Program 1:Exercise 1: Create a class with a method which can calculate the sum of first n natural numbers which are divisible by 3 or 5.

Method Name	calculateSum
Method Description	Calculate Sum
Argument	int n
Return Type	int-sum
Logic	Calculate the sum of first n natural numbers which are divisible by 3 or 5.

```
package coma.example;
import java.util.Scanner;
class Findsum
```

```
{
	int sum=0;
	void calculatesum(int n)
	{
		for(int i=1;i<=n;i++)

	{
		if(i%3==0||i%5==0)
		{
			sum=sum+i;
		}
	}
		System.out.println(sum);
	}
}
public class Sum1 {
	public static void main(String[] args) {
		int x;

		Scanner sc=new Scanner(System.in);
		System.out.println("\n enter range");
		x=sc.nextInt();
		Findsum f=new Findsum();
		f.calculatesum(x);
	}
}
```

Program2:Exercise 2: Create a class with a method to find the difference between the sum of the squares and the square of the sum of the first n natural numbers.

Method Name	calculateDifference
Method Description	Calculate the difference
Argument	int n
Return Type	int - Sum
Logic	Find the difference between the sum of the squares of the first n natural numbers and the square of their sum. For Example if n is 10,you have to find (1^2+2^2+3^2+….9^2+10^2)-(1+2+3+4+5…+9+10)^2

```
package coma.example;
import java.util.Scanner;

class Cal
{
	int sum=0;
	int sum1=0;
	int sum2=0;int k;
	void calculateDifference(int n)
	{
		for(int i=1;i<=n;i++)
		{
			sum=(int) (sum+Math.pow(i,2));
			sum1=sum1+i;
		}
```

```
            sum2=(int) (Math.pow((sum1),2));
            k=sum-sum2;
            System.out.println(k);
        }
}
public class Sum2 {
        public static void main(String[] args) {

            int x;
            Scanner sc=new Scanner(System.in);
            System.out.println("\n enter range");
            x=sc.nextInt();
            Cal c=new Cal();
            c.calculateDifference(x);

        }

}
```

Program3:Exercise 3: Create a class containing a method to create the mirror image of a String. The method should return the two Strings separated with a pipe(|) symbol .

Method Name	getImage
Method Description	Generate the mirror image of a String and add it to the existing string.
Argument	String
Return Type	String
Logic	Accepts One String Find the mirror image of the String Add the two Strings together separated by a pipe(|) symbol.

	For Example Input : EARTH Output : EARTH|HTRAE Hint: Use StringBuffer API (Ex: For this problem reverse method in Stringbuffer can be used) Note: Learn the other APIs in StringBuffer

```
package coma.example;
import java.util.Scanner;
class GetImage
{
	String temp="";
	void getImage(String s)
	{
		int l=s.length();

		for(int i=l-1;i>=0;i--)
		{
			temp=temp+s.charAt(i);
		}
		System.out.println(s+"|"+temp);
	}
}
public class Mirror3 {
	public static void main(String[] args) {
		String a;
		Scanner sc=new Scanner(System.in);
		System.out.println("\n enter string");
		a=sc.nextLine();

		GetImage g=new GetImage();
		g.getImage(a);
	}
```

}

Program4:Exercise 4: Create a method to check if a number is an increasing number

Method Name	checkNumber
Method Description	Check if a number is an increasing number
Argument	int number
Return Type	boolean
Logic	A number is said to be an increasing number if no digit is exceeded by the digit to its left. For Example : 134468 is an increasing number

```
package coma.example;
import java.util.Scanner;

class Chknum
{
	boolean checkNumber(String n)
	{
		int l=n.length();
		//String t;
		int i=l-1;
		while(((n.charAt(i))>(n.charAt(i-1))) && i>1)
			i=i-1;
		return(i==1);
	}
}
```

```
public class Num4 {
	public static void main(String[] args) {
		String x;
		boolean a;
		Scanner sc=new Scanner(System.in);
		System.out.println("\n enter number");
		x=sc.nextLine();
		Chknum c=new Chknum();
		a=c.checkNumber(x);
		if(a)
			System.out.println("the given number is in increasing
order");
		else
			System.out.println("the given number is not in increasing
order");

	}

}
```

Program5:Example 5: Create a method to check if a number is a power of two or not

Method Name	checkNumber
Method Description	Checks if the entered number is a power of two or not
Argument	int n

Return Type	boolean
Logic	Check if the input is a power of two. Ex: 8 is a power of 2

```
package coma.example;
import java.util.Scanner;

class Chcknum
{

int k;
	boolean checknumber(int n)
	{

		while((n%2==0)&&n>1)
			n=n/2;
			return (n==1);

	}
}

public class Pow5 {
	public static void main(String[] args)
	{
		int x;
		boolean a;
		Scanner sc=new Scanner(System.in);
		System.out.println("\n enter number");
		x=sc.nextInt();
		Chcknum c=new Chcknum();
		a=c.checknumber(x);
		if(a)
			System.out.println("the given number is power of 2");
		else
```

```
                System.out.println("the given number is not a power of
2");

        }

}
```

Program6:Example 6: A school offers medals to the students of tenth based on the following criteria
If(Marks>=90) : Gold
If(Marks between 80 and 90) : Silver
If(Marks between 70 and 80) : Bronze
Note: Marks between 80 and 90 means → marks>=80 and marks<90
Write a function which accepts the marks of students as a Hashmap and return the details of the students eligible for the medals along with type of medal.
The input hashmap contains the student registration number as key and mark as value.
The output hashmap should contain the student registration number as key and the medal type as value.

Method Name	getStudents
Method Description	Generate the list of students eligible for scholarship
Argument	Hashmap
Return Type	Hashmap
Logic	The method should return the details of the students eligible for the medals along with the medal type.

```
package com.cognizant.geometry;
import java.util.HashMap;
```

```
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
public class StuMedalHashMap6 {
	static Map<Integer, String> stumedal(Map<Integer, Integer> m){
		Map<Integer, String> rm=new HashMap<Integer, String>();
		Set<Integer> s=m.keySet();
		Iterator<Integer> it=s.iterator();
		int id;
		while(it.hasNext()){
			id=it.next();
			int  n=m.get(id);
			System.out.println(n);
			int nn=n/10;
			System.out.println(nn);
			switch(nn){
			case 9:	rm.put(id,"Gold");			break;
			case 8:  rm.put(id, "Silver");		break;
			case 7:	rm.put(id,"Bronze");			break;
		}
		}return rm;
	}
	public static void main(String[] args) {
		HashMap<Integer,Integer> h=new HashMap<Integer,
Integer>();
		h.put(121,90);		h.put(122,80);
		h.put(123,70);		h.put(124,60);
		h.put(125,50);		h.put(126,98);
		h.put(127,85);		h.put(128,77);
		h.put(129,69);
		System.out.println(h);
		Map<Integer, String>res=StuMedalHashMap6.stumedal(h);
		System.out.println(res);
	}
}
```

Program9:

Example 9: Create a method which can perform the following operations on two String objects S1 and S2. The output of each operation should be added to an arraylist and the arraylist should be returned.(Assume S2 is of smaller size)

Examples for below statements are shown in the Logic part

1. Character in each alternate index of S1 should be replaced with S2
2. If S2 appears more than once in S1, replace the last occurrence of S2 in S1 with the reverse of S2, else return S1+S2
3. If S2 appears more than once in S1, delete the first occurrence of S2 in S1, else return S1
4. Divide S2 into two halves and add the first half to the beginning of the S1 and second half to the end of S1.

