

Java Test

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

```
import java.util.*;

public class DayDifference {

    public static void main(String[] args)
    {
        int[] firstdate = new int[3];
        int[] nextdate = new int[3];
        System.out.println("Enter first date in the format yyyy mm dd");
        Scanner s = new Scanner(System.in);
        for(int i =0; i<3;i++)
        {firstdate[i] = s.nextInt();}

        System.out.println("Enter second date in the format yyyy mm dd");
        Scanner t = new Scanner(System.in);
        for(int i =0; i<3;i++)
        {nextdate[i] = t.nextInt();}

        // Creates two calendars instances
        Calendar cal1 = Calendar.getInstance();
        Calendar cal2 = Calendar.getInstance();

        // Set the date for both of the calendar instance
        cal1.set(firstdate[0], firstdate[1], firstdate[2]);
        cal2.set(nextdate[0], nextdate[1], nextdate[2]);

        // Get the represented date in milliseconds
        long millis1 = cal1.getTimeInMillis();
        long millis2 = cal2.getTimeInMillis();

        // Calculate difference in milliseconds
        long diff = millis2 - millis1;

        // Calculate difference in seconds
        long diffSeconds = diff / 1000;

        // Calculate difference in minutes
        long diffMinutes = diff / (60 * 1000);

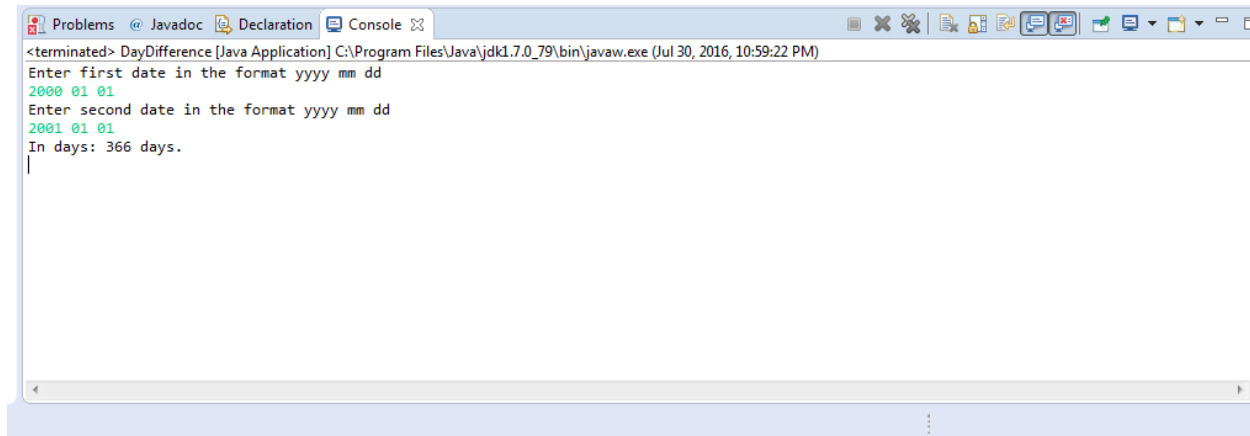
        // Calculate difference in hours
        long diffHours = diff / (60 * 60 * 1000);

        // Calculate difference in days
        long diffDays = diff / (24 * 60 * 60 * 1000);

        System.out.println("In days: " + diffDays + " days.");
    }
}
```

```
}
```

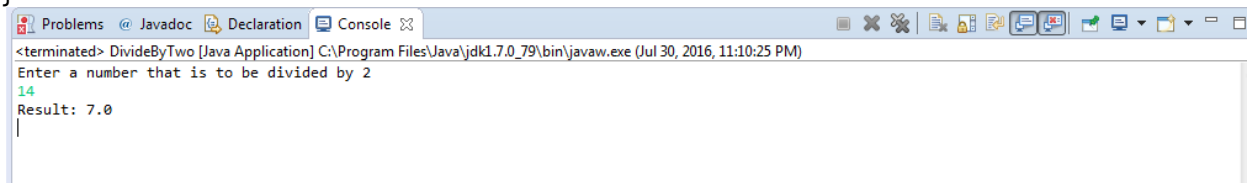
OutPut:



```
<terminated> DayDifference [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 30, 2016, 10:59:22 PM)
Enter first date in the format yyyy mm dd
2000 01 01
Enter second date in the format yyyy mm dd
2001 01 01
In days: 366 days.
```

