1.     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.

CODE:

package Tokenizer;

import java.util.Scanner;

import java.util.StringTokenizer;

public class StringSplitter {

public static void main(String[] args) {

//create a Scanner object for reading input from the user

Scanner s = new Scanner(System.in);

//prompt the user to enter a string

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

//read the entire line of input

String input = s.nextLine();

//create a StringTokenizer to split the string into words

StringTokenizer tokenizer = new StringTokenizer(input);

//print each token (word) on a new line

while (tokenizer.hasMoreTokens()) {

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

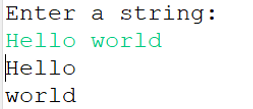
}

s.close();

}

}

OUTPUT:



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

CODE:

package StringToken;

import java.util.Scanner;

import java.util.StringTokenizer;

public class WordCounter {

public static void main(String[] args) {

//create a Scanner object for reading input from the user

Scanner s = new Scanner(System.in);

//prompt the user to enter a string

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

//read the entire line of input

String input = s.nextLine();

//create a StringTokenizer to split the string into words

StringTokenizer t = new StringTokenizer(input);

//count the number of tokens (words)

int wordCount = t.countTokens();

//print the number of words

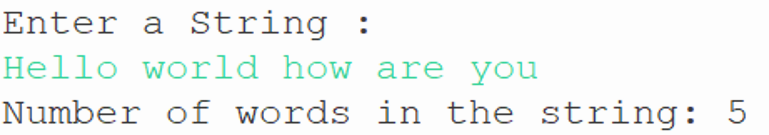
System.out.println("Number of words in the string: " + wordCount);

s.close();

}

}

OUTPUT:



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

CODE:

package LinkedList;

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("End"); // Add to the end

list.addFirst("Beginning"); // Add to the beginning

//add an element to the middle

list.add(1, "Middle");

//print the LinkedList

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

for (String element : list) {

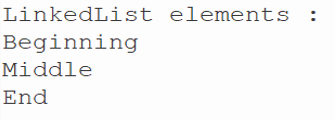
System.out.println(element);

}

}

}

OUTPUT:



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

CODE:

package ArrayList;

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("Mango");

list.add("Cherry");

list.add("Date");

//print the ArrayList before sorting

System.out.println("ArrayList before Sorting:");

for (String element : list) {

System.out.println(element);

}

//sort the ArrayList

Collections.sort(list);

//print the ArrayList after sorting

System.out.println("\nArrayList after sorting:");

for (String element : list) {

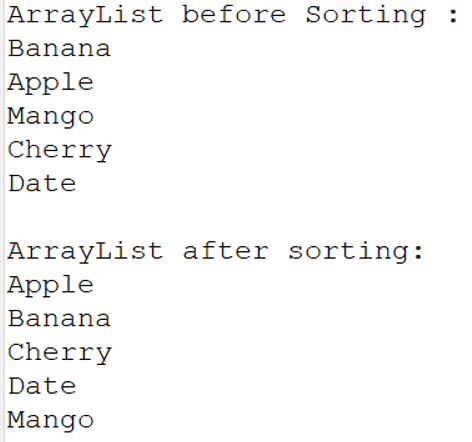
System.out.println(element);

}

}

}

OUTPUT:



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

CODE:

package Array;

import java.util.ArrayList;

public class ReplaceElement {

public static void main(String[] args) {

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

//add elements to the ArrayList

list.add("ONE");

list.add("TWO");

list.add("THREE");

list.add("FOUR");

list.add("FIVE");

//print the ArrayList before replacement

System.out.println("ArrayList before replacement:");

for (String element : list) {

System.out.println(element);

}

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

String newElement = "New Second";

list.set(1, newElement);

//print the ArrayList after replacement

System.out.println("\nArrayList after replacement:");

for (String element : list) {

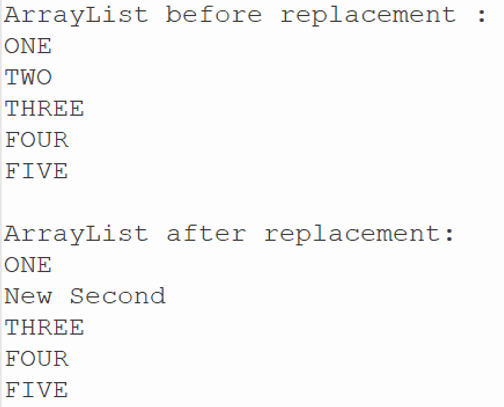
System.out.println(element);

}

}

}

OUTPUT:



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

CODE:

package Linked;

import java.util.LinkedList;

import java.util.ListIterator;

public class ReverseLinkedList {

public static void main(String[] args) {

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

//add elements to the LinkedList

list.add("ONE");

list.add("TWO");

list.add("THREE");

list.add("FOUR");

list.add("FIVE");

//print the LinkedList elements

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

for (String element : list) {

System.out.println(element);

}

//iterate the LinkedList in reverse order

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

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

while (iterator.hasPrevious()) {

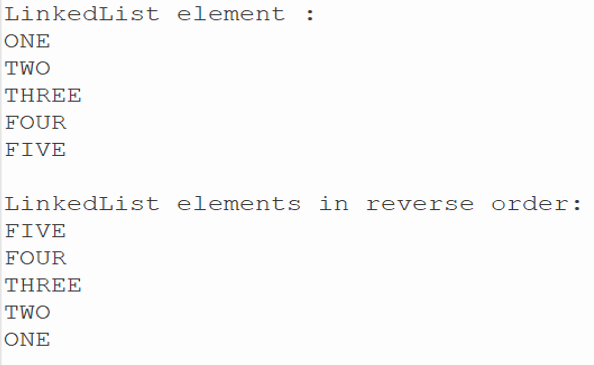
System.out.println(iterator.previous());

}

}

}

OUTPUT:



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

CODE:

package Retrieve;

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("Dog");

list.add("Rat");

list.add("Cat");

list.add("Bat");

//retrieve, but do not remove, the last element

String lastElement = list.peekLast();

//print the last element

System.out.println("The last element is: " + lastElement);

//print the LinkedList to show it has not been modified

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

for (String element : list) {

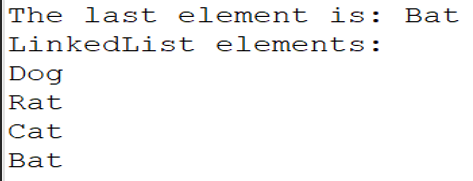
System.out.println(element);

}

}

}

OUTPUT:



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

CODE:

package Elements;

import java.util.LinkedList;

public class LinkedListOfIntegers {

public static void main(String[] args) {

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 the LinkedList elements

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

for (Integer element : list) {

System.out.println(element);

}

}

}

OUTPUT:

