1.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.

2.Count Palindromes

Problem Description

A contest closes in n days hh hours, mm minutes and ss seconds. Given two values of n, how many palindromes of the format nhhmmss would we find in the indicated interval?

A string is said to be palindrome if it reads the same backwards as forwards.

Constraints

n2 - n1 <= 10

Input

One line containing two integer n1 and n2, where n1<n2

Output

One integer representing the number of palindromes in this countdown interval

Time Limit

3

Examples

Example 1

Input

1 2

Output

472

Explanation

We need to check the numbers 1000000 through 2235959 including only numbers where the last 6 digits correspond to times. We find 472 such numbers: 1000001, 1001001, 1002001, 1003001, 1004001, ..., 2231322, 2232322, 2233322, 2234322, 2235322

Example 2

Input

0 2

Output

708

Explanation

There are 708 palindromes: 0000000, 0001000, 0002000, 0003000, 0004000, ..., 2231322, 2232322, 2233322, 2234322, 2235322

3.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

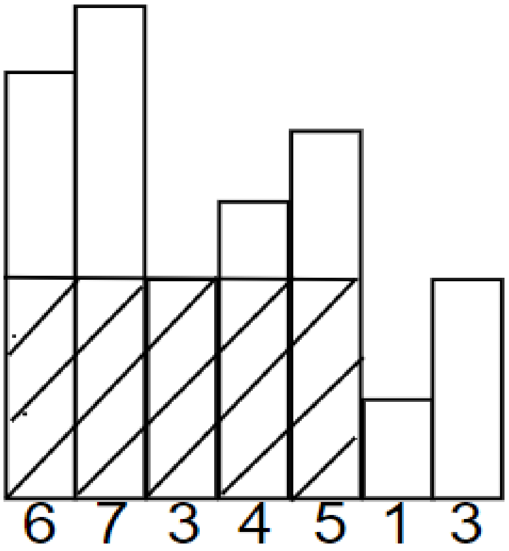
1 1

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

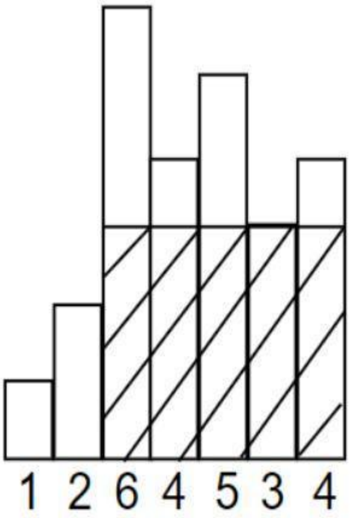
1 2

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.