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

```
3. import java.util.Scanner;
4.
5. public class DivideByTwo {
6.
7.     public static void main(String[] args) {
8.         // TODO Auto-generated method stub
9.         System.out.println("Enter a number that is to be divided by 2");
10.        Scanner s = new Scanner(System.in);
11.        int n = s.nextInt();
12.        double quotient = (n>>1);
13.        System.out.println("Result: " +quotient);
14.
15.    }
16.
17. }
```



```
<terminated> DivideByTwo [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 30, 2016, 11:10:25 PM)
Enter a number that is to be divided by 2
14
Result: 7.0
```

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

```
import java.util.Scanner;

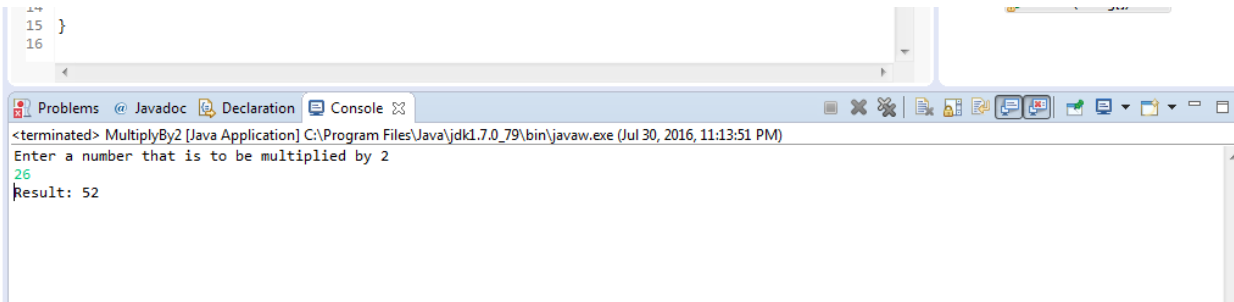
public class MultiplyBy2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter a number that is to be multiplied by 2");
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        int result = (n<<1);
    }
}
```

```

        System.out.println("Result: " + result);
    }
}

```



4. How to swap two variables, by using pass by reference method ?

```

import java.util.Scanner;

public class SwapReference {
    int x;
    SwapReference(int x)
    { this.x = x;}

    void swap(SwapReference s)
    {
        int temp = this.x;
        this.x = s.x;
        s.x = temp;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter 1st number to be swapped");
        Scanner s = new Scanner(System.in);
        int a = s.nextInt();

        System.out.println("Enter 2nd number to be swapped");
        Scanner t = new Scanner(System.in);
        int b = t.nextInt();
        SwapReference s1 = new SwapReference(a);
        SwapReference s2 = new SwapReference(b);

        System.out.println("Before swapping");
        System.out.println("S1 has integer value->" + s1.x);
        System.out.println("S2 has integer value->" + s2.x);

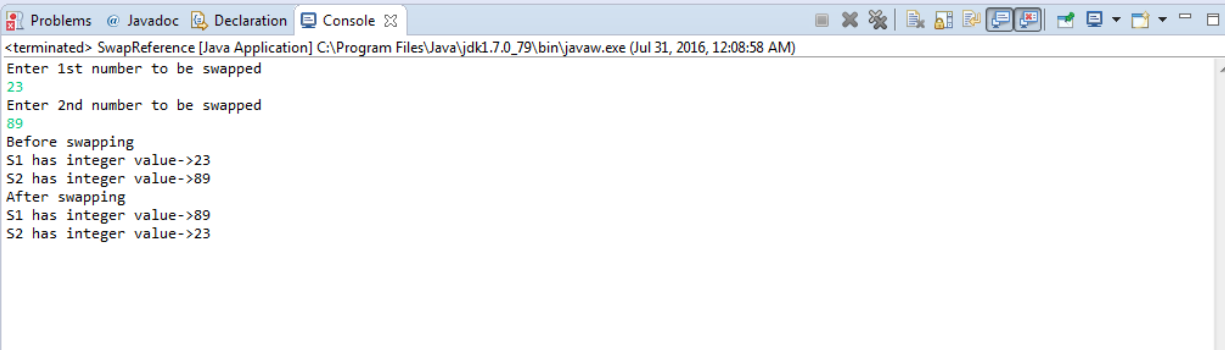
        s1.swap(s2);
        System.out.println("After swapping");
        System.out.println("S1 has integer value->" + s1.x);
        System.out.println("S2 has integer value->" + s2.x);
    }
}

