Problem 504: Base 7

Problem Information

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

Problem Description

Given an integer

num

, return

a string of its

base 7

representation

.

Example 1:

Input:

num = 100

Output:

"202"

Example 2:

Input:

num = -7

Output:

"-10"

Constraints:

-10

7

<= num <= 10

7

Code Snippets

C++:

```
class Solution {
public:
string convertToBase7(int num) {

}
};
```

Java:

```
class Solution {
public String convertToBase7(int num) {

}
}
```

Python3:

```
class Solution:
def convertToBase7(self, num: int) -> str:
```

Python:

```
class Solution(object):
def convertToBase7(self, num):
"""
:type num: int
:rtype: str
"""
```

JavaScript:

```
/**
* @param {number} num
* @return {string}
*/
var convertToBase7 = function(num) {

};
```

TypeScript:

```
function convertToBase7(num: number): string {

};
```

C#:

```
public class Solution {
public string ConvertToBase7(int num) {

}
}
```

C:

```
char* convertToBase7(int num) {

}
```

Go:

```
func convertToBase7(num int) string {

}
```

Kotlin:

```
class Solution {
fun convertToBase7(num: Int): String {

}
}
```

Swift:

```
class Solution {
func convertToBase7(_ num: Int) -> String {

}
}
```

Rust:

```
impl Solution {
pub fn convert_to_base7(num: i32) -> String {

}
}
```

Ruby:

```
# @param {Integer} num
# @return {String}
def convert_to_base7(num)

end
```

PHP:

```
class Solution {

/**
* @param Integer $num
* @return String
*/
function convertToBase7($num) {

}
}
```

Dart:

```
class Solution {
String convertToBase7(int num) {

}
}
```

Scala:

```
object Solution {
def convertToBase7(num: Int): String = {

}
}
```

Elixir:

```
defmodule Solution do
@spec convert_to_base7(num :: integer) :: String.t
def convert_to_base7(num) do

end
end
```

Erlang:

```
-spec convert_to_base7(Num :: integer()) -> unicode:unicode_binary().
convert_to_base7(Num) ->
.
```

Racket:

```
(define/contract (convert-to-base7 num)
(-> exact-integer? string?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Base 7
* 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 convertToBase7(int num) {

}
};
```

Java Solution:

```
/**
* Problem: Base 7
* 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 convertToBase7(int num) {

}
}
```

Python3 Solution:

```
"""
Problem: Base 7
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 convertToBase7(self, num: int) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def convertToBase7(self, num):
"""
:type num: int
:rtype: str
"""
```

JavaScript Solution:

```
/**
* Problem: Base 7
* 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 {number} num
* @return {string}
*/
var convertToBase7 = function(num) {
};
```

TypeScript Solution:

```
/**
* Problem: Base 7
* 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 convertToBase7(num: number): string {

};
```

C# Solution:

```
/*
* Problem: Base 7
* 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 ConvertToBase7(int num) {

}
}
```

C Solution:

```
/*
* Problem: Base 7
* 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* convertToBase7(int num) {

}
```

Go Solution:

```
// Problem: Base 7
// 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 convertToBase7(num int) string {

}
```

Kotlin Solution:

```
class Solution {
fun convertToBase7(num: Int): String {

}
}
```

Swift Solution:

```
class Solution {
func convertToBase7(_ num: Int) -> String {

}
}
```

Rust Solution:

```
// Problem: Base 7
// 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_to_base7(num: i32) -> String {

}
}
```

Ruby Solution:

```
# @param {Integer} num
# @return {String}
def convert_to_base7(num)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $num
* @return String
*/
function convertToBase7($num) {

}
}
```

Dart Solution:

```
class Solution {
String convertToBase7(int num) {

}
}
```

Scala Solution:

```
object Solution {
def convertToBase7(num: Int): String = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec convert_to_base7(num :: integer) :: String.t
def convert_to_base7(num) do

end
end
```

Erlang Solution:

```
-spec convert_to_base7(Num :: integer()) -> unicode:unicode_binary().
convert_to_base7(Num) ->
.
```

Racket Solution:

```
(define/contract (convert-to-base7 num)
(-> exact-integer? string?)
)
```