21. Date Format Conversion

Given a date string in the format dd/mm/yyyy, write a program to convert the given date to the format dd-mm-yy.

Include a class **UserMainCode** with a static method "**convertDateFormat**" that accepts a String and returns a String.

Create a class **Main** which would get a String as input and call the static method **convertDateFormat** present in the UserMainCode.

Input and Output Format:
Input consists of a String.
Output consists of a String.

Sample Input:
12/11/1998

Sample Output:
12-11-98

import java.text.ParseException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

```
public class Main {
public static void main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
String n=sc.next();
      System.out.println(User.convertDateFormat(n));

}
}
```

```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
importjava.util.ArrayList;
import java.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstatic String convertDateFormat(String n) throws ParseException{
      SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
      Date d= sdf.parse(n);
      SimpleDateFormat sdf1=new SimpleDateFormat("dd-MM-yyyy");
      String s=sdf1.format(d);
      returns;
      }
}
```

22. Valid Date

Given a date string as input, write a program to validate if the given date is in any of the following formats:
dd.mm.yyyy
dd/mm/yy
dd-mm-yyyy

Include a class **UserMainCode** with a static method "**validateDate**" that accepts a String and returns an integer. This method returns 1 if the date is valid, else return -1.

Create a class **Main** which would get a String as input and call the static method **validateDate** present in the UserMainCode.

Input and Output Format:
Input consists of a String.
Output consists of a String that is either 'Valid' or 'Invalid'.

Sample Input 1:
12.03.2012

Sample Output 1:
Valid

Sample Input 2:
27#01#1977

Sample Output 2:
Invalid

```
	publicstaticvoid main(String[] args) {

Scanner sc=new Scanner(System.in);
String s=sc.nextLine();
SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
sdf.setLenient(false);
int res=0;
if(s.matches("[0-9]{2}(/)[0-9]{2}(/)[0-9]{4}"))
{
	try {
		Date d=sdf.parse(s);
		res=1;
	} catch (ParseException e) {
		res=-1;
	}

System.out.println(res);

	}
}
}
```

23. Convert Format

Given a 10 digit positive number in the format XXX-XXX-XXXX as a string input, write a program to convert this number to the format XX-XX-XXX-XXX.

Include a class **UserMainCode** with a static method "**convertFormat**" that accepts a String argument and returns a String.

Create a class **Main** which would get a String as input and call the static method **convertFormat** present in the UserMainCode.

Input and Output Format:
Input consists of a String.
Output consists of a String.

Sample Input:

555-666-1234

Sample Output:

55-56-661-234

```
import java.text.ParseException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Scanner;

public class Main {
public static void main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
String n=sc.next();
System.out.println(User.convertFormat(n));

}
}
```

```
import java.text.ParseException;
importjava.text.SimpleDateFormat;
importjava.util.ArrayList;
importjava.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
import java.util.StringTokenizer;


publicclass User {
publicstatic String convertFormat(String s) throws ParseException{
	StringTokenizer st=new StringTokenizer(s,"-");
	int i=0;
	String[] s1=new String[st.countTokens()];
	while(st.hasMoreTokens())
	{
		s1[i]=st.nextToken();
		i++;
	}
	StringBuffer sb=new StringBuffer();
	sb.append(s1[0].substring(0,2));
		sb.append("-");
		sb.append(s1[0].substring(2)).append(s1[1].substring(0,1));
		sb.append("-");
		sb.append(s1[1].substring(1)).append(s1[2].substring(0,1));
		sb.append("-");
		sb.append(s1[2].substring(1));

		return sb.toString();
	}
```

```
}
```

```
import java.util.StringTokenizer;
publicclass UserMainCode {
        publicstatic String convertFormat(String s)
        {
                StringBuffer sb=new StringBuffer();
                StringTokenizer st=new StringTokenizer(s,"-");
                String s1=st.nextToken();
                String s2=st.nextToken();
                String s3=st.nextToken();
           sb.append(s1.substring(0,2));
           sb.append("-");
           sb.append(s1.substring(s1.length()-1));
           sb.append(s2.substring(0,1));
           sb.append("-");
           sb.append(s2.substring(1));
           sb.append(s3.substring(0,1));
           sb.append("-");
           sb.append(s3.substring(1));
        return sb.toString();

        }
}
```

24. Add and Reverse

Given an int array and a number as input, write a program to add all the elements in the array greater than the given number. Finally reverse the digits of the obtained sum and print it.

Include a class **UserMainCode** with a static method "**addAndReverse**" that accepts 2 arguments and returns an integer.The first argument corresponds to the integer array and the second argument corresponds to the number.

Create a class **Main** which would get the required input and call the static method **addAndReverse** present in the UserMainCode.

