cogito.

Home Coding Test

General Notes:

- You have a **total of 1.5 hours** to complete the two questions, from the time you received this mail. When time is up, please **send your code** by mail (reply to this mail).
- Please also send additional code/resources if any (code you used for testing, your IDE's “project” files)
- Your solution will be evaluated in terms of **correctness, cleanliness, and readability**.
- You can also score **“bonus points” for coding speed**, if you send your solution before the deadline. However, correctness/cleanliness/readability are **more important than speed**.
- You may use any programming language of your choice.
- Time management: it is better to produce two imperfect solutions than a perfect solution for one of the questions and a poor solution for the other. Please manage your time accordingly.

[SEE THE NEXT PAGES FOR THE QUESTIONS]

QUESTION 1 (of 2): Connect-Four

Write a function that receives a "connect-four" board as an input. The board is represented by a matrix of integers, where each cell holds one of the following values:

- 0 => empty
- 1 => player 1's piece
- 2 => player 2's piece

The function should return the winner (if any): 0 (no winner yet), 1, or 2.

Notes:

- No input validation tests are necessary. You can assume the board given to you is valid (e.g. no "floating pieces").
- You don't need to implement the "moves", where the next piece goes, etc.
- Your solution doesn't have to be efficient.

[QUESTION 2 IS ON THE NEXT PAGE]

QUESTION 2 (of 2): Forbidden Integers

Given a range of integers, we define a list of forbidden integers within that range. The forbidden integers are given as a list of inclusive ranges in a file, such that each range is in a new line.

For example, given a range of 0-9 and a forbidden integers list of [5-8, 0-2, 4-7], the only integers allowed are 3 and 9.

Assuming a range of 0-4294967295 (all 32-bit unsigned integers) and using the forbidden numbers list attached, write a program that outputs the smallest allowed integer and the total number of allowed integers (add your result to the code as a comment).

Notes:

- If *n* is the number of forbidden ranges and *u* is the length of the total range, your solution should be more efficient than $O(n^2)$ (it should not be a function of *u*).