

JAVA PROGRAMMING LAB

Exercise-1

1. John found another manuscript of ancient mathematicians.

According to this manuscript an integer k is a lucky number if $k = a_1 + a_2 + \dots + a_n$, where: $a_i = 7^p$. p may be any positive integer. if i and j are distinct, $a_i \neq a_j$

For example 7 is a lucky number: $7 = 7^1$. 56 is a lucky number $56 = 7^2 + 7^1$.

John has an array of n integers. He wants to determine how many members of this array are lucky. He is not good at programming and needs your help. Write a program which takes an integer n and array consisting of n integers and determines quantity of lucky integers in this array.

Input

The first line of input contains integer n : number of elements in the array. The second line of input contains n space separated integers.

Output

Print the number of lucky integers in a given array.

Constraints

$1 \leq n \leq 100$

$1 \leq \text{arr}[i] \leq 1000000$. $\text{arr}[i]$ is the i^{th} element of array.

Example #1

Input

2

49 50

Output

1

$49 = 7^2$. 50 can't be represented as a sum of distinct powers of 7.

Example #2

Input

2

7 49

Output

2

$7 = 7^1$ and $49 = 7^2$.

Program:

```
import java.util.*;

import java.lang.*;

public class Ln
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);

        System.out.println("Enter size of an array");

        int n=s.nextInt();

        if(n<1 || n>100)
        {
            System.out.println("Try Again");
        }
        else
        {
            System.out.println("Enter"+n+"Elements");

            int a[]=new int[n];

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

```

        {
            a[i]=s.nextInt();
        }
        find_Luckynumber(n,a);
    }
}

public static void find_Luckynumber(int n,int a[])
{
    int c=0,sum=0;
    for(int i=0;i<n;i++)
    {
        if(a[i]<1 || a[i]>100000000)
        {
            continue;
        }
        else
        {
            sum=0;
            for(int j=1;j<=8;j++)
            {

```

```
        sum+=(int)Math.pow(7,j);
        if(sum==a[i])
        {
            c++;
            break;
        }
    }
}

System.out.println("no.of Lucky number:"+c);
}
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Ln.java

D:\20481A05J9 (JAVA)>java Ln
Enter size of an array
5
Enter 5 Elements
7
49
56
399
349
no.of Lucky number:3

D:\20481A05J9 (JAVA)>
```

- 2.** Write a Java program that reads an integer number (between 1 and 255) from the user and prints the binary representation of the number. The answer should be printed as a String.

Note: The output displayed should contain 8 digits and should be padded with leading 0s(zeros), in case the returned String contains less than 8 characters.

For example, if the user enters the value 16, then the output should be 00010000 and if the user enters the value 100, the output should be 01100100

(Hint : You may use String.format() method for the expected output)

Program:

```
import java.lang.*;

import java.util.*;

public class Binary
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);

        System.out.println("ENTER  AN INTEGER BETWEEN 1 TO 255");

        int n=s.nextInt();

        if(n<1 || n>255)

            System.out.println(" INPUT MUST ENTER BETWEEN 1 TO 255");

        else

            { String b=Integer.toBinaryString(n);

              //String b=String.format("
%08d",Integer.valueOf(Integer.toBinaryString(n)));

              System.out.println("  Binary value is:"+b);

            }

        }

    }
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Binary.java

D:\20481A05J9 (JAVA)>java Binary
ENTER AN INTEGER BETWEEN 1 TO 255
245
    Binary value is:11110101

D:\20481A05J9 (JAVA)>java Binary
ENTER AN INTEGER BETWEEN 1 TO 255
452
    INPUT MUST ENTER BETWEEN 1 TO 255
```

Exercise-2

3. *Software is being developed by a university that displays SGPA of your current semester. You are given the task to develop a module that calculates the SGPA with respect to the secured grade points corresponding to given number of credits in each subject.*

The credits for the courses are:

Graphics: 2, PPS: 4, JAVA: 3, Chemistry: 3, English: 2, Technical Skills: 1.5, Data Structures: 4 Complete your Module by displaying the SGPA of current semester.

Program:

```
import java.lang.*;

import java.util.*;

public class Week2a
{
    public static void main(String args[])
    {
```

```
String sub[]={"Graphics","PPS","JAVA","Chemistry","English","Technical
Skills","Data Structures"};

double credits[]={2,4,3,3,2,1.5,4};

int points=0;

double sgpa=0,sum=0;

