------------------------------ SET 3 --

1) Find the sum of the numbers in the given input string array
   Input{“2AA”,”12”,”ABC”,”c1a”)
   Output:6 (2+1+2+1)
   Note in the above array 12 must not considered as such
   i.e,it must be considered as 1,2

```
package Set3;
public class ClassSeT01 {
public static void main(String[] args) {
        String[] s1={"2AA","12","A2C","C5a"};
        getSum(s1);     }
public static void getSum(String[] s1) {
        int sum=0;
        for(int i=0;i<s1.length;i++)
                for(int j=0;j<s1[i].length();j++){
                        char c=s1[i].charAt(j);
                        if(Character.isDigit(c)){
                                String t=String.valueOf(c);
                                int n=Integer.parseInt(t);
                                sum=sum+n; } }
        System.out.println(sum);        }
}
```

--||

2) Create a program to get the hashmap from the given input string array where the key for the hashmap
   is first three letters of array element in uppercase and the value of hashmap is the element itself
   Input:{“goa”,”kerala”,”gujarat”}   [string array]
   Output:{{GOA,goa},{KER,kerala},{GUJ,Gujarat}}   [hashmap]

```
package Set3;
import java.util.*;
public class ClassSeT02 {
public static void main(String[] args) {
        String[] s1={"goa","kerala","gujarat"};
        putvalues(s1);
}
public static void putvalues(String[] s1) {
        ArrayList<String> l1=new ArrayList<String>();
        HashMap<String,String> m1=new HashMap<String,String>();
        ArrayList<String> l2=new ArrayList<String>();
        for(String s:s1)
                l1.add(s.toUpperCase().substring(0, 3));
        for(String s:s1)
                l2.add(s);
        for(int i=0;i<l1.size();i++)
                m1.put(l1.get(i),l2.get(i));
        System.out.println(m1);
}
}
```

--||

3) String[] input1=["Vikas","Lokesh",Ashok]
   expected output String: "Vikas,Lokesh,Ashok"

```
package Set3;
public class ClassSeT03 {
public static void main(String[] args) {
    String[] ip={"Vikas","Lokesh","Ashok"};
    System.out.println(getTheNamesinGivenFormat(ip));
}
public static String getTheNamesinGivenFormat(String[] ip) {
    StringBuffer sb=new StringBuffer();
    for(int i=0;i<ip.length;i++)
        sb.append(ip[i]).append(',');
    sb.deleteCharAt(sb.length()-1);
    return sb.toString();
}
}
-----------------------------------------------------------------------------------------------------||
4) Email Validation
  String input1="test@gmail.com"
        1)@ & . should be present;
        2)@ & . should not be repeated;
        3)there should be five charcters between @ and .;
        4)there should be atleast 3 characters before @ ;
        5)the end of mail id should be .com;

package Set3;
import java.util.*;
public class ClassSeT04 {
public static void main(String[] args) {
    String ip="test@gmail.com";
    boolean b=emailIdValidation(ip);
    if(b==true)
        System.out.println("valid mail Id");
    else
        System.out.println("not a valid Id");
}
public static boolean emailIdValidation(String ip) {
    int i=0;
    boolean b=false;
    StringTokenizer t=new StringTokenizer(ip,"@");
        String s1=t.nextToken();
        String s2=t.nextToken();
    StringTokenizer t1=new StringTokenizer(s2,".");
        String s3=t1.nextToken();
        String s4=t1.nextToken();
    if(ip.contains("@") && ip.contains("."))
        i++;

    if(i==1)
        if(s3.length()==5)
            if(s1.length()>=3)
                if(s4.equals("com"))
                    b=true;
    return b;
}
```

```
}
---------------------------------------------------------------------------------------------------||
5) Square root calculation of
   ((x1+x2)*(x1+x2))+((y1+y2)*(y1+y2))
   o/p should be rounded of to int;

package Set3;
public class ClassSeT05 {
public static void main(String[] args) {
        int x1=4,x2=8;
        int y1=3,y2=5;
        sqrt(x1,x2,y1,y2);
}
public  static void sqrt(int x1, int x2, int y1, int y2) {
        int op;
        op=(int) (Math.sqrt((x1+x2)*(x1+x2))+((y1+y2)*(y1+y2)));
        System.out.println(op);
}
}
---------------------------------------------------------------------------------------------------||
6) I/P hashmap<String String>{"ram:hari","cisco:barfi","honeywell:cs","cts:hari"};
   i/p 2="hari";
   o/p string[]={"ram","cts"};

package Set3;
import java.util.*;
import java.util.Map.Entry;
public class ClassSeT06 {
public static void main(String[] args) {
        HashMap<String,String> m1=new HashMap<String,String>();
        m1.put("ram","hari");
        m1.put("cisco","barfi");
        m1.put("honeywell","cs");
        m1.put("cts","hari");
        String s2="hari";
        getvalues(m1,s2);
}
public static void getvalues(HashMap<String, String> m1, String s2) {
        ArrayList<String>l1=new ArrayList<String>();
        for(Entry<String, String> m:m1.entrySet()){
                m.getKey();
                m.getValue();
                if(m.getValue().equals(s2))
                        l1.add(m.getKey());             }
        String[] op= new String[l1.size()];
        for(int i=0;i<l1.size();i++){
                op[i]=l1.get(i) ;
                System.out.println(op[i]);      }
}
}
---------------------------------------------------------------------------------------------------||
7) Input1={“ABX”,”ac”,”acd”};
   Input2=3;
   Output1=X$d
```

```
package Set3;
import java.util.*;
public class ClassSeT07 {
public static void main(String[] args) {
        String[] s1={"abc","da","ram","cat"};
        int ip=3;
        getStr(s1,ip);
}
public  static void getStr(String[] s1, int ip) {
        String op=" ";
        String s2=" ";
        ArrayList<String> l1=new ArrayList<String>();
        for(String s:s1)
                if(s.length()==ip)
                l1.add(s);
        StringBuffer buff=new StringBuffer();
        for(String l:l1){
            s2=l.substring(l.length()-1);
            buff.append(s2).append("$"); }
        op=buff.deleteCharAt(buff.length()-1).toString();
        System.out.println(op);
}
}
```

--||

8) INPUT1= helloworld
   INPUT2= 2. delete the char,if rpted twice.
   if occurs more than twice,leave the first occurence and delete the duplicate
   O/P= helwrd;

```
package Set3;
public class ClassSeT08 {
public static void main(String[] args) {
        String input1="HelloWorld";
        int input2=2;
    System.out.println(deletingtheCharOccuringTwice(input1,input2));
}
public static String deletingtheCharOccuringTwice(String input1, int input2) {
        StringBuffer sb=new StringBuffer(input1);
        int c=1;
        for(int i=0;i<sb.length();i++){
                c=1;
                for(int j=i+1;j<sb.length();j++)
                        if(sb.charAt(i)==sb.charAt(j))
                                c++;
                if(c>=input2){
                        for(int j=i+1;j<sb.length();j++)
                                if(sb.charAt(i)==sb.charAt(j))
                                        sb.deleteCharAt(j);
                        sb.deleteCharAt(i);
                        i--;   }        }
        return sb.toString();
}
}
```

---||

9) String[] input={"100","111","10100","10","1111"} input2="10"
  output=2;count strings having prefix"10" but "10" not included in count
  operation-- for how many strings input2 matches the prefix of each string in input1

  String[] input={"01","01010","1000","10","011"}
  output=3; count the strings having prefix"10","01" but "10","01" not included

```
package Set3;
import java.util.*;
public class ClassSeT09 {
public static void main(String[] args) {
	String[] ip={"100","111","10100","10","1111"};
	gteCount(ip);
}
public static void gteCount(String[] ip) {
	int op=0;
	ArrayList<String> l1=new ArrayList<String>();
	for(String s:ip)
		if(s.startsWith("10") || s.startsWith("01") &&(s.length()>2))
			l1.add(s);
	op=l1.size();
	System.out.println(op);
}
}
```

---||

10) input1=1 ,input2=2 ,input3=3 --- output=6;
   input1=1 ,input2=13,input3=3 --- output=1;
   input1=13,input2=2 ,input3=8 --- output=8;
   if value equal to 13,escape the value '13', as well as the next value to 13.
   sum the remaining values

```
package Set3;
import java.util.*;
public class ClassSeT10 {
public static void main(String[] args) {
	int ip1=13,ip2=2,ip3=8;
	System.out.println(thirteenLapse(ip1,ip2,ip3));
}
public static int thirteenLapse(int ip1, int ip2, int ip3) {
	List<Integer> l=new ArrayList<Integer>();
	l.add(ip1);
	l.add(ip2);
	l.add(ip3);
	int s=0;
	for(int i=0;i<l.size();i++){
		if(l.get(i)!=13)
			s+=l.get(i);
		if(l.get(i)==13)
			i=i+1;}
	return s;
}
}
```

---||

11) input="hello"
   output="hlo"; Alternative positions...

```
package Set3;
public class ClassSeT11 {
public static void main(String[] args) {
        String s="Hello";
        System.out.println(alternatingChar(s));
}
public static String alternatingChar(String s){
        StringBuffer sb=new StringBuffer();
        for(int i=0;i<s.length();i++)
                if(i%2==0)
                        sb.append(s.charAt(i));
        return sb.toString();
}
}
```

---||

12) Input1="Hello World"; output-------à "dello WorlH".

```
package Set3;
public class ClassSeT12 {
public static void main(String[] args) {
        String s="Hello World";
        System.out.println(reArrangingWord(s));
}
public static String reArrangingWord(String s) {
        StringBuffer sb=new StringBuffer();
        sb.append(s.substring(s.length()-1));
        sb.append(s.substring(1, s.length()-1));
        sb.append(s.substring(0, 1));
        return sb.toString();
}
}
```

---||

13) Collect no's frm list1 which is not present in list2
   & Collect no's frm list2 which is not present in list1
   and store it in output1[].
   ex: input1={1,2,3,4}; input2={1,2,3,5}; output1={4,5};

```
package Set3;
import java.util.*;
public class ClassSeT13 {
public static void main(String[] args) {
        List<Integer> l1=new ArrayList<Integer>();
        l1.add(1);
        l1.add(2);
        l1.add(3);
        l1.add(4);
        List<Integer> l2=new ArrayList<Integer>();
        l2.add(1);
        l2.add(2);
        l2.add(3);
        l2.add(5);
        int o[]=commonSet(l1,l2);
        for(int i:o)
        System.out.println(i);
}
public static int[] commonSet(List<Integer> l1, List<Integer> l2) {
        List<Integer> l3=new ArrayList<Integer>();
        List<Integer> l4=new ArrayList<Integer>();

        l3.addAll(l1);l4.addAll(l2);
        l1.removeAll(l2);l4.removeAll(l3);
        l1.addAll(l4);

        int o[]=new int[l1.size()];
        for(int j=0;j<o.length;j++)
                o[j]=l1.get(j);
        return o;
}
}
```

--||

14) String array will be given.if a string is Prefix of an any other string in that array means count.

```
package Set3;
public class ClassSeT14 {
public static void main(String[] args) {
        String[] a={"pinky","preethi","puppy","preeth","puppypreethi"};
        System.out.println(namesWithPreFixes(a));
}
public static int namesWithPreFixes(String[] a) {
        int n=0;
        for(int i=0;i<a.length;i++)
                for(int j=i+1;j<a.length;j++){
                String s1=a[i];
                String s2=a[j];
                if(s2.startsWith(s1)||s1.startsWith(s2))
                        n++;    }
        return n;
}
}
```