Note: If there are odd number of letters in S2, then add (n/2)+1 letters to the beginning and the remaining letters to the end. (n is the number of letters in S2)

5. If S1 contains characters that is in S2 change all such characters to *

Method Name	modifyStrings
Method Description	Perform the above mentioned actions on a String
Argument	String,String
Return Type	Arraylist
Logic	Do the above mentioned actions on the entered String. For Example S1=”JAVAJAVA” S2=”VA’ 1. **VA**A**VA**A**VA**A**VA**A (J replaced with VA, V replaced with VA etc.) 2. JAVAJAAV 3. JAJAVA 4. VJAVAJAVAA

package com.cognizant.geometry;

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class StrRStr9 {
	/**
	 * @param args
	 * @return
	 */
	//ArrayList<String>
	List<String>strrstr(String f, String s){
		StringBuffer sb=new StringBuffer(f);
		StringBuffer sb1=new StringBuffer(f);
		StringBuffer sb2=new StringBuffer(s).reverse();
		List<String> l=new ArrayList<String>();
		//char c[]=f.toCharArray();
		//char d[]=s.toCharArray();
		for(int i=0;i<sb.length();i=i+(s.length()+1)){
			sb=sb.replace(i,i+1,s);
		}
		System.out.println(sb+"    ");
		String ss=sb.toString();
		String s1="";
		l.add(ss);
		s1=matchLast(f,s);
		l.add(s1);
		s1=matchfirst(f, s);
		l.add(s1);
		s1=firLastapp(f, s);
		l.add(s1);
		s1=charCh(f, s);
		l.add(s1);
		return l;
	}
	String matchLast(String f, String s){
		StringBuffer sb1=new StringBuffer(f);
```

```
            StringBuffer sb2=new StringBuffer(s).reverse();
            int cnt=0,j=0;
            String s1="";
            for(int i=0;i<sb1.length();i=1+j){
                  System.out.println("i :"+i);
                  j=sb1.indexOf(s,i);
                  System.out.println("j :"+j);
                  if(j>0){
                        System.out.println(cnt);
                        cnt++;
                  }else{
                        break;
                  }
            }
            if(cnt>1){

      sb1=sb1.replace(sb1.lastIndexOf(s),sb1.lastIndexOf(s)+s.length(),sb2.
toString());
                  s1=sb1.toString();
            }else{
                  s1=f+s;
            }
            System.out.println(cnt);
            System.out.println(s1);
            return s1;
      }
      String matchfirst(String f, String s){
            StringBuffer sb1=new StringBuffer(f);
            StringBuffer sb2=new StringBuffer(s).reverse();
            int cnt=0,j=0;
            String s1="";
            for(int i=0;i<sb1.length();i=1+j){
                  System.out.println("i :"+i);
                  j=sb1.indexOf(s,i);
                  System.out.println("j :"+j);
                  if(j>0){
```

```
                System.out.println(cnt);
                cnt++;
            }else{
                break;
            }
        }
        if(cnt>1){

sb1=sb1.replace(sb1.indexOf(s),sb1.indexOf(s)+s.length(),"");
            s1=sb1.toString();
        }else{
            s1=f;
        }
        System.out.println(cnt);
        System.out.println(s1);
        return s1;

}
String firLastapp(String f,String s){
        int n=s.length();
        String s1="";
        if(n%2==0){
            s1=s.substring(0,n/2)+f+s.substring(n/2,n);
        }else{
            s1=s.substring(0, n/2+1)+f+s.substring(n/2+1,n);
        }
        System.out.println(s1);
        return s1;
}
String charCh(String f,String s){
        //for(int i=0;i<s.length();i++){
//      for(int j=0;j<f.length();j++){
        //      int n=f.indexOf(s.charAt(i),j);
            //if(n>0){
        for(int i=0;i<s.length();i++){
            f=f.replace(s.charAt(i),'*');
```

```
            }
            System.out.println(f);
        //      f=f.replace(start, end, str)
        return f;
        }
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            Scanner sc=new Scanner(System.in);
            System.out.println("enter the first string");
            String s1=sc.next();
            System.out.println("enter the second string");
            String s2=sc.next();
            StrRStr9 st=new StrRStr9();
            List<String>ll=st.strrstr(s1, s2);
            System.out.println(ll);
        }
}
```

Program10:

Example 10: Create a method that accepts a number and modifies it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number. The digit in the units place can be left as it is.

Note: Take the absolute value of the difference. Ex: 6-8 = 2

Method Name	modifyNumber
Method Description	Accepts a number and modify it as per the requirement
Argument	int number1
Return Type	int
Logic	Accept a number and modify it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number.

	For example. Input: 45862 Output:13242 **Algorithm:** Convert number into String Extract each char using charAt method Convert char to int and find the difference Create new StringBuffer object and keep adding the difference Finally convert StringBuffer to int

```
package com.cognizant.geometry;

import java.util.Scanner;

public class NumMod10 {
	/**
	 * @param args
	 */
	String cpy;
	char a1,b;
	int modifyNUm(int a){
		System.out.println("Given input is :"+a);
		cpy=Integer.toString(a);
		int len=cpy.length();
		//System.out.println(len);
		int f,s,r;
		//String resc;
		String s1="",resc="";
		for(int i=0;i<len-1;i++){
			a1=cpy.charAt(i);
			b=cpy.charAt(i+1);
```

```
                f=Character.getNumericValue(a1);
                s=Character.getNumericValue(b);
                r=Math.abs(f-s);
                if(r==0){
                        resc+="0";

