**Programming - Python for Beginners - Pattern Problems - Day 4**

**Source:Codevita**

Constellation

Three characters { #, \*, . } represents a constellation of stars and galaxies in space. Each galaxy is demarcated by # characters. There can be one or many stars in a given galaxy. Stars can only be in the shape of vowels { A, E, I, O, U }. A collection of \* in the shape of the vowels is a star. A star is contained in a 3x3 block. Stars cannot be overlapping. The dot(.) character denotes empty space.

Given 3xN matrix comprising of { #, \*, . } character, find the galaxy and stars within them.

Note: Please pay attention to how vowel A is denoted in a 3x3 block in the examples section below.

Constraints

3 <= N <= 10^5

Input

Input consists of a single integer N denoting the number of columns.

Output

The output contains vowels (stars) in order of their occurrence within the given galaxy. The galaxy itself is represented by the # character.

Example 1

Input

18

\* . \* # \* \* \* # \* \* \* # \* \* \* . \* .

\* . \* # \* . \* # . \* . # \* \* \* \* \* \*

\* \* \* # \* \* \* # \* \* \* # \* \* \* \* . \*

Output

U#O#I#EA

Explanation

As it can be seen that the stars make the image of the alphabets U, O, I, E, and A respectively.

Example 2

Input

12

\* . \* # . \* \* \* # . \* .

\* . \* # . . \* . # \* \* \*

\* \* \* # . \* \* \* # \* . \*

Output

U#I#A

Explanation

As it can be seen that the stars make the image of the alphabet U, I, and A.

# Function to find the vowel hidden

def FindVowel(InpMat,Column):

# Parse the matrix column-wise

col = 0

while(True):

if (col>=Column): break

# If all '#' is encountered in the given column, print '#'

if (InpMat[0][col]=='#'

and InpMat[1][col]=='#'

and InpMat[2][col]=='#'):

print('#', end ='')

# If all '.' is encountered skip the column

elif (InpMat[0][col] == '.'

and InpMat[1][col] == '.'

and InpMat[2][col] == '.') :

pass

# A combination of # & .

else:

i = col;

# Get the 3\*3 matrix to find the vowel

Row11 = InpMat[0][i];

Row12 = InpMat[0][i + 1];

Row13 = InpMat[0][i + 2];

Row21 = InpMat[1][i];

Row22 = InpMat[1][i + 1];

Row23 = InpMat[1][i + 2];

Row31 = InpMat[2][i];

Row32 = InpMat[2][i + 1];

Row33 = InpMat[2][i + 2];

# Check if the arrangement forms character 'A'

if (Row11 == '.' and Row12 == '\*'

and Row13 == '.' and Row21 == '\*'

and Row22 == '\*' and Row23 == '\*'

and Row31 == '\*' and Row32 == '.'

and Row33 == '\*'):

print("A", end ='')

# Increment column number

col = col + 2

# Check if the arrangement forms character 'E'

if (Row11 == '\*' and Row12 == '\*'

and Row13 == '\*' and Row21 == '\*'

and Row22 == '\*' and Row23 == '\*'

and Row31 == '\*' and Row32 == '\*'

and Row33 == '\*'):

print("E", end ='')

# Increment column number

col = col + 2

# Check if the arrangement forms character 'I'

if (Row11 == '\*' and Row12 == '\*'

and Row13 == '\*' and Row21 == '.'

and Row22 == '\*' and Row23 == '.'

and Row31 == '\*' and Row32 == '\*'

and Row33 == '\*'):

print ("I", end ='')

# Increment column number

col = col + 2

# Check if the arrangement forms 'O'

if (Row11 == '\*' and Row12 == '\*'

and Row13 == '\*' and Row21 == '\*'

and Row22 == '.' and Row23 == '\*'

and Row31 == '\*' and Row32 == '\*'

and Row33 == '\*'):

print ("O", end ='')

# Increment column number

col = col + 2

# Check if the arrangement forms 'U'

if (Row11 == '\*' and Row12 == '.'

and Row13 == '\*' and Row21 == '\*'

and Row22 == '.' and Row23 == '\*'

and Row31 == '\*' and Row32 == '\*'

and Row33 == '\*'):

print ("U", end ='')

# Increment column number

col = col + 2

col+=1

TheInp =[[ '\*', '.', '\*', '#', '\*', '\*', '\*', '#', '\*',

'\*', '\*', '#', '\*', '\*', '\*', '.', '\*', '.' ],

[ '\*', '.', '\*', '#', '\*', '.', '\*', '#', '.',

'\*', '.', '#', '\*', '\*', '\*', '\*', '\*', '\*' ],

[ '\*', '\*', '\*', '#', '\*', '\*', '\*', '#', '\*',

'\*', '\*', '#', '\*', '\*', '\*', '\*', '.',

'\*' ] ];

FindVowel(TheInp, 18)