01 Fill the cube

Problem Description

A company manufactures walls which can be directly implanted at the site.
The company uses small square bricks of material C and material D which have similar looks but have huge
difference in quality.
The company manufactures walls of square shapes only to optimize their costs.

A novice employee created a square wall using bricks of material C and D.
However, the client had asked the wall to be made of only high-quality material - material C.

To solve this problem, they will place the wall in a special furnace and heat it such that the material D melts and only material C remains.
Material C brick will move down due to gravity if a material D brick below it melts.
The new empty space created will be filled by new material C square walls.
They also want to use biggest possible C square wall while building the final wall.
For this they will position the wall in the furnace in an optimal way i.e.
rotate by 90-degrees any number of times, if required, such that the biggest space possible for new material
C wall is created.
No rotations are possible when the furnace starts heating.

Given the structure of the original wall created by the novice employee,
you need to find out the size of the new C square wall which can be fitted
in the final wall which will be delivered to the client.

Constraints

$1 < N < 100$

Input

First Line will provide the size of the original wall N.

Next N lines will provide the type of material (C and D) used for each brick by the novice employee.

Output

Size of the biggest possible C square wall which can be fitted in the final wall.

Time Limit

1

Examples

Example 1

Input

4

C D C D

C C D C

D D D D

C D D D

Output

3

Explanation

If the wall is placed with its left side at the bottom, space for a new C wall of size 2x2 can be created. This can be visualized as follows

D C D D

C D D D

D C D D

C C D C

The melted bricks can be visualized as follows

- - - -

- C - -

C C - -

C C - C

Hence, the maximum wall size that can be replaced is 2x2.

If the wall is placed as it is with its original bottom side at the bottom,
space for a new C wall of size 3x3 can be created. Post melting, this can be visualized as follows.

- - - -

C - - -

C - - -

C C C C

Hence, the maximum wall size that can be replaced is 3x3 in this approach.

Since no rotations followed by heating is going to a yield a space greater than 3x3, the output is 3.

Example 2

Input

7

C D D C D D D

C D D C D D D

D D D D D D C

D C D C D D D

D D D C D C D

C D D C D C C

C D C D C C C

Output

5

Explanation

If the wall is placed with its left side at the bottom, a space for new C wall of size 5x5 can be created. This can be visualized as follows

D D C D D C C

D D D D C C C

D D D D D D C

C C D C C C D

D D D D D D C

D D D C D D D

C C D D D C C

When this orientation of the wall is heated, a space for new C wall of size 5x5 is created after the D bricks melt

_ _ _ _ _ _ _

_ _ _ _ _ _ _

_ _ _ _ _ _C

_ _ _ _ _ _ C

_ _ _ _ _ C C

C C _ C C C C

C C C C C C C

Whereas, if the rotation was not done, the wall formed after the D bricks melt will be as follows

_ _ _ _ _ _ _

_ _ _ _ _ _ _

_ _ _ C _ _ _

C _ _ C _ _ _

C _ _ C _ _ C

C _ _ C _ C C

C C C C C C C

When this orientation of the wall is heated, a space for new C wall of size 3x3 only is created after the D bricks melt

Hence rotation is important and correct answer is 5x5

Since no rotations followed by heating is going to a yield a space greater than 5x5, the output is 5.

---

02 Binary Equivlent

Binary Equivalent

Problem Description

Mr. Binary is lost and wants to be found but the problem is he understands only binary. His house is located at a maximum binary equivalence possible, from the given set of numbers. A set is a binary equivalence if the number of 0 zeros and ones from a set of number are equal.

Constraints

1 <= N <= 20

1 <= Arr[i] <= 10^5, where Arr[i] is the ith element in the set of N numbers in second line of input

Arr[i] will be unique

Input

First line contains N denoting the number of decimal numbers

Next line contains N space separated decimal numbers

Output

Single line output printing possible binary equivalence where number of digits in this number is equal to number of bits present in the largest element in second line of input. If there is no set which has binary equivalence then return 0 padded to number of bits present in the largest element in second line of input.

Time Limit

1

Examples

Example 1

Input

3

2 7 10

Output

0011

Explanation

2 -> 0010 - 1's = 1, 0's = 3

7 -> 0111 - 1's = 3, 0's = 1

10 -> 1010 - 1's = 2, 0's = 2

Here we have taken up to 4 bits because the maximum number is 10 which needs 4 bits to be represented in binary. The number of zeroes and ones across the set is, 6 each. Hence, the set of [2,7,10] has binary equivalence. Similarly, if you consider set[2,7], it also has binary equivalence, 4 each. But set [7,10] does not have binary equivalence. Likewise, set[10] has binary equivalence of 2 each.

