ANZAC for BEGINNERS

2016 SEASON

Problems Sourced from Previous ANZACs
and
Regionals from various places

This page is intentionally left (almost) blank.

A: Arithmetic

Time Limit: 1 second(s)

It doesn't get much simpler than this. Calculate the sum of the two input numbers and the difference between the two input numbers.

Make sure you look at the input ranges and think about the data type needed to calculate the possible results.

Input

The input consists of two integer values, a and b ($-2\ 000\ 000\ 000 \leq a, b, \leq 2\ 000\ 000\ 000$), on a single line separated by a single space .

Output

Print, on a single line, firstly the sum of a and b and secondly the difference between a and b.

Sample Input and Output

Sample Input 1	Output for Sample Input
`10 2`	`12 8`

Sample Input 2	Output for Sample Input
`2 10`	`12 -8`

Sample Input 3	Output for Sample Input
`5 5`	`10 0`

Sample Input 4	Output for Sample Input
`-5 -5`	`-10 0`

Sample Input 5	Output for Sample Input
`2000000000 2000000000`	`4000000000 0`

This page is intentionally left (almost) blank.

B: Bottled Up

Time Limit: 1 second(s)

Peter is expecting a large shipment of fuel oil, but he has a small problem (doesn't everyone in these programming problems!). The only containers he has are a set of large bottles (each with the same volume) and a set of smaller bottles (also each with the same, but smaller volume). Given the volume of the shipment of oil, he would like to store the oil in the bottles so that:

1. all of the oil is stored,
2. each bottle is filled to the top, and
3. the minimum number of bottles is used.

While Peter thinks he has solved this problem for his given bottle sizes, he often spends hours wondering what would happen if his bottles had different volumes (apparently Peter doesn't lead the most exciting life).

Input

The input consists of a single line containing three positive integers, s, v_1 and v_2, where $s \leq 1\,000\,000$ is the volume of the shipment, and $v_1, v_2 \leq 1\,000\,000$ are the volumes of the two types of bottles, with $v_1 > v_2$.

Output

Output the number of bottles of size v_1 and the number of bottles of size v_2 which satisfy Peter's three conditions. If the conditions cannot be met, output `Impossible`.

Sample Input and Output

Sample Input 1	Output for Sample Input
`1000 9 7`	`108 4`

Sample Input 2	Output for Sample Input
`1000 900 7`	`Impossible`

Sample Input 3	Output for Sample Input
`1000 10 7`	`100 0`

This page is intentionally left (almost) blank.

C: Class Time

Time Limit: 1 second(s)

It's the first day of class! Tom is teaching class and first has to take attendance to see who is in class. He needs to call the students' names in alphabetical order by last name. If two students have the same last name, then he calls the students with that same last name in alphabetical order by first name. Help him!

Input

The first line of input contains an integer n ($1 \leq n \leq 100$), the number of students in Tom's class. Each of the following n lines contains the name of a single student: first name, followed by a single space, then last name. The first and last name both start with an upper-case letter ('A''Z') and these are followed by one or more lower-case letters ('a''z'). The first and last name of each student is no more than 10 letters long each.

It is guaranteed that no two students have exactly the same name, though students may share the same first name, or the same last name.

Output

Output n lines, the names of the students as Tom calls them in the desired order.

Sample Input and Output

Sample Input 1	Output for Sample Input
3 John Adams Bob Adam Bob Adams	Bob Adam Bob Adams John Adams

Sample Input 2	Output for Sample Input
1 Coursera Educators	Coursera Educators

This page is intentionally left (almost) blank.

D: Desert Bit Map

Time Limit: 1 second(s)

This problem requires you to search a black and white satellite image of a desert for a secret building complex with a given shape. A complex of this given shape may host an installation for producing the strategic xeenium macgillicudamate ingredient, and must keep its orientation with regard to cardinal axes (North-East-South-West). Rotations and mirror images are not allowed because they would interfere with the delicate alchemy required for the production process. You must determine how many times the given complex may possibly occur in the image.