                }else{
                        s1=Integer.toString(r);
                        resc+=s1;
                }
                //System.out.println(resc);
        }

        return Integer.parseInt(resc);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("ente any num ");
        int num=sc.nextInt();
        NumMod10 n=new NumMod10();
        int a=n.modifyNUm(num);
        System.out.println("Output with adjacent diff num is:"+a);
    }
}
```

Program11:

Example 11: Create a method which accepts the date of birth of person and date format and print the day (SUNDAY, MONDAY…) on which he was born.

Note: The output should be in upper case

Method Name	getDayofWeek
Method Description	Finds the day of the week in which a person is born
Argument	String date, String dateFormat
Return Type	String – Day of week
Logic	Use Calendar API and switch case to get the day of the week Ex: Input1 = 25/06/2012 Input2 = dd/MM/yyyy Output= MONDAY

```
package com.cognizant.geometry;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class DateTOWeekName11 {

	/**
	 * @param args
	 * @throws ParseException
	 */
	public static void main(String[] args) throws ParseException {
		// TODO Auto-generated method stub
		Scanner sc=new Scanner(System.in);
		System.out.println("enter the date");
		String s=sc.next();
		//Date d=new Date(s);
		System.out.println("enter the format");
		String f=sc.next();
		SimpleDateFormat sdf=new SimpleDateFormat(f);
		Date sf=sdf.parse(s);
		System.out.println(sf);
		SimpleDateFormat sdf2=new SimpleDateFormat("EEEE");
		System.out.println(sdf2.format(sf));

	}

}
```

Program12:

Example 12: You are asked to create an application for registering the details of jobseeker. The requirement is:
Username should always end with **_job** and there should be atleast minimum of 8 characters to the left of **_job**. Write a function to validate the same. Return true in case the validation is passed. In case of validation failure return false.

Method Name	validateUserName
Method Description	Checks if the username is valid
Argument	String userName
Return Type	boolean
Logic	Checks if the username ends with _job and contains at least 8 characters to the left of _job. If valid return true. Else return false.

```
package com.cognizant.geometry;

import java.util.Scanner;

public class StrValidate12 {
	/**
	 * @param args
	 */
	void validateUserName(String n){
		if(n.endsWith("_job")){
			int l=n.length();
			String ln=n.substring(0,l-4);
			//System.out.println(ln);
			//System.out.println(l);
			if(ln.length()>=8){
				System.out.println("valid input: "+n);
			}
```

```
                else{
                        System.out.println("Invalid input:\t'" +n +"'\t must
contain altleast 8 chars before _job");
                }
        }
        else{
                System.out.println("Invalid input:\t'" +n +"'\t must end
with _job");
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        StrValidate12 sv=new StrValidate12();
        Scanner sc=new Scanner(System.in);
        System.out.println("ente any string: ending with _job and
having atleast 8 char to left of _job");
        String str=sc.next();
        sv.validateUserName(str);

    }
}
```

Program13:

Example 13: Create a method that can accept an array of String objects and sort in alphabetical order. The elements in the left half should be completely in uppercase and the elements in the right half should be completely in lower case. Return the resulting array.

Note: If there are odd number of String objects, then (n/2)+1 elements should be in UPPPERCASE

Method Name	getArrayList
Method Description	Converts the String array to ArrayList and sorts it
Argument	String []elements
Return Type	String [] modifiedArray
Logic	Load the elements in to an ArrayList ,sort it, convert the left half element to uppercase and

	right half elements to lower case . Hint : 1. Use Collection 2. Use String API

```
package com.cognizant.geometry;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
public class StrSortUL13 {
	/**
	 * @param args
	 */
	static String[] strsort(String s[]){
	ArrayList<String> a=new ArrayList<String>();
	for(int i=0;i<s.length;i++){
		a.add(s[i]);
	}
	System.out.println(a);
	//Collection<String> strings=
	Collections.sort(a);
	System.out.println(a);
	String s1[]=new String[a.size()];
	System.out.println(a.size());
	if(a.size()%2==0){
		for(int i=0;i<a.size();i++){
			if(i<a.size()/2)
				s1[i]=a.get(i).toUpperCase();
			else
			s1[i]=a.get(i).toLowerCase();
		}
	}
	return s1;
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
```

```
        String s[]={"kanth","raj","mani","rk"};
        String d[]=strsort(s);
        for (String string : d) {
            System.out.println(string);
        }
    }}
```

Program14 and 15:

Example 14: Create a method which can remove a List from another List

Method Name	removeElements
Method Description	Removes the elements in one list that is present in the second list also.
Argument	List<String> list1, List<String> list2;
Return Type	List- ArrayList contains the resulting List after the removal process.
Logic	Accept two List objects list1 and list2 and remove the elements from list1 that are present in list2. This should be done in single step process without using loop. Hint: Use the List API which removes all the items in List1 which are contained in List2

Example 15: Create a method which can remove all the elements from a list other than the list of elements specified.

Method Name	removeElements
Method Description	Remove all the elements from a list other than the list of elements specified.
Argument	List<String> list1, List<String> list2;

Return Type	List- ArrayList contains the resulting List after the removal process.
Logic	Accept two List objects list1 and list2 and remove all the elements from list 1 other than the elements contained in list2.This should be done in single step process without using loop. **Hint**: Use the List API method which can retain the elements available in the second list only

```
package com.cognizant.geometry;
import java.util.ArrayList;
import java.util.List;
public class ListRmvListq14q15 {
	ArrayList<String>listMinus(ArrayList<String> l1,List<String> l2){
		@SuppressWarnings("unchecked")
		List<String> s= (List<String>) l1.clone();
		s.removeAll(l2);
		//System.out.println(l1);
		return (ArrayList<String>) s;
	}
	ArrayList<String>listIntersect(ArrayList<String> l1,List<String> l2){
		@SuppressWarnings("unchecked")
		List<String> s= (List<String>) l1.clone();
		s.retainAll(l2);
		//System.out.println(l1);
		return (ArrayList<String>) s;
	}
	public static void main(String[] args) {
		ArrayList<String> l1=new ArrayList<String>();
		l1.add("kanth");
		l1.add("rk");
		l1.add("sai");
		l1.add("raj");
```

```
            List<String> l2=new ArrayList<String>();
            l2.add("mani");
            l2.add("kanth");
            l2.add("raj");
            l2.add("srirag");
            ListRmvListq14q15 ls=new ListRmvListq14q15();
            ArrayList<String> lll=ls.listMinus(l1, l2);
            System.out.println(lll);
            System.out.println(l1 +"       "+l2);
            ArrayList<String> ll=ls.listIntersect(l1, l2);
            System.out.println(ll);


        }

}
```

Program16:Example 16: Create a method which accepts an array of numbers and returns the numbers and their squares in an HashMap

Method Name	getSquares
Method Description	Accepts a list of numbers and return their squares
Argument	int[]
Return Type	Map
Logic	Iterate through the list, find the square of each number and add the elements to a map object with the number as the key and the square as the value.