Example:
Input Array = {10,15,20,25,30,100}
Number = 15
sum = 20 + 25 + 30 + 100 = 175
output = 571

Input and Output Format:
The first line of the input consists of an integer that corresponds to the number of elements in the array.
The next n lines of the input consists of integers that correspond to the elements in the array.
The last line of the input consists of an integer that corresponds to the number.

Output consists of a single integer.

Sample Input
6
10
15
20
25
30
100
15

Sample Output
571

```
import java.text.ParseException;
```

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

public class Main {
public static void main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
int n=sc.nextInt();
int[] a=new int[n];
for(int i=0;i<n;i++)
      a[i]=sc.nextInt();
int x=sc.nextInt();
System.out.println(User.addAndReverse(a,x));

}
}
```

```
importjava.text.ParseException;
importjava.text.SimpleDateFormat;
importjava.util.ArrayList;
importjava.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticint addAndReverse(int[] a,int x) {
      int sum=0;
      int rev=0,r=0;
      for(int i=0;i<a.length;i++)
      {
            if(x<a[i])
                  sum=sum+a[i];
      }

      while(sum!=0)
      {
            r=sum%10;
            rev=rev*10+r;
            sum=sum/10;
      }

            return rev;
      }
}
```

25. Next Year day

Given a date string in dd/mm/yyyy format, write a program to calculate the day which falls on the same date next year. Print the output in small case.

The days are sunday, monday, tuesday, wednesday, thursday, friday and saturday.

Include a class **UserMainCode** with a static method "**nextYearDay**" that accepts a String and returns a String.

Create a class **Main** which would get a String as input and call the static method **nextYearDay** present in the UserMainCode.

Input and Output Format:
Input consists of a String.
Output consists of a String.

Sample Input:
13/07/2012

Sample Output:
saturday

```
import java.text.ParseException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;



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

System.out.println(User.nextYearDay(n));

}
}
```

```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
importjava.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstatic String nextYearDay(String s) throws ParseException {
      SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
      SimpleDateFormat sdf1=new SimpleDateFormat("EEEE");
      Date d= sdf.parse(s);
      Calendar c=Calendar.getInstance();
      c.setTime(d);
    c.add(Calendar.YEAR,1);
    Date year=c.getTime();
    String day=sdf1.format(year);

return day;
}
}
```

26. Sum Squares of Digits

Write a program that accepts a positive number as input and calculates the sum of squares of individual digits of the given number.

Include a class **UserMainCode** with a static method "**getSumOfSquaresOfDigits**" that accepts an integer argument and returns an integer.

Create a class **Main** which would get an integer as input and call the static method **getSumOfSquaresOfDigits** present in the UserMainCode.

Input and Output Format:
Input consists of an integer.
Output consists of an integer.

Sample Input:
321

Sample Output:
14

```
import java.text.ParseException;
```

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

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

System.out.println(User.getSumOfSquaresOfDigits(n));

}
}
```

```
importjava.text.ParseException;
importjava.text.SimpleDateFormat;
importjava.util.ArrayList;
importjava.util.Calendar;
importjava.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticint getSumOfSquaresOfDigits(int n) {
	int sum=0,r=0;
	while(n!=0)
	{
		r=n%10;
		sum=sum+(r*r);
		n=n/10;
	}
	return sum;
}
}
```

27. Even and Odd Index Sum

Write a program that accepts a positive number as input and calculates the sum of digits at even indexes (say evenSum) and sum of digits at odd indexes (say oddSum) in the given number. If both the sums are equal , print 'yes', else print no.

Example:
input = 23050
evenSum = 2 + 0 + 0 = 2
oddSum = 3 + 5 = 8
output = no

Include a class **UserMainCode** with a static method "**sumOfOddEvenPositioned**" that accepts an integer and returns an integer. The method returns 1 if the 2 sums are equal. Else the method returns -1.

Create a class **Main** which would get an integer as input and call the static method **sumOfOddEvenPositioned** present in the UserMainCode.

Input and Output Format:
Input consists of an integer.
Output consists of a string that is either "yes" or "no".

Sample Input 1:
23050

Sample Output 1:
no

Sample Input 2:
231

Sample Output 2:
yes

```
import java.text.ParseException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Scanner;

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

int res=User.sumOfOddEvenPositioned(n);
if(res==1)
      System.out.println("yes");
else
      System.out.println("no");

}
}
importjava.text.ParseException;
importjava.text.SimpleDateFormat;
importjava.util.ArrayList;
```

```java
importjava.util.Calendar;
importjava.util.Date;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticint sumOfOddEvenPositioned(int n) {
	int even=0,odd=0;
	int res=0,r=0,m=0;
	int j=0,count=0;
int n1=n;
	while(n>0)
	{
		n=n/10;
		count++;
	}
	System.out.println(count);
	int[] a=newint[count];
	while(n1!=0)
	{
		r=n1%10;
		a[j]=r;
		j++;
		n1=n1/10;
	}
	int[] b=newint[j];
	for(int k=j-1;k>=0;k--)
	{
		b[m]=a[k];
		m++;
	}

	for(int i=0;i<m;i++)
	{
		System.out.println("a:"+b[i]);
		if(i%2==0)
			even=even+b[i];
		else
			odd=odd+b[i];
	}
	System.out.println(even);
	System.out.println(odd);
	if(even==odd)
		res=1;
	else
		res=-1;
	return res;
}
}
```

28. Remove 3 Multiples

Write a program that accepts an ArrayList of integers as input and removes every 3rd element and prints the final ArrayList.

