# Python Lists and Strings Coding

## Exam 3:

## Answers:

1: String Length Check [3 marks]

Write a function `check\_string\_length(s)` that returns True if the string length is

between 5 and 10 characters (inclusive), False otherwise.

Example:

Input: &quot;Hello&quot;

Output: True

Input: &quot;Hi&quot;

Output: False

# Write your answer here

**Answer:**

def check\_string\_length(s):

return 5 <= len(s) <= 10

#test cases:

print(check\_string\_length("Hello"))

# True

print(check\_string\_length("Hi"))

# False

# Write your answer here

2: List Sum [3 marks]

Write a function `sum\_list(numbers)` that takes a list of numbers and returns

their sum. Handle empty lists by returning 0.

Example:

Input: [1, 2, 3, 4, 5]

Output: 15

Input: []

Output: 0

# Write your answer here

**Answer:**

def sum\_list(numbers):

return sum(numbers) if numbers else 0

#Test case:

print(sum\_list([1, 2, 3, 4, 5]))

# 15

print(sum\_list([]))

# 0

# Write your answer here

3: String Case Converter [3 marks]

Write a function `convert\_case(s)` that converts a string to uppercase if it

contains more uppercase letters, and lowercase if it contains more lowercase

letters. If equal, return the original string.

Example:

Input: &quot;HeLLo&quot;

Output: &quot;HELLO&quot;

Input: &quot;hello&quot;

Output: &quot;hello&quot;

# Write your answer here

**Answer:**

def converting\_case(s):

upper\_count = sum(1 for char in s if char.isupper())

lower\_count = sum(1 for char in s if char.islower())

if upper\_count > lower\_count:

return s.upper()

elif lower\_count > upper\_count:

return s.lower() return s

#Test case:

print(convert\_case("HeLLo"))

"HELLO"

print(convert\_case("hello"))

"hello"

# Write your answer here

4: List Duplicate Remover [3 marks]

Write a function `remove\_duplicates(lst)` that removes duplicate elements

from a list while preserving the order of first occurrence.

Example:

Input: [1, 2, 2, 3, 3, 4, 5, 5]

Output: [1, 2, 3, 4, 5]

Input: [&quot;apple&quot;, &quot;banana&quot;, &quot;apple&quot;, &quot;cherry&quot;]

Output: [&quot;apple&quot;, &quot;banana&quot;, &quot;cherry&quot;]

# Write your answer here

**Answer:**

def remove\_duplicates(lst):

seen = set()

return [x for x in lst if not (x in seen or seen.add(x))]

#Test case:

print(remove\_duplicates([1, 2, 2, 3, 3, 4, 5, 5]))

# [1, 2, 3, 4, 5] print(remove\_duplicates(["apple", "banana", "apple", "cherry"])

# ["apple", "banana", "cherry"]

# Write your answer here

5: String Word Counter [3 marks]

Write a function `count\_words(s)` that counts the number of words in a string.

Words are separated by spaces.

Example:

Input: &quot;Hello world&quot;

Output: 2

Input: &quot;This is a test sentence&quot;

Output: 5

# Write your answer here

**Answer:**

def counting\_words(s):

return len(s.split())

# Test case:

print(count\_words("Hello world"))

# 2

print(count\_words("This is a test sentence"))

# 5

# Write your answer here

6: List Even Numbers [3 marks]

Write a function `get\_even\_numbers(lst)` that returns a new list containing

only the even numbers from the input list.

Example:

Input: [1, 2, 3, 4, 5, 6, 7, 8]

Output: [2, 4, 6, 8]

Input: [1, 3, 5, 7]

Output: []

# Write your answer here

**Answer:**

def get\_even\_numbers(lst):

return [num for num in lst if num % 2 == 0]

# Test cases

print(get\_even\_numbers([1, 2, 3, 4, 5, 6, 7, 8]))

# [2, 4, 6, 8]

print(get\_even\_numbers([1, 3, 5, 7]))

# []

# Write your answer here

7: String Reverser [3 marks]

Write a function `reverse\_string(s)` that reverses a string without using the

built-in reverse() method.

Example:

Input: &quot;hello&quot;

Output: &quot;olleh&quot;

Input: &quot;python&quot;

Output: &quot;nohtyp&quot;

# Write your answer here

**Answer:**

def reversing\_string(s):

return s[::-1]

# Test cases

print(reverse\_string("hello"))

# "olleh"

print(reverse\_string("python"))

# "nohtyp"

# Write your answer here

8: List Element Counter [3 marks]

Write a function `count\_element(lst, element)` that counts how many times a

specific element appears in a list.

Example:

Input: lst = [1, 2, 2, 3, 2, 4], element = 2

Output: 3

Input: lst = [&quot;a&quot;, &quot;b&quot;, &quot;a&quot;, &quot;c&quot;, &quot;a&quot;], element = &quot;a&quot;

Output: 3

# Write your answer here

**Answer:**

def counting\_element(lst, element):

return lst.count(element)

# Test cases

print(count\_element([1, 2, 2, 3, 2, 4], 2))

# 3

print(count\_element(["a", "b", "a", "c", "a"], "a"))

# 3

# Write your answer here

9: String Vowel Counter [3 marks]

Write a function `count\_vowels(s)` that counts the number of vowels (a, e, i, o,

u) in a string, ignoring case.

Example:

Input: &quot;Hello World&quot;

Output: 3

Input: &quot;Python Programming&quot;

Output: 4

# Write your answer here

**Answer:**

def counting\_vowels(s):

vowels = "aeiouAEIOU"

return sum(1 for char in s if char in vowels)

# Test cases

print(count\_vowels("Hello World"))

# 3

print(count\_vowels("Python Programming"))

# 4

# Write your answer here

Bonus Challenge [Extra Practice]

[Extra Practice]

Write a function `is\_anagram(s1, s2)` that checks if two strings are anagrams of each other.

An anagram is a word formed by rearranging the letters of another word, using all the

original letters exactly once. For example, &quot;listen&quot; and &quot;silent&quot; are anagrams because they

contain the same letters in different order.

Example:

Input: s1 = &quot;listen&quot;, s2 = &quot;silent&quot;

Output: True

Input: s1 = &quot;hello&quot;, s2 = &quot;world&quot;

Output: False

# Write your answer here

Answer:

def is\_anagram(s1, s2):

return sorted(s1) == sorted(s2)

# Test cases

print(is\_anagram("listen", "silent"))

# True

print(is\_anagram("hello", "world"))

# False

# Write your answer here