University of Lethbridge

A: Grapheme–Colour Synesthesia

Grapheme–colour synesthesia is a condition that some people have that affects how they see letters. A person with synesthesia would look at a word and perceive each letter as being a specific colour. It is a completely involuntary response. For example, the title of this page is written in the colours of how Kayleigh sees those letters. Here is how she perceives the alphabet:

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz0123456789

Aa	Red	Gg	Brown	Mm	Orange	Ss	Black	Yy	Yellow	4	Purple
Bb	Blue	Hh	Orange	Nn	Green	Tt	Blue	Zz	Black	5	Orange
Cc	Yellow	Ii	Gray	Oo	Gray	Uu	Orange	0	Gray	6	Green
Dd	Brown	Jj	Purple	Pp	Yellow	Vv	Yellow	1	Gray	7	Pink
Ee	Green	Kk	Pink	Qq	Pink	Ww	Green	2	Green	8	Purple
Ff	Brown	Ll	Purple	Rr	Brown	Xx	Black	3	Pink	9	Black

Table 1: Colours in "simple" terms. Kayleigh actually describes the colours in more detail. For example '4' as *lavender*, but we won't use her detailed classification here.

On top of looking cool, Kayleigh uses this ability for remembering different things. For example, she may remember that a person's name starts with "blue". This would help her to remember the name of the person later on in life. She also remembers her passwords this way. For each password, she remembers the colour sequence of the letters exactly. So for the password "hello", she remembers orange-green-purple-purple-gray. She doesn't think that there are not too many possibilities for each colour sequence, so she plans on trying all words that have that colour sequence.

Unfortunately, websites have a limit of how many passwords you are allowed to try before your account gets locked. If the limit is L, once you enter L incorrect passwords, your account is locked. Given the password and L, is it guaranteed that Kayleigh will be able to login to her account?

Input

Input will start with an integer T ($1 \leq T \leq 200$) denoting the number of test cases. For each test case, you will be given p, the password, and L ($1 \leq L \leq 400000000$). The length of p will be at least one and no more than 100 and will only contain lowercase letters and numbers. You may assume that Kayleigh never uses capital letters in her password (so you do not need to consider capital letters at all in your calculations).

Output

For each test case, output `Yes` if Kayleigh is guaranteed to guess the password before her account is locked or `No` if she is not.

Sample Input

```
2
abc 1000
abc 2
```

Sample Output

```
Yes
No
```

University of Lethbridge

B: Bookshelves

3... 2... 1... YES! After five grueling hours in an online bidding war, Kayleigh has just won against MrGnomeLover562! At stake was a box of books on garden gnomes. Filled with joy, Kayleigh ran to the nearest store where she bought a bookshelf (even though she had no idea how many books were inside). Two weeks later, the box arrived and she opened it. On top were such titles as *The Care and Maintenance of Pointy Red Hats*, *21st Century Gnome Culture* and *Gnowing Your Gnome*.

It wasn't long before she noticed that there were a lot more books here than she thought there was going to be. She counted the number of books that was in the box and went back to the store to purchase more bookshelves if they were needed. Kayleigh likes to save space, so after she counted the number of books, she will not have bought more bookshelves than what was needed. Except don't forget that she had already bought one bookshelf, so if the box is empty, she still has one shelf. Each bookshelf is identical and can hold a maximum of C books. How many bookshelves will she have in her house after she has unpacked everything?

Input

The first line of the input contains a positive integer T ($T \leq 100$), denoting the number of test cases to follow. Each test case will contain two integers B ($0 \leq B \leq 10^7$) and C ($1 \leq C \leq 10^7$) denoting the number of books in the box and the capacity of each bookshelf, respectively.

Output

For each test case, output one line in the form "`Case #x: y`" where `x` is the case number (starting at 1) and `y` is the number of bookshelves that Kayleigh will have in her house.