4.Unlocker

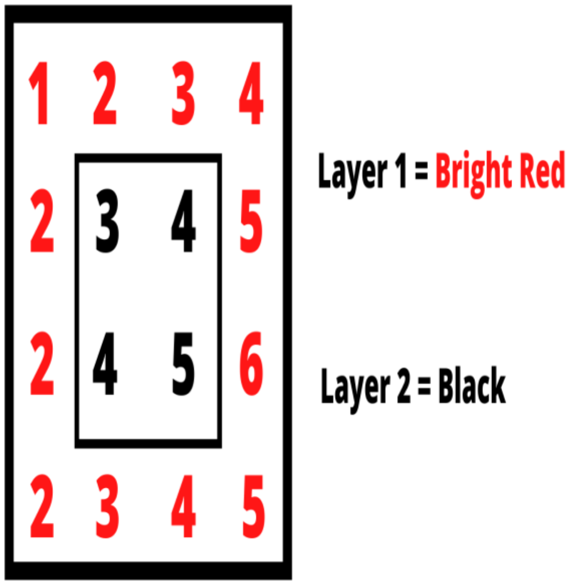
Problem Description

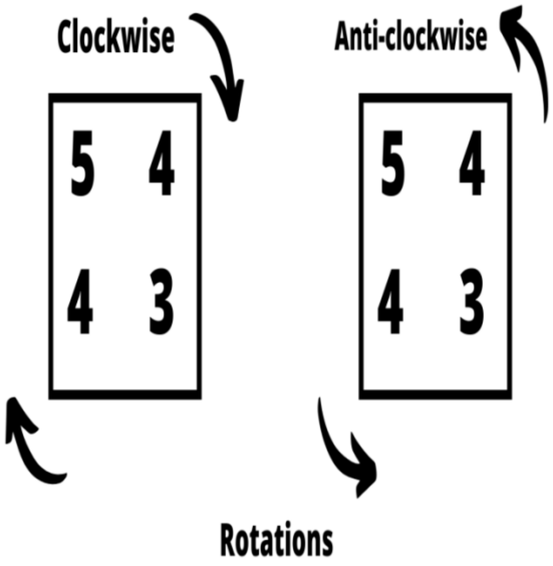
A locker is comprised of one or more layers. Each layer can be rotated only in one direction. Odd numbered layers rotate in anti-clockwise direction (left to right), and even numbered layers rotate in clockwise direction (right to left).

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.





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

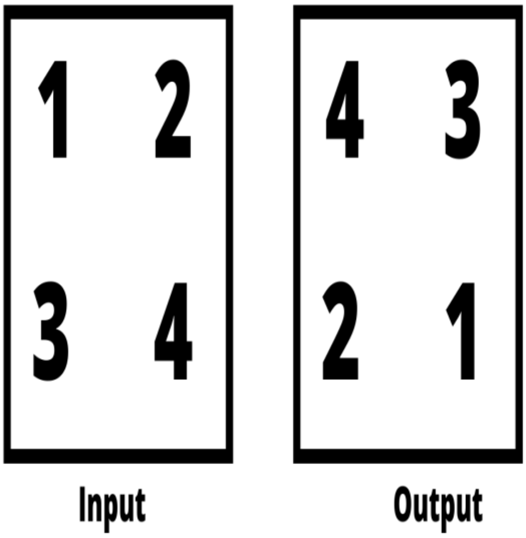
Output

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

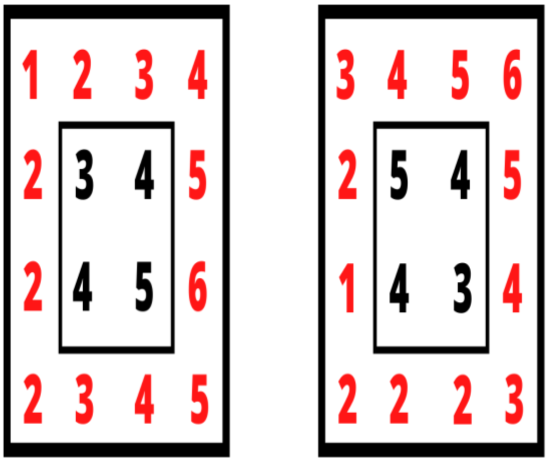
2 5 4 5

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.



