

# Elections

## Problem Description

Elections are going on, and there are two candidates A and B, contesting with each other. There is a queue of voters and in this queue some of them are supporters of A and some of them are supporters of B. Many of them are neutral. The fate of the election will be decided on which side the neutral voters vote. Supporters of A and supporters of B make attempt to win the votes of neutral voters.

The way this can be done is explained below:

1. The voter queue is denoted by three characters, viz {-, A, B}. The - denotes neutral candidate, A denotes supporter of candidate A and B denotes supporter of candidate B.
2. Supporters of A can only move towards the left side of the queue.
3. Supporters of B can only move towards the right side of the queue.
4. Since time is critical, supporters of both A and B will move simultaneously.
5. They both will try and influence the neutral voters by moving in their direction in the queue. If supporter of A reaches the neutral voter before supporter of B reaches him, then that neutral voter will become a supporter of candidate A.
6. Similarly, if supporter of B reaches the neutral voter before supporter of A reaches him, then that neutral voter will become a supporter of candidate B.
7. Finally, if both reach at the same time, the voter will remain neutral. A neutral vote cannot decide the outcome of the election.
8. If finally, the queue has more votes for candidate A, then A wins the election. If B has more votes, then B wins that election. If both have equal votes, then it will be a coalition government.

Refer Examples section for understanding the dynamics of how the supporters influence the neutral voters.

Your task is to find the outcome of the election.

**Note:** There are no test cases where all votes are neutral.

## Constraints

$1 \leq \text{length of queue} \leq 10^5$

## Input

First line contains an integer which is length of queue of voters.

Second line contains characters {-, A, B}, in which denotes

- A = voter who is supporter of candidate A
- B = voter who is supporter of candidate B
- - = neutral voter

## Output

Print candidate with maximum number of votes. If they have equal number of votes, print "Coalition government".

## Time Limit

1

## Examples

### Example 1

Input

14

--AB--AB---A--

Output

A

Explanation:

For starting positions where there is no opposition from supporter of B, supporter of A can promote in left side of the queue. The voting queue will then look like below:

A A A B - - A B - - - A - -

From 4<sup>th</sup> place (in voting queue) B supporter is moving towards the right side, simultaneously 7<sup>th</sup> placed A supporter is also moving towards the left side. Then the voting queue will look like below:

A A A B B A A B - - - A - -

From 8<sup>th</sup> place B supporter is moving towards the right side, simultaneously 12<sup>th</sup> placed A supporter is also moving towards the left side. Then the voting queue will look like below:

A A A B B A A B B - A A - -

Since supporters of both A and B will reach the 10<sup>th</sup> voter at the same time, 10<sup>th</sup> voter will remain neutral.

Since supporter of A at 12<sup>th</sup> place cannot move towards right, last 2 voters will not be influenced and remain neutral. Then the voting queue will look like below:

A A A B B A A B B - A A - -

Since all voter have now cast their votes, election results can now be declared.

So final result is: A A A B B A A B B - A A - -

A has 7 votes, B has 4 votes hence, A wins the election.

### Example 2

Input

4

A---

Output

A

Explanation:

Since supporter of A at 1<sup>st</sup> place cannot move towards right, last 3 voters will not be influenced and will remain neutral. Then the voting queue will look like below:

A - - -

Since all voter have now cast their votes, election results can now be declared.

So final result is: A - - -

A has 1 vote, B has 0 votes hence, A wins the election.

Example 3

Input

5

A---B

Output

Coalition government

Explanation:

Since supporter of A at 1<sup>st</sup> place cannot move towards right, supporter of B at 5<sup>th</sup> cannot move towards left, middle 3 voters will not be influenced and will remain neutral. Then the voting queue will look like below:

A - - - B

So final result is: A - - - B

A has 1 vote, B has 1 vote hence, output will be "Coalition government".

# Single Lane Highway

## Problem Description

Certain number of cars are passing a single lane road. Speeds of all cars vary. It is easy to see, that depending on the speeds of the cars various groups will be formed.

Being a single lane road passing/overtaking is not allowed. Given speeds of cars, calculate how many groups can be formed if all possible permutations are taken into account. Refer example1 for better understanding.

Print number of groups divided by the number of permutations.

