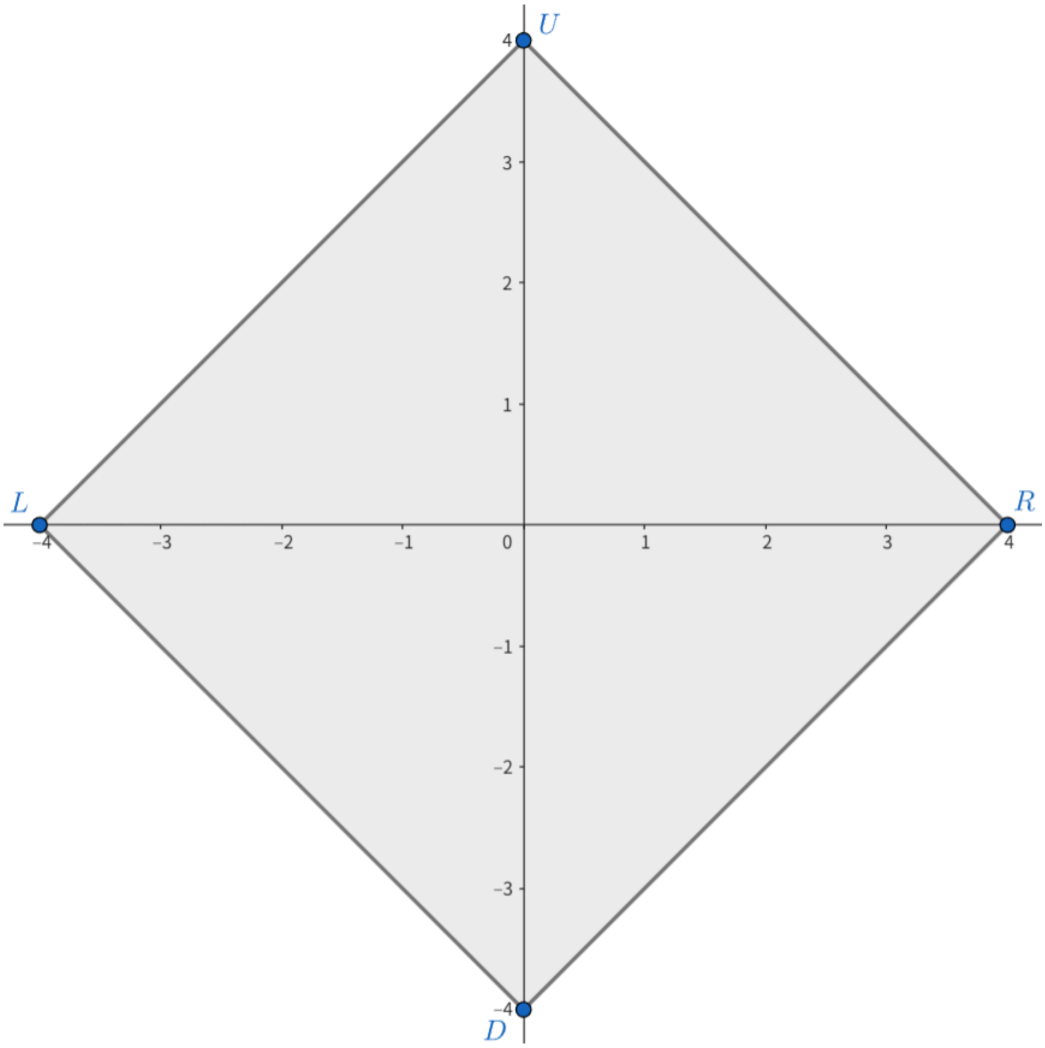
A. Draw a Square

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

The pink soldiers have given you 4 **distinct points** on the plane. The 4 points' coordinates are $(-l, 0)$, $(r, 0)$, $(0, -d)$, $(0, u)$ correspondingly, where l, r, d, u are positive integers.



In the diagram, a square is drawn by connecting the four points L, R, D, U.

