Q1.public class SquareRoot {

public static int mySqrt(int x) {

if (x == 0) {

return 0;

}

long left = 1;

long right = x;

while (left <= right) {

long mid = left + (right - left) / 2;

if (mid \* mid == x) {

return (int) mid;

} else if (mid \* mid > x) {

right = mid - 1;

} else {

left = mid + 1;

}

}

return (int) right;

}

public static void main(String[] args) {

System.out.println(mySqrt(4)); // Output: 2

System.out.println(mySqrt(8)); // Output: 2

}

}

Q2class ListNode {

int val;

ListNode next;

ListNode(int val) {

this.val = val;

}

}

public class AddTwoNumbers {

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

ListNode dummyHead = new ListNode(0);

ListNode current = dummyHead;

int carry = 0;

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

int sum = carry;

if (l1 != null) {

sum += l1.val;

l1 = l1.next;

}

if (l2 != null) {

sum += l2.val;

l2 = l2.next;

}

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

current = current.next;

carry = sum / 10;

}

if (carry > 0) {

current.next = new ListNode(carry);

}

return dummyHead.next;

}

public static void main(String[] args) {

// Create the first number: 2 -> 4 -> 3

ListNode l1 = new ListNode(2);

l1.next = new ListNode(4);

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

// Create the second number: 5 -> 6 -> 4

ListNode l2 = new ListNode(5);

l2.next = new ListNode(6);

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

AddTwoNumbers solution = new AddTwoNumbers();

ListNode sum = solution.addTwoNumbers(l1, l2);

// Print the sum: 7 -> 0 -> 8

while (sum != null) {

System.out.print(sum.val + " ");

sum = sum.next;

}

}

}