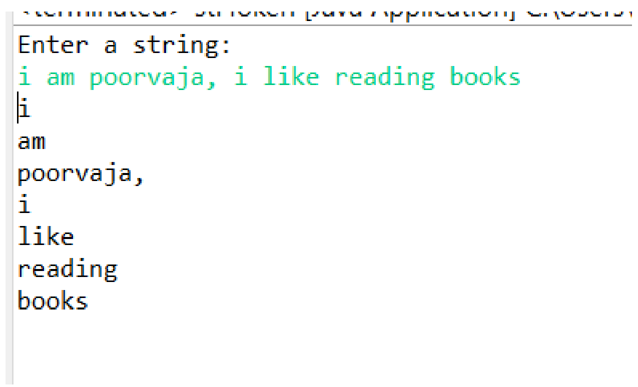
Poorvaja Bhorunde

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.

```
package lab;
import java.util.StringTokenizer;
import java.util.Scanner;


public class strToken {

        public static void main(String[] args) {
    // Create a Scanner object for getting input from the user
    Scanner s = new Scanner(System.in);

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

    // Create a StringTokenizer object 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());
    }
        }
}
```

Output:



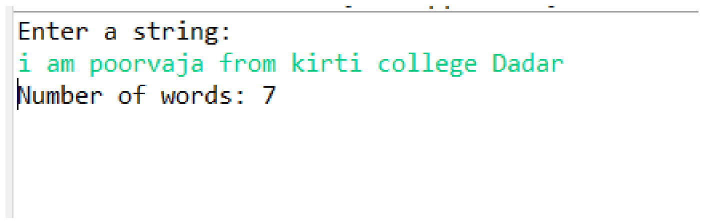
```
<terminated> strToken [Java Application] C:\Users\
Enter a string:
i am poorvaja, i like reading books
i
am
poorvaja,
i
like
reading
books
```

Poorvaja Bhorunde

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

```java
package lab;

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

public class WordCount {

        public static void main(String[] args) {
    Scanner a = new Scanner(System.in);

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

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

    // Counting the number of words
    int wordCount = 0;
    while (tokenizer.hasMoreTokens()) {
        tokenizer.nextToken();
        wordCount++;
    }

    // Printing the word count
    System.out.println("Number of words: " + wordCount);

                }

        }
```

Output:



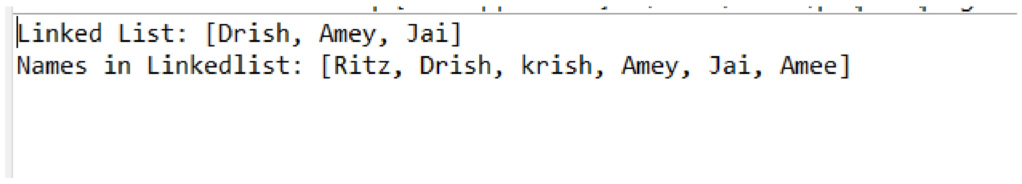
```
Enter a string:
i am poorvaja from kirti college Dadar
Number of words: 7
```

Poorvaja Bhorunde

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

```
package lab;

import java.util.LinkedList;

public class LinkedlistExamp {

	public static void main(String[] args) {

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

		linkedList.add("Drish");
		linkedList.add("Amey");
		linkedList.add("Jai");

		System.out.println("Linked List: "+linkedList);
		linkedList.addFirst("Ritz");       // Adding Ritz at the beginning

	linkedList.add(2, "krish"); // Adding krish at index 2

		linkedList.addLast("Amee");  // Adding amee at the last



		System.out.println("Names in Linkedlist: "+linkedList);//printing list of names



	}

}
```

Output:



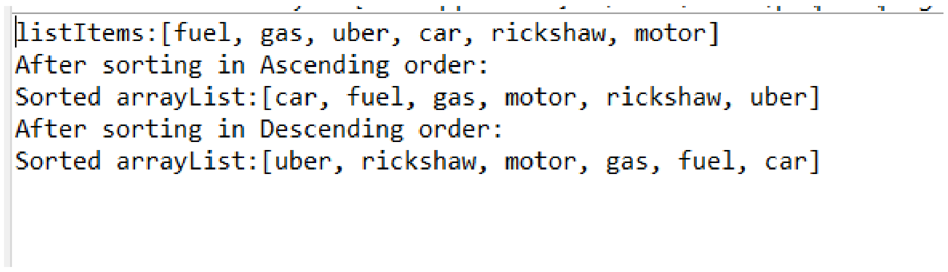
```
Linked List: [Drish, Amey, Jai]
Names in Linkedlist: [Ritz, Drish, krish, Amey, Jai, Amee]
```

Poorvaja Bhorunde

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

```
package lab;
import java.util.ArrayList;
import java.util.Collections;


public class SortArrayList {

	public static void main(String[] args) {

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

		arrlist.add("fuel"); //adding elements to array list
		arrlist.add("gas");
		arrlist.add("uber");
		arrlist.add("car");
		arrlist.add("rickshaw");
		arrlist.add("motor");
		System.out.println("listItems:"+arrlist); //original array list


		Collections.sort(arrlist);	// Sorting the ArrayList

		System.out.println("After sorting in Ascending order: ");
		System.out.println("Sorted arrayList:"+arrlist);

		Collections.sort(arrlist,Collections.reverseOrder());
		System.out.println("After sorting in Descending order: ");
		System.out.println("Sorted arrayList:"+arrlist);	// Printing the sorted
ArrayList

	}

}
```

