Problem 405: Convert a Number to Hexadecimal

Problem Information

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

Problem Description

Given a 32-bit integer

num

, return

a string representing its hexadecimal representation

. For negative integers,

two’s complement

method is used.

All the letters in the answer string should be lowercase characters, and there should not be any leading zeros in the answer except for the zero itself.

Note:

You are not allowed to use any built-in library method to directly solve this problem.

Example 1:

Input:

num = 26

Output:

"1a"

Example 2:

Input:

num = -1

Output:

"ffffffff"

Constraints:

-2

31

<= num <= 2

31

- 1

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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

Python:

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

JavaScript:

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

};
```

TypeScript:

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

};
```

C#:

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

}
}
```

C:

```
char* toHex(int num) {

}
```

Go:

```
func toHex(num int) string {

}
```

Kotlin:

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

}
}
```

Swift:

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

}
}
```

Rust:

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

}
}
```

Ruby:

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

end
```

PHP:

```
class Solution {

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

}
}
```

Dart:

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

}
}
```

Scala:

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

}
}
```

Elixir:

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

end
end
```

Erlang:

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

Racket:

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

Solutions

C++ Solution:

```
/*
* Problem: Convert a Number to Hexadecimal
* 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 toHex(int num) {

}
};
```

Java Solution:

```
/**
* Problem: Convert a Number to Hexadecimal
* 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 toHex(int num) {
}
}
```

Python3 Solution:

```
"""
Problem: Convert a Number to Hexadecimal
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 toHex(self, num: int) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

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

JavaScript Solution:

```
/**
* Problem: Convert a Number to Hexadecimal
* 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 toHex = function(num) {

};
```

TypeScript Solution:

```
/**
* Problem: Convert a Number to Hexadecimal
* 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 toHex(num: number): string {

};
```

C# Solution:

```
/*
* Problem: Convert a Number to Hexadecimal
* 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 ToHex(int num) {

}
}
```

C Solution:

```
/*
* Problem: Convert a Number to Hexadecimal
* 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* toHex(int num) {

}
```

Go Solution:

```
// Problem: Convert a Number to Hexadecimal
// 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 toHex(num int) string {

}
```

Kotlin Solution:

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

}
}
```

Swift Solution:

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

}
}
```

Rust Solution:

```
// Problem: Convert a Number to Hexadecimal
// 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 to_hex(num: i32) -> String {

}
}
```

Ruby Solution:

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

end
```

PHP Solution:

```
class Solution {

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

}
}
```

Dart Solution:

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

}
}
```

Scala Solution:

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

}
}
```

Elixir Solution:

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

end
end
```

Erlang Solution:

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

Racket Solution:

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