--||

15) count the number of words in the string
   Input string="i work in cognizant.";
   output=4;

```
package Set3;
import java.util.StringTokenizer;
public class ClassSeT15 {
public static void main(String[] args) {
        String s="I work for cognizant";
        System.out.println(noOfWordsInString(s));
}
public static int noOfWordsInString(String s) {
        StringTokenizer t=new StringTokenizer(s," ");
        return t.countTokens();
}
}
----------------------------------------------------------------------------------------------------||
16) int[] input={2,1,4,1,2,3,6};
   check whether the input has the sequence of "1,2,3". if so-
   output=true;
   int[] input={1,2,1,3,4,5,8};
   output=false

package Set3;
public class ClassSeT16 {
public static void main(String[] args) {
      //int[] a={2,1,4,1,2,3,6};
      int[] a={1,2,1,3,4,5,8};
      System.out.println(sequenceInArray(a));
}
public static boolean sequenceInArray(int[] a) {
      boolean b=false;
      int n=0;
      for(int i=0;i<a.length-1;i++)
            if((a[i+1]-a[i])==1)
                  n++;
      if(n==2)
            b=true;
      return b;
}
}
----------------------------------------------------------------------------------------------------||
17) input-- String input1="AAA/abb/CCC"
   char input2='/'
   output-- String[] output1;
   output1[]={"aaa","bba","ccc"};

   operation-- get the strings from input1  using stringtokenizer
            reverse each string
            then to lower case
            finally store it in output1[] string array

package Set3;
import java.util.*;
public class ClassSeT17 {
public static void main(String[] args) {
      String ip1="AAA/abb/CCC";
      char ip2='/';
      String op[]=loweringCasenReverseofaString(ip1,ip2);
      for(String s:op)
      System.out.println(s);
}
public static String[] loweringCasenReverseofaString(String ip1, char ip2){
      List<String> l=new ArrayList<String>();
      StringTokenizer t=new StringTokenizer(ip1,"/");
      while(t.hasMoreTokens()){
            StringBuffer sb=new StringBuffer(t.nextToken().toLowerCase());
            l.add(sb.reverse().toString()); }
      String op[]=new String[l.size()];
```

```
        for(int i=0;i<op.length;i++)
                op[i]=l.get(i);
        return op;
}
}
----------------------------------------------------------------------------------------------------||
18) Input1=”cowboy”; Output1=”cowcow”;
    Input1=”so”;output1=”sososo”;
    HINT: if they give 3 letter word u have to display 2 time;

package Set3;

public class ClassSeT18 {
public static void main(String[] args) {
        String ip1="cowboy";
        String ip2="cow";
        System.out.println(printingStringDependingOncharCount(ip1,ip2));
}
public static String printingStringDependingOncharCount(String ip1,String ip2) {
        StringBuffer sb=new StringBuffer();
        int n1=ip2.length();
        if(n1==3)
                for(int i=0;i<n1-1;i++)
                        sb.append(ip1.substring(0, n1));
        else if(n1==2)
                for(int i=0;i<n1+1;i++)
                        sb.append(ip1.substring(0, n1));
        return sb.toString();
}
}
----------------------------------------------------------------------------------------------------||
19) input---input1=1;
            input2=4;
            input3=1;
    output1=4;
    operation--- print the element which is not repeated
    if all the inputs r different sum all inputs
    input---input1=1;
            input2=2;
            input3=3;
    output1=6;

package Set3;
public class ClassSeT19 {
public static void main(String[] args) {
        int ip1=1,ip2=4,ip3=1;
        //int ip1=1,ip2=2,ip3=3;
        //int ip1=1,ip2=1,ip3=1;
        System.out.println(sumOfNonRepeatedChars(ip1,ip2,ip3));
}
public static int sumOfNonRepeatedChars(int ip1, int ip2, int ip3){
        int n=0;
        if(ip1!=ip2 && ip2!=ip3 && ip3!=ip1)
                n=ip1+ip2+ip3;
        else if(ip1==ip2 && ip2==ip3)
                n=0;
        else{
                if(ip1==ip2)
                        n=ip3;
                else if(ip1==ip3)
                        n=ip2;
                else if(ip2==ip3)
                        n=ip1;  }
        return n;
}
}
```

--||

20) input1-List1-{apple,orange,grapes}
input2-List2-{melon,apple,mango}
output={mango,orange}
operation-- In 1st list remove strings starting with 'a' or 'g'
In 2nd list remove strings ending with 'n' or 'e'
Ignore case, return in string array

```
package Set3;
import java.util.*;
public class ClassSeT20 {
public static void main(String[] args) {
        List<String> l1=new ArrayList<String>();
        l1.add("apple");
        l1.add("orange");
        l1.add("grapes");
        List<String> l2=new ArrayList<String>();
        l2.add("melon");
        l2.add("apple");
        l2.add("mango");
        String[] s2=fruitsList(l1,l2);
        for(String s3:s2)
                System.out.println(s3);
}
public static String[] fruitsList(List<String> l1, List<String> l2){
        List<String> l3=new ArrayList<String>();
        for(int i=0;i<l1.size();i++){
                String s1=l1.get(i);
                if(s1.charAt(0)!='a' && s1.charAt(0)!='A' && s1.charAt(0)!='g' && s1.charAt(0)!='G')
                        l3.add(s1); }
        for(int i=0;i<l2.size();i++){
                String s1=l2.get(i);
                if(s1.charAt(s1.length()-1)!='n' && s1.charAt(s1.length()-1)!='N' && s1.charAt(s1.length()-1)!='e' &&
s1.charAt(s1.length()-1)!='E')
                        l3.add(s1); }
        Collections.sort(l3);
        String[] s2=new String[l3.size()];
        for(int i=0;i<s2.length;i++)
                s2[i]=l3.get(i);
        return s2;
}
}
```

---||

21) input1-- Hello*world
output-- boolean(true or false)
operation-- if the character before and after * are same return true else false
if there in no star in the string return false(Ignore case)

```
package Set3;
import java.util.*;
public class ClassSeT21 {
public static void main(String[] args) {
        String input="Hello*world";
        System.out.println(characterCheck(input));
}
public static boolean characterCheck(String input) {
        boolean b=false;
        StringTokenizer t=new StringTokenizer(input,"*");
        String s1=t.nextToken();
        String s2=t.nextToken();
        String s3=s1.substring(s1.length()-1);
        String s4=s2.substring(0,1);
        if(s3.equalsIgnoreCase(s4))
                b=true;
        return b;
}
}
```

---||

22) input --String input1 ="xaXafxsd"
output--String output1="aXafsdxx"
operation-- remove the character "x"(only lower case) from string and place at the end

```
package Set3;
public class ClassSeT22 {
public static void main(String[] args) {
        String input="xaXafxsd";
        System.out.println(removalOfx(input));
}
public static String removalOfx(String input) {
        StringBuffer sb=new StringBuffer(input);
        int j=0;
        for(int i=0;i<sb.length();i++)
                if(sb.charAt(i)=='x'){
                        sb.deleteCharAt(i);
                        j++;}
        for(int i=0;i<j;i++)
                sb.append('x');
        return sb.toString();
}
}
```

---||

23) HashMap<String,Integer> h1={“abc”:50,”efg”:70};
if the mark is less than 60 then put the output in the
HashMap<String,String> h2={“abc”:”fail”,”efg”:”pass”}

```
package Set3;
```

```
import java.util.*;
public class ClassSeT23 {
public static void main(String[] args) {
    Map<String, Integer> m1=new HashMap<String, Integer>();
    m1.put("abc", 90);
    m1.put("efg", 50);
    m1.put("mno", 60);
    m1.put("rst", 75);
    m1.put("xyz", 35);
    System.out.println(examResult(m1));
}
public static Map<String,String> examResult(Map<String, Integer> m1) {
    Map<String,String> m2=new HashMap<String, String>();
    String s1=new String();
    String s2=new String();
    int n=0;
    Iterator<String> i=m1.keySet().iterator();
    while(i.hasNext()){
        s1=(String) i.next();
        n=m1.get(s1);
        if(n>=60)
            s2="PASS";
        else
            s2="FAIL";
        m2.put(s1, s2); }
    return m2;
}
}
----------------------------------------------------------------------------------------------------||
```

24) String i/p1=2012;
   sTRING i/p2=5
   IF EXPERIENCE IS GREATER THAN INPUT 2 THEN TRUE;

```
package Set3;
import java.text.*;
import java.util.*;
public class ClassSeT24 {
public static void main(String[] args) throws ParseException {
    String ip1="2012";
    String ip2="5";
    System.out.println(experienceCalc(ip1,ip2));
}
public static boolean experienceCalc(String ip1, String ip2) throws ParseException {
    boolean b=false;
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy");
        Date d1=sdf.parse(ip1);
        Date d2=new Date();
    int n1=d1.getYear();
    int n2=d2.getYear();
    int n3=Integer.parseInt(ip2);
    if((n2-n1)>n3)
        b=true;
    return b;
}
}
----------------------------------------------------------------------------------------------------||
25) input string="hello", n=2
   output: lolo

package Set3;
public class ClassSeT25 {
public static void main(String[] args) {
        String s1="hello";
        int n1=2;
        System.out.println(formattingOfString(s1,n1));
}
public static String formattingOfString(String s1, int n1) {
        String s2=s1.substring(s1.length()-n1, s1.length());
        StringBuffer sb=new StringBuffer();
        for(int i=0;i<n1;i++)
                sb.append(s2);
        return sb.toString();
}
}
----------------------------------------------------------------------------------------------------||
26) prove whether a number is ISBN number or not
   input="0201103311"
   ISBN number:  sum=0*10 +2*9+ 0*8 +1*7+ 1*6 +0*5+ 3*4 +3*3+ 1*2 +1*1
   sum%11==0 then it is ISBN number

package Set3;
public class ClassSeT26 {
public static void main(String[] args) {
        String ip="0201103311";
        boolean b=ISBNnumber(ip);
        if(b==true)
                System.out.println("valid ISBN number");
        else
                System.out.println("check ur data");
}
public static boolean ISBNnumber(String ip) {
        boolean b=false;
        int sum=0;
        for(int i=0,j=ip.length();i<ip.length();i++,j--){
                String s=String.valueOf(ip.charAt(i));
                int n=Integer.parseInt(s);
                sum+=(n*j);     }
        //System.out.println(sum);
        if(sum%11==0)
                b=true;
        return b;
}
}
----------------------------------------------------------------------------------------------------||
27) Validate Password
   validation based on following criteria:
  -> minimum length is 8
  -> should contain any of these @/_/#
```

-> should not start with number/special chars(@/#/_)
-> should not end with special chars
-> can contain numbers,letters,special chars

```
package Set3;
import java.util.*;
public class ClassSeT40 {
public static void main(String[] args) {
	Scanner s=new Scanner(System.in);
	String s1=s.next();
	boolean b=passwordValidation(s1);
	if(b==true)
		System.out.println("valid password");
	else
		System.out.println("not a valid password");
}
public static boolean passwordValidation(String s1) {
	boolean b=false,b1=false,b2=false;
	if(s1.length()>=8)
		if(!Character.isDigit(s1.charAt(0)))
			if(s1.charAt(0)!='@' && s1.charAt(0)!='_' && s1.charAt(0)!='#')
				if(s1.charAt(s1.length()-1)!='@' && s1.charAt(s1.length()-1)!='_' && s1.charAt(s1.length()-1)!='#')
					b1=true;
	if(b1==true)
		for(int i=0;i<s1.length();i++)
			if(Character.isAlphabetic(s1.charAt(i)) || Character.isDigit(s1.charAt(i)) || s1.charAt(i)=='#' || s1.charAt(i)=='@' || s1.charAt(i)=='_')
				b2=true;
	if(b2==true)
		if(s1.contains("#") || s1.contains("@") || s1.contains("_"))
			b=true;
	return b;
}
}
-----------------------------------------------------------------------------------------------||
```

28) pan card number validation:
all letters shud be in caps,shud be of 8 chars.
first three letters must be alphabets.
next 4 letters shud be digits and last letter shud be an alphabet

```
package Set3;
import java.util.*;
public class ClassSeT28 {
public static void main(String[] args) {
	Scanner s=new Scanner(System.in);
	String pan=s.next();
	boolean b=panNumberValidation(pan);
	if(b==true)
		System.out.println("valid Pancard Number");
	else
		System.out.println("not a valid credential");
}
public static boolean panNumberValidation(String pan) {
        boolean b=false,b1=false,b2=false;
        String s1=pan.substring(0, 3);
        String s2=pan.substring(3, 7);
        if(pan.length()==8)
                if(Character.isAlphabetic(pan.charAt(pan.length()-1)) && Character.isUpperCase(pan.charAt(pan.length()-
1)))
                        b1=true;
        if(b1==true)
                for(int i=0;i<s1.length();i++)
                        if(Character.isAlphabetic(s1.charAt(i)) && Character.isUpperCase(s1.charAt(i)))
                                b2=true;
                        else
                                {b2=false;break;}

        if(b2==true)
                for(int i=0;i<s2.length();i++)
                        if(Character.isDigit(s2.charAt(i)))
                                b=true;
                        else
                        {b=false;break;}
        return b;
}
}
```

