KTH Challenge 2014

Stockholm, 13th April 2014

Problems

This page is intentionally left (almost) blank.

Problem A
Numbers On a Tree
Problem ID: numbertree

Photo by Chris Darling

Lovisa is at KTH listening to Stefan Nilsson lecturing about perfect binary trees. “A perfect binary tree has a distinguished node called the *root* which is usually drawn at the top. Each node has two children except the nodes in the lowest layer, which we call *leaves*.” Lovisa knows all this already, so she is a bit bored. Noticing this, Stefan comes up with a new challenge for Lovisa.

First, we label the nodes of a perfect binary tree with numbers as follows. We start at the bottom right leaf which gets number 1 and then label nodes on the same level in increasing order from right to left. After finishing a level, we move to the rightmost node in the level above and label all the nodes on that level from right to left. We proceed in this fashion until we reach the root.

When we want to describe a node in the tree, we can do it by describing a path starting at the root and going down toward the leaves. At each non-leaf node we can either go left (`'L'`) or right (`'R'`).

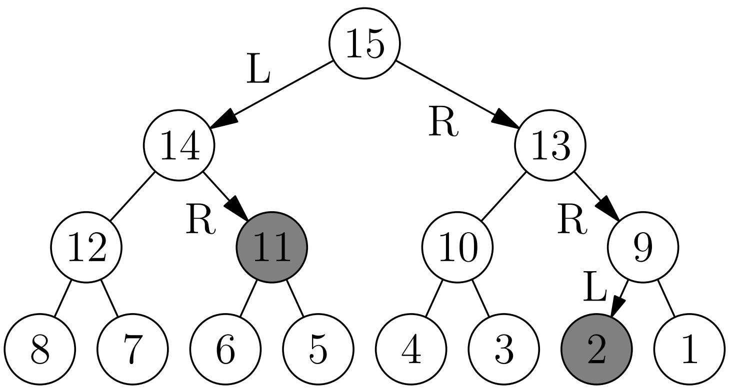


Figure A.1: Labeled binary tree of height 3 with two marked paths from the root. Path `LR` leads to label 11 while path `RRL` leads to 2. The root has number 15.

Task

Lovisa’s task is to calculate the label of a node, given the height of the tree H and the description of the path from the root.

Input

The only line of input contains the height of the tree H, $1 \leq H \leq 30$ and a string consisting of the letters `'L'` and `'R'`, denoting a path in the tree starting in the root. The letter `'L'` denotes choosing the left child, and the letter `'R'` choosing the right child. The description of the path may be empty and is at most H letters.

Output

Output one line containing the label of the node given by the path.

Sample Input 1	Sample Output 1
3 LR	11

Sample Input 2	Sample Output 2
3 RRL	2

Sample Input 3	Sample Output 3
2	7

Problem B
Absurdistan Roads II
Problem ID: absurdistan2

Photo by Ivan McClellan

The people of Absurdistan discovered how to build roads only last year. After the discovery, each city decided to build its own road, connecting the city with some other city. Each newly built road can be used in both directions.

Absurdistan is full of absurd coincidences. It took all N cities precisely one year to build their roads. And even more surprisingly, when the roads were finished it was possible to travel from every city to any other city using the newly built roads. We say that such a road network is *connected*. Being interested in mathematics and probability, you started wondering how unlikely this coincidence really is.

Task

Each city picked uniformly at random another city to which they built a road. Calculate the probability that the road network ends up being connected.

Input

The first line contains an integer N $(2 \leq N \leq 140)$ – the number of cities.

Output

Output one line containing a floating point number denoting the probability that the randomly built road network with N cities and N roads is connected. Your answer should have an absolute error of at most 10^{-8}.

Sample Input 1	Sample Output 1
4	0.962962962963

Sample Input 2	Sample Output 2
2	1.000000000000

This page is intentionally left (almost) blank.

