

SUPERIOR UNIVERSITY LAHORE

GOLD CAMPUS

**NAME:** MUHAMMAD MOIZ RAZA

**ROLL NO:** SU92-BSAIM-F24-**061**

**PROGRAM:** ARTIFICIAL INTELLIGENCE

**SEMESTER:** 3rd SEMESTER

**SECTION:** BSAI-3A

**SUBJECT:** ARTIFICIAL INTELLIGENCE

**Submission Title:** LAB TASK-4

* python program to remove punctuations from the given string
* Write a python program to sort the sentence in alphabetical order?
* Code for LUHN Algorithm

**Submitted To: Sir RASHIK ALI**

**LAB TASK-4**

**TASK:** Write a python program to sort the sentence in alphabetical order?

**Why we write this code?**

## We write this code to filter out only the alphabets (like a–z or A–Z) from a mixed string that contains letters, numbers, symbols, and punctuation. It helps us separate meaningful text from random characters.

## **How the Code Works**

The code reads the string one character at a time, just like scanning letters in a line. For each character, it checks if it’s an alphabet using the isalpha() function. If it is, that character gets printed; if it’s a symbol, number, or space, the code simply ignores it. This way, only the letters from the string come out as the final output, leaving all the extra symbols behind. We write a senetence: “ kuch bhi , kuch bhi , ! , @ , # , $ , % , ^ , & , \* , ( , ) , \_ , - , + , = , { , } , [ , ] , | , \\ , : , ; , ' , \" , < , > , , , . , ? , /"

**OUTPUTS:**

****

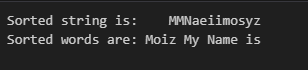
**TASK:** python program to remove punctuations from the given string

### ****Why we write this code?****

### We write this code to practice how sorting works both at the character level and the word level. It helps us understand bubble sort by arranging letters in alphabetical order and also shows how words from a sentence can be sorted. Basically, it’s a way to learn sorting step by step on text data.

## **How the Code Works**

First, the program takes the string "My Name is Moiz" and breaks it into characters. Using bubble sort, it compares letters one by one and rearranges them in alphabetical order, then joins them back to show a sorted string. Next, it splits the sentence into separate words and again applies bubble sort, this time to arrange the words in alphabetical order. Finally, it prints both the sorted characters and the sorted words.

**OUTPUTS:**

**TASK:** Code for LUHN Algorithm

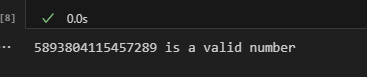
**Why we write this code?**

## We write this code to check if a number, like a credit card number, is valid or not. It uses the **Luhn algorithm**, which is commonly used in banking and payment systems to quickly verify numbers before processing.

## **How the Code Works**

The code takes the number and reverses it. Then, moving left to right, it doubles every second digit. If doubling makes the digit bigger than 9, it subtracts 9. After that, it adds all digits together. If the total ends in 0 (divisible by 10), the number is considered valid; otherwise, it’s not.

**OUTPUTS:**



THE END