

Problem Set (Python)

Complete as much as you want, when you want. No rules on what you can lookup/google, though I do recommend spending more time googling how to do a specific operation you want then the problem itself, particularly for easier problems.

Problems here are specified as function headers which take input and with an expected output, though some problems may involve multiple functions if there's class design or similar involved.

Generally problems are intended to loosely get harder for later numbers, though obviously it depends on your comfort level with various concepts. I generally make the last problem or two quite a bit harder for fun though...

Problems

in_celsius

```
def in_celsius(temp : float) -> float:
    """
    Returns the celsius equivalent of the farhenheit `temp`

    Examples:
    in_celsius(32.0) ~> 0.0
    in_celsius(212.0) ~> 100.0
    in_celsius(-40.0) ~> -40.0
    """
```

min_four

```
def min_four(x : int, y : int, z : int, w : int) -> int:
    """
    Returns the minimum of the four given integers

    Examples:
    min_four(1, 2, 3, 4) -> 1
    min_four(3, 2, 4, 6) -> 2
    min_four(0, 0, -5, 0) -> -5
    """
```

second_vowel

```
def second_vowel(s : str) -> int:
    """
```

Returns the index of the second vowel in `s`
 If no vowels are in `s`, returns -1

A vowel is defined to be 'aeiou', and a 'y' that isn't the first character

Examples:

```
second_vowel('test') -> -1
second_vowel('aeiou') -> 1
second_vowel('slowly') -> 5
second_vowel('yip') -> -1
"""
```

reverse_string

```
def reverse_string(s : str) -> str:
    """
    Returns the string `s` in reverse order
```

Examples:

```
reverse_string('abc') -> 'cba'
reverse_string('') -> ''
reverse_string('abab') -> 'baba'
"""
```

max_even

```
def max_even(lst : list[int]) -> int | None:
    """
    Returns the maximum even number in `lst`, or `None` if none exists
    Note that negatives are even if the associated positive is
```

Examples:

```
max_even([1, 2, 3]) -> 2
max_even([-4, 3, -1, -6]) -> -4
max_even([1, 3, 5]) -> None
max_even([0]) -> 0
"""
```

sum_all

```
def sum_all(table : list[list[int]]) -> list[int]:
    """
    Returns the sum of each row of the table as a list
```

Examples:

```
sum_all([[1, 2, 3], [4, 5, 6]]) -> [5, 7, 9]
```

```
sum_all([[2], [4, 7], [-1, -2, -4, -5]]) -> [2, 11, -12]  
sum_all([]) -> []  
"""
```