TASK	MJEHURIC	DATUM	ROT	SLIKAR	TREZOR	PERIODNI
input	standard input					
output	standard output					
time limit	1 second	1 second	1 second	1 second	3 seconds	5 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
points	**40**	**40**	**70**	**100**	**120**	**130**
	500					

Goran has five wooden pieces arranged in a sequence. There is a number between 1 and 5 inscribed on every piece, so that every number appears on exactly one of the five pieces.

Goran wants to order the pieces to form the sequence 1, 2, 3, 4, 5 and does it like this:

1. If the number on the first piece is greater than the number on the second piece, swap them.
2. If the number on the second piece is greater than the number on the third piece, swap them.
3. If the number on the third piece is greater than the number on the fourth piece, swap them.
4. If the number on the fourth piece is greater than the number on the fifth piece, swap them.
5. If the pieces don't form the sequence 1, 2, 3, 4, 5, go to step 1.

Write a program that, given the initial ordering of the pieces, outputs the ordering after each swap.

INPUT

The first line contains five integers separated by single spaces, the ordering of the pieces.

The numbers will be between 1 and 5 (inclusive) and there will be no duplicates.

The initial ordering will not be 1, 2, 3, 4, 5.

OUTPUT

After any two pieces are swapped, output the ordering of the pieces, on a single line separated by spaces.

EXAMPLES

input
 	
`2 1 5 3 4`
 	
output
 	
`1 2 5 3 4`
`1 2 3 5 4`
`1 2 3 4 5`
	input
	
	`2 3 4 5 1`
	
	output
	
	`2 3 4 1 5`
	`2 3 1 4 5`
	`2 1 3 4 5`
	`1 2 3 4 5`

Write a program that, given a date in 2009, determines the day of week on that date.

INPUT

The first line contains two positive integers D and M separated by a space. The numbers will be a valid date in 2009.

OUTPUT

Output the day of the week on D. M. 2009. The output should be one of the words "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" or "Sunday".

EXAMPLES

input	**input**	**input**
`1 1`	`17 1`	`25 9`
output	**output**	**output**
`Thursday`	`Saturday`	`Friday`

Damir likes to rotate. Right now he is rotating tables of letters. He wrote an R×C table onto a piece of paper. He has also chosen an angle K, a multiple of 45, and wants to rotate his table that many degrees clockwise.

It turns out this task is a bit too hard for Damir, so help him out.

INPUT

The first line contains two integers R and C separated by a space ($1 \leq R \leq 10$, $1 \leq C \leq 10$) the number of rows and columns in Damir's table.

Each of the next R lines contains one row of Damir's table, a string of C lowercase letters.

The last line contains an integer K, a multiple of 45 between 0 and 360 (inclusive).

OUTPUT

Output Damir's table rotated K degrees clockwise, like shown in the examples. The output should contain the smallest number of rows necessary. Some rows may have leading spaces, but no rows may have trailing spaces.

EXAMPLES

input

```
3 5
damir
marko
darko
45
```

output

```
  d
 m a
d a m
 a r i
  r k r
   k o
    o
```

input

```
3 5
damir
marko
darko
90
```

output

```
dmd
aaa
rrm
kki
oor
```

input

```
5 5
abcde
bcdef
cdefg
defgh
efghi
315
```

output

```
    e
   d f
  c e g
 b d f h
a c e g i
 b d f h
  c e g
   d f
    e
```

Josip is a strange painter. He wants to paint a picture consisting of N×N pixels, where N is a power of two (1, 2, 4, 8, 16 etc.). Each pixel will be either black or white. Josip already has an idea of how each pixel will be coloured.

This would be no problem if Josip's painting process wasn't strange. He uses the following recursive process:

- If the picture is a single pixel, he colours it the way he intended.
- Otherwise, split the square into four smaller squares and then:
  1. Select one of the four squares and colour it white.
  2. Select one of the three remaining squares and colour it black.
  3. Consider the two remaining squares as new paintings and use the same three-step process on them.

Soon he noticed that it was not possible to convert all his visions to paintings with this process. Your task is to write a program that will paint a picture that differs as little as possible from the desired picture. The difference between two pictures is the number of pairs of pixels in corresponding positions that differ in colour.

