1. An array of 40 elements contains unique numbers between 1 to 100. Write a java program to find out the missing numbers which are not part of array.

import java.util.Arrays;

public class MyClass {

    public static void main(String args[]) {

         int ar[]= {1,3,4,5,8,10,15,20,42,43,44,45,46,47,50,54,55,56,26,33,34,35,36,37};

           Arrays.sort(ar);

       for (int j = 0; j < ar.length; j++) {

           System.out.println(ar[j] +" "); }

     System.out.println("\nMissing elements between 1 to 100 in array :");

        int j=0;

    for(int i=1;i<=100;i++){

      if(j<ar.length && i==ar[j])

     {  j++;  }

    else

     { System.out.println(i+" "); }

}     }

}

 2.Write a java program to find the rth highest number in an array of n

public class NthLargestInArrayExample{

public static int getNthLargest(int[] a, int total,int N){

   int temp;

for(int i=0;i<total;i++)

{

for(int j=i+1;j<total;j++)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

return a[total-N];

}

public static void main(String args[]){

int a[]={1,2,5,6,3,2},N=4;

System.out.println("Nth Largest:"+getNthLargest(a,6,N));

}}

3.Write a program to find the sum of first n Fibonacci numbers where the 1st 2 numbers are 3 and 7

public class Fibonacci {

    public static void main(String[] args) {

        int n = 10, t1 = 3, t2 = 7,add=0;

        System.out.print("First " + n + " terms: ");

        for (int i = 1; i <= n; ++i)

        {

            System.out.print(t1+" " );

            int sum = t1 + t2;

            add=add+t1;

            t1 = t2;

            t2 = sum;

        }

       System.out.print("Sum of the n numbers:"+add);

    }

}

4. Write a program to add the sum of first n non-prime numbers .

import java.util.\*;

public class NonPrimeNumber

{

static int maxValue(int arr[])

{

    int max\_e = Integer.MIN\_VALUE;

    for(int i = 0; i < arr.length; i++)

    {

    max\_e = Math.max(max\_e, arr[i]);

    }

    return max\_e;

}

static int nonPrimeSum(int arr[], int n)

{

    int max\_val = maxValue(arr);

    boolean p1[] = new boolean[max\_val + 1];

    for(int i = 0; i < p1.length; i++) {

    p1[i] = true; }

    p1[0] = false;

    p1[1] = false;

    for (int p = 2; p \* p <= max\_val; p++)

    {

        if (p1[p] == true)

        {

            for (int i = p \* 2; i <= max\_val; i += p) {

                p1[i] = false; }

        }

    }

    int sum = 0;

    for (int i = 0; i < n; i++)

        if (!p1[arr[i]]) {

         System.out.println(arr[i]);

            sum += arr[i]; }

    return sum;

}

public static void main(String args[])

{

  int arr[] = { 1, 4, 7, 12, 9, 8 };

    int n = arr.length;

    System.out.println( nonPrimeSum(arr, n));

}

}

 5. Write a java program to print odd even numbers using two threads

ThreadsConcept.Java

public class ThreadsConcept {

    public static void main(String... args) {

    PrintData print = new PrintData();

        Thread t1 = new Thread(new ThreadOneEven(print, 20, false));

        Thread t2 = new Thread(new ThreadOneEven(print, 20, true));

        t1.start();

        t2.start();

    }

}

PrintData.Java

 class PrintData {

 boolean isOdd = false;

    synchronized void printEven(int number) {

        while (isOdd == false) {

            try {

                wait();

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

        System.out.println("Even:" + number);

        isOdd = false;

        notifyAll();

    }

    synchronized void printOdd(int number) {

        while (isOdd == true) {

            try {

                wait();

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

        System.out.println("Odd:" + number);

        isOdd = true;

        notifyAll();

    }

}

ThreadOneEven.Java

 class ThreadOneEven implements Runnable {

 private int max;

    private PrintData print;

    private boolean isEvenNumber;

    ThreadOneEven(PrintData print, int max, boolean isEvenNumber) {

        this.print = print;

        this.max = max;

        this.isEvenNumber = isEvenNumber;