Please determine if it is possible to draw a square* with the given points as its vertices.

---

* A *square* is defined as a polygon consisting of 4 vertices, of which all sides have equal length and all inner angles are equal. No two edges of the polygon may intersect each other.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains four integers l, r, d, u ($1 \le l, r, d, u \le 10$).

Output

For each test case, if you can draw a square using the four points, output "Yes". Otherwise, output "No".

You can output the answer in any case. For example, the strings "yEs", "yes", and "YES" will also be recognized as positive responses.

Standard Input	Standard Output
2 2 2 2 2 1 2 3 4	Yes No

Note

On the first test case, the four given points form a square, so the answer is "Yes".

On the second test case, the four given points do not form a square, so the answer is "No".

,

B. The Third Side

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

The pink soldiers have given you a sequence a consisting of n positive integers.

You must repeatedly perform the following operation **until there is only 1 element left**.

- Choose two **distinct** indices i and j.
- Then, choose a positive integer value x such that there exists a **non-degenerate triangle*** with side lengths a_i, a_j, and x.
- Finally, remove two elements a_i and a_j, and append x to the end of a.

Please find the maximum possible value of the only last element in the sequence a.

*A triangle with side lengths a, b, c is non-degenerate when $a + b > c$, $a + c > b$, $b + c > a$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1000$) — the elements of the sequence a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the maximum possible value of the only last element on a separate line.

Standard Input	Standard Output
4 1 10 3 998 244 353 5 1 2 3 4 5 9 9 9 8 2 4 4 3 5 3	10 1593 11 39

Note

On the first test case, there is already only one element. The value of the only last element is 10.

On the second test case, a is initially $[998, 244, 353]$. The following series of operations is valid:

1. Erase $a_2 = 244$ and $a_3 = 353$, and append 596 to the end of a. a is now $[998, 596]$.
2. Erase $a_1 = 998$ and $a_2 = 596$, and append 1593 to the end of a. a is now $[1593]$.

It can be shown that the only last element cannot be greater than 1593. Therefore, the answer is 1593.

,

C. XOR and Triangle

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

This time, the pink soldiers have given you an integer x ($x \geq 2$).

Please determine if there exists a **positive** integer y that satisfies the following conditions.

- y is **strictly** less than x.
- There exists a **non-degenerate triangle*** with side lengths x, y, $x \oplus y$. Here, $\oplus$ denotes the bitwise XOR operation.

Additionally, if there exists such an integer y, output any.

---

*A triangle with side lengths a, b, c is non-degenerate when $a + b > c$, $a + c > b$, $b + c > a$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 2000$). The description of the test cases follows.

The only line of each test case contains a single integer x ($2 \leq x \leq 10^9$).

Output

For each test case, print one integer on a separate line. The integer you must output is as follows:

- If there exists an integer y satisfying the conditions, output the value of y ($1 \leq y < x$);
- Otherwise, output -1.

If there exist multiple integers that satisfy the conditions, you may output any.

Standard Input	Standard Output
7 5 2 6 3 69 4 420	3 -1 5 -1 66 -1 320

Note

In the first test case, there exists a non-degenerate triangle with side lengths 3, 5, and $3 \oplus 5 = 6$. Therefore, $y = 3$ is a valid answer.

In the second test case, 1 is the only possible candidate for y, but it cannot make a non-degenerate triangle. Therefore, the answer is -1.

,

D. Counting Points

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

The pink soldiers drew n circles with their center **on the x-axis** of the plane. Also, they have told that **the sum of radii is exactly m^***.

Please find the number of integer points **inside or on the border of** at least one circle. Formally, the problem is defined as follows.

You are given an integer sequence $x_1, x_2, \ldots, x_n$ and a positive integer sequence $r_1, r_2, \ldots, r_n$, where it is known that $\sum_{i=1}^{n} r_i = m$.

You must count the number of integer pairs (x, y) that satisfy the following condition.

- There exists an index i such that $(x - x_i)^2 + y^2 \le r_i^2$ ($1 \le i \le n$).