Consider the following images, both on the same scale, where a # (sharp) is a "black" pixel representing a part of a building, and a . (dot) is a "white" pixel, representing sand. On the left is an image of the complex you are trying to locate, on the right is an image of the desert with some buildings on it.

```
#..
#.#
#..
```

```
#..... #........................#...
#.#....#.#.. #........ #............
#......#.#...##....#...##......###..
.......#.....#...#.#...##........#.#
...................#.....#.......#..
```

- How many possible locations for the given secret buildings do we count?
- The answer is *four*: one at the top-left corner, two overlapped possibilities to its right, and one in the bottom right. The shapes near the top-right corner, and in the centre bottom don't count because they are rotated (remember that rotated and/or mirrored images do not count).
- Note that, as this answer implies, the sand pixels in the image of the building complex simply establish the necessary relationships between the building parts. In the actual image they may contain *either* sand *or* other building parts (possibly for disguising the true nature of the complex).
- *Assume* that images representing strategic complexes are already trimmed of any unneeded dot "white" pixels on the edges, i.e., these images will always contain *at least one # character on each edge* (as our example shows). An edge here is the first or last row or column.

Input

Each test case will give you the specification for the building complex image followed by the specification for the desert image.

In each problem the input is:

- Line 1: 2 positive integers, b_1, b_2, respectively representing the number of lines and the number of columns in the following buildings image. Both numbers will be in the range 1 to 16 inclusive.
- Next b_1 lines: b_2 characters (# or .) on each line to represent part of the image of the building complex.
- Next Line: 2 positive integers, d_1, d_2, respectively representing the number of lines and the number of columns in the following Desert image. Both numbers will be in the range 1 to 64 inclusive.
- Next d_1 lines: d_2 characters (# or .) on each line to represent the desert image.

Output

The output for each test case consists of a single integer value on a line by itself being the number of matches found.

Sample Input and Output

Sample Input 1	Output for Sample Input

<pre>2 2
#.
##
3 5
#.#.#
#####
.###.</pre>
	<pre>4</pre>

Sample Input 2	Output for Sample Input

<pre>1 3
#.#
3 6
##..##
.#.#.#
#.#...</pre>
	<pre>3</pre>

Sample Input 3	Output for Sample Input

<pre>3 3
#..
#.#
#..
5 36
#......#........................#...
#.#....#.#...#.........#............
#......#.#...##....#...##......###..
.......#.....#...#.#...##........#.#
...................#.....#.......#..</pre>
	<pre>4</pre>

E: Excellence

Time Limit: 1 second(s)

The World Coding Federation is setting up a huge on-line programming tournament of teams comprised of pairs of programmers. Judge David is in charge of putting teams together from the Southeastern delegation. Every student must be placed on exactly one team of two students. Luckily, he has an even number of students who want to compete, so that he can make sure that each student does compete. However, he'd like to maintain his pristine reputation amongst other judges by making sure that each of the teams he fields for the competition meet some minimum total rating. We define the total rating of a team to be the sum of the ratings of both individuals on the team.

Help David determine the maximum value, X, such that he can form teams, each of which have a total rating greater than or equal to X.

Input

The first line of input contains a single positive integer $n (1 < n < 105, n$ is even), the number of students who want to enter the on-line programming tournament. Each of the following n lines contains one single integer s_i $(1 \leq s_i \leq 1\ 000\ 000)$, the rating of student i.

Output

Print, on a single line, the maximum value, X, such that David can form teams where every team has a total rating greater than or equal to X.

Sample Input and Output

Sample Input 1	Output for Sample Input
4 1 2 3 5	5

Sample Input 2	Output for Sample Input
2 18 16	34

Sample Input 3	Output for Sample Input
4 13 12 19 14	27

This page is intentionally left (almost) blank.

F: Footballers

Time Limit: 1 second(s)

The All Golds Rugby Club categorises members as Senior or Junior. Anyone over 17 years old is a `Senior`, as is anyone weighing 80 kg or more. Everyone else is a `Junior`. Your task is to correctly classify club members.

Input

The input consists of a single line containing a player's name followed by two positive integers, a ($5 \leq a \leq 60$) and w ($25 \leq w \leq 180$) being the age in years and weight in kilograms of the player. The player's name consists only of alphabetic characters i.e. there are no spaces in the name.

Output

Print, on a single line, the player's name, followed by a space, followed by the player's category, either `Junior` or `Senior`.

Sample Input and Output

Sample Input 1	Output for Sample Input
`Joe 16 34`	`Joe Junior`

Sample Input 2	Output for Sample Input
`Bill 18 65`	`Bill Senior`

Sample Input 3	Output for Sample Input
`Billy 17 65`	`Billy Junior`

Sample Input 4	Output for Sample Input
`Sam 17 85`	`Sam Senior`

This page is intentionally left (almost) blank.

G: Complexity

Time Limit: 1 second(s)

Define the *complexity* of a string to be the number of distinct letters in it. For example, the string `string` has complexity 6 and the string `letter` has complexity 4.