## Constraints

$0 \leq N < 10^5$

$0 \leq \text{speed of individual vehicle} < 10^9$

## Input

First line contains an integer N, which denotes the number of vehicles

Second line contains N space separated integers which denotes the speed of individual vehicle.

## Output

Print number of groups divided by the number of permutations rounded upto 6 decimal places.

## Time Limit

1

## Examples

Example 1

Input

3

10 20 30

Output

1.833333

Explanation:

So all possible permutations are:

{10 20 30}

{10 30 20}

{20} {10 30}

{20 30} {10}

{30} {10 20}

{30 20} {10}

So here there are total 6 permutations, and total number of groups are 11.

So, output is  $11/6 = 1.833333$

## Example 2

Input

4

56 78 13 92

Output

2.083333

Explanation:

So here there are total 24 permutations,

For example:

{56 78 13 92}

{92} {13 78 56}

{56} {13 92 78}

{78 92} {13 56}

.

.

So on and so forth. The total number of groups are 50.

So, the output is  $50/24 = 2.083333$

# Corona Virus

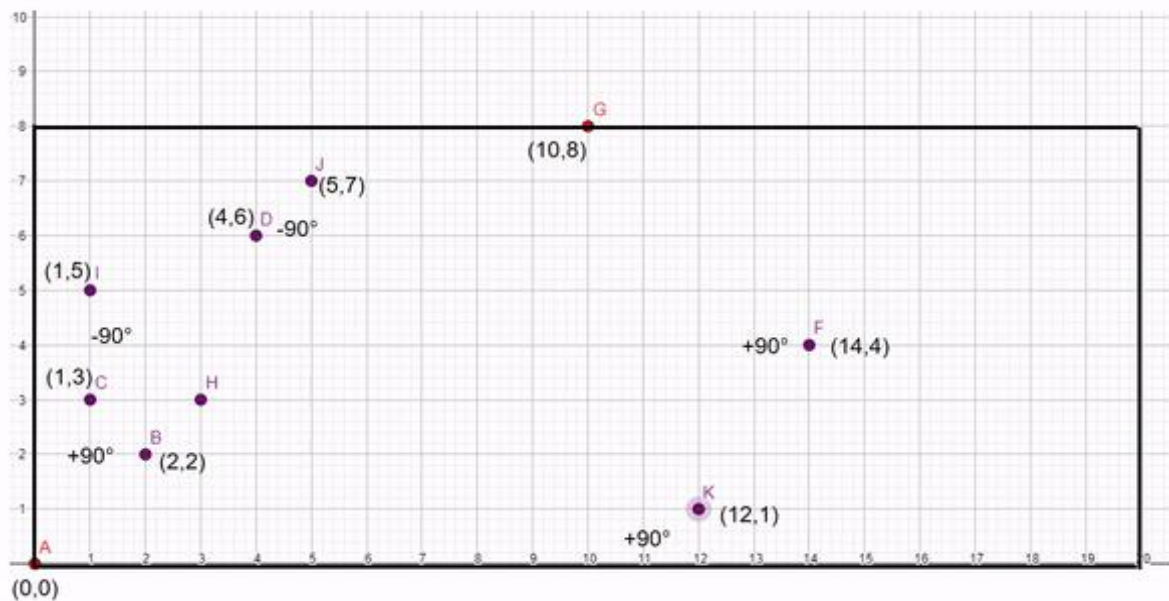
## Problem Description

A city is represented as two-dimensional rectangular matrix. The outer wall of the given matrix denotes the boundaries of the city. Citizens are dispersed in the city at different locations. They are either depicted by {a, c}. Corona Virus has already infected the city.

The Corona Virus enters the city from coordinate (0, 0) and traverses along a diagonal path until it encounters a human. If it encounters a human, designated as **a**, its trajectory rotates anti clockwise (right to left) by 90 degrees. Similarly, if it encounters a human, designated as **c**, its trajectory rotates clockwise (left to right) by 90 degrees. After infecting the person, the virus continues to move along its diagonal path.

During its traversal if it hits the boundary of the city for the first time, it rotates 90 degrees to reenter the city. However if it hits any of the boundary wall, the second time, the virus gets destroyed.

