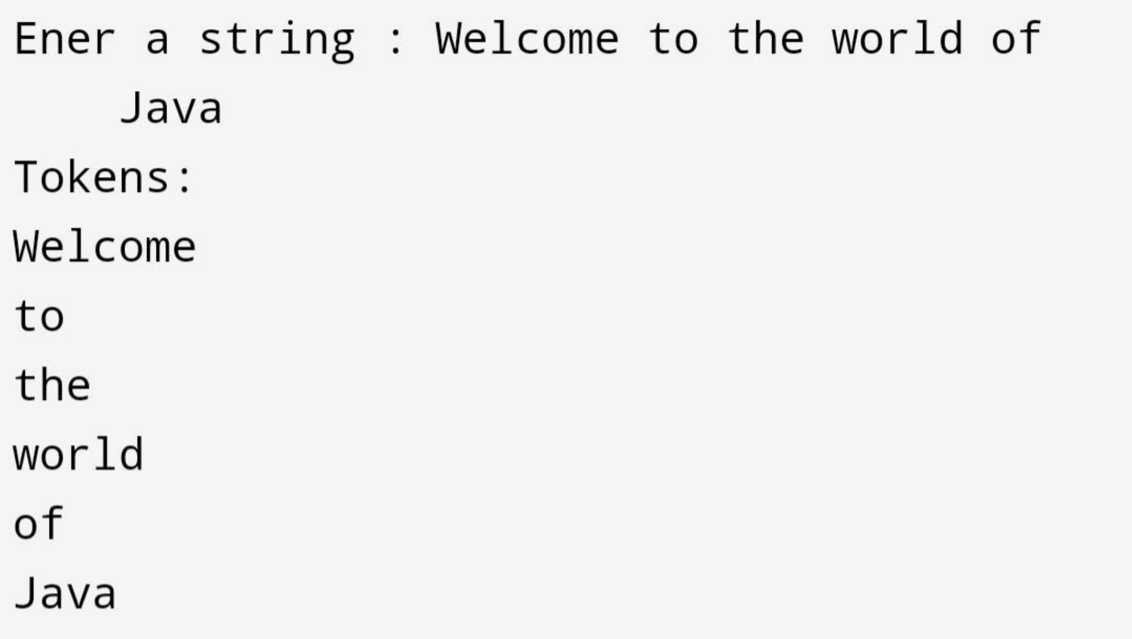
1.Write a Java program that reads a string from the user and uses StringTokenizer to split the string into individual words. Print each word on a new line.

Code:-

```
package MyPackage;
//importing packages
Import java.util.Scanner;
Import java.util.StringTokenizer;
Public class StringTokenizerDemo
{
  Public static void main (String[] args)
  {
    //creating object of Scanner
    Scanner sc = new Scanner(System.in);

    //prompt user to enter a string
    System.out.print(“Ener a string : “);
    String str = sc.nextLine();

    //creating object of StringTokenizer and passing string and delimiter
    StringTokenizer st = new StringTokenizer(str, “ “);

    //printing tokens
    System.out.println(“Tokens:”);
    While (st.hasMoreTokens()) {
      System.out.println(st.nextToken());
    }
  }
}
```

Output:-



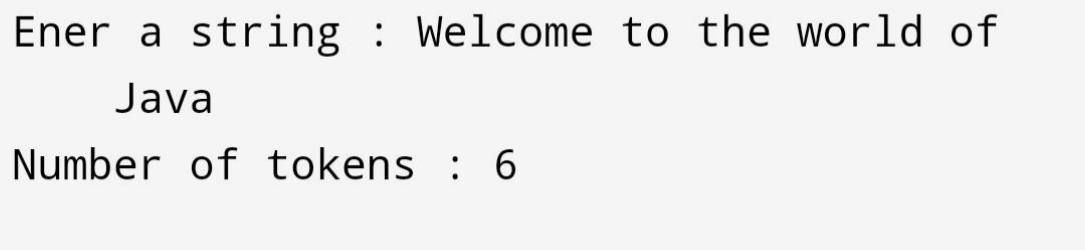
```
Ener a string : Welcome to the world of
    Java
Tokens:
Welcome
to
the
world
of
Java
```

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 MyPackage;
//importing packages
Import java.util.Scanner;
Import java.util.StringTokenizer;
Public class StringTokenizerDemo
{
  Public static void main (String[] args)
  {
    //creating object of Scanner
    Scanner sc = new Scanner(System.in);

    //prompt user to enter a string
    System.out.print(“Ener a string : “);
    String str = sc.nextLine();

    //creating object of StringTokenizer and passing string and delimiter
    StringTokenizer st = new StringTokenizer(str, “ “);