---

*Is this information really useful? Don't ask me; I don't really know.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le m \le 2 \cdot 10^5$).

The second line of each test case contains $x_1, x_2, \ldots, x_n$ — the centers of the circles ($-10^9 \le x_i \le 10^9$).

The third line of each test case contains $r_1, r_2, \ldots, r_n$ — the radii of the circles ($1 \le r_i$, $\sum_{i=1}^{n} r_i = m$).

It is guaranteed that **the sum of m over all test cases** does not exceed $2 \cdot 10^5$.

Output

For each test case, output the number of integer points satisfying the condition on a separate line.

Standard Input	Standard Output
4 2 3 0 0 1 2 2 3 0 2 1 2 3 3 0 2 5 1 1 1 4 8 0 5 10 15 2 2 2 2	13 16 14 52

Note

On the first test case, the circle with $r_1 = 1$ is completely inside the circle with $r_2 = 2$. Therefore, you only have to count the number of integer points inside the latter. There are 13 integer points such that $x^2 + y^2 \le 2^2$, so the answer is 13.

On the second test case, the circle with $r_1 = 1$ is not completely inside the circle with $r_2 = 2$. There are 3 additional points that are inside the first circle but not inside the second circle, so the answer is $3 + 13 = 16$.

,

E. Empty Triangle

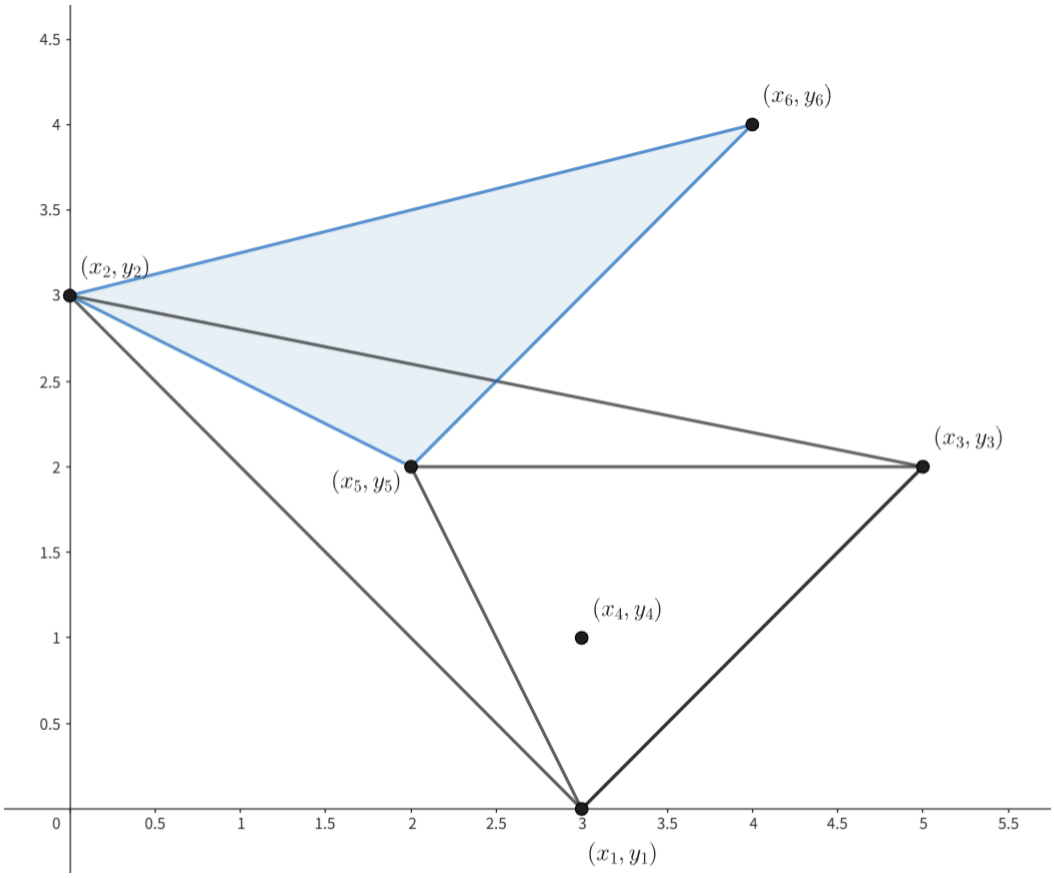
Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

This is an interactive problem.

