**Name: Usman Ul Haq**

**Roll# BSAIM-035**

**AI LAB TASKs**

**Documentation**

**Documentation for LUHN Algorithm**

This program validates a credit card number using the Luhn algorithm, a widely used checksum formula for detecting errors in card numbers.

Functioning of the Program

1. User Input:
   * The user is prompted to enter a credit card number as a string.
2. Reversing the Number:
   * The program reverses the card number to process it according to the Luhn algorithm.
3. Processing Each Digit:
   * The program iterates through each digit of the reversed number:
     + Odd-positioned digits (when counting from 0) are doubled.
     + If doubling results in a value greater than 9, 9 is subtracted from the result (equivalent to summing the individual digits).
     + Even-positioned digits remain unchanged.
   * The processed digits are summed together.
4. Final Validation Check:
   * If the total sum is divisible by 10, the card number is valid.
   * Otherwise, the card number is invalid.

This program effectively checks whether a given credit card number follows the Luhn algorithm, ensuring its correctness before processing transactions.

**2. Write a python program to remove punctuations from the given string?**

This program **removes punctuation** from a given sentence, ensuring that only letters, numbers, and spaces remain.

**Functioning of the Program**

1. **User Input:**
   * The user is prompted to enter a sentence.
2. **Defining Punctuation Characters:**
   * A string containing common punctuation marks (!()-[]{};:'"\,<>./?@#$%^&\*\_~) is stored.
3. **Processing Each Character:**
   * The program iterates through each character of the input sentence.
   * If the character **is not** a punctuation mark, it is added to the cleaned\_text string.
4. **Displaying the Result:**
   * The modified sentence, now free of punctuation, is printed.

This simple yet effective program is useful for **text preprocessing** in natural language processing (NLP), improving readability, and ensuring cleaner text formatting.

**3. Write a python program to sort the sentence in alphabetical order?**

This program **sorts the words of a sentence in alphabetical order**, ignoring case sensitivity.

**Functioning of the Program**

1. **User Input:**
   * The user enters a sentence.
2. **Splitting Words:**
   * The sentence is broken down into individual words using the split() method, creating a list of words.
3. **Sorting Words Alphabetically:**
   * The words are sorted **in a case-insensitive manner** using sorted(words, key=str.lower), ensuring that uppercase and lowercase letters do not affect the order.
   * The sorted words are stored in a new list.
4. **Reconstructing the Sentence:**
   * The sorted words are joined back into a sentence using ' '.join(sorted\_words).
5. **Displaying the Output:**
   * The sorted sentence is printed.

This program is useful for **text processing** tasks such as **dictionary ordering, natural language analysis, and improving readability**.