DAPC 2025

Delft Algorithm Programming Contest 2025

Problems

A	Alto Adaptation
B	Bottle of New Port
C	Calculation Obfuscation
D	Dralinpome
E	Entropy Evasion
F	Friendly Formation
G	Genealogy Gumbo
H	Hidden Sequence
I	Ingredient Intervals
J	Journal Publication
K	Koehandel
L	Landgrave

A Alto Adaptation

Time limit: 2s

Concerto by Lorenzo Costa. Public domain on Wikimedia Commons

It is always so frustrating when the song you want to sing is *just* outside the range of your beautiful alto singing voice... Thanks to music theory, a note outside your vocal range sounds similar to the intended note when you *transpose* it by one or multiple octaves up or down. However, just transposing a single note that is out of your range sounds weird: you would prefer to sing longer sequences of notes transposed by the same number of octaves.

To simplify some music theory, you represent the song as a sequence of numbers. Each number represents the *pitch* (or "height") of a music note. The span of your vocal range is represented by two numbers, and you can sing any note between these two endpoints (inclusive). There are twelve notes in an octave, so transposing a note by some number of octaves up or down makes this note some multiple of 12 higher or lower than in the original song. You can switch between different transpositions at any point during the song. This divides the song into intervals of notes transposed by the same number of octaves. Your goal is to make the shortest such interval as long as possible. The original notes of the song can be treated as transposed by 0 octaves.

As an example, consider the third sample case, visualized in Figure A.1. The first three notes comfortably fit in your vocal range. The subsequent three notes need to be transposed two octaves down to fit your vocal range. The next three notes should be transposed one octave down. The final two notes need to be transposed one octave up. So, the shortest interval of notes in the same transposition consists of the two notes at the end of the song.

Input

The input consists of:

- One line with three integers n, ℓ, and h ($1 \leq n \leq 1000$, $0 \leq \ell < h < 120$, $\ell + 11 \leq h$), the number of notes, and the low and high end of your vocal range (inclusive).
- One line with n integers a ($0 \leq a < 120$), the notes in the song that you want to sing.

Output

Output the maximum length of the shortest interval of notes that should be transposed by the same number of octaves to make them fit inside your vocal range.

