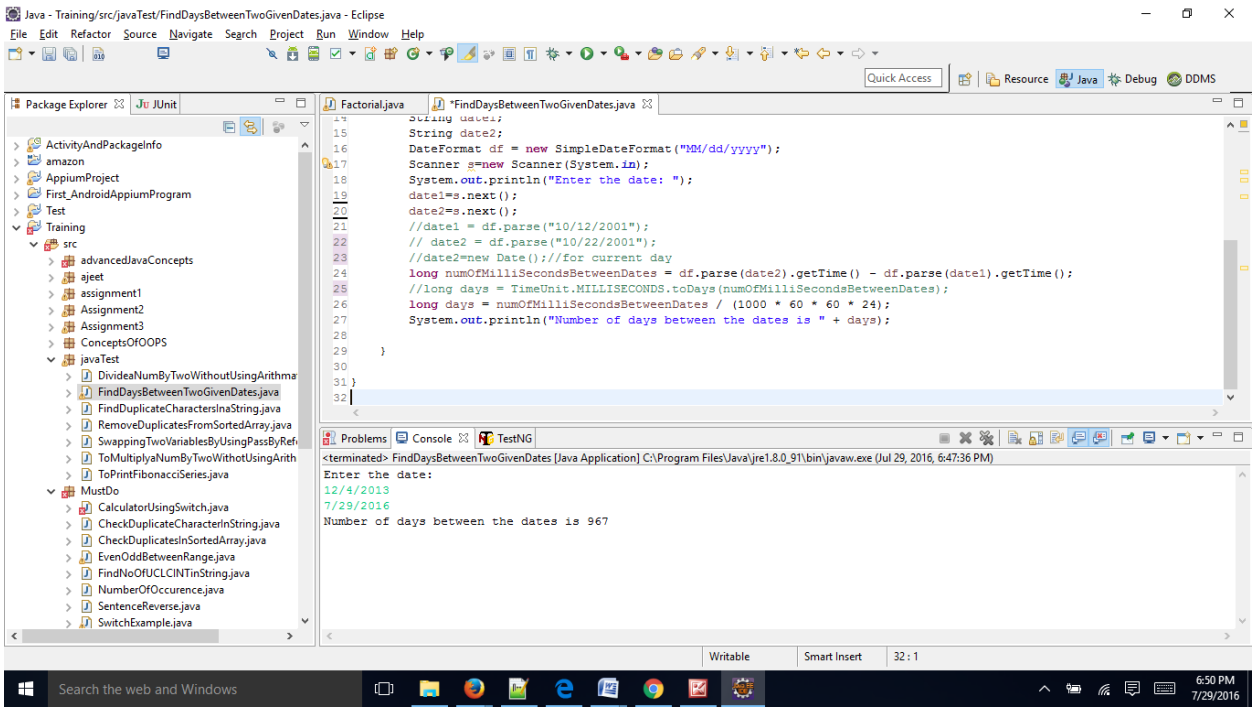
## 1.Find out the number of days in between two  given dates ?

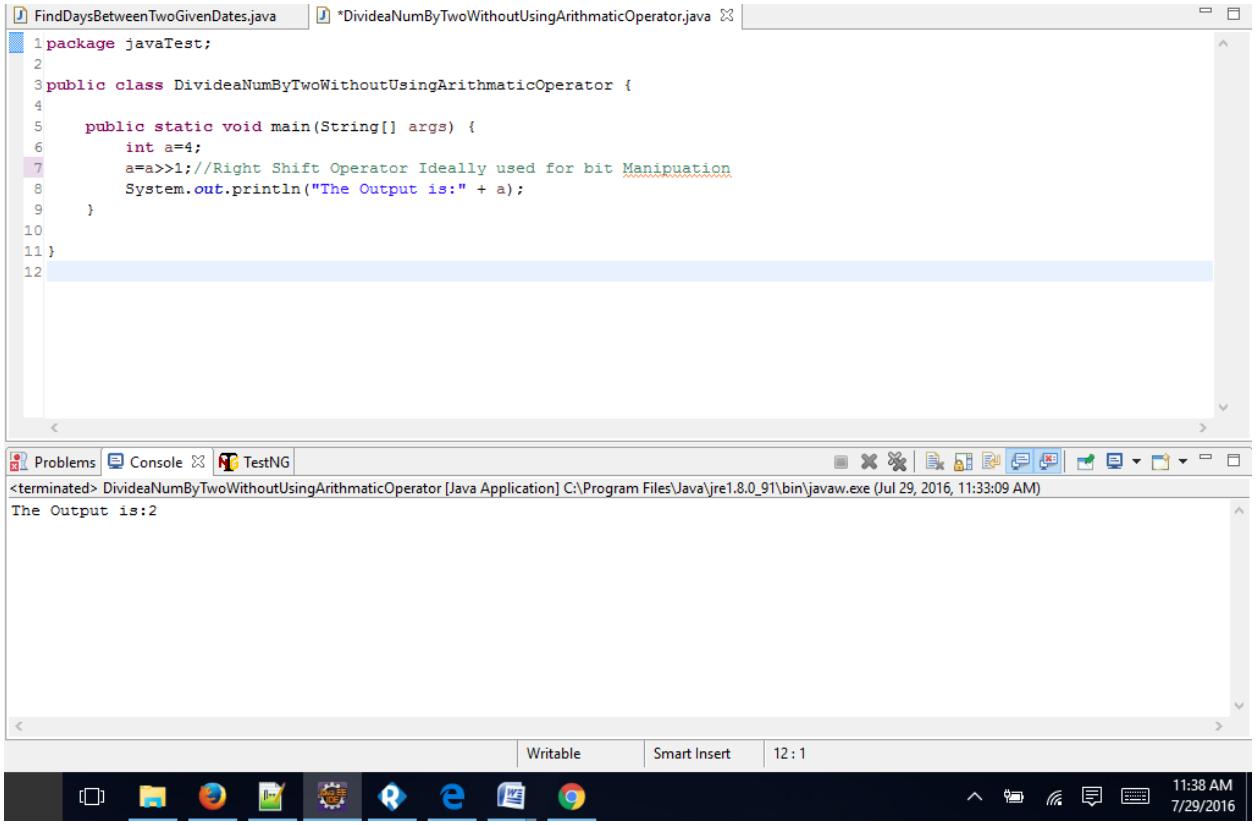
```
package javaTest;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;

public class FindDaysBetweenTwoGivenDates {

	public static void main(String[] args) throws ParseException {
		String date1;
		String date2;
		DateFormat df = new SimpleDateFormat("MM/dd/yyyy");
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the date: ");
		date1=s.next();
		date2=s.next();
		//date1 = df.parse("10/12/2001");
	       // date2 = df.parse("10/22/2001");
		//date2=new Date();//for current day
		long numOfMilliSecondsBetweenDates = df.parse(date2).getTime() -
df.parse(date1).getTime();
		//long days =
TimeUnit.MILLISECONDS.toDays(numOfMilliSecondsBetweenDates);
		long days = numOfMilliSecondsBetweenDates / (1000 * 60 * 60 *
24);
		System.out.println("Number of days between the dates is " +
days);

	}

}
```




