## **Qube QRT**

Codingame - 2022 Sep04 (4 questions in 30 minutes)

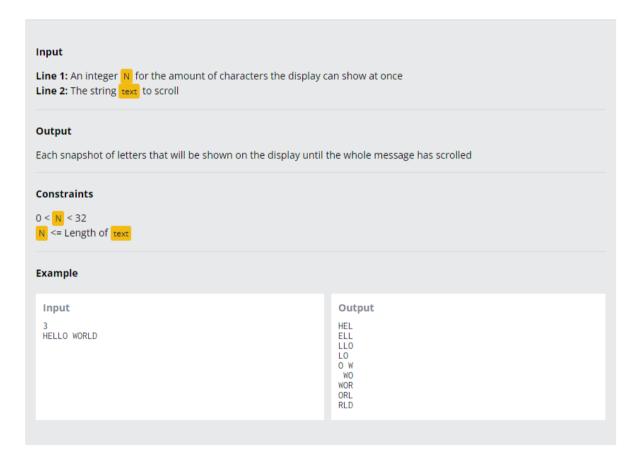


Elapsed time: 00:07/12:00 ^

You can change the programming language at the top right of the screen.

### ⊚ Goal

You have just recently purchased a LED display that can show up to N characters at once. Your goal is to scroll a message across the display, moving from the right to the left, one character at a time.



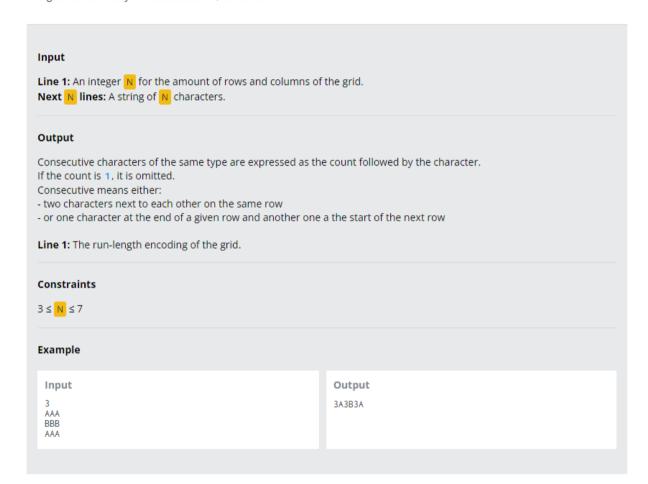
## Question 2 / 4 - Programming

Elapsed time: 00:03/18:00 ^

You can change the programming language at the top right of the screen.



The goal of your program is to compress a N by N grid into one single line using **run-length encoding**. The grid contains only the characters "A", "B" or "C".



# Question 3 / 4 - Programming

Elapsed time: 00:02/10:00 ^

You can change the programming language at the top right of the screen.

### ⊚ Goal

Clap@7 is a game played by a group of people, whereby each person will call out numbers, in ascending order, starting from 1. However, if the number satisfies any of the following conditions:

- > The number is divisible by 7
- > The number has the digit 7 in it
- > The sum of the digits of the number is divisible by 7

Then the person has to clap instead of calling out the number. If the person does not perform the correct action, he loses.

Given an integer N, determine how many claps there have been, if the game has terminated at N, including N.

An example is shown, where three players A, B and C are playing this game.

- A: 1
- B: 2
- C: 3
- A: 4 B: 5
- C: 6
- A: CLAP
- B: 8
- C: 9
- A: 10
- B: 11
- C: 12
- A: 13
- B: CLAP C: 15
- A: CLAP
- B: CLAP C: 18

# Input

**Line 1:** An integer N which is the number at which the game has ended.

#### Output

Print the number of **CLAPS** 



( Elapsed time: 00:03/20:00 ^

You can change the programming language at the top right of the screen.

#### The program:

Given the coordinates of the points 1 and 2, fill the grid of the given size in order to demonstrate which of the two points is the closest to each cell. We will use the Manhattan distance: d(A, B) = abs(Ax - Bx) + abs(Ay - By).

Print a grid that contains, for each cell, the following character:

- 'X' if one of the two given points is in the cell
- '0' if the cell is equidistant to the two points
- '1' if the closest point to the cell is point 1
- '2' if the closest point to the cell is point 2

```
INPUT:
Line 1 : two space separated integers giving the width and the height of the grid.
Line 2: two space separated integers giving the coordinates of point 1.
Line 3: two space separated integers giving the coordinates of point 2.
OUTPUT:
A grid of height rows and width columns.
CONSTRAINTS:
1 < width , height ≤ 200
0 \le x1 , y1 , x2 , y2 < 200
EXAMPLE:
                               Output
1111111022
  Input
10 10
                               1111111022
                                1111111022
                                1111111022
                                1111111022
                                1111X11022
                               1111110222
                                11111022X2
                                1111102222
                                1111102222
```