Milestone Project 1:

- 3 String Manipulation code in Python

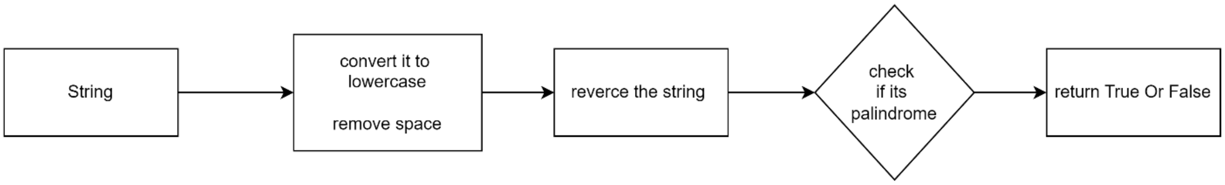
  palindrome function

  - Start: Begin the process.
  - Input the String
  - Convert to Lowercase and Remove Spaces
  - Check if String is Equal to its Reverse

    If Yes: The string is a palindrome.

    If No: The string is not a palindrome.

  - Output Result: Return whether the string is a palindrome or not.
  - End: End the process.



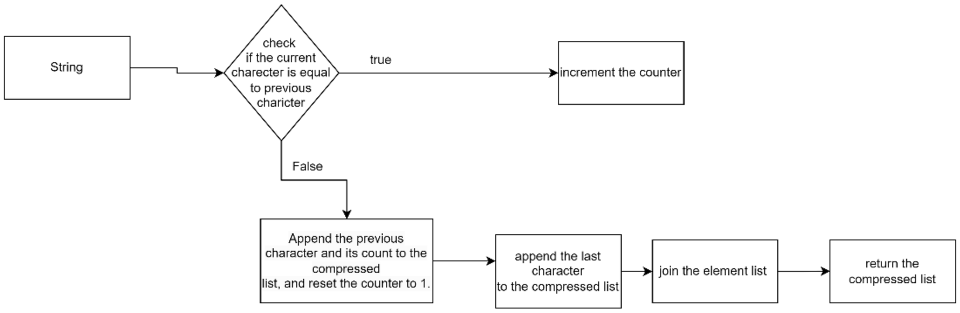
String compression function

- Start: The process begins.
- Input the String
- Initialize Variables: Initialize an empty list compressed and set count = 1.
- Iterate Over String: Loop through the string starting from the second character.
- Check if Current Character Equals Previous Character:

  Yes: Increment the count.

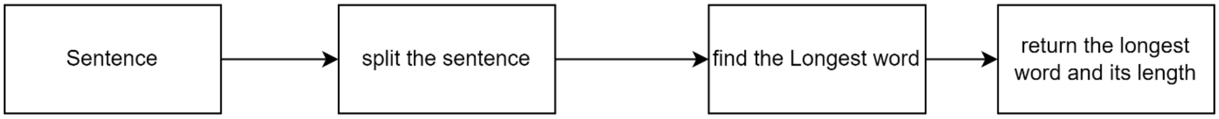
  No: Append the previous character and its count to the compressed list, and reset the count to 1.

- Append Last Character and Count: After the loop, append the last character and its count to the compressed list.
- Join the List: Join the elements of the compressed list into a single string.
- Return Compressed String: Output the compressed string.
- End: The process ends.


String
check if the current charecter is equal to previous charicter
true
increment the counter
False
Append the previous character and its count to the compressed list, and reset the counter to 1.
append the last character to the compressed list
join the element list
return the compressed list