## 2.How to divide a number by 2 without using / operator?

Solution:

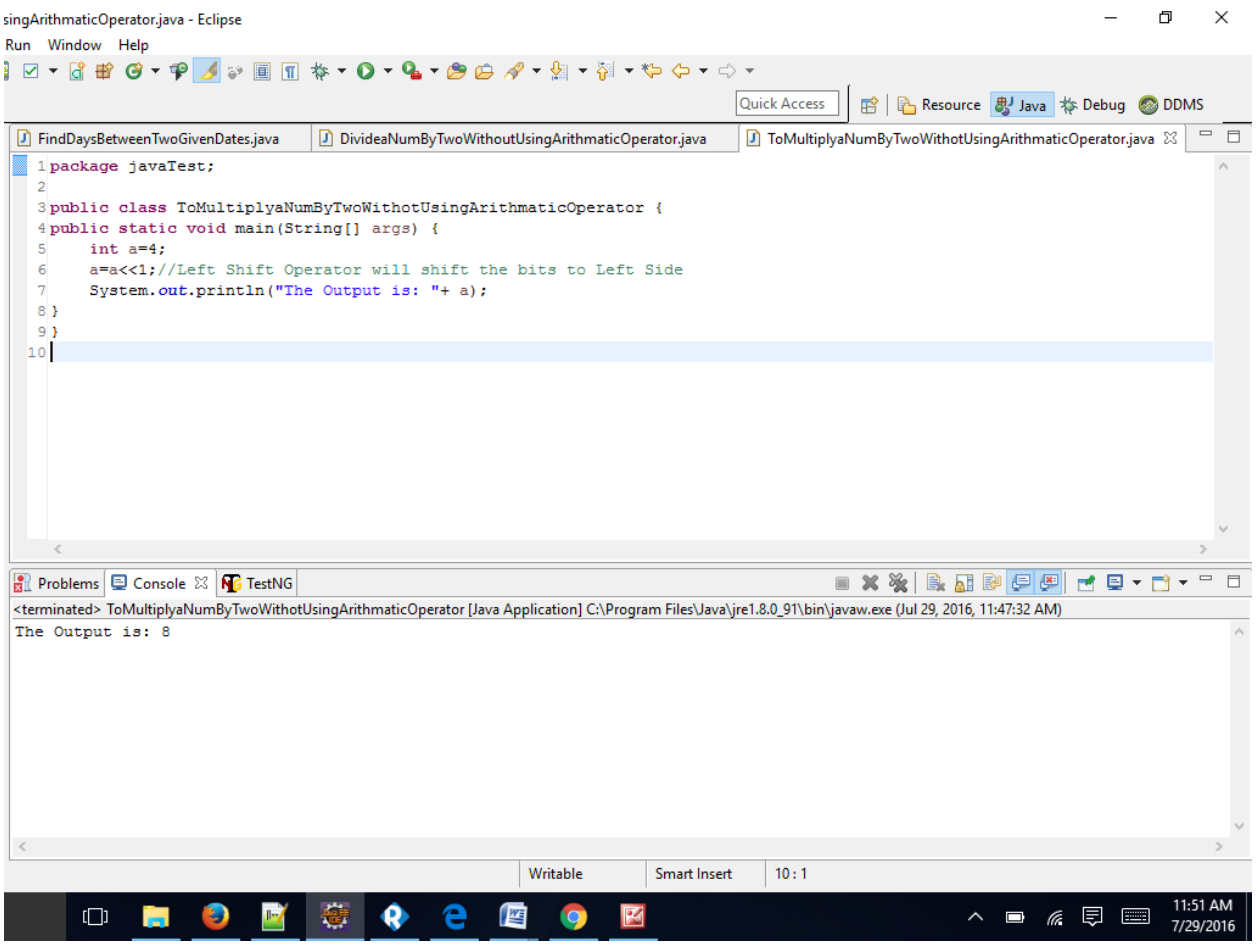
```
package javaTest;

public class DivideaNumByTwoWithoutUsingArithmaticOperator {

public static void main(String[] args) {

                int a=4;

                a=a>>1;//Right Shift Operator Ideally used for bit Manipuation

                System.out.println("The Output is:" + a);

        }

}
```




### 3.How to multiply a number by 2 without using * operator?

### Solution:

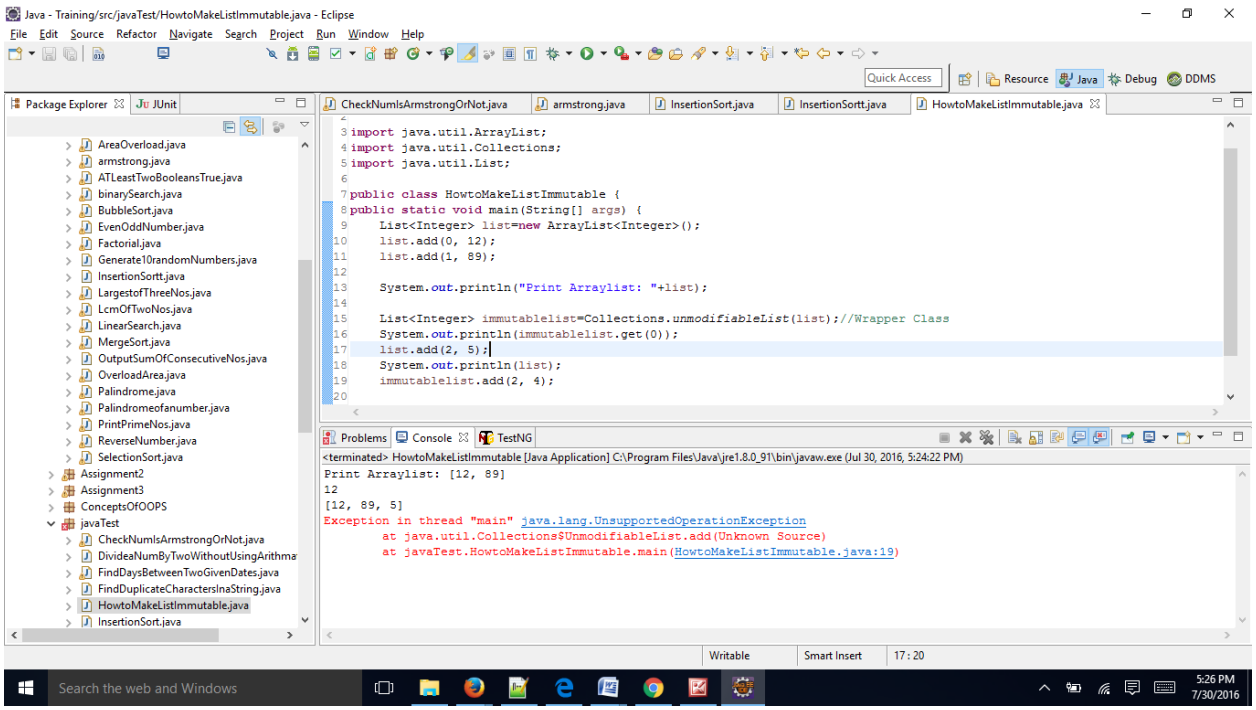
```
package javaTest;

public class ToMultiplyaNumByTwoWithotUsingArithmaticOperator {
public static void main(String[] args) {
      int a=4;
      a=a<<1;//Left Shift Operator will shift the bits to Left Side
      System.out.println("The Output is: "+ a);
}


   }
```




