

Question #1:

Ultimate Battleships (100 points)

Please read [important information](#) first.

Description

In this homework, your task is to implement the game of Ultimate Battleships. The game is for two people and consists of two phases.

First phase is placement phase where players place their 5 available ships on the board. The second phase is battle phase where players try to sink each other's ships. The player with the last ship standing wins.

We provided many functions to you, to be used in specific cases during the game. Below, we explain where each should be used.

- Firstly, the rules are printed once at the beginning of the game by the `print_rules()` function. Check the rules for some details about the game.
- There is a function named `print_3d_list()` which expects a 3d list formed of two 2d lists which are supposed to be the player boards and prints them in the expected format. You need to use this every time you need to print the boards.
- The function `print_ships_to_be_placed()` prints the remaining ships to be placed for the player who is taking their turn at that time. It is to be printed immediately after the board during the placement phase.
- The function `print_single_element()` is used to print a single element followed by a single whitespace, and the cursor stays on the same line afterwards. Great for printing elements of a list, such as the names of remaining ships to be placed.
- The function `print_empty_line()` is needed when you need to move the cursor to a new line.
- The function `print_player_turn_to_place()` is used after the remaining ships to be placed are printed. It shows whose turn it is to place those ships.
- The function `print_to_place_ships()` is used immediately after whose turn to place ships is now is announced. It asks for the information to place a ship.
- The function `print_incorrect_input_format()` is to be used during both phases when the user provides bad input. For example, the coordinates might be floats when they should have been integers, or there may not be enough arguments provided. If the user provides more than needed arguments, take the first 4 and disregard the rest.
- The function `print_incorrect_coordinates()` is to be used during both phases when the user provides coordinates outside the board.
- The function `print_incorrect_ship_name()` is to be used during the placement phase when the ship name the user provides is not a valid one.
- The function `print_incorrect_orientation()` is to be used during the placement phase when the orientation the user provides is not a valid one.
- The function `print_ship_is_already_placed()` is to be used during the placement phase when the ship name the player chose to place is already placed.
- The function `print_ship_cannot_be_placed_outside()` is to be used during the placement phase when the coordinates and orientation the player chose for a ship exceeds the bounds of the board.
- The function `print_ship_cannot_be_placed_occupied()` is to be used during the placement phase when the coordinates and orientation the player chose for a ship coincides with an already placed ship by them.
- The function `print_confirm_placement()` is to be used at the end of the placement phase when a player has finished placing their ships. We ask if they are sure of their placement. If so, we either move on to the next player or we begin the battle phase. If they want to change their placement, we reset all their ships and start placement again for that player.
- The function `print_player_turn_to_strike()` is to be used during the battle phase, immediately after printing the board to show which player's turn it is.
- The function `print_choose_target_coordinates()` is to be used during the battle phase, after whose turn to strike is announced. It asks the player to choose coordinates to strike on the board.
- The function `print_miss()` is used during the battle phase, after a player takes their turn to strike a coordinate on the enemy board. This is used when the strike fails to hit any enemy ships.
- The function `print_hit()` is used during the battle phase, after a player takes their turn to strike a coordinate on the enemy board. This is used when the strike successfully hits an enemy ship.
- The function `print_type_done_to_yield()` is used during the battle phase, after a player takes their turn to strike and they miss. This is to prevent players accidentally seeing each other's boards.
- The function `print_tile_already_struck()` is used during the battle phase, after a player takes their turn to strike a tile. This is used when the player accidentally chooses an already attacked tile.
- The function `print_player_won()` is used at the end of the battle phase, when a player successfully sinks the other's ships.
- The function `print_thanks_for_playing()` is used at the end of the game, after the winner is announced.

Certain implementation details:

- You must handle inputs in a case insensitive manner for this homework. For example, "carrier 1 5 h", "Carrier 1 5 h" and "cARRier 1 5 H" are all valid inputs. "DONE" or "dONe" works the same as "done", "Y" and "y" are both supposed to work.
- Make sure to strip the inputs you get, otherwise you may run into problems during grading even if your code works

when you run it.

- If the player gives a bad input, or makes an invalid move, we don't take away their turns. We tell them what they did wrong and let them take their turn again.
- The ship tiles are denoted with the "#" character, the unknown sea tiles are denoted with "-", the missed strikes are denoted with "O", and hits are denoted with "!" on player boards.
- When each player takes their turn, their own board should be fully visible, but the enemy ships should not be visible on the other board, only the hits and misses should be shown.
- When a player misses their strike, they yield their turn. When they hit, they get to make another strike until they miss.
- If a player types anything that is not a case insensitive done, they are repeatedly asked to write done until they yield.

Check the example input/output files for further clarification. Keep in mind that we will be grading your code not just based on these examples, but other cases as well, so try to write code which can handle all possible cases.

Warning: You are not allowed to use imports and topics that haven't been covered this semester.

Important: Never take input prompts throughout this homework. Never print anything by yourselves.

Hint: Nested lists to represent the boards makes your job easier.