Total number of unique sets where binary equivalence is possible from all combinations are 3 viz. Sets are [2,7,10], [2,7] and [10] which is the final answer. But as Mr. Binary only understands zeroes and ones, return the binary of 3.

Since 10 is the largest element in the input on line 2, the number of bits required to represent 10 in binary is 4. Hence output needs to be padded upto 4 digits. Since binary of 3 represented as a 4-digit number is 0011, the answer is 0011

Note

Do not consider empty subset

Example 2

Input

1

7

Output

000

Explanation

7 -> 111 - 1's = 3, 0's = 1

Since there is only one element in the set and it also does not have binary equivalence, the answer is 0. However, keeping output specifications in mind, the answer should be printed as 000 since the highest element in second line of input viz. 7 has 3 bits when represented in binary format.

---

03 Largest Gold Ingot

Largest Gold Ingot

Problem Description

Ramesh is a goldsmith, who brought a large number of gold ingot each of different length(L) but equal breadth(B) and height(H). He wants to weld the ingots of same length with each other. He tasks his new employee, Akash, to weld the ingots of same length with each other. But Akash forgot that he had to weld the ingots of same length, instead he welded the ingots in a random manner.

Later Ramesh found out what he had done. He then ordered Akash to cut the welded ingot such that a cuboid with the largest volume from the welded gold ingot is obtained.

Find the volume of summation of gold ingots minus volume of the largest cuboid.

Constraints

0 < G < 10^5

Input

First Line contains one integer G, denoting number of gold ingots

Second line contains two space separated integers B and H, where B denotes the breadth and H denotes the height of individual ingot

Third line contains G space separated integers, denoting the length of the individual gold ingots that are welded together in adjacent manner

Output

An integer corresponding to the volume of summation of gold ingots minus volume of the largest cuboid, mod 10^9+7.

Time Limit

1

Examples

Example 1

Input

7

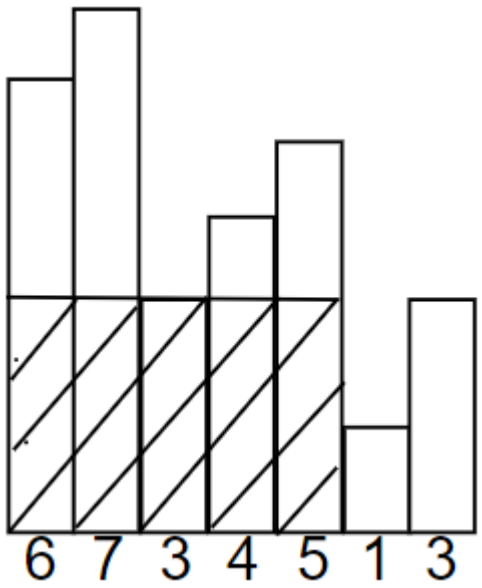
1 1

6 7 3 4 5 1 3

Output

14

Explanation



Total volume of shaded region is 15 and the total volume is 29. So the volume of summation of gold ingots minus largest cuboid obtained is 14, since the height is 1 and breadth is 1.

Example 2

Input

7

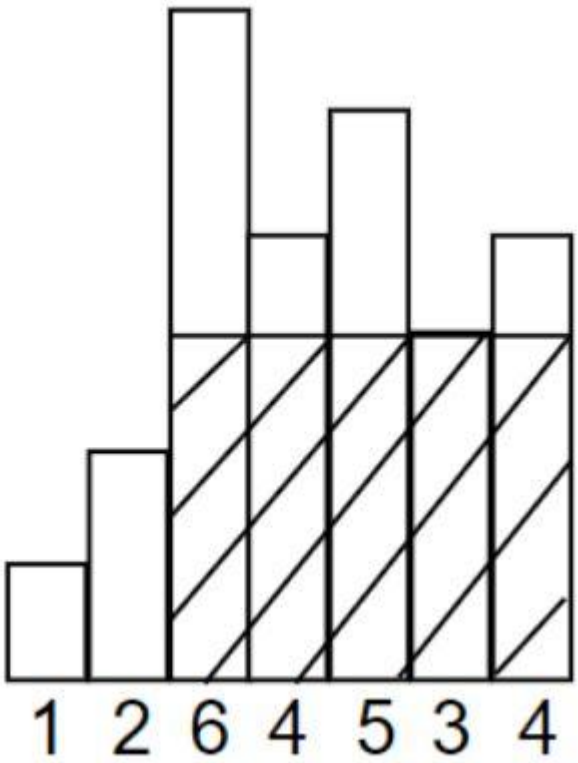
1 2

1 2 6 4 5 3 4

Output

20

Explanation



The volume of summation of gold ingots minus largest cuboid obtained is 20, since the height is 2 and breadth is 1.

