**Lab-5**

Q1. Write a Java program that reads a string from the user and uses StringTokenizer to split the string into individual words. Print each word on a new line.

**Program**

package demo;

import java.util.Scanner;

import java.util.StringTokenizer;

public class StringSplitter {

public static void main(String[] args) {

// Create a Scanner object to read input from the user

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string:");

String input = scanner.nextLine();

// Create a StringTokenizer object to split the string

StringTokenizer tokenizer = new StringTokenizer(input);

// Print each word on a new line

while (tokenizer.hasMoreTokens()) {

System.out.println(tokenizer.nextToken());

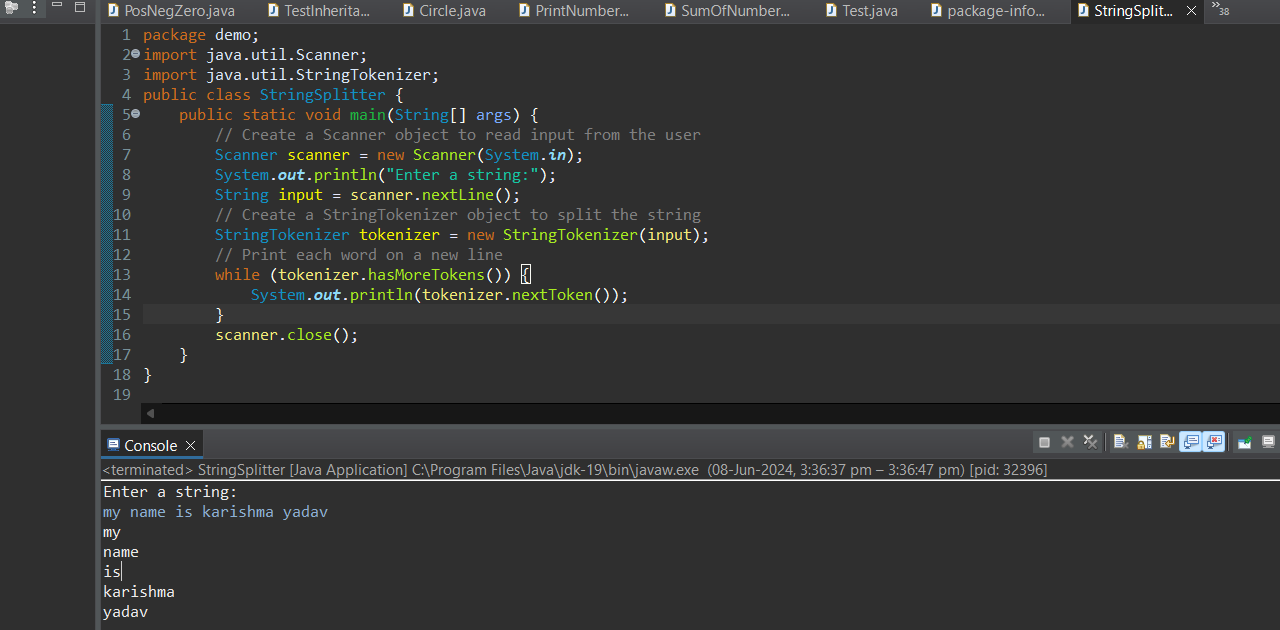
}

scanner.close();

}

}

**Output**



Q2.     Write a Java program that reads a string from the user and uses StringTokenizer to count the number of words in the string.

**Program**

package demo;

import java.util.Scanner;

import java.util.StringTokenizer;

public class WordCounter {

public static void main(String[] args) {

// Create a Scanner object to read input from the user

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string:");

String input = scanner.nextLine();

// Create a StringTokenizer object to split the string

StringTokenizer tokenizer = new StringTokenizer(input);

// Initialize a counter for the words

int wordCount = 0;

// Count the number of words

while (tokenizer.hasMoreTokens()) {

tokenizer.nextToken();

wordCount++;