---||

29) In a hashmap if key is odd then find average of value as integer
   ex: h1={1:4,2:6,4:7,5:9}
        output=(4+9)/2

```
package Set3;
import java.util.*;
public class ClassSeT29 {
public static void main(String[] args) {
        Map<Integer, Integer> m1=new HashMap<Integer, Integer>();
        m1.put(1, 4);
        m1.put(2, 6);
        m1.put(4, 7);
        m1.put(5, 9);
        System.out.println(avgValuesOfOddKeys(m1));
}
public static int avgValuesOfOddKeys(Map<Integer, Integer> m1) {
        int l=0,m=0;
        Iterator<Integer> i=m1.keySet().iterator();
        while(i.hasNext()){
                int n=(Integer) i.next();
                if(n%2!=0){
                        m+=m1.get(n);
                        l++;}   }
        return m/l;
}
}
```

---||

30) Return 1 if the last & first characters of a string are equal else
   return -1. Consider case.
   Eg: Input = "this was great"

	Output= 1

```
package Set3;
public class ClassSeT30 {
public static void main(String[] args) {
	String input="this was great";
	System.out.println(checkForFirstAndLastChar(input));
}
public static int checkForFirstAndLastChar(String input) {
	int n=0;
	if(input.charAt(0)==input.charAt(input.length()-1))
		n=1;
	else n=-1;
	return n;
}
}
```

--||

31) concat two string if length of two string is equal.
   if length of one string is greater, then remove the character from
   largest string and then add. The number of characters removed from
   largest string is equal to smallest string's length
   for example: input 1="hello";
                input 2="helloworld";
                output="worldhello";

```
package Set3;
public class ClassSeT31 {
public static void main(String[] args) {
	String ip1="hello";
	String ip2="helloworld";
	System.out.println(removalOfCharFromLargestString(ip1,ip2));
}
public static String removalOfCharFromLargestString(String ip1,String ip2){
	StringBuffer sb=new StringBuffer();
	int n1=ip1.length();
	int n2=ip2.length();
	if(n1<n2)
		sb.append(ip2.substring(n1, n2)).append(ip1);
	return sb.toString();
}
}
```

--||

32) i/p: Honesty is my best policy
   o/p: Honesty
   Return the maximum word length from the given string.
   If there are two words of same length then,
   return the word which comes first based on alphabetical order.

```
package Set3;
import java.util.*;
public class ClassSeT32 {
public static void main(String[] args) {
	String s1="is a poppppy preethi";
	System.out.println(lengthiestString(s1));
}
public static String lengthiestString(String s1) {
	int max=0;
	String s2=new String();
	StringTokenizer t=new StringTokenizer(s1," ");
	loop:
	while(t.hasMoreTokens()){
		String s3=t.nextToken();
		int n=s3.length();
		if(n>max){
			max=n;
			s2=s3;}
		if(n==max)
			for(int i=0;i<s3.length();i++){
				char c1=s2.charAt(i);
				char c2=s3.charAt(i);
				if(c1!=c2){
					if(c2<c1)
						s2=s3;
					continue loop;}   }
	}
	return s2;
}
}
```

--||

33) In a string check whether all the vowels are present
   if yes return 1 else 2.
   ex: String 1="education"
      output=1.

```
package Set3;
public class ClassSeT33 {
public static void main(String[] args) {
	String s1="education";
	System.out.println(vowelsCheck(s1));
}
public static boolean vowelsCheck(String s1) {
	boolean b=false;
	int n1=0,n2=0,n3=0,n4=0,n5=0;
	for(int i=0;i<s1.length();i++){
		char c=s1.charAt(i);
		if(c=='a' || c=='A')
			n1++;
		if(c=='e' || c=='E')
			n2++;
		if(c=='i' || c=='I')
			n3++;
		if(c=='o' || c=='O')
			n4++;
		if(c=='u' || c=='U')
			n5++;}
	if(n1==1 && n2==1 && n3==1 && n4==1 && n5==1)
		b=true;
	return b;
}
}
```

---||

34) swap the every 2 chrecters in the given string
If size is odd number then keep the last letter as it is.
Ex:- input: forget
output: ofgrte
Ex:- input : NewYork
output : eNYwrok

```
package Set3;
public class ClassSet34 {
public static void main(String[] args) {
        String s1="newyork";
        System.out.println(formattingGivenString(s1));
}
public static String formattingGivenString(String s1) {
        StringBuffer sb=new StringBuffer();
        int j=0;
        if(s1.length()%2==0)
                j=s1.length();
        else
                j=s1.length()-1;

        for(int i=0;i<j;i=i+2){
                String s2=(s1.substring(i, i+2));
                StringBuffer sb1=new StringBuffer(s2);
                sb.append(sb1.reverse());}

        String s3=new String();
        if(s1.length()%2==0)
                s3=sb.toString();
        else
                s3=sb.append(s1.charAt(s1.length()-1)).toString();
        return s3;
}
}
```

---||

35) i/p: bengal
o/p: ceogbl
if z is there replace with a

```
package Set3;
public class ClassSeT35 {
public static void main(String[] args) {
        String s1="bengal";
        System.out.println(stringFormatting(s1));
}
public static String stringFormatting(String s1) {
        StringBuffer sb=new StringBuffer();
        for(int i=0;i<s1.length();i++){
                char c=s1.charAt(i);
                if(i%2==0){
                        if(c==122)
                c=(char) (c-25);
            else{
                c=(char) (c+1);}
                sb.append(c);}
        else
            sb.append(c);}
    return sb.toString();
}
}
```

--||

36) find the maximum chunk of a given string
   i/p: this isssss soooo good
   o/p=5

```
package Set3;
import java.util.*;
public class ClassSeT36 {
public static void main(String[] args) {
    String s1="this is sooo good";
    System.out.println(maxChunk(s1));
}
public static int maxChunk(String s1) {
    int max=0;
    StringTokenizer t=new StringTokenizer(s1," ");
    while(t.hasMoreTokens()){
        String s2=t.nextToken();
        int n=0;
        for(int i=0;i<s2.length()-1;i++)
            if(s2.charAt(i)==s2.charAt(i+1))
                n++;
        if(n>max)
            max=n;
    }
    return (max+1);
}
}
```

--||

37) i/p1: new york
   i/p2: new jersey
   o/p: new y+r+

solution:

```
package Set3;
public class ClassSeT37 {
public static void main(String[] args) {
    String s1="New york";
    String s2="New jersey";
    System.out.println(commonCharsinAstring(s1,s2));
}
private static String commonCharsinAstring(String s1, String s2) {
    int f;
    StringBuffer sb=new StringBuffer();
    for(int i=0;i<s1.length();i++){
        f=0;
        char c1=s1.charAt(i);
        for(int j=0;j<s2.length();j++)
            if(c1==s2.charAt(j))
                f=1;
        if(f!=0)
            sb.append(c1);
        else
            sb.append('+');}
    return sb.toString();
}
}
```

--||

38) input1={2,4,3,5,6};
   if odd find square
   if even find cube
   finally add it
   output1=208(4+16+27+125+36)

```
package Set3;
public class ClassSeT38 {
public static void main(String[] args) {
    int a[]={2,4,3,5,6};
    System.out.println(summationPattern(a));
}
public static int summationPattern(int[] a) {
    int n1=0,n2=0;
    for(int i=0;i<a.length;i++)
        if(a[i]%2==0)
            n1+=(a[i]*a[i]);
        else
            n2+=(a[i]*a[i]*a[i]);
    return n1+n2;
}
}
```

--||

39) input1="the sun raises in the east";
   output1=raises;
   count no vowels in each word and print the word which has max
   no of vowels if two word has max no of vowel print the first one

```
package Set3;
import java.util.*;
public class ClassSeT39 {
public static void main(String[] args) {
    String s1="the sun rises in the east";
    System.out.println(wordWithMaxVowelCount(s1));
}
public static String wordWithMaxVowelCount(String s1) {
    int max=0;
    String s2="aeiouAEIOU";
    String s3=new String();
    StringTokenizer t=new StringTokenizer(s1," ");
    while(t.hasMoreTokens()){
        String s4=t.nextToken();
        int c=0;
        for(int i=0;i<s4.length();i++)
            for(int j=0;j<s2.length();j++)
                if(s4.charAt(i)==s2.charAt(j))
                    c++;
        if(c>max){
            max=c;
            s3=s4;} }
    return s3;
}
}
---------------------------------------------------------------------------------------------------||
40) String format : CTS-LLL-XXXX
   ip1: CTS-hyd-1234
   ip2: hyderabad
   -> LLL must be first 3 letters of ip2.
   -> XXXX must be a 4-digit number

package Set3;
import java.util.*;
public class ClassSeT41 {
public static void main(String[] args) {
    String s1="CTS-hyd-1234";
    String s2="hyderabad";
    boolean b=formattingString(s1,s2);
    if(b==true)
        System.out.println("String format:CTS-LLL-XXXX");
    else
        System.out.println("not in required format");
}
public static boolean formattingString(String s1, String s2) {
    String s3=s2.substring(0, 3);
    boolean b=false;
    StringTokenizer t=new StringTokenizer(s1,"-");
    String s4=t.nextToken();
    String s5=t.nextToken();
    String s6=t.nextToken();
    if(s4.equals("CTS") && s5.equals(s3) && s6.length()==4)
        for(int i=0;i<s6.length();i++){
            if(Character.isDigit(s6.charAt(i)))
                b=true;
            else{
                b=false;}}
    return b;
}
}
---------------------------------------------------------------------------------------------------||
41) ip: "this is sample test case"
   op: "this amplec"
   remove the duplicates in the given string

package Set3;
import java.util.*;
public class ClassSeT42 {
```

```
public static void main(String[] args) {
        String s1="this is sample test case";
        System.out.println(removeDuplicates(s1));
}
public static String removeDuplicates(String s1) {
        StringBuffer sb=new StringBuffer();
        Set<Character> c1=new LinkedHashSet<Character>();
        for(int i=0;i<s1.length();i++)
                c1.add(s1.charAt(i));
        for(char c2:c1)
                sb.append(c2);
        return sb.toString();
}
}
----------------------------------------------------------------------------------------------------||
42) input1 is a map<Integer,Float>
    {1:2.3,2:5.6,3:7.7,4:8.4}
    get the even number from keys and find the avg of values of these keys.
    answer should be rounded to two numbers after decimal
    eg:- the output number 15.2499999 should be 15.25

package Set3;
import java.util.*;
public class ClassSeT43 {
public static void main(String[] args) {
        Map<Integer,Float> m1=new HashMap<Integer, Float>();
        m1.put(1, (float) 12.93);
        m1.put(2, (float) 15.67);
        m1.put(3, (float) 17.27);
        m1.put(4, (float) 14.88);
        System.out.println(avgOfEvenKeyValues(m1));
}
public static float avgOfEvenKeyValues(Map<Integer, Float> m1) {
        int n1=0,n3=0;
        float n2=0;
        Iterator<Integer> i=m1.keySet().iterator();
        while(i.hasNext()){
                n1=(Integer) i.next();
                if(n1%2==0){
                        n2+=m1.get(n1);
                        n3++;}  }
        float n=Math.round((n2/n3)*100)/100f;
        return n;
}
}
----------------------------------------------------------------------------------------------------||
43) Color Code Validation:
        String should starts with the Character '#'.
        Length of String is 7.
        It should contain 6 Characters after '#' Symbol.
        It should contain Characters Between 'A-F' and Digits '0-9'.
        if String is acceptable then Output1=1
        else  Output1=-1;
```

```
package Set3;
import java.util.*;
public class ClassSeT44 {
public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        String s1=s.next();
        boolean b=colorCodeValidation(s1);
        if(b==true)
                System.out.println("valid color code");
        else
                System.out.println("invalid color code");
}
public static boolean colorCodeValidation(String s1) {
        boolean b=false,b1=false;
        String s2=s1.substring(1,s1.length());
        if(s1.length()==7)
                if(s1.charAt(0)=='#')
                        b1=true;

        if(b1==true)
                for(int i=0;i<s2.length();i++){
                        char c=s2.charAt(i);
                        if(c!='#'){
                        if((Character.isAlphabetic(c)&& Character.isUpperCase(c)) || Character.isDigit(c))
                                b=true;
                        else{
                                b=false;
                                break;}}}
        return b;
}
}
```