Sample Input 1	Sample Output 1

`6 20 42`
`22 29 32 19 21 23`
	`3`

Sample Input 2	**Sample Output 2**
6 40 64 42 42 42 42 42 42	6

Sample Input 3	**Sample Output 3**
11 12 23 15 19 18 40 42 44 26 29 28 4 2	2

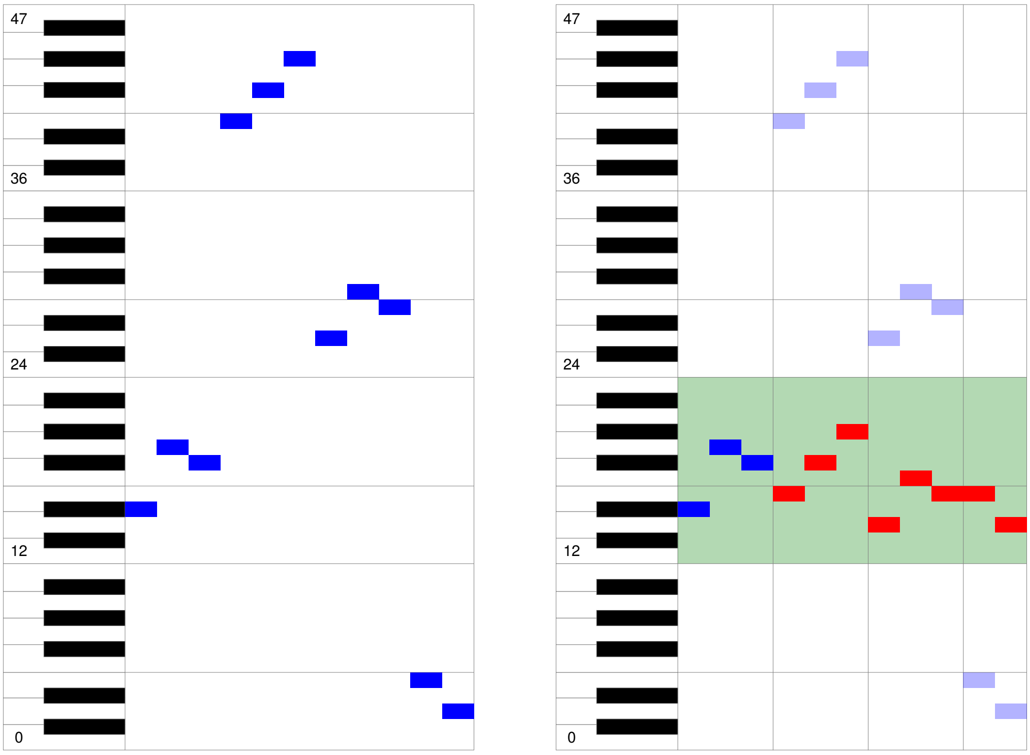


Figure A.1: Illustration of the third sample case. The green shaded background corresponds to your vocal range. The blue notes are the notes in the original song. The red notes are sung one or two octaves higher or lower than in the original song.

B Bottle of New Port

Time limit: 1s

At the Bottles and Port Company, you produce and bottle only the finest of vintages. To satisfy the desires of your highly exclusive clientele, it is necessary to study what happens to your wines once a new bottle is opened. When that happens, liquids in the bottle immediately start evaporating. The alcohol in the bottle evaporates at a different rate than the other liquids in the bottle, meaning that over time, the delicate balance of the port wine will be disturbed.

A bottle of your favourite port, won at a programming competition. Picture by Reinier Schmiermann

Your goal is to compute how the alcohol percentage changes in the days after opening a fresh bottle. By placing the opened bottle in an advanced chemical apparatus, you are able to measure how fast each liquid in the bottle evaporates.

Input

The input consists of:

- One line with an integer d ($0 \leq d \leq 10^6$), the number of days your bottle of new port has remained open.
- One line with two integers a and o ($1 \leq a, o \leq 10^{12}$), the initial volume of alcohol in the bottle and the initial total volume of other liquids in the bottle, both in µL.
- One line with two integers Δ_a and Δ_o ($0 \leq \Delta_a, \Delta_o \leq 10^{12}$), the evaporation rate of the alcohol and the evaporation rate of the other liquids in the bottle, both in µL/day.

It is guaranteed that the bottle is not empty after leaving it open for d days ($d \cdot \Delta_a < a$ or $d \cdot \Delta_o < o$).

Output

Output the alcohol percentage after leaving the bottle open for d days.

Your answer should have an absolute or relative error of at most 10^{-6}.

Sample Input 1

```
2
3 8
1 1
```

Sample Output 1

```
14.2857142857143
```

Sample Input 2

```
5
11 89
2 1
```

Sample Output 2

```
1.17647058823529
```

Sample Input 3

```
10
10 40
1 1
```

Sample Output 3

```
0
```

Sample Input 4

```
10
40 10
0 3
```

Sample Output 4

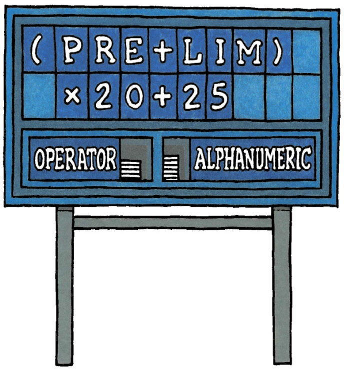
```
100
```

C Calculation Obfuscation

Time limit: 2s

Marijn wrote an expression to calculate the answer to the ultimate question. Unfortunately, his friend Jeroen decided to pull a prank and shuffled all the characters in his expression!

The syntax of the expression is similar to that of most programming languages (in fact, the expression is valid Python):



8 out of 10 jury members does *Calcdown.*

- Expressions consist of numbers, variables, and parenthesized expressions (enclosed between '`(`' and '`)`'), which are joined together by the operators '`+`' and '`*`'.
- Numbers consist of digits (`0-9`) and do not start with a leading '`0`', except for the number '`0`' itself.
- Variables start with a letter, followed by zero or more letters and/or digits.
- The '`*`' operator takes precedence over '`+`'.

