Asientos Privilegiados

With final exams just around the corner, Juan and his friends have decided to spend the week in the library studying. Since they are going to spend so much time in there, it is essential that they choose the best seats to concentrate. The best seats are the ones where there is a little air flowing: the ones by the window and by the aisle.

The library has f rows and a seats per row, and library seats follow the numbering shown in the picture: on the left is the window, on the right is the aisle, and the seats are numbered in ascending order from left to right in the first row, from right to left in the second row, from left to right again in the third row, and so on.

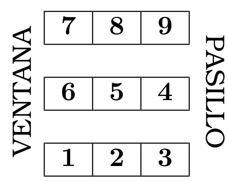

Figure 1: Numbering example with $f = 3, a = 3$.

Would you be able to tell, given a seat number, whether it is in window or aisle?

Input and output

The first line of the input contains the number of cases T.

For each case there will be a line with 3 numbers: f, the number of rows, a, the number of seats, and n, the seat number to query.

For each case, you must print a line with the answer: `VENTANA` if the seat is next to the window, `PASILLO` if it is next to the aisle, `AMBOS` if it is next to both, or `:(` if it is next to neither.

Example

Input:

```
4
3 3 1
3 3 4
3 3 5
5 1 1
```

Output:

```
VENTANA
PASILLO
:(
AMBOS
```

Constraints

$1 \leq T \leq 100\,000$.

$1 \leq f, a \leq 30\,000$.

$1 \leq n \leq f \cdot a$.

Subtasks

1. (15 points) $a = 1$.
2. (20 points) $f = 1$.
3. (30 points) $T \leq 1\,000$, $f \cdot a \leq 1\,000$.
4. (35 points) No additional constraints.

Pétalos

Szeged is a Hungarian city that in addition to hosting the IOI 2023, also has great monuments, beautiful monuments with great gardens, and therefore it is normal that the gardeners of this city take care to have a garden as beautiful as the monument the goes along with it.

In the garden of the Votive Church of Szeged there are n flowers, the i-th flower has a_i petals and to achieve perfection we want to make all flowers have the same number of petals, ie, $a_i = a_j$ for all i, j, for this we can tear petals from the flowers, and of course you cannot add petals.

At the same time the flowers also have to look nice, so we can't pluck petals in any way, we will pluck them with the following procedure:

- If the flower has k petals, we number these in order from 1 to k starting from any petal clockwise.
- We tear off the odd petals.

This way, if we have a flower with 16 petals we can get one with 8, and if we have one with 11, we can get one with 5. Note that this procedure can be applied more than once to the same flower.

Gardeners want to work as little as possible and wonder what is the minimum number of times they will have to apply this procedure to get all the flowers to have the same number of petals.

Input and output

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

Each case starts with a line containing n, the number of flowers.

The second line contains n integers a_i, the number of petals of each flower.

For each case you must print a line with a single integer, the minimum number of times the gardeners must apply the described procedure for all flowers to have the same number of petals.

Example

Input:

```
3
2
11 9
1
1000000000
2
1 2
```

Output:

```
4
0
1
```

Constraints

$1 \leq T \leq 5000$.

$1 \leq n \leq 10^4$, the sum of n over all cases will be less than or equal to 10^4.

$1 \leq a_i \leq 10^9$.

Subtasks

1. (11 points) $a_i = 1$ or $a_i = 2$ for each $1 \leq i \leq n$.
2. (19 points) $n = 2$.
3. (11 points) a_i is a power of 2 for every $1 \leq i \leq n$.
4. (24 points) The sum of n for all cases is at most 500.
5. (35 points) No additional constraints.

Sumar Número Compuesto

We say that a positive integer x is *composite* if it can be expressed as $x = y \cdot z$ with $y, z \geq 2$ integers.

Given n, you are asked to print a sequence of n positive integers $a_1, \ldots, a_n$ that satisfies the following two conditions:

- For all $1 \leq i < j \leq n$, $\sum_{k=i}^{j} a_k = a_i + \ldots + a_j$ is a composite number.
- For all $1 \leq i < n$, the numbers a_i and a_{i+1} are coprime, i.e., $\gcd(a_i, a_{i+1}) = 1$.

Input and output

The input contains a line with the integer n.

You must print a line with the n integers $a_1, \ldots, a_n$. In case there are several possible solutions, you can print any of them.

Example

Input:

```
3
```

Output:

```
5 7 3
```

Explanation: $\gcd(5, 7) = \gcd(7, 3) = 1$ and $5 + 7 = 12, 7 + 3 = 10$ and $5 + 7 + 3 = 15$ are all composite numbers.