---

04 Unlocker

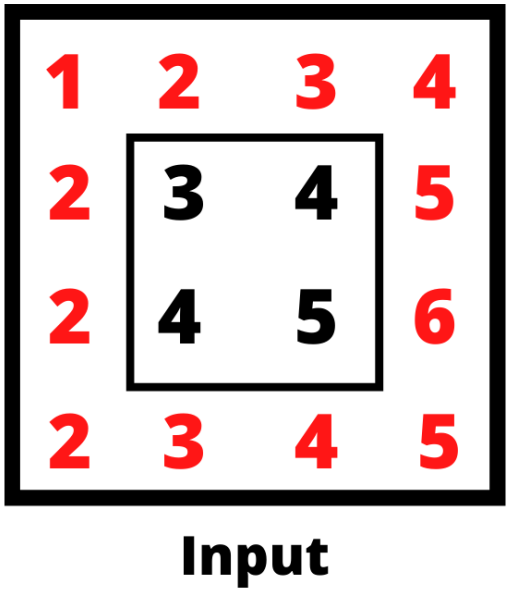
Problem Description

A locker is comprised of one or more layers. Each layer can be rotated only in one direction. Odd numbered layers rotate in anti-clockwise direction (left to right), and even numbered layers rotate in clockwise direction (right to left).

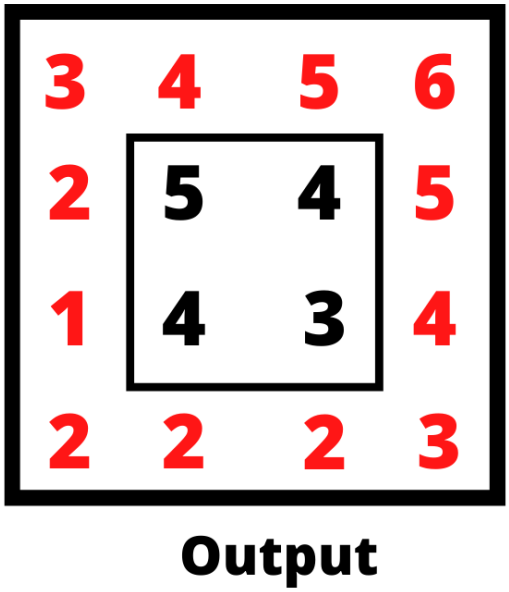
You are given a locker, in the form of a matrix. The matrix will be rectangular in shape. The outer most layer of this matrix is layer1. In context of the diagram below, the numbers painted in red are layer1 and the inner numbers constitute layer2. Bigger matrices will have more layers.

One rotation defined as a given number moving in the neighbouring spot i.e. one spot left for clockwise rotation and one spot right for anti-clockwise rotation.

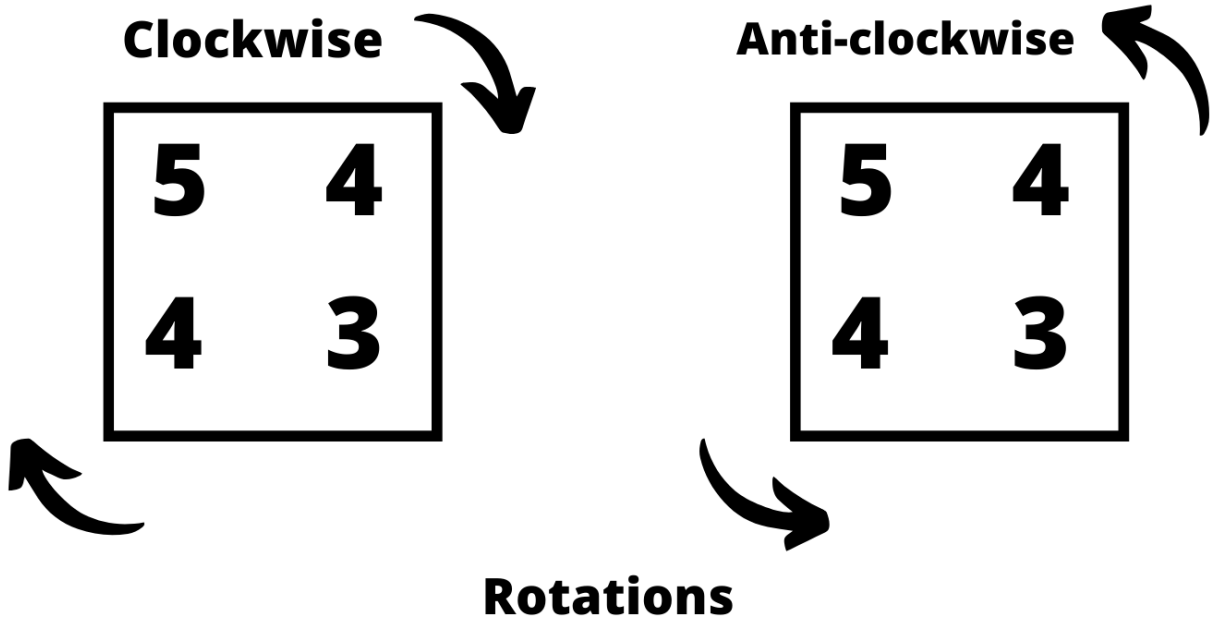
Number of rotations for each layer required to unlock the locker will be provided as input. Print the final unlocked matrix as output.



