Problem A. Sentence Reconstructing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow wants to play a sentence-making game. His friend prepares a sentence and meow must look very carefully to find a new sentence within his friend's! But there's a catch. **Each letter from Meow's friend's sentence can only be used once**. Fortunately, the game is quite simple because it doesn't involve any punctuation. For example, given the list, *"friend icing on the sung"* can the sentence *"coding is fun"* be formed? The answer is yes. However, given the list *"please bake umpteen cookies"* you can't form the sentence *"bugs are problems"*. Note that spaces are also considered as a letter.

Input

Two strings will be inputted, and each string will comprise at least one character.

The strings can vary in length ($1 \leq Length \leq 10^6$). For example, the second string may be longer, shorter, or the same length as the first string. All letters input will be lowercase.

Output

Display the word *"possible"* if the second string can be created using the letters contained in the first string.

Display the word *"not possible"* if the second string can't be created using the letters contained in the first string.

Examples

standard input	standard output
friend icing on the snug coding is fun	possible
please bake umpteen cookies bugs are problems	not possible
meowmeowmeow meow meow	not possible

Note

For test case 2, Meow's friend sentence does not contain the letter 'g'.

For test case 3, Meow's friend sentence does not contain a space.

Problem B. Pawsome Cards

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In Meow's world, all doors are installed very highly advanced locks, they are integrated with different features, such as paw reading, purr detection, fur scan, biometric eye scanning and more! These digital locks are even used in vaults and high security facilities. But there's something even more a-paw-ling, they have yet to realize a major flaw in their locks!

Meow is a legendary spy known for its ability to break into any secret bases in its world. It was able to achieve this because it realized the flaw, which is that all doors and vaults in their world will always have a key-card access or length, x and width, y. Meow was able to create key cards that can access many different locks at once, but it has no control over the size of these cards. Meow's uncle, Meowtini is a leatherworker and wants to help Meow make a card holder for it to carry these cards on its missions, but the problem is that Meow has no idea how big of a card holder it needs to keep all these cards when it goes on a mission.

Your job is to find the dimensions of a card holder with length, l and width, w that can fit these cards. Your job is to create a system that accepts 2 types of queries:

1. $+$ x y – Meow creates a card of size $x \times y$
2. ? h w – Meow checks if the card holder with size h $\times$ w fits all the cards up to that moment

The card holder can hold cards when if it fulfils one of the two condition:

1. $x \leq h$ and $y \leq w$
2. $y \leq h$ and $x \leq w$

You are required to output an **INTEGER** that indicates if the wallet fits all the previous cards mentioned. (0 – doesn't fit, 1 – fits all cards)

Input

The first line contains an integer n ($2 \leq n \leq 10^4$) – the number of queries.

The next n lines follows these 2 types of query formats:

1. $+$ x y – Meow creates a card of size $x \times y$
2. ? h w – Meow checks if the card holder with size $h \times w$ fits all the cards up to that moment ($1 \leq h, w \leq 10^9$)

It is guaranteed that the queries start with type **1** before having any type **2** queries and that there will be at least one type **2** queries within the input data.

Output

For each query type **2**, print 1 if the card holder can fit all cards up to that point and 0 if it cannot.

Example

standard input	standard output
14 + 2 4 + 4 2 ? 1 10 + 4 5 ? 5 5 + 1 5 ? 100 10 ? 1 5 + 105 24 + 40 25 ? 200 23 ? 110 25 + 195230571 928276543 ? 1000000000 1000000000	0 1 1 0 0 1 1

Problem C. Purrhaps a Jog?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow loves taking walks and jogs around the city, to the point where it knew the number of steps, a it'd need to finish a full lap for each town! One day, Meow wanted to train his algorithmic and mathematic skills while jogging! So, it started counting the number of steps, and jotting down the last digit of the steps it has taken when the number of steps taken reaches the number divisible by b.

For example, if Meow reaches the 105th step (a), and the divider Meow has chosen is 24 (b), the steps divisible will be 24, 48, 72, 96. Their last digits will then be 4, 8, 2 and 6. After the run, Meow then wants to sum up this number! (Total sum of last digit= 20)

Now, Meow wants to develop its skills even further, but it doesn't know if it's calculating the numbers correctly. Your job is to develop an algorithm to help Meow calculate the sum of all last digits that Meow has written down! The number of towns (test cases) that Meow runs through will be indicated with an n.

Input

First line is the number of test cases ($1 \leq n \leq 1000$).

The following n lines contains each test case (towns that Meow has ran in). Each test case has 2 integers, a and b ($1 \leq a, b \leq 10^9$) where a that indicates the total steps jogged; b indicates the divisor chosen.

Output

Print the integer for each query, the sum of the last digits written down by Meow.

Example

standard input	standard output
5 1 1 105 24 111 21 313 24 198273 28	1 20 15 54 28328

Problem D. Money Bag

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow is a legendary bank robber. One day, when carrying out his usual heist, he found that the bank had more gold bars than he expected. He only brought a money bag with volume V. However, the bank has n gold bars, each with their own volume.

Meow knew that the larger the volume of the gold bar, the more it was supposed to be worth. Hence, to maximize his profit, he would like to take some (or all) of those n gold bars and put it in his money bag, such that the remaining space in the bag is minimized.

Input

The first line contains an integer V ($0 \leq V \leq 20000$) – the volume of the money bag

The second line contains an integer n ($1 \leq n \leq 30$) – the number of gold bars available inside the bank

The third line contains n positive integers $a_1, a_2, ..., a_n$, each representing the weight of the gold bar. ($1 \leq a_i \leq 1000$)

Output

Print a single integer, the remaining space of the bag.

Examples

standard input	standard output
20 3 10 5 5	0
24 3 20 1 5	3

Note

In the first example, Meow is able to fit all gold bars inside his bag with no extra remaining space.

In the second example, Meow is only able to fit gold bar with volume 20 and 1 in his bag, leaving 3 extra remaining space.

Problem E. Ice Cream

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Meow loves to eat ice cream. He will always have an ice cream in his hand every day. There are a lot of flavours like vanilla, chocolate, durian and so on.

One day, he tries to have multiple scoops of ice cream on one stick. He suddenly thought of how many combinations can he makes if he chooses r numbers of scoops from n different flavour. The flavours can be repeated. For example, if he needs to choose 3 scoops, it can be vanilla, vanilla, vanilla; vanilla, vanilla, chocolate and so on. Help Meow calculate the numbers of combination that he can make, note that the order of the flavour taken is not important.

Input

The first line contains a single integer t ($1 \leq t \leq 1000$) – the number of test cases

Each test case contains a line with 2 integers n and r ($1 \leq n, r \leq 1000$), denoting the total number of different flavour and number of ice cream scopes to be taken, respectively.

Output

Output the possible number of variations of flavours for the ice-cream. Note that the output may be very big and may exceed 2^{64}

If the case is impossible, output 0 as the answer.

Example

standard input	standard output
3 4 2 4 3 10 4	10 20 715

Note

For the first test case, assume the 4 flavours are 'a', 'b', 'c', 'd'.

Meow can take aa, ab, ac, ad, bb, bc ,bd, cc, cd, dd, making total of 10 different variations.