**Test2**

**1.**

function mySqrt(x) {

if (x === 0) return 0; // Special case for x = 0

let prev = 0;

let curr = 1;

while (curr !== prev) {

prev = curr;

curr = (curr + x / curr) / 2;

}

return Math.floor(curr);

}

**Example1:**

const x = 4;

const result = mySqrt(x);

console.log(result);

Output: 2

**Example2:**

const x = 8;

const result = mySqrt(x);

console.log(result);

Output: 2

2.

class ListNode {

constructor(val) {

this.val = val;

this.next = null;

}

}

function addTwoNumbers(l1, l2) {

let dummyHead = new ListNode(0);

let current = dummyHead;

let carry = 0;

while (l1 !== null || l2 !== null) {

let sum = carry;

if (l1 !== null) {

sum += l1.val;

l1 = l1.next;

}

if (l2 !== null) {

sum += l2.val;

l2 = l2.next;

}

carry = Math.floor(sum / 10);

current.next = new ListNode(sum % 10);

current = current.next;

}

if (carry > 0) {

current.next = new ListNode(carry);

}

return dummyHead.next;

}

// Example 1

let l1 = new ListNode(2);

l1.next = new ListNode(4);

l1.next.next = new ListNode(3);

let l2 = new ListNode(5);

l2.next = new ListNode(6);

l2.next.next = new ListNode(4);

let result = addTwoNumbers(l1, l2);

console.log(result);

// Output: [7,0,8]

// Example 2

l1 = new ListNode(0);

l2 = new ListNode(0);

result = addTwoNumbers(l1, l2);

console.log(result);

// Output: [0]

// Example 3

l1 = new ListNode(9);

l1.next = new ListNode(9);

l1.next.next = new ListNode(9);

l1.next.next.next = new ListNode(9);

l1.next.next.next.next = new ListNode(9);

l1.next.next.next.next.next = new ListNode(9);

l1.next.next.next.next.next.next = new ListNode(9);

l2 = new ListNode(9);

l2.next = new ListNode(9);

l2.next.next = new ListNode(9);

l2.next.next.next = new ListNode(9);

result = addTwoNumbers(l1, l2);

console.log(result);

// Output: [8,9,9,9,0,0,0,1]