The pink soldiers hid from you n $(3 \le n \le 1500)$ **fixed** points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, **whose coordinates are not given to you**. Also, it is known that no two points have the same coordinates, and no three points are collinear.

You can ask the Frontman about three **distinct** indices i, j, k. Then, he will draw a triangle with points (x_i, y_i), (x_j, y_j), (x_k, y_k), and respond with the following:

- If at least one of the hidden points lies inside the triangle, then the Frontman gives you the index of one such point. Do note that if there are multiple such points, **the Frontman can arbitrarily select one of them**.
- Otherwise, the Frontman responds with 0.



Your objective in this problem is to find a triangle not containing any other hidden point, such as the blue one in the diagram.

Using at most 75 queries, you must find any triangle formed by three of the points, **not containing** any other hidden point inside.

Do note that the Frontman may be **adaptive** while choosing the point to give you. In other words, the choice of the point can be determined by various factors including but not limited to the orientation of points and the previous queries. However, note that **the sequence of points will never be altered**.

Hacks are disabled for this problem. Your solution will be judged on exactly $\mathbf{35}$ input files, including the example input.

Interaction

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 20$). The description of the test cases follows.

The first line of each test case contains one positive integer n — the number of points ($3 \le n \le 1500$).

Afterwards, you can output the following on a separate line to ask a query:

- "`? i j k`" ($1 \le i, j, k \le n$, $i \ne j$, $i \ne k$, $j \ne k$): Ask about the triangle formed by (x_i, y_i), (x_j, y_j), (x_k, y_k).

Then, the interactor responds with one integer on a new line:

- If your query is invalid or the number of queries has exceeded $\mathbf{75}$, then the interactor responds with -1;
- If there exists an index p ($1 \le p \le n$) such that the point (x_p, y_p) is inside the triangle, then **any such index** p is given;
- Otherwise, the interactor responds with 0.

If you want to print an answer, you can output the following on a separate line:

- "`! i j k`" ($1 \le i, j, k \le n$, $i \ne j$, $i \ne k$, $j \ne k$): The triangle formed by (x_i, y_i), (x_j, y_j), (x_k, y_k) contains no other hidden point.

Then, the following interaction happens:

- If the answer is invalid or the triangle contains another hidden point, the interactor responds with -1;
- If the answer is correct and there are more test cases to solve, you are given the value of n for the next test case;
- Otherwise, it means that all test cases are solved correctly, and your program may terminate gracefully.

Do note that printing the answer does **not** count towards the number of queries made.

Do note that the interactor may be **adaptive** while choosing the point to give you. In other words, the choice of the point can be determined by various factors including but not limited to the orientation of points and the previous queries. However, note that **the sequence of points will never be altered**.

After printing each query do not forget to output the end of line and flush* the output. Otherwise, you will get `Idleness limit exceeded` verdict.

If, at any interaction step, you read -1 instead of valid data, your solution must exit immediately. This means that your solution will receive `Wrong answer` because of an invalid query or any other mistake. Failing to exit can result in an arbitrary verdict because your solution will continue to read from a closed stream.

Hacks

Hacks are disabled for this problem. Your solution will be judged on exactly $\mathbf{35}$ input files, including the example input.