}

// Print the word count

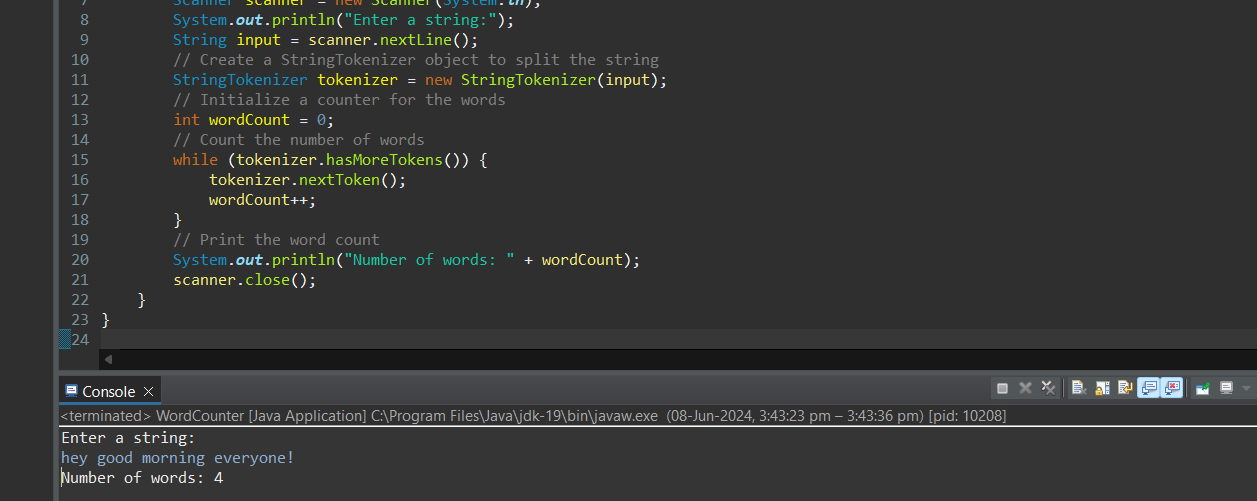
System.out.println("Number of words: " + wordCount);

scanner.close();

}

}

**Output**



Q3.     Write a Java program to create a LinkedList of strings, add elements at specific positions (beginning, middle, end), and print the list.

**Program**

package demo;

import java.util.LinkedList;

public class LinkedListExample {

public static void main(String[] args) {

// Create a LinkedList of strings

LinkedList<String> list = new LinkedList<>();

// Add elements to the LinkedList

list.add("Middle");

list.addFirst("Beginning");

list.addLast("End");

// Add an element in the middle (in this case, after the first element)

list.add(2, "New Middle");

System.out.println("LinkedList elements:");

for (String element : list) {

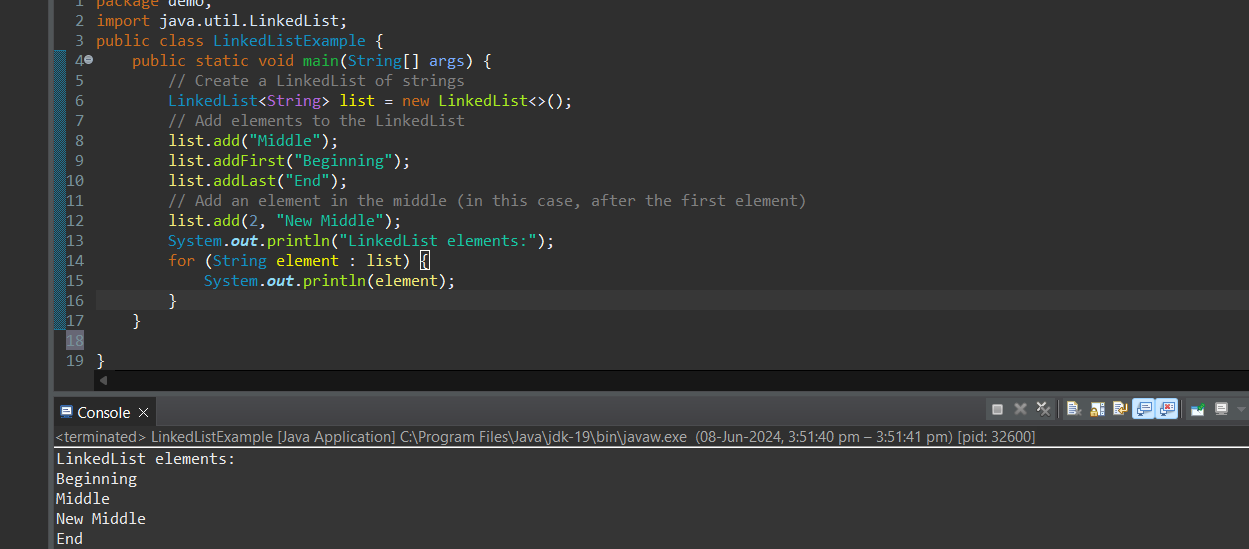
System.out.println(element);

}

}

}

**Output**



Q4.     Write a Java program to sort a given array list.

