Problem 3120: Count the Number of Special Characters I

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given a string

word

. A letter is called

special

if it appears

both

in lowercase and uppercase in

word

.

Return the number of

special

letters in

word

.

Example 1:

Input:

word = "aaAbcBC"

Output:

3

Explanation:

The special characters in

word

are

'a'

,

'b'

, and

'c'

.

Example 2:

Input:

word = "abc"

Output:

0

Explanation:

No character in

word

appears in uppercase.

Example 3:

Input:

word = "abBCab"

Output:

1

Explanation:

The only special character in

word

is

'b'

.

Constraints:

1 <= word.length <= 50

word

consists of only lowercase and uppercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int numberOfSpecialChars(string word) {

}
};
```

Java:

```
class Solution {
public int numberOfSpecialChars(String word) {

}
}
```

Python3:

```
class Solution:
def numberOfSpecialChars(self, word: str) -> int:
```

Python:

```
class Solution(object):
def numberOfSpecialChars(self, word):
"""
:type word: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} word
* @return {number}
*/
```

```
var numberOfSpecialChars = function(word) {

};
```

TypeScript:

```
function numberOfSpecialChars(word: string): number {

};
```

C#:

```
public class Solution {
public int NumberOfSpecialChars(string word) {

}
}
```

C:

```
int numberOfSpecialChars(char* word) {

}
```

Go:

```
func numberOfSpecialChars(word string) int {

}
```

Kotlin:

```
class Solution {
fun numberOfSpecialChars(word: String): Int {

}
}
```

Swift:

```
class Solution {
func numberOfSpecialChars(_ word: String) -> Int {
}
}
```

Rust:

```
impl Solution {
pub fn number_of_special_chars(word: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} word
# @return {Integer}
def number_of_special_chars(word)

end
```

PHP:

```
class Solution {

/**
* @param String $word
* @return Integer
*/
function numberOfSpecialChars($word) {

}
}
```

Dart:

```
class Solution {
int numberOfSpecialChars(String word) {

}
}
```

Scala:

```
object Solution {
def numberOfSpecialChars(word: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec number_of_special_chars(word :: String.t) :: integer
def number_of_special_chars(word) do

end
end
```

Erlang:

```
-spec number_of_special_chars(Word :: unicode:unicode_binary()) -> integer().
number_of_special_chars(Word) ->
.
```

Racket:

```
(define/contract (number-of-special-chars word)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/
```

```
class Solution {
public:
int numberOfSpecialChars(string word) {

}
};
```

Java Solution:

```
/**
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

class Solution {
public int numberOfSpecialChars(String word) {

}
}
```

Python3 Solution:

```
"""
Problem: Count the Number of Special Characters I
Difficulty: Easy
Tags: string, hash

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(n) for hash map
"""

class Solution:
def numberOfSpecialChars(self, word: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def numberOfSpecialChars(self, word):
"""
:type word: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

/**
* @param {string} word
* @return {number}
*/
var numberOfSpecialChars = function(word) {

};
```

TypeScript Solution:

```
/**
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

function numberOfSpecialChars(word: string): number {
};
```

C# Solution:

```
/*
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

public class Solution {
public int NumberOfSpecialChars(string word) {

}
}
```

C Solution:

```
/*
* Problem: Count the Number of Special Characters I
* Difficulty: Easy
* Tags: string, hash
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(n) for hash map
*/

int numberOfSpecialChars(char* word) {

}
```

Go Solution:

```
// Problem: Count the Number of Special Characters I
// Difficulty: Easy
// Tags: string, hash
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

func numberOfSpecialChars(word string) int {

}
```

Kotlin Solution:

```
class Solution {
fun numberOfSpecialChars(word: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func numberOfSpecialChars(_ word: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Count the Number of Special Characters I
// Difficulty: Easy
// Tags: string, hash
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

impl Solution {
pub fn number_of_special_chars(word: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} word
# @return {Integer}
def number_of_special_chars(word)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $word
* @return Integer
*/
function numberOfSpecialChars($word) {

}
}
```

Dart Solution:

```
class Solution {
int numberOfSpecialChars(String word) {

}
}
```

Scala Solution:

```
object Solution {
def numberOfSpecialChars(word: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec number_of_special_chars(word :: String.t) :: integer
def number_of_special_chars(word) do
```

```
end
end
```

Erlang Solution:

```
-spec number_of_special_chars(Word :: unicode:unicode_binary()) -> integer().
number_of_special_chars(Word) ->
.
```

Racket Solution:

```
(define/contract (number-of-special-chars word)
(-> string? exact-integer?)
)
```