You like strings which have complexity either 1 or 2. Your friend has given you a string and you want to turn it into a string that you like. You have a magic eraser which will delete one letter from any string. Compute the minimum number of times you will need to use the eraser to turn the string into a string with complexity at most 2.

Input

The input consists of a single line that contains a single string of at most 100 lowercase ASCII letters (`'a'`–`'z'`).

Output

Print, on a single line, the minimum number of times you need to use the eraser.

Sample Input and Output

Sample Input 1	Output for Sample Input
`string`	`4`

Sample Input 2	Output for Sample Input
`letter`	`2`

Sample Input 3	Output for Sample Input
`aaaaaa`	`0`

Sample Input 4	Output for Sample Input
`uncopyrightable`	`13`

Sample Input 5	Output for Sample Input
`ambidextrously`	`12`

Sample Input 6	Output for Sample Input
`assesses`	`1`

Sample Input 7	Output for Sample Input
`assassins`	`2`

This page is intentionally left (almost) blank.

H: Vowels

Time Limit: 1 second(s)

The English alphabet consists of 26 letters. Five of these (`a`, `e`, `i`, `o` and `u`) are classified as vowels, the remaining 21 as consonants. Almost every English word contains at least one vowel (`rhythm` is one of the few exceptions).

In this problem you will be given a piece of English text. Your task is to determine the frequency of each vowel that is found in the piece, and to display the answers sorted by frequency, highest frequency first. Where two vowels are equally frequent, they are to be displayed in alphabetical order.

As you can see from the examples below, upper case and lower case letters are considered to be the same letter in this problem. Use lower case in your output. As you can see from the second example, a frequency of zero must still be displayed.

Input

Input will consist of a single line of text with length 0 to 200 characters, inclusive.

Output

Output a single line with each vowel in lower case, followed by a colon, followed by the frequency of that vowel. There must be one space between each vowel and its count.

Sample Input and Output

Sample Input 1	Output for Sample Input
`Ugh!!`	`u:1 a:0 e:0 i:0 o:0`

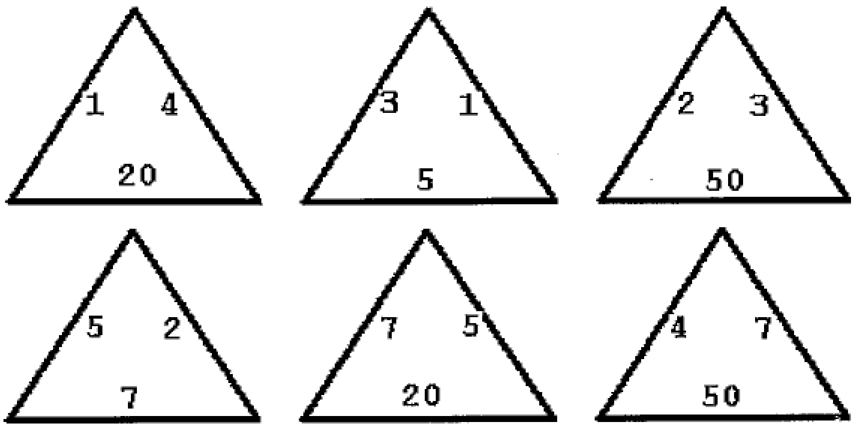
Sample Input 2	Output for Sample Input
`This piece of text was written in the city of Auckland.`	`e:5 i:5 a:3 o:2 u:1`

Sample Input 3	Output for Sample Input
`ACM Programming Contest.`	`a:2 o:2 e:1 i:1 u:0`

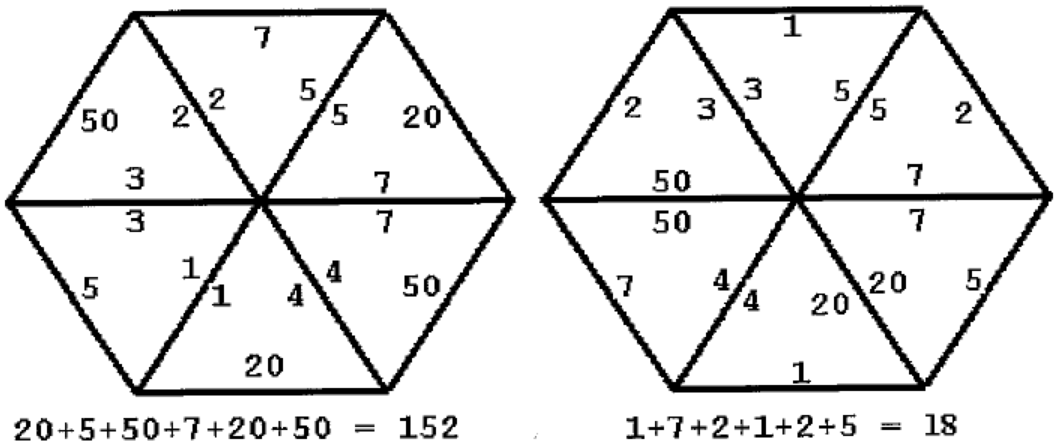
This page is intentionally left (almost) blank.

