

❖ First Non Repeated Character:

Explanation: -

The program identifies the first non-repeated character in a given string using recursion. It first counts the frequency of each character with a helper function, then recursively checks for the first character that appears only once. The result is printed, showing either the first non-repeated character or a message indicating none were found.

Time Complexity: - $O(n)$

Space Complexity: - $O(n)$

[Start]

|

v

[Input: "Enter a String: "] -> str

|

v

[Initialize: frequency[256]]

|

v

[Call: countCharacterFrequency(str, frequency, 0)]

|

v

[Check: index == str.length()] ---> Yes ---> [Return]

| No

v

[Get: currentChar = str.charAt(index)]

|

v

[frequency[currentChar]++]

|

v

[Call: countCharacterFrequency(str, frequency, index + 1)]

|

v

```
[ Call: findFirstNonRepeatedCharacter(str, frequency, 0) ]
|
v
[ Check: index == str.length() ] ---> Yes ---> [ Return '\0' ]
| No
v
[ Get: currentChar = str.charAt(index) ]
|
v
[ Check: frequency[currentChar] == 1 ] ---> Yes ---> [ Return
currentChar ]
| No
v
[ Call: findFirstNonRepeatedCharacter(str, frequency, index + 1) ]
|
v
[ Output: Print first non-repeated character or message ]
|
v
[ End ]
```