Problem A : Progression

(blue balloon)

According to Wikipedia, an arithmetic progression (AP) is a sequence of numbers such that the difference of any two successive members of the sequence is a constant. For instance, the sequence 3, 5, 7, 9, 11, 13, . . . is an arithmetic progression with common difference 2. For this problem, we will limit ourselves to arithmetic progression whose common difference is a non-zero integer.

On the other hand, a geometric progression (GP) is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed non-zero number called the common ratio. For example, the sequence 2, 6, 18, 54, . . . is a geometric progression with common ratio 3. For this problem, we will limit ourselves to geometric progression whose common ratio is a non-zero integer.

Given three successive members of a sequence, you need to determine the type of the progression and the next successive member.

Standard Input

Your program will be tested on one or more test cases. Each case is specified on a single line with three integers ($-10,000 < a_1, a_2, a_3 < 10,000$) where a_1, a_2, and a_3 are distinct.

The last case is followed by a line with three zeros.

Standard Output

For each test case, you program must print a single line of the form:

XX u v

where XX is either AP or GP depending if the given progression is an Arithmetic or Geometric Progression. v is the next member of the given sequence. All input cases are guaranteed to be either an arithmetic or geometric progressions.

Sample Input	Sample Output
4 7 10 2 6 18 0 0 0	AP 13 GP 54

Problem B: Kids

(red balloon)

Your son's birthday is corning soon (assume that you have a son), and you promised to make the best party ever for him. He will be very happy if he can invite all his friends to this party (he has many friends), but unfortunately you can't invite everyone because you have a limited number of candies, and you want everyone to be happy.

As we all know, kids love to eat a lot of candies of the same type, let's say a kid will be happy only if he can eat at least K candies of the same type.

Given K, and the number of available candies of each type, calculate the maximum number of kids where you can make all of them happy by giving each one at least K candies of the same type.

Standard Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cases ($1 \leq T \leq 100$). Followed by the test cases, each test case is on two lines. The first line of each test case contains two integers N, the number of different candies ($1 \leq N \leq 100$), and K, the minimum number of candies which will make a kid happy as described above ($1 \leq K \leq 100$). The second line of each test case contains N integers, separated by a single space, which are the available number of candies of each type. There will be at least 1 candy and at most 100 candies of each type.

Standard Output

For each test case, print on a single line one integer, the maximum number of kids you are asked to calculate as described above.

Sample Input	Sample Output
2 3 2 4 5 7 3 8 4 5 7	7 0

Problem C : Fuel
(yellow balloon)

The government of Africaland has recently announced a new petrol rationing plan with an unexpected price hike. According to the new plan, each person receives a quota of 60 liters per month in a fuel card. Each liter of petrol costs 1500 nairas if it is within quota. Any extra fueling costs 3000 nairas per liter.

After recovering from the shock, Mahya is trying to figure out how dark is the future. The current month is coming to an end, and Mahya has some quota left in her fuel card, remaining available for the next month. A quota of 60 liters will also be added to her fuel card just at the beginning of the next month. She also has a prediction of the amount of petrol that will be used in the next month. She now wants to know how much she should pay for petrol in the next month. However, she is too lazy to do that on her own. So, she needs your help to calculate the cost for her.

Standard Input

The first line of input contains an integer **N** which is the number of datasets that follow. Each dataset consists of two lines. The first line contains an integer n ($0 \leqslant n \leqslant 200$), specifying the amount of petrol that will be used in the next month. The second line contains an integer k ($0 \leqslant k \leqslant 360$), showing the quota left in Mahya's fuel card at the end of current month.

Standard Output

For each dataset, print out on a single line, the amount of money (in nairas) that Mahya will pay for petrol in the next month.

Sample Input	Sample Output
2 41 0 125 40	61500 225000

Problem D : Africaland
(Orange balloon)

Again, Africaland has experienced a very bad economic condition over the past few months. The value of naira, the national currency of Africaland, changes against one unit of gold very rapidly. People in Africaland, all wondering about their savings, are trying to exchange their savings with gold coins.

Dr. Adebayor Kujiku who is a data scientist, has obtained a prediction of the price (in nairas) of a gold coin for the next n days based on the existing data over the past 40 years. He believes his prediction, and now he want to increase his savings based on it. He was wondering how much savings he has at the end of the n-th day assuming that he has c nairas at the beginning of the first day. Since Dr. Adebayor Kujiku is not a programmer, he asks you to help to find his answer.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq \mathbf{P} \leq 1000$), which is the number of data sets that follow. Each data set begins with a line containing two integers c ($0 \leqslant c \leqslant 3000$), Dr. Adebayor Kujiku's initial savings in nairas, and n ($0 \leqslant n \leqslant 30$), the period of his prediction. Each of the next following n lines of the dataset contains an integer pi ($1000 \leqslant pi \leqslant 2000$) denoting the price of a gold coin at day i ($1 \leqslant i \leqslant n$) in nairas.

Standard Output

The output contains just an integer, which indicates the maximum savings he can obtain at the end of the n-th day assuming that Dr. Adebayor Kujiku exchanges all his remaining gold coins (if there is any) to nairas at the end of the n-th day.

Sample Input	Sample Output
2 1000 3 1000 1100 1200 2000 4 1000 2000 1500 1800	1200 4600

Problem E : Emails
(Purple balloon)

This year, many people registered for the internet contest with several email addresses. We want to see how many valid and distinct email addresses registered.

A valid email address consists of a username and a domain name separated by a character '`@`'. A username is a string containing letters (`a-z` and `A-Z`), digits (`0-9`), underscores (`_`), and periods (`.`). Usernames cannot begin or end with a period and cannot contain two consecutive periods. Other than this rule, periods do not matter in email addresses (they can be removed without changing the address). Uppercase and lowercase letters in the usernames are considered the same. So, usernames `AliBaba` and `ali.baba` are considered the same. Usernames should contain 6 to 30 characters, after removing all of its periods.

A valid domain name is a string of length between 3 and 30 (inclusive), consisting of domain parts separated by periods (`.`). A domain name must not start or end with a period. Each domain part is a non-empty string of letters (`a-z` and `A-Z`), digits (`0-9`), and dash (`-`). Uppercase and lowercase letters in the domain names are also considered the same. So, `Foo.bar` is the same as `foo.Bar`, but not the same as `Foo-Bar` or `Foobar`.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq \mathbf{P} \leq 1000$), which is the number of data sets that follow. The first line of the dataset contains a positive integer n ($1 \leqslant n \leqslant 1000$), the number of the registered email addresses. Each of the next n lines contains one email address of length at most 100 and consisting of alphabets, digits, '`@`', '`.`', '`_`', and '`-`'.

Standard Output

For each input data set, the output is a line containing a single integer which is the number of distinct email addresses that are valid.

Sample Input	Sample Output
1 5 acmacm@icpc.ir acmacm..a@icpc.ir .a1a1.abc@icpc.ir acma.c.m@icpc.ir acmacm@icpc.com	2