1. Strings – Unique & Existing Characters

Solution:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = newScanner(System.in);
String s1 = sc.nextLine();
String s2 = sc.nextLine();
System.out.println(User.replacePlus(s1, s2));
}
}

publicclass User {
publicstatic String replacePlus(String s1, String s2) {
String ss1 = s1.toLowerCase();
String ss2 = s2.toLowerCase();
StringBuffer sb = newStringBuffer();
for (int i = 0; i < s1.length(); i++) {
char c = ss1.charAt(i);
if (ss2.indexOf(c) == -1)
sb.append('+');
else
sb.append(s1.charAt(i));
}
return sb.toString();
}
}
```

2. Unique Even Sum

Solutions:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = newScanner(System.in);
int n = sc.nextInt();
int[] a = newint[20];
for (int i = 0; i < n; i++)
a[i] = sc.nextInt();
int res = User.addUniqueEven(a);
if (res == -1)
System.out.println("no even numbers");
else
System.out.println(res);
```

```
}
}

publicclass User {
publicstaticint addUniqueEven(int a[]) {
int i = 0, j = 0, count = 0, sum = 0;
int n = a.length;
for (i = 0; i < n; i++) {
count = 0;
for (j = i + 1; j < n; j++) {
if (a[i] == a[j])
count++;
}
if (count == 0) {
if (a[i] % 2 == 0)
sum = sum + a[i];
}
}
if (sum == 0)
return -1;
else
return sum;
}
}
```

```
package uniqueevensum;

import java.util.HashSet;
import java.util.Set;

public class UserMainCode {

	public static int addUniqueEven(int[] a) {
		int sum = -1;
		Set<Integer> s = new HashSet<Integer>();
		for (int i = 0; i < a.length; i++) {
			s.add(a[i]);
		}
		for (int x : s) {
			if (x % 2 == 0) {
				sum = sum + x;
			}
		}
		return sum;
```

```
        }

}
```

3. String Occurences

Solution:

```
import java.util.Scanner;

publicclass Main {

publicstaticvoid main(String[] args) {
Scanner s = newScanner(System.in);
String s1 = s.nextLine();
String s2 = s.nextLine();
System.out.println(User.countNoOfWords(s1, s2));
}

}


import java.util.StringTokenizer;

publicclass User {
publicstaticint countNoOfWords(String s1, String s2) {
String[] a = new String[s1.length()];
String[] b = new String[s2.length()];
int i = 0, j = 0, count = 0;
StringTokenizer st1 = newStringTokenizer(s1, " ");
StringTokenizer st2 = newStringTokenizer(s2, " ");
while (st1.hasMoreTokens()) {
a[i] = st1.nextToken();
i++;
}
while (st2.hasMoreTokens()) {
b[j] = st2.nextToken();
j++;
}
for (int k = 0; k < i; k++) {
if (b[1].equals(a[k])) {
count++;
}
}
return count;
}
}
```

```
public class UserMain {

	public static int stringOccurence(String s1, String s2) {
		String[] a1 = s1.split(" ");
		String[] a2 = s2.split(" ");
		int count = 0;
		for (int i = 0; i < a1.length; i++) {
			String a = a1[i];
			for (int j = 0; j < a2.length; j++) {
				String b = a2[j];
				if (a.equals(b)) {
					count++;
				}
			}
		}
		return count;
	}
}
```

4. ArrayList Manipulation

Solution :

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

publicclass Main {

publicstaticvoid main(String[] args) {
Scanner s = newScanner(System.in);
int n = s.nextInt();
ArrayList<Integer> al1 = new ArrayList<Integer>();
ArrayList<Integer> al2 = new ArrayList<Integer>();
ArrayList<Integer> a = new ArrayList<Integer>();
for (int i = 0; i < n; i++)
al1.add(s.nextInt());
for (int i = 0; i < n; i++)
al2.add(s.nextInt());
List a = User.generateOddEvenList(al1, al2);
for (inti = 0; i< a.size(); i++)
System.out.println(a.get(i));
}
}
```

```
import java.util.ArrayList;

publicclass User {
publicstatic ArrayList<Integer> generateOddEvenList(ArrayList<Integer> a1,
ArrayList<Integer> a2) {
ArrayList<Integer> a = new ArrayList<Integer>();
int i = 0;
for (i = 0; i < a1.size(); i++) {
if (i % 2 == 0)
a.add(a2.get(i));
else
a.add(a1.get(i));
}
return a;

}
}
```

5. Mastering Hashmap

Solution:

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

publicclass Main {

publicstaticvoid main(String[] args) {
Scanner s = newScanner(System.in);
int n = s.nextInt();
HashMap<Integer, Integer> hm1 = new HashMap<Integer, Integer>();
for (int i = 0; i < n; i++)
hm1.put(s.nextInt(), s.nextInt());
System.out.println(User.getAverageOfOdd(hm1));
}
}


import java.util.HashMap;
import java.util.Iterator;

publicclass User {
publicstaticint getAverageOfOdd(HashMap<Integer, Integer> hm1) {
int sum = 0, count = 0;
Iterator<Integer> itr = hm1.keySet().iterator();
while (itr.hasNext()) {
int key = itr.next();
if (key % 2 != 0) {
count++;
int val = hm1.get(key);
```

```
sum = sum + val;
}
}
int avg = sum / count;
return avg;
}
}

public class UserMain {

	public static int getAverageOfOdd(HashMap<Integer, Integer> hm) {
		int sum = 0;
		int count = 0;
		int avg = 0;
		Iterator<Map.Entry<Integer, Integer>> i = hm.entrySet().iterator();
		while (i.hasNext()) {
			Map.Entry<Integer, Integer> e = i.next();
			if (e.getKey() % 2 != 0) {
				sum = sum + e.getValue();
				count++;
			}
			avg = Math.abs(sum / count);
		}
		return avg;
	}

}
```