An example of a valid expression is "`(abc+var0)*(12+0)`", where the addition of two variables "`abc`" and "`var0`" is multiplied by the addition of "`12`" and "`0`".

Luckily, Marijn remembers that he did not use any unnecessary parentheses in his code. This means that for example "`a+(b+c)`" does not appear because the '`+`' operator is associative, nor does "`(a*b)+c`" since the '`*`' operator takes precedence over '`+`'. **Note:** it is not required that removing any of the parentheses actually changes the outcome of the expression.

Given a shuffled expression, help Marijn find any expression that could have been his original expression, if such an expression exists.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 3 \cdot 10^5$), the length of the shuffled expression.
- One line with a string of length n, the shuffled expression, consisting of the characters in "`()+*`", digits (`0-9`), and English lowercase letters (`a-z`).

Output

If an expression exists that satisfies all the requirements, output "`possible`" followed by such a permutation of the input string. Otherwise, output "`impossible`".

If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
7
123test
```

Sample Output 1

```
possible
test321
```

Sample Input 2

```
10
012var+*()
```

Sample Output 2

```
possible
(var2+0)*1
```

Sample Input 3

```
7
(1+2)+3
```

Sample Output 3

```
impossible
```

Sample Input 4

```
7
(000+*)
```

Sample Output 4

```
possible
(0+0)*0
```

Sample Input 5

```
2
00
```

Sample Output 5

```
impossible
```

Sample Input 6

```
9
((1+2))*3
```

Sample Output 6

```
impossible
```

Sample Input 7

```
10
bilmseiops
```

Sample Output 7

```
possible
impossible
```

D Dralinpome

Time limit: 1s

Your highly intelligent girlfriend, Jeannetta Ewell from Mississippi, loves to concatenate letters. For each of your anniversaries, you prepare a jar of letter jelly as a gift for Jeannetta. One of her favorite linguistic constrictions is that of a *palindrome*. In a palindrome, each letter reappears in place after a reorientation, reversing the string. For example, the word "`racecar`" is a palindrome, as it reads the same forwards and backwards.

The letter jelly that Jeannetta received last anniversary.

After writing out "`jeannetta`" with her letter jelly, by arranging the letters in the order "`natejetan`", she suddenly realized that she made a palindrome! This incidence lead to Jeannetta introducing her inventive definition of a *dralinpome*. In a dralinpome, the letters can be rearranged to form a palindrome.

Jeannetta now seeks to know all dralinpomes in the dictionary. However, due to her disinterestedness in programming, she would need to manually consider each word in the dictionary. While she has no idea what to do with an array or a queue, you and your teammates reparticipate in programming competitions each year. Therefore, you decide to help her and prepare a program that testifies of any given word whether it is a dralinpome or not.

Input

The input consists of:

- One line with a string s $(1 \leq |s| \leq 10^5)$, the word to check. The word consists of only English lowercase letters (`a-z`).

Output

Output "`yes`" if the given word is a dralinpome, or "`no`" otherwise.

Sample Input 1	Sample Output 1
`zoo`	`yes`

Sample Input 2	Sample Output 2
`yes`	`no`

Sample Input 3	Sample Output 3
`cacao`	`yes`

Sample Input 4	Sample Output 4
`multinomiaalcoefficient`	`yes`

Sample Input 5	Sample Output 5
`racecars`	`no`

E Entropy Evasion

Time limit: 1s

A primitive satellite, not yet equipped with a BAPC.
CC BY 2.0 by NASA on Flickr

Last week, you were contacted by aliens. Strangely, no one you tell seems to believe you – this is likely some form of alien brainwashing. Nonetheless, you decided to reply to the message, and with the help of a few video tutorials, a rocket with a picture of you and your cat is currently long underway to the aliens. Perhaps more importantly, the rocket also contains an array of n bits, all set to 0. This was specifically requested by the aliens: "State your intentions. Send an array of bits set to 1 to indicate you wish to remain peaceful. In any other case, we will destroy your planet." Wait, what? Set to 1? Whoops.

