A. Little Elephant and Chess

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Little Elephant loves chess very much.

One day the Little Elephant and his friend decided to play chess. They've got the chess pieces but the board is a problem. They've got an 8×8 checkered board, each square is painted either black or white. The Little Elephant and his friend know that a *proper chessboard* doesn't have any side-adjacent cells with the same color and the upper left cell is white. To play chess, they want to make the board they have a proper chessboard. For that the friends can choose any row of the board and cyclically shift the cells of the chosen row, that is, put the last (rightmost) square on the first place in the row and shift the others one position to the right. You can run the described operation **multiple times** (or not run it at all).

For example, if the first line of the board looks like that "BBBBBBWW" (the white cells of the line are marked with character "W", the black cells are marked with character "B"), then after one cyclic shift it will look like that "WBBBBBBW".

Help the Little Elephant and his friend to find out whether they can use any number of the described operations to turn the board they have into a proper chessboard.

Input

The input consists of exactly eight lines. Each line contains exactly eight characters "W" or "B" without any spaces: the j-th character in the i-th line stands for the color of the j-th cell of the i-th row of the elephants' board. Character "W" stands for the white color, character "B" stands for the black color.

Consider the rows of the board numbered from 1 to 8 from top to bottom, and the columns — from 1 to 8 from left to right. The given board can initially be a proper chessboard.

Output

In a single line print "YES" (without the quotes), if we can make the board a proper chessboard and "NO" (without the quotes) otherwise.

Examples

input

```
WBWBWBWB
BWBWBWBW
BWBWBWBW
BWBWBWBW
WBWBWBWB
WBWBWBWB
BWBWBWBW
WBWBWBWB
```

output

```
YES
```

input

```
WBWBWBWB
WBWBWBWB
BBWBWWWB
BWBWBWBW
BWBWBWBW
BWBWBWWW
BWBWBWBW
BWBWBWBW
```

output

```
NO
```

Note

In the first sample you should shift the following lines one position to the right: the 3-rd, the 6-th, the 7-th and the 8-th.

In the second sample there is no way you can achieve the goal.

B. Little Elephant and Magic Square

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Elephant loves magic squares very much.

A *magic square* is a 3×3 table, each cell contains some positive integer. At that the sums of integers in all rows, columns and diagonals of the table are equal. The figure below shows the magic square, the sum of integers in all its rows, columns and diagonals equals 15.

The Little Elephant remembered one magic square. He started writing this square on a piece of paper, but as he wrote, he forgot all three elements of the main diagonal of the magic square. Fortunately, the Little Elephant clearly remembered that all elements of the magic square did not exceed 10^5.

Help the Little Elephant, restore the original magic square, given the Elephant's notes.

Input

The first three lines of the input contain the Little Elephant's notes. The first line contains elements of the first row of the magic square. The second line contains the elements of the second row, the third line is for the third row. The main diagonal elements that have been forgotten by the Elephant are represented by zeroes.

It is guaranteed that the notes contain exactly three zeroes and they are all located on the main diagonal. It is guaranteed that all positive numbers in the table do not exceed 10^5.

Output

Print three lines, in each line print three integers — the Little Elephant's magic square. If there are multiple magic squares, you are allowed to print any of them. Note that all numbers you print must be positive and not exceed 10^5.

It is guaranteed that there exists at least one magic square that meets the conditions.

Examples

input

```
0 1 1
1 0 1
1 1 0
```

output

```
1 1 1
1 1 1
1 1 1
```

input

```
0 3 6
5 0 5
4 7 0
```

output

```
6 3 6
5 5 5
4 7 4
```

C. Little Elephant and Bits

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Little Elephant has an integer a, written in the binary notation. He wants to write this number on a piece of paper.

To make sure that the number a fits on the piece of paper, the Little Elephant **ought** to delete exactly one any digit from number a in the binary record. At that a new number appears. It consists of the remaining binary digits, written in the corresponding order (possible, with leading zeroes).

The Little Elephant wants the number he is going to write on the paper to be as large as possible. Help him find the maximum number that he can obtain after deleting exactly one binary digit and print it in the binary notation.

Input

The single line contains integer a, written in the binary notation without leading zeroes. This number contains more than 1 and at most 10^5 digits.

Output

In the single line print the number that is written without leading zeroes in the binary notation — the answer to the problem.

Examples

input

```
101
```

output

```
11
```

input

```
110010
```

output

```
11010
```

Note

In the first sample the best strategy is to delete the second digit. That results in number $11_2 = 3_{10}$.

In the second sample the best strategy is to delete the third or fourth digits — that results in number $11010_2 = 26_{10}$.

D. Little Elephant and Elections

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There have recently been elections in the zoo. Overall there were 7 main political parties: one of them is the Little Elephant Political Party, 6 other parties have less catchy names.

Political parties find their number in the ballot highly important. Overall there are m possible numbers: $1, 2, \ldots, m$. Each of these 7 parties is going to be assigned in some way to exactly one number, at that, two distinct parties cannot receive the same number.

The Little Elephant Political Party members believe in the lucky digits 4 and 7. They want to evaluate their chances in the elections. For that, they need to find out, how many correct assignments are there, such that the number of lucky digits in the Little Elephant Political Party ballot number is strictly larger than the total number of lucky digits in the ballot numbers of 6 other parties.

Help the Little Elephant Political Party, calculate this number. As the answer can be rather large, print the remainder from dividing it by 1000000007 $(10^9 + 7)$.

Input

A single line contains a single positive integer m $(7 \le m \le 10^9)$ — the number of possible numbers in the ballot.

Output

In a single line print a single integer — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
7
```

output

```
0
```

input

```
8
```

output

```
1440
```

E. Little Elephant and LCM

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Little Elephant loves the LCM (least common multiple) operation of a non-empty set of positive integers. The result of the LCM operation of k positive integers $x_1, x_2, ..., x_k$ is the minimum positive integer that is divisible by each of numbers x_i.

Let's assume that there is a sequence of integers $b_1, b_2, ..., b_n$. Let's denote their LCMs as $lcm(b_1, b_2, ..., b_n)$ and the maximum of them as $max(b_1, b_2, ..., b_n)$. The Little Elephant considers a sequence b *good*, if $lcm(b_1, b_2, ..., b_n) = max(b_1, b_2, ..., b_n)$.

The Little Elephant has a sequence of integers $a_1, a_2, ..., a_n$. Help him find the number of good sequences of integers $b_1, b_2, ..., b_n$, such that for all i $(1 \le i \le n)$ the following condition fulfills: $1 \le b_i \le a_i$. As the answer can be rather large, print the remainder from dividing it by 1000000007 $(10^9 + 7)$.

Input

The first line contains a single positive integer n $(1 \le n \le 10^5)$ — the number of integers in the sequence a. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^5)$ — sequence a.

Output

In the single line print a single integer — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
4
1 4 3 2
```

output

```
15
```

input

```
2
6 3
```

output

```
13
```