Assignment 2

Deadline: Deadline: 10pm, Monday, April 13, 2020

- You can submit your solution to the OJ system for multiple times.
- You can discuss with your classmates about the assignment but do NOT copy others' solutions. The OJ system can detect the duplication. *If confirmed that there is a plagiarism, both parties will get 0 mark for this assignment as a penalty.*

Problem A

Description

A rectangle is represented as a list $[x1, y1, x2, y2]$, where $(x1, y1)$ are the coordinates of its bottom–left corner, and $(x2, y2)$ are the coordinates of its top–right corner.

Two rectangles overlap if the area of their intersection is positive. To be clear, two rectangles that only touch at the corner or edges do not overlap.

Given many rectangles, return whether they overlap.

Input

The first line is an integer n, which means their are n pairs rectangles.
Next n lines contain the coordinates of two rectangles.

Output

If two rectangles overlap, you should print **True**. If not, you should print **False**

Sample input

```
2
0 0 2 2 1 1 3 3
0 0 1 1 1 0 2 1
```

Sample output

```
True
False
```

Problem B

Description

You are given two jugs with capacities x and y litres. There is an infinite amount of water supply available. You need to determine whether it is possible to measure exactly z litres using these two jugs.

If z liters of water is measurable, you must have z liters of water contained within one or both buckets by the end.

Operations allowed:

- Fill any of the jugs completely with water.
- Empty any of the jugs.
- Pour water from one jug into another till the other jug is completely full or the first jug itself is empty.

Input

The first line is an integer n, which means their are n different cases.
Next n lines contain the capacities of three jugs (x, y, z).

Output

If you can left z liters of water in jugs, you should print **True**. If not, you should print **False**

Sample input

```
2
3 5 4
2 6 5
```

Sample output

```
True
False
```

Problem C - Triangle

Description

Three points that are **not** on the same line can construct a **triangle** by connecting with each other.

There are many points in a **3-D space**, and they can construct many triangles.

You need find the **largest** one among all possible triangles, and tell its **area**.

Take it easy.

Enjoy.

Input

Using `System.in` to get

The first line contains an integer **n**, which indicates n points totally. ($0 \leq n \leq 1000$)

Then following n lines, each line contains 3 integers: `x y z`, which is the **coordinates** of a point.

$(-10^6 \leq x \leq 10^6, -10^6 \leq y \leq 10^6, -10^6 \leq z \leq 10^6)$

A sample:

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

Output

You only need to print the **area** of the **largest** triangle among all possible triangle constructed by the points given in input.

If **no triangle** can be constructed, please **do not** print anything.

Note: Round to **two** decimal.

A sample:

```
123.45
```

More examples

Input

```
1
1 1 1
```

Output

Nothing

Input

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

Output

Nothing

Input

```
3
0 0 0
```

```
3 0 0
0 4 0
```

Output

```
6.00
```

Input

```
4
0 0 0
0 0 1
0 1 0
1 1 0
```

Output

```
0.71
```

Problem D - Spiral encryption

Description

ZS has received a piece of confidential information from M78 nebula, he should write a letter to inform his superior Yanyan as soon as possible. For security, the content should be encrypted by **spiral encryption**!

But ZS need to heal the world, so have no time to encrypt it. What to do?

"**What can I do for you?**" you say.

Input

Using `System.in` to get

The input only have **one** line, which contains a long **string**.

The string consists of **numbers**, **white-space** character and **English letters**.

Like this:

```
sadADASDkjlka 123
```

Output

- First of all, you should **remove** all white-spaces in input string.

  Only remaining characters need encrypt.

  Original string:

  ```
  123 45 6 7
  ```

  After removing:

  ```
  1234567
  ```

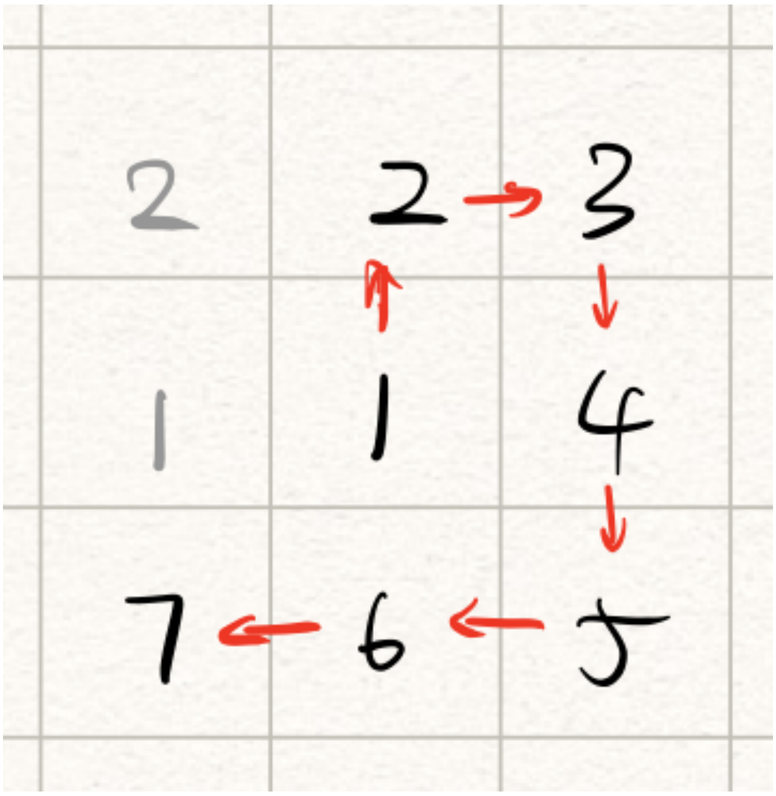
- Then, you should print the remaining string in a **spiral way**, and always **start toward up**, like this:

  For string:

  ```
  1234567
  ```

  Spiral print:

  ```
  23
  14
  765
  ```



2 2 3
1 1 4
7 6 5

- Finally, for aesthetics, we want to make it a rectangle. So we fill it with the leading characters of the string.

  After filling:

  ```
  223
  114
  765
  ```

And this is what you should output finally.

More examples

Input

```
a
```

Output

```
a
```

Input

```
a 3
```

Output

```
3
a
```

Input

```
a 2 g
```

Output

```
2g
aa
```

Input

```
a1b             2c 3            d4e5f6            z
```

Output

```
5f6z
e1ba
4a21
d3cb
```