Luckily, the rocket is equipped with a Bit Array Protection Controller (BAPC). The BAPC can be opened up to expose a contiguous section of the bit array to interstellar radiation, independently setting each of the exposed bits to either 0 or 1, each with a probability of 50%. Your idea is to exploit this randomness to set most of the bits in the array to 1. The aliens sounded serious, but not *that* serious, so setting at least 70% of the bits to 1 is probably fine. Unfortunately, the rocket has almost reached its destination, so you can open up the BAPC at most 125 times before the aliens read the array. Send the correct commands to the BAPC to ensure at least 70% of the bits are set to 1.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads from the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

The interactor first sends one line with an integer n ($1 \leq n \leq 1000$), the length of the bit array v.

Then, your program should send at most 125 commands to set at least 70% of the bits to 1. Each command is sent by printing one line of the form "ℓ r" ($1 \leq \ell \leq r \leq n$), which will expose the bits in positions $[\ell, r]$ to interstellar radiation. The interactor will respond with two lines. The first line consists of n integers $v_1, v_2, \ldots, v_n$ ($v_i \in \{0, 1\}$ for each i), the new state of the array. The second line contains a single integer p ($0 \leq p \leq 100$), the current percentage of ones, rounded down.

When the percentage of ones is at least 70, the interaction will stop. Sending any additional commands after this point or using more than 125 commands will result in a wrong answer.

For each of your submissions, the interactor uses a new random seed. The interactor is not adaptive.

Your submission will be run on at most 100 test cases.

Make sure you flush the buffer after each write.

A testing tool is provided to help you develop your solution.

Sample Interaction 1

Read	Write
4	
	1 4
0 0 0 0 0	
	1 3
1 1 1 0 75	

Sample Interaction 2

Read	Write
3	
	1 3
1 0 0 33	
	2 2
1 0 0 33	
	2 2
1 1 0 66	
	3 3
1 1 1 100	

F Friendly Formation

Time limit: 3s

Last year's BAPC. For a short while, all was still fine. CC BY-SA 3.0 by Stefan Krause on Wikimedia Commons

This year, n players showed up for the Big Annual Paintball Competition (BAPC), a large national contest in which a blue and red team battle for the national title. Some of the players that showed up know each other already, while others do not. At last year's BAPC, this led to some unfortunate communication incidents, causing the entire event to be cut short: half of the blue team had somehow made their way to an adjacent field, and the red team miraculously managed to capture *their own* flag. This year's organization has therefore decided that in both teams, each pair of players should already know each other. Additionally, to keep the game fair, all players should be assigned a team, and both teams should be equally large. Is it possible to split the group into two such teams?

Input

The input consists of:

- One line with two integers n and m ($2 \leq n \leq 10^6, 0 \leq m \leq 10^6$), the number of players and the number of relations.
- m lines, each with two integers a and b ($1 \leq a, b \leq n$, $a \neq b$), indicating that players a and b already know each other.

A relation between any pair of players will be listed at most once.

Output

If it is possible to split the group into two teams with the above requirements, then for each player i, output "`r`" if player i should join the red team, and "`b`" if player i should join the blue team. Otherwise, output "`impossible`".

If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
2 1
1 2
```

Sample Output 1

```
r
b
```

Sample Input 2

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

Sample Output 2

```
impossible
```

Sample Input 3

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

Sample Output 3

```
impossible
```

Sample Input 4

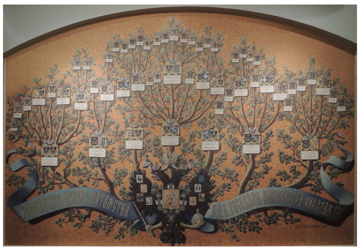
```
4 4
1 2
2 3
3 4
4 1
```

Sample Output 4

```
r
b
b
r
```

G Genealogy Gumbo

Time limit: 3s



A reconstruction of the family tree you have found. CC BY-SA 4.0 by Shakko on Wikimedia

You just got a new job as a Byzantine Ancestry Project Coordinator where you try to piece together ancient family trees of people long dead. Because you can no longer ask them any questions, you have to rely on documents left from that era where names are written patrilineally: ancestry is only traced by the name of someone's father. One example of such name would be "Basil, son of Alexios". You are trying to prove your research hypothesis: the entire Byzantine population is descended from just a single common ancestor.