INPUT

The first line contains an integer N (1 ≤ N ≤ 512), the size of the picture Josip would like to paint. N will be a power of 2.

Each of the following N lines contains N digits 0 or 1, white and black squares in the target picture.

OUTPUT

On the first line, output the smallest possible difference that can be achieved.

On the next N lines, output a picture that can be painted with Josip's process and achieves the smallest difference. The picture should be in the same format as in the input.

Note: The second part of the output (the picture) may not be unique. Any correct output will be accepted.

SCORING

In test cases worth 50% points, N will be at most 8.

EXAMPLES

input 4 0001 0001 0011 1110 **output** 1 0001 0001 0011 1111	**input** 4 1111 1111 1111 1111 **output** 6 0011 0011 0111 1101	**input** 8 01010001 10100011 01010111 10101111 01010111 10100011 01010001 10100000 **output** 16 00000001 00000011 00000111 00001111 11110111 11110011 11110001 11110000

Mirko decided to open a new business – bank vaults. A branch of the bank can be visualized in a plane, vaults being points in the plane. Mirko's branch contains exactly L·(A+1+B) vaults, so that each point with integer coordinates inside the rectangle with corners (1, −A) and (L, B) contains one vault.

The vaults are watched by two guards – one at (0, −A), the other at (0, B). A guard can **see** a vault if **there are no other vaults** on the line segment connecting them.

A vault is not secure if **neither** guard can see it, secure if **only one** guard can see it and super-secure if **both** guards can see it.

Given A, B and L, output the number of insecure, secure and super-secure vaults.

INPUT

The first line contains integers A and B separated by a space ($1 \le A \le 2000$, $1 \le B \le 2000$).

The second line contains the integer L ($1 \le L \le 1000000000$).

OUTPUT

Output on three separate lines the numbers of insecure, secure and super-secure vaults.

SCORING

In test cases worth 50% of points, L will be at most 1000.

In test worth another 25% of points, A and B will be at most 100 (but L can be as large as one billion).

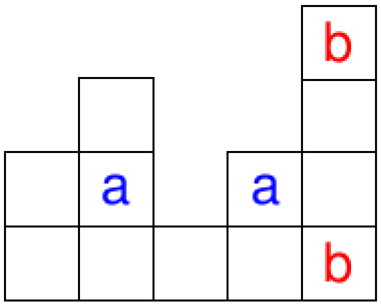
EXAMPLES

input	input	input
1 1 3	2 3 4	7 11 1000000
output	**output**	**output**
2 2 5	0 16 8	6723409 2301730 9974861

Luka is bored in chemistry class so he is staring at a large periodic table of chemical elements hanging from a wall above the blackboard. To kill time, Luka decided to make his own table completely different from the one in the classroom.

His table consists of N columns, each with some height, aligned at the bottom (see example below). After he draws the table he needs to fill it with elements. He first decided to enter the noble gases of which there are K. Luka must put them in the table so that no two noble gases are **close** to each other.

Two squares in the table are close to each other if they are in the same column or row, and all squares between them exist. In the example below, the 'a' squares are not close, but the 'b' squares are.



Write a program that, given N, K and the heights of the N columns, calculates the total number of ways for Luka to place the noble gases into the table. This number can be large, so output it modulo 1000000007.

INPUT

The first line contains the integers N and K separated by a space ($1 \le N \le 500$, $1 \le K \le 500$), the number of columns in Luka's table and the number of noble gases.

The next line contains N positive integers, separated by spaces. These are heights of the columns from left to right. The heights will be at most 1000000.

OUTPUT

Output the number of ways for Luka to fill his table with noble gases, modulo 1000000007.

SCORING

In test cases worth 40% of points, all numbers in the input will be less than 15.

In test cases worth 70% of points, all numbers in the input will be less than 100.

EXAMPLES

input 3 3 2 1 3 **output** 2	**input** 4 1 1 2 3 4 **output** 10	**output** 5 2 2 3 1 2 4 **output** 43	**input** 3 2 999999 999999 999999 **output** 990979013