I: Triangle Game

Time Limit: 2 second(s)



In the triangle game you start off with six triangles numbered on each edge, as in the example above. You can slide and rotate the triangles so they form a hexagon but the hexagon is only legal if edges common to two triangles have the same number on them. You may not flip any triangle over. Two legal hexagons formed from the six triangles are illustrated below.



The score for a legal hexagon is the sum of the numbers on the outside six edges.

Your problem is to find the highest score that can be achieved with any six particular triangles.

Input

The input will contain one test case which consists of six lines with three integers separated by spaces. All values will be between 1 and 100 inclusive. Each line contains the numbers on the triangles in clockwise order. Compare the top image with the first sample input for clarification.

Output

On a single line output only the word `none` if there are no legal hexagons or the highest score if there is a legal hexagon.

Sample Input and Output

Sample Input 1	Output for Sample Input
1 4 20 3 1 5 50 2 3 5 2 7 7 5 20 4 7 50	152

Sample Input 2	Output for Sample Input
10 1 20 20 2 30 30 3 40 40 4 50 50 5 60 60 6 10	21

Sample Input 3	Output for Sample Input
10 1 20 20 2 30 30 3 40 40 4 50 50 5 60 10 6 60	none

J: Jelly

Time Limit: 1 second(s)

A local school provides jelly for their pupils every day and the school staff are very careful to see that each child has exactly the same amount.

The jelly is prepared the previous day by pouring the liquid jelly into rectangular sided moulds, one mould per child, and then put in the fridge where it sets. The moulds may differ by the length and width of their sides but are filled to different heights so that they all have the same volume. Length, width, and height are always integer values.

Unfortunately, one of the cleaners loves practical jokes! Whenever he can, before the jelly has set, he tips liquid jelly from one of the moulds into another. He is happy if he succeeds just once and doesn't repeat the joke with other moulds.

Your task is to help the school staff by preparing a report for them. They need to know who has lost jelly and who has gained it so that they can correct matters before the children arrive.

Input

The input consists of one test case. The test case begins with a single integer n, $1 \leq n \leq 100$, representing the number of children for whom jelly was prepared. Following this are n lines, each line representing one child. The data for a child consists of the child's name followed by a single space and three space-separated integers, l, w and h ($1 \leq l, w, h \leq 100$) being the length, width and height of the jelly in that child's mould. A child's name consists of a sequence of 1 up to 10 alphabetic characters (upper and/or lower case). No two children have the same name.

Output

Output consists of one line of text. If the cleaner did not manage to transfer any jelly before it set, output:

`No child has lost jelly.`

If the cleaner did manage to transfer jelly, your output must be in the form:

`ChildA` `has lost jelly to` *`ChildB`*`.`

Sample Input and Output

Sample Input 1	Output for Sample Input
3 Joe 10 10 2 Susan 10 5 4 Bill 5 5 8	No child has lost jelly.

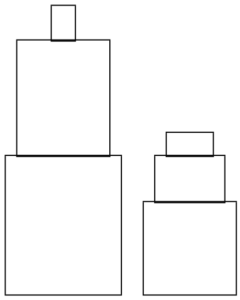
Sample Input 2	Output for Sample Input
4 Zoe 10 2 2 Lee 6 5 2 Alan 5 4 4 Tommy 12 5 1	Zoe has lost jelly to Alan.

This page is intentionally left (almost) blank.