```
package com.cognizant.geometry;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
public class MapSqr16 {
	/**
```

```
	 * @param args
	 */
	static Map<Integer, Integer> numsqr(List<Integer> l){
		Map<Integer, Integer> m=new HashMap<Integer, Integer>();
		Iterator<Integer> it=l.iterator();
		while(it.hasNext()){
			int n=it.next();
		m.put(n,(int) Math.pow(n, 2));
		}
		return m;
	}
	static Map<Integer, Integer> numsqrArr(int[] a){

		Map<Integer, Integer> m=new HashMap<Integer, Integer>();
		for(int i=0;i<a.length;i++){
			m.put(a[i],(int) Math.pow(a[i], 2));
		}
		return m;
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		List<Integer> l=new ArrayList<Integer>();
		l.add(2);
		l.add(4);
		l.add(6);
		Map<Integer,Integer> m=MapSqr16.numsqr(l);
		System.out.println(m);
		int a[]={2,5,8};
		Map<Integer,Integer> m1=MapSqr16.numsqrArr(a);
		System.out.println(m1);
	}
}
```

Program17:Example 17: Create a method which accepts the id and the age of people as a Map and decide if they are eligible for vote. A person is eligible for vote if his age is greater than 18. Add the IDs of all the eligible persons to list and return the list. (Assume date is in DD/MM/yyyy format)

Method Name	votersList
Method Description	Generate the list of voters based on the ages of the people
Argument	Map
Return Type	List
Logic	Accept a map with ID as key and Date of Birth as value and check if the person is eligible to vote. A person is eligible for vote for if his age is greater than 18. If the person is eligible add his ID to the list. Hint: Use Calendar API and SimpleDateFormat

```
package lab;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
public class VoterList17 {
	static ArrayList<Integer> votersList(Map<Integer,Date> dob)
	{
		Set<Map.Entry<Integer,Date>> set=dob.entrySet();
		ArrayList<Integer> vl=new ArrayList<Integer>();
		Iterator it =set.iterator();
		while(it.hasNext())
		{
			Map.Entry<Integer,Date> me=(Entry) it.next();
			Date dt=me.getValue();
			System.out.println(dt);
			Date cd=new Date();
			long diff=Math.abs(dt.getTime()-
cd.getTime())/(1000*60*60);
```

```
                long age=diff/(24*30*12);
                if(age>=18)
                        vl.add(me.getKey());
                }
        return vl;
    }
    public static void main(String[] args) throws ParseException {
        Map<Integer,Date> dob=new HashMap<Integer,Date>();
        SimpleDateFormat sdf=new
SimpleDateFormat("dd/MM/yyyy");
        dob.put(111, sdf.parse("12/3/1986"));
        dob.put(307, sdf.parse("1/5/1987"));
        dob.put(306, sdf.parse("12/8/1999"));
        dob.put(444, sdf.parse("12/9/1987"));
        ArrayList vl=votersList(dob);
        System.out.println(dob);
        System.out.println(vl);
    }}
```

Program18:Example 18: Create a method which accepts an integer array, reverse the numbers in the array and returns the resulting array in sorted order

Method Name	getSorted
Method Description	Return the resulting array after reversing the numbers and sorting it
Argument	int []
Return Type	int
Logic	Accept and integer array, reverse the numbers in the array, sort it and return the resulting array. Hint : 1. Convert the numbers to String to reverse it 2. Use Collection APIs to sort it **Ex:** {12,23,96,45} **Step 1:** Reverse numbers {21,32,69,54} **Step2:** Sort it {21,32,54,69} **Hint**: Use String to reverse number

	To sort it, Convert array to ArrayList and use Collections.sort

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

public class ArraySorting18 {
	static ArrayList<Integer> getSorted(int a[] )
	{
		ArrayList<Integer> ar=new ArrayList<Integer>();

		for(int i=0;i<a.length;i++)
		{
			StringBuffer str=new StringBuffer(String.valueOf(a[i]));
			String str1=str.reverse().toString();
			ar.add(Integer.parseInt(str1));
		}
		System.out.print("After Reversing:  ");
		System.out.println(ar);
		Collections.sort(ar);
		return ar;
	}
	public static void main(String[] args) {
		int arr[]={12,23,96,45};
		System.out.print("Original Array  :");
		for(int i=0;i<arr.length;i++)
		{
			System.out.print(arr[i]+",");
		}
		System.out.println();
		ArrayList<Integer> ar=getSorted(arr);
		System.out.print("After Sorting   :");
```

```
            System.out.println(ar);
        }

}
```

Program19:Example 19: Create a method which accepts an integer array and removes all the duplicates in the array. Return the resulting array in descending order

Method Name	modifyArray
Method Description	Remove duplicates
Argument	int []
Return Type	int []
Logic	Remove the duplicate elements in the array and sort it in descending order Hint: 1. Use Collection API (TreeSet) to remove duplicates and sort the result in ascending order 2. Create a new array, iterate through elements in TreeSet and add it in the reverse order

```
package lab;

import java.util.Collections;
import java.util.Iterator;
import java.util.TreeSet;

public class ModifyArray19 {
        static int [] modifyArray(int a[])
        {
                TreeSet<Integer> ts=new TreeSet<Integer>();
                for(int i : a)
                {
                        ts.add(i);
```

```
            }
            int arr[]=new int[ts.size()];
            Iterator<Integer> it=ts.iterator();
            int i=ts.size()-1;
            while(it.hasNext())
            {
                  arr[i--]=it.next();
            }
            return arr;
      }
      /**
       * @param args
       */
      public static void main(String[] args) {
            // TODO Auto-generated method stub

            int arr[]={42,23,34,23,32,21,32};
            for(int x : arr)
                  System.out.print(x+",");
            System.out.println();
            arr=modifyArray(arr);
            for(int x : arr)
            System.out.print(x+",");
      }
}
```

Program:20Example 20: Create a method that accepts a character array and count the number of times each character is present in the array. Add how many times each character is present to a hash map with the character as key and the repetitions count as value

Method Name	countCharacter
Method Description	Count the number of occurrence of each character in a Character array
Argument	char[]
Return Type	map

