Problem 3280: Convert Date to Binary

Problem Information

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

Problem Description

You are given a string

date

representing a Gregorian calendar date in the

yyyy-mm-dd

format.

date

can be written in its binary representation obtained by converting year, month, and day to their binary representations without any leading zeroes and writing them down in

year-month-day

format.

Return the

binary

representation of

date

.

Example 1:

Input:

date = "2080-02-29"

Output:

"100000100000-10-11101"

Explanation:

100000100000, 10, and 11101 are the binary representations of 2080, 02, and 29 respectively.

Example 2:

Input:

date = "1900-01-01"

Output:

"11101101100-1-1"

Explanation:

11101101100, 1, and 1 are the binary representations of 1900, 1, and 1 respectively.

Constraints:

date.length == 10

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

, and all other

date[i]

's are digits.

The input is generated such that

date

represents a valid Gregorian calendar date between Jan 1

st

, 1900 and Dec 31

st

, 2100 (both inclusive).

Code Snippets

C++:

```
class Solution {
public:
string convertDateToBinary(string date) {

}
};
```

Java:

```
class Solution {
public String convertDateToBinary(String date) {

}
}
```

Python3:

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

Python:

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

JavaScript:

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

};
```

TypeScript:

```
function convertDateToBinary(date: string): string {

};
```

C#:

```
public class Solution {
public string ConvertDateToBinary(string date) {

}
}
```

C:

```
char* convertDateToBinary(char* date) {

}
```

Go:

```
func convertDateToBinary(date string) string {

}
```

Kotlin:

```
class Solution {
fun convertDateToBinary(date: String): String {

}
}
```

Swift:

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

}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

```
class Solution {
String convertDateToBinary(String date) {

}
}
```

Scala:

```
object Solution {
def convertDateToBinary(date: String): String = {

}
}
```

Elixir:

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

end
end
```

Erlang:

```
-spec convert_date_to_binary(Date :: unicode:unicode_binary()) ->
unicode:unicode_binary().
convert_date_to_binary(Date) ->
.
```

Racket:

```
(define/contract (convert-date-to-binary date)
(-> string? string?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Convert Date to Binary
* 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:
string convertDateToBinary(string date) {

}
};
```

Java Solution:

```
/**
* Problem: Convert Date to Binary
* 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 String convertDateToBinary(String date) {

}
}
```

Python3 Solution:

```
"""
Problem: Convert Date to Binary
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 convertDateToBinary(self, date: str) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Convert Date to Binary
* 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 {string}
*/
```

```
var convertDateToBinary = function(date) {

};
```

TypeScript Solution:

```
/**
* Problem: Convert Date to Binary
* 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 convertDateToBinary(date: string): string {

};
```

C# Solution:

```
/*
* Problem: Convert Date to Binary
* 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 string ConvertDateToBinary(string date) {

}
}
```

C Solution:

```
/*
* Problem: Convert Date to Binary
* 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
*/

char* convertDateToBinary(char* date) {

}
```

Go Solution:

```
// Problem: Convert Date to Binary
// 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 convertDateToBinary(date string) string {

}
```

Kotlin Solution:

```
class Solution {
fun convertDateToBinary(date: String): String {

}
}
```

Swift Solution:

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

}

}
```

Rust Solution:

```
// Problem: Convert Date to Binary
// 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 convert_date_to_binary(date: String) -> String {

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

```
class Solution {
String convertDateToBinary(String date) {

}
}
```

Scala Solution:

```
object Solution {
def convertDateToBinary(date: String): String = {

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

```
-spec convert_date_to_binary(Date :: unicode:unicode_binary()) ->
unicode:unicode_binary().
convert_date_to_binary(Date) ->
.
```

Racket Solution:

```
(define/contract (convert-date-to-binary date)
(-> string? string?)
)
```