Problem C
Cow Crane
Problem ID: cowcrane


Photo by Stuart Heath

Farmer Laura has a barn. In her barn, she has two cows, Monica and Lydia. Monica and Lydia love food, and they are quite lazy. For most of the day they chill out in the barn, waiting for Laura to come serve them a nice meal. Farmer Laura is always very precise about when to serve them food, so Monica and Lydia know exactly when to expect food, the same time every day.

This might sound surprising to you but there's a problem. Farmer Laura needs your help. She will be replacing some planks in the floor of the barn, which means that the cows have to be moved temporarily from their favorite spots. Since the cows are infinitely lazy, they refuse to walk themselves. Farmer Laura has rented an excellent tool to resolve this issue – a *cow crane*, designed and crafted specifically for the cow's comfort.

We visualize the barn as a one-dimensional line. The cow crane starts at time $t = 0$ at position $x = 0$, and it can move one distance unit per second. The crane can only carry one cow at a time, but it may pick up and drop off a cow as many times as necessary. Monica's current location is at $x = m$, and Lydia is located at $x = l$. Monica will be moved to the temporary location at $x = M$ and Lydia to $x = L$. Monica and Lydia always have their daily meal t_m and t_l seconds into the day, so the cows had better be in their respective temporary locations exactly by these times. You may assume that it takes no time for the crane to pick up or drop off a cow and that the two cows can be at the same position at the same time.

Task

Farmer Laura would like to know if she can move the cows so that both of them are in place at their temporary location no later than their daily meal occurs.

Input

Input consists of three lines. The first line consists of two integers m and l, the current positions of the cows. The second line consists of two integers M and L, the new positions of the cows. The third line consists of two integers t_m and t_l, the time at which the two cows will be served their daily meal. It is guaranteed that $-10^8 \leq m, l, M, L \leq 10^8$ and $1 \leq t_m, t_l \leq 10^8$. It is also guaranteed that both cows will actually move, i.e., $m \neq M$ and $l \neq L$.

Output

Output should consist of a single word. Print "`possible`" if it is possible to move both cows before they are served their daily meal. Otherwise, print "`impossible`".

Sample Input 1

```
-1 1
-2 2
6 6
```

Sample Output 1

```
possible
```

Sample Input 2

```
-1 1
-2 2
5 5
```

Sample Output 2

```
impossible
```

Sample Input 3

```
-1 1
1 -1
3 5
```

Sample Output 3

```
possible
```

Sample Input 4

```
0 1
2 3
6 3
```

Sample Output 4

```
possible
```

Problem D
Tomosynthesis
Problem ID: tomosynthesis

Tomosynthesis is a medical imaging modality in which a 3D dataset is obtained algorithmically from a set of X-ray images taken in different directions within a limited range of angles. A larger range of angles normally gives a better reconstruction, but is more difficult to acquire. Arvid is working on a reconstruction algorithm for obtaining the 3D image, but so far it doesn't seem to work when there are overlapping structures in any of the input images. The sample he will first reconstruct is a test object consisting of parallel equal-length cylinders of varying diameters that will be rotated around the axis of the cylinders.

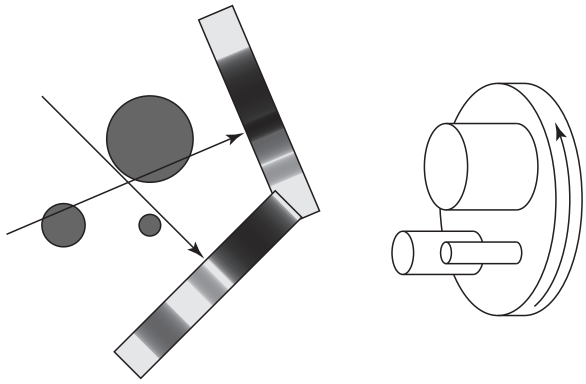


Figure D.1: To the left, a cross section of the test object of sample input 1. The upper projection is not acceptable since the two larger cylinders overlap. In the lower projection no cylinders overlap, so this direction and a range of angles around it are okay. To the right, an illustration of the same test object from the side.

