Outline Pseudocode for AnagramChecker for Topic 1: Introduction, CST-201

By Joseph Abraham

Main Procedure:

1. **Start the program**
2. **Display Greeting and Instructions:**
    - **a.** Print the greeting message:
        - "Welcome to the Joseph Abraham Anagram Checker for CST-201"
        - "Go Antilopes!!"
    - **b.** Print an instruction message: "Enter two words to check if they are anagrams."
    - **c.** Print a blank line for readability.
3. **Begin Infinite Loop (Continuous Checking):**
    - **a. Prompt for First Word:**
        - Display: "Enter the first word: "
        - Read and store user input in variable firstWord.
    - **b. Prompt for Second Word:**
        - Display: "Enter the second word: "
        - Read and store user input in variable secondWord.
    - **c. Check for Anagram:**
        - Call the function areAnagrams(firstWord, secondWord).
        - If function returns true, then:
            - Print: "We have an anagram!"
        - Otherwise, if the function returns false, then:
            - Print: "Dat not anagram!"
    - **d. Print a Blank Line:**
        - Print an empty line to separate each check.
    - **e. Repeat Step 3 Indefinitely:**
        - loop continues automatically until the program is manually terminated (e.g., by pressing Ctrl+C).

Function: areAnagrams(word1, word2):

1. **Normalize the Input Words:**
    a. Convert word1 and word2 to lowercase.
    b. Remove any spaces from both words.
2. **Check Word Lengths:**
    a. If the lengths of word1 and word2 are different:
        - Return false (since words of different lengths cannot be anagrams).
3. **Initialize Frequency Array:**
    a. Create an integer array of size 26 (representing letters ‘a’ to ‘z’), initializing all elements to zero.
4. **Count Frequency for the First Word:**
    a. For each character in word1:
    b. Calculate the index as character - 'a'.
    c. Increment the value at that index in the frequency array.
5. **Adjust Frequency Using the Second Word:**
    a. For each character in word2:
        - Calculate the index as character - 'a'.
        - Decrement the value at that index in the frequency array.
6. **Determine if Words are Anagrams:**
    a. Traverse the frequency array:
        - If every element is zero:
            - Return true (the words are anagrams).
        - If any element is not zero:
            - Return false (the words are not anagrams).
7. **End Function.**