

Problem 3099: Harshad Number

Problem Information

Difficulty: [Easy](#)

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

An integer divisible by the

sum

of its digits is said to be a

Harshad

number. You are given an integer

x

. Return

the sum of the digits

of

x

if

x

is a

Harshad

number, otherwise, return

-1

.

Example 1:

Input:

$x = 18$

Output:

9

Explanation:

The sum of digits of

x

is

9

.

18

is divisible by

9

. So

18

is a Harshad number and the answer is

9

.

Example 2:

Input:

$x = 23$

Output:

-1

Explanation:

The sum of digits of

x

is

5

.

23

is not divisible by

5

. So

23

is not a Harshad number and the answer is

-1

.

Constraints:

$1 \leq x \leq 100$

Code Snippets

C++:

```
class Solution {  
public:  
    int sumOfTheDigitsOfHarshadNumber(int x) {  
  
    }  
};
```

Java:

```
class Solution {  
public int sumOfTheDigitsOfHarshadNumber(int x) {  
  
}  
}
```

Python3:

```
class Solution:  
    def sumOfTheDigitsOfHarshadNumber(self, x: int) -> int:
```

Python:

```
class Solution(object):  
    def sumOfTheDigitsOfHarshadNumber(self, x):  
        """  
        :type x: int
```

```
:rtype: int  
"""
```

JavaScript:

```
/**  
 * @param {number} x  
 * @return {number}  
 */  
var sumOfTheDigitsOfHarshadNumber = function(x) {  
  
};
```

TypeScript:

```
function sumOfTheDigitsOfHarshadNumber(x: number): number {  
  
};
```

C#:

```
public class Solution {  
    public int SumOfTheDigitsOfHarshadNumber(int x) {  
  
    }  
}
```

C:

```
int sumOfTheDigitsOfHarshadNumber(int x) {  
  
}
```

Go:

```
func sumOfTheDigitsOfHarshadNumber(x int) int {  
  
}
```

Kotlin:

```
class Solution {  
    fun sumOfTheDigitsOfHarshadNumber(x: Int): Int {  
        }  
    }  
}
```

Swift:

```
class Solution {  
    func sumOfTheDigitsOfHarshadNumber(_ x: Int) -> Int {  
        }  
    }  
}
```

Rust:

```
impl Solution {  
    pub fn sum_of_the_digits_of_harshad_number(x: i32) -> i32 {  
        }  
    }  
}
```

Ruby:

```
# @param {Integer} x  
# @return {Integer}  
def sum_of_the_digits_of_harshad_number(x)  
end
```

PHP:

```
class Solution {  
  
    /**  
     * @param Integer $x  
     * @return Integer  
     */  
    function sumOfTheDigitsOfHarshadNumber($x) {  
  
    }  
}
```

Dart:

```
class Solution {  
    int sumOfTheDigitsOfHarshadNumber(int x) {  
  
    }  
}
```

Scala:

```
object Solution {  
    def sumOfTheDigitsOfHarshadNumber(x: Int): Int = {  
  
    }  
}
```

Elixir:

```
defmodule Solution do  
  @spec sum_of_the_digits_of_harshad_number(x :: integer) :: integer  
  def sum_of_the_digits_of_harshad_number(x) do  
  
  end  
end
```

Erlang:

```
-spec sum_of_the_digits_of_harshad_number(x :: integer()) -> integer().  
sum_of_the_digits_of_harshad_number(X) ->  
.
```

Racket:

```
(define/contract (sum-of-the-digits-of-harshad-number x)  
  (-> exact-integer? exact-integer?)  
)
```

Solutions

C++ Solution:

```

/*
 * Problem: Harshad Number
 * 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 sumOfTheDigitsOfHarshadNumber(int x) {

    }
};

```

Java Solution:

```

/**
 * Problem: Harshad Number
 * 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 sumOfTheDigitsOfHarshadNumber(int x) {

}
}

```

Python3 Solution:

```

"""
Problem: Harshad Number
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 sumOfTheDigitsOfHarshadNumber(self, x: int) -> int:
# TODO: Implement optimized solution
pass

```

Python Solution:

```

class Solution(object):

def sumOfTheDigitsOfHarshadNumber(self, x):
"""

:type x: int
:rtype: int

"""

```

JavaScript Solution:

```

/**
 * Problem: Harshad Number
 * 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} x
 * @return {number}
 */
var sumOfTheDigitsOfHarshadNumber = function(x) {

};


```

TypeScript Solution:

```

/**
 * Problem: Harshad Number
 * 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 sumOfTheDigitsOfHarshadNumber(x: number): number {
}

```

C# Solution:

```

/*
 * Problem: Harshad Number
 * 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 SumOfTheDigitsOfHarshadNumber(int x) {
        return 0;
    }
}

```

C Solution:

```

/*
 * Problem: Harshad Number
 * 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 sumOfTheDigitsOfHarshadNumber(int x) {  
  
}
```

Go Solution:

```
// Problem: Harshad Number  
// 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 sumOfTheDigitsOfHarshadNumber(x int) int {  
  
}
```

Kotlin Solution:

```
class Solution {  
    fun sumOfTheDigitsOfHarshadNumber(x: Int): Int {  
  
    }  
}
```

Swift Solution:

```
class Solution {  
    func sumOfTheDigitsOfHarshadNumber(_ x: Int) -> Int {  
  
    }  
}
```

Rust Solution:

```
// Problem: Harshad Number  
// 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 sum_of_the_digits_of_harshad_number(x: i32) -> i32 {

}
}

```

Ruby Solution:

```

# @param {Integer} x
# @return {Integer}
def sum_of_the_digits_of_harshad_number(x)

end

```

PHP Solution:

```

class Solution {

/**
 * @param Integer $x
 * @return Integer
 */
function sumOfTheDigitsOfHarshadNumber($x) {

}
}

```

Dart Solution:

```

class Solution {
int sumOfTheDigitsOfHarshadNumber(int x) {

}
}

```

Scala Solution:

```
object Solution {  
    def sumOfTheDigitsOfHarshadNumber(x: Int): Int = {  
        }  
    }  
}
```

Elixir Solution:

```
defmodule Solution do  
  @spec sum_of_the_digits_of_harshad_number(x :: integer) :: integer  
  def sum_of_the_digits_of_harshad_number(x) do  
  
  end  
  end
```

Erlang Solution:

```
-spec sum_of_the_digits_of_harshad_number(x :: integer()) -> integer().  
sum_of_the_digits_of_harshad_number(X) ->  
.
```

Racket Solution:

```
(define/contract (sum-of-the-digits-of-harshad-number x)  
  (-> exact-integer? exact-integer?)  
)
```