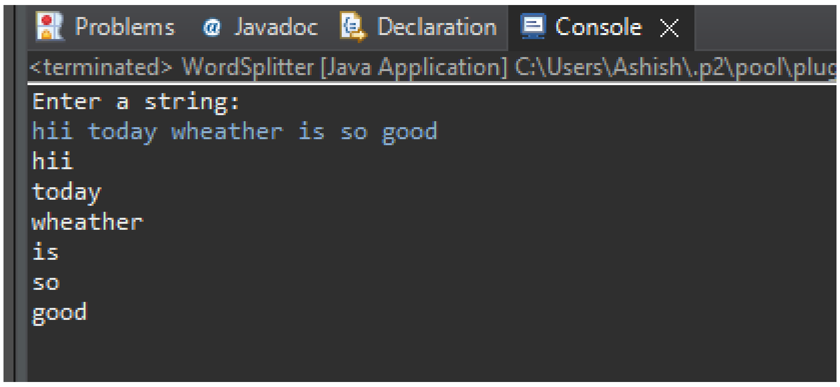
Lab 5

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:
```java
package hellow;

import java.util.Scanner;
import java.util.StringTokenizer;

public class WordSplitter {
  public static void main(String[] args) {
    // Create a Scanner object to read input from the user
    Scanner scanner = new Scanner(System.in);

    // Prompt the user to enter a string
    System.out.println("Enter a string:");
    String input = scanner.nextLine();

    // Create a StringTokenizer to split the string into words
    StringTokenizer tokenizer = new StringTokenizer(input);

    // Print each word on a new line
    while (tokenizer.hasMoreTokens()) {
      System.out.println(tokenizer.nextToken());
    }

    // Close the scanner
    scanner.close();
  }
}
```

Output:



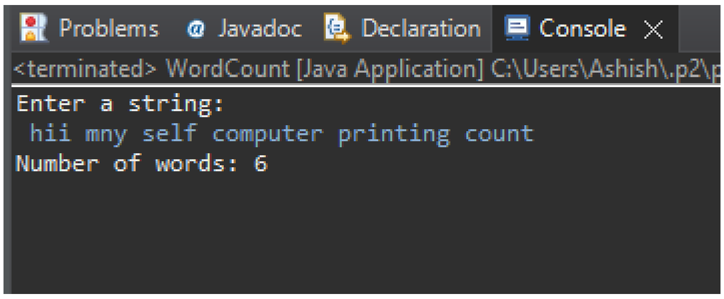
```
<terminated> WordSplitter [Java Application] C:\Users\Ashish\.p2\pool\plug
Enter a string:
hii today wheather is so good
hii
today
wheather
is
so
good
```

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:
```java
package hellow;
import java.util.Scanner;
import java.util.StringTokenizer;
```

```java
public class WordCount {
  public static void main(String[] args) {
    // Create a Scanner object to read input from the user
    Scanner scanner = new Scanner(System.in);

    // Prompt the user to enter a string
    System.out.println("Enter a string:");
    String input = scanner.nextLine();

    // Create a StringTokenizer to split the string into words
    StringTokenizer tokenizer = new StringTokenizer(input);

    // Initialize word count
    int wordCount = 0;

    // Count the number of words
    while (tokenizer.hasMoreTokens()) {
      tokenizer.nextToken(); // Move to the next token
      wordCount++;
    }

    // Print the number of words
    System.out.println("Number of words: " + wordCount);

    // Close the scanner
    scanner.close();
  }
}
```

Output:



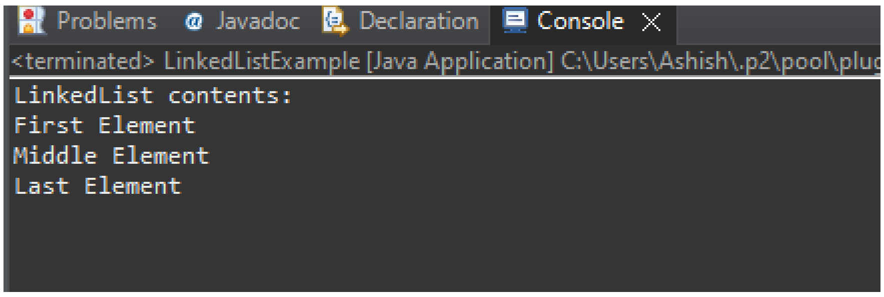
```
<terminated> WordCount [Java Application] C:\Users\Ashish\.p2\p
Enter a string:
 hii mny self computer printing count
Number of words: 6
```