6. Anagram

Solutions:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner s = newScanner(System.in);
String s1 = s.nextLine();
String s2 = s.nextLine();
int result = User.getAnagrams(s1, s2);
if (result == 1)
System.out.println("Anagrams");
else
System.out.println("Not Anagrams");
}
}
```

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

publicclass User {
publicstaticint getAnagrams(String s1, String s2) {

String str1 = s1.toLowerCase();
String str2 = s2.toLowerCase();
ArrayList<Character> al1 = new ArrayList<Character>();
ArrayList<Character> al2 = new ArrayList<Character>();
ArrayList<Character> al3 = new ArrayList<Character>();
int res = 0;
for (int i = 0; i < s1.length(); i++)
al1.add(str1.charAt(i));
for (int i = 0; i < s2.length(); i++)
al2.add(str2.charAt(i));
al3.add(' ');
al1.removeAll(al3);
al2.removeAll(al3);
Collections.sort(al1);
Collections.sort(al2);
if (al1.equals(al2))
res = 1;
else
res = -1;
return res;
}
}

public class UserMain {

	public static int anagram(String s1, String s2) {
		String a1 = s1.toLowerCase();
		String a2 = s2.toLowerCase();
		String b1[] = a1.split("");
		String b2[] = a2.split("");
		for (int i = 0; i < b1.length; i++) {
			Arrays.sort(b1);
		}
		for (int i = 0; i < b1.length; i++) {
			Arrays.sort(b2);
		}
		if (Arrays.equals(b1, b2)) {
			return 1;
		} else
			return -1;
	}
```

```
}
```

7. Retirement

Solution:

```
import java.text.ParseException;
importjava.util.LinkedHashMap;
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) throws ParseException {
Scanner s = newScanner(System.in);
int n = s.nextInt();
LinkedHashMap<String, String> hm = newLinkedHashMap<String, String>();
for (int i = 0; i < n; i++)
hm.put(s.next(), s.next());
System.out.println(User.retirementEmployeeList(hm));
}
}

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.LinkedHashMap;

publicclass User {
publicstaticArrayList<String> retirementEmployeeList(
LinkedHashMap<String, String> hm) throws ParseException {
ArrayList<String> al = new ArrayList<String>();
SimpleDateFormat sdf = newSimpleDateFormat("dd/MM/yyyy");
String s = "01/01/2014";
Date d2 = sdf.parse(s);
Date d1 = newDate();
Iterator<String> itr = hm.keySet().iterator();
while (itr.hasNext()) {
String key = itr.next();
String val = hm.get(key);
d1 = sdf.parse(val);
Calendar c = Calendar.getInstance();
c.setTime(d1);
int y1 = c.get(Calendar.YEAR);
int m1 = c.get(Calendar.MONTH);
int day1 = c.get(Calendar.DAY_OF_MONTH);
```

```
c.setTime(d2);
int y2 = c.get(Calendar.YEAR);
int m2 = c.get(Calendar.MONTH);
int day2 = c.get(Calendar.DAY_OF_MONTH);
int y = Math.abs(y1 - y2);
if (m1 == m2) {
if (day1 > day2)
y--;
} elseif (m1 > m2)
y--;
if (y >= 60)
al.add(key);
}
return al;
}
}
```

8. Kaprekar Number

Solution:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = newScanner(System.in);
int n = sc.nextInt();
int i = User.getKaprekarNumber(n);
if (i == 1)
System.out.println("Kaprekar Number");
else
System.out.println("Not Kaprekar Number");
}
}


publicclass User {
publicstaticint getKaprekarNumber(inttemp) {
intn = temp;
intsq = n * n;
intsqr=sq;
intres = 0;
intcount = 0;
while (sq != 0) {
count++;
sq= sq / 10;
}
String a = Integer.toString(sqr);
```

```
String n1 = a.substring(count/2);
String n2 = a.substring(0,count/2);
inti = Integer.parseInt(n1);
intj = Integer.parseInt(n2);
if ((i + j) == temp)
res = 1;
else
res = -1;
returnres;
}
}