```

```

}
}

```



5. How to make a list immutable?

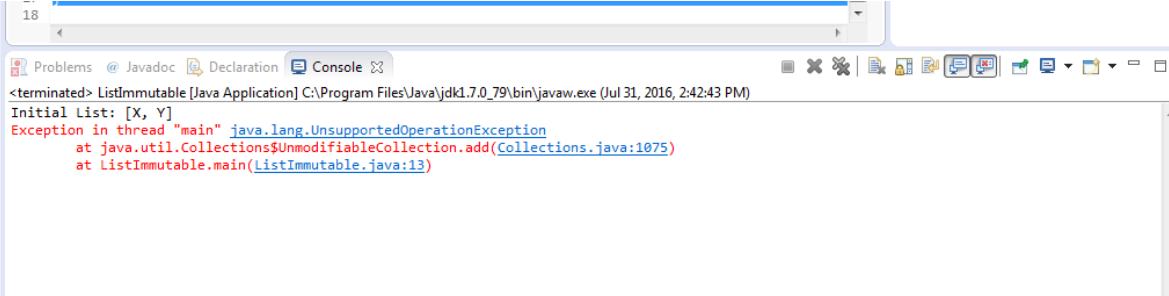
```

import java.util.*;
public class ListImmutable {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        List<Character> list = new ArrayList<Character>();
        list.add('X');
        list.add('Y');
        System.out.println("Initial List: " +list);

        //create a new list
        List<Character> newList = Collections.unmodifiableList(list);
        newList.add('A');
        System.out.println("New List: " +newlist);
    }
}

```



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

```

public class LinkedList {
    public LinkedList next;
    public int value;

    public LinkedList(int value)
    {this.value = value;
    this.next = null; }
}

```

```

public String toString(){
    String data = "";
    LinkedList current = this;
    do{
        data = data + current.value+" ";
        current = current.next;
    }while(current!=null);
    return data;
}
}

```

```

import java.util.*;

public class ReverseLinkedList {
    public static LinkedList reverseList(LinkedList list)
    {
        if(list == null || list.next==null)
        {return list;}

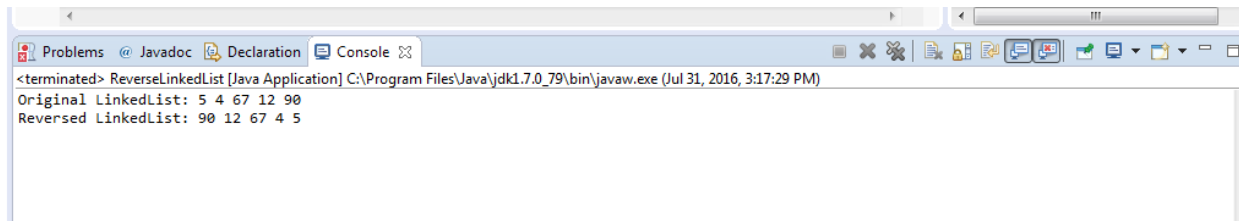
        LinkedList prevNode, curNode, nextNode;
        prevNode = null;
        nextNode = null;
        curNode= list;
        while(curNode!=null)
        {
            nextNode = curNode.next;
            curNode.next = prevNode;
            prevNode = curNode;
            curNode = nextNode;
        }
        return prevNode;    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        LinkedList list = new LinkedList(5);
        list.next = new LinkedList(4);
        list.next.next = new LinkedList(67);
        list.next.next.next = new LinkedList(12);
        list.next.next.next.next = new LinkedList(90);

        System.out.println("Original LinkedList: "+list.toString());
        //reverse the list iteratively

        list = reverseList(list);
        System.out.println("Reversed LinkedList: "+list.toString());
    }
}

```



7. Write a program to implement ArrayList and Linked list

Array list:

```
import java.util.*;

public class ArrayList {
    private Object[] mylist;
    private int size = 0;

    public ArrayList()
    { mylist = new Object[10];}

    public Object get(int index)
    {if(index < size)
        return mylist[index];
    else
        throw new ArrayIndexOutOfBoundsException();}

    public void add(Object o)
    {
        if(mylist.length - size <= 5)
            {increaseSize();} mylist[size++] = o;
    }

    public Object remove(int index)
    {
        if(index < size){
            Object o = mylist[index];
            mylist[index] = null;
            int temp = index;
            while(temp < size)
            {mylist[temp] = mylist[temp+1];
            mylist[temp+1] = null; temp++;}
            size--;
            return o;}
        else
            throw new ArrayIndexOutOfBoundsException();
    }

    public int Size(){return size;}
    private void increaseSize()
    {mylist = Arrays.copyOf(mylist, mylist.length*2);
    System.out.println("\n New Length: " +mylist.length);}

    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