Scanner s=new Scanner(System.in);

for(int i=0;i<sub.length;i++)
{
    System.out.println("Enter the subject"+sub[i]+"Grade");

String str=s.next();

if(str.equals("A+"))

    points=10;

else if(str.equals("A"))

    points=9;

    else if(str.equals("B"))

        points=8;

else if(str.equals("C"))

    points=7;

else if(str.equals("D"))

    points=6;

else if(str.equals("E"))

    points=5;

else if(str.equals("F"))
```



```
        points=0;
else{
    System.out.println("Invalid Grade");
    i--;
}
if(points==0)
    break;
sgpa=sgpa+points*credits[i];
sum=sum+credits[i];
}
    if(points==0)
        System.out.println("The Person Fails");
    else
        System.out.println("The SGPA is "+(sgpa/sum));
}
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Week2a.java

D:\20481A05J9 (JAVA)>java Week2a
Enter the subjectGraphicsGrade
A+
Enter the subjectPPSGrade
A
Enter the subjectJAVAGrade
B
Enter the subjectChemistryGrade
D
Enter the subjectEnglishGrade
B
Enter the subjectTechnical SkillsGrade
A
Enter the subjectData StructuresGrade
B
The SGPA is8.179487179487179
```

```
D:\20481A05J9 (JAVA)>java Week2a
Enter the subjectGraphicsGrade
A
Enter the subjectPPSGrade
B
Enter the subjectJAVAGrade
V
Invalid Grade
Enter the subjectJAVAGrade
F
The Person Fails
```

4. You are given a string consisting of n lowercase Latin letters. You must find the count of number of larger alphabets for every character of the string (according to lexicographical order).

Input

The first line of input contains an integer n , the length of the given string. The second line of input contains a string.

Output

Print the count of number of larger alphabets for every character of the string on a single line. Separate elements by white spaces.

Constraints

$$1 \leq n \leq 100$$

Example#1

Input

3

abc

Output

2 1 0

Explanation: $a - 2$: ' a ' < ' b ', ' a ' < ' c '. ' b ' - 1: ' b ' < ' c '. ' c ' - 0: There is no letter in this string, which is larger than ' c '.

Example#2

Input 5

aaabb

2 2 2 0 0

Explanation: $a - 2$: ' a ' < ' b ' (b at index 3 and index 4). There is no letter in this string, which is larger than ' b '.

Program:

```
import java.lang.*;
```

```
import java.util.*;
```

```
public class Week2b
```

```
{  
    public static void main(String args[])  
    {  
        int count;  
Scanner s=new Scanner(System.in);  
System.out.println("Enter the string length");  
int n=s.nextInt();  
System.out.println("Enter the string length upto "+n);  
String str=s.next();  
char ch[]=str.toCharArray();  
for(int i=0;i<n;i++)  
{  
    count=0;  
    for(int j=i+1;j<n;j++)  
    {  
        if(ch[i]<ch[j])  
{  
            count++;  
        }  
    }  
    System.out.print(count);  
    }  
}  
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Week2b.java

D:\20481A05J9 (JAVA)>java Week2b
Enter the string length
4
Enter the string length upto 4
abcd
3210
D:\20481A05J9 (JAVA)>java Week2b
Enter the string length
6
Enter the string length upto 6
781524
103010
D:\20481A05J9 (JAVA)>java Week2b
Enter the string length
3
Enter the string length upto 3
999
000
```

Exercise-3

5. *Tom and Jerry found two bags of apples. The bag that Jerry chose contains 5 apples and the bag chosen by Tom has 3 apples. Tom wants to have more apples, so he swaps the bags. Write a program to display the apples in the two bags before and after swapping.*

Hint :-(Try using call by value and call by reference; Write which can be used to swap)

Program:

//call by value

import java.lang.*;

```

import java.util.*;

class Week
{
    int x,y;

    public void show(Week z)
    {
        int temp;

        temp=z.x;

        z.x=z.y;

        z.y=temp;

        System.out.println("The value of method is"+z.x+" "+z.y);
    }
}

class Week3b
{
    public static void main(String args[])
    {
        Week c=new Week();

        c.x=3;

        c.y=5;

        System.out.println("Before swapping values are"+c.x+" "+c.y);

        c.show(c);

        System.out.println("After swapping values are"+c.x+" "+c.y);
    }
}

```

Output:

```
D:\20481A05J9 (JAVA)>javac Week3a.java

D:\20481A05J9 (JAVA)>java Week3a
enter the 2 values:
3
9
Before swapping3 9
After swapping3 9
```

//Call by reference

```
import java.lang.*;
import java.util.*;
class Week
{
    int x,y;
    public void show(Week z)
    {
        int temp;
        temp=z.x;
        z.x=z.y;
        z.y=temp;
        System.out.println("The value of method is"+z.x+" "+z.y);
    }
}
class Week3
{
```

```

public static void main(String args[])
{
    Week c=new Week();

    c.x=3;

    c.y=5;

    System.out.println("Before swapping values are"+c.x+" "+c.y);

    c.show(c);

    System.out.println("After swapping values are"+c.x+" "+c.y);
}
}

```

Output:

```

D:\20481A05J9 (JAVA)>javac Week3.java

D:\20481A05J9 (JAVA)>java Week3
Before swapping values are3 5
The value of method is5 3
After swapping values are5 3

```

(call by reference should be used for swap)

6. *Access the instance variables by using “this” and super keywords.*

Program:

```

class Box
{
    int length,breadth,height;

    Box(int length,int breadth,int height)

```



```

        {
            this.length=length;
            this.breadth=breadth;
            this.height=height;
        }
    }

class Box1 extends Box
{
    int base;

    Box1(int length,int breadth,int height,int base)
    {
        super(length,breadth,height);
        this.base=base;
    }

    void display()
    {
        System.out.println("The parent class
variable:"+length="+super.length+"breadth="+super.breadth+"height="+super.h
eight);

        System.out.println("The child class variable is:"+base);
    }
}

class Exp6
{
    public static void main(String args[])

```

```

{
    Box1 obj=new Box1(10,20,30,40);
    obj.display();
}
}

```

Output:

```

D:\20481A05J9 (JAVA)>javac Exp6.java

D:\20481A05J9 (JAVA)>java Exp6
The parent class variable:length=10breadth=20height=30
The child class variable is:40

```

Exercise-4

7. Create an abstract class named shape, that contains an empty method named numberOfSides(). Define three classes named Trapezoid, Triangle and Hexagon, such that each one of the classes contains only the method numberOfSides(), that contains the number of sides in the given geometrical figure.

Program:

abstract class shape

```

{
    abstract void number_of_sides();
}

```

class Trapezoid extends shape

```

{
    void number_of_sides()
}

```

```

    {
        System.out.println(" The number of sides of Trapezoid=4");
    }
}

class Triangle extends shape
{
    void number_of_sides()
    {

        System.out.println(" The number of sides of Triangle =3");
    }
}

class Hexagon extends shape
{
    void number_of_sides()
    {

        System.out.println(" The number of sides of Hexagon =6");
    }
}

class Week4a
{
    public static void main(String args[])
    {

```

```
        shape obj;  
  
obj=new Trapezoid();  
obj.number_of_sides();  
  
obj=new Triangle ();  
obj.number_of_sides();  
  
obj=new Hexagon ();  
obj.number_of_sides();  
  
    }  
  
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Week4a.java  
  
D:\20481A05J9 (JAVA)>java Week4a  
The number of sides of Trapezoid=4  
The number of sides of Triangle =3  
The number of sides of Hexagon =6
```

8. *You are supposed to calculate the area of a polygon based on number of inputs given by the user.*

Polygon can be a square, a rectangle or a triangle.

Program:

```
import java.lang.*;  
  
import java.util.*;  
  
class Area  
{
```

```
int Square(int a)
```

```
{
```

```
    return a*a;
```

```
}
```

```
int Square(int length,int breadth)
```

```
{
```

```
    return (length*breadth);
```

```
}
```

```
double Square(int a,int b,int c)
```

```
{
```

```
    int s;
```

```
    s=(a+b+c)/2;
```

```
    double area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
```

```
    return area;
```

```
}
```

```
}
```

```
class Week4b
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        Area obj=new Area();
```

```
        System.out.println("Enter the sides of polygon:");
```

```

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();


    System.out.println("The area of ploygon:"+obj.Square(n));


int length=sc.nextInt();
int breadth=sc.nextInt();

    System.out.println("the area of polygon="+obj.Square(length,breadth));

    int a=sc.nextInt();
int b=sc.nextInt();
int c=sc.nextInt();

    System.out.println("The area is:"+obj.Square(a,b,c));
}
}

```

Output:

```

D:\20481A05J9 (JAVA)>javac Week4b.java

D:\20481A05J9 (JAVA)>java Week4b
Enter the sides of polygon:
6
The area of ploygon:36
2
9
the area of polygon=18
9
12
8
The area is:28.982753492378876

```

Exercise-5

9. Write an interface called *Playable*, with a method *void play()*; Let this interface be placed in a package called *music*.

Write a class called *Veena* which implements *Playable* interface. Let this class be placed in a package *music.string*

Write a class called *Saxophone* which implements *Playable* interface. Let this class be placed in a package *music.wind*

Write another class *Test* in a package

called *live*. Then, **a.** Create an

instance of *Veena* and call *play()*

method

b. Create an instance of *Saxophone* and call *play()* method

c. Place the above instances in a variable of type *Playable* and then call *play()*

Program:

//playable

```
package music;
```

```
public interface Playable
```

```
{
```

```
    void play();
```

```
}
```

//Veena

```
package music.string;
```

```
import music.*;
```

```
public class Veena implements Playable
{
    public void play()
    {
        System.out.println("Veena is playing .....");
    }
}
```

//Saxophone

```
package music.wind;
import music.*;
public class Saxophone implements Playable
{
    public void play()
    {
        System.out.println("Saxophone is playing .....");
    }
}
```

//Main program

```
package live;
import music.*;
import music.string.Veena;
import music.wind.Saxophone;
public class Test
{
```



```

public static void main(String args[])
{
    Playable obj1=new    Veena();
    Playable obj2=new Saxophone();
    obj1.play();
    obj2.play();
}
}

```

Output:

```

D:\20481A05J9 (JAVA)>javac -d . Playable.java
D:\20481A05J9 (JAVA)>javac -d . Saxophone.java
D:\20481A05J9 (JAVA)>javac -d . Veena.java
D:\20481A05J9 (JAVA)>javac Test.java
D:\20481A05J9 (JAVA)>java live.Test
Veena is playing .....
Saxophone is playing .....

```

10. *Create and access a user defined package where the package contains a class named CircleDemo, which in turn contains a method called circleArea() which takes radius of the circle as the parameter and returns the area of the circle.*

Program:

```

package Mypack;

```

```
public class CircleDemo
```

```
{
```

```
    public double circleArea(double r)
```

```
    {
```

```
        return 3.14*r*r;
```

```
    }
```

```
}
```

```
import Mypack.CircleDemo;
```

```
import java.util.*;
```

```
public class Area
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        double area;
```

```
        double r;
```

```
        Scanner sc=new Scanner(System.in);
```

```
        System.out.println("Enter the radius of the circle:");
```

```
        r=sc.nextDouble();
```

```
        CircleDemo x=new CircleDemo();
```

```
        area=x.circleArea(r);
```

```
        System.out.println("The area of the circle is:"+area);
```

```
    }
```

```
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac -d . CircleDemo.java

D:\20481A05J9 (JAVA)>javac Area.java

D:\20481A05J9 (JAVA)>java Area
Enter the radius of the circle:
3.5
The area of the circle is:38.465

D:\20481A05J9 (JAVA)>java Area
Enter the radius of the circle:
4
The area of the circle is:50.24
```

Exercise-6

11. *Handle the following exceptions using exception handling mechanism in java. (Note: Handle all exceptions in single program using command line arguments)*

- a. *ArithmeticException*
- b. *ArrayIndexOutOfBoundsException*
- c. *NullPointerException*
- d. *StringIndexOutOfBoundsException*
- e. *NumberFormatException*

Program:

```
import java.util.*;

import java.lang.*;

public class Week6a
{
    public static void main(String args[])
    {
```

```
int a=args.length;
int b=10;
try
{
int c=b/a;
System.out.println("Division:"+c);
int d[]={10,20,30,40,50};
int e=Integer.parseInt(args[0]);
d[e]=25;
System.out.println("The updated array ele is"+d[e]);
String f=args[1];
String str;
if(f.equals("true"))
    str=null;
else
str="Hello CSE-D welcome to Java lab";
int g=Integer.parseInt(args[2]);
System.out.println(str.substring(g));

