**Name:** Danish Akram  
**Roll No:** 063  
**Section:** BSDS – 3A  
**Subject:** AI (Lab)

# Task 04

## Question No. 1

Task:  
A string was given containing alphabets, punctuation marks, and symbols. The goal was to display only alphabetic characters from that string.

Explanation:  
- The program takes a string that contains multiple symbols such as commas, dots, brackets, etc.  
- It checks each character in the string using a condition that filters out only alphabet letters (A–Z or a–z).  
- These letters are then printed in sequence, ignoring all other characters.

Output Example:  
All alphabets from the given string are displayed continuously without any symbols.

## Question No. 2

Task:  
To perform two sorting operations on a given text:  
1. Sort all characters in the string.  
2. Sort all words alphabetically.

Explanation:  
- First, the program converts the string into a list of characters and applies a Bubble Sort algorithm to arrange them in ascending order (A–Z).  
- The sorted characters are then joined back to form a new string.  
- Secondly, the program splits the original sentence into individual words, sorts them alphabetically using the same logic, and then joins them back into a new sentence.

Output Example:  
- Sorted string (by characters): shows characters arranged in increasing order.  
- Sorted words: shows words arranged alphabetically (e.g., “danish me” → “danish me”).

## Question No. 3

Task:  
To verify whether a given number is valid according to the Luhn algorithm, commonly used for validating credit or debit card numbers.

Explanation:  
- The program reverses the given number and processes each digit.  
- Every second digit (from the right) is doubled; if the result is greater than 9, 9 is subtracted from it.  
- The sum of all digits is then calculated.  
- If the total sum is divisible by 10, the number is considered valid; otherwise, it is invalid.

Output Example:  
The program prints whether the provided number (e.g., 6011111111111117) is valid or not according to the Luhn check.