Input



Output



Constraints

1 < M, N <= 300

0 <= Numbers in matrix < 100

1 <= Number of rotations <= 10^9

M%2=0 && N%2=0

Input

First line contains two space separated integer M and N which denotes the number of rows and number of columns, respectively

Next M lines contain N space separated integers depicting the locked matrix

Last line contains L space separated integers, where L is the number of layers. Each number on this line denotes the number of rotations for every layer from 1 to L

Output

Print unlocked matrix

Time Limit

2

Examples

Example 1

Input

2 2

1 2

3 4

2

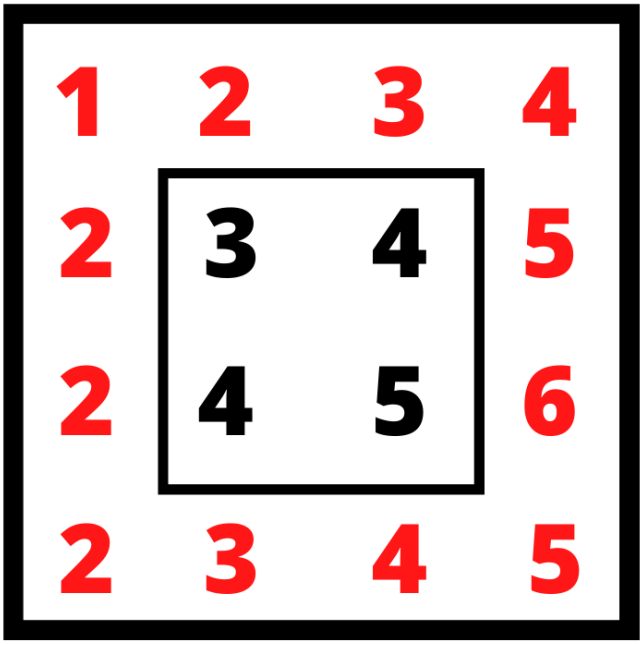
Output

4 3

2 1

Explanation:

There is only one layer. So, we have to rotate it in anti-clockwise direction with 2 rotations.



