MATHEMATICS DEPARTMENT

"A computer is the mathematicians best friend"

μ - Games 12th edition
Mathematics Department Utrecht

22 November 2024

Rules:

The idea of this event is to gap the bridge between mathematics and programming. When working on these exercises, we hope the participant will get a better understanding of the underlying mathematical concepts. You will not be required to do a lot of difficult programming. With array manipulation and basic functionality, you should be able to solve all the exercises.

When working on these exercises, you must conform to the following rules.

- You are allowed to work in groups of maximum 4 people.
- You will have 3 hours to solve the problems.
- For the problems, you can use the default mathematics library of your programming language (for example *import math* in Python). Other libraries like *numpy* cannot be used.
- You cannot look up any computer code that may help you with solving the problem.

After 3 hours, the solutions to the exercises will be discussed. To check your own solution, one can go to the website `http://clover.science.uu.nl/dj`.

A standard setup of your code could look like this, where *function()* is the method used to solve the problem.

```
# First, we read the size of the testset (if necessary).
n = input()
arr = []

for _ in range(n):
    row = input().split()
    arr.append(row)

def function():
    pass

answer = function()

# Finally, we print the answer of the testset.
print(answer)
```

Problem 1: Rotating Points

Difficulty: ★ ☆ ☆ ☆ ☆
Key words: Linear Algebra, Geometry

We want to rotate a line segment with endpoints (x_0, x_1) and (y_0, y_1) multiple times around the origin $(0, 0)$.

Input

- One line with two floats $-1000 \leq x_0, x_1 \leq 1000$, denoting the first end point of the line segment.
- One line with two floats $-1000 \leq y_0, y_1 \leq 1000$, denoting the second end point of the line segment.
- One line with an integer $0 \leq n \leq 1000$ denoting the number of rotations we are doing.
- n lines with various integers $-360 \leq a \leq 360$ denoting the size of the rotation in degrees.

Output

- Output the sum of the two endpoint of the rotated line segment, i.e. $x_0 + x_1 + y_0 + y_1$. Round your answer up to the first four decimals (for example use the function ***round()*** in Python.)

Examples

Input	Output
0 0 0 1 2 30 60	-1

Problem 2: Factorial Fiasco

Difficulty: ★ ☆ ☆ ☆ ☆
Key words: Linear algebra

As a mathematician, you should be familiar with the factorial function. The factorial ! is a function $! : \mathbb{N} \to \mathbb{N}$ defined by $n \mapsto n! := n \cdot (n-1) \cdot (n-2) \cdots 1$. Frank would like to expand the definition of a factorial to 2×2 matrices with integral entries (this space is also called $\mathcal{M}_{\mathbb{Z}}(2)$). His first attempt is the following. Let:

$$\begin{aligned} ! : \mathcal{M}_{\mathbb{Z}}(2) &\to \mathcal{M}_{\mathbb{Z}}(2) \\ A &\mapsto A! := A \cdot (A - I) \cdot (A - 2I) \cdots (B + I), \end{aligned}$$

where B is the first matrix in the sequence $A, A - I, A - 2I, \ldots$ such that $\det(B) = 0$.
However, Frank soon notices such a B does not always exists, so this function is not well-defined. However, such a B *does* exist if the matrix A is of the form

$$A = \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix},$$

where λ, μ are positive integers.
Determine $\det(A!)$ for matrices of this form.

Input

- The input is one line with two space separated integers $1 \leq \lambda, \mu \leq 10^4$, from which you can determine the matrix A.

Output

- Output the determinant of $A!$ mod $10^9 + 7$, where the factorial of such a matrix is given above.

Examples

Input	Output
1 1	1

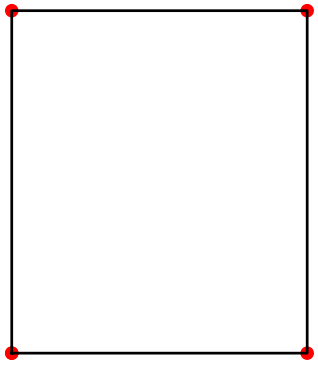
Input	Output
2 5	40

Problem 3: Polygon Area

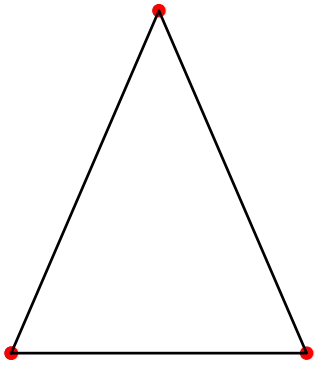
Difficulty: ★★☆☆☆
Key words: Linear Algebra, Matrices

Given the vertices of a polygon, specified in clockwise order, we are interested in the area of this polygon.

Example 1



Example 2



Input

- One line containing an integer $2 \leq n \leq 10000$ denoting the number of vertices of the polygon.
- n lines, each containing two integers $-10000 \leq x, y \leq 10000$ that represent the x and y coordinates of each vertex of the polygon.

