

**Name:**  Muhammad Awais Asghar

**Roll Number:** SU92-BSAIM-F24-065

**SUBJECT:** Artifical Intelligence (LAB)

**Task No -4**

**Question 1:**

**Why this code was made:**

The goal of this application is to use the Luhn algorithm to validate card numbers.  
In the real world, the Luhn algorithm is frequently used to verify SIM serial numbers, debit cards, and credit cards.This application assists in determining whether an entered number is a valid card number.

**How this code works:**

**Number of Input**  
  
A string of numbers is provided, such as "5 8 9 3 8 0 4 1 1 5 4 5 7 2 8 9".  
  
Spaces are not extracted; only the numbers are.  
 **Taking the Check Digit Separately**  
  
The check digit is the number's final digit.  
  
Until the very end, this digit is not used in the main processing.  
  
**Turning the Last Few Digits**  
  
Every digit is inverted, with the exception of the check digit.  
  
This facilitates accurate processing of alternating positions.  
  
**Every second digit is doubled.**  
  
Each digit at an even number is doubled, beginning on the left side of the inverted list.  
  
For instance, 4 = 8 and 7 =14.

If 9 is greater than 9, subtract 9.  
  
Subtract 9 from a doubled value if it is greater than 9.  
  
For instance, 14 → 5.  
  
**Conclusion**  
  
Add up all of the numbers that have been processed.  
  
Add the check digit after that.  
  
**Verification Check**  
  
A number is considered valid if the sum is divisible by 10.  
  
It is invalid otherwise.  
  
**Detailed Description**  
Each step is clearly printed by the luhn\_explain() function:  
  
The initial number  
  
Taking the check digit out  
  
Digits in reverse  
  
doubling  
  
When necessary, subtract 9.  
  
Totaling everything  
  
Final judgment



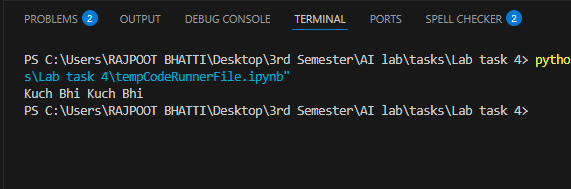
**Question 2:**

**Why this code was made:**

This program's goal is to completely remove all punctuation from a string.  
When you only want to keep the letters, numbers, and spaces from a particular text such as when preparing data for text analysis or word counting), this is helpful.

**How this code works:**

A text string has been given:  
  
"Kuch, Bhi; Kuch? Bhi!"  
  
  
The cleaned text is stored in an empty string result.  
  
One by one, the program analyzes every character in the text:  
  
The character is added to the result if it is a letter, number, or space.  
  
The character is disregarded if it is a punctuation mark (such as, ;?!).  
  
The final cleaned string is saved in result after every character has been checked.  
  
After cleaning, the text is printed.



**Question 3:**

**Why this code was made:**

This code is used to illustrate the bubble sort algorithm and its two possible applications:  
  
arranging the words in a sentence in alphabetical order.  
  
arranging every character in a sentence in alphabetical order, ignoring spaces.  
  
This application helped in understand the real-life operation of bubble sort.

**How this code works:**

Function of Bubble Sorting  
  
The list is processed through by the bubble\_sort(arr) function.  
  
It compares two nearby items each time.  
  
They change places if the left item is larger than the right.  
  
Until the entire list is sorted, this process is repeated.  
  
Organizing Words  
  
The words "I", "am", and "Awais" make up the sentence "I am Awais."  
  
They are arranged alphabetically using bubble sort.  
  
After sorting, the words are reassembled into a sentence.  
  
Character-Based Sorting  
  
Only the following characters remain after spaces are removed: ['I','a','m','A','w','a','i','s'].  
  
These characters are arranged alphabetically using bubble sort.  
  
A string is then created by combining the sorted characters.  
  
Results:  
  
prints the result of the word sorting.  
  
prints the result of the character sorting.

