A. MUH and Sticks

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

Two polar bears Menshykov and Uslada from the St.Petersburg zoo and elephant Horace from the Kiev zoo got six sticks to play with and assess the animals' creativity. Menshykov, Uslada and Horace decided to make either an elephant or a bear from those sticks. They can make an animal from sticks in the following way:

- Four sticks represent the animal's legs, these sticks should have the same length.
- Two remaining sticks represent the animal's head and body. The bear's head stick must be shorter than the body stick. The elephant, however, has a long trunk, so his head stick must be as long as the body stick. Note that there are no limits on the relations between the leg sticks and the head and body sticks.

Your task is to find out which animal can be made from the given stick set. The zoo keeper wants the sticks back after the game, so they must never be broken, even bears understand it.

Input

The single line contains six space-separated integers l_i ($1 \le l_i \le 9$) — the lengths of the six sticks. It is guaranteed that the input is such that you cannot make both animals from the sticks.

Output

If you can make a bear from the given set, print string "`Bear`" (without the quotes). If you can make an elephant, print string "`Elephant`" (without the quotes). If you can make neither a bear nor an elephant, print string "`Alien`" (without the quotes).

Examples

input
```
4 2 5 4 4 4
```
output
```
Bear
```

input
```
4 4 5 4 4 5
```
output
```
Elephant
```

input
```
1 2 3 4 5 6
```
output
```
Alien
```

Note

If you're out of creative ideas, see instructions below which show how to make a bear and an elephant in the first two samples. The stick of length 2 is in red, the sticks of length 4 are in green, the sticks of length 5 are in blue.

B. MUH and Important Things

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

It's time polar bears Menshykov and Uslada from the zoo of St. Petersburg and elephant Horace from the zoo of Kiev got down to business. In total, there are n tasks for the day and each animal should do each of these tasks. For each task, they have evaluated its difficulty. Also animals decided to do the tasks in order of their difficulty. Unfortunately, some tasks can have the same difficulty, so the order in which one can perform the tasks may vary.

Menshykov, Uslada and Horace ask you to deal with this nuisance and come up with individual plans for each of them. The plan is a sequence describing the order in which an animal should do all the n tasks. Besides, each of them wants to have its own unique plan. Therefore three plans must form three different sequences. You are to find the required plans, or otherwise deliver the sad news to them by stating that it is impossible to come up with three distinct plans for the given tasks.

Input

The first line contains integer n ($1 \le n \le 2000$) — the number of tasks. The second line contains n integers $h_1, h_2, ..., h_n$ ($1 \le h_i \le 2000$), where h_i is the difficulty of the i-th task. The larger number h_i is, the more difficult the i-th task is.

Output

In the first line print "YES" (without the quotes), if it is possible to come up with three distinct plans of doing the tasks. Otherwise print in the first line "NO" (without the quotes). If three desired plans do exist, print in the second line n distinct integers that represent the numbers of the tasks in the order they are done according to the first plan. In the third and fourth line print two remaining plans in the same form.

If there are multiple possible answers, you can print any of them.

Examples

input

```
4
1 3 3 1
```

output

```
YES
1 4 2 3
4 1 2 3
4 1 3 2
```

input

```
5
2 4 1 4 8
```

output

```
NO
```

Note

In the first sample the difficulty of the tasks sets one limit: tasks 1 and 4 must be done before tasks 2 and 3. That gives the total of four possible sequences of doing tasks : [1, 4, 2, 3], [4, 1, 2, 3], [1, 4, 3, 2], [4, 1, 3, 2]. You can print any three of them in the answer.

In the second sample there are only two sequences of tasks that meet the conditions — [3, 1, 2, 4, 5] and [3, 1, 4, 2, 5]. Consequently, it is impossible to make three distinct sequences of tasks.

C. MUH and House of Cards

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

Polar bears Menshykov and Uslada from the zoo of St. Petersburg and elephant Horace from the zoo of Kiev decided to build a house of cards. For that they've already found a hefty deck of n playing cards. Let's describe the house they want to make:

1. The house consists of some non-zero number of floors.
2. Each floor consists of a non-zero number of rooms and the ceiling. A room is two cards that are leaned towards each other. The rooms are made in a row, each two adjoining rooms share a ceiling made by another card.
3. Each floor besides for the lowest one should contain less rooms than the floor below.

Please note that the house may end by the floor with more than one room, and in this case they also must be covered by the ceiling. Also, the number of rooms on the adjoining floors doesn't have to differ by one, the difference may be more.