   //using countTokens() method to count tokens
    Int count = st.countTokens();

    //printing numbers of tokens
    System.out.println(“Number of tokens : “ + count);
  }
}
```

Output:-



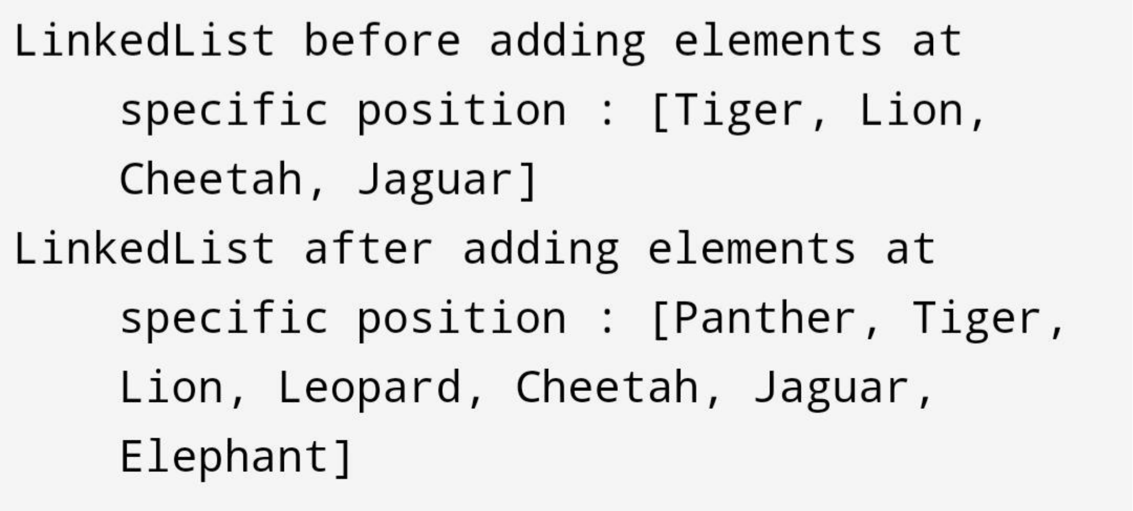
```
Ener a string : Welcome to the world of
    Java
Number of tokens : 6
```

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 MyPackage;
import java.util.*;
public class LinkedListDemo
{
  Public static void main (String[] args)
  {
    //creating string LinkedList
    LinkedList<String> animals = new LinkedList<>();

    //adding elements in LinkedList
    Animals.add(“Tiger”);
    Animals.add(“Lion”);
    Animals.add(“Cheetah”);
    Animals.add(“Jaguar”);

    //printing LinkedList
    System.out.println(“LinkedList before adding elements at specific
position : “ + animals);

    //adding elements at specific position
    Animals.addFirst(“Panther”);
    Animals.add(3, “Leopard”);
    Animals.addLast(“Elephant”);

    //printing LinkedList
    System.out.println(“LinkedList after adding elements at specific
position : “ + animals);
  }
}
```

Output:-



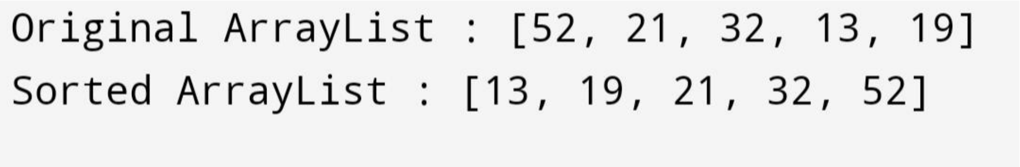
```
LinkedList before adding elements at
    specific position : [Tiger, Lion,
    Cheetah, Jaguar]
LinkedList after adding elements at
    specific position : [Panther, Tiger,
    Lion, Leopard, Cheetah, Jaguar,
    Elephant]
```

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

Code:-

```
package MyPackage;
//importing packages
Import java.util.ArrayList;
Import java.util.Collections;
Public class ArrayListSorter
{
  Public static void main (String[] args)
  {
    //creating array list of integer type
    ArrayList<Integer> numbers = new ArrayList<>();

    //adding elements in array list
    Numbers.add(52);
    Numbers.add(21);
    Numbers.add(32);
    Numbers.add(13);
    Numbers.add(19);

    //printing Original ArrayList
    System.out.println(“Original ArrayList : “ + numbers);

    //sorting array
    Collections.sort(numbers);

    //printing Sorted ArrayList
    System.out.println(“Sorted ArrayList : “ + numbers);
  }
}
```