**Output:**



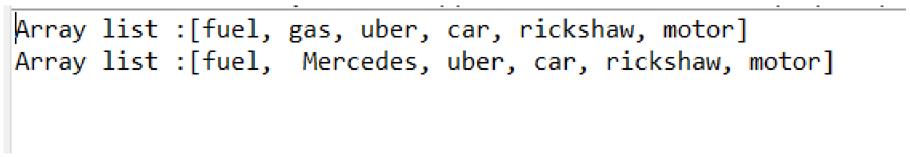
```
listItems:[fuel, gas, uber, car, rickshaw, motor]
After sorting in Ascending order:
Sorted arrayList:[car, fuel, gas, motor, rickshaw, uber]
After sorting in Descending order:
Sorted arrayList:[uber, rickshaw, motor, gas, fuel, car]
```

Poorvaja Bhorunde

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

```java
package lab;
import java.util.ArrayList;


public class SortArrayList {

	public static void main(String[] args) {


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

		arrlist.add("fuel"); //adding elements to array list
		arrlist.add("gas");
		arrlist.add("uber");
		arrlist.add("car");
		arrlist.add("rickshaw");
		arrlist.add("motor");
		System.out.println("Array list :"+arrlist); //original array list

		arrlist.set(1, " Mercedes");
		// Replace the second element (index 1) with a specified element


		System.out.println("Array list :"+arrlist);

	}
}
```

**Output:**

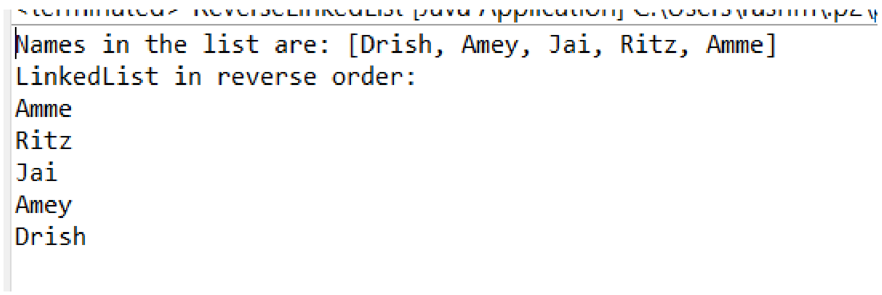


```
Array list :[fuel, gas, uber, car, rickshaw, motor]
Array list :[fuel,  Mercedes, uber, car, rickshaw, motor]
```

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

```
package lab;

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

public class ReverseLinkedList {

        public static void main(String[] args) {


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

                list.add("Drish");
                list.add("Amey");
                list.add("Jai");
                list.add("Ritz");
                list.add("Amme");

                System.out.println("Names in the list are: "+list);

                System.out.println("LinkedList in reverse order:");
     ListIterator<String> iterator = list.listIterator(list.size());
     while (iterator.hasPrevious()) {
        System.out.println(iterator.previous());
     }


        }

}
```

**Output:**

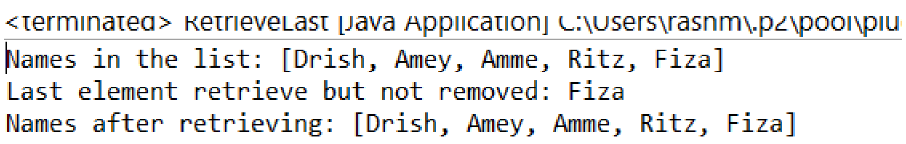


```
<terminated> ReverseLinkedList [Java Application] C:\Users\rashmi\.p2\
Names in the list are: [Drish, Amey, Jai, Ritz, Amme]
LinkedList in reverse order:
Amme
Ritz
Jai
Amey
Drish
```

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

```java
package lab;

import java.util.LinkedList;

public class RetrieveLast {

	public static void main(String[] args) {

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

		List.add("Drish");
		List.add("Amey");
		List.add("Amme");
		List.add("Ritz");
		List.add("Fiza");

		System.out.println("Names in the list: "+List);

	String lastElement = List.getLast();  // Retrieve, but do not remove, the last element

	System.out.println("Last element retrieve but not removed: " + lastElement);


		System.out.println("Names after retrieving: "+List);

	}

}
```

Output:



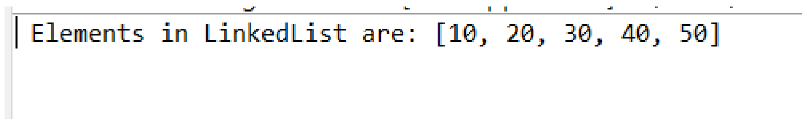
```
<terminated> RetrieveLast [Java Application] C:\Users\rashm\.p2\pool\plu
Names in the list: [Drish, Amey, Amme, Ritz, Fiza]
Last element retrieve but not removed: Fiza
Names after retrieving: [Drish, Amey, Amme, Ritz, Fiza]
```

Poorvaja Bhorunde

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

```
package lab;
import java.util.LinkedList;



public class IntegerLinkedlist {

	public static void main(String[] args) {
	LinkedList<Integer> list = new LinkedList<>();

	list.add(10);
	list.add(20);
	list.add(30);
	list.add(40);
	list.add(50);

	// Printing the elements of the LinkedList
	System.out.println(" Elements in LinkedList are: " + list);

	}

}
```

**Output:**



```
 Elements in LinkedList are: [10, 20, 30, 40, 50]
```