## 5.How to make a list immutable?

```
package javaTest;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class HowtoMakeListImmutable {
public static void main(String[] args) {
      List<Integer> list=new ArrayList<Integer>();
      list.add(0, 12);
      list.add(1, 89);

      System.out.println("Print Arraylist: "+list);

      List<Integer> immutablelist=Collections.unmodifiableList(list);
    System.out.println(immutablelist.get(0));
    list.add(2, 5);
    System.out.println(list);
      immutablelist.add(2, 4);
```

```
    }

}
```

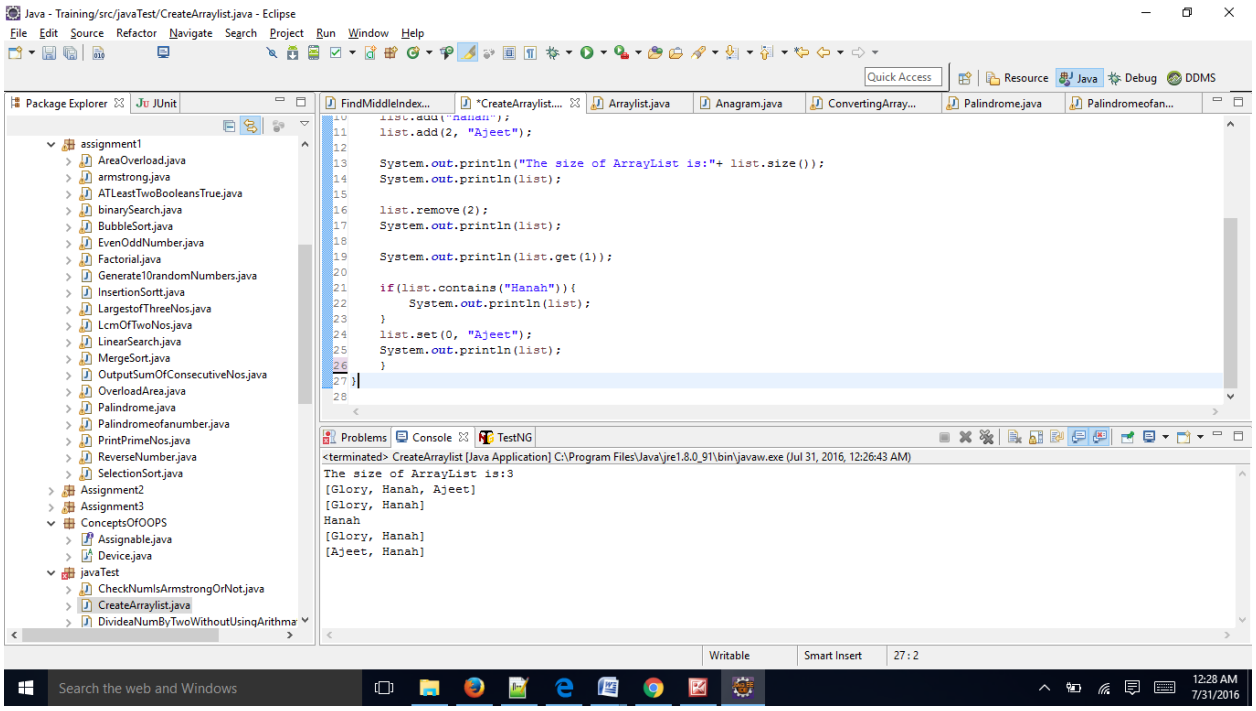



## 6. Write a sample code to reverse Singly Linked List by iterating through it only once.

```
package javaTest;

import javax.xml.soap.Node;

public class ReverseSinglyLinkedListByItThroughOnlyOnce {
public static void main(String[] args) {
      int data;
      Node link;
    Node previous=null;
      Node current=start;

      while(current!=null){
            Node next=current.getLink;
            current.setLink(previous);
            previous=current;
            current=next;
      }

}
}
```

## 7. Write a program to implement ArrayList and Linked list

```java
package javaTest;

import java.util.ArrayList;
import java.util.List;

public class CreateArraylist {
public static void main(String[] args) {
	List<String>list=new ArrayList<String>();
	list.add("Glory");
	list.add("Hanah");
	list.add(2, "Ajeet");

	System.out.println("The size of ArrayList is:"+ list.size());
	System.out.println(list);

	list.remove(2);
	System.out.println(list);

	System.out.println(list.get(1));

	if(list.contains("Hanah")){
		System.out.println(list);
	}
	list.set(0, "Ajeet");
	System.out.println(list);
	}
}
```



