Documentation for Python Tasks

# Task 1: Sort Characters in a String

This task implements bubble sort on characters of a string. The program converts the string into a list of characters, then sorts them based on their ASCII values using the bubble sort algorithm.

Code Example:

text = "Hi mehwish"  
class Sort\_string:  
 def \_\_init\_\_(self):  
 self.list = list(text)  
 def sort(self):  
 n = len(self.list)  
 for i in range(n - 1):  
 for j in range(n - 1 - i):  
 if ord(self.list[j]) > ord(self.list[j + 1]):  
 self.list[j], self.list[j + 1] = self.list[j + 1], self.list[j]  
 return "".join(self.list)  
   
n = Sort\_string()  
sorted = n.sort()  
print("original:", text)  
print("updated text:", sorted)

Output: The characters of the string are sorted alphabetically.

# Task 2: Sort Words in a String

This task modifies the bubble sort logic to work with words instead of characters. The string is split into words and then sorted alphabetically using bubble sort.

Code Example:

text = "Hi mehwish how are you"  
  
class Sort\_string:  
 def \_\_init\_\_(self):  
 self.words = text.split()  
 def sort(self):  
 n = len(self.words)  
 for i in range(n - 1):  
 for j in range(n - 1 - i):  
 if self.words[j].lower() > self.words[j + 1].lower():  
 self.words[j], self.words[j + 1] = self.words[j + 1], self.words[j]  
 return " ".join(self.words)  
  
n = Sort\_string()  
sorted\_text = n.sort()  
print("original:", text)  
print("updated text:", sorted\_text)

Output: The words of the string are sorted alphabetically.

# Task 3: Luhn Algorithm

The Luhn Algorithm is used to validate credit card numbers. It works by doubling every second digit from the right, subtracting 9 if the result is greater than 9, and then summing all digits. If the total sum is divisible by 10, the number is valid.

Code Example:

def luhn\_algorithm(card\_number):  
 digits = [int(d) for d in str(card\_number)]  
 for i in range(len(digits) - 2, -1, -2):  
 digits[i] \*= 2  
 if digits[i] > 9:  
 digits[i] -= 9  
 return sum(digits) % 10 == 0  
  
card\_number = 4532015112830366  
print("Valid:" if luhn\_algorithm(card\_number) else "Invalid")

Output: The program checks if the card number is valid using the Luhn Algorithm.

# Task 4: Remove Punctuation from Text

This task removes punctuation marks from a given string using the string.punctuation library. It is useful in text preprocessing tasks like Natural Language Processing (NLP).

Code Example:

import string  
  
text = "Hello, world! How's it going?"  
cleaned\_text = "".join(ch for ch in text if ch not in string.punctuation)  
print("Original:", text)  
print("Without punctuation:", cleaned\_text)

Output: The text is cleaned by removing all punctuation marks.