In order to prove this hypothesis, you need to determine whether such a family tree exists: find one possible family tree with a single root. You cannot create imaginary people or construct familial relations for which there is no historical evidence. One father can have multiple children, but one child cannot have multiple fathers. It is possible that different people have the same name.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), the number of parental relations specified.
- n lines, each with a name of the form "A`, son of `B".

Each name has at most 20 characters and consists of one English uppercase letter (`A-Z`), followed by only English lowercase letters (`a-z`).

Output

If it is possible for the given people to have a single common ancestor, output "`possible`". Otherwise, output "`impossible`".

Sample Input 1	Sample Output 1
3 Jacob, son of Isaac Isaac, son of Abraham Ishmael, son of Abraham	possible

Sample Input 2	Sample Output 2
2 Basil, son of Alexios Procopius, son of Constantine	impossible

Sample Input 3

```
3
Alvin, son of Bert
Bert, son of Charles
Charles, son of Alvin
```

Sample Output 3

```
possible
```

Sample Input 4

```
2
James, son of Harry
Harry, son of James
```

Sample Output 4

```
possible
```

Sample Input 5

```
3
Albert, son of Brent
Cody, son of Brent
Brent, son of Daniel
```

Sample Output 5

```
possible
```

Sample Input 6

```
3
Aaron, son of Aaron
Aaron, son of Bobby
Bobby, son of Chris
```

Sample Output 6

```
possible
```

Sample Input 7

```
3
Chris, son of Bobby
Bobby, son of Aaron
Aaron, son of Aaron
```

Sample Output 7

```
possible
```

H Hidden Sequence

Time limit: 4s

Three-player chess; one of the games of the evening. Public domain by Dr Jacek Filek on Wikimedia Commons

Three players are having a game night, playing a long sequence of different games. In each game, one of the three players wins. In order to keep track of the order of winners, each of the three players decides to keep a list of all winners in order. However, due to the dopamine rush as the result of winning a game, everyone always forgets to write themselves on their own list of winners. Thus, the list of player 1 only contains the order in which players 2 and 3 won the different games, the list of player 2 only contains the order in which players 1 and 3 won, and the list of player 3 only contains the order in which players 1 and 2 won.

Given these three lists, determine the actual order in which all three players won their games.

Input

The input consists of:

- Three lines, the ith of which contains a string s_i ($1 \leq |s_i| \leq 10^5$), the order in which the two players different from i won their games.

It is guaranteed that it is always possible to determine the unique order in which all three players won their games.

Output

Output a single string containing the order in which all three players won their games.

Sample Input 1	Sample Output 1
2 1 21	21

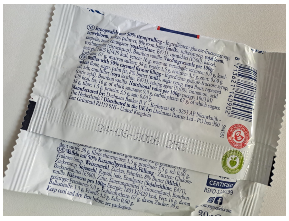
Sample Input 2	Sample Output 2
23 13 12	123

Sample Input 3	Sample Output 3
23232323 11331133 12121212	121323121323

This page is intentionally left blank.

I Ingredient Intervals

Time limit: 1s



The ingredient list of a stroopwafel, where only two percentages are shown.

The label of a supermarket product describes the composition of its ingredients as follows:

> **Ingredients**: Chocolate (41.1%), Sugar, Peanuts (20%), Starch, Thickener, Glaze.

In accordance with regulations from the Bureau for the Analysis of Product Composition, the ingredients are listed in descending order by percentage, and the percentages add up to 100%. While some ingredients have their percentage specified on the label, others do not.