public class UserMain {
        public static int kapnum(int n) {
                String a = (n * n) + "";
                int x = Integer.parseInt(a.substring(0, a.length() / 2));
                int y = Integer.parseInt(a.substring(a.length() / 2));
                int z = x + y;
                if (n == z) {
                        return 1;
                } else
                        return -1;
        }
}
```

9. Vowels

Solution:

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

publicclass Main {
publicstaticvoid main(String[] args) throws ParseException {
Scanner sc = newScanner(System.in);
String s = sc.nextLine();
System.out.println(User.storeMaxVowelWord(s));
}
}


import java.util.StringTokenizer;

publicclass User {
publicstatic String storeMaxVowelWord(String s) {
StringTokenizer st = new StringTokenizer(s, " ");
int count = 0, max = 0;
String s2 = null;
```

```
while (st.hasMoreTokens()) {
String s1 = st.nextToken();
count = 0;
for (int i = 0; i <s1.length(); i++) {
if (s1.charAt(i) == 'a' || s1.charAt(i) == 'e'
|| s1.charAt(i) == 'i' || s1.charAt(i) == 'o'
|| s1.charAt(i) == 'u' || s1.charAt(i) == 'A'
|| s1.charAt(i) == 'E' || s1.charAt(i) == 'I'
|| s1.charAt(i) == 'O' || s1.charAt(i) == 'U')
count++;
}
if (count > max) {
max = count;
s2 = s1;
}
}
return s2;
1
}
}
```

```
public class UserMain {
	public static String maxVowels(String s) {
		String s1[] = s.split(" ");
		int max = 0;
		int count = 0;
		String res = null;
		for (int i = 0; i < s1.length; i++) {
			count = 0;
			if ((s1[i].charAt(i) + "").matches("[aeiouAEIOU]{1}")) {
				count++;
			}
			if (count > max) {
				max = count;
				res = s1[i];
			}}
		return res;
	}
}
```

10. ArrayList and Set Operations

Solution:

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

public class Main {
```

```
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
ArrayList<Integer> al1 = new ArrayList<Integer>();
ArrayList<Integer> al2 = new ArrayList<Integer>();
ArrayList<Integer> res = new ArrayList<Integer>();
for (int i = 0; i < n; i++)
al1.add(sc.nextInt());
for (int i = 0; i < n; i++)
al2.add(sc.nextInt());
char c = sc.next().charAt(0);
res = User.performSetOperations(al1, al2, c);
for (int i = 0; i < res.size(); i++)
System.out.println(res.get(i));
}
}
```

```
import java.util.ArrayList;
import java.util.LinkedHashSet;

public class User {
public static ArrayList<Integer> performSetOperations(
ArrayList<Integer> al1, ArrayList<Integer> al2, char c) {

LinkedHashSet<Integer> h = new LinkedHashSet<Integer>();
ArrayList<Integer> al3 = new ArrayList<Integer>();
switch (c) {
case '+':
al1.addAll(al2);
h.addAll(al1);
al3.addAll(h);
break;
case '*':
for (int i = 0; i < al1.size(); i++) {
for (int j = 0; j < al2.size(); j++) {
if (al1.get(i) == al2.get(j)) {
al3.add(al1.get(i));
}
}
}
break;
case '-':
for (int i = 0; i < al1.size(); i++) {
for (int j = 0; j < al2.size(); j++) {
if (al1.get(i) == al2.get(j)) {
```

```
al1.remove(i);
}
}
}
al3.addAll(al1);
break;

}

return al3;
}
}
```

```
public class UserMain {

	public static List<Integer> setOperation(int[] a, int[] b, char c) {
		List<Integer> l = new ArrayList<>();
		// Set<Integer> h=new LinkedHashSet<>();
		Set<Integer> l1 = new LinkedHashSet<>();
		for (int i = 0; i < a.length; i++) {
			l1.add(a[i]);
		}
		Set<Integer> l2 = new LinkedHashSet<>();
		for (int i = 0; i < b.length; i++) {
			l2.add(b[i]);
		}
		if (c == '+') {
			l1.addAll(l2);
			l.addAll(l1);
		} else if (c == '*') {
			l1.retainAll(l2);
			l.addAll(l1);
		} else if (c == '-') {
			l1.removeAll(l2);
			l.addAll(l1);
		}
		return l;
	}

}
```

11. max Scorer

Solution:

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

publicclass Main {
publicstaticvoid main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
intn=sc.nextInt();
ArrayList<String>a=new ArrayList<String>();
for(inti=0;i<n;i++)
a.add(sc.next());
System.out.println(User.highestScorer(a));
sc.close();
}
}
```

```
import java.util.ArrayList;
import java.util.StringTokenizer;

publicclass User {
publicstatic String highestScorer(ArrayList<String>a) {
String ss=null,name=null,Name=null;
intm1=0,m2=0,m3=0,sum=0,max=0;
for(inti=0;i<a.size();i++)
{
ss=a.get(i);
StringTokenizer st=new StringTokenizer(ss,"-");
while(st.hasMoreTokens())
{
name=st.nextToken();
m1=Integer.parseInt(st.nextToken());
m2=Integer.parseInt(st.nextToken());
m3=Integer.parseInt(st.nextToken());
sum=m1+m2+m3;
if(max<sum)
{
max=sum;
Name=name;
}
}
}
returnName;
}
}
```

12. Max Vowels

Solution:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = newScanner(System.in);
String s = sc.nextLine();
System.out.println(User.getWordWithMaximumVowels(s));
}
}


import java.util.StringTokenizer;

publicclass User {
publicstatic String getWordWithMaximumVowels(String s) {
StringTokenizer st = new StringTokenizer(s, " ");
int count = 0, max = 0;
String res = null;
String f = null;
while (st.hasMoreTokens()) {
res = st.nextToken();
count = 0;
for (int k = 0; k < res.length(); k++) {
if (res.charAt(k) == 'a' || res.charAt(k) == 'e'
|| res.charAt(k) == 'i' || res.charAt(k) == 'o'
|| res.charAt(k) == 'u' || res.charAt(k) == 'A'
|| res.charAt(k) == 'E' || res.charAt(k) == 'I'
|| res.charAt(k) == 'O' || res.charAt(k) == 'U')
count++;
if (count > max) {
max = count;
f = res;
}