You have to calculate the trajectory taken by the virus, print the city map where infected citizens can be found and finally report the number of safe and infected citizens.



## Input

An input matrix of 9 rows and 20 columns comprising of {\*, a, c, .} characters where

- \* denotes an element on the boundaries of the city
- a denotes citizen after encountering whom the virus trajectory changes by 90 degrees (anti clockwise direction)
- c denotes citizen after encountering whom the virus trajectory changes by 90 degrees (clockwise direction)
- . (dot) denotes empty location within the city

## Output

Random number of lines each denoting the coordinates of the trajectory of the virus.

From next line an output matrix of 9 rows and 20 columns comprising of {\*, a, c, ., - } characters where

- \* denotes an element on the boundaries of the city
- a denotes citizen after encountering whom the virus trajectory changes by 90 degrees (anti clockwise direction)
- c denotes citizen after encountering whom the virus trajectory changes by 90 degrees (clockwise direction)
- . (dot) denotes empty location within the city
- - denotes the location of the infected citizen

And the next two lines print the number of safe and infected citizens in the city

Refer Examples section for better understanding.

## Constraints

$$0 \leq x \leq 20$$

$$0 \leq y \leq 8$$

The virus cannot hit the three corners (20, 8) (20, 0) (0, 8)

## Time Limit

1

## Examples

Example 1

Input

\*\*\*\*\*

\*...C.....\*

\*...C.....\*

\*C.....\*

\*.....a....\*

\*C.C.....\*

\*.a.....\*

\*.....C.....\*

\*\*\*\*\*

## Output

0 0

1 1

2 2

1 3

2 4

3 5

4 6

5 5

6 4

7 3

8 2

9 1

10 0

11 1

12 2

13 3

14 4

13 5

12 6

11 7

10 8

\*\*\*\*\*

\* ...C..... \*

\* ...-..... \*

\* C..... \*

\* .....-... \*

\* -.C..... \*

\* .-..... \*



```
* .....c..... *
```

```
*****
```

safe=4

infected=4

### Explanation

The virus trajectory starts from (0,0) and crosses (1,1) (2,2). At (2,2) we have a citizen of type **a** causing the virus trajectory to be rotated by 90 degrees anti clockwise. It moves until the next citizen in its path is encountered at (1, 3) of type **c** causing the virus trajectory to be rotated by 90 degree clockwise. It continues on its path till it reaches the next human in its path at (4,6) of type **c** Virus trajectory is again rotated by 90 degrees clockwise until it hits the boundary at (10,0). Since this is the first time that the virus hits the boundary, it rotates by 90 degrees anticlockwise to reenter the city. The trajectory then continues towards (11,1) (12,2) (13,3) and finally a citizen at (14,4) of type **a** rotating the trajectory to 90 degree anticlockwise. From there it continues its trajectory and hits the boundary at (10,8).

Since this is the second time the virus hits the boundary, the virus is destroyed.

So, along its trajectory starting from (0,0) it has infected 4 citizens at location (2,2) (1,3) (4,6) (14,4). The other 4 citizens who did not come in the virus trajectory are deemed to be safe.

### Example 2

#### Input

```
*****
```

```
* ..... *
```

```
* ..c..... *
```

```
* ...c..... *
```

```
* .....a..... *
```

```
* ..... *
```

```
* .....a.....c... *
```

```
* ..... *
```

```
*****
```

#### Output

0 0

1 1

2 2

3 3

4 4

5 5

6 4

7 3

8 2

9 3

10 4

9 5

8 6

7 7

6 8

5 7

4 6

3 5

2 4

1 3

0 2

\*\*\*\*\*

\* ..... \*

\* ..C..... \*

\* ...-..... \*

\* .....-..... \*

\* ..... \*

\* .....-.....C... \*

\* ..... \*

\*\*\*\*\*

safe=2

infected=3

Explanation

The virus trajectory starts from (0,0) and crosses (1,1) (2,2) (5,5). At (5,5) we have a citizen of type c causing the virus trajectory to be rotated by 90 degrees clockwise. It moves until the next citizen in its path is encountered at (8,2) of type a causing the virus trajectory to be rotated by 90 degree

anti clockwise. It continues on its path till it reaches the next human in its path at (10,4) of type **a** Virus trajectory is again rotated by 90 degrees clockwise until it hits the boundary at (6,8). Since this is the first time that the virus hits the boundary, it rotates by 90 degrees anticlockwise to reenter the city. The trajectory then continues towards (9,5) (8,6) (7,7) (6,8) and reenters the city by rotating the trajectory by 90 degrees anti clockwise to follow the trajectory (5,7) (4,6) (3,5) (2,4) (1,3) (0,2).