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

Input Output

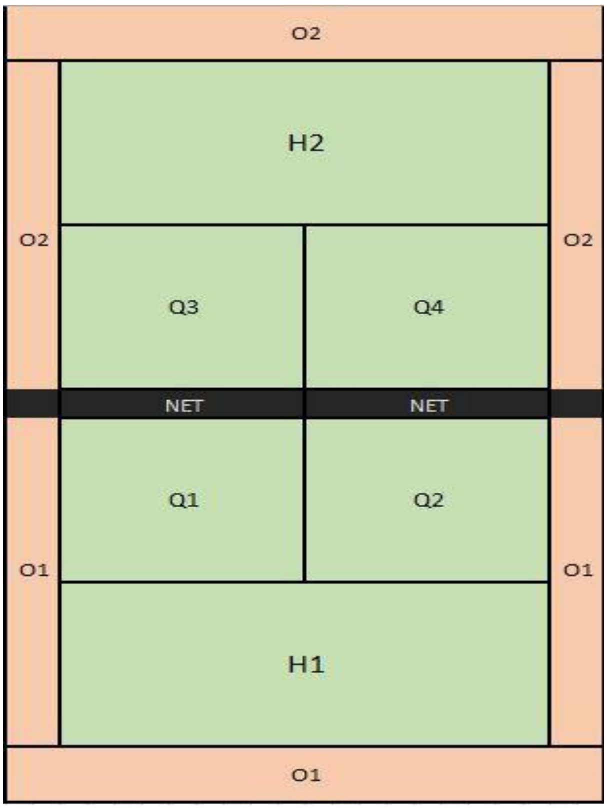
5.Tennis Score

Problem Description

A game of tennis (singles) is played. Regions of the court are named as shown in the image.

The running score of each game is described in the manner: "0", "15", "30", and "40". If at least three points have been scored by each side and a player has one more point than his opponent, the score of the game is advantage ("Advantage") for the player in the lead. (source: Wikipedia)

A set consists of a sequence of games played with service alternating between games, ending when a player wins a set by winning at least six games and at least two games more than the opponent. If one player has won six games and the opponent five, an additional game is played. If the leading player wins that game, the player wins the set 7-5. If the trailing player wins the game (tying the set 6-6) a tie-break is played. A tie-break, allows one player to win one more game and thus the set, to give a final set score of 7-6 or 6-7.(source: Wikipedia)



Aim is to toss the ball on the green part of the other side of the court. A set of strings will be provided signifying where the tennis ball has been tossed on the court. Assume that no player will hit the ball directly (i.e. without the ball being tossed on his (green) side of the court).

For Example, a string Q4 Q1 Q3 O1 will mean that server serves and the ball hits the area Q4. Receiver returns and ball hits ground in area Q1, server hits back again to area Q3 and then the receiver hits 'long' to area O1 and loses a point. So the set output is 0-0 and the current game output will be 15-0.

Following are the rules of the game:

- Game always starts with Server on H1 side (lower side in the image)
- Serve changes after every game is won
- On 'Serve' ball should hit on any 'Q' part of the other side. Hitting on 'H' part will be considered a fault
- While serving, if the server makes a double fault (ball should fall on his side or outside region twice), the server loses one point
- Points scored by the current server are mentioned first, for example if server wins the first point score will be 15-0
- At the end of a set, a changeover happens i.e. players change sides of the court.

- In case of game score 40-40, display "Deuce". In case of Server's Advantage, display "Advantage Server".

In case of Receiver's Advantage, display "Advantage Receiver"

- Number of sets played may not exceed 5

- In case a set is complete a set score of 7-5 will be denoted as:

  7 0 (first player set score)

  5 0 (second player set score)

  Since the second set is about to start, a score of 0 0 is displayed. Please read these scores vertically.

- When a new game is about to start, display 0 - 0 for new game. For example:

  0 0 (current game score)

Departure from Tennis rule:

- In real game of tennis a server is required to serve *cross court*. However, in this problem the server can serve in any court {Q3, Q4} for server 1 and {Q1, Q2} for server 2

- In real game of tennis a tie-break is counted in scores of 1, 2, 3, 4 instead of 15, 30, 40 etc. However, in this problem a tie-break is won according to regular rule i.e. by scoring points like in a regular game. For example, lets say the score is 6-6. In regular tennis a tie break would follow, but in this problem the 13th game will be played and scored using the same rules applied for first 12 games. Whoever wins the 13th game, wins the set i.e. either 7-6 or 6-7

- In a real game of tennis, a *changeover* (players switching court sides) happens at the end of a set or after the first game is played in a new set. However, in this problem changeover happens at the end of the set

Constraints

Number of space separated strings in input < 500

Input

One line containing a string representing the sequence where the ball drops, separated by space

Output

First line containing the set score of all the sets of the Server (starting from first and separated by space)

Second line containing the set score of all the sets of the Receiver (starting from first and separated by space)

Third line containing the game score of the current game (separated by space) or "Deuce" or "Advantage Receiver" or "Advantage Server", as may be the case.