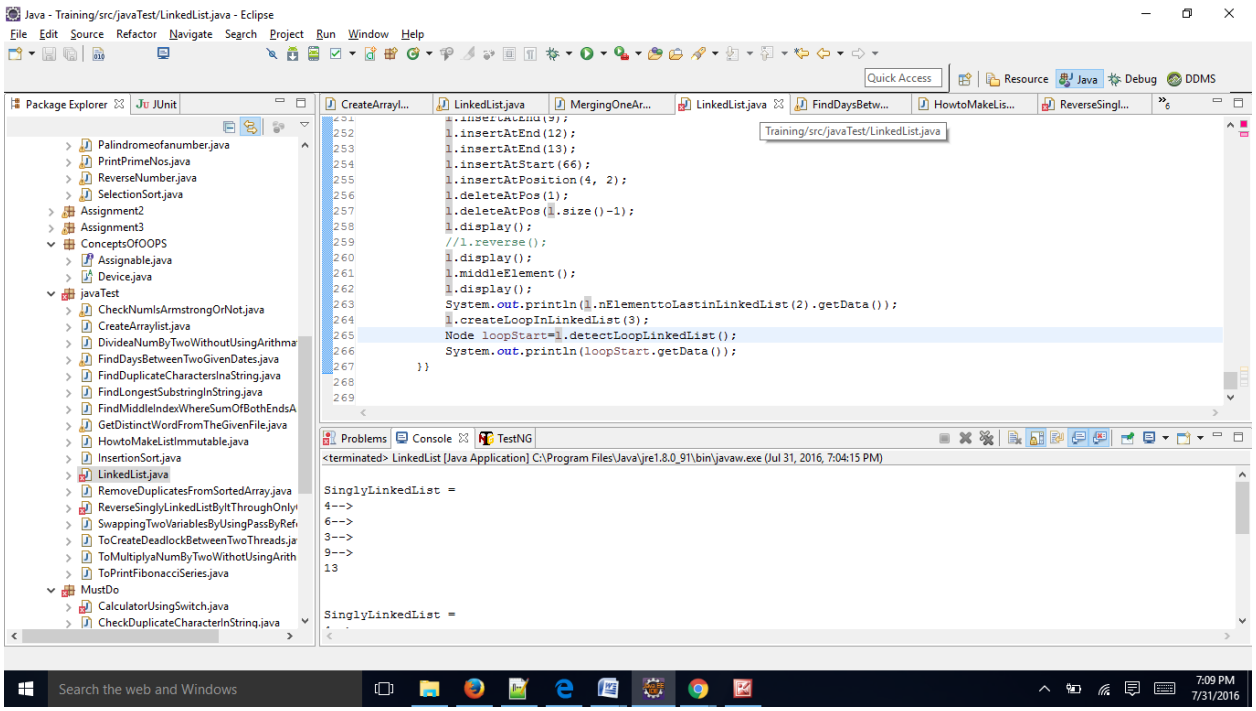

LinkedList:

```java
import advancedJavaConcepts.Node;
```

```
class Node {

	private int data;
	Node link;
	public Object next;

	public Node(){
		data=0;
		link=null;
	}
	public Node(int d,Node n ){
		data=d;
		link=n;
	}
	public void setLink(Node n){
		link=n;
	}
	public void setData(int d){
		data=d;
	}
	public Node getLink(){
		return link;
	}
	public int getData(){
		return data;
	}
	}

	public class LinkedList{
		private static final Node Node = null;
		protected Node start;
		protected Node end;
		int size;
		public Object next;

		public LinkedList(){
			start=null;
			end=null;
			size=0;
		}
		//Adding node at beginning
		public void insertAtStart(int val){
			Node nptr=new Node(val,null);
			if(start==null){
				start=nptr;
				end=start;
			}else{
				nptr.setLink(start);
				start=nptr;
			}
			size++;
		}

		//Adding node at end
		public void insertAtEnd(int val){
			Node nptr=new Node(val,null);
```

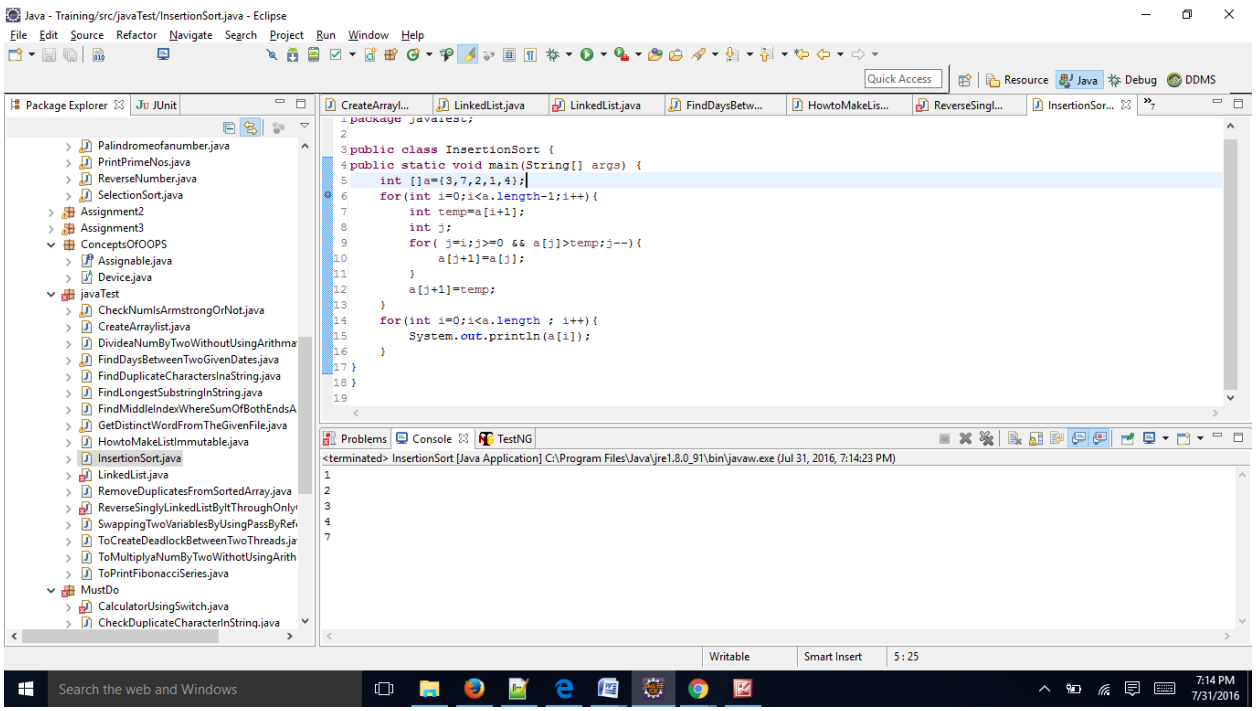
```
        if(start==null){
                start=nptr;
                end=start;
        }else{
                end.setLink(nptr);
                end=nptr;
        }
        size++;
}

//To insert an element at a position

public void insertAtPosition(int val,int pos){
        Node nptr=new Node(val,null);
        Node ptr=start;
        pos=pos-1;
        for(int i=1;i<size;i++){
                if(i==pos){
                        Node tmp=ptr.getLink();
                        ptr.setLink(nptr);
                        nptr.setLink(tmp);
                }
                ptr=ptr.getLink();
        }size++;
                }

//To delete an element at the start

public void deleteAtPos(int pos){
        if(pos==1){
                start=start.getLink();
                size--;
                return;
        }

        //To delete an element at the end
        if(pos==size){
                Node s=start;
                Node t=start;
                while( s!= end){
                        t=s;
                        s=s.getLink();
                }
                end=t;
                t.setLink(null);
                size--;
                return;
                        }


//To delete an element at a position

                        Node ptr=start;
                        pos=pos-1;
                        for(int i=1;i<size;i++){
                                if(i==pos){
                                        Node tmp=ptr.getLink().getLink();
```

```
                                    ptr.setLink(tmp);
                                }
                                ptr=ptr.getLink();
                            }size--;
                                }

//Create Loop in LinkedList
            public void createLoopInLinkedList(int n){
                  Node loopStart=start;
                  for(int i=0;i<n;i++){
                    loopStart=loopStart.getLink();
                  }
                  end.setLink(loopStart);
            }

            //To dispay elements
            public void display(){
                  System.out.println("\nSinglyLinkedList =");
                  if(size==0){
                        System.out.println("empty\n");
                        return;
                  }
                  if(start.getLink()==null){
                        System.out.println(start.getData());
                        return;
                  }
                  Node ptr=start;
                  System.out.println(start.getData()+"-->");
                  ptr=start.getLink();
                  while(ptr.getLink()!=null){
                        System.out.println(ptr.getData()+"-->");
                        ptr=ptr.getLink();
                  }
                  System.out.println(ptr.getData()+"\n");
                  }
            public static void main(String[] args) throws Exception {
                  LinkedList l=new LinkedList();
                  l.insertAtStart(3);
                  l.insertAtStart(6);
                  l.insertAtEnd(9);
                  l.insertAtEnd(12);
                  l.insertAtEnd(13);
                  l.insertAtStart(66);
                  l.insertAtPosition(4, 2);
                  l.deleteAtPos(1);
                  l.deleteAtPos(l.size()-1);
                  l.display();
                  //l.reverse();
                  l.display();
                  l.middleElement();
                  l.display();

      System.out.println(l.nElementtoLastinLinkedList(2).getData());
                  l.createLoopInLinkedList(3);
                  Node loopStart=l.detectLoopLinkedList();
                  System.out.println(loopStart.getData());
            }}
```



