A. Mezo Playing Zoma

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today, Mezo is playing a game. Zoma, a character in that game, is initially at position $x = 0$. Mezo starts sending n commands to Zoma. There are two possible commands:

- 'L' (Left) sets the position $x := x - 1$;
- 'R' (Right) sets the position $x := x + 1$.

Unfortunately, Mezo's controller malfunctions sometimes. Some commands are sent successfully and some are ignored. If the command is ignored then the position x doesn't change and Mezo simply proceeds to the next command.

For example, if Mezo sends commands "LRLR", then here are some possible outcomes (underlined commands are sent successfully):

- "<u>LRLR</u>" — Zoma moves to the left, to the right, to the left again and to the right for the final time, ending up at position 0;
- "LRLR" — Zoma recieves no commands, doesn't move at all and ends up at position 0 as well;
- "<u>L</u>R<u>L</u>R" — Zoma moves to the left, then to the left again and ends up in position -2.

Mezo doesn't know which commands will be sent successfully beforehand. Thus, he wants to know how many different positions may Zoma end up at.

Input

The first line contains n $(1 \le n \le 10^5)$ — the number of commands Mezo sends.

The second line contains a string s of n commands, each either 'L' (Left) or 'R' (Right).

Output

Print one integer — the number of different positions Zoma may end up at.

Example

input

```
4
LRLR
```

output

```
5
```

Note

In the example, Zoma may end up anywhere between -2 and 2.

B. Just Eat It!

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today, Yasser and Adel are at the shop buying cupcakes. There are n cupcake types, arranged from 1 to n on the shelf, and there are infinitely many of each type. The tastiness of a cupcake of type i is an integer a_i. There are both tasty and nasty cupcakes, so the tastiness can be positive, zero or negative.

Yasser, of course, wants to try them all, so he will buy exactly one cupcake of each type.

On the other hand, Adel will choose some segment $[l, r]$ $(1 \le l \le r \le n)$ that does not include all of cupcakes (he can't choose $[l, r] = [1, n]$) and buy exactly one cupcake of each of types $l, l + 1, \dots, r$.

After that they will compare the total tastiness of the cupcakes each of them have bought. Yasser will be happy if the total tastiness of cupcakes he buys is **strictly** greater than the total tastiness of cupcakes Adel buys **regardless of Adel's choice**.

For example, let the tastinesses of the cupcakes be $[7, 4, -1]$. Yasser will buy all of them, the total tastiness will be $7 + 4 - 1 = 10$. Adel can choose segments $[7], [4], [-1], [7, 4]$ or $[4, -1]$, their total tastinesses are $7, 4, -1, 11$ and 3, respectively. Adel can choose segment with tastiness 11, and as 10 is not strictly greater than 11, Yasser won't be happy :(

Find out if Yasser will be happy after visiting the shop.

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 n ($2 \le n \le 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($-10^9 \le a_i \le 10^9$), where a_i represents the tastiness of the i-th type of cupcake.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5.

Output

For each test case, print "YES", if the total tastiness of cupcakes Yasser buys will always be **strictly** greater than the total tastiness of cupcakes Adel buys regardless of Adel's choice. Otherwise, print "NO".

Example

input

```
3
4
1 2 3 4
3
7 4 -1
3
5 -5 5
```

output

```
YES
NO
NO
```

Note

In the first example, the total tastiness of any segment Adel can choose is less than the total tastiness of all cupcakes.

In the second example, Adel will choose the segment $[1, 2]$ with total tastiness 11, which is not less than the total tastiness of all cupcakes, which is 10.

In the third example, Adel can choose the segment $[3, 3]$ with total tastiness of 5. Note that Yasser's cupcakes' total tastiness is also 5, so in that case, the total tastiness of Yasser's cupcakes isn't strictly greater than the total tastiness of Adel's cupcakes.

C. Fadi and LCM

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today, Osama gave Fadi an integer X, and Fadi was wondering about the minimum possible value of $max(a, b)$ such that $LCM(a, b)$ equals X. Both a and b should be positive integers.

$LCM(a, b)$ is the smallest positive integer that is divisible by both a and b. For example, $LCM(6, 8) = 24$, $LCM(4, 12) = 12$, $LCM(2, 3) = 6$.

Of course, Fadi immediately knew the answer. Can you be just like Fadi and find any such pair?

Input

The first and only line contains an integer X ($1 \le X \le 10^{12}$).

Output

Print two positive integers, a and b, such that the value of $max(a, b)$ is minimum possible and $LCM(a, b)$ equals X. If there are several possible such pairs, you can print any.