    }

    @Override

    public void run() {

        //System.out.println("Run method");

        int number = isEvenNumber == true ? 2 : 1;

        while (number <= max) {

            if (isEvenNumber) {

                //System.out.println("Even :"+ Thread.currentThread().getName());

                print.printEven(number);

                //number+=2;

            } else {

                //System.out.println("Odd :"+ Thread.currentThread().getName());

                print.printOdd(number);

                // number+=2;

            }

            number += 2;

        }

    }

}

6.Write a java program to demonstrate java serialization

import java.io.\*;

 class Sample implements java.io.Serializable

{

    public int a;

    public String b;

    public Sample(int a, String b)

    {

        this.a = a;

        this.b = b;

    }

}

public class TestSample

{

    public static void main(String[] args)

    {

        Sample object = new Sample(1, "geeksforgeeks");

        String filename = "Samplefile.ser";

  // Serialization

        try

        {

            //Saving of object in a file

            FileOutputStream file = new FileOutputStream(filename);

            ObjectOutputStream out = new ObjectOutputStream(file);

            // Method for serialization of object

            out.writeObject(object);

            out.close();

            file.close();

            System.out.println("Object has been serialized");

        }

        catch(IOException ex)

        {

            System.out.println("IOException is caught");

        }

        Sample object1 = null;

        // Deserialization

        try

        {

            // Reading the object from a file

            FileInputStream file = new FileInputStream(filename);

            ObjectInputStream in = new ObjectInputStream(file);

            // Method for deserialization of object

            object1 = (Sample)in.readObject();

            in.close();

            file.close();

            System.out.println("Object has been deserialized ");

            System.out.println("a = " + object1.a);

            System.out.println("b = " + object1.b);

        }

        catch(IOException ex)

        {

            System.out.println("IOException is caught");

        }

        catch(ClassNotFoundException ex)

        {

            System.out.println("ClassNotFoundException is caught");

        }

    }

}

7.Write a java program to modify a given json file

import java.io.FileWriter;

import java.io.IOException ;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

public class JSONExample

{

    @SuppressWarnings("unchecked")

    public static void main( String[] args )

    {

        JSONObject employeeDetails = new JSONObject();

        employeeDetails.put("firstName", "Lokesh");

        employeeDetails.put("lastName", "Gupta");

        employeeDetails.put("website", "howtodoinjava.com");

        JSONObject employeeObject = new JSONObject();

        employeeObject.put("employee", employeeDetails);

        JSONObject employeeDetails2 = new JSONObject();

        employeeDetails2.put("firstName", "Brian");

        employeeDetails2.put("lastName", "Schultz");

        employeeDetails2.put("website", "example.com");

        JSONObject employeeObject2 = new JSONObject();

        employeeObject2.put("employee", employeeDetails2);

        JSONArray employeeList = new JSONArray();

        employeeList.add(employeeObject);

        employeeList.add(employeeObject2);

        try (FileWriter file = new FileWriter("employees.json")) {

            file.write(employeeList.toJSONString());

            file.flush();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

8.Write a java program to count the number of unique words in a paragraph and to find the count of each word in the same paragraph

// Java program to print unique words

// from a string

import java.util.HashMap;

import java.util.Iterator;

import java.util.Set;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Sample

{

    static void printUniquedWords(String str)

    {

        Pattern p = Pattern.compile("[a-zA-Z]+");

        Matcher m = p.matcher(str);

        HashMap<String, Integer> hm = new HashMap<>();

        while (m.find())

        {

            String word = m.group();

            if(!hm.containsKey(word))

                hm.put(word, 1);

            else

                hm.put(word, hm.get(word) + 1);

        }

        Set<String> s = hm.keySet();

        Iterator<String> itr = s.iterator();

        int counter =0;

        System.out.println("Count of Each word");

        while(itr.hasNext())

        {

            String w = itr.next();

              System.out.println(w+"-"+hm.get(w));

                counter++;

        }

        System.out.println("Total No of unique words in paragraph:"+counter);

    }

    public static void main(String[] args)

    {

        String str = "If you think you can do it, you can do it.";

        printUniquedWords(str);

    }

}