For each unspecified ingredient, determine the minimum and maximum possible percentage, given these conditions.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 100$), the number of ingredients on the label.
- n lines, each starting with a string w ($1 \leq |w| \leq 20$), the name of the ingredient, optionally followed by a single number p ($0 < p \leq 100$), the specified percentage of the ingredient.

The ingredient names are unique and only consist of English lowercase letters (`a-z`). Percentages are given with at most one digit after the decimal point. There is at least one ingredient whose percentage is not specified. The input describes a valid label.

Output

Output the unspecified ingredients in the order given in the input. For each such ingredient, output its name, followed by its minimum and maximum percentages.

Your answer should have an absolute or relative error of at most 10^{-6}.

Sample Input 1

```
6
chocolate 41.1
sugar
peanuts 20
starch
thickener
glaze
```

Sample Output 1

```
sugar 20 38.9
starch 0 18.9
thickener 0 9.45
glaze 0 6.3
```

Sample Input 2

```
1
water
```

Sample Output 2

```
water 100 100
```

Sample Input 3

```
4
flour
milk
egg
salt
```

Sample Output 3

```
flour 25 100
milk 0 50
egg 0 33.33333333
salt 0 25
```

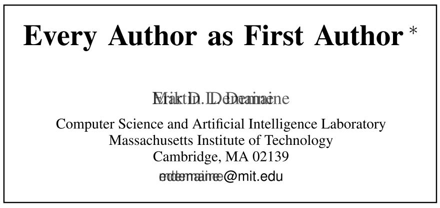
J Journal Publication

Time limit: 4s

Both winning BAPC and publishing research papers is a team effort. And every member of the team wants to be appreciated.

Every research paper has a list of authors at the top. Each author is listed using their full name, which might consist of several parts.

To avoid arguments about the order of names, your team decided to just order the list of names in the following way:



Every Author as First Author *

Erik D. Demaine
Martin L. Demaine

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
edemaine@mit.edu
mdemaine@mit.edu

Another way to make every author feel equally appreciated is to superimpose their names.
© Erik D. Demaine
Martin L. Demaine, arXiv:2304.01393 [**cs.DL**],
used non-commercially

1. Pick a single name part for each author.
2. Sort authors in non-descending order, comparing only the selected parts lexicographically.

However, your team already submitted the draft for your paper before you came up with this idea.

You are given the list of author as presented in the draft. Is it possible to pick name parts in a way that the list of authors in the draft is sorted?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), the number of authors.
- n lines (one for each author name), each with an integer p ($1 \leq p \leq 10$), the number of name parts, followed by p name parts.

Each name part has at most 10 characters and consists of one English uppercase letter (`A-Z`), followed by only English lowercase letters (`a-z`).

Output

If it is possible to select a name part for each author such that the list is sorted, output for each author the selected name part. Otherwise, output “`impossible`”.

If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
6
2 Maria Douglas
3 Ozzy Levi Carpenter
3 Quentin Aaron Potter
2 Christy Iglesias
2 Mo Mansur
3 Sam Marlon Scully
```

Sample Output 1

```
impossible
```

Sample Input 2

```
4
2 Maria Douglas
3 Ozzy Levi Carpenter
3 Quentin Aaron Potter
3 Sam Marlon Scully
```

Sample Output 2

```
Maria
Ozzy
Potter
Scully
```

Sample Input 3

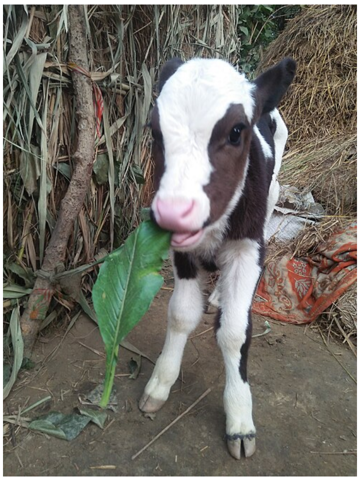
```
3
1 Sam
1 Sam
1 Sam
```

Sample Output 3