```

        ArrayList al = new ArrayList();
        al.add(new Integer(4));
        al.add(new Integer(12));
        al.add(new Integer(67));
        al.add(new Integer(18));
        al.add(new Integer(23));

        for(int i=0;i<al.size;i++)
            {System.out.println(al.get(i)+" ");}
        al.add(new Integer(31));
        al.remove(2);
        System.out.println("Array list after adding element 31 and removing
element at index 2 ");
        for(int i=0;i<al.size;i++)
            System.out.println(al.get(i)+" ");
    }}

```

```

<terminated> ArrayList [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 3:59:52 PM)
4
12
67
18
23

New Length: 20
Array list after adding element 31 and removing element at index 2
4
12
18
23
31

```

LinkedList

```

public class Node<T> implements Comparable<T> {
    private T value;
    private Node<T> nextRef;

    public T getValue(){
        return value;}

    public void SetValue(T value)
    {this.value = value;}

    public Node<T> getNextRef(){return nextRef;}
    public void setNextRef(Node<T> ref){this.nextRef = ref;}

    public int compareTo(T arg){
        if(arg==this.value)
        {return 0;}
        else {return 1;}
    }
}

public class SinglyLinkedListImpl<T> {

    private Node<T> head;
    private Node<T> tail;
}

```

```

public void add(T element){

    Node<T> nd = new Node<T>();
    nd.SetValue(element);
    System.out.println("Adding: "+element);
    /**
     * check if the list is empty
     */
    if(head == null){
        //since there is only one element, both head and
        //tail points to the same object.
        head = nd;
        tail = nd;
    } else {
        //set current tail next link to new node
        tail.setNextRef(nd);
        //set tail as newly created node
        tail = nd;
    }
}

public void addAfter(T element, T after){

    Node<T> tmp = head;
    Node<T> refNode = null;
    System.out.println("Traversing to all nodes..");
    /**
     * Traverse till given element
     */
    while(true){
        if(tmp == null){
            break;
        }
        if(tmp.compareTo(after) == 0){
            //found the target node, add after this node
            refNode = tmp;
            break;
        }
        tmp = tmp.getNextRef();
    }
    if(refNode != null){
        //add element after the target node
        Node<T> nd = new Node<T>();
        nd.SetValue(element);
        nd.setNextRef(tmp.getNextRef());
        if(tmp == tail){
            tail = nd;
        }
        tmp.setNextRef(nd);
    } else {
        System.out.println("Unable to find the given element...");
    }
}

```



```

public void deleteFront(){
    if(head == null){
        System.out.println("Underflow...");
    }
    Node<T> tmp = head;
    head = tmp.getNextRef();
    if(head == null){
        tail = null;
    }
    System.out.println("Deleted: "+tmp.getValue());
}

public void deleteAfter(T after){
    Node<T> tmp = head;
    Node<T> refNode = null;
    System.out.println("Traversing to all nodes..");
    /**
     * Traverse till given element
     */
    while(true){
        if(tmp == null){
            break;
        }
        if(tmp.compareTo(after) == 0){
            //found the target node, add after this node
            refNode = tmp;
            break;
        }
        tmp = tmp.getNextRef();
    }
    if(refNode != null){
        tmp = refNode.getNextRef();
        refNode.setNextRef(tmp.getNextRef());
        if(refNode.getNextRef() == null){
            tail = refNode;
        }
        System.out.println("Deleted: "+tmp.getValue());
    } else {
        System.out.println("Unable to find the given element...");
    }
}

public void traverse(){
    Node<T> tmp = head;
    while(true){
        if(tmp == null){
            break;
        }
        System.out.println(tmp.getValue());
        tmp = tmp.getNextRef();
    }
}

