

Is a number a palindrome (without stringify):

const inputNumber = 132456;

const isPalindrome = num => {

  let reversedNumber = 0;

  const originalNumber = num;

  while(num > 0) {

    let rightDigit = Math.floor(num % 10);

    console.log('First floor: ' + rightDigit)

    reversedNumber = reversedNumber \* 10 + rightDigit;

    num = Math.floor(num / 10);

    console.log('New num: ' + num)

  }

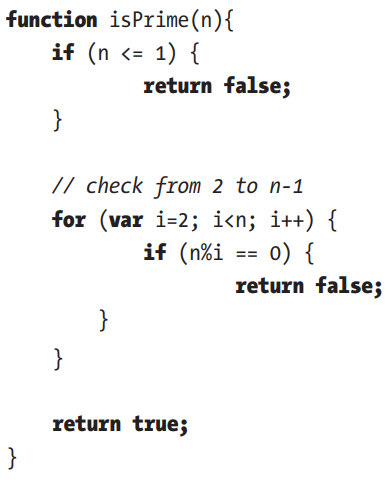
  return reversedNumber === originalNumber;

}

console.log(isPalindrome(inputNumber))

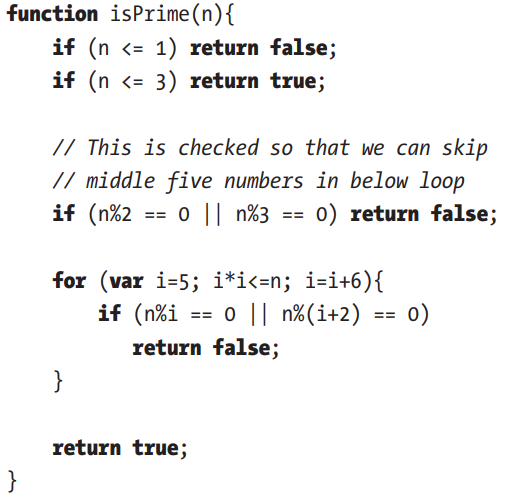
**Test if a number is prime:**

* O(n) is where we:



Improve:

* All prime numbers follow this formula: 6\*k + 1  
  **7** = 6\*1+1, **13**=6\*2+1  
  2 and 3 are exceptions to this rule  
    
  We only need to test until the square root (**if the square root of n is not a prime number, then n is not a prime number, by definition**)

  
O(sqrt(n))