MATHEMATICS DEPARTMENT

"A computer is the mathematicians best friend"

μ - Games
Mathematics Utrecht

June 2023

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 persons.
- You will have 3 hours to solve the problems.
- For the problems, you can use the default mathematics library of your programming language (*import math* in Python).
- 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`.

Problem 1: Equilibrium points

Difficulty: ★ ☆ ☆ ☆ ☆
Keywords: Dynamical systems, Local stability.

We are looking into the equilibrium points of a particular discrete dynamical system defined as

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_0 &\in \mathbb{R}. \end{aligned} \tag{1}$$

We want to find all possible equilibrium points of this system for a function of the form

$$f(x) = ax^2 + bx + c. \tag{2}$$

Input

- A line with the natural number $0 < a < 10^3$.
- A line with the natural number $0 < b < 10^3$.
- A line with the natural number $0 < c < 10^3$.

Output

- One or more lines printing the values of all the equilibrium points of System 1. Print these values in increasing order. Solutions are checked up to 10^{-7} in precision.

Examples

Input	Output
1	-1.0
2	
1	

Input	Output
1	-2.6180340
3	-0.3819660
1	

Problem 2: Enigmatic Evan

Difficulty: ★ ☆ ☆ ☆ ☆
Keywords: Combinatorics

Evan is a magical pirate travelling in a D-dimensional space. He starts at the origin of the galaxy and tries to get to some point P (with integer coordinates). Here his treasure is buried in some additional compactified dimensions. You also want to find Evan's treasure. To do so, you need to follow Evan to see how the treasure is hidden in the compactified dimensions. However, you are very busy, so you only want to follow him if you know he might have gone to the point P.

For this, you put a tracker on him which broadcasts his movements in terms of the letters A - Z. Each letter encodes a fixed D-dimensional translation vector with integer coefficients. However, you lost the packaging of your tracker, so you don't know exactly what vectors the letters A - Z encode. Given a movement string of Evan, is it possible that he ended up at the point P?

Input

- A line with an integer $1 \leq D \leq 100$ the dimensions of space.
- A line with D integers specifying the point P whose absolute value is smaller that 10^9.
- A line with a string of letters from the set A - Z. This string is non-empty and contains no more than 10^6 characters.

Output

- One line with a Y if Evan can reach P and a N if Evan cannot reach P.

Examples

Input	Output
3 7 8 2 A	Y

Input	Output
1 1 AA	N

Problem 3: Cory's Coins

Difficulty: ★ ★ ☆ ☆ ☆
Keywords: Set theory

Cory, an avid enthusiast of coin puzzles, seeks a reliable means to verify her solutions. She has approached you to develop a program capable of determining the minimum number of moves required to transform one coin configuration into another.

In Cory's puzzles, coins are arranged on an infinite grid. To facilitate your program's functionality, Cory will provide two images as input, which contain all the coins. In these images, a "0" signifies an empty spot, while a "1" represents the presence of a coin. Your program should then output the minimum number of moves necessary to transition from the initial image to the final one.

It is important to note that while the final image may be subjected to translation relative to the original, the orientation of the final image remains significant. Therefore, an arrow pointing down versus an arrow pointing up are considered different.

Input

- A line with the width and height of the first picture $1 \leq w1, h1 \leq 100$.
- $h1$ lines of length $w1$ containing 0 and 1 specifying the locations of the coins in the first configuration. Note that the number of coins (i.e. the number of 1's) is given by some number $1 \leq n \leq 1000$.
- A line with the width and height of the first picture $1 \leq w2, h2 \leq 100$.
- $h2$ lines of length $w2$ containing 0 and 1 specifying the locations of the coins in the second configuration. Note that no coins are lost, so in the second picture there will also be n coins.

Output

- One line with the least number of coin changes necessary to go from one picture to the other.

Examples

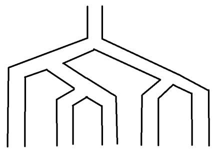
Input	Output
3 2	1
010	
111	
3 2	
111	
010	

Input	Output
1 3	2
1	
1	
1	
3 1	
111	

Problem 4: Multiple Mazes

Difficulty: ★ ★ ★ ☆ ☆
Key words: Trees, Generating function

Theresa would love to have a very weird maze in her garden. This maze runs from north to south with one entrance in the north. Then, as you start walking to the south you have two options: either you arrive at an exit or you arrive at a junction where your current path splits into two. There are no loops in the maze. For instance, from above her maze could look like this:



Suppose she wants her maze to have n exits. How many possible mazes can she construct?
Hint: First, find a recurrence relation, then consider a generating function with as coefficients the number of mazes with n exits. You may also use the identity $\binom{\frac{1}{2}}{n} = \frac{(-1)^{n+1}}{4^n(2n-1)}\binom{2n}{n}$. You are allowed to use Pythons math.comb for this exercise.

Input

- The number $1 \leq n < 100$ of exits.

Output

- The total number of mazes of this form with exactly n exits.

Examples

Input	Output
2	1

Input	Output
3	2

Problem 5: Walking Around

Difficulty: ★★★★☆
Key words: Probability, Random walks, Networks

We are considering a random walk on an undirected, connected graph G with vertex set V and edge set E. We assume that this graph contains at least one cycle of odd length. We define a simple random walk on a graph as a stochastic process, where we start at an initial state (vertex) s_1, and at every step we move to a vertex $s_{i+1} \in V$, with probabilities given by the transition matrix

$$P_{s_i,s_{i+1}} = \begin{cases} 1/d_{s_i} & \text{if } (s_i, s_{i+1}) \in E \\ 0 & \text{else,} \end{cases} \tag{3}$$

where d_{s_i} is the degree of vertex s_i. In Figure 1 we show a possible random walk on a network.