---

*To flush, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `sys.stdout.flush()` in Python;

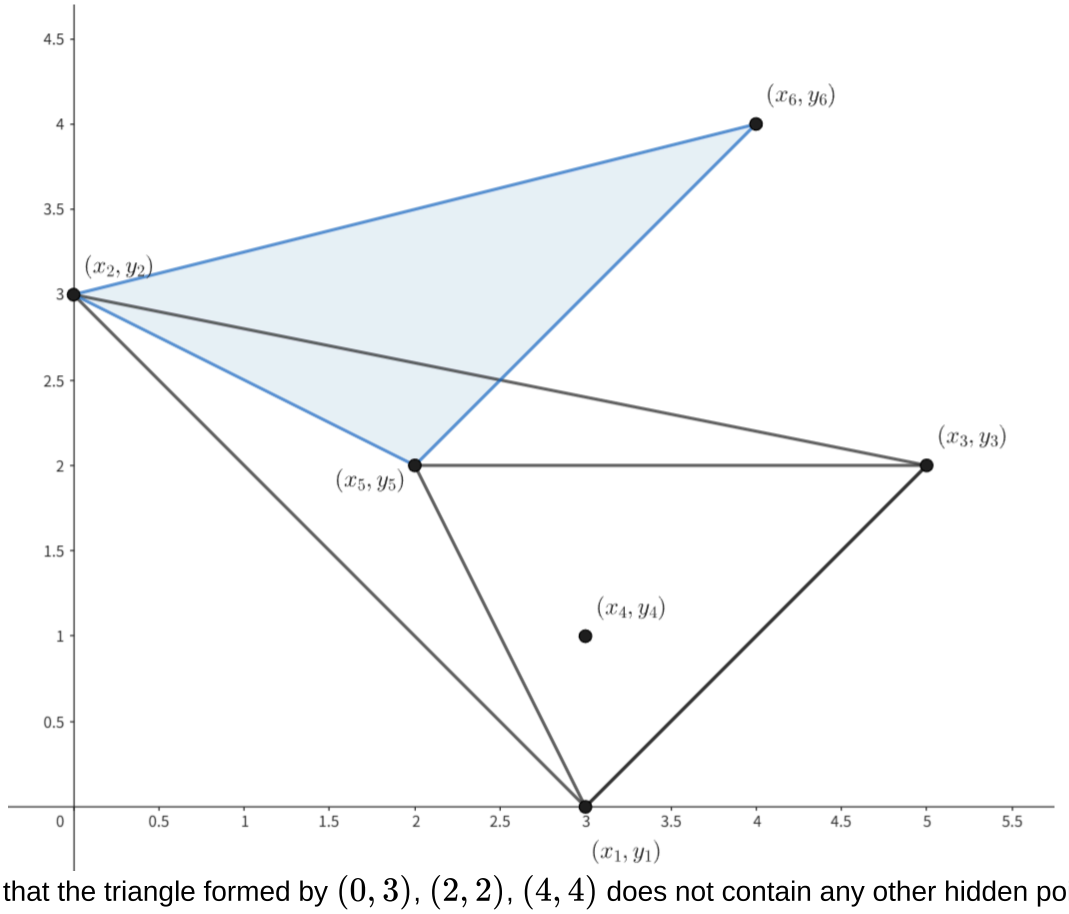
- see the documentation for other languages.

Standard Input	Standard Output
2	
6	? 1 2 3
5	? 1 5 3
4	? 2 5 6
0	! 2 5 6
3	! 1 2 3

Note

In the first test case, the points are $(3, 0), (0, 3), (5, 2), (3, 1), (2, 2), (4, 4)$.

The triangles corresponding to the three queries are as follows.



You can see that the triangle formed by $(0, 3)$, $(2, 2)$, $(4, 4)$ does not contain any other hidden point inside. Therefore, it is a valid answer.

Do note that the interaction example only shows a valid interaction, and **not necessarily the actual response**. For example, it is possible that the Frontman responds with 4 when you query "`? 1 2 3`". However, as the Frontman does not alter the sequence of points, he never responds with 6 for the same query.

In the second test case, there are only 3 points. Therefore, we know that the unique triangle formed by the points does not contain any other hidden point inside.