Logic	Count the number of times each character appears in the array. Add the details into a hash map with character as key and count as value. Example: {‘A’,’P’,’P’,’L’,’E’} Output: Will be hashmap with the following contents{‘A’:1,’P’:2,’L’:1,’E’:1}

```
package lab;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;
import java.util.Map.Entry;
public class CountCharacter20 {
	static HashMap countCharacter(char []ch)
	{
		HashMap<Character,Integer> hm=new
HashMap<Character,Integer>();
		for(int i=0;i<ch.length;i++)
		{
			int count=1;
			for(int j=i+1;j<=ch.length-1;j++)
				if(ch[i]==ch[j])count++;
			if(!hm.containsKey(ch[i]))
				hm.put(ch[i], count);
		}return hm;
	}
public static void main(String[] args) {
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter String ");
	String str=sc.next();
	char ch[]=str.toCharArray();
	HashMap<Character,Integer>map;
	map=countCharacter(ch);
```

```
        Set<Map.Entry<Character, Integer>> set=map.entrySet();
        terator it =set.iterator();
        System.out.print("[ ");
        while(it.hasNext())
        {
                Map.Entry<Character,Integer> me=(Entry) it.next();
                System.out.print(" '"+me.getKey()+"'"+":"+"'"+me.getValue()+"'
");
        }
        System.out.print(" ]");}}
```

Program21:Example 21: A String contains a list of states and capitals. Write a method which can parse the string and return the states and capitals as map with state as key and capital as value.
The String is in the below format.
The state and capital is separated by a delimiter (del1). There will be multiple state-capital pairs and each state – capital pair is separated by another delimiter (del2).
Ex: Input will be **tamilnadu||chennai-karanataka||bengaluru.**
Here, || will be provided as del1 and - will be provided as del2.

Method Name	getStates
Method Description	Accepts the states and capitals as a String and return a map
Argument	String data, char del1,char del l2
Return Type	Map
Logic	Parse the string based on the delimiters and load it to a map with the state name as key and capital as value. Hint: Use Stringtokenizer or split method in String class. Try both the above ways to get familiarized with both APIs

```
package lab;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
```

```
import java.util.Scanner;
import java.util.Set;
import java.util.Map.Entry;
public class GetStatess21 {
	static HashMap getStates(String data, String del1,String del2)
	{
		HashMap<String,String> map=new HashMap<String,String>();
		String stcap[]=data.split(del2);
		for(int i=0;i<stcap.length;i++)
		{
			//System.out.println(stcap[i]);		String
s[]=stcap[i].split("@");
			//System.out.println(s[0]+" "+s[1]);	map.put(s[0], s[1]);
		}
		return map;
	}
public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter state & capital String  state capital
pairs seperated by - and state and capital seperated by @");
		String str=sc.next();
		char ch[]=str.toCharArray();
		HashMap<String,String>map;
		map=getStates(str,"@","-");
		Set<Map.Entry<String,String>> set=map.entrySet();
		Iterator it =set.iterator();
		System.out.print("[ ");
		while(it.hasNext())
		{
			Map.Entry<Character,Integer> me=(Entry) it.next();
			System.out.print("
'"+me.getKey()+"'"+":"+"'"+me.getValue()+"' ");
		}
		System.out.print(" ]");	}		}
```

Program22:Example 22: In a certain television game show, a couple is considered as a perfect couple if both the husband's and wife's name

contains the same set of characters. Each couple is provided with an ID. Write a method which can accept a Hashmap with ID as key and the husband's and wife's name separated with "-" as value. The method should generate the list of perfect couples based on the above mentioned criteria and return their IDs as List object.

Method Name	checkPerfectCouple
Method Description	Select the set of perfect couples
Argument	Map
Return Type	List
Logic	Accept the Map Iterate through it Separate the husband's and wife's names If they contain the same characters, add the ID to the List object. Ex: Assuming VIMAL-MALIV is the value, this is a perfect couple since both these names contains same characters (in different order).

```
package lab;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.Map.Entry;
public class CheckPerfectCouple22 {
	static ArrayList toCharArr(String str)
	{
		ArrayList<Character> al=new ArrayList<Character>();
		char ch[]=str.toCharArray();
		for(int i=0;i<ch.length;i++)
			al.add(ch[i]);
		return al;
	}

	static ArrayList checkPerfectCouple(Map<Integer,String> map)
```

```
	{
		Set<Map.Entry<Integer,String>> set=map.entrySet();
		Iterator it =set.iterator();
		ArrayList<Character> al1,al2;
		ArrayList<Integer> lst1=new ArrayList<Integer>();
		while(it.hasNext())
		{
			Map.Entry<Integer,String> me=(Entry) it.next();
			String st=me.getValue();
			String str[]=st.split("-");
			al1=toCharArr(str[0]);
			al2=toCharArr(str[1]);
			if(al1.size()==al2.size())
			{
				if(al1.containsAll(al2))
				{
					lst1.add(me.getKey());
				}
			}

		}
		return lst1;

	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		HashMap<Integer,String> hm=new HashMap<Integer,String>();
		hm.put(111, "vimal-malvi");
		hm.put(121, "simal-lamisraj");
		hm.put(131, "smal-lams");
		ArrayList<Integer> ar=checkPerfectCouple(hm);
System.out.println(ar);
	}
```

```
}
```

Program23:Example 23: Create a method which can perform a particular String operation based on the user's choice. The method should accept

the String object and the user's choice and return the output of the operation.
Options are
A: Add the String to itself
B: Replace alternate positions with *
C: Remove duplicate characters in the String
D: Change alternate characters to upper case

Method Name	changeString
Method Description	Modify the string based on user choice
Argument	String string, char ch
Return Type	String
Logic	Perform the required operation based on the user choice and return the resulting string

```
package corejava;
import java.util.Scanner;
public class StringMod23{
	String changeString(String  str,char ch)
	{
		if(ch=='A' || ch=='a')
		{
			str+=str;
		}
		else if(ch=='B' || ch=='b')
		{
			char st[]=str.toCharArray();
			for(int i=0;i<st.length;i=i+2)
				st[i]='*';
			str=new String(st);
		}
		else if(ch=='C' || ch=='c')
		{
			char st1[]=str.toCharArray();
			for(int j=0;j<st1.length;j++)
			{
```

```
                        for(int k=j-1;k>=0;k--)
                        {
                                if(st1[j]==st1[k]) st1[j]='\0';
                        }
                }
                str=new String(st1);
        }
        else
        {
                char st[]=str.toCharArray();
                for(int i=0;i<st.length;i=i+2)
                        st[i]=(char) (st[i]-32);
                str=new String(st);
        }
        return str;
    }
public static void main(String args[])
    {
        StringMod23 sm=new StringMod23();
        Scanner sc=new Scanner(System.in);
        String strr,chstr;
        System.out.println("Enter String");
        strr=sc.next();
        char ch;
        System.out.println("Enter A: Add the String to itself \nB:
Replace alternate positions with * \nC: Remove duplicate characters in the
String \nD: Change alternate characters to upper case");
        ch=sc.next().toCharArray()[0];
        System.out.println("Changed string is
"+sm.changeString(strr,ch));

        }


    }
```