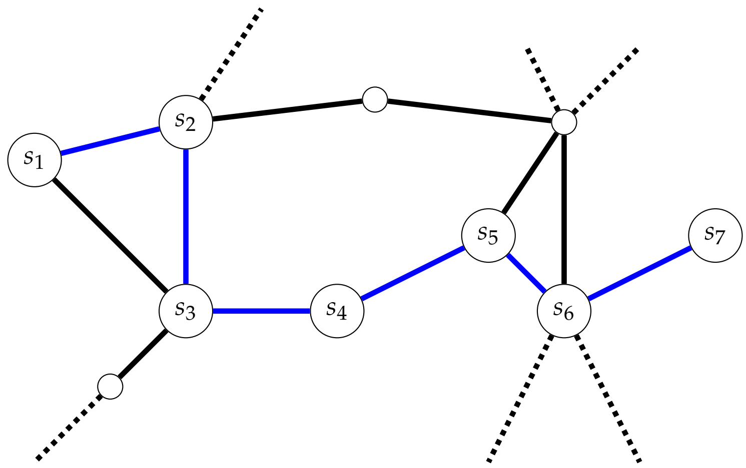


Figure 1: Simple random walk on an undirected network. Blue edges indicate the route of the random walk, while s_1 denotes the initial state of the random walk.

We are interested in the expected return time to an intial state. That is, the expected number of steps one has to take before returning to the initial state s_1.

Input

- An integer $0 < m = |E| < 10^6$ indicating the number of edges in the graph.
- One line containing a single integer $i \in \{1, 2, \ldots, n\}$, indicating the initial state of the graph.
- m lines , each containing two space separated integers, indicating the undirected edge in the graph.

Output

- The expected return time to the initial state s_1. Answer will be checked up to precision 10^{-7}.

Examples

Input	Output
3 1 1 2 2 3 3 1	3.0

Problem 6: Crazy Cricket

Difficulty: ★ ★ ★ ★ ☆
Key words: Combinatorics

Living in the first house of his street, Chris the cricket is responsible for delivering mail to all his neighbours. All houses in his street are equally spaced with distance 1 between adjacent houses. Every day, Chris starts at his house, visits the other houses in the street, and comes back to his own house. He never visits a house more than once during his trip.

One day, after eating some peculiar grass, Chris the cricket had unfortunately gone somewhat mad. Instantaneously, he developed an enormous fear of becoming dull, and in order to keep his job and not get scared of his daily mail duties, Chris needs to visit the houses in such a way that he never travels the same distance between two houses on his journey. Unfortunately, this makes it impossible to visit every house in the street, but Chris decides that skipping a single house should not be that big of a deal; they can just come to his house to pick up their mail. However, Chris is not very smart, and he has tasked you with finding a route that allows him to keep his job.

Input

- A single integer $3 \leq n \leq 10^6$, representing the number of houses in the street.

Output

- A list of n space-separated integers, denoting the route that Chris should follow, starting and ending with his own house. If no such route is possible, output `impossible` instead.

Examples

Input	Output
3	impossible

Input	Output
4	1 2 4 1

Problem 7: Deriving Denise

Difficulty: ★ ★ ★ ★ ☆
Key words: Calculus

Denise is a high school student, who is currently following calculus. She learns about derivatives and integrals, the latter of which she considers to be much more interesting. She thinks the grunt work of differentiating is beneath her; she would much rather spend her time integrating. So, she has decided to write a program that will do her differentiation homework sets for her.

As she is in high school, the functions to be differentiated are still rather simple. In fact, they are always a composition of the following simpler functions: exp : $\mathbb{R} \to \mathbb{R}$, mul : $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$, pow : $\mathbb{R} \times \mathbb{N} \to \mathbb{R}$, add : $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and div : $\mathbb{R} \times (\mathbb{R} \setminus \{0\}) \to \mathbb{R}$, which are defined by

$$\begin{aligned} \exp(x) &= e^x \\ \mathrm{mul}(a,b) &= a \cdot b \\ \mathrm{pow}(x,n) &= x^n \\ \mathrm{add}(a,b) &= a+b \\ \mathrm{div}(a,b) &= \frac{a}{b}. \end{aligned}$$

Denise's homework consists of questions of the following form: derive the function $f(x)$ with respect to x and evaluate this at some point p. Denise's teachers are rather lazy themselves, so they only want the final numerical answer of the above calculation with a precision of 10^{-5}. Furthermore, they also are not nefarious, so the exercises are given in such a form that the denominator of a fraction is non-zero at p.

Denise specifies the input as a computation graph. The first line specifies how many operations are used. The next n lines are given in the format of operations followed by space separated arguments. For the arguments there are three options. Either a single integer referencing the output of a previous operation. So, add 1 3 specifies the first and the third result in the list of operations should be added together. Secondly, there is the option of c followed by a number. This indicates the number should be used as is in the computation. So, mul c6 3 says that the third term should be multiplied by 6. Finally, we have the option x. This indicates this function depends directly on the variable x. So, pow x c2 denotes the term x^2.

Denise wants to know the derivative as computed at the final node in the computation graph, at the specified point p.

Input

- 1 line specifying the number of operations $1 \leq n \leq 100$.
- n lines each containing an operation followed by its arguments.
- 1 line specifying $-100 \leq p \leq 100$.

Output

- 1 line with $f'(p)$ with an accuracy of 10^{-5}.

Examples

Input	Output
2 mul x c3.345 add c4.234 1 2938418	3.345

Input	Output
9 pow x c2 pow x c3 pow x c4 mul c1.232928343 1 mul 2 c3.298313904 mul c6.534728934 3 add 4 5 add x 6 div 7 8 0.56832	-0.046385026372272056