```
Sam
Sam
Sam
```

K Koehandel

Time limit: 1s



This cute cow you would not to want to part with for anything.
CC BY-SA 4.0 by Ashish Rajvanshi on Wikimedia Commons

You are playing a game of *Betting and Purchasing Cows* (BAPC), and Old MacDonald wants to buy one of your cows using a bid-off. This works as follows.

He bets some number of BAPC coins, which he puts in a cup that he places upside down on the table. You then openly bet some number of your own coins. The cup is removed to reveal MacDonald's bet. The bets are exchanged: you receive the bet from MacDonald and he receives your bet. The player that bet the most BAPC coins receives one cow from the opponent, regardless of who started the bidding. In case of a tie, no cows are exchanged.

You love the cows in the game, so your primary goal is to have the most possible cows after the trade. Your secondary goal is to have the highest possible number of coins after the trade. You pay close attention when Old MacDonald is secretly putting their coins in the cup, and by listening very carefully, you have determined the exact number of coins that he put in the cup. Given this information and the number of coins you have, how many coins should you bet?

Input

The input consists of:

- One line with two integers c and n ($0 \leq c, n \leq 10^9$), the number of BAPC coins in Old MacDonald's cup and the number of coins that you have.

Output

Output the number of coins that you should bet, keeping your goals in mind.

Sample Input 1	Sample Output 1
6 19	7

Sample Input 2	Sample Output 2
42 37	0

Sample Input 3	Sample Output 3
25 25	25

This page is intentionally left blank.

L Landgrave

Time limit: 5s

Artist's impression of your castle after completion. CC0 by Perituss on Wikimedia Commons

Congratulations on your new title of Landgrave, along with the generous stretch of land that accompanies it. Your predecessor was certainly industrious – he scattered towers across the region like seeds in a field – yet somehow never managed to assemble even one proper castle.

You will immediately remedy this oversight, for your fiefdom needs a castle to which the population can retreat in times of strife. To make a castle, you construct a cycle from some of the existing towers, connecting them with straight walls to enclose a region.

What makes the task difficult are the mysterious rules of the Builders' Guild, in particular, the arcane Bylaws of Acuteness Prevention in Castles. These rules decree that wherever two walls meet, they must form an **interior** angle of at least 90 degrees. Also, the walls must enclose a single connected region.

Given the locations of the towers, decide where the walls should go. It does not matter how many of the existing towers you use, how large the resulting castle is, or how many bricks you use. All you need is a single polygonal castle from which to defend your lands.

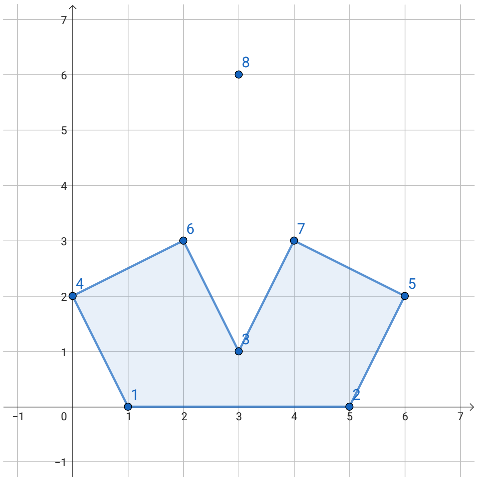


Figure L.1: Illustration of the third sample input, with one of the multiple possible ways to construct a castle.

Input

The input consists of:

- One line with an integer n ($3 \leq n \leq 3000$), the number of towers.
- n lines, each with two integers x and y ($|x|, |y| \leq 10^9$), the coordinates of the towers.

No two towers have the same coordinates.

Output

If it is possible to build a castle, output the number of towers used, followed by a list of (1-based) indices of the towers in either clockwise or counter-clockwise order. Otherwise, output "`impossible`".

If a tower t is located on a wall between two other towers, including t in the output is optional.

If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
5
0 0
1 0
-1 0
0 1
0 -1
```

Sample Output 1

```
4
2 4 3 5
```

Sample Input 2

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

Sample Output 2

```
impossible
```

Sample Input 3

```
8
1 0
5 0
3 1
0 2
6 2
2 3
4 3
3 6
```

Sample Output 3

```
7
1 2 5 7 3 6 4
```