

# Name: Tahir Fareed

# Roll No: SU92-BSAIM-F24-036

# Section: BSAI (3A)

# Subject: AI Lab

# Task : 04

Card Number Validation & Text Processing Documentation

This document explains the code implemented in the Jupyter Notebook 'card number.ipynb'. It includes text cleaning, character sorting, word sorting, and card number validation using the Luhn Algorithm. A flowchart is also provided for better understanding.

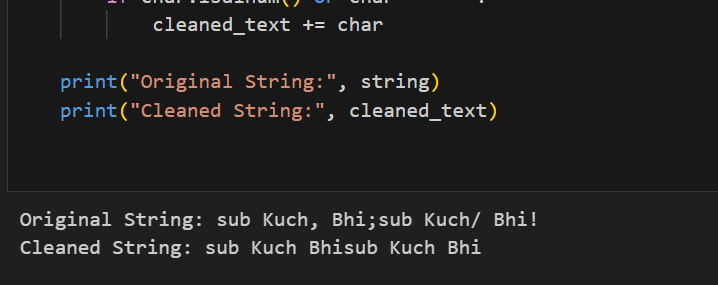
# 1. Text Cleaning

In this section, punctuation and special characters are removed from the given string, keeping only alphabets, numbers, and spaces.

Example Code:

string = "sub Kuch, Bhi;sub Kuch/ Bhi!"  
cleaned\_text = ""  
for char in string:  
 if char.isalnum() or char == " ":  
 cleaned\_text += char  
print("Original String:", string)  
print("Cleaned String:", cleaned\_text)

**screenshot :**



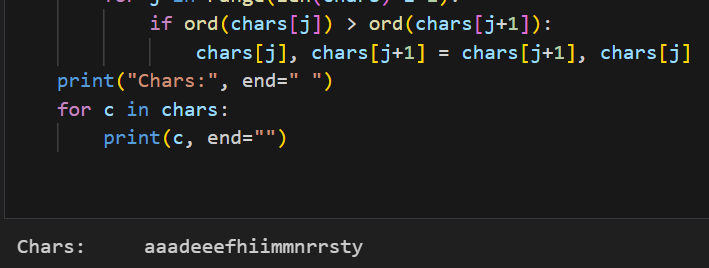
# 2. Character Sorting

This section sorts characters of a string in ascending alphabetical order using a bubble sort approach.

Example Code:

text = "my name is tahir fareed"  
chars = [c for c in text]  
for i in range(len(chars)):  
 for j in range(len(chars)-i-1):  
 if ord(chars[j]) > ord(chars[j+1]):  
 chars[j], chars[j+1] = chars[j+1], chars[j]  
print("Sorted Characters:", "".join(chars))

**screenshot :**



# 3. Word Sorting

This section splits a string into words and sorts them alphabetically by comparing letters one by one.

Example Code:

text = "my name is tahir fareed"  
words = text.split()  
for i in range(len(words)):  
 for j in range(len(words)-i-1):  
 if words[j] > words[j+1]:  
 words[j], words[j+1] = words[j+1], words[j]  
print("Sorted Words:", " ".join(words))

**screenshot :**

A computer screen shot of a computer code

AI-generated content may be incorrect.

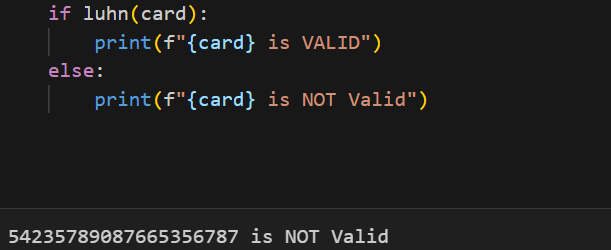
# 4. Card Validation using Luhn Algorithm

The Luhn Algorithm is used to check whether a card number is valid or not. It works as follows:  
1. Remove the last digit (check digit).  
2. Reverse the remaining digits.  
3. Double every second digit. If the result is greater than 9, subtract 9.  
4. Add all digits including the check digit.  
5. If the sum is divisible by 10, the card number is valid.

Example Code:

def luhn(card\_number):  
 check\_digit = int(card\_number[-1])  
 digits = [int(d) for d in card\_number[:-1]]  
 digits.reverse()  
 for i in range(0, len(digits), 2):  
 digits[i] \*= 2  
 if digits[i] > 9:  
 digits[i] -= 9  
 total = sum(digits) + check\_digit  
 return total % 10 == 0  
  
card = "54235789087665356787"  
if luhn(card):  
 print(f"{card} is VALID")  
else:  
 print(f"{card} is NOT Valid")

**screenshot :**

****