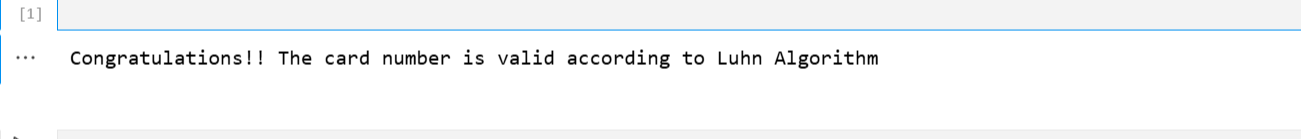
**Task # 4 Description**

**Question # 1:**

1. **Function(num)**
   * Converts the number into a list of digits (arrange).
   * Removes the **last digit** (ending\_value) which is the check digit.
   * Reverses the remaining digits (because Luhn works right-to-left).
2. **Double Every Second Digit**
   * Starting from index 0 (the rightmost digit after reversal), it doubles every alternate digit.
   * If doubling makes a number > 9, it subtracts 9 (equivalent to adding the digits of the product).
3. **Final Sum**
   * Adds up the transformed digits + the ending\_value.
   * If the total modulo 10 is 0, the number is valid.
4. **Usage**
   * The string "5893804115457289" is checked.
   * If valid → prints a success message.
   * If not → prints an error message

**Output Screen shots:**

****

**Question # 2:**

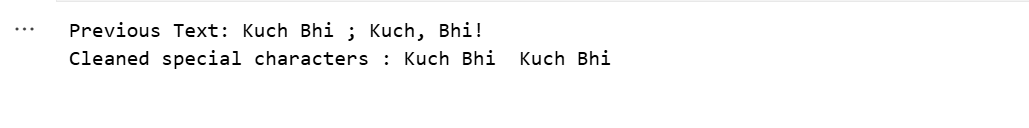
This program removes all **special characters** (like ; , ! @ #) from a given text and keeps only **letters, digits, and spaces**

**Function finish\_specials(txt)**

* + Creates an empty string clean\_txt to store the cleaned version.
  + Loops through each character (symbol) in the input text.
  + Uses two checks:
    - symbol.isalnum() → keeps alphabets and numbers.
    - symbol.isspace() → keeps spaces.
  + If either condition is true, that character is added to clean\_txt.

1. **Input Example**
   * Input: "Kuch Bhi ; Kuch, Bhi!"
   * Special characters ;, ,, and ! are ignored.
2. **Output**
   * Original: Kuch Bhi ; Kuch, Bhi!
   * Cleaned: Kuch Bhi Kuch Bhi (notice extra space where punctuation was removed).

**Output Screen shots:**

****

**Question # 3:**

This program sorts both the **characters** and **words** of a given string using the **Bubble Sort algorithm**

1. **Function sort(arr)**
   * Finds the length of the list n.
   * Uses two nested loops to compare adjacent elements.
   * If the current element arr[j] is greater than the next one arr[j+1], they are swapped.
   * After all iterations, the list becomes sorted in **ascending order** (A → Z, a → z).
2. **Character Sorting**
   * Takes input from the user (s).
   * Removes all spaces using replace(" ", "") and converts it to a list of characters.
   * Calls sort(chars) to alphabetically sort characters.
   * Joins the sorted list back into a string and prints it.
3. **Word Sorting**
   * Splits the original string into a list of words using split().
   * Sorts the words alphabetically using the same sort() function.
   * Joins and prints them back as a sentence.

**Output Screen shots:**

****