Longest word function

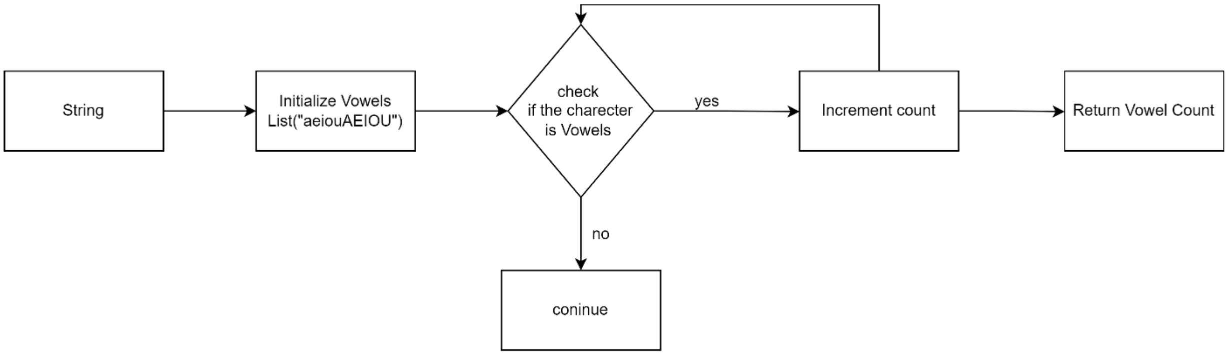
- Start: The process begins.
- Input the Sentence: The sentence.
- Split the Sentence into Words: Use the split() method to break the sentence into individual words.
- Find the Longest Word: Use the max() function to find the word with the maximum length.
- Return Longest Word and Length: Return the longest word and its length.
- End: The process ends.



3 Array based code

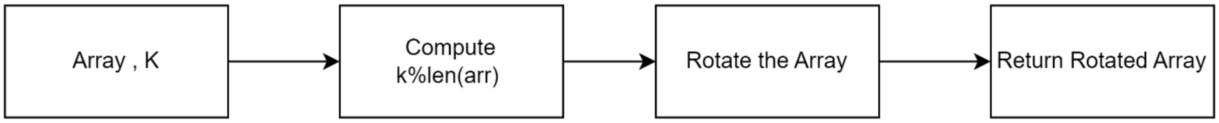
Count vowels Function

- Start: The process begins.
- Input the String.
- Initialize Vowels: Define a string containing all vowels (“aeiouAEIOU”).
- Iterate Through the String: Loop through each character in the string.
    - Check if Character is a Vowel: For each character, check if it is present in the vowels string.
        - Yes: Increment the count.
        - No: Continue to the next character.
- Return Vowel Count: Return the final count of vowels.
- End: The process ends.


String
Initialize Vowels List("aeiouAEIOU")
check if the charecter is Vowels
yes
Increment count
Return Vowel Count
no
coninue

Rotate array Function

- Start: The process begins.
- Input the Array and Rotation Value (k): The array (e.g., [1, 2, 3, 4, 5]) and the value of k are provided as input.
- Compute Effective Rotation: Calculate k % len(arr) to handle cases where k is larger than the length of the array.
- Rotate the Array: Split the array into two parts:
  - The last k elements: arr[-k:]
  - The first len(arr)-k elements: arr[:-k]
  - Concatenate the two parts.
- Return Rotated Array: Return the rotated array.
- End: The process ends.



Read a CSV file using Pandas and manipulate it.

Step 1: Importing the Required Library

Step 2: Reading the CSV File

Step 3: Renaming Columns

Step 4: Cleaning the Data

Step 5: Dropping Unnecessary Columns

Step 6: Sorting the Data

Step 7: Saving the Manipulated Data

Transform a JSON file into a CSV file.

Step 1: Importing the Required Library

Step 2: Reading a JSON File

Step 3: Converting the JSON Data to CSV

Take a sample CSV and plot a graph using Matplotlib.

Step 1: Importing the Required Libraries

Step 2: Reading the CSV File

Step 3: Plotting a Graph

df['Installs']: Represents the x-axis values (number of installs).

df['Size']: Represents the y-axis values (app size).

Step 4: Setting Graph Title and Axis Labels

Step 5: Displaying the Graph

Count word problem using Python Dictionary

Step 1: Splitting the Sentence into Words

Step 2: Creating an Empty Dictionary

Step 3: Counting Word Occurrences

Step 4: Returning the Word Dictionary