and Y__you, the starting coordinates of your car.

Sample Input

```
1
5,5
1,1
```

Output

The output consists of 1 line for each test case that contains the amount of time it takes (in minutes) to catch up to the agent.

Sample Output

```
8
```

B: Find the Mole

You are part of an elite spy organization, the Hidden Spy Protection Coalition (HSPC). Unfortunately, your organization has faced issues with specific employees not completing enough missions. We suspect that these employees may be moles working for the enemy. To determine if this is the case, we need to analyze the efficiency of each spy given the number of missions they complete on average each day.

You are tasked with writing a program that will determine the name of the agent who, on average, completed less than 4 missions per day. In order to find this average, find the mean of the number of missions completed daily. You will be given a list of agents and the number of missions they completed daily for the past few months. The agent who completed less than 4 missions per day is the mole — output this agent's codename and we will fire them!

Input

The first line in the test data file contains the number of test cases. Each test case starts with a line denoting the number of data points n ($1 \leq n \leq 1000$). The following n lines consist of an agent's codename, followed by a comma, followed by the number of missions completed on a day. The codename will be a string of length l ($1 \leq l \leq 100$), consisting of alphanumeric characters (a-z, A-Z, and 0-9). The number of missions completed will be an integer m ($1 \leq m \leq 1000$). **No spaces or whitespace will be present anywhere in the input.**

Output

For each test case, output the name of the agent who, on average, completed less than 4 missions per day on a separate line. It is guaranteed that only one agent will complete less than 4 missions per day.

Sample Input

```
1
6
Phantom,5
Shadow,10
Goof,3
Shadow,12
Phantom,15
Goof,2
```

Sample Output

```
Goof
```

C: Decrypt the Hacker's Message

You've just become a junior special agent with the Hacker Service Protection Coalition (HSPC). The notorious hacker, known only by the alias SW4N, has recently published an encrypted message. Luckily, a member of your team was able to come up with some steps to decrypt the message.

Your first mission, should you choose to accept it, is to write a program that will decrypt the hacker's message. Write a function that takes in a string and returns the decrypted message. The message will be encrypted using the following steps:

1. The letters in the message will be rotated by 13 places in the alphabet.
2. The numbers in the message will be rotated by 5 places in the number line. (Include 0)
3. After performing the previous two steps, any consecutive repeated **letters** will be "compressed" by replacing the repeated letters with the number of repetitions followed by the letter itself.

Input

The first line in the test data file contains a number of test cases n. Each test case will be one string of length l, consisting of characters A-Z (uppercase only), spaces, and digits 0-9. The input will follow $0 < n < 1000$ and $0 < l < 10\_000$.

Output

For each test case, output the decrypted message.

Sample Input

```
1
FJAAAA VF UNPXRE AHZORE 6
```

Sample Output

```
SW4N IS HACKER NUMBER 1
```

D: Combination Lock

You are trying to access a secret room in a high-security building, but the room's door is locked with a number of combination locks. Each lock has a number of wheels n, where each wheel has a highest digit d. The correct combination for each lock is a sequence of n digits consisting of the numbers 0 up to and including d. Write a program that outputs all possible combinations for each given lock, in numerically increasing order of the combinations.

Input

The first line in the test data file contains the number of test cases. Each test case consists of an integer n, the number of wheels ($1 \leq n \leq 5$), and the integer d, the highest digit on the wheel ($0 \leq d \leq 9$), separated by a space.

Output

For each test case, output all possible combinations for the lock in numerically increasing order. Each combination should be on a separate line.

Sample Input

```
2
1 6
2 1
```

Sample Output

```
0
1
2
3
4
5
6
00
01
10
11
```

E: Hidden Signals

You are an elite codebreaker for the HSPC (Highly Secretive Protocol Command), a secret organization dedicated to intercepting and decrypting enemy transmissions. Your latest assignment involves a coded message sent by a notorious double agent. The agent's messages are usually layered with complex instructions designed to mislead anyone trying to decipher them.

Your task is to decode the message by following a series of commands issued by HSPC's decryption protocol. Each command alters the encoded string through character replacements, deletions, or insertions. As an agent of HSPC, it's your job to carefully execute each command in order to reveal the hidden message.

Input

The first line contains an integer T, the number of test cases.

For each test case:

The first line contains an integer N, the number of commands. The second line contains the original encoded message, a string of lowercase letters. Each of the next N lines contains one command in one of three possible formats:

- replace `old_char` with `new_char` - replaces each occurrence of `old_char` in the message with `new_char`.
- delete `char` - removes each occurrence of `char` from the message.
- insert `char` before `position` - inserts `char` before the character at the specified (1-based) `position` in the message.

The encoded message will have a length between 1 and 100 characters. There will be at most 30 commands. The position for insertion will always be valid (within the bounds of the current message).

Output

Output the decrypted string message after applying all the commands in the specified order.