While bears are practicing to put cards, Horace tries to figure out how many floors their house should consist of. The height of the house is the number of floors in it. It is possible that you can make a lot of different houses of different heights out of n cards. It seems that the elephant cannot solve this problem and he asks you to count the number of the distinct heights of the houses that they can make using **exactly** n cards.

Input

The single line contains integer n ($1 \le n \le 10^{12}$) — the number of cards.

Output

Print the number of distinct heights that the houses made of exactly n cards can have.

Examples

input
13
output
1

input
6
output
0

Note

In the first sample you can build only these two houses (remember, you must use all the cards):

Thus, 13 cards are enough only for two floor houses, so the answer is 1.

The six cards in the second sample are not enough to build any house.

D. MUH and Cube Walls

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polar bears Menshykov and Uslada from the zoo of St. Petersburg and elephant Horace from the zoo of Kiev got hold of lots of wooden cubes somewhere. They started making cube towers by placing the cubes one on top of the other. They defined multiple towers standing in a line as a wall. A wall can consist of towers of different heights.

Horace was the first to finish making his wall. He called his wall an elephant. The wall consists of w towers. The bears also finished making their wall but they didn't give it a name. Their wall consists of n towers. Horace looked at the bears' tower and wondered: in how many parts of the wall can he "see an elephant"? He can "see an elephant" on a segment of w contiguous towers if the heights of the towers on the segment match as a sequence the heights of the towers in Horace's wall. In order to see as many elephants as possible, Horace can raise and lower his wall. He even can lower the wall below the ground level (see the pictures to the samples for clarification).

Your task is to count the number of segments where Horace can "see an elephant".

Input

The first line contains two integers n and w ($1 \le n, w \le 2 \cdot 10^5$) — the number of towers in the bears' and the elephant's walls correspondingly. The second line contains n integers a_i ($1 \le a_i \le 10^9$) — the heights of the towers in the bears' wall. The third line contains w integers b_i ($1 \le b_i \le 10^9$) — the heights of the towers in the elephant's wall.

Output

Print the number of segments in the bears' wall where Horace can "see an elephant".

Examples

input

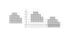
```
13 5
2 4 5 5 4 3 2 2 2 3 3 2 1
3 4 4 3 2
```

output

```
2
```

Note

The picture to the left shows Horace's wall from the sample, the picture to the right shows the bears' wall. The segments where Horace can "see an elephant" are in gray.



E. MUH and Lots and Lots of Segments

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Polar bears Menshykov and Uslada from the zoo of St. Petersburg and elephant Horace from the zoo of Kiev decided to do some painting. As they were trying to create their first masterpiece, they made a draft on a piece of paper. The draft consists of n segments. Each segment was either horizontal or vertical. Now the friends want to simplify the draft by deleting some segments or parts of segments so that the final masterpiece meets three conditions:

1. Horace wants to be able to paint the whole picture in one stroke: by putting the brush on the paper and never taking it off until the picture is ready. The brush can paint the same place multiple times. That's why all the remaining segments must form a single connected shape.
2. Menshykov wants the resulting shape to be simple. He defines a simple shape as a shape that doesn't contain any cycles.
3. Initially all the segment on the draft have integer startpoint and endpoint coordinates. Uslada doesn't like real coordinates and she wants this condition to be fulfilled after all the changes.

As in other parts the draft is already beautiful, the friends decided to delete such parts of the draft that the sum of lengths of the remaining segments is as large as possible. Your task is to count this maximum sum of the lengths that remain after all the extra segments are removed.

Input

The first line of the input contains integer n ($1 \le n \le 2 \cdot 10^5$) — the number of segments on the draft. The next n lines contain four integers each: x_1, y_1, x_2, y_2 $(-10^9 \le x_1 \le x_2 \le 10^9;\ -10^9 \le y_1 \le y_2 \le 10^9)$ — the two startpoint and the two endpoint coordinates of a segment. All segments are non-degenerative and either are strictly horizontal or strictly vertical.

No two horizontal segments share common points. No two vertical segments share common points.

Output

Print a single integer — the maximum sum of lengths for the remaining segments.

Examples

input

```
2
0 0 0 1
1 0 1 1
```

output

```
1
```

input

```
4
0 0 1 0
0 0 0 1
1 -1 1 2
0 1 1 1
```

output

```
5
```

Note

The shapes that you can get in the two given samples are:

In the first sample you need to delete any segment as the two segments together do not form a single connected shape.

In the second sample the initial segments form a cycle, there are four ways to break the cycle: delete the first, second or fourth segment altogether or delete the middle of the third segment. The last way is shown on the picture.