MENTORMATE™ DevCamp

Assignment 1: Video Recording

Your first assignment is to record yourself answering the questions below **in English**. The video should be up to **5 minutes long**.

Questions:

1. Present yourself shortly to us.
2. Why do you want to become part of DevCamp?
3. Which technology is your favorite and why?

Here are some **tips** on how to record your video:

- Shoot yourself in landscape (horizontal) for higher quality. If audio pickup isn't great, use
  headphones.
- Shoot yourself in good lighting, natural light is your best friend. Do not shoot yourself in direct sunlight. Facing a window could be helpful.
- The shooting angle shouldn’t be much below or above your head. Just imagine you’re talking to someone.
- Don’t talk too fast and try to pause between thoughts.
- Shorter is better than longer.

Upload the video to any Cloud (file transfer/sharing) platform - Youtube, Google Drive, WeTransfer, pCloud, Dropbox, One Drive, etc. Make sure that the link is shared externally, so we can open it. Please insert the link in the relevant field in the application form.

Assignment 2: Brickwork

Here is the second part of your assignment. You should build your solution with one of the following programming languages: JavaScript (you can use any framework or library), .NET, Java, Ruby, Python, PHP, Go, Swift, Objective C. Please upload the code for this project to GitHub and insert the link to your repository in the relevant field in the application form.

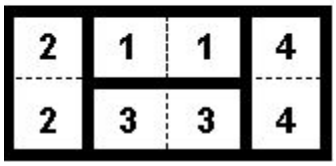
Brickwork Assignment:

The builders must cover a rectangular area of size *M* × *N* (*M* and *N* are even numbers) with two layers of bricks that are rectangles of size 1 × 2. The first layer of the bricks has already been completed. The second layer (in an effort to make the brickwork really strong) must be created in a way that no brick in it lies exactly on a brick from the first layer. However, it is allowed **half** of the same brick to lie on the same brick on the second layer.

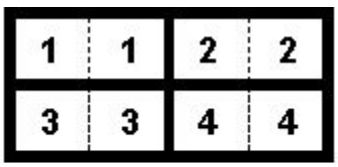
Create a console app that accepts input parameters for the given layout of the bricks for the first layer, determine the possible layout of the **second** one, or prove that it is impossible to create the **second** layer and print it in the console.

Example. The two pictures show the layout of the two layers, respectively. The size of the area is 2 × 4. Each brick is marked with its number on both halves.

Layer 2 (output)



Layer 1 (input)



Input

1. *N*, *M* — dimensions of the area (both layers' dimension a.k.a wall thickness/width and length).
2. Then, add a single value separated by a space for each line *N* and following column *M* describing the bricks layout in the first layer.

   NOTE: Each brick is marked with two equal numbers written in the squares of the area that are covered by this brick. All bricks are marked with whole numbers ranging from 1 to the total number of the bricks. *M* and *N* are even numbers not exceeding 100.

Output

Write *N* lines with *M* numbers each that describe the layout of the second layer in the way shown above

Assessment

1. If the solution exists, write *N* lines with *M* numbers each that describe the layout of the second layer in the way shown above.
2. Print output `-1` with a message that no solution exists.
3. Validations - N and M should define a valid area of less than 100 lines/ columns. Validate input has exactly that number of rows and columns. Validate there are no bricks spanning 3 rows/ columns.
4. Add comments on each class, method, and instantiated variable.
5. Surround each brick of the second layer with asterisk and/ or dash symbols - `*` and/ or `-`. There should be a single line of symbols between two bricks.

Sample

input	output
2 4 1 1 2 2 3 3 4 4	2 1 1 4 2 3 3 4

Example 2

Layer 2 (output)

2 1 1 4 5 5 6 6

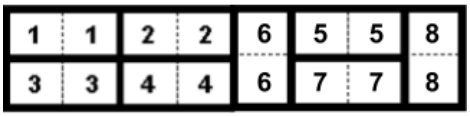
2 3 3 4 7 7 8 8

Layer 1 (input)

2 8

1 1 2 2 6 5 5 8

3 3 4 4 6 7 7 8



Example 3

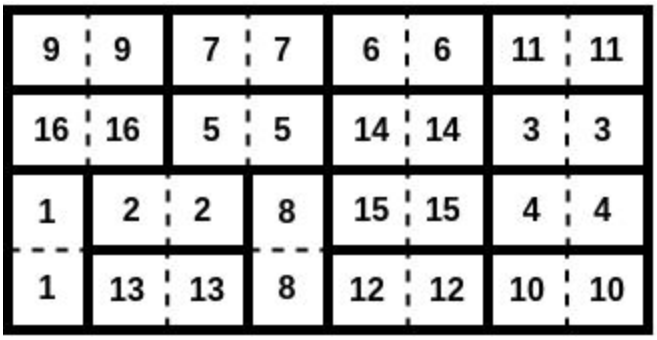
Layer 2 (output)

9 9 7 7 6 6 11 11

16 16 5 5 14 14 3 3

1 2 2 8 15 15 4 4

1 13 13 8 12 12 10 10



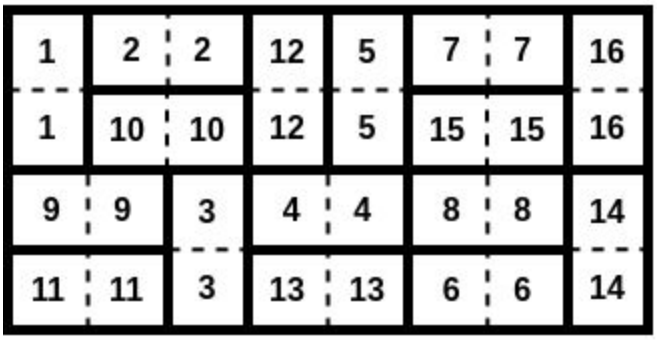
Layer 1 (input)
4 8

1 2 2 12 5 7 7 16

1 10 10 12 5 15 15 16

9 9 3 4 4 8 8 14

11 11 3 13 13 6 6 14



Error examples