Disregarding the fact that his algorithm will not work in practice, Arvid asks you for help. What is the largest range of angles in which the test object can be imaged without any cylinders overlapping in any of the images? An image is a plane projection of the structure perpendicular to the axes of the cylinders.

Input

The first line of input contains a single integer $2 \leq N \leq 100$ denoting the number of cylinders that constitute the test object. This is followed by N rows, each containing three floating point numbers x, y and r, denoting the x- and y-coordinate of the center of a cylinder, and the radius of that cylinder, respectively. The coordinates are in the range $-1\,000 \leq x, y \leq 1\,000$ and the radius $0 < r \leq 1\,000$. None of the cylinders touch or overlap.

Output

Output a single number, the size in radians of the largest continuous range of projection directions over which no cylinders overlap. If no such angle exists output `0`. Your answer should have an absolute error of at most 10^{-8}.

Sample Input 1	Sample Output 1
3 -1 -1 0.5 1 -1 0.25 1 1 1	0.511268019

Sample Input 2	Sample Output 2
2 0 0 1 2 2 1	1.570796327

Problem E
Pizza Problems
Problem ID: pizzaproblems

Photo by Sam DeLong

Me and my friends are ordering a big pizza to share. As you can imagine this is quite complicated, since everyone has different wishes about what should be on the pizza. For instance Gunnar wants bananas on the pizza, Emma doesn't want bananas but wants olives, Marc wants there to be tomatoes, and so on. Fortunately, against all odds, we managed to come up with a selection of toppings such that everyone had at least $2/3$'s of their wishes fulfilled, which we unanimously decided was good enough.

But then, disaster struck! We sent out Lukáš to buy the pizza, but he accidentally lost the piece of paper on which we had written down our carefully selected list of toppings. Now we're back at square one, and have to construct a new selection of toppings. Given how long it took us to find the original selection of toppings, we have decided to lower our standards a bit and just try to find a selection such that everyone has *strictly* more than $1/3$ of their wishes fulfilled.

Can you help us with this? If you do, you'll get some pizza!

Input

The first line of input contains an integer $1 \leq N \leq 10\,000$, the number of friends in the group (including yourself). Each of the next n lines contains the list of wishes of one of the friends. This list starts with an integer $1 \leq w \leq 30$, the number of wishes this friend has, followed by a space-separated list of wishes. Each wish is either "`+<topping>`" or "`-<topping>`" where `<topping>` is the name of a topping, indicating that this friend wants or does not want this topping. Each topping name appears at most once in each list.

Topping names are non-empty strings of up to 15 lower-case English letters '`a`'-'`z`'. There are at most 250 different toppings.

Output

Output a list of toppings (without repetitions, separated by spaces or newlines) such that each friend has strictly more than $1/3$ of their wishes fulfilled. You may assume that there exists a list such that every friend has at least $2/3$ of their wishes fulfilled.

Your list of toppings is not allowed to contain any toppings that are not mentioned in the input, nor is it allowed to contain repetitions.

Sample Input 1

```
1
4 +zucchini +mozzarella +mushrooms -artichoke
```

Sample Output 1

```
zucchini
mozzarella
mushrooms
artichoke
```

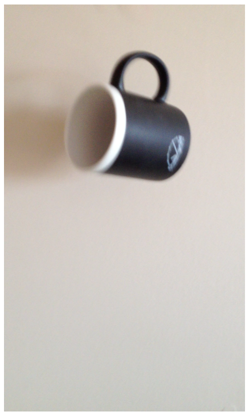
Sample Input 2

```
3
3 +redbeans +soylentgreen -bluecheese
3 +redbeans -soylentgreen +bluecheese
3 -redbeans +soylentgreen +bluecheese
```

Sample Output 2

```
redbeans
soylentgreen
bluecheese
```

Problem F
Falling Mugs
Problem ID: falling



Susan is making high-speed videos of falling coffee mugs. When analyzing the videos she wants to know how big the mugs are, but unfortunately they all got destroyed during filming. Susan knows some physics though, so she has figured out how far they moved between different video frames. The camera was filming at a speed of 70 frames per second, which means that at frame n, counted from when the mug was released, the number of millimeters it has moved is $d = n^2$. The counting of the frames starts at 0.

