Lab2 Anagram Checker

1 Essential Parts

-
  ```
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  ADD R3, R3, #-2
  BRz LOOP1
  ```

  This code is used to determine whether the current character is a space, the ASCII code of the space is 32, so the ASCII code of the character needs to be subtracted by 32, if it is 0, it is a space.Since the ADD instruction supports a maximum of 32, a maximum of 15 can be subtracted each time

-
  ```
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  ADD R3, R3, #-15
  BRnz BIG_TO_SMALL1
  ```

  This piece of code is used to determine case. Since the maximum ASCII code of an uppercase letter is 90, subtract 90 from this character. If it is less than or equal to 0, it is uppercase and needs to be converted to lowercase.

-
  ```
  LD R4, ARY
  ADD R4, R3, R4
  LDR R5, R4, #0
  ADD R5, R5, #1
  STR R5, R4, #0
  ```

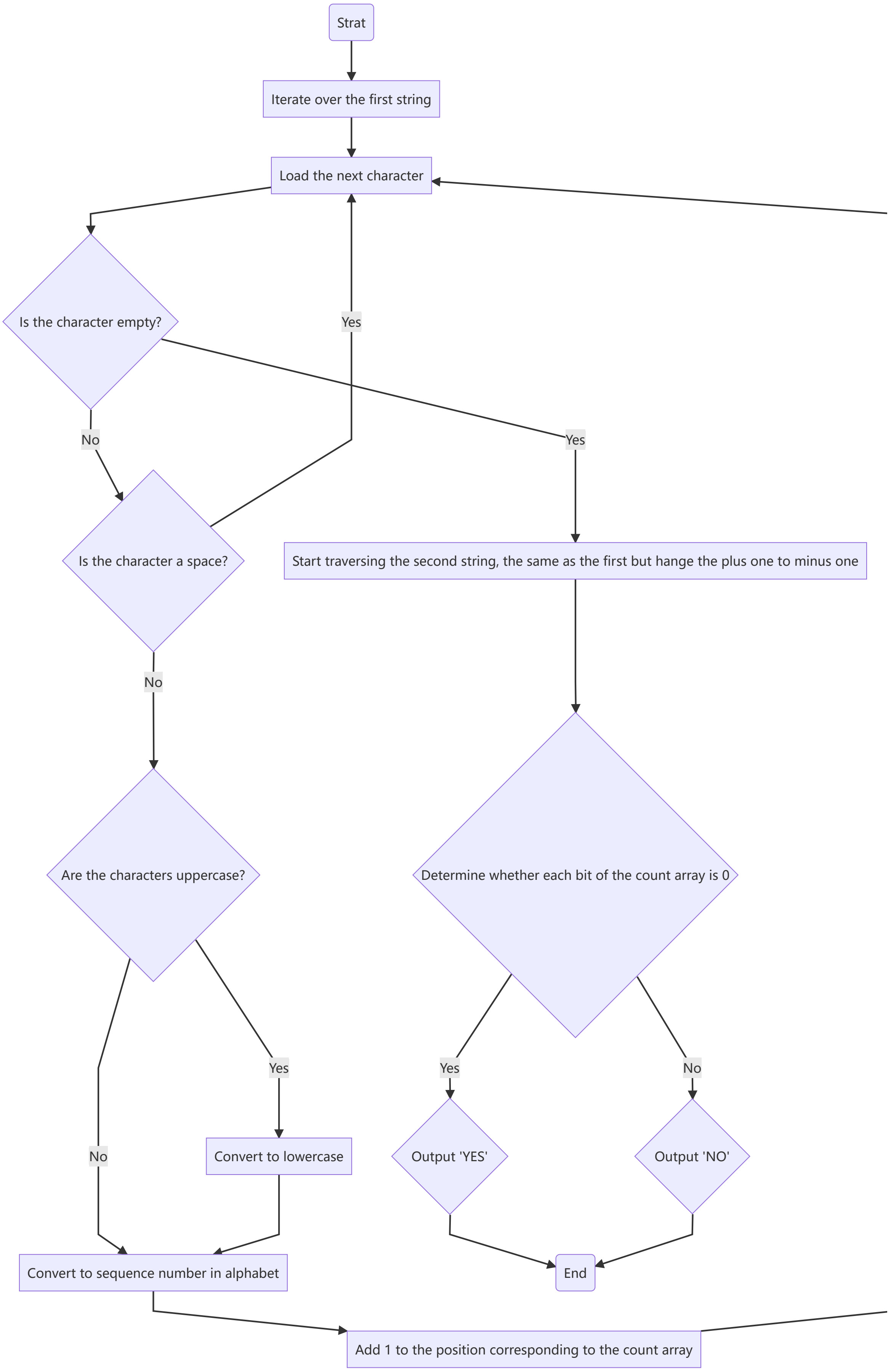
  This piece of code implements the operation of adding one to the corresponding position of the letter in the counting array.

2 Q & A

Q : What is the specific process of adding this letter to the corresponding position of the counting array?

A : The specific process is: add the first address of the counting array to R4, write the character and the address in the corresponding array to R4, and R5 corresponds to the data at the address in R4, add one to R5, and then send it back to the corresponding address.

3 Algorithm


Strat
Iterate over the first string
Load the next character
Is the character empty?
Yes
No
Yes
Is the character a space?
Start traversing the second string, the same as the first but hange the plus one to minus one
No
Are the characters uppercase?
Determine whether each bit of the count array is 0
Yes
Yes
No
No
Convert to lowercase
Output 'YES'
Output 'NO'
Convert to sequence number in alphabet
End
Add 1 to the position corresponding to the count array