Sample Input

```
2
10 10
32 9
```

Sample Output

```
Case #1: 1
Case #2: 4
```

University of Lethbridge

C: Coffee Time

Coffee shops are extremely popular places at Monash University. A new shop that just opened up has made the mistake of only having one very slow coffee grinder – it only grinds two teaspoons of coffee per minute! Because the owner knows that it is going to be very busy, he has made the decision that he will run the coffee grinder consistently from open until close. This way, he can keep a pile of reserve coffee grounds.

The coffee shop serves five sizes of coffee: extra-small (XS), small (S), medium (M), large (L) and extra-large (XL). To make a cup of coffee, the following amount of coffee grounds are needed:

Size	XS	S	M	L	XL
Teaspoons	2	3	4	5	6

Given the orders throughout the day, how many people will have to wait to be served and how many can be served immediately?

Input

The input will start with an integer T ($1 \leq T \leq 100$), the number of test cases to follow. Each test case will start with an integer n ($1 \leq n \leq 10000$) denoting the number of customers that will be served today. The following n lines contain the orders throughout the day. Each line contains two items: an integer k ($0 \leq k \leq 1440$) and a string s where s is one of `XS`, `S`, `M`, `L` and `XL`. This means that the customer ordered that size of drink k minutes after the restaurant opened. The input is given in chronological order, so the time given for the ith person will be no later than the time given for the $(i+1)$th person.

Output

For each test case, output one number: the number of people who have to wait to have their coffee. The customers must be served in the order that is given in the input (even if multiple customers arrive at the same minute – see the second and third sample input cases). You may assume that the staff works infinitely fast. This means that if there are enough coffee grounds to make the coffee, they will make it instantaneously.

Sample Input

```
3
3
1 XS
2 XS
4 M
3
2 XL
2 M
4 XS
4
2 XL
2 M
4 XS
7 XS
```

Sample Output

```
0
3
3
```

University of Lethbridge

D: Dice Rolling

Rolling dice is a very important part of many board games. Dice are cubes with a different number of dots on each side. For this problem, we need to know all of the different ways that a certain number can be rolled with one or two dice. When one die is rolled, the value is the number of dots facing upwards. When we roll two dice, the value is the total number of dots between both dice (the sum of both dice).

Each die must be drawn in the following format. The dots are to help you with the exact spacing of each drawing. Each dot ('.') represents a space (' ') character. In the actual output of the program, you must output the spaces, NOT the dots! (For example, see the sample input).

```
    .-------.       .-------.       .-------.
    |.......|       |.....o.|       |.....o.|
1 : |...o...|   2: |.......|   3 : |...o...|
    |.......|       |.o.....|       |.o.....|
    .-------.       .-------.       .-------.

    .-------.       .-------.       .-------.
    |.o...o.|       |.o...o.|       |.o...o.|
4 : |.......|   5: |...o...|   6 : |.o...o.|
    |.o...o.|       |.o...o.|       |.o...o.|
    .-------.       .-------.       .-------.
```

Input

The first line of the input contains a positive integer T ($T \leq 20$), denoting the number of test cases to follow. Each test case will contain one integer n ($1 \leq n \leq 12$) each on their own line.

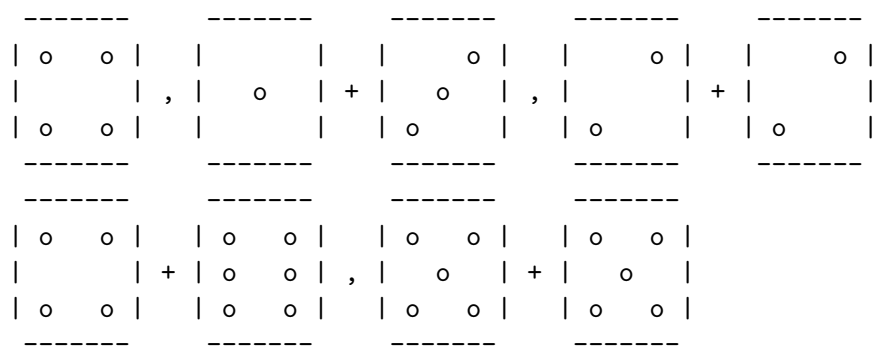
Output

For each test case, output all of the possible combinations of one or two dice that sum to n. All images should appear in one horizontal strip. Print all of the possibilities with just one die before any with two dice. When two dice are used, the first die should be no larger than the second die. Print the pairs in increasing order based on the first die. Separate two different possibilities with a comma (','). When using two dice, put a plus sign ('+') in between the two dice. See the Sample Output for the exact speficiations.

