Problem 1360: Number of Days Between Two Dates

Problem Information

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

Problem Description

Write a program to count the number of days between two dates.

The two dates are given as strings, their format is

YYYY-MM-DD

as shown in the examples.

Example 1:

Input:

date1 = "2019-06-29", date2 = "2019-06-30"

Output:

1

Example 2:

Input:

date1 = "2020-01-15", date2 = "2019-12-31"

Output:

15

Constraints:

The given dates are valid dates between the years

1971

and

2100

.

Code Snippets

C++:

```
class Solution {
public:
int daysBetweenDates(string date1, string date2) {

}
};
```

Java:

```
class Solution {
public int daysBetweenDates(String date1, String date2) {

}
}
```

Python3:

```
class Solution:
def daysBetweenDates(self, date1: str, date2: str) -> int:
```

Python:

```
class Solution(object):
def daysBetweenDates(self, date1, date2):
"""
:type date1: str
:type date2: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} date1
* @param {string} date2
* @return {number}
*/
var daysBetweenDates = function(date1, date2) {

};
```

TypeScript:

```
function daysBetweenDates(date1: string, date2: string): number {

};
```

C#:

```
public class Solution {
public int DaysBetweenDates(string date1, string date2) {

}
}
```

C:

```
int daysBetweenDates(char* date1, char* date2) {

}
```

Go:

```
func daysBetweenDates(date1 string, date2 string) int {
}
```

Kotlin:

```
class Solution {
fun daysBetweenDates(date1: String, date2: String): Int {

}
}
```

Swift:

```
class Solution {
func daysBetweenDates(_ date1: String, _ date2: String) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn days_between_dates(date1: String, date2: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} date1
# @param {String} date2
# @return {Integer}
def days_between_dates(date1, date2)

end
```

PHP:

```
class Solution {

/**
* @param String $date1
* @param String $date2
* @return Integer
*/
function daysBetweenDates($date1, $date2) {

}
}
```

Dart:

```
class Solution {
int daysBetweenDates(String date1, String date2) {

}
}
```

Scala:

```
object Solution {
def daysBetweenDates(date1: String, date2: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec days_between_dates(date1 :: String.t, date2 :: String.t) :: integer
def days_between_dates(date1, date2) do

end
end
```

Erlang:

```
-spec days_between_dates(Date1 :: unicode:unicode_binary(), Date2 ::
unicode:unicode_binary()) -> integer().
days_between_dates(Date1, Date2) ->
.
```

Racket:

```
(define/contract (days-between-dates date1 date2)
(-> string? string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Number of Days Between Two Dates
* 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 daysBetweenDates(string date1, string date2) {

}
};
```

Java Solution:

```
/**
* Problem: Number of Days Between Two Dates
* 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 daysBetweenDates(String date1, String date2) {

}
}
```

Python3 Solution:

```
"""
Problem: Number of Days Between Two Dates
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 daysBetweenDates(self, date1: str, date2: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def daysBetweenDates(self, date1, date2):
"""
:type date1: str
:type date2: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Number of Days Between Two Dates
* 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} date1
* @param {string} date2
* @return {number}
*/
var daysBetweenDates = function(date1, date2) {

};
```

TypeScript Solution:

```
/**
* Problem: Number of Days Between Two Dates
* 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 daysBetweenDates(date1: string, date2: string): number {

};
```

C# Solution:

```
/*
* Problem: Number of Days Between Two Dates
* 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 DaysBetweenDates(string date1, string date2) {

}

}
```

C Solution:

```
/*
* Problem: Number of Days Between Two Dates
* 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 daysBetweenDates(char* date1, char* date2) {

}
```

Go Solution:

```
// Problem: Number of Days Between Two Dates
// 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 daysBetweenDates(date1 string, date2 string) int {

}
```

Kotlin Solution:

```
class Solution {
fun daysBetweenDates(date1: String, date2: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func daysBetweenDates(_ date1: String, _ date2: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Number of Days Between Two Dates
// 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 days_between_dates(date1: String, date2: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} date1
# @param {String} date2
# @return {Integer}
def days_between_dates(date1, date2)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $date1
* @param String $date2
* @return Integer
*/
function daysBetweenDates($date1, $date2) {
}
}
```

Dart Solution:

```
class Solution {
int daysBetweenDates(String date1, String date2) {

}
}
```

Scala Solution:

```
object Solution {
def daysBetweenDates(date1: String, date2: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec days_between_dates(date1 :: String.t, date2 :: String.t) :: integer
def days_between_dates(date1, date2) do

end
end
```

Erlang Solution:

```
-spec days_between_dates(Date1 :: unicode:unicode_binary(), Date2 ::
unicode:unicode_binary()) -> integer().
days_between_dates(Date1, Date2) ->
.
```

Racket Solution:

```
(define/contract (days-between-dates date1 date2)
(-> string? string? exact-integer?)
)
```