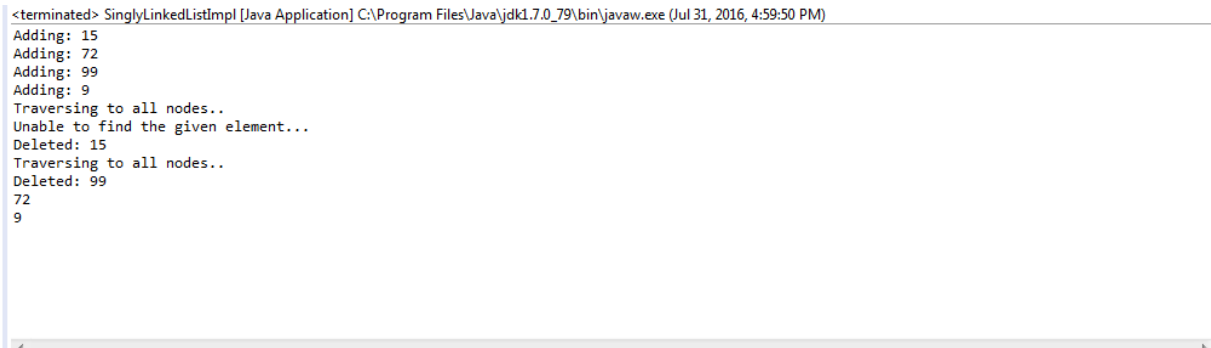
```

```

    }

    public static void main(String a[]){
        SinglyLinkedListImpl<Integer> sl = new SinglyLinkedListImpl<Integer>();
        sl.add(15);
        sl.add(72);
        sl.add(99);
        sl.add(9);
        sl.addAfter(99, 54);
        sl.deleteFront();
        sl.deleteAfter(72);
        sl.traverse();
    }
}

```



```

<terminated> SinglyLinkedListImpl [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 4:59:50 PM)
Adding: 15
Adding: 72
Adding: 99
Adding: 9
Traversing to all nodes..
Unable to find the given element...
Deleted: 15
Traversing to all nodes..
Deleted: 99
72
9

```

8. Write a program for Insertion Sort in java.

```

public class InsertionSort {

    public static void printNumbers(int[] num){
        for(int i=0;i<num.length;i++){
            System.out.print(num[i]+" ");
        }
        System.out.println();
    }

    public static void insertSort(int[] input)
    {
        int n = input.length;
        for(int j=1;j<n;j++){
            int key = input[j];
            int i = j -1;
            while((i>-1) && (input[i]>key))
            { input[i+1] = input[i];
              i--;}
            input[i+1] = key;
            printNumbers(input);
        }
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}

```

```

        int[] input = {12, 7, 34, 18, 41, 35, 51};
        insertSort(input);
    }
}

```

```

<terminated> InsertionSort [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 5:20:42 PM)
7 12 34 18 41 35 51
7 12 34 18 41 35 51
7 12 18 34 41 35 51
7 12 18 34 41 35 51
7 12 18 34 35 41 51
7 12 18 34 35 41 51
7 12 18 34 35 41 51

```

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

```

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.StringTokenizer;

public class MyDistinctFileWords {

    public List<String> getDistinctWordList(String fileName){

        FileInputStream fis = null;
        DataInputStream dis = null;
        BufferedReader br = null;
        List<String> wordList = new ArrayList<String>();
        try {
            fis = new FileInputStream(fileName);
            dis = new DataInputStream(fis);
            br = new BufferedReader(new InputStreamReader(dis));
            String line = null;
            while((line = br.readLine()) != null){
                StringTokenizer st = new StringTokenizer(line, " ,.:\\");
                while(st.hasMoreTokens()){
                    String tmp = st.nextToken().toLowerCase();
                    if(!wordList.contains(tmp)){
                        wordList.add(tmp);
                    }
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally{
            try{if(br != null) br.close();}catch(Exception ex){}

```

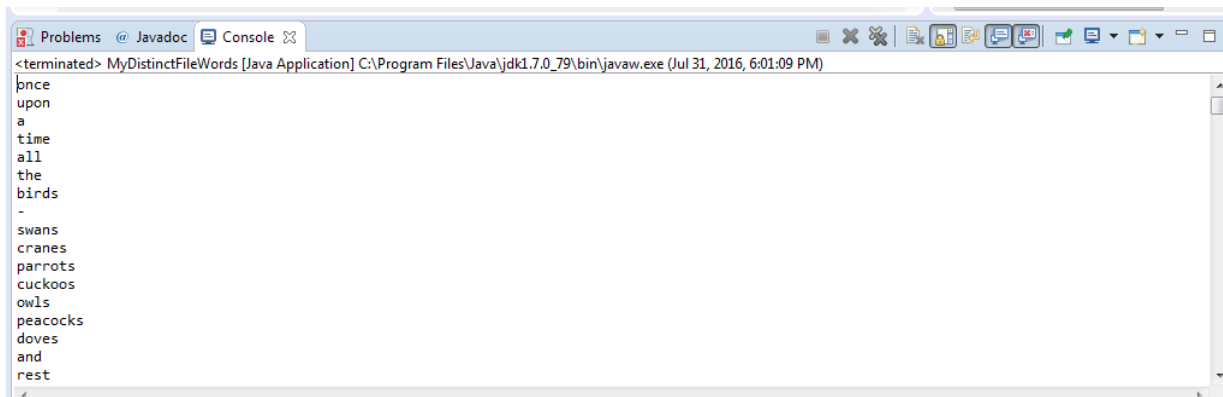
```