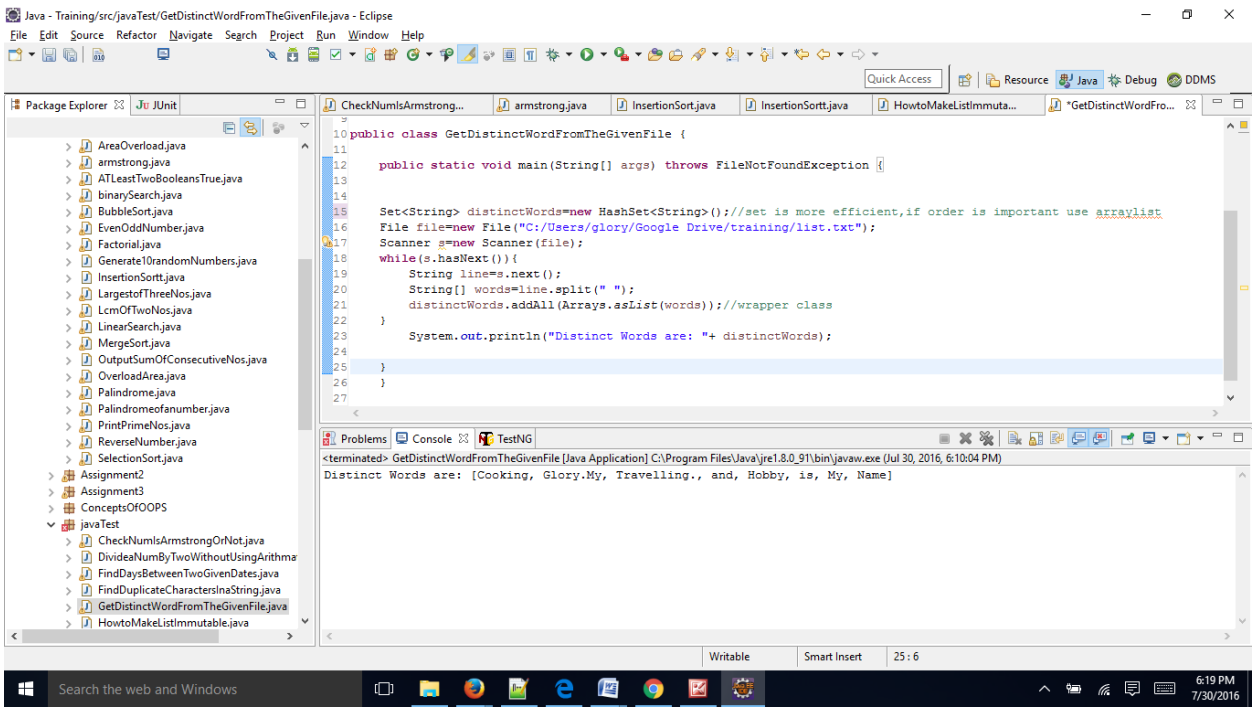

## 8.Write a program for Insertion Sort in java.

```
package javaTest;

public class InsertionSort {
public static void main(String[] args) {
	int []a={3,7,2,1,4};
	for(int i=0;i<a.length-1;i++){
		int temp=a[i+1];
		int j;
	  for( j=i;j>=0 && a[j]>temp;j--){
			a[j+1]=a[j];
		}
		a[j+1]=temp;
	}
	for(int i=0;i<a.length ; i++){
		System.out.println(a[i]);
	}
}
}
```




## 9.Write a program to get distinct word list from the given file.

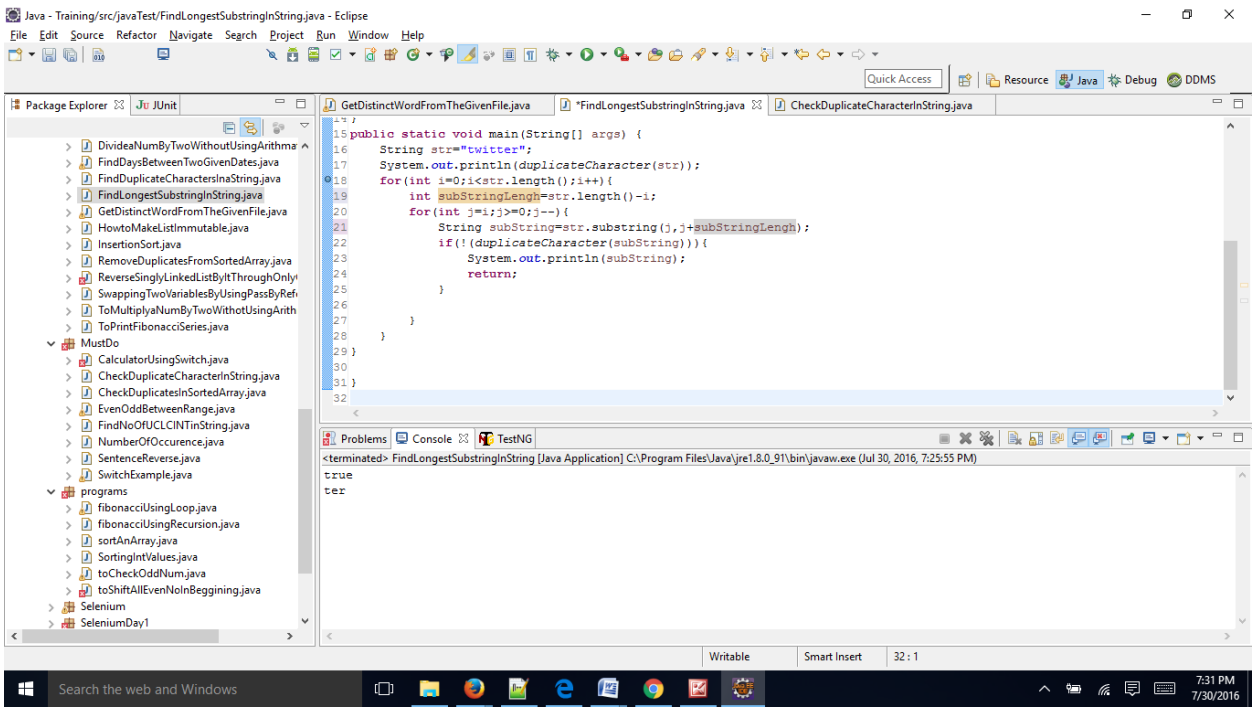
```
package javaTest;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class GetDistinctWordFromTheGivenFile {

	public static void main(String[] args) throws FileNotFoundException {


	Set<String> distinctWords=new HashSet<String>();//set is more
efficient,if order is important use arraylist
	File file=new File("C:/Users/glory/Google Drive/training/list.txt");
    Scanner s=new Scanner(file);
    while(s.hasNext()){
      String line=s.next();
      String[] words=line.split(" ");
      distinctWords.addAll(Arrays.asList(words));//wrapper class
    }
      System.out.println("Distinct Words are: "+ distinctWords);

    }
      }
```




