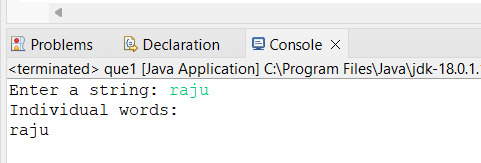
Lab 5

Aditya Kamble

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.
2. **package** lab5;
3. **import** java.util.Scanner;
4. **import** java.util.StringTokenizer;
5. **public** **class** que1 {

8. **public** **static** **void** main(String[] args) {
9. Scanner scanner = **new** Scanner(System.***in***);
11. System.***out***.print("Enter a string: ");
12. String input = scanner.nextLine();
14. StringTokenizer tokenizer = **new** StringTokenizer(input);
16. System.***out***.println("Individual words:");
17. **while** (tokenizer.hasMoreTokens()) {
18. System.***out***.println(tokenizer.nextToken());
19. }
21. scanner.close();
22. }
23. }

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.

-

**package** lab5;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** que2 {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String input = scanner.nextLine();

StringTokenizer tokenizer = **new** StringTokenizer(input);

System.***out***.println("Individual words:");

**while** (tokenizer.hasMoreTokens()) {

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

}

scanner.close();

}

}

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

-

**package** lab5;

**import** java.util.LinkedList;

**public** **class** que3 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

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

linkedList.addFirst("Apple");

linkedList.addFirst("Banana");

linkedList.add(linkedList.size() / 2, "Orange");

linkedList.addLast("Grapes");

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

**for** (String element : linkedList) {

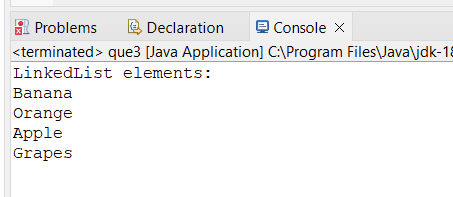
System.***out***.println(element);

}

}

}

Output-



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

-

**package** lab5;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** que4 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<Integer> numbers = **new** ArrayList<>();

numbers.add(5);

numbers.add(2);

numbers.add(8);

numbers.add(1);

numbers.add(3);

System.***out***.println("Original ArrayList: " + numbers);

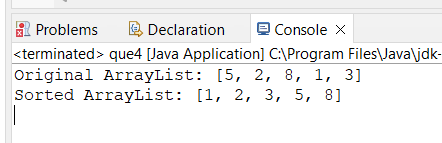
Collections.*sort*(numbers);

System.***out***.println("Sorted ArrayList: " + numbers);

}

}

Output-



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

-

**package** lab5;

**import** java.util.ArrayList;

**public** **class** que5 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<String> color = **new** ArrayList<String>();

color.add("Red");

color.add("Green");

System.***out***.println("Original array list: " + color);

String new\_color = "White";

color.set(1,new\_color);

**int** num=color.size();

System.***out***.println("Replace second element with 'White'.");

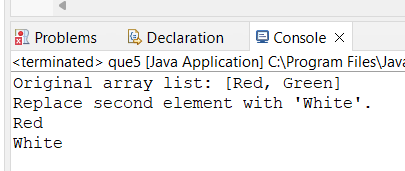
**for**(**int** i=0;i<num;i++)

System.***out***.println(color.get(i));

}

}

Output-



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

-**package** lab5

**import** java.util.LinkedList;

**import** java.util.Iterator;

**public** **class** que6 {

**public** **static** **void** main(String[] args) {

LinkedList<String> l\_list = **new** LinkedList<String>();

l\_list.add("Red");

l\_list.add("Green");

l\_list.add("Black");

l\_list.add("Pink");

l\_list.add("orange");

System.***out***.println("Original linked list:" + l\_list);

Iterator it = l\_list.descendingIterator();

System.***out***.println("Elements in Reverse Order:");

**while** (it.hasNext()) {

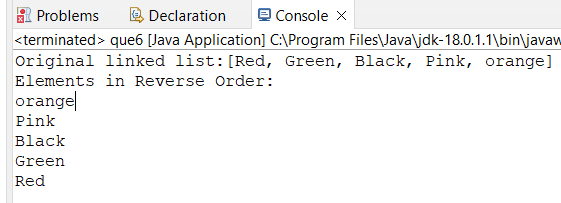
System.***out***.println(it.next());

}

}

}

Output-



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

**package** lab5;

**import** java.util.\*;

**public** **class** que7 {

**public** **static** **void** main(String[] args) {

LinkedList <String> c1 = **new** LinkedList <String> ();

c1.add("Red");

c1.add("Green");

c1.add("Black");

c1.add("White");

c1.add("Pink");

System.***out***.println("Original linked list: " + c1);

String x = c1.peekLast();

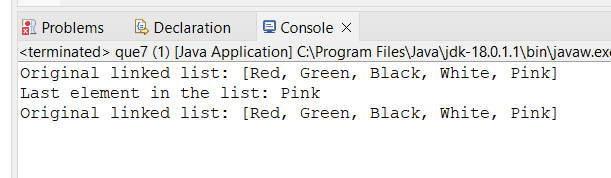
System.***out***.println("Last element in the list: " + x);

System.***out***.println("Original linked list: " + c1);

}

}

Output-



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

-

**package** lab5;

**import** java.util.LinkedList;

**public** **class** que8 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

LinkedList<String> num = **new** LinkedList<>();

num.add("1");

num.add("2");

num.add("3");

System.***out***.println("LinkedList: " + num);

}

}

Output-