Output:-



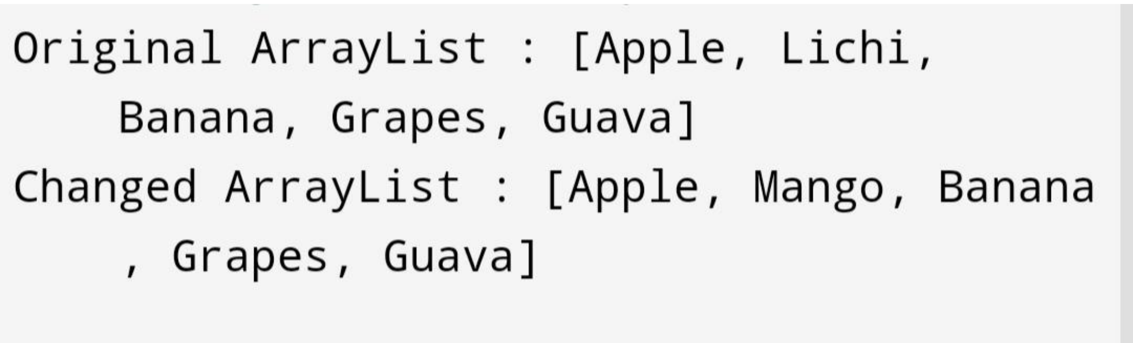
```
Original ArrayList : [52, 21, 32, 13, 19]
Sorted ArrayList : [13, 19, 21, 32, 52]
```

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

Code:-

```
package MyPackage;
import java.util.ArrayList;
public class ArrayListElementReplacer
{
  Public static void main (String[] args)
  {
    //creating array list of integer type
    ArrayList<String> fruits = new ArrayList<>();

    //adding elements to array list
    Fruits.add(“Apple”);
    Fruits.add(“Lichi”);
    Fruits.add(“Banana”);
    Fruits.add(“Grapes”);
    Fruits.add(“Guava”);

    //printing Original ArrayList
    System.out.println(“Original ArrayList : “ + fruits);

    //replacing the second element of array list
    Fruits.set(1, “Mango”);

    //printing changed ArrayList
    System.out.println(“Changed ArrayList : “ + fruits);
  }
}
```

Output:-



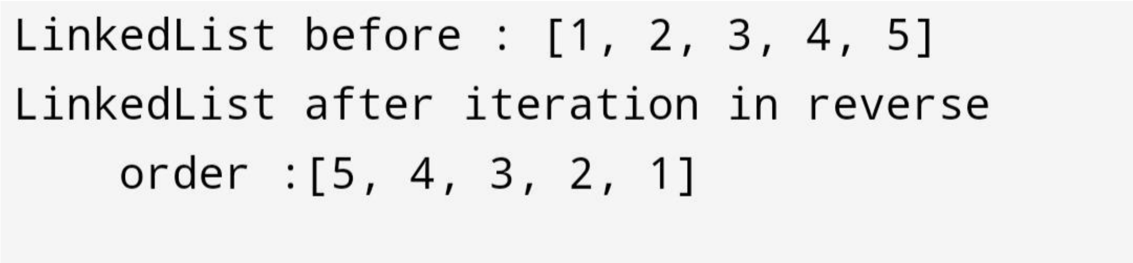
```
Original ArrayList : [Apple, Lichi,
    Banana, Grapes, Guava]
Changed ArrayList : [Apple, Mango, Banana
    , Grapes, Guava]
```

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

Code:-

```
package MyPackage;
import java.util.*;
public class LinkedListDemo
{
  Public static void main (String[] args)
  {
    //creating linked list of integer type
    LinkedList<Integer> numbers = new LinkedList<>();

    //adding elements in linked list
    Numbers.add(1);
    Numbers.add(2);
    Numbers.add(3);
    Numbers.add(4);
    Numbers.add(5);

    //printing linked list
    System.out.println(“LinkedList before : “ + numbers);

    //sorting linked list in reverse order
    Collections.sort(numbers, Collections.reverseOrder());

    //printing linked list
    System.out.println(“LinkedList after iteration in reverse order :” +
numbers);
  }
}
```