}
}
return f;
}
}

public class UserMain {

        public static String count(String s) {
                String s1[] = s.split(" ");
```

```
                String v[] = new String[10];
                int max = 0;
                String res = null;
                for (int i = 0; i < s1.length; i++) {
                        v[i] = s1[i].replaceAll("aeiouAEIOU", "");
                        if (v[i].length() > max) {
                                max = v[i].length();
                                res = s1[i];
                        }
                }
                return res;
        }
}
```

13. Adjacent Swaps

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = new Scanner(System.in);
String string=sc.nextLine();
System.out.println(User.swapPairs(string));
sc.close();
}
}


publicclass User {
publicstatic String swapPairs(String s) {
StringBuffer sb=new StringBuffer();
if(s.length()%2==0)
{
for(inti=0;i<s.length()-1;i=i+2)
{
sb.append(s.charAt(i+1)).append(s.charAt(i));
}
}
else
{
for(inti=0;i<s.length()-1;i=i+2)
{
sb.append(s.charAt(i+1)).append(s.charAt(i));
}
sb.append(s.charAt(s.length()-1));
}
returnsb.toString();
}
```

```
}

public class UserMain {

	public static String swap(String s) {
		char[] c = s.toCharArray();
		StringBuffer sb = new StringBuffer();
		if (c.length % 2 == 0) {
			for (int i = 0; i < c.length; i = i + 2)
				sb = sb.append(c[i + 1]).append(c[i]);
		} else {
			for (int i = 0; i < c.length - 1; i = i + 2) {
				sb = sb.append(c[i + 1]).append(c[i]);
			}
			sb = sb.append(c[c.length - 1]);
		}
		return sb.toString();

	}
}
```

14. Password

Solution:

```
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = newScanner(System.in);
String s = sc.next();
boolean flag = User.validatePassword(s);
if (flag == true)
System.out.println("valid");
else
System.out.println("invalid");
}
}

publicclass User {
publicstaticboolean validatePassword(String s) {
int number = 0, c = 0, sp = 0;
boolean flag = false;
for (int i = 0; i < s.length(); i++) {
if (s.length() >= 8) {
if (Character.isDigit(s.charAt(i))) {
number++;
}
```

```
if (Character.isLetter(s.charAt(i))) {
c++;
} else {
if (s.charAt(i) != ' '&& !Character.isDigit(s.charAt(i))
&& !Character.isLetter(s.charAt(i)))
sp++;
}
}
}
if (number >= 1 && c >= 1 && sp >= 1)
flag = true;
return flag;
}
}
```

15. Employee Bonus

Solution:

```
import java.text.ParseException;
import java.util.*;

publicclass Main {
publicstaticvoid main(String[] args) throws ParseException {
Scanner sc = new Scanner(System.in);
intn=sc.nextInt();
TreeMap<Integer,Integer>t=new TreeMap<Integer,Integer>();
HashMap<Integer,String>h1=new HashMap<Integer,String>();
HashMap<Integer,Integer>h2=new HashMap<Integer,Integer>();
for(inti=0;i<n;i++)
{
        intid=sc.nextInt();
        h1.put(id, sc.next());
        h2.put(id, sc.nextInt());
}
t=User.calSalary(h1,h2);
Iterator<Integer>it1=t.keySet().iterator();
while(it1.hasNext())
{
        intid=it1.next();
        intval=t.get(id);
        System.out.println(id);
        System.out.println(val);
}
sc.close();
}
}
```

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;

publicclass User {
publicstatic TreeMap<Integer,Integer> calSalary(HashMap<Integer,String>h1,
HashMap<Integer,Integer>h2) throws ParseException {
        TreeMap<Integer,Integer>t=new TreeMap<Integer,Integer>();
        Iterator<Integer>it1=h1.keySet().iterator();
        SimpleDateFormat sd=new SimpleDateFormat("dd-MM-yyyy");
        String ss="01-09-2014";
        intnew_sal=0;
        while(it1.hasNext())
        {
                intid1=it1.next();
                String dob=h1.get(id1);
                intsalary=h2.get(id1);
                Date d1=sd.parse(dob);
                Date d2=sd.parse(ss);
                d1=sd.parse(dob);
                inty1=d1.getYear();
                inty2=d2.getYear();
                intyear=Math.abs(y1-y2);
                if(year>=25 &&year<=30)
                {
                        new_sal=salary+(salary*20/100);
                        t.put(id1,new_sal);
                }
                elseif(year>=31 &&year<=60)
                {
                        new_sal=salary+(salary*30/100);
                        t.put(id1,new_sal);
                }
                else
                        ;

        }
        returnt;
}
}
```

16. Date Format

Solution:

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

```
publicclass Main {
	publicstaticvoid main(String[] args) throwsParseException {
		Scanner s=new Scanner(System.in);
		String s1=s.next();
		String s2=s.next();
		System.out.println(User.findOldDate(s1,s2));
	}

