CUETO, ALEXA JOYCE G.

TW23

PRACTICAL EXAM 1

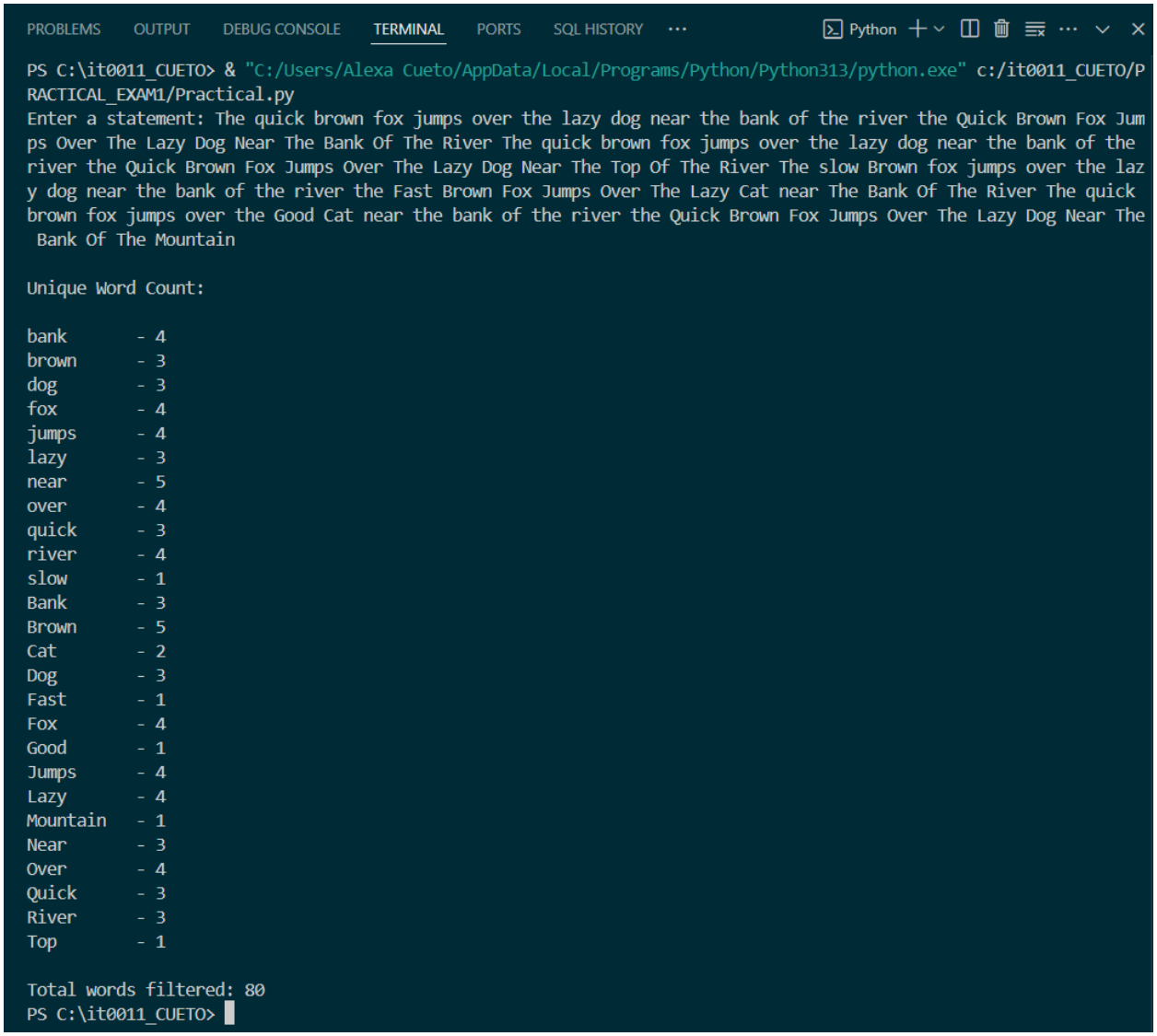
Source Code:

