**Text and Number Processing Programs**

Name: Ayesha Sikandar  
Roll No.: 044  
Section: BSDS 3A

**1. Luhn Algorithm**

**Code Explanation:**

This program checks whether a credit card number is valid according to the Luhn Algorithm.

* The number is split into individual digits.
* Starting from the right, every second digit is doubled. If doubling produces a number greater than 9, 9 is subtracted from it.
* All digits are then summed, and the result is valid if the sum is divisible by 10.
* The program also validates input to ensure the user enters only digits.

**Why I Made This:**

I made this program to understand real-world number validation algorithms and how they can be implemented in Python.  
It helps in learning loops, conditions, and list operations in practical scenarios like credit card verification.

**How I Made It:**

* Converted the input number into a list of digits.
* Used a for loop with enumeration to handle doubling and summing according to the Luhn rules.
* Checked if the sum modulo 10 equals 0 to determine validity.
* Printed friendly messages to guide the user.

**2. Punctuation Algorithm**

**Code Explanation:**

This program removes all punctuation and special characters from a user-input sentence.

* The program loops through each character in the input.
* It keeps only letters, numbers, and spaces based on ASCII values.
* The cleaned text is then printed without punctuation.

**Why I Made This:**

I made this program to learn how to filter strings in Python and clean text data.  
It is useful in applications like text processing, NLP, and data cleaning where punctuation might need to be removed.

**How I Made It:**

* Used the ord() function to get ASCII values of characters.
* Applied a conditional check to keep letters, digits, and spaces only.
* Built the cleaned string character by character using a loop.
* Printed both the original and cleaned text for comparison.

**3. Ascii Algorithm**

**Code Explanation:**

This program sorts the words of a sentence alphabetically.

* The sentence is split into words using the split() method.
* The list of words is sorted alphabetically in a case-insensitive manner.
* Finally, the sorted words are joined back into a sentence and printed.

**Why I Made This:**

I made this program to learn how to manipulate text and lists in Python.  
It demonstrates basic string and list methods, and how sorting can be applied to real-world text data.

**How I Made It:**

* Split the sentence into a list of words.
* Used the sorted() function with key=str.lower for case-insensitive sorting.
* Joined the sorted words using " ".join() and printed the result.

**Conclusion:**

Through these three programs, I learned how to:

* Validate numbers with algorithms like Luhn.
* Process text by removing unwanted characters.
* Manipulate and sort strings and lists efficiently in Python.

These exercises helped me practice loops, conditions, string and list operations, and user input handling, which are fundamental concepts in Python programming.