}


import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;

publicclass User {
	publicstatic String findOldDate(String s1,String s2) throws ParseException
	{
		SimpleDateFormat sd1=new SimpleDateFormat("dd-MM-yyyy");
		Date d1=sd1.parse(s1);
		Date d2=sd1.parse(s2);
		Calendar c=Calendar.getInstance();
		c.setTime(d1);
		intday1=c.get(Calendar.DAY_OF_MONTH);
		intm1=c.get(Calendar.MONTH);
		inty1=c.get(Calendar.YEAR);
		c.setTime(d2);
		intday2=c.get(Calendar.DAY_OF_MONTH);
		intm2=c.get(Calendar.MONTH);
		inty2=c.get(Calendar.YEAR);
		SimpleDateFormat sd2=new SimpleDateFormat("MM/dd/yyyy");
		String res=null;
		if(y1==y2)
		{
			if(m1==m2)
			{
				if(day1==day2)
				{
					res=sd2.format(d1);
				}
			}
				else
				{
					if(m1>m2)
						res=sd2.format(d2);
					else
						res=sd2.format(d1);
```

```
                        }
                    }
                else
                {
                    if(y1>y2)
                        res=sd2.format(d2);
                    else
                        res=sd2.format(d1);
                }

                returnres;
        }
}
```

17. Maximum Difference

```
package practice;

import java.util.Scanner;

public class Main {

        public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
int n=sc.nextInt();
int a[]=new int[n];
for (int i = 0; i < a.length; i++) {
        a[i]=sc.nextInt();
}
System.out.println(UserMainCode.findMaxDistance(n,a));
        }

}
```

```
package practice;

public class UserMainCode {

        public static int findMaxDistance(int n, int[] a) {
                int diff=0;
                int max=0;
                int index=0;
for (int i = 0; i < a.length-1; i++) {
        if(a[i]>a[i+1]){
        diff=a[i]-a[i+1];
        }
        else{
```

```
                diff=a[i+1]-a[i];
        }
        if(diff>max){
                max=diff;
                if(a[i]>a[i+1])
                index=i;
                else
                        index=i+1;
        }
}               return index;
        }

}
```

18. PAN Card

```
package practice;

import java.util.Scanner;

public class Main {

        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                String s = sc.next();
                int res = 0;
                res = UserMainCode.validatePAN(s);
                if (res == 1) {
                        System.out.println("Valid");
                } else {
                        System.out.println("Invalid");
                }
        }

}


package practice;

public class UserMainCode {

        public static int validatePAN(String s) {
                if (s.length() == 8) {
                        if (s.matches("[A-Z]{3}[0-9]{4}[A-Z]{1}")) {
                                return 1;
                        } else
                                return -1;
                } else
```

```
                return -1;
        }
}
```

19. Last Letters

```
package prac;

import java.util.Scanner;

public class Main {

        public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
String s=sc.nextLine();
System.out.println(UserMain.cal(s));
        }

}
```

```
package prac;

public class UserMain {

        public static String cal(String s) {
String a[]=s.split(" ");
StringBuffer sb=new StringBuffer();
for (int i = 0; i < a.length; i++) {
        sb.append(Character.toUpperCase(a[i].charAt(a[i].length()-1))).append("$");
}
sb.deleteCharAt(sb.length()-1);
return sb.toString();
        }
}
```

20. Largest Key in HashMap

```
public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        HashMap<Integer,String> hm= new HashMap<Integer,String>();
        for(int i=0;i<n;i++)
                hm.put(sc.nextInt(),sc.next());
        System.out.println(UserMain.reshape(hm));
}
```

```
public class UserMain {

	public static String reshape(HashMap<Integer, String> hm) {
		int max=0;
		String res=null;
		Iterator<Entry<Integer, String>> i=hm.entrySet().iterator();
		while(i.hasNext()){
			Map.Entry<Integer, String> e=i.next();
			if(e.getKey()>max){
				max=e.getKey();
				res=e.getValue();
			}
		}
		return res;
	}
}
```

21. Day of the Week

```
public class UserMain {

	public static String dayname(String s) {
SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
sdf.setLenient(false);
Date d=null;

	try {
		d=sdf.parse(s);

	} catch (ParseException e) {

	}

Calendar c1= new GregorianCalendar();
c1.setTime(d);
c1.add(Calendar.YEAR, 1);
SimpleDateFormat sdf1=new SimpleDateFormat("EEEE");
return sdf1.format(c1.getTime()).toLowerCase();
	}
}
```

22. Transfer from Hashmap to Arraylist

```
public class UserMain {

	public static List<String> reshape(HashMap<Integer, String> hm) {
		List<String> l=new ArrayList<String>();
		Iterator<Integer> i=hm.keySet().iterator();
```

```
		while(i.hasNext()){
			int m=i.next();
			String n=hm.get(m);
					if(n.matches("[a-z]{1}.*[0-9]{1,}.*[A-Z]{1}")){
						l.add(n);
					}
		}
		return l;
	}
}
```

23. Date Format Conversion

```
public class UserMain {

	public static String rightformat(String s) {
SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
sdf.setLenient(false);
Date d1=null;
try {
	d1=sdf.parse(s);
} catch (ParseException e) {
}
SimpleDateFormat sd=new SimpleDateFormat("dd-MM-yyyy");
String n=sd.format(d1);
return n;
		}
}
```

24. String Processing – ZigZag

```
public class UserMain {

	public static int yy(String s) {

int k=0;
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
sdf.setLenient(false);
Calendar c=Calendar.getInstance();
try {
	Date d=sdf.parse(s);
	c.setTime(d);
	k=c.getActualMaximum(Calendar.DATE);
} catch (ParseException e) {