Program24:Example 24: Create a method that accepts a String and checks if it is a positive string. A string is considered a positive string, if on moving

from left to right **each** character in the String comes after the previous characters in the Alphabetical order.
For Example
ANT is a positive String (Since T comes after N and N comes after A)
APPLE is not positive since L comes before P in the alphabetical order.
The method should return true if the entered string is positive

Method Name	checkPositive
Method Description	Checks if a String is positive
Argument	String
Return Type	boolean
Logic	Check if a string is positive based on the above criteria and return true if positive. Hint: **Step 1:** Convert to Char array. **Step 2:** Iterate through array, subtract 1st two characters (A-N). This will give the ASCII difference **Step 3:** If result is negative, then return false and break. Else continue to next loop

```
package com.cognizant.geometry;
import java.util.Scanner;
public class PositiveStr24 {
	char a1,b;
	boolean checkPositive(String a){
		System.out.println("Given input is :"+a);
		int f,s;
		for(int i=0;i<a.length()-1;i++){
			a1=a.charAt(i);
			b=a.charAt(i+1);
			f=Character.getNumericValue(a1);
			s=Character.getNumericValue(b);
			if(f-s>0)
				return false;
			else
				continue ;
		}
```

```
            return true;
            }
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            PositiveStr24 n=new PositiveStr24();
            Scanner sc=new Scanner(System.in);
            System.out.println("ente the string to check for positive:");

            String s=sc.next();
            boolean a=n.checkPositive(s);
            if(a){
                System.out.println("positive");
            }
            else{
                System.out.println("Negative");
            }
}}
```

Program:25: Example 25: Create a method which accepts two Arraylist containing characters. Merge both arrays lists, sort the elements in the resulting list and return the resulting array.

Method Name	mergeData
Method Description	Merge two arraylist , sort it and return the result as an integer array.
Argument	List, List
Return Type	char[]
Logic	Merge both arrays lists, sort the elements in the resulting list and return it as a char array.

```
package CoreJava_practical_lab;

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

```
public class ArrayList25 {
	char[] mergeData(List l1,List l2){
		l1.addAll(l2);
		Collections.sort(l1);
		StringBuffer sb=new StringBuffer();
		for(int i=0;i<l1.size();i++){
			sb.append(l1.get(i)+" ");
		}
		String s=sb.toString();
		char c[]=s.toCharArray();

		return c;

	}

	public static void main(String[] args) {
		List<Character> a1=new ArrayList<Character>();

	a1.add('k');a1.add('b');a1.add('l');a1.add('m');a1.add('j');a1.add('p');
		ArrayList<Character> a2=new ArrayList<Character>();

	a2.add('p');a2.add('j');a2.add('m');a2.add('l');a2.add('b');a2.add('k');

		ArrayList25 a25=new ArrayList25();
	char[] ch=a25.mergeData(a1,a2);
		System.out.println(ch);

	}
}
```

Program26: Example 26: Create a method that searches for a particular String in a List. If found, the element should be replaced with a string having only half of the characters in the actual string

Method Name	modifyElement
Method Description	Search for an element in the arraylist and modifies it.
Argument	List<String> arrayList , String element
Return Type	List
Logic	Accept an arraylist and search for an element in the list and replace with a string having only first half of the characters in the actual string. For Example if a search was done for APPLE and if APPLE is found in the list, replace it with APP. Return the modified list Hint: Iterate through list and find the index where the String is present. Take the first half of the String and set it at that index in the arraylist. (Use set method)

```
package CoreJava_practical_lab;

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

public class List26 {
	List<String>modifyElement(List<String> l,String s){
		String s1="";
		for(int i=0;i<l.size();i++){
```

```
                    if(l.get(i).equals("kotesh")){
                        StringBuffer sb=new StringBuffer(l.get(i));

                        int j=sb.length()/2;
                        sb=sb.replace(0,j,s);
                        s1=sb.toString();
                        l.set(i, s1);
                        }
            }
            return l;
 }

        public static void main(String[] args) {
            List<String> l1=new ArrayList<String>();

        l1.add("koti");l1.add("kotesh");l1.add("bk");l1.add("kb");l1.add("ks");
l1.add("km");l1.add("kotesh");
            List26 l26=new List26();
            System.out.println(l1);
            List<String> lr=l26.modifyElement(l1,"sm");
            System.out.println(lr);

        }
}
```

Program27: Example 27: Create a method to find the sum of the first n even numbers that are divisible by 3 and 8

Method Name	findSum
Method Description	Find the sum of first n even numbers that are divisible by 3 and 8
Argument	Int
Return Type	Int

Logic	Sum of the multiples of first n even numbers that are divisible by 3 and 8

```
package CoreJava_practical_lab;
import java.util.Scanner;

public class Sumd38_27{

	static int findSum(int n){
		int sum=0;
		for(int i=1;i<=n;i++){
			if(i%2==0 && i%3==0 && i%8==0){
				sum+=i;
			}
		}
		return sum;
	}

	public static void main(String[] args) {

		int n;
		Scanner sc=new Scanner(System.in);

		System.out.println("enter a number");
		n=sc.nextInt();
		System.out.println(findSum(n));

	}

}
```

Program28: Example 28: Create a method to find the sum of the cubes of the digits of an n digit number

Method Name	findSum
Method Description	Find the sum cubes of the digits of an n digit number
Argument	Int
Return Type	Int
Logic	Return the sum of cubes of the digits of an n digit number Example Input : 123 Output : 1^3+2^3+3^3= 1+8+27=36 Hint: Use %(mod) operator to separate each digit

```
package CoreJava_practical_lab;

import java.util.Scanner;

public class SumCube28 {
	static int findSum(int n){
		int sum=0;
		while(n>0){
		int r=n%10;
		sum=sum+(r*r*r);
		n=n/10;
		}
```

```
            return sum;
        }

        public static void main(String[] args) {

            int n;
            Scanner sc=new Scanner(System.in);

            System.out.println("enter a number");
            n=sc.nextInt();
            System.out.println(findSum(n));
        }

}
```

Program29: Example 29: Create a method which accepts a hash map and return the values of the map in sorted order as a List.

Method Name	getValues
Method Description	Get the values of a map in sorted order
Argument	HashMap
Return Type	List
Logic	Return the values of a hash map in sorted order