**Program**

package demo;

import java.util.ArrayList;

import java.util.Collections;

public class SortArrayList {

public static void main(String[] args) {

// Create an ArrayList of strings

ArrayList<String> list = new ArrayList<>();

// Add elements to the ArrayList

list.add("Banana");

list.add("Apple");

list.add("Orange");

list.add("Mango");

list.add("Pineapple");

// Print the original list

System.out.println("Original list:");

for (String fruit : list) {

System.out.println(fruit);

}

// Sort the ArrayList

Collections.sort(list);

// Print the sorted list

System.out.println("\nSorted list:");

for (String fruit : list) {

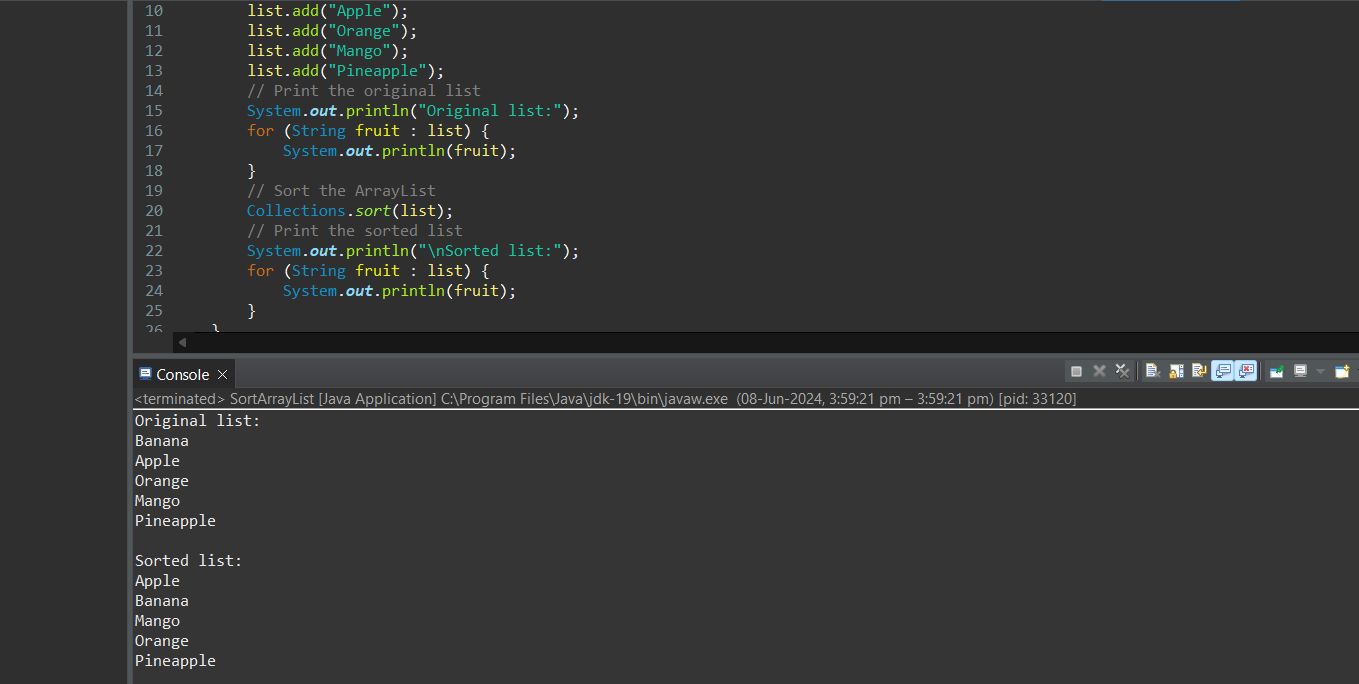
System.out.println(fruit);

}

}

}

**Output**



Q5.     Write a Java program to replace the second element of an ArrayList with the specified element.