---||

44) Find the Maximum span of the given array.
span is the number of elements between the duplicate element
including those 2 repeated numbers.
if the array as only one elemnt,then the span is 1.
input[]={1,2,1,1,3}
output1=4
input[]={1,2,3,4,1,1,5}
output1=6

```
package Set3;
public class ClassSeT45 {
public static void main(String[] args) {
        int[]a={1,2,1,1,3};
        System.out.println(maxSpan(a));
}
public static int maxSpan(int[] a) {
        String s2 = null;
        int n=0;
        StringBuffer sb=new StringBuffer();
        for(int i=0;i<a.length;i++)
                sb.append(String.valueOf(a[i]));
        String s1=sb.toString();
        for(int i=0;i<s1.length();i++)
                for(int j=i+1;j<s1.length();j++)
                        if(s1.charAt(i)==s1.charAt(j))
                                s2=String.valueOf(s1.charAt(j));
        int n1=s1.indexOf(s2);
        int n2=s1.lastIndexOf(s2);
        for(int i=n1+1;i<n2;i++)
                n++;
        return (n+2);
}
}
```

---||

45) Getting the first and last n letters from a word where wordlength > 2n.
Ex: Input: california,3.
output: calnia.

```
package Set3;
public class ClassSeT46 {
public static void main(String[] args) {
        String s1="california";
        int n1=3;
        System.out.println(subStringOfgivenString(s1,n1));
}
public static String subStringOfgivenString(String s1, int n1) {
        StringBuffer sb=new StringBuffer();
        sb.append(s1.substring(0, n1)).append(s1.substring(s1.length()-n1,s1.length()));
        return sb.toString();
}
}
```

---||

46) input1="aBrd";
input2="aqrbA";
input3=2;
output1=boolean true;
2nd char of ip1 and last 2nd char of ip2 show be equal

```
package Set3;
public class ClassSeT46 {
public static void main(String[] args) {
        String ip1="aBrd";
        String ip2="aqrbA";
        int ip3=2;
        System.out.println(charCheck(ip1,ip2,ip3));
}
public static boolean charCheck(String ip1, String ip2, int ip3){
        boolean b=false;
        String s1=String.valueOf(ip1.charAt(ip3-1));
        String s2=String.valueOf(ip2.charAt(ip2.length()-ip3));
        if(s1.equalsIgnoreCase(s2))
                b=true;
        return b;
}
}
```

---||

```
47) Add elements of digits:9999
    output:9+9+9+9=3+6=9;

package Set3;
public class ClassSeT47 {
public static void main(String[] args) {
        int n=9999;
        System.out.println(conversiontoaSingleDigit(n));
}
public static int conversiontoaSingleDigit(int n){
        loop:
                while(n>10){
                        int l=0,m=0;
                        while(n!=0){
                                m=n%10;
                                l=l+m;
                                n=n/10; }
                        n=l;
                        continue loop; }
        return n;
}
}
-----------------------------------------------------------------------------------------------------||
48) leap year or not using API?

package Set3;
import java.util.*;
public class ClassSeT48 {
public static void main(String[] args) {
        String s="2013";
        System.out.println(leapYear(s));
}
public static boolean leapYear(String s) {
        int n=Integer.parseInt(s);
        GregorianCalendar c=new GregorianCalendar();
        boolean b=c.isLeapYear(n);
        return b;
}
}
-----------------------------------------------------------------------------------------------------||
49) perfect no or not?

package Set3;
public class ClassSeT49 {
public static void main(String[] args) {
        int n=28;
        System.out.println(perfectNumber(n));
}
public static boolean perfectNumber(int n) {
        int n1=0;
        boolean b=false;
        for(int i=1;i<n;i++)
                if(n%i==0)
                        n1+=i;
        //System.out.println(n1);
        if(n1==n)
                b=true;
        return b;
}
}
-----------------------------------------------------------------------------------------------||
50) HashMap<String,String> input1={“mouse”:”100.2”,”speaker”:”500.6”,”Monitor”:”2000.23”};
   String[] input2={“speaker”,”mouse”};
   Float output=600.80(500.6+100.2);

package Set3;
import java.util.*;
public class ClassSeT50 {
public static void main(String[] args) {
        HashMap<String, String> m1=new HashMap<String, String>();
        m1.put("mouse", "100.2");
        m1.put("speaker","500.6");
        m1.put("monitor", "2000.23");
        String[] s={"speaker","mouse"};
        System.out.println(getTheTotalCostOfPheripherals(m1,s));
}
public static float getTheTotalCostOfPheripherals(HashMap<String,String> m1,
                String[] s) {
        Float f=(float) 0;
        Iterator<String> i=m1.keySet().iterator();
        while(i.hasNext()){
        String s1=(String) i.next();
        Float f1=Float.parseFloat(m1.get(s1));
        for(int j=0;j<s.length;j++)
                if(s[j].equals(s1))
                        f+=f1;  }
        return f;
}
}
-----------------------------------------------------------------------------------------------||
51) Input1=845.69, output=3:2;
   Input1=20.789; output=2:3;
   Input1=20.0; output=2:1;
   output is in Hashmap format.
   Hint: count the no of digits.

package Set3;
import java.util.*;
public class ClassSeT51 {
public static void main(String[] args) {
        double d=845.69;
        System.out.println(noOfDigits(d));
}
public static String noOfDigits(double d) {
        int n1=0,n2=0;
        String s=String.valueOf(d);
        StringTokenizer t=new StringTokenizer(s,".");
        String s1=t.nextToken();
        String s2=t.nextToken();
        n1=s1.length();
        n2=s2.length();
        if(s1.charAt(0)=='0')
                n1=s1.length()-1;
        if(n2!=1)
                if(s2.charAt(s2.length()-1)=='0')
                        n2=s2.length()-1;
        String s3=String.valueOf(n1)+":"+String.valueOf(n2);
        return s3;
}
}
```

```
--------------------------------------------------------------------------------------------------||
--------------------------------------------------------------------------------------------------||
--------------------------------------------------------------------------------------------------||
--------------------------------------------------------------------------------------------------||
--------------------------------------------------------------------------------------------------||

--------------------------------------SET 2-----------------------------------------------------
```

1) compare two strings, if the characters in string 1 are present in
string 2, then it should be put as such in output, else '+' should be
put in output...ignore case difference.

```
        input 1:"New York"
        input 2:"NWYR"

        output:N+w+Y+r+
```

```
package Set2;
public class ClassSET1 {
public static void main(String[] args) {
                String s1="New York";
                String s2="NWYR";
                System.out.println(StringFormatting(s1,s2));
        }
public static String StringFormatting(String s1, String s2) {
        StringBuffer s4=new StringBuffer();
        String s3=s1.toUpperCase();
        for(int i=0;i<s2.length();i++)
                for(int j=0;j<s3.length();j++)
                        if(s2.charAt(i)==s3.charAt(j))
                                s4.append(s1.charAt(j)).append('+');
        return s4.toString();
}
}
```

```
---------------------------------------------------------------------------------------------------------||
```

2) input:

```
        Searchstring s1="GeniusRajkumarDev";
        String s2="Raj";
        String s3="Dev";
   output:
        Return 1 if s2 comes before s3 in searchstring else return 2
```

```
package Set2;
public class ClassSET2 {
public static void main(String[] args) {
        String srch="MaanVeerSinghKhurana";
        String s1="Veer";
        String s2="Singh";
        int n=searchString(srch,s1,s2);
        if(n==1)
                System.out.println(s1+" comes before "+s2);
        else
                System.out.println(s2+" comes before "+s1);
}
public static int searchString(String srch, String s1, String s2) {
        int n=0;
        int n1=srch.indexOf(s1);
        int n2=srch.indexOf(s2);
        if(n1<n2)
                n=1;
        else
                n=2;
        return n;
}
}
```

--||

3) months between two dates

```
package Set2;
import java.text.*;
import java.util.*;
public class ClassSET3 {
public static void main(String[] args) throws ParseException {
        String s1="30/05/2013";
        String s2="01/06/2013";
        System.out.println(monthsBetweenDates(s1,s2));
}
public static int monthsBetweenDates(String s1, String s2) throws ParseException {
        SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
                Date d1=sdf.parse(s1);
                Date d2=sdf.parse(s2);
        Calendar cal=Calendar.getInstance();

        cal.setTime(d1);
                int months1=cal.get(Calendar.MONTH);
                int year1=cal.get(Calendar.YEAR);
        cal.setTime(d2);
                int months2=cal.get(Calendar.MONTH);
                int year2=cal.get(Calendar.YEAR);
        int n=((year2-year1)*12)+(months2-months1);
        return n;
}
}
------------------------------------------------------------------------------------------------------------||
```

4) intput="xyzwabcd"
    intput2=2;
    input3=4;
    output=bawz

```
package Set2;
public class ClassSET4 {
public static void main(String[] args) {
        String s1="xyzwabcd";
        int n1=2,n2=4;
        System.out.println(retrievalOfString(s1,n1,n2));
}

public static String retrievalOfString(String s1, int n1, int n2) {
        StringBuffer sb=new StringBuffer(s1);
        sb.reverse();
        String s2=sb.substring(n1, n1+n2);
        return s2;
}
}
------------------------------------------------------------------------------------------------------------||
```

5)   Given integer array
        input :int[] arr={2,3,5,4,1,6,7,7,9};

Remove the duplicate elements and print sum of even numbers in the array..
print -1 if arr contains only odd numbers

```
package Set2;
import java.util.*;
public class ClassSET5 {
public static void main(String[] args) {
        int a[]={2,3,5,4,1,6,7,7,9};
        System.out.println(sumOfEvenNos(a));
}
public static int sumOfEvenNos(int[] a) {
        List<Integer> l1=new ArrayList<Integer>();
        for(int i=0;i<a.length;i++)
```

```
            l1.add(a[i]);
        List<Integer> l2=new ArrayList<Integer>();
        for(int i=0;i<a.length;i++)
        for(int j=i+1;j<a.length;j++)
            if(a[i]==a[j])
                l2.add(a[j]);
        l1.removeAll(l2);
        l1.addAll(l2);
        int n=0,n1;
        for(int i=0;i<l1.size();i++)
            if(l1.get(i)%2==0)
                n+=l1.get(i);
        if(n==0)
            n1=-1;
        else
            n1=n;
        return n1;
}
}
----------------------------------------------------------------------------------------------------||
```

6)   input1="abc2012345"
        input2="abc2112660"
        input 3=4
here "abc**" refers to customer id.
12345 refers to last month eb reading and 12660 refers to this month eb reading
find the difference between two readings and multiply it by input3

ie., output=(12660-12345)*4

```
package Set2;
public class ClassSET6 {
public static void main(String[] args) {
        String input1="abc2012345";
        String input2="abc2112660";
        int input3=4;
        System.out.println(meterReading(input1,input2,input3));
}
public static int meterReading(String input1, String input2, int input3) {
        int n1=Integer.parseInt(input1.substring(5, input1.length()));
        int n2=Integer.parseInt(input2.substring(5, input2.length()));
        int n=Math.abs((n2-n1)*input3);
        return n;
}
}
----------------------------------------------------------------------------------------------------||
```

7)Given array of intergers,print the sum of elements sqaring/cubing as per the power of their indices.
//answer= sum=sum+a[i]^i;
     eg:input:{2,3,5}
        Output:29

```
package Set2;
public class ClassSET7 {
public static void main(String[] args) {
        int a[]={2,3,5};
        System.out.println(sumOfElementsWrtIndices(a));
        }
public static int sumOfElementsWrtIndices(int[] a) {
        int s=0;
        for(int i=0;i<a.length;i++)
                s+=(Math.pow(a[i], i));
        return s;
}
}
```