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

Code:

```java
package hellow;

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 beginning of the list
    list.addFirst("First Element");

    // Add elements to the end of the list
    list.addLast("Last Element");

    // Add elements at a specific position (e.g., middle)
    // For this example, we'll add the element at index 1
    list.add(1, "Middle Element");

    // Print the entire LinkedList
    System.out.println("LinkedList contents:");
    for (String element : list) {
      System.out.println(element);
    }
  }
}
```

Output:



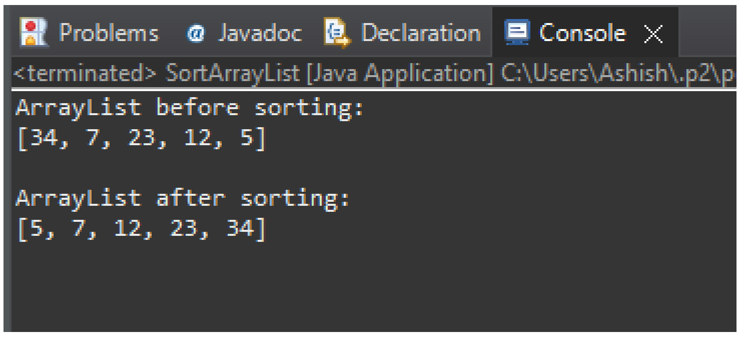
```
<terminated> LinkedListExample [Java Application] C:\Users\Ashish\.p2\pool\plug
LinkedList contents:
First Element
Middle Element
Last Element
```

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

```
package hellow;

import java.util.*;

public class SortArrayList {
  public static void main(String[] args) {
    // Create an ArrayList of integers
    ArrayList<Integer> numbers = new ArrayList<>();

    // Add some integers to the list
    numbers.add(34);
    numbers.add(7);
    numbers.add(23);
    numbers.add(12);
    numbers.add(5);

    // Print the list before sorting
```

```
        System.out.println("ArrayList before sorting:");
        System.out.println(numbers+"\n");

        // Sort the ArrayList
        Collections.sort(numbers);

        // Print the list after sorting
        System.out.println("ArrayList after sorting:");
        System.out.println(numbers);
    }
}
```

Output:

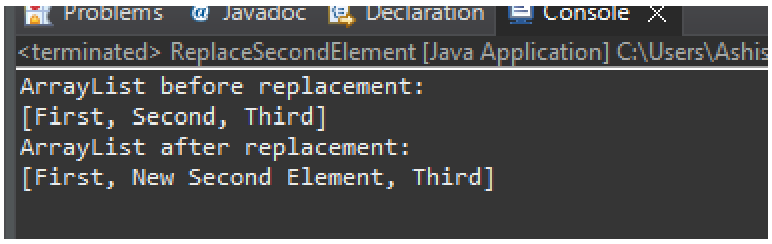


```
Problems  Javadoc  Declaration  Console
<terminated> SortArrayList [Java Application] C:\Users\Ashish\.p2\p
ArrayList before sorting:
[34, 7, 23, 12, 5]

ArrayList after sorting:
[5, 7, 12, 23, 34]
```

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

```
package hellow;

import java.util.ArrayList;

public class ReplaceSecondElement {
    public static void main(String[] args) {
        // Create an ArrayList of strings
        ArrayList<String> list = new ArrayList<>();

        // Add some elements to the list
        list.add("First");
        list.add("Second");
        list.add("Third");

        // Print the list before replacement
        System.out.println("ArrayList before replacement:");
        System.out.println(list);

        // Define the new element to replace the second element
        String newElement = "New Second Element";

        // Replace the second element (index 1) with the new element
        if (list.size() > 1) { // Ensure there are at least 2 elements
            list.set(1, newElement);
        } else {
            System.out.println("The list does not have a second element.");
        }

        // Print the list after replacement
        System.out.println("ArrayList after replacement:");
        System.out.println(list);
    }
}
```

Output:



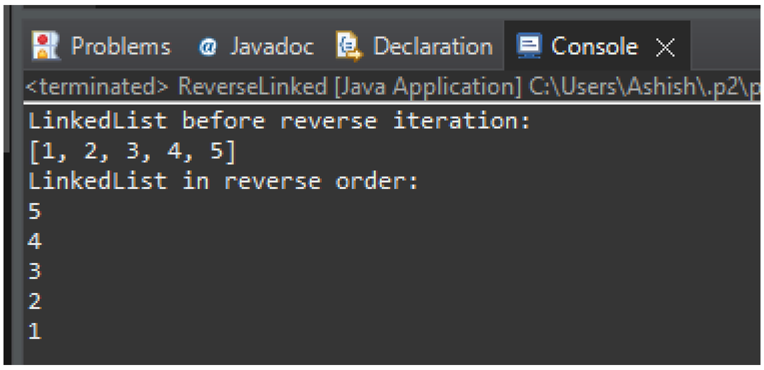
```
<terminated> ReplaceSecondElement [Java Application] C:\Users\Ashis
ArrayList before replacement:
[First, Second, Third]
ArrayList after replacement:
[First, New Second Element, Third]
```