**Program**

package demo;

import java.util.ArrayList;

public class ReplaceElement {

public static void main(String[] args) {

// Create an ArrayList of strings

ArrayList<String> list = new ArrayList<>();

// Add elements to the ArrayList

list.add("Apple");

list.add("Banana");

list.add("Orange");

list.add("Mango");

list.add("Pineapple");

// Print the original list

System.out.println("Original list:");

for (String fruit : list) {

System.out.println(fruit);

}

// Specify the new element to replace the second element

String newElement = "Grapes";

// Replace the second element (index 1) with the specified element

list.set(1, newElement);

// Print the modified list

System.out.println("\nModified list:");

for (String fruit : list) {

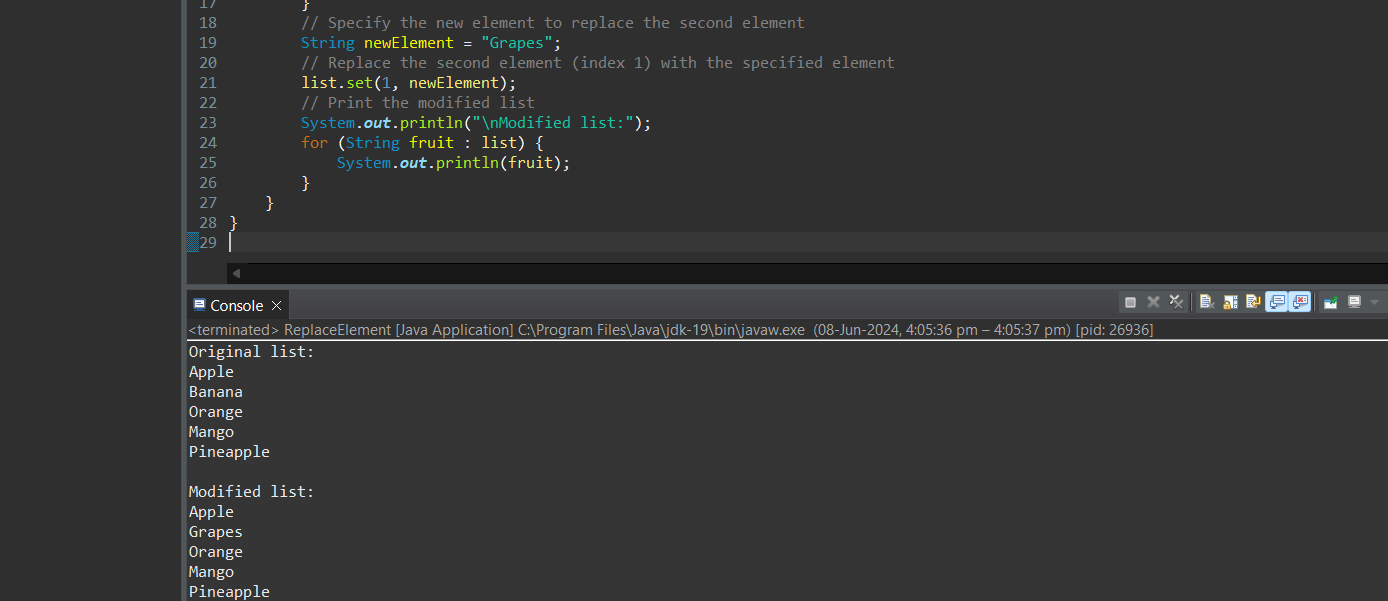
System.out.println(fruit);

}

}

}

**Output**



Q6.     Write a Java program to iterate a linked list in reverse order.

**Program**

package demo;

import java.util.LinkedList;

import java.util.ListIterator;

public class ReverseLinkedList {

public static void main(String[] args) {

// Create a LinkedList of strings

LinkedList<String> list = new LinkedList<>();

// Add elements to the LinkedList

list.add("Apple");

list.add("Banana");

list.add("Orange");

list.add("Mango");

list.add("Pineapple");

// Print the original list

System.out.println("Original list:");

for (String fruit : list) {

System.out.println(fruit);

}

// Print the list in reverse order

System.out.println("\nList in reverse order:");

ListIterator<String> iterator = list.listIterator(list.size());