Examples

input

```
2
```

output

```
1 2
```

input

```
6
```

output

```
2 3
```

input

```
4
```

output

```
1 4
```

input

```
1
```

output

```
1 1
```

D. Dr. Evil Underscores

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today, as a friendship gift, Bakry gave Badawy n integers $a_1, a_2, \dots, a_n$ and challenged him to choose an integer X such that the value $\underset{1 \le i \le n}{\max} (a_i \oplus X)$ is minimum possible, where $\oplus$ denotes the bitwise XOR operation.

As always, Badawy is too lazy, so you decided to help him and find the minimum possible value of $\underset{1 \le i \le n}{\max} (a_i \oplus X)$.

Input

The first line contains integer n ($1 \le n \le 10^5$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 2^{30} - 1$).

Output

Print one integer — the minimum possible value of $\underset{1 \le i \le n}{\max} (a_i \oplus X)$.

Examples

input

```
3
1 2 3
```

output

```
2
```

input

```
2
1 5
```

output

```
4
```

Note

In the first sample, we can choose $X = 3$.

In the second sample, we can choose $X = 5$.

E. Delete a Segment

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n segments on a Ox axis $[l_1, r_1]$, $[l_2, r_2]$, ..., $[l_n, r_n]$. Segment $[l, r]$ covers all points from l to r inclusive, so all x such that $l \le x \le r$.

Segments can be placed *arbitrarily* — be inside each other, coincide and so on. Segments can degenerate into points, that is $l_i = r_i$ is possible.

Union of the set of segments is such a set of segments which covers exactly the same set of points as the original set. For example:

- if $n = 3$ and there are segments $[3, 6]$, $[100, 100]$, $[5, 8]$ then their union is 2 segments: $[3, 8]$ and $[100, 100]$;
- if $n = 5$ and there are segments $[1, 2]$, $[2, 3]$, $[4, 5]$, $[4, 6]$, $[6, 6]$ then their union is 2 segments: $[1, 3]$ and $[4, 6]$.

Obviously, a union is a set of pairwise non-intersecting segments.

You are asked to erase exactly one segment of the given n so that the number of segments in the union of the rest $n - 1$ segments is maximum possible.

For example, if $n = 4$ and there are segments $[1, 4]$, $[2, 3]$, $[3, 6]$, $[5, 7]$, then:

- erasing the first segment will lead to $[2, 3]$, $[3, 6]$, $[5, 7]$ remaining, which have 1 segment in their union;
- erasing the second segment will lead to $[1, 4]$, $[3, 6]$, $[5, 7]$ remaining, which have 1 segment in their union;
- erasing the third segment will lead to $[1, 4]$, $[2, 3]$, $[5, 7]$ remaining, which have 2 segments in their union;
- erasing the fourth segment will lead to $[1, 4]$, $[2, 3]$, $[3, 6]$ remaining, which have 1 segment in their union.

Thus, you are required to erase the third segment to get answer 2.

Write a program that will find the maximum number of segments in the union of $n - 1$ segments if you erase any of the given n segments.

Note that if there are multiple equal segments in the given set, then you can erase only one of them anyway. So the set after erasing will have exactly $n - 1$ segments.

Input

The first line contains one integer t ($1 \le t \le 10^4$) — the number of test cases in the test. Then the descriptions of t test cases follow.

The first of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of segments in the given set. Then n lines follow, each contains a description of a segment — a pair of integers l_i, r_i ($-10^9 \le l_i \le r_i \le 10^9$), where l_i and r_i are the coordinates of the left and right borders of the i-th segment, respectively.

The segments are given in an arbitrary order.

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

Output

Print t integers — the answers to the t given test cases in the order of input. The answer is the maximum number of segments in the union of $n - 1$ segments if you erase any of the given n segments.

Example

input

```
3
4
1 4
2 3
3 6
5 7
3
5 5
5 5
5 5
6
3 3
1 1
5 5
1 5
2 2
4 4
```

output

```
2
1
5
```

F. Classical?

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given an array a, consisting of n integers, find:

$$\max_{1 \le i < j \le n} LCM(a_i, a_j),$$

where $LCM(x, y)$ is the smallest positive integer that is divisible by both x and y. For example, $LCM(6, 8) = 24$, $LCM(4, 12) = 12$, $LCM(2, 3) = 6$.

Input

The first line contains an integer n ($2 \le n \le 10^5$) — the number of elements in the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) — the elements of the array a.

Output

Print one integer, the maximum value of the least common multiple of two elements in the array a.

Examples

input

```
3
13 35 77
```

output

```
1001
```

input

```
6
1 2 4 8 16 32
```

output

```
32
```