K: Towering Problem

Time Limit: 1 second(s)

You've been put in charge of an art exhibit from the famous minimalist sculptor J (even his name is minimalist!). J's work involves the careful layout of vertically dispositioned orthogonal parallelepipeds in a set of tapering obelisks — in other words, he puts smaller boxes on top of larger boxes. His most recent triumph is called "2 by 3's Decreasing," in which he has various sets of six boxes arranged in two stacks of three boxes each. One such set is shown below:



J has sent you the art exhibit and it is your job to set up each of the six-box sets at various locations throughout the museum. But when the sculptures arrived at the museum, uncultured barbarians (i.e., delivery men) simply dropped each set of six boxes on the floor, not realizing the aesthetic appeal of their original layout. You need to reconstruct each set of two towers, but you have no idea which box goes on top of the other! All you know is the following: for each set of six, you have the heights of the two towers, and you know that in any tower the largest height box is always on the bottom and the smallest height box is on the top. Armed with this information, you hope to be able to figure out which boxes go together before tomorrow night's grand opening gala.

Input

The input consists of eight positive integers. The first six represent the heights of the six boxes. These values will be given in no particular order and no two will be equal.

The last two values (which will never be the same) are the heights of the two towers.

All box heights will be ≤ 100 and the sum of the box heights will equal the sum of the tower heights.

Output

Output the heights of the three boxes in the first tower (i.e., the tower specified by the first tower height in the input), then the heights of the three boxes in the second tower. Each set of boxes should be output in order of decreasing height. Each test case will have a unique answer.

Sample Input and Output

Sample Input 1	Output for Sample Input
`12 8 2 4 10 3 25 14`	`12 10 3 8 4 2`

Sample Input 2	Output for Sample Input
`12 17 36 37 51 63 92 124`	`63 17 12 51 37 36`

This page is intentionally left (almost) blank.

L: Post Office

Time Limit: 1 second(s)

Other than postcards, the post office department of some country recognizes three classes of mailable items: letters, packets and parcels. The three dimensions of a mailable item are called length, height and thickness, with length being the largest and thickness the smallest of the three dimensions.

A letter's length must be at least 125 mm but not more than 290 mm, its height at least 90 mm but not more than 155 mm and its thickness at least 0.25 mm but not more than 7 mm.

All three of a packet's dimensions must be greater than or equal to the corresponding minimum dimension for a letter and at least one of its dimensions must exceed the corresponding maximum for a letter. Furthermore, a packet's length must be no more than 380 mm, its height no more than 300 mm and its thickness no more than 50 mm.

All three of a parcel's dimensions must be greater than or equal to the corresponding minimum dimension for a letter and at least one of its dimensions must exceed the corresponding maximum for a packet. Furthermore, the parcel's combined length and girth may not exceed 2 100 mm. The girth is the full perimeter measured around the parcel, perpendicular to the length.

Input

The input will contain data for a number of problem instances. For each problem instance, the input will consist of the three dimensions of an item, measured in mm, in any order. The length and width will be positive integers. The thickness will be either a positive integer or a positive floating point number.

The input will be terminated by a line containing three zeroes.

Output

For each problem instance, output the classification of the item on a single line as `letter`, `packet`, `parcel` or `not mailable`. Use only lower case letters.

Sample Input and Output

Sample Input 1	Output for Sample Input
100 120 100 0.5 100 200 100 10 200 200 75 100 0 0 0	not mailable letter packet parcel

This page is intentionally left (almost) blank.

M: Magic Trick

Time Limit: 1 second(s)

Your friend has come up with a math trick that supposedly will blow your mind. Intrigued, you ask your friend to explain the trick.

First, you generate a random positive integer, k, between 1 and 100 inclusive. Then, your friend will give you n operations to execute. An operation consists of one of the four arithmetic operations `ADD`, `SUBTRACT`, `MULTIPLY`, or `DIVIDE`, along with an integer-valued operand x. You are supposed to perform the requested operations in order.

You don't like dealing with fractions or negative numbers though, so if during the process, the operations generate a fraction or a negative number, you will tell your friend that he messed up.

You know the n operations your friend will give. How many of the first 100 positive integers, i.e. 1 to 100, will cause your friend to mess up?

Input

The first line of input contains a single positive integer n ($1 \leq n \leq 10$). Each of the next n lines consists of an operation, followed by an operand. The operation is one of the strings `ADD`, `SUBTRACT`, `MULTIPLY`, or `DIVIDE`. Operands are positive integers not exceeding 5.

Output

Print, on a single line, a single integer indicating how many of the first 100 positive integers will result in you telling your friend that he messed up.

Sample Input and Output

Sample Input 1	Output for Sample Input
1 SUBTRACT 5	4

Sample Input 2	Output for Sample Input
1 DIVIDE 2	50

Sample Input 3	Output for Sample Input
2 ADD 5 DIVIDE 5	80