Susan thinks a certain mug is D millimeters high, but to verify this she needs to find two frames between which the mug has moved exactly this distance. Can you help her do this?

Input

The input file contains the single positive integer $D \leq 200\,000$, the distance to be measured.

Output

Output two non-negative integers n_1 and n_2, the numbers of the frames that Susan should compare. They should fulfill $n_2^2 - n_1^2 = D$. If no two such integers exist, instead output "`impossible`".

Sample Input 1	Sample Output 1
88	9 13

Sample Input 2	Sample Output 2
86	impossible

This page is intentionally left (almost) blank.

Problem G
Intercept
Problem ID: intercept


Photo by Patrik Neckman

Fatima commutes from KTH to home by subway every day. Today Robert decided to surprise Fatima by baking cookies and bringing them to an intermediate station. Fatima does not always take the same route home, because she loves to admire the artwork inside different stations in Stockholm. However, she always optimizes her travel by taking the shortest route. Can you tell Robert which station he should go to in order to surely intercept Fatima?

Input

The first line contains two integers N and M, $1 \leq N, M \leq 100\,000$, where N is the number of subway stations and M is the number of subway links. M lines follow, each with three integers u, v, w, $0 \leq u, v < N$, $0 < w \leq 1\,000\,000\,000$, meaning that there is a one-way link from u to v that takes w seconds to complete. Note that different subway lines may serve the same route.

The last line contains two integers s and t, $0 \leq s, t < N$ the number of the station closest to KTH and closest to home, respectively. It is possible to reach t from s.

Output

A space separated list of all the station numbers u such that all shortest paths from s to t pass through u, in increasing order.

Sample Input 1	Sample Output 1
4 4 0 1 100 0 2 100 1 3 100 2 3 100 0 3	0 3

Sample Input 2	Sample Output 2
7 8 0 1 100 0 2 100 1 3 100 2 3 100 3 4 100 3 5 100 4 6 100 5 6 100 0 6	0 3 6

This page is intentionally left (almost) blank.

Problem H
Radar
Problem ID: radar

Photo by alex.ch

After your boat ran out of fuel in the middle of the ocean, you have been following the currents for 80 days. Today, you finally got your radar equipment working. And it's receiving signals!

Alas, the signals come from the "radar" station owned by the eccentric lighthouse keeper Hasse. Hasse's radar station (which does not work quite like other radar stations) emits continuous signals of three different wave-lengths. Therefore, the only interesting thing you can measure is the phase of a signal as it reaches you. For example, if the signal you tuned on to has a wave-length of 100 meters and you are 1456 meters from the station, your equipment can only tell you that you are either 56, or 156, or 256, or $\ldots$ meters away from the lighthouse.

So you reach for your last piece of paper to start calculating – but wait, there's a catch! On the display you read: "ACCURACY: 3 METERS". So, in fact, the information you get from this signal is that your distance from Hasse's radar station is in the union of intervals $[53, 59] \cup [153, 159] \cup [253, 259] \cup \ldots$.

What to do? Since the key to surviving at sea is to be optimistic, you are interested in what the smallest possible distance to the lighthouse could be, given the wavelengths, measurements and accuracies corresponding to the three signals.

Task

Given three positive prime numbers m_1, m_2, m_3 (the wavelengths), three nonnegative integers x_1, x_2, x_3 (the measurements), and three nonnegative integers y_1, y_2, y_3 (the accuracies), find the smallest nonnegative integer z (the smallest possible distance) such that z is within distance y_i from x_i modulo m_i for each $i = 1, 2, 3$. An integer x' is *within distance* y from x modulo m if there is some integer t such that $x \equiv x' + t \pmod{m}$ and $|t| \leq y$.