}
return k;
```

```
}

}
```

25. Age for Voting

```
public class UserMainCode {

	public static String getAgee(String s, String c) {

LocalDate dob=LocalDate.parse(s,DateTimeFormatter.ofPattern("dd/MM/yyyy"));
LocalDate curr=LocalDate.parse(c,DateTimeFormatter.ofPattern("dd/MM/yyyy"));
int res=(int) ChronoUnit.YEARS.between(dob, curr);
if(res>=18){
	return "eligible";
}
else{
	return "not eligible";
}
	}

}
```

26. Constructor Overloading

Main.java

```
import java.util.Scanner;

public class Main {
	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the product id");
		long id = in.nextLong();
		in.nextLine(); // to avoid skipping the input
		System.out.println("Enter the product name");
		String productName = in.nextLine();
		System.out.println("Is the product supplied by Nivas Suppliers? Type yes or
no (not case sensitive)");
		String ans = in.nextLine();
		if(ans.equalsIgnoreCase("no")){
			System.out.println("Enter the supplier name");
			String supplierName = in.nextLine();
			Product pro = new Product(id,productName,supplierName);
			pro.display();
		}
		else{
			Product pro = new Product(id,productName);
			pro.display();
```

```
                }
        }
}
```

Product.java

```
public class Product {
                private long id;
                private String productName;
                private String supplierName;

                public Product(){

                }
                public Product(long id,String productName, String supplierName){
                        this.id = id;
                        this.productName = productName;
                        this.supplierName = supplierName;
                }
                public Product(long id,String productName){
                        this.id = id;
                        this.productName = productName;
                        this.supplierName = "Nivas";
                }
                public void display(){
                        System.out.println("Product Id is "+id);
                        System.out.println("Product Name is "+productName);
                        System.out.println("Supplier Name is "+supplierName);
                }
                public long getId() {
                        return id;
                }
                public void setId(long id) {
                        this.id = id;
                }
                public String getProductName() {
                        return productName;
                }
                public void setProductName(String productName) {
                        this.productName = productName;
                }
                public String getSupplierName() {
                        return supplierName;
                }
                public void setSupplierName(String supplierName) {
                        this.supplierName = supplierName;
                }
```

}

27. Book

Main.java

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

public class Main {
	public static void main(String args[]){
		Scanner in = new Scanner(System.in);
		String name;
		List<Author> authorList = new ArrayList<Author>();
		double price;
		int qtyInStock = 0;
		int numAuthors;
		String ans;
		String authName;
		String email;
		String gender;
		Book book;

		System.out.println("Enter the book name");
		name = in.nextLine();

		System.out.println("Enter the number of authors");
		numAuthors = in.nextInt();
		in.nextLine();

		for(int i=0;i<numAuthors;i++){
			System.out.println("Enter the author name");
			authName = in.nextLine();

			System.out.println("Enter the author email id");
			email = in.nextLine();

			System.out.println("Enter the author's gender");
			gender = in.nextLine();

			authorList.add(new Author(authName, email, gender));
		}

		System.out.println("Enter the book price");
		price = in.nextDouble();
```

```
                System.out.println("Is the book currently available? Type Yes/No (Not case
sensitive)");
                ans = in.next();

                if(ans.equalsIgnoreCase("yes")){
                        System.out.println("Enter the number of books available");
                        qtyInStock = in.nextInt();
                        book = new Book(name, authorList, price, qtyInStock);
                }
                else{
                        book = new Book(name, authorList, price);
                }
                System.out.println(book.toString());
        }
}
```

Author.java

```
public class Author implements Comparable<Author>{
        private String name;
        private String email;
        private String gender;



        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }

        public String getEmail() {
                return email;
        }

        public void setEmail(String email) {
                this.email = email;
        }

        public String getGender() {
                return gender;
        }
```

```
	public void setGender(String gender) {
		this.gender = gender;
	}

	public Author(){

	}

	public Author(String name, String email, String gender) {
		this.name = name;
		this.email = email;
		this.gender = gender;
	}

	public int compareTo(Author auth) {
		//Author auth1 = (Author)auth;
		int diff;
		diff = this.name.compareTo(auth.name);
		return diff;
	}

	public String toString(){
		String str = name +" ("+gender+") contact at "+email;
		return str;
	}


}
```

Book.java

```
import java.util.Iterator;
import java.util.List;


public class Book {
	private String name;
	private List<Author> authorList;
	private double price;
	private int qtyInStock = 0;
	public String getName() {
		return name;
	}
	public List<Author> getAuthorList() {
		return authorList;
```

```
	}
	public double getPrice() {
		return price;
	}
	public int getQtyInStock() {
		return qtyInStock;
	}
	public Book(String name, List<Author> authorList, double price,
			int qtyInStock) {
		this.name = name;
		this.authorList = authorList;
		this.price = price;
		this.qtyInStock = qtyInStock;
	}
	public Book(String name, List<Author> authorList, double price) {
		this.qtyInStock = 0;
		this.name = name;
		this.authorList = authorList;
		this.price = price;
	}

	public String toString(){
		String str;
		Iterator<Author> it = authorList.iterator();
		Author author;
		str = name + " authored by";
		while(it.hasNext()){
			author = it.next();
			str = str+" "+author.getName();
		}
		str = str+" costs Rs."+String.format("%.1f",price)+" : ";
		if(qtyInStock == 0){
			str = str+"Not Available";
		}
		else{
			str = str+"Available";
		}
		return str;
	}


}
```