    }
    return wordList;
}

public static void main(String a[]){

    MyDistinctFileWords distFw = new MyDistinctFileWords();
    List<String> wordList =
distFw.getDistinctWordList("C:/Users/DELL/Documents/Sample.txt");
    for(String str:wordList){
        System.out.println(str);
    }
}
}

```



```

<terminated> MyDistinctFileWords [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 6:01:09 PM)
once
upon
a
time
all
the
birds
-
swans
cranes
parrots
cuckoos
owls
peacocks
doves
and
rest

```

10. Find longest substring without repeating characters.

11. Write a program to remove duplicates from sorted array

```

import java.util.Arrays;

public class DuplicateArray {
    public int[] removeDuplicates(int a[]) {
        int previousElement = a[0];
        int pos = 0;
        for(int i=1; i < a.length; i++) {
            int currentElement = a[i];
            if((currentElement ^ previousElement) == 0) {
                // Same Element again
            }
            else {
                previousElement = currentElement;
                pos++;
                a[pos] = currentElement;
            }
        }
        return Arrays.copyOf(a, pos+1);
    }

    public static void main(String []args){
        int a[] = {1,2,2,2,3,3,3,4,4,5};
        int b[] = new DuplicateArray().removeDuplicates(a);
    }
}

```

```

    for(int cur: b) {
        System.out.println(cur);
    }
}
}

```

```

20
21 public static void main(String []args){
22     int a[] = {1,2,2,2,3,3,3,4,4,5};
23     int b[] = new DuplicateArray().removeDuplicates(a);
24     for(int cur: b) {
25         System.out.println(cur);
26     }
27 }
28 }

```

Problems @ Javadoc Console

<terminated> DuplicateArray [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 7:29:02 PM)

```

1
2
3
4
5

```

12. Write a program to print fibonacci series.

```

import java.util.Scanner;

public class Fibonacci {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int length = 15;
        int[] fibcount = new int[length];
        fibcount[0] = 0;
        fibcount[1] = 1;
        for(int i =2; i<length;i++)
        {
            fibcount[i] = fibcount[i-1] + fibcount[i-2];
        }
        System.out.println("Fibonacci Series");
        for (int i =0;i<length;i++)
            System.out.print(" " +fibcount[i]);
    }
}

```

Problems @ Javadoc Console

<terminated> Fibonacci [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Jul 31, 2016, 7:33:31 PM)

```

Fibonacci Series
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

```

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

```

public class DuplicateChar {

```

```

public void findDup(String str){
    Map<Character, Integer> dupchar = new HashMap<Character, Integer>();
    char[] chrs = str.toCharArray();
    for(Character ch:chrs){
        if(dupchar.containsKey(ch)){
            dupchar.put(ch, dupchar.get(ch)+1);
        } else {
            dupchar.put(ch, 1);
        }
    }
    Set<Character> keys = dupchar.keySet();
    for(Character ch:keys){
        if(dupchar.get(ch) > 1){
            System.out.println(ch+"--->"+dupchar.get(ch));
        }
    }
}

public static void main(String a[]){
    DuplicateChar dcs = new DuplicateChar();
    dcs.findDup("application");
}
}

```

```

26 public static void main(String a[]){
27     DuplicateChar dcs = new DuplicateChar();
28     dcs.findDup("application");
29 }
30 }
31
32

```

Problems @ Javadoc Console

<terminated> DuplicateChar [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (Aug 1, 2016, 2:58:10 AM)

```

p--->2
a--->2
i--->2

```

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

```

import java.util.Scanner;

public class ArmstrongCheck{
    public static void main(String[] args) {
        int c=0,a,temp;
        System.out.println("Enter a number to check if it is an Armstrong number");
        Scanner s = new Scanner(System.in);

        int n=s.nextInt();
        temp=n;
        while(n>0)
        {

```

```
a=n%10;
n=n/10;
c=c+(a*a*a);
}
if(temp==c)
System.out.println("armstrong number");
else
    System.out.println("Not armstrong number");
}
}
```