Constraints

$2 \leq n \leq 1000$.

The numbers you write must satisfy $1 \leq a_i \leq 5000$ for all $1 \leq i \leq n$. It can be shown that there is always a solution that satisfies this constraint.

Subtasks

1. (18 puntos) $n = 5$.
2. (18 puntos) $n = 8$.
3. (18 puntos) $n = 12$.
4. (18 puntos) $n = 20$.
5. (28 puntos) $n = 1000$.

Volteando Segmentos

Let n be a given positive integer. The judge has a hidden number $1 \leq x \leq n$ that you must determine. To do this, you can ask a series of questions.

There is a permutation $p_1, \ldots, p_n$ where initially $p_i = i$ for all $1 \leq i \leq n$. You can ask the following query: you give an index i and the judge answers if $p_i < x$, $p_i = x$ or $p_i > x$. But after each query, if j is the index such that $p_j = x$, you flip the $[i, j]$ segment (if $i \leq j$) or the $[j, i]$ segment (if $i > j$) of the permutation, ie, denoting by p' the new permutation after the question, you have that $p'_i = p_j, p'_{i+1} = p_{j-1}, \ldots, p'_j = p_i$ if $i \leq j$ and analogously $p'_j = p_i, p'_{j+1} = p_{i-1}, \ldots, p'_i = p_j$ if $i > j$.

Your goal is to determine the number x without asking too many questions.

Input and output

This is an interactive problem. You must refresh the output each time you print data (`cout <<endl` or `cout <<flush` in C++, `System.out.flush()` in Java, `stdout.flush()` in Python).

The first line of the input contains the number of cases T.

For each case, you must first read a line with the number n. After reading this value, you must ask questions to the interactor:

To ask a question, you must print a line with the format `? i`, where $1 \leq i \leq n$ is the index. You will then be answered with a line with the character `<`, `>`, or `=`, depending on whether $p_i < x$, $p_i > x$, or $p_i = x$ respectively. In case you ask an invalid question or exceed the question limit, you will be answered with `-`, and in that case your program must terminate immediately.

To give the answer, you must print a line in the format `! x`, where $1 \leq x \leq n$ is the value of x that you have determined. Next, you must read a response with one character, which will be:

- `+`, if you go to the next case. In that case your program should read the new integer n and continue with the new case.
- `-`, if you do not go to the next case, either because the cases are finished or because you have given an incorrect answer. In that case, your program must terminate immediately.

In either case, your program should terminate only after having read a `-`.

Example

Here we show an example interaction in which, for clarification, we show the value of x for each case and the state of the p permutation when asking each question.

Judge	Program	Commentary
2		
5		$x = 3$
	? 1	$[1, 2, 3, 4, 5]$
$<$		$p_1 = 1 < 3$
	? 2	$[3, 2, 1, 4, 5]$
$<$		$p_2 = 2 < 3$
	? 3	$[2, 3, 1, 4, 5]$
$<$		$p_3 = 1 < 3$
	? 4	$[2, 1, 3, 4, 5]$
$>$		$p_4 = 4 > 3$
	? 5	$[2, 1, 4, 3, 5]$
$>$		$p_5 = 5 > 3$
	? 3	$[2, 1, 4, 5, 3]$
$>$		$p_3 = 4 > 3$
	! 3	
+		
3		$x = 1$
	? 3	$[1, 2, 3]$
$>$		$p_3 = 3 > 1$
	? 3	$[3, 2, 1]$
$=$		$p_3 = 1 = x$
	? 1	$[3, 2, 1]$
$>$		$p_1 = 3 > 1$
	? 2	$[1, 2, 3]$
$>$		$p_2 = 2 > 1$
	! 1	
−		

Constraints

$1 \le T \le 100$.

$2 \le n \le 30\,000$.

You can ask at most 50 questions. Giving the answer does not count as a question.

The value of x is fixed at the beginning of the interaction, i.e., it does not change adaptively according to the questions you ask.

Subtasks

1. (4 points) $n = 2$.
2. (18 points) $n \leq 40$.
3. (14 points) Either $x \leq 40$, or $n - x + 1 \leq 40$, or $\left|\frac{n}{2} - x\right| \leq 20$.
4. (31 points) $n \leq 600$.
5. (33 points) No additional restrictions.

