

## Calvin's Stars

---

Calvin likes to lie in a field and look at the night sky. Since he does not know any real star constellations, he makes them up: if two stars are close to each other, they must belong to the same constellation. He wants to name them all, but fears to run out of names. Can you help him and count how many constellations there are in the sky?

Two stars belong to the same constellation if distance between their projections on a two-dimensional sky plan isn't more than D units.

### Input

The first line contains the number of stars  $N$  (  $0 \leq N \leq 1,000$  ) and the distance  $D$  (a real number  $0.0 \leq D \leq 1,000.00$  ). Next  $N$  lines have a pair of real coordinates  $X Y$  (where  $-1,000.00 \leq X, Y \leq 1,000.00$  ) for each star in the sky. All real numbers in the input will have at most 2 (two) digits after a decimal point.

### Output

Output the **number  $N$**  that is the number of constellations in Calvin's sky.

### Example 1

```
Input:
5 1.5
1.0 0.1
2.0 0.0
5.0 0.2
6.0 0.4
3.0 -0.1
```

```
Output:
2
```

### Example 2

```
Input:
3 4.0
121.12 254.06
645.04 301.85
912.49 568.96
```

```
Output:
3
```

# Virus

Due to the outbreak of unknown virus in 2029 in the mountains of Timbet, all the monks at Sempou Temple are required to do physical checkups at Iosefka's Clinic. All checkups are scheduled on the same day. Each monk gets instructions in which he is given

1. his unique patient number from the set  $\{1 \dots n\}$
2. the time of the day when he is supposed to show up at Iosefka's Clinic
3. a list of doctors' offices that he is to visit in the specified order

Doctors' offices in Iosefka's Clinic are numbered with numbers from the set  $\{1 \dots m\}$ .

Since a doctor can only check one monk at a time, if several people show up a doctor's office at time  $t$ , they form a queue in increasing order of their numbers and join the end of the queue already formed by monks who arrived earlier.

If at time  $t$  in front of office  $x$  there is a queue of people who arrived earlier or at time  $t$ , then the first monk from the queue enters office  $x$ . This monk exits the office after one time unit and at time  $t+1$  appears at the next office from his list of offices to visit. At that time the next person from the queue enters office  $x$ .

If a monk was supposed to show up at the clinic at time  $t$ , then at time  $t$  he shows up at the first doctors' office on his list. If a visit at office  $x$  at time  $t$  was for the given monk the last visit on his list, then at time  $t+1$  this monk leaves the clinic.

Your task is to find the time when the last monk leaves the Iosefka's Clinic.

## Input

The first line of input contains 2 natural numbers  $n$  and  $m$ ,  $1 \leq n, m \leq 10000$ , giving the number of the monks and the number of doctors' offices. Each of the following  $n$  lines contains a sequence of natural numbers. Among these lines, line  $i$  ( $1 \leq i \leq n$ ) has the following format

$t \ k \ g_1 \ g_2 \ \dots \ g_k$

meaning that the  $i$ th monk arrives at time  $t$  and has to visit  $k$  offices in the order given by  $g_1 \ g_2 \ \dots \ g_k$  where each  $g_j$  is a number of doctor's office,  $1 \leq g_j \leq m$ . We have that  $0 \leq t \leq 1,000,000$  and there is no more than 1,000,000 visits scheduled for a day at the clinic.

## Output

Print one line giving the time when the last monk leaves the hospital

### Example 1

Input :

```
5 3
1 3 3 2 1
0 7 2 3 1 1 1 1 2
2 1 1
1 2 3 3
4 3 1 1 1
```

Output :

12

### Example 2

Input :

```
5 10
3 1 6
2 3 3 2 8
2 1 4
```

```
2 4 7 9 9 6
0 2 8 7
```

Output:  
6

## Math Class

---

Kou is having her first math class. Her teacher gave her  $N$  positive integers and asked her to build a largest possible number. The only allowed operation is to concatenate the numbers one after another. Kou can reorganize the numbers as she sees fit, but she cannot remove or rearrange the digits within the numbers.

However, Kou got stuck on this problem. She turns to you for help.

### Input

The first line of the input contains a single integer  $N$  ( $1 \leq N \leq 50$ ), indicating the number of integers. The next line contains  $N$  positive integers between 1 and  $10^{100}$ .

### Output

You should print a single integer, the largest number that can be created by concatenating the given numbers in any order.

### Example 1

Input:

4  
123 124 56 90

Output:

9056124123

### Example 2

Input:

5  
123 124 56 90 9

Output:

99056124123

### Example 3

Input:

5  
9 9 9 9 9

Output:

99999

# Sia's Box

---

Sia has a mysterious box and she wants to play a game with you. The box supports two types of operations:

- **1 x** : Put the number x into the box.
- **2** : Take out a number from the box.

Sia will give you a sequence of operations and the results of **2** operations. Your task is to determine what is really hidden in the box: a stack, a queue, a max priority queue or something else.

## Input

The first line of the input contains a single integer N ( $1 \leq N \leq 1000$ ), indicating the number of operations. Each of the next N lines contains a single operation described above. For operation **2**, there is an additional number x indicating the result of that operation. The value of x in both operations satisfies  $1 \leq x \leq 100$ .

## Output

You should print:

- **stack** if it is certain that the box is a stack.
- **queue** if it is certain that the box is a queue.
- **priority queue** if it is certain that the box is a max priority queue.
- **impossible** if it is certain that the box cannot be any of those three data structures.
- **not sure** if the box could possibly be more than one of those three data structures.

## Example 1

Input :

```
6
1 1
1 2
1 3
2 1
2 2
2 2
2 3
```

Output :

queue

## Example 2

Input :

```
6
1 1
1 2
1 3
2 3
2 2
2 2
2 1
```

Output :

not sure

## Example 3

Input:

2  
1 1  
2 2

Output:

impossible

#### Example 4

Input:

4  
1 2  
1 1  
2 1  
2 2

Output:

stack

#### Example 5

Input:

7  
1 2  
1 5  
1 1  
1 3  
2 5  
1 4  
2 4

Output:

priority queue

# Daily Prize

---

Trader Jane's has a new promotion to attract even more customers to already crowded supermarket.

To participate in a daily drawing of the prize a customer needs to fill out a form with their name, phone number and the amount that they paid for their groceries.

At the end of the day, Trader Jane's manager picks two forms from among the completed ones:

- first is the one that has the highest amount paid
- second is the one that has the lowest amount paid The person who paid the highest amount gets the prize equal to the difference between their bill and the lowest bill.

Given how busy Trader Jane's supermarket is, you can be certain that at the end of each day there are at least two bills to select (usually there are many many more).

The selected forms are discarded, but the other ones remain in the pool for the next day, so each customer has a chance to be selected as the prize winner on the day of their purchase, or any day after.

Your task is to compute how much money Trader Jane's pays out in prizes.

## Input

The input contains an integer  $n$ ,  $1 \leq n \leq 5,000$  on the first line - this is the number of days in the promotion. Each of the next  $n$  lines contains a sequence of non-negative integers separated by whitespace. The first number on each line indicates the number of forms submitted on that day,  $0 \leq k \leq 100,000$ . The next  $k$  numbers specify the bill amounts on the forms entered for the daily drawing on that day. Each amount is guaranteed to be no larger than 1.000.000.

## Output

Print one number that is the sum of all the prizes that Trader Jane's pays out during the promotion followed by a newline.

### Example 1

Input :

```
5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2
```

Output :

```
19
```

### Example 2

Input :

```
2
2 1 2
2 1 2
```

Output :

```
2
```