## 10.Find longest substring without repeating characters.

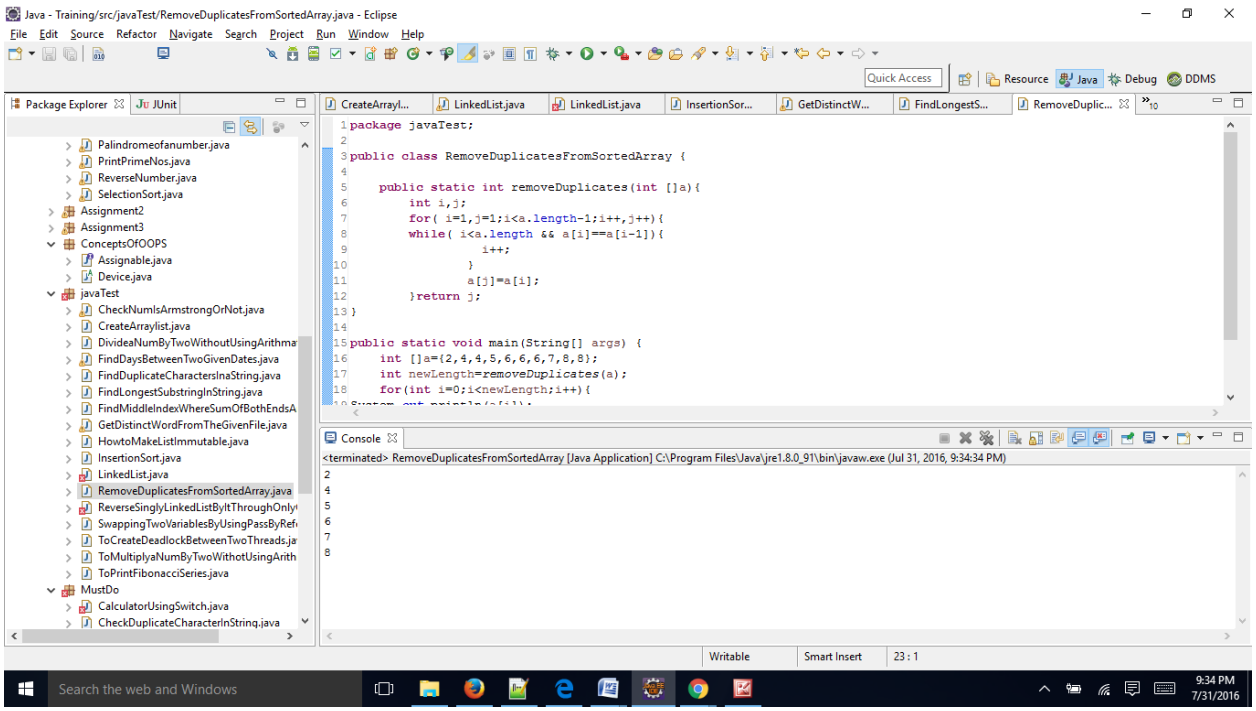
```
package javaTest;

public class FindLongestSubstringInString {
public static boolean duplicateCharacter(String str){


	for(int i=0;i<str.length();i++){
		for(int j=i+1;j<str.length();j++){
			if(str.charAt(i)==str.charAt(j))
				return true;
			}
	}
	return false;
}
public static void main(String[] args) {
	String str="twitter";
	System.out.println(duplicateCharacter(str));
	for(int i=0;i<str.length();i++){
		int subStringLengh=str.length()-i;
		for(int j=i;j>=0;j--){
			String subString=str.substring(j,j+subStringLengh);
			if(!(duplicateCharacter(subString))){
				System.out.println(subString);
				return;
			}

		}
	}
}
```

}



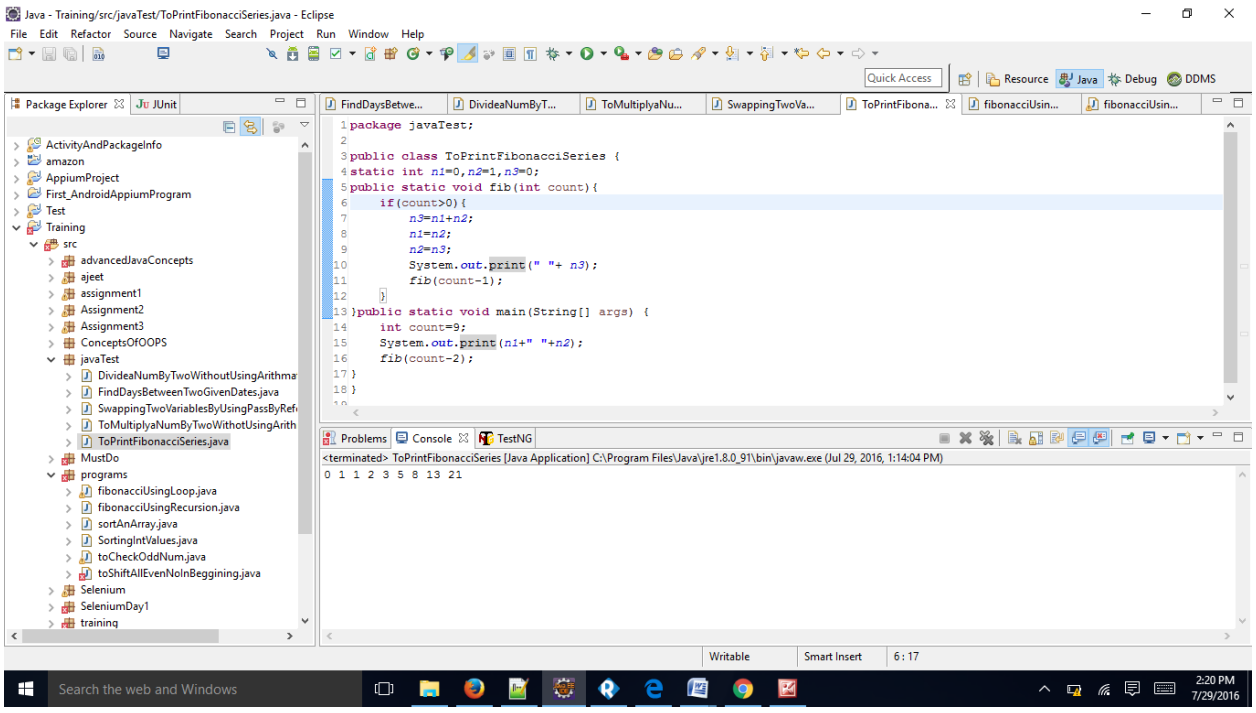

## 11.Write a program to remove duplicates from sorted array

```
package javaTest;

public class RemoveDuplicatesFromSortedArray {

	public static int removeDuplicates(int []a){
		int i,j;
		for( i=1,j=1;i<a.length-1;i++,j++){
		while( i<a.length && a[i]==a[i-1]){
			i++;
		}
			a[j]=a[i];
		}return j;
}

public static void main(String[] args) {
	int []a={2,4,4,5,6,6,6,7,8,8};
	int newLength=removeDuplicates(a);
	for(int i=0;i<newLength;i++){
System.out.println(a[i]);
	}
}
}
```