--||

8)Given array of string check whether all the elements contains only digits not any alaphabets.
if condn satisfied print 1 else -1.
Input:{"123","23.14","522"}
output:1

Input1:{"asd","123","42.20"}
output:-1

```
package Set2;
public class ClassSET8 {
public static void main(String[] args) {
        String[] input1={"123","23.14","522"};
        //String[] input1={"asd","123","42.20"};
        System.out.println(stringOfDigits(input1));
}
public static int stringOfDigits(String[] input1) {
        int n=0;
        for(int i=0;i<input1.length;i++){
                String s1=input1[i];
                for(int j=0;j<s1.length();j++){
                        char c1=s1.charAt(j);
                        if(Character.isDigit(c1))
                                n=1;
                        else {n=-1;break;} }}
        return n;
}
}
```

--||

9) int[] a={12,14,2,26,35}
find the difference b/w max and min values in array
output:35-2=33.

```
package Set2;
import java.util.Arrays;
public class ClassSET9 {
public static void main(String[] args) {
        int a[]={12,14,2,26,35};
        System.out.println(diffBwMaxAndMin(a));
}
public static int diffBwMaxAndMin(int[] a) {
        Arrays.sort(a);
        int n=a[a.length-1]-a[0];
        return n;
}
}
----------------------------------------------------------------------------------------------------||
```

```
10) Given an array find the largest 'span'(span is the number of elements between two same digits)
    find their sum.
        a[]={1,4,2,1,4,1,5}
        Largest span=5
```

```
package Set2;
public class ClassSET10 {
public static void main(String[] args) {
        int a[]={1,4,2,1,4,1,5};
        System.out.println("sum of largest span elements:"+largestSpan(a));
}
public static int largestSpan(int[] a) {
        int max=0;
        int p1=0;
        int p2=0;
        int n=0;
        int sum=0;
        for(int i=0;i<a.length-1;i++){
                for(int j=i+1;j<a.length;j++)
                        if(a[i]==a[j])
                                n=j;
                if(n-i>max){
                        max=n-i;
                        p1=i;
                        p2=n; }
        }
        System.out.println("largest span:"+(p2-p1));
        for(int i=p1;i<=p2;i++)
                sum=sum+a[i];
        return (sum);
}
}
```

--||

11) input={"1","2","3","4"}
   output=10
ie,if string elements are nos,add it.
   input={"a","b"}
   output=-1;

```
package Set2;
public class ClassSET11 {
public static void main(String[] args) {
	String s[]={"1","2","3","4"};
	//String s[]={"a","b","3","4"};
	System.out.println(checkForStringElements(s));
}
public static int checkForStringElements(String[] s) {
	int n=0;
	boolean b=false;
	for(int i=0;i<s.length;i++){
		String s1=s[i];
		for(int j=0;j<s1.length();j++){
			char c=s1.charAt(j);
			if(Character.isDigit(c))
				b=true;
			else{
				b=false;
				break;}		}
		if(b==true)
			n+=Integer.parseInt(s1);
		else{
			n=-1;
			break;}		}
	return n;
}
}
```

--||

12) arraylist1={ 1,2,3,4,5}
   arraylist2={ 6,7,8,9,10}
size of both list should be same
output={6,2,8,4,10}

```
package Set2;
public class ClassSET12 {
public static void main(String[] args) {
	int a[]={1,2,3,4,5};
	int b[]={6,7,8,9,10};
	int c[]=alternativeIndicesElements(a,b);
```

```
        for(int d:c)
        System.out.println(d);
}
public static int[] alternativeIndicesElements(int[] a, int[] b){
        int c[]=new int[a.length];
        if(a.length==b.length)
                for(int i=0;i<a.length;i++)
                        if(i%2==0)
                                c[i]=b[i];
                        else
                                c[i]=a[i];
        return c;
}
}
```

--||

13) count the number of times the second word in second string occurs in first string-case sensitive

input1=hai hello hai where hai Hai;
input2=what hai
output=3;

```
package Set2;
import java.util.StringTokenizer;
public class ClassSET13 {
public static void main(String[] args) {
        String input1="hai hello how are you?? hai hai";
        String input2="what hai";
        System.out.println(stringOccurance(input1,input2));
}

public static int stringOccurance(String input1, String input2){
        int count=0;
        StringTokenizer t1=new StringTokenizer(input2," ");
        String s1=t1.nextToken();
        String s2=t1.nextToken();
        StringTokenizer t2=new StringTokenizer(input1," ");
        while(t2.hasMoreTokens()){
                if(t2.nextToken().equals(s2))
                        count++;        }
        return count;
}
}
```

---||

14) find the no.of charcters,that has repeated 3 consecutive times
input1:"aaabbccc"
ouput1=2;

```
package Set2;
public class ClassSET14 {
public static void main(String[] args) {
        String input1="aaabbccc";
        System.out.println(consecutiveRepeatitionOfChar(input1));
}

public static int consecutiveRepeatitionOfChar(String input1) {
        int c=0;
        int n=0;
        for(int i=0;i<input1.length()-1;i++){
                if(input1.charAt(i)==input1.charAt(i+1))
                        n++;
                else
                        n=0;

                if(n==2)
                        c++; }
        return c;
}
}
```

--||

15)What will be the DAY of current date in next year

```
package Set2;
import java.text.SimpleDateFormat;
import java.util.*;
public class ClassSet15 {
public static void main(String[] args) {
        Date d=new Date();
        System.out.println(sameDayOnUpcomingYear(d));
}
public static String sameDayOnUpcomingYear(Date d) {
        Date d1=new Date();
        d1.setYear(d.getYear()+1);
        SimpleDateFormat sdf=new SimpleDateFormat("EEEE");
        String s=sdf.format(d1);
        return s;
}
}
```

--||

16)Get all the numbers alone from the string and return the sum.

Input"123gif"
Output:6

```
package Set2;
public class ClassSET16 {
public static void main(String[] args) {
	String s="1234gif";
	System.out.println(summationOfNosInaString(s));
}
public static int summationOfNosInaString(String s) {
	int n=0;
	for(int i=0;i<s.length();i++){
		char c=s.charAt(i);
		if(Character.isDigit(c)){
			String s1=String.valueOf(c);
			n+=Integer.parseInt(s1);} }
	return n;
}
}
```

---||

17)input array={red,green,blue,ivory}
sort the given array
reverse the given array
if user input is 1 it should give oth element of an reversed array.

```
package Set2;
import java.util.*;
public class ClassSET17 {
public static void main(String[] args) {
	String[] s={"red","green","blue","ivory","yellow"};
	int n=1;
	System.out.println(retrievingRequiredColor(s,n));
}
public static String retrievingRequiredColor(String[] s, int n) {
	String s1=new String();
	List<String> l=new ArrayList<String>();
	for(int i=0;i<s.length;i++)
		l.add(s[i]);
	Collections.sort(l,Collections.reverseOrder());
	for(int i=0;i<l.size();i++)
		if(i==(n-1))
			s1=l.get(i);
	return s1;
}
}
```

---||

18)String a="a very fine day"
output:A Very Fine Day

```
package Set2;
import java.util.StringTokenizer;
public class ClassSET18 {
public static void main(String[] args){
        String s1="its a very fine day";
        System.out.println(capsStart(s1));
}
public static String capsStart(String s1){
        StringBuffer s5=new StringBuffer();
        StringTokenizer t=new StringTokenizer(s1," ");
        while(t.hasMoreTokens()){
                String s2=t.nextToken();
                String s3=s2.substring(0,1);
                String s4=s2.substring(1, s2.length());
                s5.append(s3.toUpperCase()).append(s4).append(" ");     }
        return s5.toString();
}
}
--------------------------------------------------------------------------------------------------------||
```

19)Take the word with a max length in the given sentance
in that check for vowels if so count the no.of occurances !

```
  Input 1="Bhavane is a good girl"
  Output =3
  Input 1="Bhavanee is a good girl"
  Output =4
```

```
package Set2;
import java.util.StringTokenizer;
public class ClassSET19 {
public static void main(String[] args) {
        String s1="Bhavane is a good girl";
        System.out.println(countVowelsInMaxLengthString(s1));
}
public static int countVowelsInMaxLengthString(String s1) {
        int n1=0, max=0;
        String s4="AEIOUaeiou";
        String s3=new String();
        StringTokenizer t=new StringTokenizer(s1," ");
        while(t.hasMoreTokens()){
        String s2=t.nextToken();
        int n2=s2.length();
        if(n2>max){
                max=n2;
                s3=s2;} }
        for(int i=0;i<s3.length();i++)
                for(int j=0;j<s4.length();j++)
                        if(s3.charAt(i)==s4.charAt(j))
                                n1++;
        return n1;
}
}
----------------------------------------------------------------------------------------------------||
```

20) for the given string display the middle 2 value, case satisfies only if string is of even length
ip - java
op - av

```
package Set2;
public class ClassSET20 {
public static void main(String[] args) {
	String s1="sumithra";
	System.out.println(printingSubstringOfMiddleChars(s1));
}
public static String printingSubstringOfMiddleChars(String s1) {
	String s2=s1.substring((s1.length()/2)-1, (s1.length()/2)+1);
	return s2;
}
}
----------------------------------------------------------------------------------------------------||
```

21) Given an array int a[]. Add the sum at even indexes.do the same with odd indexes.
	if both the sum is equal return 1 or return -1.

```
package Set2;
public class ClassSET21 {
public static void main(String[] args) {
	int a[]={9,8,5,3,2,6,4,7,5,1};
	System.out.println(oddEvenIndicesCount(a));
}
public static int oddEvenIndicesCount(int[] a) {
	int n1=0,n2=0,n3=0;
	for(int i=0;i<a.length;i++)
		if(i%2==0)
			n1+=a[i];
		else
			n2+=a[i];
	if(n1==n2)
		n3=1;
	else
		n3=-1;
	return n3;
}
}
---------------------------------------------------------------------------------------------------||
```

22)no of days in a month in specific year

```
package Set2;
import java.util.*;
public class ClassSET22 {
public static void main(String[] args){
        Calendar ca=new GregorianCalendar(2013,Calendar.FEBRUARY,03);
        System.out.println(noOfDaysInaMonth(ca));
}
public static int noOfDaysInaMonth(Calendar ca){
        int n=ca.getActualMaximum(Calendar.DAY_OF_MONTH);
        return n;
}
}
```

```
----------------------------------------------------------------------------------------------------||
----------------------------------------------------------------------------------------------------||
----------------------------------------------------------------------------------------------------||
----------------------------------------------------------------------------------------------------||
----------------------------------------------------------------------------------------------------||

---------------------------------------------SET 1---------------------------------------------------
```

1. A number is given as input. Find the odd digits in the number, add them and find if the sum is odd or not.if even return -1, if odd return 1
input:52315
logic:5+3+1+5=14(even)
output:-1
input:1112
logic:1+1+1=3(odd)
output:1

```
package Set1;

public class ClassSet1 {
        public static int SumOfOddsAndEvens(int n){
                int n1,n2=0,n3;
                while(n!=0)
                {
                        n1=n%10;
                        if((n1%2)!=0)
                                n2+=n1;
                        n/=10;
                }
                if(n2%2==0)
                        n3=-1;
                else
                        n3=1;

                return n3;
        }
```

```
    public static void main(String[] args) {
        int n=12346;
        System.out.println(SumOfOddsAndEvens(n));
    }
}
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
```

2.Find the day(Friday) of a date input given as MM-dd-yyyy (format)
input:12-27-2012
output:Thursday

```
package Set1;
import java.util.*;
import java.text.*;
public class ClassSet2 {
    public static String getDay(Date d1){
        String s1;
        SimpleDateFormat sdf=new SimpleDateFormat("EEEEE");
        s1=sdf.format(d1);
        return s1;
    }
    public static void main(String[] args) {
        Date d1=new Date(2012/12/27);
        System.out.println("day is:"+getDay(d1));
    }
}
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
```