Example 2

Input

4 4

1 2 3 4

2 3 4 5

2 4 5 6

2 3 4 5

2 2

Output

3 4 5 6

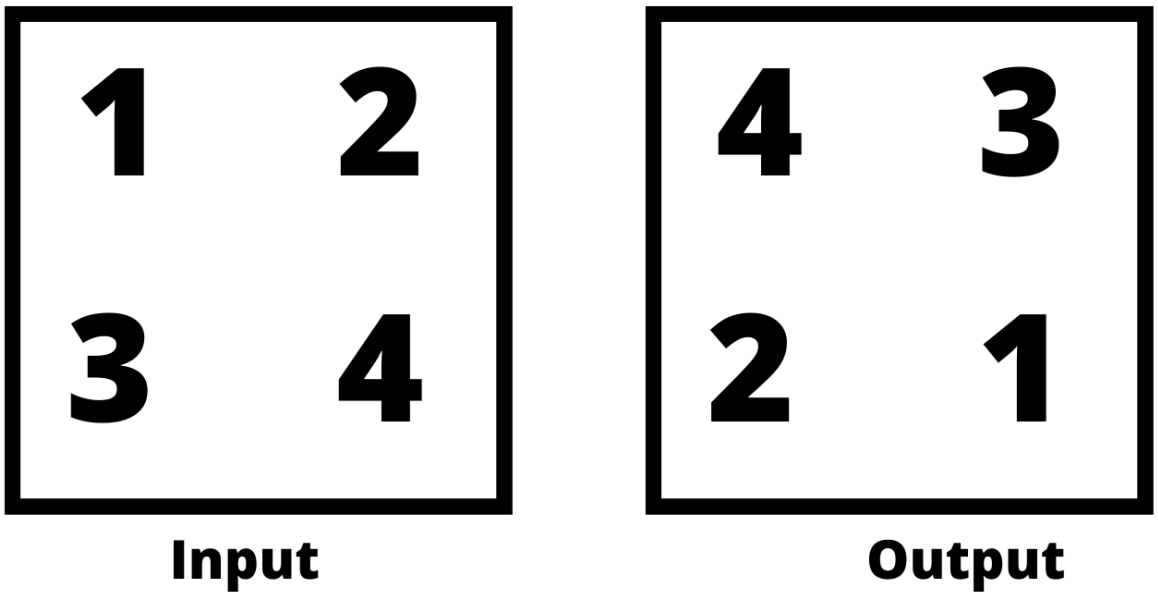
2 5 4 5

1 4 3 4

2 2 2 3

Explanation:

Here we have to rotate layer1 in anti-clockwise direction with 2 rotations, and layer2 clockwise with 2 rotations.



---

05 CODU sudoku

7X7

Problem Description

CODU is solving a 7x7 sudoku. Help him in solving the unique Sudoku.

Rules are as follows

1. There are 7 regions coloured differently. Each region must have a single occurrence of numbers between range [1, 7].

2. Regions don't have a fix shape and it can change from input to input.

3. Each row must have a single occurrence of numbers between range [1, 7] across all input.

4. Each column must have a single occurrence of numbers between range [1, 7] across all input.

Some numbers in some rows, columns and regions will be given. These will be between [1, 7].

Zero (0) denotes that the number is covered. Uncovering it will give a number between [1, 7].

Your task is to fill the numbers [1,7] where there is a 0 such that the 7x7 Sudoku is solved.

7x7 Sudoku is said to be solved when every region, every column, every row has exactly one occurrence of numbers [1,7].

Constraints

7 < Known/Given numbers in Entire Sudoku < 14

Input

Input consists of 14 lines.

First 7 lines denote the positions of numbers [1,7] in respective row and column.

Next 7 lines denote the shape of the regions inside the Sudoku. These will be denoted by 7 unique characters between alphabets [a-z].

Output

Print the solved Sudoku.

7 lines, each line containing 7 space separated integers honoring all the conditions.

Time Limit

1

Examples

Example 1

Input

0 0 0 0 0 6 0

0 0 0 0 0 0 0

2 6 5 1 7 4 3

0 0 0 3 0 0 0

0 0 0 0 0 0 0

0 0 0 0 0 0 0

0 0 0 0 0 0 0

a a a b b b b

a a a a b b c

d d d e e b c

d d d d e e c

f f f h e e c

f f h h e c c

f f h h h h c

The above input can be visualized as follows-

com.tcs.cv.automata.ei.middleware.DocxToHtmlConverter@66f44e8b:image1.png

Output

1 2 4 5 3 6 7

3 5 6 7 1 2 4

2 6 5 1 7 4 3

4 7 1 3 2 5 6

7 1 2 6 4 3 5

5 4 3 2 6 7 1

6 3 7 4 5 1 2

Explanation

There could be many different solutions. Producing any solution as output is acceptable.

Example 2

Input

0 0 0 0 0 0 0

0 0 0 0 4 0 0

3 0 0 6 0 0 0

0 0 0 0 6 0 1

5 0 0 0 0 0 3

0 0 1 0 0 0 2

2 0 0 0 0 0 5

r r r b b b b

g r r r r b o

g g g g b b o

p p g o o o o

p p g d o l l

p p p d l l l

d d d d d l l

The above input can be visualized as follows-

com.tcs.cv.automata.ei.middleware.DocxToHtmlConverter@66f44e8b:image2.png

Note that the shape of the regions in both the inputs are different.

Output

7 1 3 4 5 2 6

1 6 5 2 4 3 7

3 5 2 6 1 7 4

4 2 7 3 6 5 1

5 7 4 1 2 6 3

6 3 1 5 7 4 2

2 4 6 7 3 1 5

Explanation

There could be many different solutions. Producing any solution as output is acceptable.

---

06 3_Palindrome

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 <= the length of input sting <= 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].