F. Counting Necessary Nodes

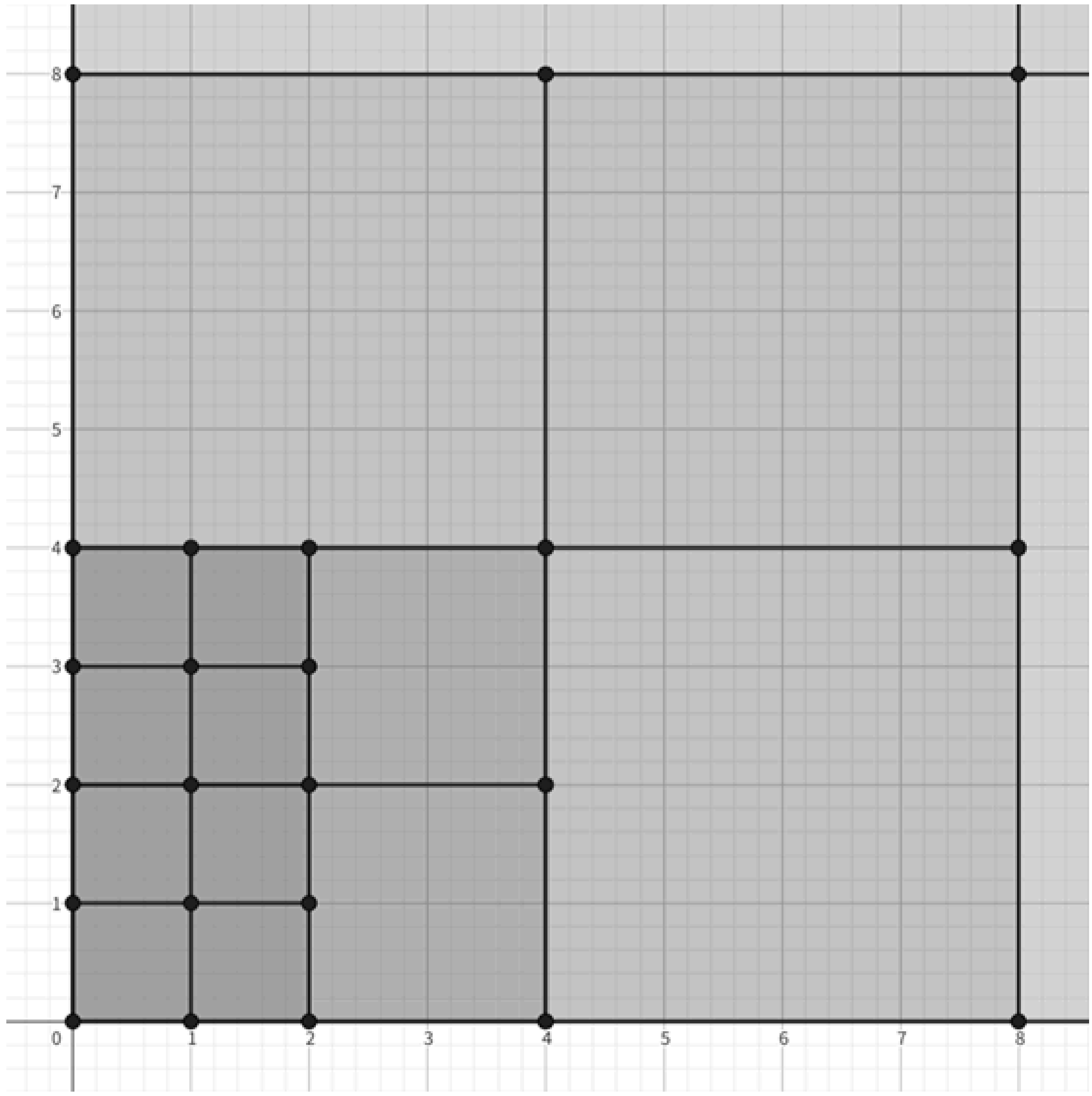
```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

A **quadtree** is a tree data structure in which each node has at most four children and accounts for a square-shaped region.

Formally, **for all tuples** of nonnegative integers $k, a, b \geq 0$, there exists **exactly one node** accounting for the following region*.

$$[a \cdot 2^k, (a+1) \cdot 2^k] \times [b \cdot 2^k, (b+1) \cdot 2^k]$$

All nodes whose region is larger than 1×1 contain four children corresponding to the regions divided equally into four, and the nodes whose region is 1×1 correspond to the leaf nodes of the tree.



A small subset of the regions accounted for by the nodes is shown. The relatively darker regions are closer to leaf nodes.

The Frontman hates the widespread misconception, such that the quadtree can perform range queries in $\mathcal{O}(\log n)$ time when there are n leaf nodes inside the region. In fact, sometimes it is necessary to query much more than $\mathcal{O}(\log n)$ regions for this, and the time complexity is $\mathcal{O}(n)$ in some extreme cases. Thus, the Frontman came up with this task to educate you about this worst case of the data structure.

The pink soldiers have given you a finite region $[l_1, r_1] \times [l_2, r_2]$, where l_i and r_i ($l_i < r_i$) are nonnegative integers. Please find the minimum number of nodes that you must choose in order to make the union of regions accounted for by the chosen nodes **exactly** the same as the given region. Here, two sets of points are considered different if there exists a point included in one but not in the other.

---

*Regions are sets of points **with real coordinates**, where the point (x, y) is included in the region $[p, q] \times [r, s]$ if and only if $p \le x \le q$ and $r \le y \le s$. Here, $\times$ formally refers to Cartesian product of sets.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The only line of each test case contains four integers l_1, r_1, l_2, r_2 — the boundaries of the region in each axis ($0 \le l_i < r_i \le 10^6$).