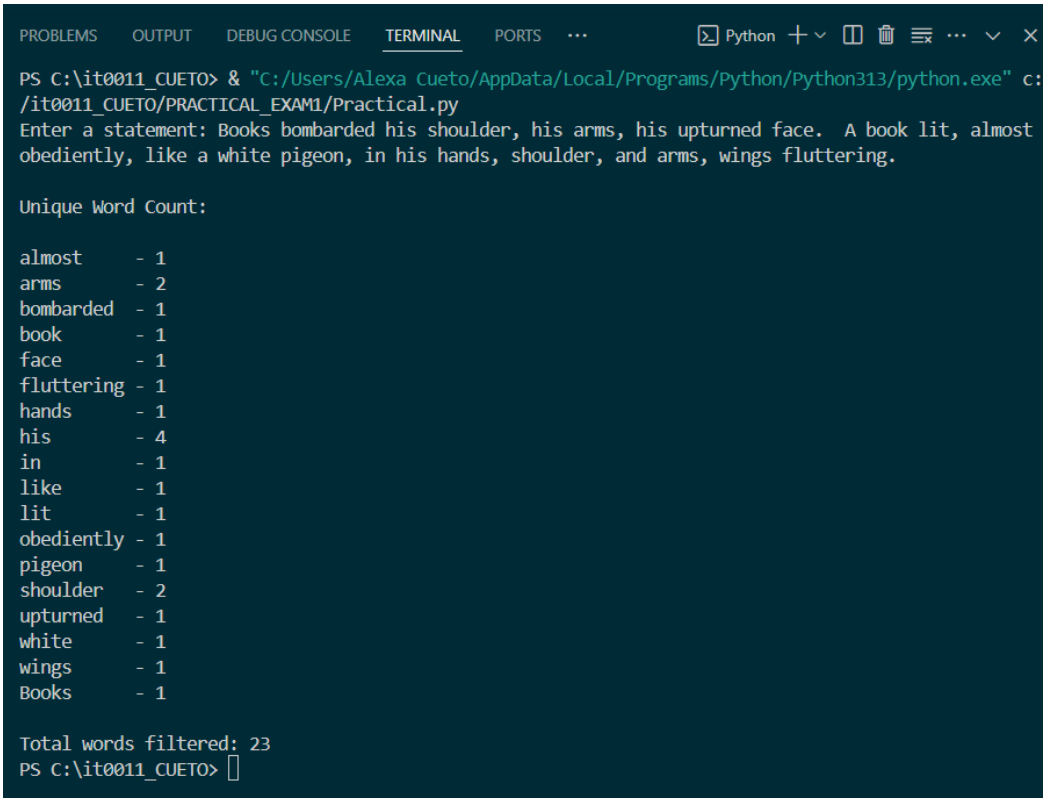
```
#CUETO, ALEXA JOYCE G
#TW23
#PRACTICAL EXAM 1


def uniqueWordCounter(statement):
    excludedWords = ["and", "but", "or", "nor", "for", "so", "yet", "of", "a", "an", "the"]


#Remove punctuation marks
    removedPunctuation = ""
    for char in statement:
        if char.isalnum() or char.isspace(): #Checks whether the character is alphanumeric or a space
            removedPunctuation += char #Join the characters without punctuation marks


    words = removedPunctuation.split() #Split the statement into words
    filteredWords = [word for word in words if word.lower() not in excludedWords] #Filter out the excluded words. Also word.lower() is used to make it case-insensitive


    uniqueWords = {} #Dictionary to store the word count
    for word in filteredWords:
```

```
        uniqueWords[word] = uniqueWords.get(word, 0) + 1 #If the
word is in the dictionary, increment to one, otherwise count as
zero


    # **Sorting Logic Change**
    lowerCaseWords = []
    upperCaseWords = []

    # Classify words into lowercase and uppercase lists
    for word in uniqueWords:
        if word[0].islower(): #The zero index of the word is
checked if it's lowercase
            lowerCaseWords.append(word)
        else:
            upperCaseWords.append(word)

    # Sort each list separately
    lowerCaseWords.sort()
    upperCaseWords.sort()

    # Combine the sorted lists
    allWords = lowerCaseWords + upperCaseWords

    # Display output
    print("\nUnique Word Count:\n")
    for word in allWords:
        print(f"{word:<10} - {uniqueWords[word]}")  # left
formatting
```

```
    print(f"\nTotal words filtered:
{sum(uniqueWords.values())}")


# Main program
statement = input("Enter a statement: ")
uniqueWordCounter(statement)
```

Output:




PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
PS C:\it0011_CUETO> & "C:/Users/Alexa Cueto/AppData/Local/Programs/Python/Python313/python.exe" c:
/it0011_CUETO/PRACTICAL_EXAM1/Practical.py
Enter a statement: Books bombarded his shoulder, his arms, his upturned face. A book lit, almost
obediently, like a white pigeon, in his hands, shoulder, and arms, wings fluttering.
Unique Word Count:
almost - 1
arms - 2
bombarded - 1
book - 1
face - 1
fluttering - 1
hands - 1
his - 4
in - 1
like - 1
lit - 1
obediently - 1
pigeon - 1
shoulder - 2
upturned - 1
white - 1
wings - 1
Books - 1
Total words filtered: 23
PS C:\it0011_CUETO>