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

Code:

```
package hellow;

import java.util.LinkedList;
import java.util.ListIterator;

public class ReverseLinked {
  public static void main(String[] args) {
    // Create a LinkedList of integers
    LinkedList<Integer> list = new LinkedList<>();

    // Add some integers to the list
    list.add(1);
    list.add(2);
    list.add(3);
    list.add(4);
    list.add(5);

    // Print the list before iteration
    System.out.println("LinkedList before reverse iteration:");
    System.out.println(list);

    // Iterate the LinkedList in reverse order
    System.out.println("LinkedList in reverse order:");
    ListIterator<Integer> iterator = list.listIterator(list.size()); // Start from
the end
    while (iterator.hasPrevious()) {
      System.out.println(iterator.previous());
    }
  }
}
```

Output:



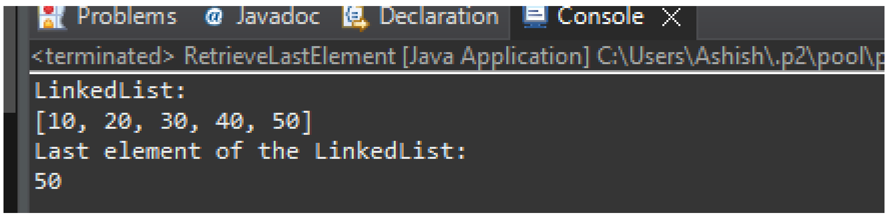
```
<terminated> ReverseLinked [Java Application] C:\Users\Ashish\.p2\p
LinkedList before reverse iteration:
[1, 2, 3, 4, 5]
LinkedList in reverse order:
5
4
3
2
1
```

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

Code:

```java
package hellow;

import java.util.LinkedList;

public class RetrieveLastElement {
  public static void main(String[] args) {
    // Create a LinkedList of integers
    LinkedList<Integer> list = new LinkedList<>();

    // Add some integers to the list
    list.add(10);
    list.add(20);
    list.add(30);
    list.add(40);
    list.add(50);

    // Print the list before retrieving the last element
    System.out.println("LinkedList:");
    System.out.println(list);

    // Retrieve the last element without removing it
    Integer lastElement = list.getLast();

    // Print the last element
    System.out.println("Last element of the LinkedList:");
    System.out.println(lastElement);
  }
}
```

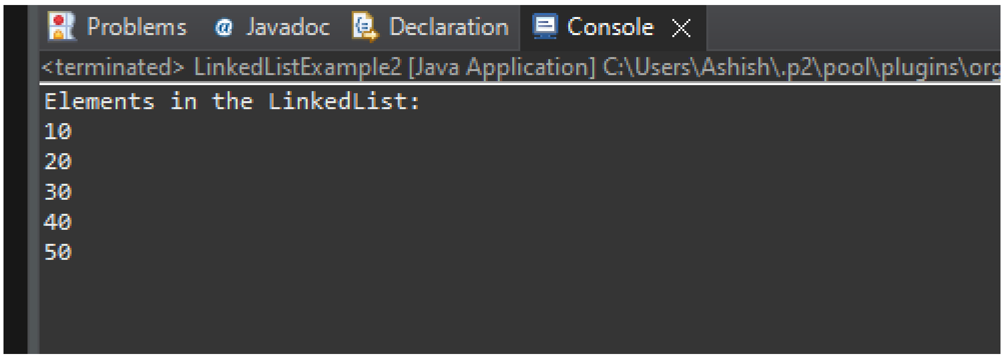
Output:



```
<terminated> RetrieveLastElement [Java Application] C:\Users\Ashish\.p2\pool\p
LinkedList:
[10, 20, 30, 40, 50]
Last element of the LinkedList:
50
```

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

Code:
```
package hellow;

import java.util.LinkedList;

public class LinkedListExample2 {
  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 in the LinkedList:");
    for (Integer element : list) {
      System.out.println(element);
    }
  }
}
```

Output:


```
<terminated> LinkedListExample2 [Java Application] C:\Users\Ashish\.p2\pool\plugins\org
Elements in the LinkedList:
10
20
30
40
50
```