28. Employee Register

```
import java.util.ArrayList;
import java.util.Iterator;
```

```
import java.util.List;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
public class Main {

	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		String firstName;
		String lastName;
		String mobile;
		String email;
		String address;
		String str;
		Set<String> emp = new TreeSet<String>();
		System.out.println("Enter The Number of Employees");
		int num = in.nextInt();
		in.nextLine();
		for(int i = 1; i<=num; i++){
			System.out.println("Enter Employee "+i+" Details:");
			System.out.println("Enter the Firstname");
			firstName = in.nextLine();
			System.out.println("Enter the Lastname");
			lastName = in.nextLine();
			System.out.println("Enter the Mobile");
			mobile = in.nextLine();
			System.out.println("Enter the Email");
			email = in.nextLine();
			System.out.println("Enter the Address");
			address = in.nextLine();
			str = String.format("%-15s %-15s %-15s %-30s %-
15s",firstName,lastName,mobile,email,address);
			emp.add(str);
		}
		List<String> empList = new ArrayList<String>(emp);

		System.out.println("Employee List:");
		System.out.format("%-15s %-15s %-15s %-30s %-
15s\n","Firstname","Lastname","Mobile","Email","Address");
		Iterator<String> it = empList.iterator();
		while(it.hasNext()){
			System.out.println(it.next());
		}
	}
}
```

29. Collect unique symbols from a set of card

Main.java

```
import java.util.Iterator;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
public class Main {

	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		String symbol;
		String number;
		Set<String> cardSet = new TreeSet<String>();
		Set<String> symbolSet = new TreeSet<String>();
		int count = 0;
		do{
			System.out.println("Enter a card :");
			symbol = in.next();
			number = in.next();
			count++;

			if(symbolSet.contains(symbol)){

			}else{
				symbolSet.add(symbol);
				cardSet.add(new Card(symbol,number).toString());
			}
		}while(cardSet.size() != 4);
		System.out.println("Four symbols gathered in "+count+" cards.");
		System.out.println("Cards in Set are :");
		Iterator<String> it = cardSet.iterator();
		while(it.hasNext()){
			System.out.println(it.next());
		}
	}
}
```

Card.java

```
public class Card {
	String symbol;
	String number;
	public Card(){

	}
	public Card(String symbol, String number) {
		this.symbol = symbol;
```

```
		this.number = number;
	}
	public String toString(){
		String str = symbol+" "+number;
		return str;
	}
}
```

30. Set of boxes

Main.java

```
import java.text.DecimalFormat;
import java.util.Iterator;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
public class Main {

	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the number of Box");
		int num = in.nextInt();
		double length;
		double width;
		double height;
		double volume;
		String str;
		Box box;
		Set<Box> boxSet = new TreeSet<Box>();

		for(int i = 1; i<= num; i++){
			System.out.println("Enter the Box "+i+" details");
			System.out.println("Enter Length");
			length = in.nextDouble();
			System.out.println("Enter Width");
			width = in.nextDouble();
			System.out.println("Enter Height");
			height = in.nextDouble();

			boxSet.add(new Box(length, width, height));
		}

		System.out.println("Unique Boxes in the Set are");
		Iterator<Box> it = boxSet.iterator();
		while(it.hasNext()){
			box = it.next();
```

```
                        volume = box.length * box.width * box.height;
                        //System.out.format("Length =%.1f Width =%.1f Height =%.1f Volume
=%.2f\n",box.length,box.width,box.height,volume);
                        DecimalFormat df = new DecimalFormat("0.0#");
                        DecimalFormat df1 = new DecimalFormat("0.0#");
                        String vol = df1.format(volume);

                        str = "Length ="+df.format(box.length)+" Width
="+df.format(box.width);
                        str = str+" Height ="+df.format(box.height)+" Volume ="+vol;
                        System.out.println(str);

                }
        }
}
```

Box.java

```
public class Box implements Comparable<Box>{
        double length;
        double width;
        double height;

        public Box(){

        }

        public Box(double length, double width, double height) {
                this.length = length;
                this.width = width;
                this.height = height;
        }

        @Override
        public int compareTo(Box obj) {
                double volume1 = this.length * this.width * this.height;
                double volume2 = obj.length * obj.width * obj.height;

                volume1 = volume1*100;
                volume2 = volume2*100;

                int diff = (int)volume1 - (int)volume2;

                return diff;
        }
        public boolean equals(Box obj){
                boolean flag = false;
                double volume1 = this.length * this.width * this.height;
```

```
                double volume2 = obj.length * obj.width * obj.height;
                if((volume1-volume2) == 0){
                        flag = true;
                }
                return flag;}}
```

31. Profit or Loss

```
import java.util.Scanner;

public class Main {
        public static void main(String[] args) {
                Scanner in = new Scanner(System.in);
                System.out.println("Enter the number of dozens of toys purchased");
                int numDozens = in.nextInt();
                System.out.println("Enter the price per dozen");
                int price = in.nextInt();
                System.out.println("Enter the selling price of 1 toy");
                int toyPrice = in.nextInt();
                double cost = price/12.0;
                double profit = toyPrice - cost;
                double profitPercent = profit/cost*100;
                String str = String.format("%.2f", profitPercent);
                System.out.println("Sam's profit percentage is "+str+" percent");
        }
}
```