Input

There are three lines of input. The first line is m_1 m_2 m_3, the second is x_1 x_2 x_3 and the third is y_1 y_2 y_3. You may assume that $0 < m_i \leq 10^6$, $0 \leq x_i < m_i$, and $0 \leq y_i \leq 300$ for each i. The numbers m_1, m_2, m_3 are all primes and distinct.

Output

Print one line with the answer z. Note that the answer might not fit in a 32-bit integer.

Sample Input 1	Sample Output 1
11 13 17 5 2 4 0 0 0	2095

Sample Input 2	Sample Output 2
941 947 977 142 510 700 100 100 100	60266

Problem I
Count von Walken's Fence
Problem ID: vonwalken

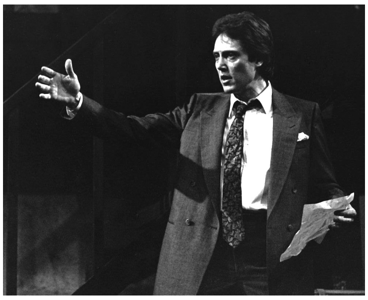


Photo by Lascher

The old Count von Walken ponders along the fence of his backyard. The fence has a repeating pattern with poles in the ground at equal distances. Since von Walken has nothing better to do, he counts the number of steps he takes between each pole.

The distance between two consecutive poles turns out not to be an integer multiple of the length of his steps because sometimes he takes two steps between the poles, and sometimes he takes three steps.

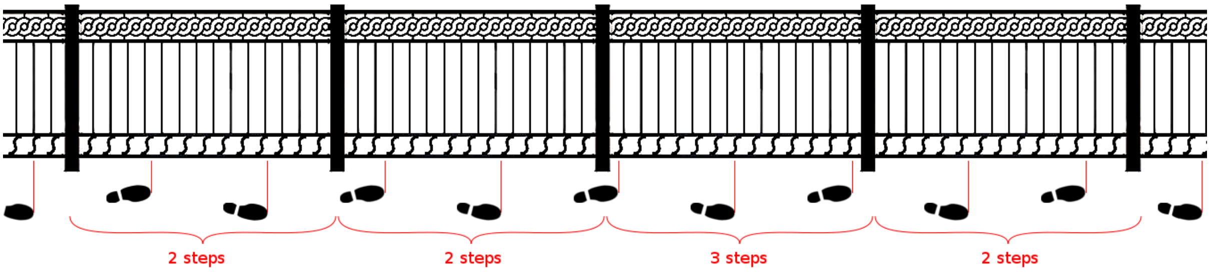


Figure I.1: A picture of sample case 2

Von Walken knows that his steps are always 1 meter long, so he starts thinking of what the distance between the poles may be. *"It must be more than 2 meters, since I occasionally can fit 3 steps between the poles, but it must be less than 3 meters, since I sometimes only fit 2 steps in between."*

Task

Given a list of step counts and a distance D, determine whether it is possible that the distance between two poles is D meters. The poles can be considered to have width 0, and each step is strictly between two poles.

To avoid problems with floating point numbers, the result is guaranteed to be the same even if any pole is moved up to 10^{-7} meters.

Input

The input consists of a line containing the real number D and an integer N, followed by a line with the space-separated list of integer step counts, $c_1, c_2, \ldots, c_N$. It holds that $2 \leq c_i, D \leq 3$ and that $0 \leq N \leq 10\,000$.

Output

The program should print "`possible`" if D meters is a possible distance between the poles, and "`impossible`" otherwise.

Sample Input 1

```
2.505 4
2 2 3 2
```

Sample Output 1

```
impossible
```

Sample Input 2

```
2.1 4
2 2 3 2
```

Sample Output 2

```
possible
```