At (0,2) it again hits the boundary and since this is the second time the virus hits the boundary, the virus is destroyed. So along its trajectory starting from (0,0) it has infected 3 citizens at location (5,5) (10,4) (8,2). The other 2 citizens who did not come in the virus trajectory are deemed to be safe.

# String Pair

## Problem Description

One person hands over the list of digits to Mr. String, But Mr. String understands only strings. Within strings also he understands only vowels. Mr. String needs your help to find the total number of pairs which add up to a certain digit D.

The rules to calculate digit D are as follow

Take all digits and convert them into their textual representation

Next, sum up the number of vowels i.e. {a, e, i, o, u} from all textual representation

This sum is digit D

Now, once digit D is known find out all unordered pairs of numbers in input whose sum is equal to D. Refer example section for better understanding.

## Constraints

$1 \leq N \leq 100$

$1 \leq \text{value of each element in second line of input} \leq 100$

Number 100, if and when it appears in input should be converted to textual representation as hundred and not as one hundred. Hence number of vowels in number 100 should be 2 and not 4

## Input

First line contains an integer N which represents number of elements to be processed as input

Second line contains N numbers separated by space

## Output

Lower case representation of textual representation of number of pairs in input that sum up to digit D

Note: - (If the count exceeds 100 print "greater 100")

## Time Limit

1

## Examples

Example 1

Input

5

1 2 3 4 5

## Output

one

## Explanation

1 -> one -> o, e

2 -> two -> o

3 -> three -> e, e

4 -> four -> o, u

5 -> five -> i, e

Thus, count of vowels in textual representation of numbers in input =  $\{2 + 1 + 2 + 2 + 2\} = 9$ . Number 9 is the digit D referred to in section above.

Now from given list of number  $\{1,2,3,4,5\}$  -> find all pairs that sum up to 9.

Upon processing this we know that only a single unordered pair  $\{4, 5\}$  sum up to 9. Hence the answer is 1. However, output specification requires you to print textual representation of number 1 which is one. Hence output is one.

Note: - Pairs  $\{4, 5\}$  or  $\{5, 4\}$  both sum up to 9. But since we are asking to count only unordered pair, the number of unordered pair is this combination is only one.

## Example 2

### Input

3

7 4 2

### Output

zero

### Explanation

7 -> seven -> e, e

4 -> four -> o, u

2 -> two -> o

Thus, count of vowels in textual representation of numbers in input =  $\{2 + 2 + 1\} = 5$ . Number 5 is the digit D referred to in section above.

Since no pairs add up to 5, the answer is 0. Textual representation of 0 is zero. Hence output is zero.

# Zoo Design

## Problem Description

Aman is a rich businessman who want to build a zoo. He wants to make enclosures for terrestrial and aquatic animals. Terrestrial animals will be of two types, namely herbivorous and carnivorous animals. So there will be three different enclosures.

Herbivores like Elephant, Deer are prime attractions. Similarly, Lion and Tiger are prime attractions amongst carnivores. Finally, Dolphins and Shark are prime attractions amongst aquatics for tourists.

Aman being a savvy businessman realizes that in order to minimize the cost of building the zoo without compromising on the attractions, he has to decide how much area to allocate to each animal type. Each animal type requires a certain area to thrive in. This in turn impacts the area allocation, which in turn has cost implications.

Your task is to help Aman workout the mathematics such that the zoo building cost is minimized subject to the following constraints:

- Zoo needs to have minimum of X herbivores, Y carnivores and Z aquatic animals
- Different types of animals will need different minimum area to thrive in
- For animals of a given type, the minimum area required is the same
- There is also a maximum limit for the overall area allocated for each animal type
- Cost of fencing etc. is included in cost of enclosure
- Exclude the essentials like pathways for tourists, from area and cost calculations

Consider all areas in square meters and cost in Rupees.

