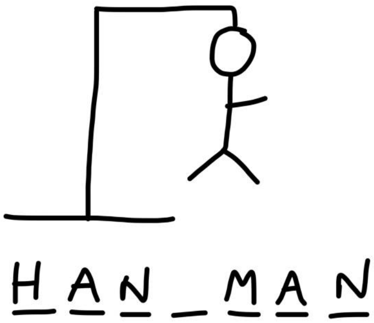
Lab 2 : Hangman



In this assignment, you will recreate the classic "Hangman" game.

Rules of the game

- The computer picks a random word, and displays it with all letters replaced with `_`.
- At each turn:
  - The player is asked for a letter.
  - If the letter was already tried before, ask for another letter
  - Otherwise, display the "hint". For each letter in the word:
    - If the letter was tried by the player, display it (uppercase)
    - Otherwise, display `_`
  - Display the number of turns remaining
- The game stops when:
  - all the letters are revealed
  - OR
  - the maximum number of tries has been reached.

Python implementation

- The maximum number of tries is defined as a global constant.
- The program opens the `words.txt` file, and picks a word from it in the `pick_random_word()` function.
- The `show_letters_in_word` takes a word and a list of letters, and **returns a string** displaying the letters present in the word.
- The `all_letters_found` function takes a word and a list of letters. It returns `True` if all letters in the word are in the list `letters`.
- Finally, the `main()` function is used to run the game
- Read the code provided (and the docstrings!)

Your code must respect the structure above. Your submissions are automatically graded with a script, based on the rules above. Look at the code provided!

Submission

- Make sure the tests pass (using pytest).
- Submit the `hangman.py` file to D2L. **DO NOT USE ZIPPED FILES!**

Grading

Submit before the end of class

Item	Grade
The game works according to the specifications, and the tests pass	100%

Note: a demo during class is required.

Submit after class, before Sunday

Item	Grade
The game works according to the specifications, and the tests pass	70%
The code is clean, documented and easy to read	10%
Additional checks informative messages have been added ("You win", "You lost", "You already tried this letter", "You did not enter a single letter", etc)	20%