Output

For each test case, output the minimum number of nodes necessary to satisfy the condition on a separate line.

Standard Input	Standard Output
5 0 1 1 2 0 2 0 2 1 3 1 3 0 2 1 5 9 98 244 353	1 1 4 5 374

Note

On the first test case, the given region is $[0, 1] \times [1, 2]$. There is one node accounting for $[0, 1] \times [1, 2]$. Choosing this node, the answer is 1.

On the second test case, the given region is $[0, 2] \times [0, 2]$. There is one node accounting for $[0, 2] \times [0, 2]$. Choosing this node, the answer is 1.

On the third test case, the given region is $[1, 3] \times [1, 3]$. There is **no node** that accounts for $[1, 3] \times [1, 3]$. Instead, you can make the union of regions exactly the same as $[1, 3] \times [1, 3]$ by choosing the following 4 nodes:

- A leaf node accounting for $[1, 2] \times [1, 2]$;
- A leaf node accounting for $[1, 2] \times [2, 3]$;
- A leaf node accounting for $[2, 3] \times [1, 2]$;
- A leaf node accounting for $[2, 3] \times [2, 3]$.

It can be shown that it is impossible to make the union of regions exactly the same as $[1, 3] \times [1, 3]$ with less than 4 nodes. Therefore, the answer is 4.

On the fourth test case, the given region is $[0, 2] \times [1, 5]$. You can make the union of regions exactly the same as $[0, 2] \times [1, 5]$ by choosing the following 5 nodes:

- A leaf node accounting for $[0, 1] \times [1, 2]$;
- A leaf node accounting for $[1, 2] \times [1, 2]$;
- A **non-leaf** node accounting for $[0, 2] \times [2, 4]$;
- A leaf node accounting for $[0, 1] \times [4, 5]$;
- A leaf node accounting for $[1, 2] \times [4, 5]$.

It can be shown that it is impossible to make the union of regions exactly the same as $[0, 2] \times [1, 5]$ with less than 5 nodes. Therefore, the answer is 5.

G. Game With Triangles: Season 2

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   512 megabytes
```

