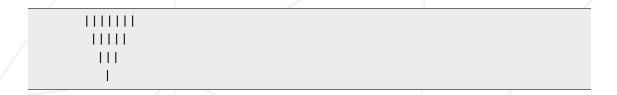# Chapter II

Subject

II.1 What you need to do

- Alum1 is a game based on matches.

- There is a fixed number of rows of matches.

- Players take turns.

- Each player can take, on a single row, between 1 and 3 matches.

- The player who takes the last match, on the last available row, loses.

- For example, this is a board :

```
   | | | | | | |
    | | | | |
     | | |
      |
```

- You are going to code a program against which a user can play.

- This program will take a text file as input. That text file will describe the starting board.

- If no argument is given to the program, you'll read the board on standard input, with an empty line signaling the end of input.

- The board will be formatted according to the following rules :

  - Each line of the text file will indicate the number of matches present on that line, followed by a newline.

  - This number must be included between 1 and 10000.

    ◦ If the board description is incorrect, you'll write `ERROR` on standard error, followed by a newline, and exit cleanly.

- Once the game is loaded, you will display the board between each turn (whether it is the player playing or the computer).

- When it is the player's turn, you will ask them the number of matches they want to remove on the last line of the board.

- If the answer provided is incorrect (the move is invalid, or can't be played at that turn), you'll ask again !

- Your IA must try to win.

- At the end of the game, the winner is announced.

II.2   Turned-it project

- You must have, at the root of your repository, an `author` file containing your 2 logins on separate lines, followed by a newline:

```
$>cat -e auteur
xlogin$
ylogin$
$>
```

- Your executable will be called `alum1`.

- You must supply a `Makefile` with the usual rules.

- Only what is present on your Git repository will be evaluated during the defense.

- You can only use the following functions :

    ○ read

    ○ write

    ○ open

    ○ close

    ○ malloc

    ○ free