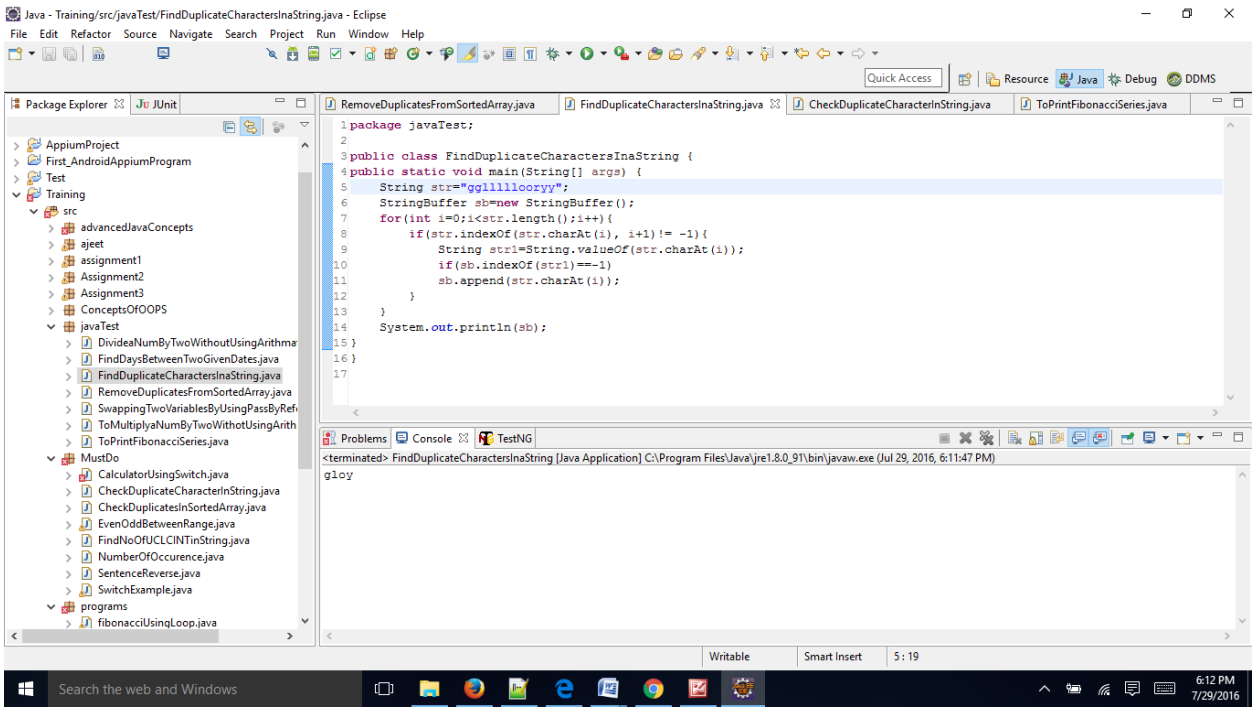

## 12.Write a program to print fibonacci series.

```
package javaTest;

public class ToPrintFibonacciSeries {
static int n1=0,n2=1,n3=0;
public static void fib(int count){
	if(count>0){
		n3=n1+n2;
		n1=n2;
		n2=n3;
		System.out.print(" "+ n3);
		fib(count-1);
	}
}public static void main(String[] args) {
	int count=9;
	System.out.print(n1+" "+n2);
	fib(count-2);
}
}
```



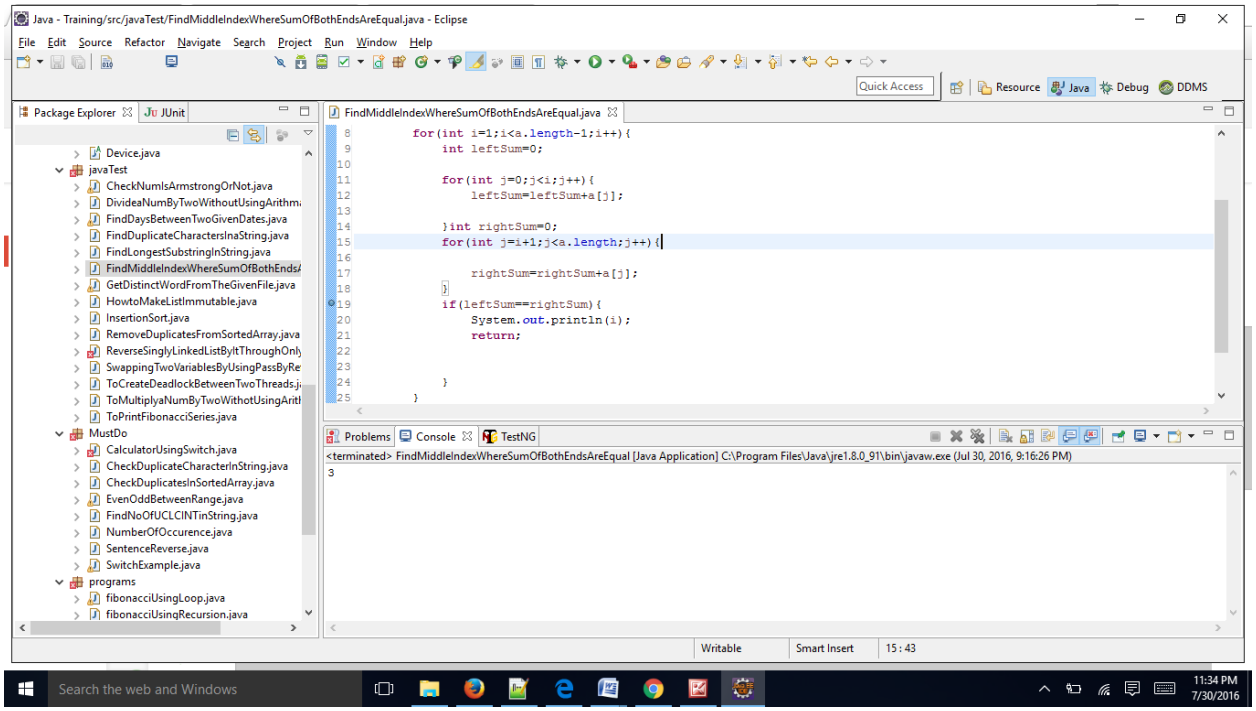

## 13.Write a program to find out duplicate characters in a string

```
package javaTest;

public class FindDuplicateCharactersInaString {
public static void main(String[] args) {
	String str="ggllllllooryy";
	StringBuffer sb=new StringBuffer();
	for(int i=0;i<str.length();i++){
		if(str.indexOf(str.charAt(i), i+1)!= -1){//duplicate character
found
			String str1=String.valueOf(str.charAt(i));
			if(sb.indexOf(str1)==-1)//no duplicate character
			sb.append(str.charAt(i));
		}
	}
	System.out.println(sb);
}
}
```



