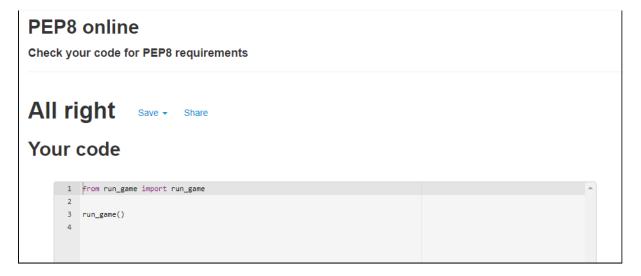
run.py



run_game.py



game_entry.py

```
PEP8 online
Check your code for PEP8 requirements
All right Save - Share
Your code
        2 Gets the players game input and updates the player pattern
        4 import compare
        5 import game_board
        6 import clear_screen
       8 available_rows = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']
       10
       11 - def get_valid_coordinates(size):
       12
       13
             Calculates the valid coordinates depending on screen size
       14
       15 valid_coordinates = []
        Check again
```

game_board.py

```
PEP8 online
Check your code for PEP8 requirements
All right
                      Save ▼ Share
Your code
         2 Prints the game board on the terminal
         4 from math import ceil
         5 import game_entry
         8 - def board_element(i, k, direction, game_pattern):
              Support function for calc_header function.

Determines the order of iteration through the board list,
        10
        11
              depending on header direction.
        12
        13
        14 - if direction == 'vertical':
        15 return game_pattern[-i][k]
```

end_game.py

compare.py

```
PEP8 online
Check your code for PEP8 requirements
All right
                     Save ▼ Share
Your code
        2 Contains functions to evaluate the game result and display it to the player
        6 → def compare_patterns(size, player_pattern, game_pattern):
             Compares the player and game patterns to calculate the number of errors
        9
        10
            compare_player_pattern = player_pattern
       13
              list_1 = list(map(lambda item: item.replace(chr(183), '0'),
                               compare_player_pattern[i]))
       15 list_2 = list(map(lambda item: item.replace(chr(0x25A1), '0'), list_1))
```

clear_screen.py

#