Output

- A single line that prints the area of the polygon (correct up to the first 6 digits).

Tip: Try to form triangles!

Examples

Input	Output
4 0 0 0 1 1 1 1 0	1

Input	Output
3 0 0 1 2 2 0	2

Problem 4: Short Trees

Difficulty: ★ ★ ★ ☆ ☆
Key words: Graph theory, Combinatorics

We will look at graphs with a single root node, n nodes connected to the root node and and attached to each of these n nodes another k nodes. We will call this graph $T_{n,k}$

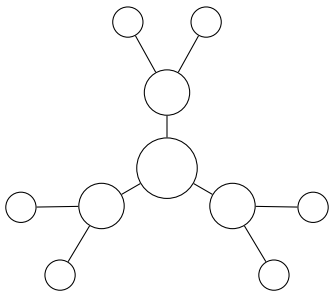


Figure 1: A picture of $T_{3,2}$

We are now interested in paths in this tree starting at the root node and having length ℓ. Your task is to count the number of paths for given n, k, ℓ.

Input

- One line consisting of 3 space-separated numbers $1 \leq n, k, \ell \leq 10^4$.

Output

- One number, the number of paths of length ℓ in the given graph.

Examples

Input	Output
1 1 1	1
3 1 2	6
3 2 2	9

Problem 5: Interior Grid Points

Difficulty: ★★★★☆
Key words: Linear Algebra, Matrices

We are investigating polygons defined on a two-dimensional grid and are interested in determining the number of grid points that lie within the boundaries of these polygons.
Given the vertices of the polygon specified in clockwise order, we seek to compute the total number of grid points contained inside the polygon.

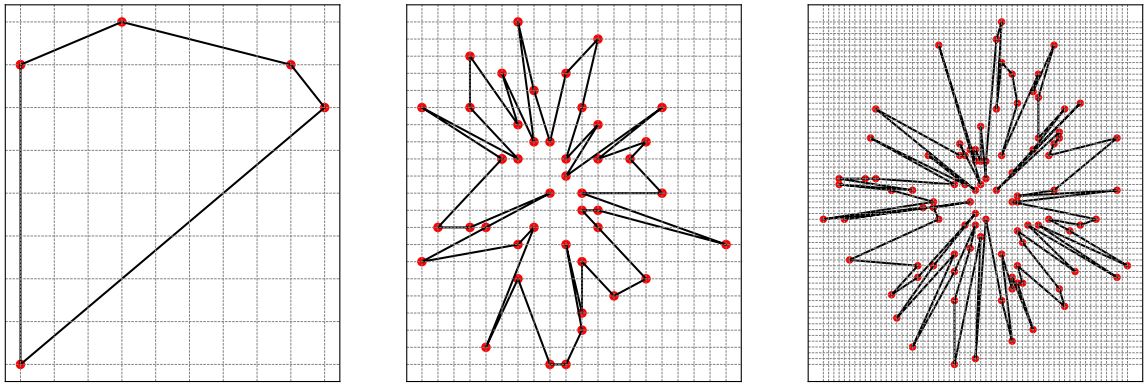


Figure 2: Various polygons in $\mathbb{R}^2$ with vertices indicated by red color.

Input

- One line containing an integer $2 \leq n \leq 10000$ denoting the number of vertices of the polygon.
- n lines, each containing two integers $-10000 \leq x, y \leq 10000$ that represent the x and y coordinates of each vertex of the polygon.

Output

- A single line that prints the number of integer grid points lying inside the polygon.

Tip: There is a relation between the area and the points in and on the polygon!

Examples

Input	Output
3 0 0 1 2 5 0	2

Input	Output
6 0 0 0 1 1 2 2 2 2 0 1 0	1

Problem 6: Associative Brackets

Difficulty: ★ ★ ★ ★ ★
Key words: Algebra

Sarah has to do calculate some products for her homework. She prefers doing her products from left to right, like $((a \cdot b) \cdot c) \cdot d$, but her teacher has given her exercises with random bracketings.

Sarah now want to know how many times she has to apply the associativity law to transform the given bracketing to a bracketing with all the brackets on the left.

The template for your solution in python can be found here https://clover.science.uu.nl/associative-brackets/ or by scanning the qr code at the bottom of the page.

Input

- We will use Polish notation[1] to denote our expressions, so $a \cdot ((b \cdot c) \cdot d)$ would be written as **x a x x b c d**. There are at least 1 and at most 10^6 characters used in the input.

Output

- A single integer, the minimum number of times you need to apply associativity to put all the brackets on the left.

Examples

Input	Output
x a x b c	1

Input	Output
x a x a a	1

Input	Output
x a x x b c d	2

Input	Output
x x a b c	0

Input	Output
x a x b x c d	2

[1] https://en.wikipedia.org/wiki/Polish_notation