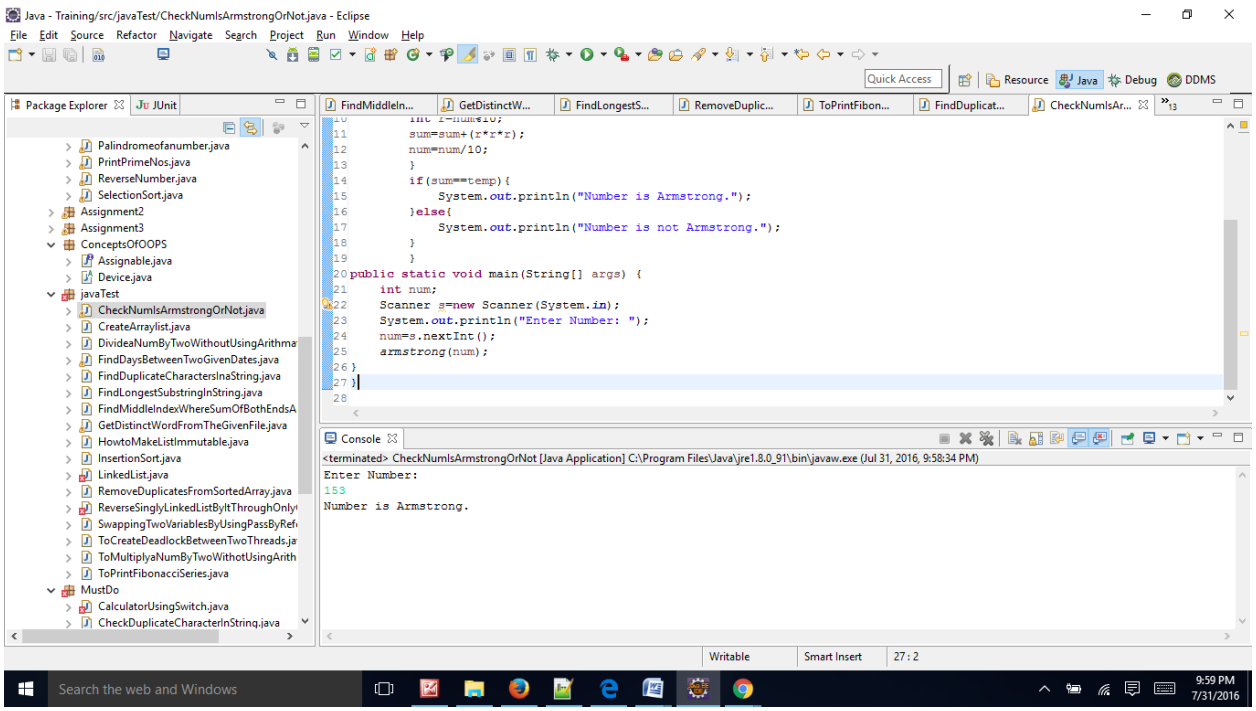

## 15.Find out middle index where sum of both ends are equal

```
package javaTest;

public class FindMiddleIndexWhereSumOfBothEndsAreEqual {
	public static void main(String[] args) {
		int a[]={1,4,3,5,8};


		for(int i=1;i<a.length-1;i++){
			int leftSum=0;

			for(int j=0;j<i;j++){
				leftSum=leftSum+a[j];

			}int rightSum=0;
			for(int j=i+1;j<a.length;j++){

				rightSum=rightSum+a[j];
			}
			if(leftSum==rightSum){
				System.out.println(i);
				return;


			}
		}
	}

}
```




## 16. Write a program to find the given number is Armstrong number or not?

```
package javaTest;

import java.util.Scanner;

public class CheckNumIsArmstrongOrNot {
	public static void armstrong(int num){
		int temp=num;
		int sum=0;
		while(num>0){
		int r=num%10;
		sum=sum+(r*r*r);
		num=num/10;
		}
	  if(sum==temp){
			System.out.println("Number is Armstrong.");
		}else{
			System.out.println("Number is not Armstrong.");
		}
		}
public static void main(String[] args) {
	int num;
	Scanner s=new Scanner(System.in);
	System.out.println("Enter Number: ");
	num=s.nextInt();
    armstrong(num);
}
}
```


Java - Training/src/javaTest/CheckNumIsArmstrongOrNot.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Resource Java Debug DDMS
Package Explorer
JUnit
Palindromeofanumber.java
PrintPrimeNos.java
ReverseNumber.java
SelectionSort.java
Assignment2
Assignment3
ConceptsOfOOPS
Assignable.java
Device.java
javaTest
CheckNumIsArmstrongOrNot.java
CreateArraylist.java
DivideaNumByTwoWithoutUsingArithma
FindDaysBetweenTwoGivenDates.java
FindDuplicateCharactersInaString.java
FindLongestSubstringInString.java
FindMiddleIndexWhereSumOfBothEndsA
GetDistinctWordFromTheGivenFile.java
HowtoMakeListImmutable.java
InsertionSort.java
LinkedList.java
RemoveDuplicatesFromSortedArray.java
ReverseSinglyLinkedListByItThroughOnly
SwappingTwoVariablesByUsingPassByRef
ToCreateDeadlockBetweenTwoThreads.ja
ToMultiplyaNumByTwoWithotUsingArith
ToPrintFibonacciSeries.java
MustDo
CalculatorUsingSwitch.java
CheckDuplicateCharacterInString.java
FindMiddleIn...
GetDistinctW...
FindLongestS...
RemoveDuplic...
ToPrintFibon...
FindDuplicat...
CheckNumIsAr...
sum=sum+(r*r*r);
num=num/10;
}
if(sum==temp){
System.out.println("Number is Armstrong.");
}else{
System.out.println("Number is not Armstrong.");
}
}
public static void main(String[] args) {
int num;
Scanner s=new Scanner(System.in);
System.out.println("Enter Number: ");
num=s.nextInt();
armstrong(num);
}
}
Console
<terminated> CheckNumIsArmstrongOrNot [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (Jul 31, 2016, 9:58:34 PM)
Enter Number:
153
Number is Armstrong.
Writable
Smart Insert
27 : 2
Search the web and Windows
9:59 PM
7/31/2016