```
package CoreJava_practical_lab;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class HashMap29 {

	List getValues(HashMap hs){

		List<String> list = new ArrayList<String>(hs.values());
		return list;

	}


	public static void main(String[] args) {
		HashMap<Integer, String> map = new HashMap<Integer,
String>();
		map.put (2, "kotesh");
		map.put (1, "koti");
		map.put (7, "bk");
		map.put (5, "kb");
		System.out.println(map);
		HashMap29 hs29=new HashMap29();
		List ls=hs29.getValues(map);
		System.out.println(ls);


	}

}
```

Program31:Example 31: Write a method to find the sum of the factorials of the first n numbers in the Fibonacci series.
Fibonacci series is given by 0 1 1 2 3 5 8……
Factorial for a number m is given by factorial= m*m-1*m-2….1

Method Name	sumOfFactorial
Method Description	Calculate sum
Argument	int n
Return Type	int
Logic	1. Generate the first n elements in the Fibonacci series 2. Find the factorial of each element 3. Find the sum of the factorial

```
package CoreJava_practical_lab;

import java.util.Scanner;

public class FibFactSum31 {
	static int sumOfFactorial(int n){

		int f1=0,f2=1,f3,sum=0;

		for (int i=0;i<n;i++)
		  {
		if (i<=1)
			f3=i;
		else
		    {
			f3=f1+f2;
```

```
                f1=f2;
                f2=f3;
              }//else
            sum=sum+fact(f3);
             }//for

            return sum;
  }//sumOfFactorial()

            static int fact(int x){
                if(x==0 || x==1)
                        return 1;
                else return x*fact(x-1);
            }//fact()

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

            System.out.println("enter a number");
            n=sc.nextInt();
            System.out.println(sumOfFactorial(n));

        }//main()
}//FibFactSum
```

Program32:Example 32: A company transmits its String data over the network as encrypted data. The encryption logic is as shown given below. For a String ad the logic is as given

a→ a+9=j

d→d+9=m

So the encrypted word would be jm.

If on addition of 9 results in a char greater than z (ASCII value 122) do the encryption in a cyclic manner starting from a. So if any	encryptString

character is ‘z’ it should be (z+9) which is equal to 127>122. In this case the character would be 9 character starting from ‘a’ which ‘i’ so for adz the encrypted value should be adi Method Name	
Method Description	Encrypt the entered string
Argument	String
Return Type	String
Logic	1. Perform the arithmetic operation of char data. 2. For example Assume char a=’b’; a++; Now the value of a will be c. This is because in java the arithmetic operation on character works on its ASCII value. The ASCII value of ‘a’ is 97 and that of ‘z’ is 122.

```
package CoreJava_practical_lab;
import java.util.Scanner;
public class Encrypt32 {
	String encryptString(String st)
	{
		char ch[]=st.toCharArray();
		for(int i=0;i<st.length();i++)
		{
			if(ch[i]>='a' && ch[i]<='q')
			ch[i]=(char) (ch[i]+9);
			else
				ch[i]=(char) (ch[i]-17);
		}
		String str=new String(ch);
		return str;
	}
```

```
public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		String str1,str2;
		System.out.println("Enter String");
		str1=sc.next();
		Encrypt32 ecr=new Encrypt32();
		str2=ecr.encryptString(str1);
		System.out.println("Encrypted string is  :"+str2);
	}
}
```

Program33: Example 33: A sales company keeps track of the product purchased and sold. The company needs to make sure that the sale date is always after the purchase date. Write a method to verify this

Method Name	compareDates
Method Description	Comparing the purchase date and selling date
Argument	String purchaseDate, String sellingDate
Return Type	boolean
Logic	1. Convert the string to Date objects 2. Return true if the selling date comes after the purchase date

```
package handson;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Ex33 {

	public static void main(String[] args) throws ParseException {
		Scanner sc=new Scanner(System.in);
```

```
            System.out.println("enter purchase date as dd-mm-yy");
            String pd=sc.next();
            System.out.println("enter selling date as dd-mm-yy");
            String sd=sc.next();
            SimpleDateFormat sdf=new SimpleDateFormat("dd-mm-
yyyy");
            Date d1=sdf.parse(pd);
            Date d2=sdf.parse(sd);
            int k=d1.compareTo(d2);
            if(k<0)
                  System.out.println("verified:true");
            else
                  System.out.println("verified:false");
      }
}
```

Program34: Example 34: A company used to keep the record of the employees in two different branches separately. There are some employees who work in both the location. The company needs to keep track of the employee working in both the branches. Write a method to accept the two lists containing the names of the employees working in the

two branches. The method should find out the names of employees present in both the list and return the names as a sorted array

Method Name	getEmployees
Method Description	Get the names of employees working two different branches
Argument	List branch1, List branch2
Return Type	String []
Logic	Find the common names of the employees in both the lists

```
package handson;

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

public class Ex34 {

	public static void main(String[] args) {
		List<String> branch1=new ArrayList<String>();
		List<String> branch2=new ArrayList<String>();
		Scanner sc=new Scanner(System.in);
		System.out.println("enter no of employees in branch1");
		int n1=sc.nextInt();
		System.out.println("enter no of employees in branch1");
		int n2=sc.nextInt();
		System.out.println("Enter names of employees in branch1");
		for(int i=0;i<n1;i++)
			branch1.add(sc.next());

		System.out.println("Enter names of employees in branch2");
		for(int i=0;i<n2;i++)
			branch2.add(sc.next());
		branch1.retainAll(branch2);
		System.out.println("Common names of employees in both the branches:"+branch1);
```

```
        }
    }
```

Program36:

Example 36: Write a method which can find the sum of the first n prime numbers. Prime numbers are numbers which have only 1 and the number itself as its factors. 2 is the only even prime number. 1 is neither prime nor composite.

Ex: the 1st 5 prime numbers are 2,3,5,7,11 and sum is 28

Method Name	getPrimeSum
Method Description	Get the sum of the first n prime numbers
Argument	int n
Return Type	Int
Logic	**Hint :** 1. Use for loop to iterate over numbers from 2 to n, say loop variable i. 2. Use an inner loop with loop variable j which loops from to 2 to i/2. If for any j the remainder on dividing i by j is zero, the number is non-prime. If it is prime add the number to the sum.