Sample Input

```
1
8
mqzsrwek
replace q with o
insert p before 3
delete z
replace m with t
insert e before 5
replace k with t
insert c before 6
delete w
```

Sample Output

```
topsecret
```

F: Find Secrets.txt

Your organization, the Hacker Service Protection Coalition (HSPC), has recently hacked into a server from a rival foreign intelligence agency. You know that the server contains a critical file named `secrets.txt` with info on the agency's next big operation. Given a directory tree of all of the files on the enemy server, your task is to write a program that will find the path to the file `secrets.txt`.

Input

The first line in the test data file contains the number of lines n in the input. Each of the following n lines will consist of either:

1. A string of length l consisting of alphanumeric characters and ending with a `/` character, representing a directory in the server.
2. A string of length l consisting of an alphanumeric file name, a `.` character, and an alphanumeric file extension, representing a file in the server.

Each line will be indented using a number of `-` (dash) characters d to represent the depth of the file or directory in the server.

The input will follow $0 < n < 200,000$, $0 < l < 100$, and $0 < d < 100$. Within a directory, you may assume there will be no duplicate names for files or subdirectories. However, there may be multiple directories or files with the same name but in different parent directories.

Output

Output the path to the file `secrets.txt` in the server. Start from the root directory `/`, followed a number of directories separated by a `/` character, and ending with the file name `secrets.txt`.

You may assume that there is only one file named `secrets.txt` in the server, and that it will always be found in the input.

Sample Input

```
7
home/
- dir1/
-- file1.txt
-- document.pdf
- dir2/
-- secrets.txt
- file2.png
```

Sample Output

```
/home/dir2/secrets.txt
```

G: Laser Beam

As a quality assurance engineer at the High Speed Phaser Collaborative (HSPC) you are tasked with measuring the reflectance of the mirror used in our product. Unfortunately Carl had a little too much sugar and ran through the testing facility, knocking all our testing lasers off course. Your task is to figure out which lasers need to be adjusted to hit the mirrors given the coordinates of the mirror and the angle of the laser.

Input

The first line of input is the number, n, of lasers you must check. This is followed by n lines which consist of the x and y coordinate of the first endpoint of the mirror, the x and y coordinate of the second endpoint of the mirror, and the angle of the laser, separated by spaces. All of the numbers are integers. The endpoints are included in the mirror. The first and second coordinates are in random order.

The angle starts at the positive x axis and increases counterclockwise. 0° is the positive x axis and 90° is the positive y axis. The laser starts at the origin, the point where x and y both equal zero. Angles are given in degrees 0° up to but not including 360°.

Assume that the laser or the mirror will never be perfectly vertical.

n will be between 1 and 10,000 inclusive. All coordinates will be between -20 and 20 inclusive.

Output

For each laser you are checking, print "true" if the laser intersects the mirror or "false" if it does not on a new line.

Sample Input

```
6
2 17 -10 -7 323
17 14 9 19 333
-1 -15 -12 -19 242
-18 7 -20 -13 186
7 10 13 17 276
19 -8 -1 -12 95
```

Sample Output

```
true
false
true
true
false
false
```

H: Portal Chaos

While on a top secret mission, one of our junior operatives, Johnny, tripped and dropped our portal generator. When the generator hit the floor, it started generating extra portals, and our operative was forced to return. As our top operative at Hyper-Space Portal Corporation (HSPC), your mission, if you chose to accept it, is to navigate the maze of portals and safely retreieve our portal generator. Luckily, the generator broadcasted the details of all the portals it generated. Each of the portals connects 2 rooms and can be used in each direction. You will start in room #0, and the portal generator will be in room #1000. You can assume that there is a path from the start room to the portal generator.

Input

The first line is the number of portals that were generated, n. This is followed by n pairs of room numbers, which are each on a new line and separated by a comma. Each pair represents a portal which connects two rooms. The portals will be given in a random order.

Output

Output the least number of portals you would have to travel through to get to the generator.

Sample Input

```
5
2,4
4,1000
2,3
0,1
1,2
```

Sample Output

```
4
```

I: Gadget Collections

The engineers at the High Stakes Pocket-filler Coalition need your help to develop the most effective collection of gadgets out of their vast collection. Unfortunately, each spy's pockets are only so big, and the gadgets take up different amounts of room. You are tasked to find the number of different ways the agents can fill their pockets with gadgets. Agents may hold multiple of the same gadget at the same time.

Input

The first line of the input data is the number of test cases, m. Each test case starts with a new line which is size of the agent's pocket, s, which is in the range of 250 to 400, inclusive. The next line is the number of unique gadgets in the collection, n, which is in the range 8 to 13, inclusive. The next n lines are the sizes of each gadget in the collection. The sizes are given in a random order.

Output

Your output should be one line, for each test case, which contains the number of possible combinations that can fill the pocket. Each gadget may be used multiple times.

Sample Input

```
2
4
2
2
1
3
1
4
```

Sample Output

```
3
0
```