

## ❖Repeated Characters:

Explanation: -

The problem involves identifying characters that appear more than once in a given string. The solution requires counting the frequency of each character and then formatting the output to display only the repeated characters. The goal is to present these characters in a specific format, such as [ 'a', 'c', 'r' ].

Time Complexity: -  $O(n)$

Space Complexity: -  $O(n)$

[ Start ]

|

v

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

|

v

[ Initialize: frequency[256] ]

|

v

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

|

v

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

| No

v

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

|

v

[ frequency[currentChar]++ ]

|

v

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

|

v

```
[ Initialize: repeatedChars[256], count = 0 ]
  |
  v
[ Loop through frequency array ]
  |
  v
[ Check: frequency[ch] > 1 ] ---> Yes ---> [ repeatedChars[count++]
= ch ]
  | No
  v
[ Output: Print repeatedChars in format [ 'a', 'c', 'r' ] ]
  |
  v
[ End ]
```