The Frontman greets you to this final round of the survival game.

There is a regular polygon with n sides ($n \geq 3$). The vertices are indexed as $1, 2, \ldots, n$ in clockwise order. On each vertex i, the pink soldiers have written a positive integer a_i. With this regular polygon, you will play a game defined as follows.

Initially, your score is 0. Then, you perform the following operation any number of times to increase your score.

- Select 3 different vertices i, j, k that you **have not chosen** before, and draw the triangle formed by the three vertices.
  - Then, your score increases by $a_i \cdot a_j \cdot a_k$.
  - However, you **can not perform this operation** if the triangle shares a positive common area with any of the triangles drawn previously.

An example of a state after two operations is on the left. The state on the right **is impossible** as the two triangles share a positive common area.

Your objective is to maximize the score. Please find the maximum score you can get from this game.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n — the number of vertices ($3 \leq n \leq 400$).

The second line of each test case contains $a_1, a_2, \ldots, a_n$ — the integers written on vertices ($1 \leq a_i \leq 1000$).

It is guaranteed that the sum of n^3 over all test cases does not exceed 400^3.

Output

For each test case, output the maximum score you can get on a separate line.

Standard Input	Standard Output
6 3 1 2 3 4 2 1 3 4 6 2 1 2 1 1 1 6 1 2 1 3 1 5 9 9 9 8 2 4 4 3 5 3 9 9 9 3 2 4 4 8 5 3	6 24 5 30 732 696

Note

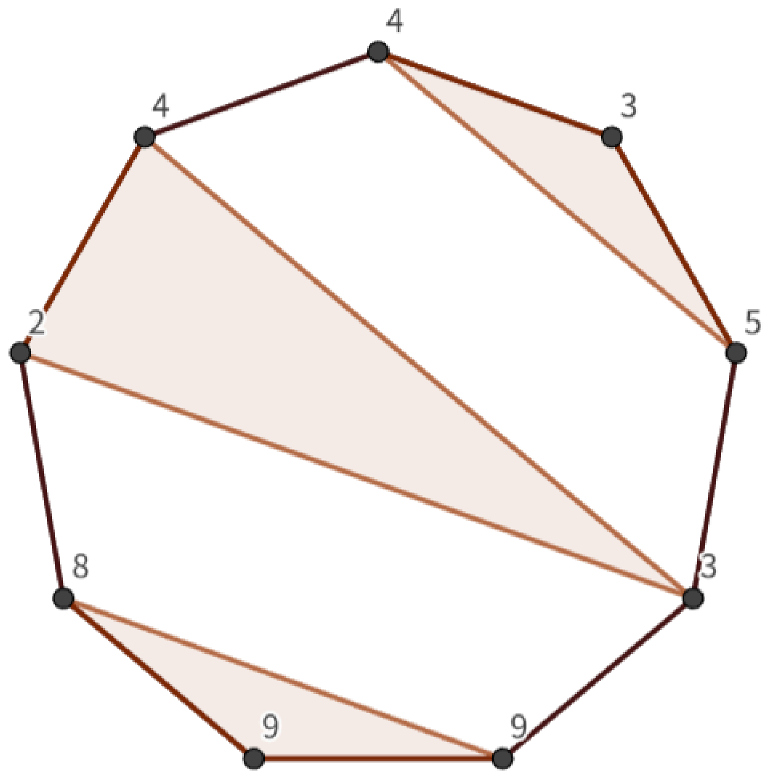
On the first test case, you can draw only one triangle. The maximum score 6 is found by drawing the triangle with $i = 1$, $j = 2$, $k = 3$.

On the second test case, you can draw only one triangle. The maximum score 24 is found by drawing the triangle with $i = 1$, $j = 3$, $k = 4$.

On the third test case, you can draw two triangles. There is a series of two operations that leads to the score 5, which is the maximum.

On the fourth test case, you can draw two triangles. However, drawing two triangles leads to a score of either $6 + 5 = 11$, $15 + 2 = 17$, or $10 + 3 = 13$. The maximum score 30 is found by drawing **only one** triangle with $i = 2$, $j = 4$, $k = 6$.

On the fifth test case, you can draw three triangles. The maximum score 732 is found by drawing three triangles as follows.


4
4
3
2
5
8
3
9
9