Output:-



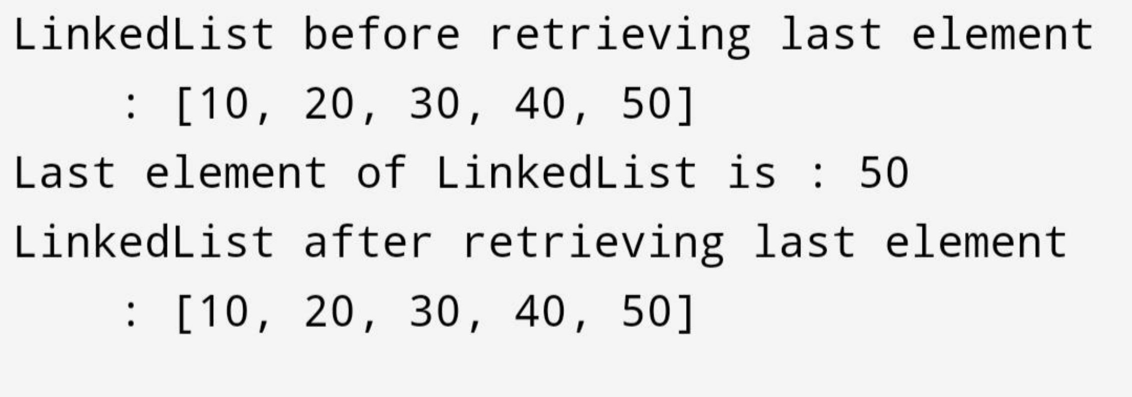
```
LinkedList before : [1, 2, 3, 4, 5]
LinkedList after iteration in reverse
    order :[5, 4, 3, 2, 1]
```

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

Code:-

```
package MyPackage;
import java.util.*;
public class LinkedListDemo
{
  Public static void main (String[] args)
  {
    //creating linked list of integer type
    LinkedList<Integer> numbers = new LinkedList<>();

    //adding elements in linked list
    Numbers.add(10);
    Numbers.add(20);
    Numbers.add(30);
    Numbers.add(40);
    Numbers.add(50);

    //printing LinkedList before retrieving last element
    System.out.println(“LinkedList before retrieving last element : “ +
numbers);

    //retrieving last element of linked list and storing in variable
    Int lastElement = numbers.getLast();

    //printing last element of LinkedList
    System.out.println(“Last element of LinkedList is : “ + lastElement);

    //printing LinkedList after retrieving last element
    System.out.println(“LinkedList after retrieving last element : “ +
numbers);
  }
}
```

Output:-



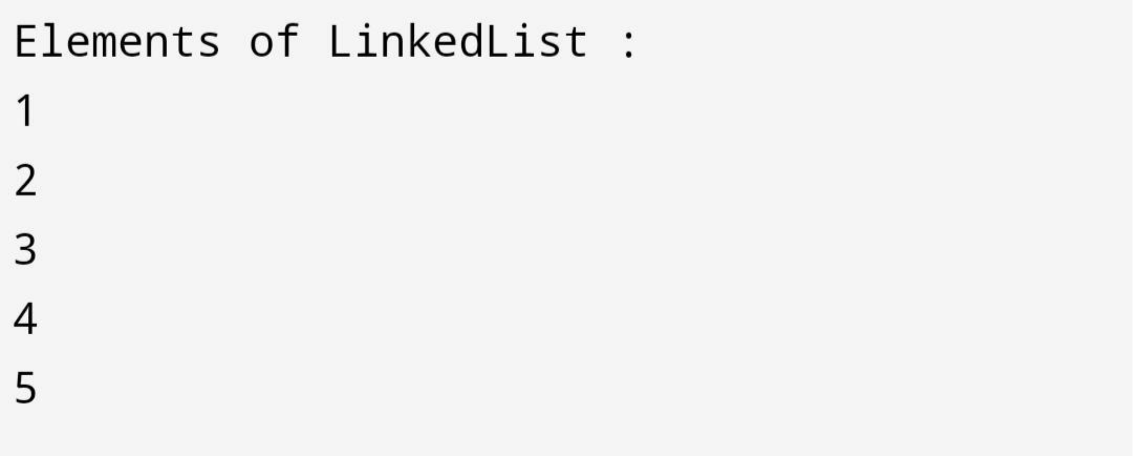
```
LinkedList before retrieving last element
    : [10, 20, 30, 40, 50]
Last element of LinkedList is : 50
LinkedList after retrieving last element
    : [10, 20, 30, 40, 50]
```

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

Code:-

```
package MyPackage;
import java.util.*;
public class LinkedListDemo
{
  Public static void main (String[] args)
  {
    //creating linked list of integer type
    LinkedList<Integer> numbers = new LinkedList<>();

    //adding elements in linked list
    Numbers.add(1);
    Numbers.add(2);
    Numbers.add(3);
    Numbers.add(4);
    Numbers.add(5);

    //printing elements of LinkedList using for each loop
    System.out.println(“Elements of LinkedList : “);
    For (Integer number : numbers) {
      System.out.println(number);
    }
  }
}
```

Output:-



```
Elements of LinkedList :
1
2
3
4
5
```