Problem 1154: Day of the Year

Problem Information

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

Problem Description

Given a string

date

representing a

Gregorian calendar

date formatted as

YYYY-MM-DD

, return

the day number of the year

.

Example 1:

Input:

date = "2019-01-09"

Output:

9

Explanation:

Given date is the 9th day of the year in 2019.

Example 2:

Input:

date = "2019-02-10"

Output:

41

Constraints:

date.length == 10

date[4] == date[7] == '-'

, and all other

date[i]

's are digits

date

represents a calendar date between Jan 1

st

, 1900 and Dec 31

st

, 2019.

Code Snippets

C++:

```
class Solution {
public:
int dayOfYear(string date) {

}
};
```

Java:

```
class Solution {
public int dayOfYear(String date) {

}
}
```

Python3:

```
class Solution:
def dayOfYear(self, date: str) -> int:
```

Python:

```
class Solution(object):
def dayOfYear(self, date):
"""
:type date: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} date
* @return {number}
*/
var dayOfYear = function(date) {
```

```
};
```

TypeScript:

```
function dayOfYear(date: string): number {

};
```

C#:

```
public class Solution {
public int DayOfYear(string date) {

}
}
```

C:

```
int dayOfYear(char* date) {

}
```

Go:

```
func dayOfYear(date string) int {

}
```

Kotlin:

```
class Solution {
fun dayOfYear(date: String): Int {

}
}
```

Swift:

```
class Solution {
func dayOfYear(_ date: String) -> Int {

}
```

```
}
```

Rust:

```
impl Solution {
pub fn day_of_year(date: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} date
# @return {Integer}
def day_of_year(date)

end
```

PHP:

```
class Solution {

/**
* @param String $date
* @return Integer
*/
function dayOfYear($date) {

}
}
```

Dart:

```
class Solution {
int dayOfYear(String date) {

}
}
```

Scala:

```
object Solution {
def dayOfYear(date: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec day_of_year(date :: String.t) :: integer
def day_of_year(date) do

end
end
```

Erlang:

```
-spec day_of_year(Date :: unicode:unicode_binary()) -> integer().
day_of_year(Date) ->
.
```

Racket:

```
(define/contract (day-of-year date)
(-> string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/
```

```
class Solution {
public:
int dayOfYear(string date) {

}
};
```

Java Solution:

```
/**
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int dayOfYear(String date) {

}
}
```

Python3 Solution:

```
"""
Problem: Day of the Year
Difficulty: Easy
Tags: string, math

Approach: String manipulation with hash map or two pointers
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

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

Python Solution:

```
class Solution(object):
def dayOfYear(self, date):
"""
:type date: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {string} date
* @return {number}
*/
var dayOfYear = function(date) {

};
```

TypeScript Solution:

```
/**
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function dayOfYear(date: string): number {
```

```
};
```

C# Solution:

```
/*
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int DayOfYear(string date) {

}
}
```

C Solution:

```
/*
* Problem: Day of the Year
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

int dayOfYear(char* date) {

}
```

Go Solution:

```
// Problem: Day of the Year
// Difficulty: Easy
```

```
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func dayOfYear(date string) int {

}
```

Kotlin Solution:

```
class Solution {
fun dayOfYear(date: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func dayOfYear(_ date: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Day of the Year
// Difficulty: Easy
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn day_of_year(date: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} date
# @return {Integer}
def day_of_year(date)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $date
* @return Integer
*/
function dayOfYear($date) {

}
}
```

Dart Solution:

```
class Solution {
int dayOfYear(String date) {

}
}
```

Scala Solution:

```
object Solution {
def dayOfYear(date: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec day_of_year(date :: String.t) :: integer
def day_of_year(date) do
```

```
end
end
```

Erlang Solution:

```
-spec day_of_year(Date :: unicode:unicode_binary()) -> integer().
day_of_year(Date) ->
.
```

Racket Solution:

```
(define/contract (day-of-year date)
(-> string? exact-integer?)
)
```