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

def check\_string\_length(s):

if len(s) >= 5 and len(s) <= 10:

return True

else:

return False

# Example test cases

print(check\_string\_length("hello")) # True

print(check\_string\_length("Python")) # True

print(check\_string\_length("tiny")) # False

print(check\_string\_length("thisislonger")) # False

Answer:

# 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

def sum\_list(numbers):

if len(numbers) == 0: # Check if the list is empty

return 0

else:

return sum(numbers) # Add all numbers in the list

# Example test cases

print(sum\_list([1, 2, 3, 4, 5])) # Output: 15

print(sum\_list([])) # Output: 0

print(sum\_list([10, -5, 7])) # Output: 12

Answer:

# 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:

# Write your answer here

def convert\_case(s):

upper = 0

lower = 0

for c in s:

if c.isupper():

upper += 1

elif c.islower():

lower += 1

if upper > lower:

return s.upper()

elif lower > upper:

return s.lower()

else:

return s

# Example test cases

print(convert\_case("HeLLo")) # Output: "HELLO"

print(convert\_case("hello")) # Output: "hello"

print(convert\_case("Hello")) # Output: "Hello"

print(convert\_case("PYthon")) # Output: "python"

print(convert\_case("PYTHON")) # Output: "PYTHON"

## 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:

# Write your answer here

def remove\_duplicates(lst):

unique\_list = [] # Create an empty list to store unique elements

for item in lst:

if item not in unique\_list: # Check if item is already in unique\_list

unique\_list.append(item) # Add it if not present

return unique\_list # Return the list without duplicates

# Example test cases

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

print(remove\_duplicates(["apple", "banana", "apple", "orange"])) # Output: ['apple', 'banana', 'orange']

print(remove\_duplicates([5, 5, 5, 5, 5])) # Output: [5]

## 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:

# Write your answer here

def count\_words(s):

words = s.split() # Split the string into words using spaces

return len(words) # Count and return the number of words

# Example test cases

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

print(count\_words("This is a test")) # Output: 4

print(count\_words("Python programming is fun")) # Output: 4

print(count\_words(" ")) # Output: 0 (empty string or spaces)

print(count\_words("OneWord")) # Output: 1

## 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:

def get\_even\_numbers(lst):

even\_list = [] # Create an empty list to store even numbers

for num in lst:

if num % 2 == 0: # Check if the number is even

even\_list.append(num) # Add even number to the list

return even\_list # Return the list of even numbers

# Example test cases

print(get\_even\_numbers([1, 2, 3, 4, 5, 6])) # Output: [2, 4, 6]

print(get\_even\_numbers([10, 15, 20, 25, 30])) # Output: [10, 20, 30]

print(get\_even\_numbers([1, 3, 5, 7])) # Output: [] (no even numbers)

print(get\_even\_numbers([8, 12, 14, 17, 19])) # Output: [8, 12, 14]

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:

# 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:

def reverse\_string(s):

reversed\_s = "" # Step 1: Start with an empty string

for char in s: # Step 2: Loop through each character in the string

reversed\_s = char + reversed\_s # Step 3: Add each character at the beginning

return reversed\_s # Step 4: Return the reversed string

# Example test cases

print(reverse\_string("hello")) # Output: "olleh"

print(reverse\_string("Python")) # Output: "nohtyP"

print(reverse\_string("12345")) # Output: "54321"

print(reverse\_string("a")) # Output: "a"

print(reverse\_string("")) # Output: ""

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

# Write your answer here

Answer:

# 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:

# Write your answer here

def count\_element(lst, element):

count = 0 # Initialize count to 0

for item in lst:

if item == element: # Check if the item matches the given element

count += 1 # Increase count

return count # Return the final count

# Example test cases

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

print(count\_element(["apple", "banana", "apple", "orange"], "apple")) # Output: 2

print(count\_element([5, 5, 5, 5, 5], 5)) # Output: 5

print(count\_element([1, 2, 3, 4], 10)) # Output: 0 (element not found)

## 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:

# Write your answer here

def count\_vowels(s):

vowels = "aeiouAEIOU" # Define vowels (both lowercase and uppercase)

count = 0 # Initialize count to 0

for char in s:

if char in vowels: # Check if the character is a vowel

count += 1 # Increase count

return count # Return the total count of vowels

# Example test cases

print(count\_vowels("hello")) # Output: 2 (e, o)

print(count\_vowels("Python Programming")) # Output: 4 (o, o, a, i)

print(count\_vowels("AEIOU")) # Output: 5 (A, E, I, O, U)

print(count\_vowels("xyz")) # Output: 0 (no vowels)

print(count\_vowels("This is a test")) # Output: 4 (i, i, a, e)

# 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:

# Write your answer here

def is\_anagram(s1, s2):

return sorted(s1) == sorted(s2) # Sort both strings and compare

# Example test cases

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

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

print(is\_anagram("triangle", "integral")) # Output: True

print(is\_anagram("Python", "nothyp")) # Output: True

print(is\_anagram("apple", "pale")) # Output: False

# 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!