while (iterator.hasPrevious()) {

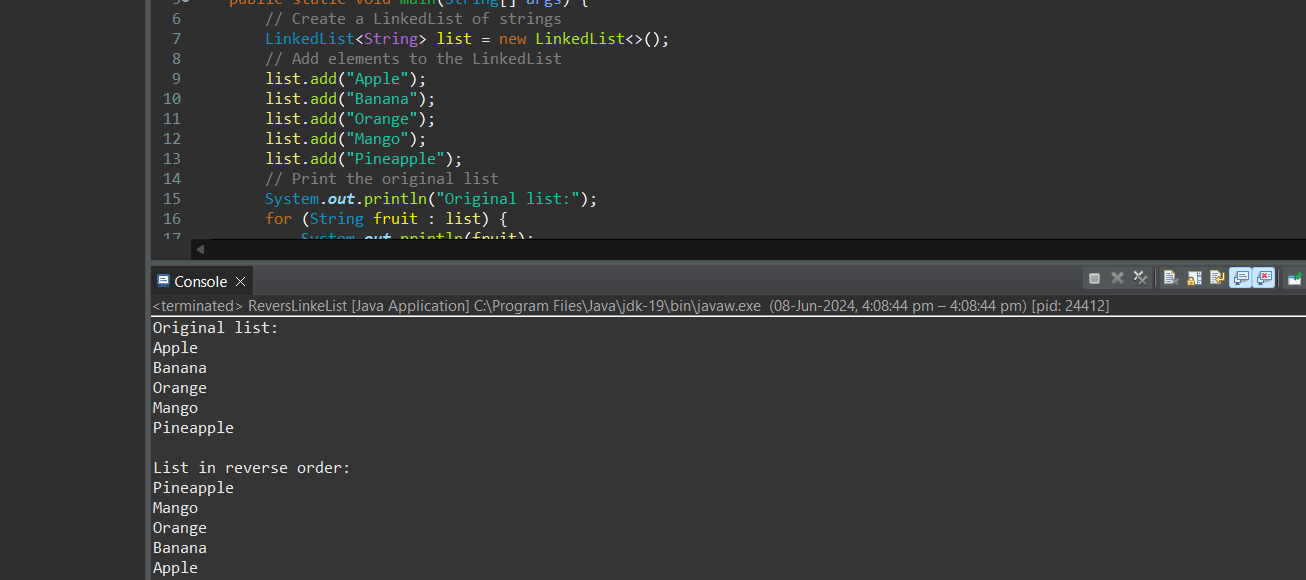
System.out.println(iterator.previous());

}

}

}

**Output**



Q7.     Write a Java program to retrieve, but not remove, the last element of a linked list.

**Program**

package demo;

import java.util.LinkedList;

public class RetrieveLastElement {

public static void main(String[] args) {

// Create a LinkedList of strings

LinkedList<String> list = new LinkedList<>();

// Add elements to the LinkedList

list.add("Apple");

list.add("Banana");

list.add("Orange");

list.add("Mango");

list.add("Pineapple");

// Retrieve but do not remove the last element

String lastElement = list.peekLast();

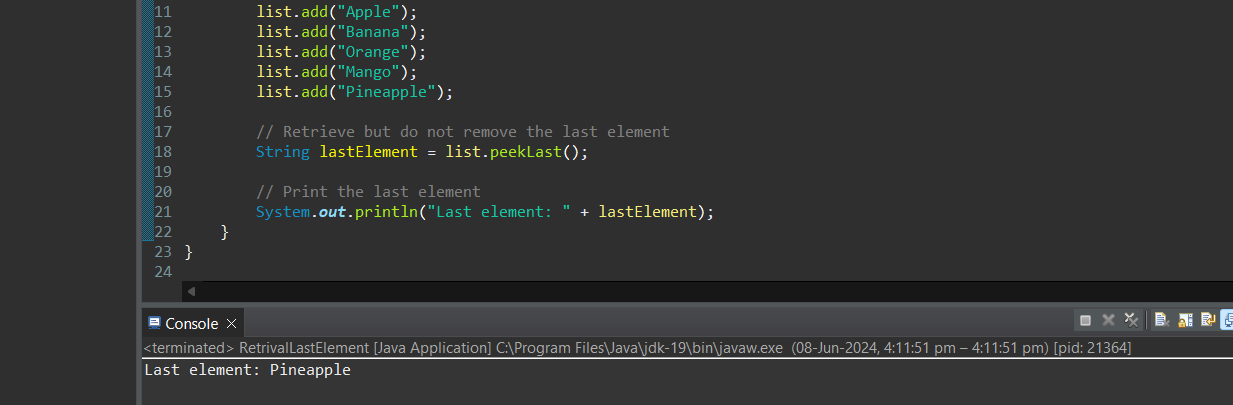
// Print the last element

System.out.println("Last element: " + lastElement);

}

}

**Output**



Q8.     Write a Java program to create a LinkedList of integers and print all the elements.

**Program**

package demo;

import java.util.LinkedList;

public class PrintLinkedList {

public static void main(String[] args) {

// Create a LinkedList of integers

LinkedList<Integer> list = new LinkedList<>();

// Add elements to the LinkedList

list.add(10);

list.add(20);

list.add(30);

list.add(40);

list.add(50);

// Print all elements of the LinkedList

System.out.println("Elements of the LinkedList:");

for (Integer number : list) {

System.out.println(number);

}

}

}

**Output**