3.A integer array is given as input. find the difference between each element. Return the index of the largest element which has the largest difference gap.
input: {2,3,4,2,3}
logic: 2-3=1,3-4=1,4-2=2,2-3=1
2 is the max diff between 4 and 2,return the index of 4(2)
output:2

```
package Set1;

public class ClassSet3 {
    public static int getDiffArray(int[] n1){
        int n2,n3=0,n4=0,i;
        for(i=0;i<n1.length-1;i++){
            n2=Math.abs(n1[i]-n1[i+1]);
            if(n2>n3){
                n3=n2;
                n4=i+1; }}
        return n4;
    }
    public static void main(String[] args) {
        int[] n1={2,9,4,7,6};
        System.out.println(getDiffArray(n1));
    }
}
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
----------------------------------------
```

4.Given two integer arrays. merge the common elements into a new array. find the sum of the elements
input1:{1,2,3,4}
input2:{3,4,5,6}
logic:{3,4}
output:7

```
package Set1;
import java.util.*;
public class ClassSet4 {
    public static int mergeArray(int a[],int b[]){
        List<Integer> l1=new ArrayList<Integer>();
        List<Integer> l2=new ArrayList<Integer>();
        int i,d=0;
        for(i=0;i<a.length;i++)
            l1.add(a[i]);
        for(i=0;i<b.length;i++)
            l2.add(b[i]);
        l1.retainAll(l2);
        for(i=0;i<l1.size();i++)
            d+=(Integer) l1.get(i);
        return d;
    }
    public static void main(String[] args) {
        int[] a={1,2,3,4};
        int[] b={3,4,5,6};
        System.out.println(mergeArray(a,b));
    }
}
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
----------------------------------------
```

5.Given two arrayslist. retrieve the odd position elements form first input and even position elemetns from second list. put it into the new arraylist at the same positions from where they are retrieved from.
(consider 0th position as even position, and two lists are of same size)
input1:{12,1,32,3}
input2:{0,12,2,23}
output:{0,1,2,3}

```
package Set1;

public class ClassSet5 {
	public static int[] retrievePosition(int[] a,int[] b){
		int[] c=new int[a.length];
		int i;
		for(i=0;i<a.length;i++){
			if(i%2==0)
				c[i]=b[i];
			if(i%2!=0)
				c[i]=a[i]; }
		return c;
	}
	public static void main(String[] args) {
		int[] a={12,1,32,3};
		int[] b={0,12,2,23};
		int[] c=retrievePosition(a,b);
		for(int d:c)
			System.out.println(d);
	}
}
```


----

6. Sum of fibonaaci series upto given input.
input:3
logic:1+1+2
output:4

```
package Set1;
import java.util.*;
public class ClassSet6 {
	public static int sumOfFibonacci(int n){
		int a=-1,b=1,c=0,d=0;
		for(int i=0;i<=n;i++){
			c=a+b;
			a=b; b=c;
			d+=c;
		}
		return d;
	}
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		int n=s.nextInt();
		System.out.println(sumOfFibonacci(n));
	}
}
```


----

7. Retrieve the odd numbers till given input number. add and subtract it consecutively and return the result.
Input:9
Output:1+3-5+7-9=-3

```
package Set1;
import java.util.*;
public class ClassSet7 {
    public static int consecutiveSumSubofOddNos(int n){
        List<Integer> l1=new ArrayList<Integer>();
        for(int i=1;i<=n;i++)
            if(i%2!=0)
                l1.add(i);
        int n1=l1.get(0);
        for(int i=1;i<l1.size();i++)
            if(i%2!=0)
                n1=n1+l1.get(i);
            else
                n1=n1-l1.get(i);
        return n1;
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        System.out.println(consecutiveSumSubofOddNos(n));
    }
}
```

----

8. Given date in dd-MM-yyyy format.return the month in full name format(January)
input:"23-01-2012"
output:January

```
package Set1;
import java.text.SimpleDateFormat;
import java.util.*;
public class ClassSet8 {
    public static String monthDiff(Date d1){
        SimpleDateFormat sdf=new SimpleDateFormat("MMMM");
        String s=(sdf.format(d1));
        return s;
    }
    public static void main(String[] args) {
        Date d1=new Date(23/01/2012);
        System.out.println(monthDiff(d1));
    }
}
```

----

9. Two dates are given as input in "yyyy-MM-dd" format. Find the number of months between the two dates
input1:"2012-12-01"
input2:"2012-01-03"
output:11

```
package Set1;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
public class ClassSet9 {
        public static int monthDiff(String s1,String s2) throws ParseException{
                SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
                Calendar c=Calendar.getInstance();
                Date d1=sdf.parse(s1);
                Date d2=sdf.parse(s2);
                c.setTime(d1);
                int m1=c.get(Calendar.MONTH);
                int y1=c.get(Calendar.YEAR);
                c.setTime(d2);
                int m2=c.get(Calendar.MONTH);
                int y2=c.get(Calendar.YEAR);
                int n=(y1-y2)*12+(m1-m2);
                return n;
        }
        public static void main(String[] args) throws ParseException {
                String s1=new String("2013-12-01");
                String s2=new String("2012-01-03");
                System.out.println(monthDiff(s1,s2));
        }
}
```

----

10. arraylist of string type which has name#mark1#mark2#mark3 format. retrieve the name of the student who has scored max marks(total of three)
input:{"arun#12#12#12","deepak#13#12#12"}
output:deepak

```
package Set1;
import java.util.*;
public class ClassSet10 {
        public static String retrieveMaxScoredStudent(String[] s1){
                Map<String, Integer> m1=new HashMap<String, Integer>();
                for(int i=0;i<s1.length;i++){
                    String s2=s1[i];
                    StringTokenizer t=new StringTokenizer(s2,"#");
                    String s3=t.nextToken();
                    int n1=Integer.parseInt(t.nextToken());
                    int n2=Integer.parseInt(t.nextToken());
                    int n3=Integer.parseInt(t.nextToken());
                    int n=n1+n2+n3;
                    m1.put(s3, n);
                    }
                //System.out.println(m1);
                int max=0;
                String m=new String();
                Iterator<String> i=m1.keySet().iterator();
                while(i.hasNext()){
                    String s4=i.next();
                    int j=m1.get(s4);
                    if(j>max){
                        max=j;
                        m=s4; }
                }
                return m;
        }
        public static void main(String[] args) {
                String[] s1={"arun#12#12#12","deepak#13#12#12","puppy#12#11#12"};
                System.out.println(retrieveMaxScoredStudent(s1));
        }
}
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------
```

11.Two inputs of a string array and a string is received. The array shld be sorted in descending order. The index of second input in a array shld be retrieved.
input1:{"ga","yb","awe"}
input2:awe
output:2

```
package Set1;
import java.util.*;
public class ClassSet11 {
    public static void main(String[] args) {
        String[] s1={"good","yMe","awe"};
        String s2="awe";
        System.out.println(stringRetrieval(s1,s2));
    }
    public static int stringRetrieval(String[] s1, String s2){
        ArrayList<String> l1=new ArrayList<String>();
        for(int i=0;i<s1.length;i++)
            l1.add(s1[i]);
        Collections.sort(l1,Collections.reverseOrder()) ;
        System.out.println(l1);
        int n=l1.indexOf(s2);
```

```
		return n;
	}
}
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

12.A time input is received as stirng. Find if the hour format is in 12 hour format. the suffix am/pm is case insensitive.
input:"09:36 am"
output:true

```
package Set1;
import java.text.*;
import java.util.*;
public class ClassSet12 {
	public static boolean hourFormat(String s) throws ParseException{
		boolean b=false;
		StringTokenizer st=new StringTokenizer(s," ");
		String s1=st.nextToken();
		String s2=st.nextToken();
		StringTokenizer st1=new StringTokenizer(s1,":");
		int n1=Integer.parseInt(st1.nextToken());
		int n2=Integer.parseInt(st1.nextToken());
		if((s2.equalsIgnoreCase("am"))|| (s2.equalsIgnoreCase("pm")))
			if((n1<=12)&&(n2<=59))
				b=true;
		return b;
	}
	public static void main(String[] args) throws ParseException {
		String s="19:36 am";
		boolean b=hourFormat(s);
		if(b==true)
			System.out.println("the time is in 12 hr format");
		else
			System.out.println("the time is in 24 hr format");
	}
}
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

13.Retrieve the palindorme-true number set from given number limit and return the sum
input1:90 input2:120
logic:99+101+111
output:311

```
package Set1;
import java.util.*;
public class ClassSet13 {
	public static int sumOfPalindromeNos(int n1,int n2){
		List<Integer> l1=new ArrayList<Integer>();
```

```
        for(int i=n1;i<=n2;i++){
            int r=0,n3=i;
            while(n3!=0){
                r=(r*10)+(n3%10);
                n3=n3/10;       }
            if(r==i)
                l1.add(i);              }
        System.out.println(l1);
        int s=0;
        for(int i=0;i<l1.size();i++)
            s+=l1.get(i);
        return s;
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the range:");
        int n1=s.nextInt();
        int n2=s.nextInt();
        System.out.println("sum of palindrome nos.within given range is:"+sumOfPalindromeNos(n1,n2));
    }
}
```


14.find the max length-word in a given string and return the max-length word. if two words are of same length return the first occuring word of max-length
input:"hello how are you aaaaa"
output:hello(length-5)

```
package Set1;
import java.util.*;
public class ClassSet14 {
    public static String lengthiestString(String s1){
        int max=0;
      String s2=null;
    StringTokenizer t=new StringTokenizer(s1," ");
    while(t.hasMoreTokens()){
        s1=t.nextToken();
        int n=s1.length();
        if(n>max){
            max=n;
            s2=s1;  }
        }
    return s2;
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the String:");
        String s1=s.nextLine();
        System.out.println("the lengthiest string is:"+lengthiestString(s1));
        }
}
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
```

15. Get a input string. reverse it and parse it with '-'.
input:computer
output:r-e-t-u-p-m-o-c

```
package Set1;
import java.util.Scanner;
public class ClassSet15 {
    public static String reversedAndParsedString(String s1){
        StringBuffer sb=new StringBuffer(s1);
        sb.reverse();
        StringBuffer sb1=new StringBuffer();
        for(int i=0;i<(2*s1.length())-1;i++)
            if(i%2!=0)
                sb1=sb.insert(i, '-');
        return sb1.toString();
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the String:");
        String s1=s.next();
        System.out.println("the formatted string is:"+reversedAndParsedString(s1));
    }
}
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
```

16. HashMap with regnum as key and mark as value. find the avg of marks whose key is odd.
input:{12:90,35:90,33:90,56:88}
output:avg of(90+90) =90

```
package Set1;
import java.util.*;
public class ClassSet16 {
    public static int averageOfMarks(Map<Integer,Integer> m1){
        int n=0,c=0;
        Iterator<Integer> i=m1.keySet().iterator();
        while(i.hasNext()){
            Integer b=i.next();
            if(b%2!=0){
                c++;
                n+=m1.get(b);} }
        return (n/c);
    }
    public static void main(String[] args) {
        Map<Integer,Integer> m1=new HashMap<Integer,Integer>();
```

```
		m1.put(367, 89);
		m1.put(368, 98);
		m1.put(369, 92);
		m1.put(366, 74);
		m1.put(365, 67);
		System.out.println(averageOfMarks(m1));
	}
}
```

----

17. A integer input is accepted. find the square of individual digits and find their sum.
input:125
output:1*1+2*2+5*5

```
package Set1;
import java.util.*;
public class ClassSet16 {
	public static int averageOfMarks(Map<Integer,Integer> m1){
		int n=0,c=0;
		Iterator<Integer> i=m1.keySet().iterator();
		while(i.hasNext()){
			Integer b=i.next();
			if(b%2!=0){
				c++;
				n+=m1.get(b);} }
		return (n/c);
	}
	public static void main(String[] args) {
		Map<Integer,Integer> m1=new HashMap<Integer,Integer>();
		m1.put(367, 89);
		m1.put(368, 98);
		m1.put(369, 92);
		m1.put(366, 74);
		m1.put(365, 67);
		System.out.println(averageOfMarks(m1));
	}
}
```