Additionally, the score you get in a subtask depends on the number of questions you ask: to get a full score you must ask at most 15 questions and to get a positive score you must ask at most 50 questions. The score of each subtask is multiplied by a multiplier $M(q)$, where q is the maximum number of questions you have asked in the cases of that subtask. The value of $M(q)$ is given by:

$$M(q) = \begin{cases} 0 & q > 50 \\ 0.5 + \frac{50-q}{70} & 50 \geq q > 15 \\ 1.0 & 15 \geq q \end{cases}$$

Mínimo por máximo

Maria has a lifelong passion for doing odd operations with sequences. She has a sequence of n positive integers $a_1, \ldots, a_n$ and she has come up with the following: she chooses two contiguous disjoint subsegments of the sequence and multiplies the minimum value of the first subsegment (the subsegment with the lowest indexes) by the maximum value of the second subsegment.

For example, if the sequence is $[a_1, a_2, a_3, a_4, a_5] = [2, 4, 1, 5, 3]$, you can choose the subsegments $[1, 3]$ and $[5, 5]$, so $\min(a_1, a_2, a_3) = 1$ and $\max(a_5) = 3$ and the result is $1 \times 3 = 3$.

Now Mary wants to know what is the result of adding up all these outcomes for each possibility that she has of choosing the two subsegments. Since this number can be very large, you must calculate the remainder of the division of this number by $10^9 + 7$.

Formally, you must calculate:

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j+1}^{n}\sum_{l=k}^{n}\min(a_i, \ldots, a_j) \times \max(a_k, \ldots, a_l) \pmod{10^9 + 7}$$

Input and output

The first line of the input contains the number of cases T.

For each case there will be two input lines: one line with the number n and the next line with the n integers $a_1, \ldots, a_n$.

For each case, you must print a line with the requested answer.

Example

Input:

```
4
3
1 2 3
5
2 4 1 5 3
2
2 3
4
1000000000 1000000000 999999999 1000000000
```

Output:

```
17
291
6
777
```

Constraints

$1 \leq T \leq 1\,000$.

$2 \leq n \leq 10^6$.

$1 \leq a_i \leq 10^9$ for all $1 \leq i \leq n$.

The sum of n for all cases is at most 10^6.

Subtasks

1. (5 points) $n = 2$.
2. (6 points) The sum of n^4 for all cases is at most $5 \cdot 10^8$.
3. (13 points) The sum of n^3 for all cases is at most $2 \cdot 10^8$.
4. (18 points) $n \leq 5000$, the sum of n^2 for all cases is at most 10^8.
5. (7 points) $a_i = 1$ for all $1 \leq i \leq n$.
6. (12 points) $a_i \leq 2$ for all $1 \leq i \leq n$.
7. (9 points) $a_i = i$ for all $1 \leq i \leq n$.
8. (30 points) No additional constraints.

Cartas

Manuel is a bit messy and has left n cards on top of his desk, the cards have a number written on one side and another number on the opposite side. Although he is not very tidy, for aesthetic reasons he wants the n numbers facing up to be two-by-two coprimes and the n numbers facing down to also be two-by-two coprime (n numbers $s_1, \ldots, s_n$ are two-to-two coprimes if and only if for all $1 \leq i < j \leq n$, $\gcd(s_i, s_j) = 1$, or, in other words, $\prod_{i=1}^{n} s_i = \operatorname{lcm}(s_1, ..., s_n)$).

To get this property satisfied, you can flip several (maybe 0) cards. Now Manuel wonders how many ways the cards can be laid out so that the numbers facing up are two-by-two coprimes and the numbers facing down are also two-by-two coprimes. Two card layouts are different if there is a card that has an opposite orientation in these layouts, even if this card has the same number on both sides.

Since this number can be very large, Manuel wants you to give as an answer the remainder of the division of the result by $10^9 + 7$.

Input and output

The first line contains t, the number of cases to process.

Each case starts with a line containing n, the number of cards.

The next n lines contain two numbers a_i and b_i, the numbers written on each side of the i-th card.

The output should print one line for each case with the number of possible distributions of cards that satisfy the described property modulo $10^9 + 7$.

Example

Input:

```
4
2
1 1
1 1
2
4 3
2 9
2
5 7
11 3
2
5 10
15 25
```

Output:

```
4
2
4
0
```

Constraints

$1 \leq t \leq 1000$.

$1 \leq n \leq 10^5$, the sum of n in all cases is less than or equal to 10^5.

$1 \leq a_i, b_i \leq 10^5$.

Subtasks

1. (11 points) $1 \leq n \leq 2$.
2. (40 points) $t = 1, 1 \leq a_i, b_i \leq 15$.
3. (49 points) No additional constraints.