Problem 3726: Remove Zeros in Decimal Representation

Problem Information

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

Problem Description

You are given a

positive

integer

n

.

Return the integer obtained by removing all zeros from the decimal representation of

n

.

Example 1:

Input:

n = 1020030

Output:

123

Explanation:

After removing all zeros from 1

0

2

00

3

0

, we get 123.

Example 2:

Input:

n = 1

Output:

1

Explanation:

1 has no zero in its decimal representation. Therefore, the answer is 1.

Constraints:

1 <= n <= 10

15

Code Snippets

C++:

```
class Solution {
public:
long long removeZeros(long long n) {

}
};
```

Java:

```
class Solution {
public long removeZeros(long n) {

}
}
```

Python3:

```
class Solution:
def removeZeros(self, n: int) -> int:
```

Python:

```
class Solution(object):
def removeZeros(self, n):
"""
:type n: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number} n
* @return {number}
*/
var removeZeros = function(n) {

};
```

TypeScript:

```
function removeZeros(n: number): number {

};
```

C#:

```
public class Solution {
public long RemoveZeros(long n) {

}
}
```

C:

```
long long removeZeros(long long n) {

}
```

Go:

```
func removeZeros(n int64) int64 {

}
```

Kotlin:

```
class Solution {
fun removeZeros(n: Long): Long {

}
}
```

Swift:

```
class Solution {
func removeZeros(_ n: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn remove_zeros(n: i64) -> i64 {

}
}
```

Ruby:

```
# @param {Integer} n
# @return {Integer}
def remove_zeros(n)

end
```

PHP:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function removeZeros($n) {

}
}
```

Dart:

```
class Solution {
int removeZeros(int n) {

}
}
```

Scala:

```
object Solution {
def removeZeros(n: Long): Long = {

}
}
```

Elixir:

```
defmodule Solution do
@spec remove_zeros(n :: integer) :: integer
def remove_zeros(n) do

end
end
```

Erlang:

```
-spec remove_zeros(N :: integer()) -> integer().
remove_zeros(N) ->
.
```

Racket:

```
(define/contract (remove-zeros n)
(-> exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Remove Zeros in Decimal Representation
* 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:
long long removeZeros(long long n) {

}
};
```

Java Solution:

```
/**
* Problem: Remove Zeros in Decimal Representation
* 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 long removeZeros(long n) {

}
}
```

Python3 Solution:

```
"""
Problem: Remove Zeros in Decimal Representation
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 removeZeros(self, n: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def removeZeros(self, n):
"""
:type n: int
:rtype: int
```

```

"""
```

JavaScript Solution:

```
/**
* Problem: Remove Zeros in Decimal Representation
* 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} n
* @return {number}
*/
var removeZeros = function(n) {

};
```

TypeScript Solution:

```
/**
* Problem: Remove Zeros in Decimal Representation
* 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 removeZeros(n: number): number {

};
```

C# Solution:

```
/*
* Problem: Remove Zeros in Decimal Representation
* 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 long RemoveZeros(long n) {

}
}
```

C Solution:

```
/*
* Problem: Remove Zeros in Decimal Representation
* 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
*/

long long removeZeros(long long n) {

}
```

Go Solution:

```
// Problem: Remove Zeros in Decimal Representation
// 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 removeZeros(n int64) int64 {

}
```

Kotlin Solution:

```
class Solution {
fun removeZeros(n: Long): Long {

}
}
```

Swift Solution:

```
class Solution {
func removeZeros(_ n: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Remove Zeros in Decimal Representation
// 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 remove_zeros(n: i64) -> i64 {

}
}
```

Ruby Solution:

```
# @param {Integer} n
# @return {Integer}
def remove_zeros(n)
```

```
end
```

PHP Solution:

```
class Solution {

/**
* @param Integer $n
* @return Integer
*/
function removeZeros($n) {

}
}
```

Dart Solution:

```
class Solution {
int removeZeros(int n) {

}
}
```

Scala Solution:

```
object Solution {
def removeZeros(n: Long): Long = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec remove_zeros(n :: integer) :: integer
def remove_zeros(n) do

end
end
```

Erlang Solution:

```
-spec remove_zeros(N :: integer()) -> integer().
remove_zeros(N) ->
.
```

Racket Solution:

```
(define/contract (remove-zeros n)
(-> exact-integer? exact-integer?)
)
```