Python Lists and Strings Coding Questions

**Total Marks:** 30  
(10 questions × 3 marks each)  
  
**Passing Marks:** 12  
**Excellent Marks:** 24 and above

**Difficulty Level:** Easy  
**Time:** 60 minutes

# Instructions

* Questions are marked based on difficulty:
* - Easy questions: 3 marks each
* - Medium questions: 5 marks each
* - Difficult questions: 10 marks each
* Write clean, well-commented code
* Include basic error handling where appropriate
* Test your code with the given test cases
* Bonus question is for extra practice and does not count towards total marks

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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: "Hello"  
Output: True  
  
Input: "Hi"  
Output: False

# Write your answer here

Answer: def check \_ string \_ length(s):

return 5 <= len (s) <= 10

# Example usage

print (check \_string \_ length("Hello")) # True

print (check \_ string \_ length("Hi")) # False

print (check \_ string \_ length("OpenAI")) # True

## 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:

# Write your answer here

def check \_ string \_ length(s):

return 5 <= len(s) <= 10

def sum \_ list(numbers):

return sum(numbers)

# Example usage

print(check \_ string \_ length("Hello")) # True

print(check \_ string \_ length("Hi")) # False

print(check \_ string \_ length("OpenAI")) # True

print(sum \_ list([1, 2, 3, 4, 5])) # 15

# 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: "HeLLo"  
Output: "HELLO"  
  
Input: "hello"  
Output: "hello"

# Write your answer here

Answer: def convert \_ 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

# Example usage

print(convert \_ case("HeLLo")) # "HELLO"

print(convert \_ case("hello")) # "hello"

print(convert \_ case("HeLlo")) # "HeLlo"

Top of Form

Bottom of Form

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

# 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))]

# Example usage

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: "Hello world"  
Output: 2  
  
Input: "This is a test sentence"  
Output: 5

# Write your answer here

Answer: def count\_words(s):

return len(s.split())

# Example usage

print(count\_words("Hello world")) # Output: 2

print(count\_words("This is a test sentence")) # Output: 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 [x for x in lst if x % 2 == 0]

# Example usage

print(get\_ even \_numbers([1, 2, 3, 4, 5, 6, 7, 8])) # Output: [2, 4, 6, 8]

print(get\_ even\_ numbers([1, 3, 5, 7])) # Output: []

# 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: "hello"  
Output: "olleh"  
  
Input: "python"  
Output: "nohtyp"

# Write your answer here

Answer: def reverse\_ string(s):

return s[::-1]

# Example usage

print(reverse\_ string("hello")) # Output: "olleh"

print(reverse\_ string("python")) # Output: "nohtyp"

This implementation efficiently reverses the string without using the built-in reverse() method. Let me know if you need any modifications! 🚀

# 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 = ["a", "b", "a", "c", "a"], element = "a"  
Output: 3

# Write your answer here

Answer: def count\_ element(lst, element):

return lst. count(element)

# Example usage

print(count\_ element([1, 2, 2, 3, 2, 4], 2)) # Output: 3

print(count\_ element(["a", "b", "a", "c", "a"], "a")) # Output: 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: "Hello World"  
Output: 3  
  
Input: "Python Programming"  
Output: 4

# Write your answer here

Answer: def count\_ vowels(s):

vowels = "aeiouAEIOU"

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

# Example usage

print(count\_ vowels("Hello World")) # Output: 3

print(count\_ vowels("Python Programming")) # Output: 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, "listen" and "silent" are anagrams because they contain the same letters in different order.

Example:

Input: s1 = "listen", s2 = "silent"  
Output: True  
  
Input: s1 = "hello", s2 = "world"  
Output: False

# Write your answer here

Answer: def is\_ anagram(s1, s2):

return sorted(s1) == sorted(s2)

# Example usage

print(is\_ anagram("listen", "silent")) # Output: True

print(is\_ anagram("hello", "world")) # Output: False

# Write your answer here

# Evaluation Criteria

* Correctness (50%)
* - Function works as expected (30%)
* - Handles edge cases correctly (20%)
* Code Readability (30%)
* - Clear variable names (10%)
* - Proper comments (10%)
* - Clean code structure (10%)
* Basic Error Handling (20%)
* - Input validation (10%)
* - Appropriate error messages (10%)

# Marking Scheme

* Easy questions: 3 marks each
* Medium questions: 5 marks each
* Difficult questions: 10 marks each
* Total marks for the paper: 30
* Passing marks: 12
* Excellent marks: 24 and above

Good Luck!