Time Limit

1

Examples

Example 1

Input

```
Q1 Q1
```

Output

```
0
0
0 15
```

Explanation

Server serves and ball hits area Q1. He will not lose point because he is serving. Again he serves and ball hits area Q1. Since, this is a double fault he loses one point. Hence the game score is 0-15.

Example 2

Input

```
Q1 Q1 Q3 Q3 Q1 Q1 Q3 Q3 Q1 Q1 Q3 Q3
```

Output

```
0
0
Deuce
```

Explanation

· String Q1 Q1 indicates a double fault. The score becomes 0-15

· Next String Q3 indicates a valid serve. The second Q3 indicates that the Receiver could not return the serve because it fell into his own half. This makes the score 15-15.

· Similar sequence repeats for two more times, making the score 30-30, then 40-40 which is Deuce.

Example 3

Input

```
Q1 Q1 Q2 Q3 O1 H2 H2 Q1 Q1 Q1 Q2
```

Output

```
1
0
0 0
```

Explanation

· Except points Q2 Q3 O1 all other string indicates a double fault on part of the server

· Q2 Q3 O1 earns the server one point

· Overall the server loses the game

· Now a new game begins and the current server becomes receiver and vice versa

· **The score is printed after the new game begins** and the score is 0 0. Hence, server score is 1 whereas the receiver score is 0

6.7X7

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

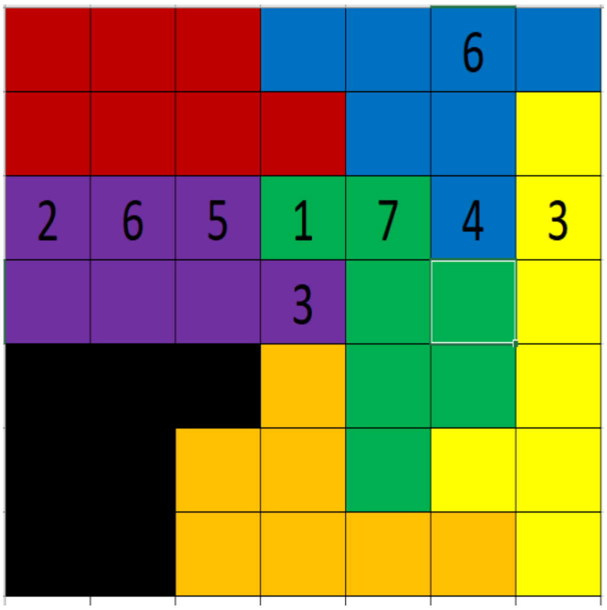
d d d d e e c

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-



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

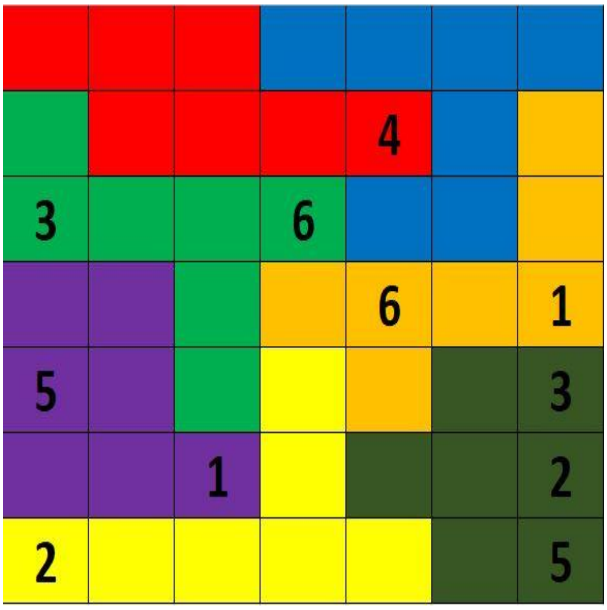
p p g o o o o

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-



				4		
3			6			
				6		1
5						3
		1				2
2						5

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.