18.Two input strings are accepted. If the two strings are of same length concat them and return, if they are not of same length, reduce the longer string to size of smaller one, and concat them
input1:"hello"
input2:"hi"
output:"lohi"
input1:"aaa"
input2:"bbb"
output:"aaabbb"

```
package Set1;
import java.util.*;
public class ClassSet18 {
    public static String concatStrings(String s1,String s2){
        StringBuffer sb=new StringBuffer();
        if(s1.length()==s2.length())
            sb.append(s1).append(s2);
        else if(s1.length()>s2.length()){
            s1=s1.substring(s1.length()/2, s1.length());
            sb.append(s1).append(s2);
        }
        else if(s1.length()<s2.length()){
            s2=s2.substring(s2.length()/2, s2.length());
            sb.append(s1).append(s2);
        }
        return sb.toString();
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        String s1=s.next();
        String s2=s.next();
        System.out.println(concatStrings(s1,s2));
    }
}
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------
```

19.accept a string and find if it is of date format "dd/mm/yyyy".
input:01/13/2012
output:false

```
package Set1;
import java.util.StringTokenizer;
public class ClassSet19 {
    public static boolean formattingDate(String s){
        boolean b1=false;
        StringTokenizer t=new StringTokenizer(s,"/");
        int n1=Integer.parseInt(t.nextToken());
        int n2=Integer.parseInt(t.nextToken());
        String s2=t.nextToken();
        int n3=Integer.parseInt(s2);
        int n4=s2.length();
        if(n4==4)
        {
            if(n3%4==0)
            {
                if((n2==2)&&(n1<=29))
                    b1=true;
				else if((n2==4)||(n2==6)||(n2==9)||(n2==11)&&(n1<=30))
					b1=true;
				else if((n2==1)||(n2==3)||(n2==5)||(n2==7)||(n2==8)||(n2==10)||(n2==12)&&(n1<=31))
					b1=true;
			}
			else
			{
				if((n2==2)&&(n1<=28))
					b1=true;
				else if((n2==4)||(n2==6)||(n2==9)||(n2==11)&&(n1<=30))
					b1=true;
				else if((n2==1)||(n2==3)||(n2==5)||(n2==7)||(n2==8)||(n2==10)||(n2==12)&&(n1<=31))
					b1=true;
			}
		}
		return b1;
	}
	public static void main(String[] args) {
	String s="31/5/2012";
	boolean b=formattingDate(s);
	if(b==true)
		System.out.println("date is in dd/MM/yyyy format");
	else
		System.out.println("date is not in dd/MM/yyyy format");
	}
}
```

----

20. Get a integer array as input. Find the average of the elements which are in the position of prime index
input:{1,2,3,4,5,6,7,8,9,10}
logic:3+4+6+8(the indeces are prime numbers)
output:21

```
package Set1;
import java.util.*;
public class ClassSet20 {
	public static int sumOfPrimeIndices(int[] a,int n){
		int n1=0;
		for(int i=2;i<n;i++){
			int k=0;
			for(int j=2;j<i;j++)
				if(i%j==0)
					k++;
			if(k==0)
				n1+=a[i];	}
		return n1;
	}
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		System.out.println("enter the array limit:");
```

```
        int n=s.nextInt();
        System.out.println("enter the array elements:");
        int[] a=new int[n];
        for(int i=0;i<n;i++)
            a[i]=s.nextInt();
        System.out.println(sumOfPrimeIndices(a,n));
    }
}
```


21.Find the extension of a given string file.
input: "hello.jpeg"
output: "jpeg"

```
package Set1;
import java.util.*;
public class ClassSet21 {
    public static String extensionString(String s1){
        StringTokenizer t=new StringTokenizer(s1,".");
        t.nextToken();
        String s2=t.nextToken();
        return s2;
    }
    public static void main(String[] args) {
        String s1="hello.jpeg";
        System.out.println(extensionString(s1));
    }
}
```


22.Get two integer arrays as input. Find if there are common elements in the arrays. find the number of common elements.
input1: {1,2,3,4}
input2: {3,4,5,6}
output: 4(3,4,3,4)

```
package Set1;
public class ClassSet22 {
    public static int commonElements(int[] a,int[] b){
        int c=0;
        for(int i=0;i<a.length;i++)
            for(int j=0;j<b.length;j++)
                if(a[i]==b[j])
                    c++;
        return (2*c);
    }
```

```
        public static void main(String[] args){
                int a[]={1,2,3,4,5};
                int b[]={3,4,5,6,7};
                System.out.println(commonElements(a,b));
        }
}
---------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

23.Find if a given pattern appears in both the input strings at same postions.
input1: "hh--ww--"
input2: "rt--er--"
output: true(false otherwise)

```
package Set1;
public class ClassSet23 {
        public static boolean stringPattern(String s1,String s2){
                String st1=s1.length()<s2.length()?s1:s2;
            String st2=s1.length()>s2.length()?s1:s2;
        boolean b=true;
        String s=st2.substring(st1.length());
            if(s.contains("-"))
                 b=false;
            else{
        loop:
            for(int i=0;i<st1.length();i++)
                if((st1.charAt(i)=='-') || (st2.charAt(i)=='-'))
                        if(st1.charAt(i)!=st2.charAt(i)){
                                 b=false;
                                 break loop;  }
                }
            return b;
        }
        public static void main(String[] args) {
                String s1="he--ll--";
                String s2="we--ll--";
                boolean b=stringPattern(s1,s2);
                if(b==true)
                        System.out.println("same pattern");
                else
                        System.out.println("different pattern");
        }
}
---------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

24. Input a int array. Square the elements in even position and cube the elements in odd position and add them as result.
input: {1,2,3,4}
output: 1^3+2^2+3^3+4^2

```
package Set1;
public class ClassSet24 {
public static int squaringAndCubingOfElements(int[] a){
        int n1=0,n2=0;
        for(int i=0;i<a.length;i++)
                if(i%2==0)
                        n1+=(a[i]*a[i]*a[i]);
                else
                        n2+=(a[i]*a[i]);
        return (n1+n2);
}
public static void main(String[] args) {
                int a[]={1,2,3,4,5,6};
                System.out.println(squaringAndCubingOfElements(a));
        }
}
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

25. Check whether a given string is palindrome also check whether it has atleast 2 different vowels
input: "madam"
output: false(no 2 diff vowels)

```
package Set1;
public class ClassSet25 {
        public static boolean palindromeAndVowelCheck(String s){
                boolean b=true;
                int c=0;
                String s1="aeiou";
                StringBuffer sb=new StringBuffer(s);
                String s2=sb.reverse().toString();
                if(!s2.equals(s))
                        b=false;
                else{
                loop:
                for(int i=0;i<s1.length();i++)
                        for(int j=0;j<s.length();j++)
                                if(s1.charAt(i)==s.charAt(j)){
                                        c++;
                                        continue loop; }
                if(c<2)
                        b=false;
                }
                return b;
        }
public static void main(String[] args) {
                String s="deed";
        System.out.println(palindromeAndVowelCheck(s));
    }
}
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
```

26. Find no of characters in a given string which are not repeated.
input: "hello"
output: 3

```
package Set1;
public class ClassSet26 {
public static int noOfnonRepeatedCharacters(String s1){
        StringBuffer sb=new StringBuffer(s1);
        for(int i=0;i<sb.length();i++){
            int n=0;
            for(int j=i+1;j<sb.length();j++)
                if(sb.charAt(i)==sb.charAt(j)){
                    sb.deleteCharAt(j);
                    n++; }
                if(n>0){
                    sb.deleteCharAt(i);
                    i--;}          }
        return sb.length();
    }
public static void main(String[] args) {
        String s1="preethi";
        System.out.println(noOfnonRepeatedCharacters(s1));
    }
}
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
----------------------------------------------
```

27. Get a input string. Find if it is a negative number, if true return the absolute value, in other cases return -1.
input: "-123"
output: 123
input: "@123"
output: -1

```
package Set1;
import java.util.*;
public class ClassSet27 {
    public static int negativeString(String s1){
        int n1=0;
        if(s1.startsWith("-")){
        int n=Integer.parseInt(s1);
        if(n<0)
            n1=Math.abs(n);}
```

```
		else
			n1=-1;
		return n1;
	}
public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		String s1=s.next();
		System.out.println(negativeString(s1));
	}
}
```

28.An arraylist of Strings is given as input. The count of the String elements that are not of size input2 string is to be returned.
input1: {"aaa","bb","cccc","d"}
input2: "bb"
output: 3("bb"-length:2)

```
package Set1;
import java.util.*;
public class ClassSet28 {
	public static int StringsNotOfGivenLength(List<String> l1,String s1){
		int n1=s1.length();
		int c=0;
		for(int i=0;i<l1.size();i++){
			int n2=l1.get(i).length();
			if(n1!=n2)
				c++;
		}
		return c;
	}
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		System.out.println("enter the no.of elements:");
		int n=s.nextInt();
		List<String> l1=new ArrayList<String>();
		for(int i=0;i<n;i++)
			l1.add(s.next());
		System.out.println("enter the input string:");
		String s1=s.next();
		System.out.println(StringsNotOfGivenLength(l1,s1));
	}
}
```

29.An array of integers is given.The given element is removed new array is returned.
input1:{10,10,20,30,76}

output: {20,20,76}(10 is asked to remove)

```
package Set1;
import java.util.*;
public class ClassSet29 {
	public static int[] removalOfGivenElementFromArray(int[] a,int c){
		List<Integer> l1=new ArrayList<Integer>();
		for(int i=0;i<a.length;i++)
			if(a[i]!=c)
				l1.add(a[i]);
		int b[]=new int[l1.size()];
		for(int i=0;i<b.length;i++)
			b[i]=l1.get(i);
		return b;
	}
	public static void main(String[] args) {
	int a[]={10,10,20,30,40,50};
	int c=10;
	int[] b=removalOfGivenElementFromArray(a,c);
	for(int d:b)
		System.out.println(d);
	}
}
```

----

30. Find the number of days between two input dates.

```
package Set1;
import java.text.*;
import java.util.*;
public class ClassSet30 {
public static int dateDifference(String s1,String s2) throws ParseException{
	SimpleDateFormat sd=new SimpleDateFormat("dd/MM/yyyy");
	Date d=sd.parse(s1);
	Calendar c=Calendar.getInstance();
	c.setTime(d);
	long d1=c.getTimeInMillis();
	d=sd.parse(s2);
	c.setTime(d);
	long d2=c.getTimeInMillis();
	int n=Math.abs((int) ((d1-d2)/(1000*3600*24)));
	return n;
	}
public static void main(String[] args) throws ParseException {
	String s1="27/12/2009";
	String s2="15/09/2012";
	System.out.println(dateDifference(s1,s2));
	}
}
```

31.Enter yout name and return a string "print the title first and then comma and then first letter of initial name.

```
package Set1;
import java.util.*;
public class ClassSet31 {
        public static String retrieveName(String s1){
                StringTokenizer t=new StringTokenizer(s1," ");
                String s2=t.nextToken();
                String s3=t.nextToken();
                StringBuffer sb=new StringBuffer(s2);
                sb.append(',').append(s3.charAt(0));
                return sb.toString();
        }
        public static void main(String[] args) {
                String s1="Bhavane Raghupathi";
                System.out.println(retrieveName(s1));
        }
}
```

32.Write a Program that accepts a string and removes the duplicate characters.