Suppose the given arrayList contains 10 elements remove the 3rd, 6th and 9th elements.

Include a class **UserMainCode** with a static method "**removeMultiplesOfThree**" that accepts an ArrayList<Integer> as arguement and returns an ArrayList<Integer>.

Create a class **Main** which would get the required input and call the static method **removeMultiplesOfThree** present in the UserMainCode.

Input and Output Format:
The first line of the input consists of an integer n, that corresponds to the number of elements to be added in the ArrayList.
The next n lines consist of integers that correspond to the elements in the ArrayList.

Output consists of an ArrayList of integers.

Sample Input:
6
3
1
11
19
17
19

Sample Output
3
1
19
17

```
publicclass Main {
publicstaticvoid main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
int n=sc.nextInt();
ArrayList<Integer> a=new ArrayList<Integer>();
ArrayList<Integer> res=new ArrayList<Integer>();
for(int i=0;i<n;i++)
```

```
        a.add(sc.nextInt());
res=User.removeMultiplesOfThree(a);
for(int i=0;i<res.size();i++)
        System.out.println(res.get(i));
}
}
```

```
 publicclass User {
publicstatic ArrayList<Integer> removeMultiplesOfThree(ArrayList<Integer> a)
{
        ArrayList<Integer> b=new ArrayList<Integer>();
        for(int i=0;i<a.size();i++)
        {
                int d=a.get(i);
                if(d%3!=0)
                {
                        b.add(a.get(i));
                }
        }
        return b;
}
}
```

29.String Occurances - II

Obtain two strings S1,S2 from user as input. Your program should count the number of times S2 appears in S1.

Return the count as output. Note - Consider case.

Include a class UserMainCode with a static method **getSubstring** which accepts two string variables. The return type is the count.

Create a Class Main which would be used to accept two Input strings and call the static method present in UserMainCode.

Input and Output Format:

Input consists of two strings with maximum size of 100 characters.

Output consists of an integer.

Refer sample output for formatting specifications.

Sample Input 1:
catcowcat
cat

Sample Output 1:
2

Sample Input 2:
catcowcat
CAT

Sample Output 2:
0

```
import java.text.ParseException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Scanner;


public class Main {

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

Scanner sc = new Scanner(System.in);

String s1=sc.next();

String s2=sc.next();

System.out.println(User.getSubstring(s1, s2));

}

}
```

```
import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.HashMap;

import java.util.Iterator;

import java.util.StringTokenizer;



public class User {

public static int getSubstring (String s1,String s2) {

        int count=0;

        int n=s1.length()-(s2.length()-1);

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

        {

                String s3=s1.substring(i,i+(s2.length()));

                if(s2.equals(s3))

                count++;

        }

        return count;

}

}
```

```
publicclass User {
publicstaticint getSubstring (String s1,String s2) {
	int count=0;
	int n=s1.length()-(s2.length()-1);
	ints2l=s2.length();
	System.out.println(n);
	for(int i=0;i<n;i++)
	{
		String s3=s1.substring(i,i+s2l);
		if(s2.equals(s3))
		count++;
	}
	return count;
}
}
```

30. Programming Logic

Write a Program that accepts three integer values (a,b,c) and returns their sum. However, if one of the values is 13 then it does not count towards the sum and the next number also does not count. So for example, if b is 13, then both b and c do not count.

Include a class UserMainCode with a static method **getLuckySum** which accepts three integers. The return type is integer representing the sum.

Create a Class Main which would be used to accept the input integers and call the static method present in UserMainCode.

Input and Output Format:

Input consists of three integers.

Output consists of a single integer.

Refer sample output for formatting specifications.

Sample Input 1:

1
2
3

Sample Output 1:
6

Sample Input 2:
1
2
13

Sample Output 2:
3

Sample Input 3:
13
3
8

Sample Output 3:
8

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Iterator;
import java.util.StringTokenizer;


public class User {
public static int getLuckySum  (int a,int b, int c) {
```

```
        int res=0;
        if(a==13)
                res=c;
        else if(b==13)
                res=a;
        else if(c==13)
                res=a+b;
        else
                res=a+b+c;

                return res;

}
}
```