## Constraints

$0 < \text{number of animals of each type} \leq 20$

$0 < \text{max area for each animal type} \leq 500$

$0 < \text{total area of land on which zoo is to be built} \leq 1000$

$0 < \text{min area required by individual animals} \leq 15$

$0 < \text{price of each type of enclosure} \leq 10000$

Area of each type of enclosure should be rounded off to the nearest integer

## Input

First line contains three space separated integers denoting the cost per square meter of building the enclosure for each type of animals viz. herbivorous, carnivorous and aquatic respectively

Second line contains three space separated integers denoting the maximum area that can be allocated to each type of animal viz. herbivorous, carnivorous and aquatic respectively

Next three lines, each will contain two space separated integers M and N, for each type of animal viz. herbivorous, carnivorous and aquatic respectively, where M denotes minimum number of animals of that type and N denotes minimum area required for that animal type

Last line contains an integer which represents the total area of land on which the zoo needs to be built

## Output

Single integer containing the minimum cost required to build the zoo

## Time Limit

1

## Examples

### Example 1

Input

10000 1000 1500

250 250 300

5 5

15 5

10 10

500

Output

837500

Explanation

·The cost of constructing the enclosure for herbivores is high. However, since we need to accommodate 5 herbivores as per given constraints, a 25 sq. meter land will need to be allocated for the herbivores.

·Since the cost of constructing the enclosure for carnivores is cheapest we are able to allocate them the maximum limit that we can allocate. Thus we are allocating 250 sq. meters for carnivores.

·The remaining 225 sq. meters can thus be allocated to aquatics without violating any constraint.

·Thus the minimum cost of constructing the zoo adhering to all constraints is  $(25 * 10000 + 250 * 1000 + 225 * 1500) = 837500$

### Example 2

Input

100 1000 1500

250 250 300

5 5

15 5

10 10

500

Output

325000

Explanation

· Since the cost of constructing the enclosure for herbivores is cheapest we are able to allocate them the maximum limit that we can allocate. Thus we are allocating 250 sq. meters for herbivores.

· The cost of constructing the enclosure for aquatics is high. However, since we need to accommodate 10 aquatics as per given constraints, a 100 sq. meter land will need to be allocated for the aquatic animals.

· The remaining 150 sq. meters can thus be allocated to carnivores without violating any constraint.

· Thus the minimum cost of constructing the zoo adhering to all constraints is  $(250 * 100 + 150 * 1000 + 100 * 1500) = 325000$



## 3 Palindrome

### Problem Description

Given an input string word, split the string into exactly 3 palindromic substrings.

Working from left to right, choose the smallest split for the first substring that still allows the remaining word to be split into 2 palindromes.

Similarly, choose the smallest second palindromic substring that leaves a third palindromic substring.

If there is no way to split the word into exactly three palindromic substrings, print "Impossible" (without quotes). Every character of the string needs to be consumed.

Cases not allowed -

After finding 3 palindromes using above instructions, if any character of the original string remains unconsumed.

No character may be shared in forming 3 palindromes.

### Constraints

$1 \leq \text{the length of input string} \leq 1000$

### Input

First line contains the input string consisting of characters between [a-z].

### Output

Print 3 substrings one on each line.

### Time Limit

1

### Examples

Example 1

Input

nayannamantenet

Output

nayan

naman

tenet

Explanation

The original string can be split into 3 palindromes as mentioned in the output.

However, if the input was nayanamantenet, then the answer would be "Impossible".

#### Example 2

Input

aaaaa

Output

a

a

aaa

Explanation

The other ways to split the given string into 3 palindromes are as follows -

[a, aaa, a], [aaa, a, a], [aa, aa, a], etc.

Since we want to minimize the length of the first palindromic substring using left to right processing, the correct way to split is [a, a, aaa].

#### Example 3

Input

aaaabaaaa

Output

a

aaabaaa

a

Explanation

The other ways to split the given string into 3 palindromes are as follows -

[aaaa, b, aaaa], [aa, aabaa, aa], etc.

Since we want to minimize the length of the first palindromic substring using left to right processing, the correct way to split is [a, aaabaaa, a].