Problem 1118: Number of Days in a Month

Problem Information

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

Problem Description

Given a year

year

and a month

month

, return

the number of days of that month

.

Example 1:

Input:

year = 1992, month = 7

Output:

31

Example 2:

Input:

year = 2000, month = 2

Output:

29

Example 3:

Input:

year = 1900, month = 2

Output:

28

Constraints:

1583 <= year <= 2100

1 <= month <= 12

Code Snippets

C++:

```
class Solution {
public:
int numberOfDays(int year, int month) {

}
};
```

Java:

```
class Solution {
public int numberOfDays(int year, int month) {


}
}
```

Python3:

```
class Solution:
def numberOfDays(self, year: int, month: int) -> int:
```

Python:

```
class Solution(object):
def numberOfDays(self, year, month):
"""
:type year: int
:type month: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number} year
* @param {number} month
* @return {number}
*/
var numberOfDays = function(year, month) {

};
```

TypeScript:

```
function numberOfDays(year: number, month: number): number {

};
```

C#:

```
public class Solution {
public int NumberOfDays(int year, int month) {

}

}
```

C:

```
int numberOfDays(int year, int month) {

}
```

Go:

```
func numberOfDays(year int, month int) int {

}
```

Kotlin:

```
class Solution {
fun numberOfDays(year: Int, month: Int): Int {

}
}
```

Swift:

```
class Solution {
func numberOfDays(_ year: Int, _ month: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn number_of_days(year: i32, month: i32) -> i32 {

}
}
```

Ruby:

```
# @param {Integer} year
# @param {Integer} month
# @return {Integer}
def number_of_days(year, month)

end
```

PHP:

```
class Solution {

/**
* @param Integer $year
* @param Integer $month
* @return Integer
*/
function numberOfDays($year, $month) {

}
}
```

Dart:

```
class Solution {
int numberOfDays(int year, int month) {

}
}
```

Scala:

```
object Solution {
def numberOfDays(year: Int, month: Int): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec number_of_days(year :: integer, month :: integer) :: integer
def number_of_days(year, month) do

end
```

```

end
```

Erlang:

```
-spec number_of_days(Year :: integer(), Month :: integer()) -> integer().
number_of_days(Year, Month) ->
.
```

Racket:

```
(define/contract (number-of-days year month)
(-> exact-integer? exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
int numberOfDays(int year, int month) {

}
};
```

Java Solution:

```
/**
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int numberOfDays(int year, int month) {

}
}
```

Python3 Solution:

```
"""
Problem: Number of Days in a Month
Difficulty: Easy
Tags: math

Approach: Optimized algorithm based on problem constraints
Time Complexity: O(n) to O(n^2) depending on approach
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def numberOfDays(self, year: int, month: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def numberOfDays(self, year, month):
"""
:type year: int
:type month: int
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {number} year
* @param {number} month
* @return {number}
*/
var numberOfDays = function(year, month) {

};
```

TypeScript Solution:

```
/**
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

function numberOfDays(year: number, month: number): number {

};
```

C# Solution:

```
/*
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
```

```
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int NumberOfDays(int year, int month) {

}
}
```

C Solution:

```
/*
* Problem: Number of Days in a Month
* Difficulty: Easy
* Tags: math
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

int numberOfDays(int year, int month) {

}
```

Go Solution:

```
// Problem: Number of Days in a Month
// Difficulty: Easy
// Tags: math
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

func numberOfDays(year int, month int) int {

}
```

Kotlin Solution:

```
class Solution {
fun numberOfDays(year: Int, month: Int): Int {

}
}
```

Swift Solution:

```
class Solution {
func numberOfDays(_ year: Int, _ month: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Number of Days in a Month
// Difficulty: Easy
// Tags: math
//
// Approach: Optimized algorithm based on problem constraints
// Time Complexity: O(n) to O(n^2) depending on approach
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn number_of_days(year: i32, month: i32) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer} year
# @param {Integer} month
# @return {Integer}
def number_of_days(year, month)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $year
* @param Integer $month
* @return Integer
*/
function numberOfDays($year, $month) {

}
}
```

Dart Solution:

```
class Solution {
int numberOfDays(int year, int month) {

}
}
```

Scala Solution:

```
object Solution {
def numberOfDays(year: Int, month: Int): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec number_of_days(year :: integer, month :: integer) :: integer
def number_of_days(year, month) do

end
end
```

Erlang Solution:

```
-spec number_of_days(Year :: integer(), Month :: integer()) -> integer().
number_of_days(Year, Month) ->
.
```

Racket Solution:

```
(define/contract (number-of-days year month)
(-> exact-integer? exact-integer? exact-integer?)
)
```