Sample Input

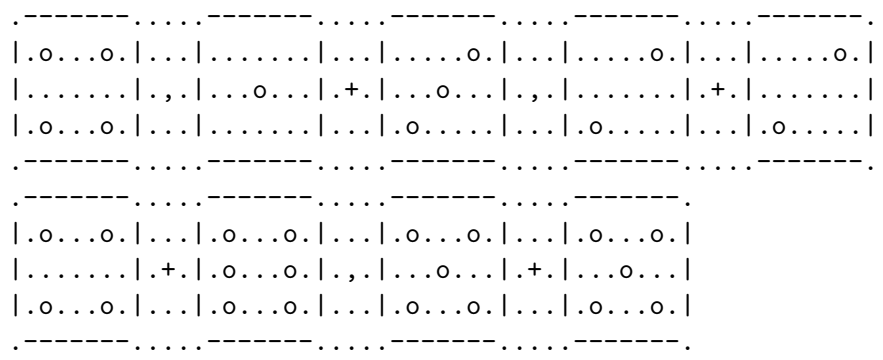
```
2
4
10
```

Sample Output



Sample Output Explained

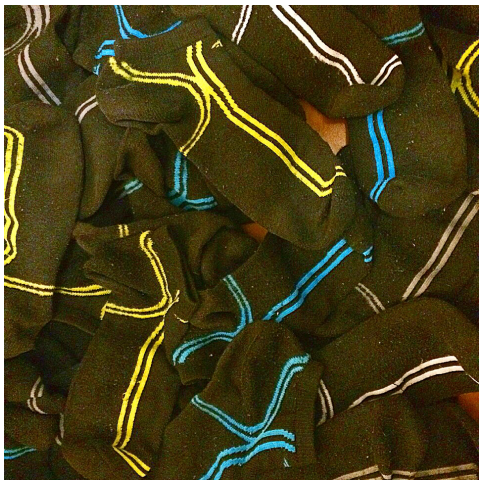
Here is the sample output again. This time, all space characters have been replaced by dots to show exactly where the spaces are. Remember: **your program should output the spaces, NOT the dots!** Your program will be considered wrong answer if any space is wrong, not presentation error.



University of Lethbridge

E: Evil Dryer

Does anyone have the same problem with their dryers that I do? They eat your socks! Somehow, I put a pair of socks into the dryer, and when I come back 45 minutes later, one of the two are gone!



Recently, I have purchased many pairs of nearly identical socks. The only difference between them is the colour. In total, there are 26 different colours (represented as `a-z`). Any two socks of the same colour can be used to make a pair of socks. This way, even if the dryer eats one of a specific colour of sock, I still have many other socks of the same colour.

I just finished a load of laundry and have thrown all of the socks onto a pile on my bed. It is at this point that I am curious how many of my socks have been eaten by the dryer. Unfortunately, I do not remember how many socks I put into the dryer to begin with (or how many of each colour), but I am confident that I only put complete pairs of socks into the dryer. What is the minimum number of socks that the dryer has eaten?

Input

The first line of the input contains a positive integer T ($T \leq 20$), denoting the number of testcases to follow. Each testcase will be on one line. Each line will contain a nonempty string of length less than 1000. This string denotes the colour of all of the socks that I have on the bed (the only characters in the string will be `a-z`). The string is in no particular order.

Output

For each test case, output one line containing one integer: the minimum number of socks that the dryer ate.

Sample Input

```
2
abcdeabcde
abcdeabcdeabcdeabcd
```

Sample Output

```
0
1
```