```
package Set1;
public class ClassSet32 {
        public static String removalOfDuplicateCharacters(String s1){
                StringBuffer sb=new StringBuffer(s1);
                for(int i=0;i<s1.length();i++)
                        for(int j=i+1;j<s1.length();j++)
                                if(s1.charAt(i)==s1.charAt(j))
                                        sb.deleteCharAt(j);
                return sb.toString();
        }
        public static void main(String[] args) {
                String s1="bhavane";
                System.out.println(removalOfDuplicateCharacters(s1));
        }
}
```

33.validate a password
i) there should be atleast one digit
ii) there should be atleast one of '#','@' or '$' .
iii)the password should be of 6 to 20 characters
iv) there should be more uppercase letter than lower case.
v) should start with an upper case and end with lower case

```
package Set1;
import java.util.*;
public class ClassSet33 {
public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        String st=s.next();
        boolean b=validatingPassword(st);
        if(b==true)
            System.out.println("valid password");
        else
            System.out.println("Invalid Password");
    }

public static boolean validatingPassword(String st)
    {
    boolean b1=false,b2=false;
    if(Character.isUpperCase(st.charAt(0)))
        if(Character.isLowerCase(st.charAt(st.length()-1)))
            if(st.length()>=6 && st.length()<=20)
                for (int i = 0; i < st.length();i++)
            {   char c = st.charAt(i);
              if(Character.isDigit(c)){
               b1=true; break; }        }
    int x=0,y=0;
    for(int i = 0; i < st.length(); i++)
       if(Character.isUpperCase(st.charAt(i)))
        x++;
       else if(Character.isLowerCase(st.charAt(i)))
          y++;
    if(b1==true)
        if(x>y)
            for (int i = 0; i < st.length();i++)
        {   char c = st.charAt(i);
          if(c=='#' || c=='@' || c=='$'){
           b2=true; break; }        }
    return b2;
}

}
```


----

34. find the average of the maximum and minimum number in an integer array

```
package Set1;
import java.util.*;
public class ClassSet34 {
    public static void main(String[] args) {
        int[] a={10,54,23,14,32,45};
        System.out.println(avgOfMaxandMinNoinArray(a));
    }
    public static int avgOfMaxandMinNoinArray(int[] a) {
```

```
                Arrays.sort(a);
                int b=a[0];
                int c=a[a.length-1];
                return (b+c)/2;
        }

}
```


35.validate the ip address in the form a.b.c.d
  where a,b,c,d must be between 0and 255
  if validated return 1 else return 2

```
package Set1;
import java.util.*;
public class ClassSet35 {
        public static void main(String[] args) {
                String ipAddress="010.230.110.160";
            boolean b=validateIpAddress(ipAddress);
            if(b==true)
                System.out.println("valid ipAddress");
            else
                System.out.println("not a valid ipAddress");
        }
        public static boolean validateIpAddress(String ipAddress) {
                boolean b1=false;
                StringTokenizer t=new StringTokenizer(ipAddress,".");
                int a=Integer.parseInt(t.nextToken());
                int b=Integer.parseInt(t.nextToken());
                int c=Integer.parseInt(t.nextToken());
                int d=Integer.parseInt(t.nextToken());
                if((a>=0 && a<=255)&&(b>=0 && b<=255)&&(c>=0 && c<=255)&&(d>=0 && d<=255))
                        b1=true;
                return b1;
        }
}
```


----

36. find the three consecutive characters in a string. if the string consists any three
  consecutive characters return 1 else return -1
  like AAAxyzaaa will return true.

```
package Set1;
public class ClassSet36 {
```

```
public static void main(String[] args) {
    String s1="aaaaxyzAAA";
    boolean b=consecutiveCharactersCheck(s1);
    if(b==true)
        System.out.println("presence of consecutive characters");
    else
        System.out.println("absence of consecutive characters");
    }

public static boolean consecutiveCharactersCheck(String s1) {
    boolean b=false;
    int c=0;
    for(int i=0;i<s1.length()-1;i++)
        if((s1.charAt(i+1)-s1.charAt(i))==1)
                c++;
    if(c>=2)
        b=true;
    return b;
}
}
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
```

37.String encrption. replace the odd-index character with next character(if it is 'z' replace with 'a'..if 'a' with 'b' as such), leave the even index as such. return the encrypted string.

```
package Set1;
public class ClassSet37 {
public static void main(String[] args) {
    String s="preethi";
    System.out.println(oddEncryptionOfString(s));
    }
public static String oddEncryptionOfString(String s) {
    StringBuffer sb=new StringBuffer();
    for(int i=0;i<s.length();i++){
        char c=s.charAt(i);
        if(c%2!=0){
            c=(char)(c+1);
            sb.append(c);  }
        else
            sb.append(c);   }
    return sb.toString();
}
}
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
```

38. Retrieve the max from array which is in a odd-index

```
package Set1;
import java.util.*;
```

```
public class ClassSet38 {
public static void main(String[] args) {
        int[] a={10,23,45,54,32,14,55,65,56};
        System.out.println(retrieveMaxFromOddIndex(a));
        }
public static int retrieveMaxFromOddIndex(int[] a) {
        List<Integer> l=new ArrayList<Integer>();
        for(int i=0;i<a.length;i++)
                if(i%2!=0)
                        l.add(a[i]);
        int b=0;
        for(int i=0;i<l.size();i++){
                int c=(Integer) l.get(i);
                if(c>b)
                        b=c; }
        return b;
}
}
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
```

39.GIVEN A STRING 555-666-1234
DISPLAY AS 55-56-661-234?

```
package Set1;
import java.util.StringTokenizer;
public class ClassSet39 {
public static void main(String[] args) {
        String s="555-666-1234";
        System.out.println(display(s));
        }
public static String display(String s) {
        StringTokenizer t=new StringTokenizer(s,"-");
        String s1=t.nextToken();
        String s2=t.nextToken();
        String s3=t.nextToken();

        StringBuffer sb=new StringBuffer();
        sb.append(s1.substring(0, s1.length()-1)).append('-');
        sb.append(s1.charAt(s1.length()-1)).append(s2.charAt(0)).append('-');
        sb.append(s2.substring(1, s2.length())).append(s3.charAt(0)).append('-');
        sb.append(s3.substring(1, s3.length()));

        return sb.toString();
}
}
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
```

40.input1="Rajasthan";
input2=2;

input3=5;
output=hts;

```
package Set1;
public class ClassSet40 {
	public static void main(String[] args) {
		String input1="Rajasthan";
		int input2=2, input3=5;
		System.out.println(retrieveString(input1,input2,input3));
	}
	public static String retrieveString(String input1, int input2, int input3) {
		StringBuffer sb=new StringBuffer(input1);
		sb.reverse();
		String output=sb.substring(input2, input3);
		return output;
	}
}
--------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------
```

41.input1={1,2,3}
input2={3,4,5}
input3;+(union)
output:inp1+inp2
input1:{1,2,3,4,5}
input 2:{2,3,4,5}
input3=*(intersection)
output:inp1*inp2
INPUT1:{1,2,3,4,5}
INPUT2:{3,6,7,8,9}
INPUT3:-(MINUS)
output:inp1-inp2

```
package Set1;
import java.util.*;
public class ClassSet41 {
	public static void main(String[] args) {
		int[] a={1,2,3,4,5};
		int[] b={0,2,4,6,8};
		Scanner s=new Scanner(System.in);
		int c=s.nextInt();
		int d[]=arrayFunctions(a,b,c);
		for(int e:d)
			System.out.println(e);
	}
```

```
    public static int[] arrayFunctions(int[] a, int[] b, int c) {
        List<Integer> l1=new ArrayList<Integer>();
        List<Integer> l2=new ArrayList<Integer>();
        List<Integer> l3=new ArrayList<Integer>();
        for(int i=0;i<a.length;i++)
            l1.add(a[i]);
        for(int i=0;i<b.length;i++)
            l2.add(b[i]);

        switch(c){
        case 1:
            l1.removeAll(l2);
            l1.addAll(l2);
            Collections.sort(l1);
            l3=l1;
            break;
        case 2:
            l1.retainAll(l2);
            Collections.sort(l1);
            l3=l1;
            break;
        case 3:
            if(l1.size()==l2.size())
                for(int i=0;i<l1.size();i++)
                    l3.add(Math.abs(l2.get(i)-l1.get(i)));
            break;
        }
        int[] d=new int[l3.size()];
        for(int i=0;i<d.length;i++)
            d[i]=l3.get(i);
        return d;
    }
}
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
```

42. input1=commitment;
  output=cmmitmnt;
c be the first index position
remove even vowels from the string

```
package Set1;
public class ClassSet42 {
    public static void main(String[] args) {
        String s1="cOmmitment";
        System.out.println(removeEvenElements(s1));
    }
    public static String removeEvenElements(String s1) {
        StringBuffer sb1=new StringBuffer();
        for(int i=0;i<s1.length();i++)
```

```
            if((i%2)==0)
                sb1.append(s1.charAt(i));
            else if((i%2)!=0)
                if(s1.charAt(i)!='a' && s1.charAt(i)!='e' && s1.charAt(i)!='i' && s1.charAt(i)!='o' &&
s1.charAt(i)!='u')
                    if(s1.charAt(i)!='A' && s1.charAt(i)!='E' && s1.charAt(i)!='I' && s1.charAt(i)!='O' &&
s1.charAt(i)!='U')
                     sb1.append(s1.charAt(i));
        return sb1.toString();
    }
}
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

43. Retrieve the odd-position digits from input integer array. Multiply them with their index and return their sum.

```
package Set1;
public class ClassSet43 {
    public static void main(String[] args) {
        int[] a={5,1,6,2,9,4,3,7,8};
        System.out.println("Sum Of Odd Position Elements Multiplied With Their Index Is:");
        System.out.println(retrievalOfOddPositionElements(a));
    }
    public static int retrievalOfOddPositionElements(int[] a) {
        int s=0;
        for(int i=0;i<a.length;i++)
            if(i%2!=0)
                s+=a[i]*i;
        return s;
    }
}
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

44. Return the number of days in a month, where month and year are given as input.

```
package Set1;
import java.util.*;
public class ClassSet44 {
    public static void main(String[] args){
        String s1="02/2011";
        System.out.println(noOfDaysInAmonth(s1));
    }
    public static int noOfDaysInAmonth(String s1){
        int n=0;
        StringTokenizer r=new StringTokenizer(s1,"/");
        int n1=Integer.parseInt(r.nextToken());
        int n2=Integer.parseInt(r.nextToken());
        if(n1==1 || n1==3 || n1==5 ||n1==7 || n1==8 || n1==10 || n1==12)
            n=31;
        else if(n1==4 || n1==6 || n1==9 || n1==11)
            n=30;
        else if(n1==2){
            if(n2%4==0)
```

```
                                n=29;
                        else
                                n=28; }
                return n;
        }
}
```

```
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------
```

45. Retrieve the non-prime numbers within the range of a given input. Add-up the non-prime numbers and return the result.

```
package Set1;
import java.util.*;
public class ClassSet45 {
public static void main(String[] args) {
        int i=20,j=40;
        System.out.println("sum of non-prime nos. is:"+additionOfnonPrimeNos(i,j));
        }
public static int additionOfnonPrimeNos(int i, int j){
        int k=0;
        List<Integer> l1=new ArrayList<Integer>();
        List<Integer> l2=new ArrayList<Integer>();
        for(int a=i;a<=j;a++){
                int c=0;
                for(int b=2;b<a;b++)
                        if(a%b==0)
                                c++;
                if(c==0)
                        l1.add(a); }
        for(int e=i;e<=j;e++)
                l2.add(e);
        l2.removeAll(l1);
        for(int d=0;d<l2.size();d++)
                k+=l2.get(d);
        return k;
}
}
```

```
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------
```

46. Retrieve the elements in a array, which is an input, which are greater than a specific input number. Add them and find the reverse of the sum.

```
package Set1;
public class ClassSet46 {
public static void main(String[] args) {
        int[] a={23,35,11,66,14,32,65,56,55,99};
        int b=37;
        System.out.println(additionOfRetrievedElements(a,b));
        }
public static int additionOfRetrievedElements(int[] a, int b) {
        int n=0;
        for(int i=0;i<a.length;i++)
        if(a[i]>b)
            n+=a[i];
    return n;
}
}
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------
```

47.Input a hashmap. Count the keys which are not divisible by 4 and return.

```
package Set1;
import java.util.*;
public class ClassSet47 {
    public static void main(String[] args) {
        Map<Integer, String> m1=new HashMap<Integer, String>();
        m1.put(24, "preethi");
        m1.put(32, "bhanu");
        m1.put(25, "menu");
        m1.put(31, "priya");
        System.out.println(fetchingKeysDivisibleByFour(m1));
    }
    public static int fetchingKeysDivisibleByFour(Map<Integer, String> m1) {
        int k=0;
        Iterator<Integer> i=m1.keySet().iterator();
        loop:
        while(i.hasNext()){
            int j=i.next();
            if(j%4!=0)
                k++;
            continue loop; }
        return k;
    }
}
```