```
package exp36;
import java.util.Scanner;
public class ex36 {
public static void main(String[] args) {
		// TODO Auto-generated method
stub
		int sum=0,count=0,num=2;
		Scanner sc=new Scanner
(System.in);
		System.out.println("enter the n value to find the sum of first n
prime numbers");
		int n=sc.nextInt();
		while(count<n) {
		int k=0;
		for(int i=2;i<num;i++)
		{
		if(num%i==0)
		{
		k=1;
		break;
		}
		}
		if(k==0)
		{
		System.out.print(num+" ");
		count++;
		sum+=num;
		}
		num++;
		}
		System.out.println("\n

sum="+sum);
		}

}
```

Program37: Example 37: Write a method which accepts a String and moves all the lower case ‘a’ to the beginning of the String.

Method Name	rearrangeCharacters
Method Description	Move the all the lower case ‘a’ to the beginning of a String
Argument	String
Return Type	String
Logic	Hint : 1. Convert the string to a character array 2. Create a Stringbuffer object 3. Create a variable(**count**) to store the number of ‘a’ present 4. Iterate over the character array and if the character is ‘a’ increment **count** for ‘a’ else add the character to the StringBuffer object. 5. Finally insert the **count** number of ‘a’ to the beginning of the StringBuffer object

```
package com.cognizant.geometry;
import java.util.Scanner;
public class StrOrdr37 {

	/**
	 * @param args
	 */
	static void straFirst(String s){
		char[] c=s.toCharArray();
		for(int i=1;i<c.length;i++){
			if(c[i]=='a'){
				for(int j=i;j >0;j--){
					if(c[j-1]!='a'){
						char d=c[j];
```

```
                        c[j]=c[j-1];
                        c[j-1]=d;
                    }else{
                        break;
                    }
                }
            }else{
                continue;
            }
        }
        String str=new String(c);
        System.out.println(str);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string");
        String s=sc.next();
        StrOrdr37.straFirst(s);
    }
}
```

Program38:Example 38: Write a method which can find the factors of a number. Factor is an integer which evenly divides a number without leaving a remainder. Return the factors as an arraylist object. For Example 1, 2 and 4 are the factor of 4

Method Name	getFators
Method Description	Get the factors of a number n
Argument	int n
Return Type	List
Logic	Hint : 1. Create a loop starting from 1 to n with loop variable say i. 2. Check if for any i , dividing n by i gives zero as remainder. Then i is a factor of n . 3. Add i to the list object

```
package examples;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class Factors38 {
	public static void main(String[] args) {
		int i,j=0;
		List<Integer> factors=new ArrayList<Integer>();
		System.out.println("enter the number");
		Scanner se=new Scanner(System.in);
		int n=se.nextInt();
		for(i=1;i<=n;i++){
			if(n%i==0)
				{ factors.add(i);
			j++;}
		}	System.out.print(factors);
	}	}
```

Program39: Example 39: Write a method which can accept an integer and return the binary, hexadecimal and octal equivalents of the number in a String array

Method Name	getFormats
Method Description	Gets the binary,hexadecimal and octal formats of a number
Argument	int
Return Type	String
Logic	Hint : 1. Use Integer wrapper class methods

```
package mypack;
import java.util.Scanner;
public class ex39 {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("enter the decimal number");
		int n=sc.nextInt();
		Integer i=new Integer(n);
		String b=i.toBinaryString(i);
```

```
        String o=i.toOctalString(i);
        String h=i.toHexString(i);
        System.out.println(b);
        System.out.println(o);
        System.out.println(h);}       }
```

Program40:

Example 40: Write a method which accepts a double number and finds the sum of the digits to the left and right of the decimal point. It should return the sum as String in the following format

Left side sum:Right side sum

For example

Input :120.520

Output: 3:7

Method Name	getSum
Method Description	Get the sum of digits on either sides of the decimal points in a double number
Argument	double
Return Type	String
Logic	Hint : 1. Convert the double number to aString 2. Separate the String to two parts based on the decimal point. 3. Find the sum of digits on each part

```
package com.cognizant.geometry;

import java.util.Scanner;

public class DoubleAddStr40 {

    /**
     * @param args
     */
    String getSum(double d){
        String s=Double.toString(d);
        int res1=0,res2=0;
```

```
            int i=s.indexOf(".");
            //System.out.println(i);
            for(int j=0;j<i;j++){
                  res1+=Character.getNumericValue(s.charAt(j));
            }
            for(int j=i+1;j<s.length();j++){
                  res2+=Character.getNumericValue(s.charAt(j));
            }
            String fnl=Integer.toString(res1)+":"+Integer.toString(res2);
            return fnl;
      }
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            Scanner sc=new Scanner(System.in);
            System.out.println("enter double number");

            DoubleAddStr40 das=new DoubleAddStr40();
            String b=das.getSum(sc.nextDouble());
            System.out.println(b);

      }

}
```

Program44:Example 44: Write a method which accepts a number and return it in words.

For Example 123 → One Two Three

Method Name	getNumber
Method Description	Get the number in words
Argument	int
Return Type	String
Logic	Use mod(%) operator, StringBuffer and switch case

```
package corejava;
import java.util.Scanner;
public class ConvertNumberToStr44{
```

```
	String getNumber(int num)
	{
		String str="";
		int d;
		while(num!=0)
		{
			d=num%10;
			switch(d)
			{
			case 0:str="Zero  "+str;		break;
			case 1:str="One  "+str;		break;
			case 2:str="Two  "+str;		break;
			case 3:str="Three  "+str;		break;
			case 4:str="Four  "+str;		break;
			case 5:str="Five  "+str;		break;
			case 6:str="six  "+str;		break;
			case 7:str="Seven  "+str;		break;
			case 8:str="Eight  "+str;		break;
			case 9:str="Nine  "+str;		break;
			default:break;
			}
			num=num/10;
		}
		return str;
	}
	public static void main(String args[])
	{
		ConvertNumberToStr44 cnt= new ConvertNumberToStr44();
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter number  :");
		int num=sc.nextInt();
		String str=cnt.getNumber(num);
		System.out.println(num+"in String format is  "+str);
	}
}
```

Program46: Example 46: Write a method which can check whether an entered number is palindrome or not.

Method Name	checkPalindrome
Method Description	Check palindrome
Argument	int
Return Type	boolean
Logic	Hint 1: 1. Convert the number to String 2. Check if the String and reverse of the String are equal

```
package exp40;
import java.util.Scanner;
public class ex46 {

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner sc=new Scanner(System.in);
		System.out.println("enter the string");
		String a=sc.next();
		int k=a.length();
		String t="";
		for(int i=k-1;i>=0;i--)
		{
			t=t+a.charAt(i);
		}
		if(a.equals(t))
```

```
                System.out.println("palindrome");
            else
                System.out.println("not a palindrome");


        }

}
```