32. Math class

```
import java.util.Scanner;

public class Main {
        public static void main(String[] args) {
                Integer num1;
                Integer num2;
                Scanner in = new Scanner(System.in);
                System.out.println("Enter the first integer");
                num1 = in.nextInt();
                System.out.println("Enter the second integer");
                num2 = in.nextInt();
                System.out.println("Absolute value of "+num1+" is "+Math.abs(num1));
                System.out.println("Absolute value of "+num2+" is "+Math.abs(num2));
                if(num1.equals(num2)){
                        System.out.println(num1+" = "+num2);
                }
                else{
                        System.out.println(num1+" != "+num2);
                }
```

```
	}
}
```

33. Wrapper class

```
import java.util.Scanner;

public class Main {
	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the binary number");
		String binary = in.next();
		System.out.println("Enter the octal number");
		String octal = in.next();
		System.out.println("Enter the hexadecimal number");
		String hex = in.next();
		System.out.println("The integer value of the binary number "+binary+" is
"+Integer.parseInt(binary, 2));
		System.out.println("The integer value of the octal number "+octal+" is
"+Integer.parseInt(octal, 8));
		System.out.println("The integer value of the hexadecimal number "+hex+" is
"+Integer.parseInt(hex, 16));

	}
}
```

34. Operations on String List

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;
public class Main {

	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		String item;
		List<String> list = new ArrayList<String>();
		boolean flag = true;
		boolean flag1;
		while(flag){
			System.out.println("1. Insert");
			System.out.println("2. Search");
			System.out.println("3. Delete");
			System.out.println("4. Display");
			System.out.println("5. Exit");

			System.out.println("Enter your choice :");
```

```
                int choice = Integer.parseInt(in.nextLine());

                switch(choice){
                case 1:
                        System.out.println("Enter the item to be inserted:");
                        item = in.nextLine();
                        list.add(item);
                        System.out.println("Inserted successfully");
                        break;
                case 2:
                        System.out.println("Enter the item to search :");
                        item = in.nextLine();
                        flag1 = listSearch(list,item);
                        if(flag1){
                                System.out.println("Item found in the list.");
                        }else{
                                System.out.println("Item not found in the list.");
                        }
                        break;
                case 3:
                        System.out.println("Enter the item to delete :");
                        item = in.nextLine();
                        flag1 = listDelete(list, item);
                        if(flag1){
                                System.out.println("Deleted successfully");
                        }else{
                                System.out.println("Item does not exist.");
                        }
                        break;
                case 4:
                        System.out.println("The Items in the list are :");
                        listDisplay(list);
                        break;
                case 5:
                        flag = false;
                        break;
                default:
                        flag = false;
                }
        }
}

public static boolean listSearch(List<String> list, String item){
        Iterator<String> it = list.iterator();
        boolean flag = false;
        while(it.hasNext()){
                if(item.equalsIgnoreCase(it.next())){
```

```
                        flag = true;
                    }
                }
                return flag;
        }
        public static boolean listDelete(List<String> list, String item){
                List<String> temp = new ArrayList<String>();
                boolean flag;
                temp.add(item);
                flag = listSearch(list, item);
                if(flag){
                        list.removeAll(temp);
                }
                return flag;
        }
        public static void listDisplay(List<String> list){
                Iterator<String> it = list.iterator();
                while(it.hasNext()){
                        System.out.println(it.next());
                }
        }
}
```

35. Collect and group cards

Main.java

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;
public class Main {

        public static void main(String[] args) {
                Scanner in = new Scanner(System.in);
                String symbol;
                int number;
                int i;
                int sum = 0;
                Map<String, List<Integer>> cardMap = new HashMap<String,
List<Integer>>();
```

```
        Set<String> symbolSet = new TreeSet<String>();

        System.out.println("Enter Number of Cards : ");
        int num = in.nextInt();
        for(i =1;i<=num;i++){
                System.out.println("Enter card "+i+":");
                symbol = in.next();
                number = in.nextInt();

                symbolSet.add(symbol);
                if(cardMap.containsKey(symbol)){
                        cardMap.get(symbol).add(number);
                }else{
                        List<Integer> list = new ArrayList<Integer>();
                        list.add(number);
                        cardMap.put(symbol, list);
                }

                }
        Iterator<String> itSet = symbolSet.iterator();
        System.out.println("Distinct Symbols are : ");
        while(itSet.hasNext()){
                System.out.print(itSet.next()+" ");
        }
        System.out.println();
        itSet = symbolSet.iterator();
        Iterator<Integer> itList;
        while(itSet.hasNext()){
                String str = itSet.next();
                itList = cardMap.get(str).iterator();
                System.out.println("Cards in "+str+" Symbol");
                while(itList.hasNext()){
                        i = itList.next();
                        System.out.println(str+" "+i);
                        sum = sum + i;
                }
                System.out.println("Number of cards : "+cardMap.get(str).size());
                System.out.println("Sum of Numbers : "+sum);
                sum = 0;
        }
}
```

```
}
```

Card.java

```
public class Card {
	String symbol;
	String number;
	public Card(){

	}
	public Card(String symbol, String number) {
		this.symbol = symbol;
		this.number = number;
	}
	public String toString(){
		String str = symbol+" "+number;
		return str;
	}
}
```