G. Puzzling Language

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this problem you will write a simple code generator for a 2D programming language derived from Brainfuck.

The code in this language is a rectangular grid of characters '.' and 'X'. The code is converted to a Brainfuck program as follows: the characters are read in the usual order (top to bottom, left to right), and each 'X' character is converted a Brainfuck instruction to be executed. The instruction is defined by the left, top and right neighbors of the 'X' character using the following conversion table:

You are given a string. Output a program in the described language which prints this string.

You can download the language interpreter used for judging here: https://assets.codeforces.com/rounds/952/puzzling-interpreter.cpp (use C++11 to compile the code). Note several implementation details:

- The first step of the language interpretation is conversion to a Brainfuck program, which is then executed.
- The code must be rectangular, with all lines of the same length. It can have at most 10,000 lines and 10,000 columns, and can have at most 500,000 'X' characters.
- The code has toroidal topology, i.e. the 'X' on the first line will have top neighbor in the last line.
- Brainfuck interpreter has 30000 memory cells which store integers from 0 to 255 with increment/decrement done modulo 256.
- Console input (`,` command) is allowed in Brainfuck code but has no effect when executed.

Input

The input consists of a single string of characters with ASCII codes between 33 ('!') and 122 ('z'), inclusive. The length of the string is between 1 and 10 characters, inclusive.

Output

Output a program in the described language which, when executed, will print the given message.

Example

input

```
$$$
```

output

```
.......X.......
......XXX......
.....XXXXX.....
....XXXXXXX....
...XXXXXXXXX...
..XXXXXXXXXXX..
.XXXXXXXXXXXXX.
...............
X.............X
X..............
X..............
X..............
```

Note

The example corresponds to the following Brainfuck program:

```
       -
      >+<
     >+++<
```

```
    >+++++<
   >+++++++<
  >+++++++++<
 >+++++++++++<

<             >
.
.
.
```

The triangular block decrements the first memory cell and sets the value of the second memory cell to 36 - the ASCII code of '$' character. The next line after the triangular block moves the memory pointer to the second memory cell, and the next three lines print the '$' character three times.