}
catch(ArithmeticException e)
{
    System.out.println(e.toString());
}
```

```
    catch(ArrayIndexOutOfBoundsException e)
    {
        System.out.println(e.toString());
    }
catch(NullPointerException e)
{
    System.out.println(e.toString());
}
catch(StringIndexOutOfBoundsException e)
{
    System.out.println(e.toString());
}
catch(NumberFormatException e)
{
    System.out.println(e.toString());
}
}
```

Output:

```

D:\20481A05J9 (JAVA)>javac Week6a.java

D:\20481A05J9 (JAVA)>java Week6a 1 2 hai
Division:3
The updated array ele is25
java.lang.NumberFormatException: For input string: "hai"

D:\20481A05J9 (JAVA)>java Week6a 67
Division:10
java.lang.ArrayIndexOutOfBoundsException: Index 67 out of bounds for length 5

D:\20481A05J9 (JAVA)>java Week6a 1 4 6
Division:3
The updated array ele is25
CSE-D welcome to Java lab

D:\20481A05J9 (JAVA)>java Week6a true
Division:10
java.lang.NumberFormatException: For input string: "true"

D:\20481A05J9 (JAVA)>java Week6a 2 50
Division:5
The updated array ele is25
java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2

D:\20481A05J9 (JAVA)>java Week6a 2 34 67
Division:3
The updated array ele is25
java.lang.StringIndexOutOfBoundsException: begin 67, end 31, length 31

```

12. Write a java program to create three userdefined exceptions and throw the exceptions using throw and write appropriate catch and finally blocks to handle.

Program:

```

import java.util.*;

class MarksException extends Exception
{
    MarksException(String msg)

    {
        super(msg);
    }
}

```

```
    }  
}  
class GradeException extends Exception  
{  
    GradeException(String msg)  
    {  
        super(msg);  
    }  
}
```

```
class RollNoException extends Exception  
{  
    RollNoException(String msg)  
    {  
        super(msg);  
    }  
}
```

```
public class Week6b  
{  
  
    public static void main(String args[])  
    {  
        int marks;
```

```

String grade="A+,A,B,C,D,E,F";
String rollno="481A05,481A54,481A42";

    try
{
System.out.println("enter the rollno:" );
Scanner s=new Scanner(System.in);
String roll=s.next ();
if(roll.length()!=10 || rollno.indexOf(roll.substring(2,8))!=-1)
throw new RollNoException("Invalid Rollno");

System.out.println("Enter the marks:");
marks=s.nextInt();
if(marks<0 || marks>100)
throw new MarksException("Invalid marks");
System.out.println("Enter the Grade:");

String gr=s.next ();

if( grade.indexOf(gr)==-1)
throw new GradeException("Invalid Grade");
System.out.println(" marks "+marks);
System.out.println(" Rollno "+roll);
System.out.println(" Grade "+gr);

}

catch(RollNoException e)
{

```



```
System.out.println(e.getMessage());
    }
    catch(MarksException e)
    {
System.out.println(e.getMessage());
    }
    catch(GradeException e)
    {
System.out.println(e.getMessage());
    }
    finally
    {
System.out.println( "Exception Handled");
    }
}
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Week6b.java

D:\20481A05J9 (JAVA)>java Week6b
enter the rollno:
20481A05J9
Enter the marks:
92
Enter the Grade:
A+
marks 92
Rollno 20481A05J9
Grade A+
Exception Handled

D:\20481A05J9 (JAVA)>java Week6b
enter the rollno:
481A05
Invalid Rollno
Exception Handled
```

Exercise-7

13. *Create three threads (by using Thread class and Runnable interface) where the first thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.*

Program:

```
import java.lang.*;

import java.util.*;

class Thread1 implements Runnable
```

```
{  
    public void run()  
    {  
        try  
        {  
            for(int i=5;i>0;i--)  
            {  
                System.out.println("Good Morning");  
                Thread.sleep(1000);  
            }  
        }  
        catch(InterruptedException e)  
        {  
            System.out.println(e.toString());  
        }  
        System.out.println("Thread1 exit");  
    }  
}
```

class Thread2 implements Runnable

```
{  
    public void run()  
    {  
        try  
        {
```

```

        for(int i=5;i>0;i--)
        {
            System.out.println("Hello");
            Thread.sleep(2000);
        }
    }
    catch(InterruptedException e)
    {
        System.out.println(e.toString());
    }
    System.out.println("Thread2 exit");
}
}

class Thread3 implements Runnable
{
    public void run()
    {
        try
        {
            for(int i=5;i>0;i--)
            {
                System.out.println("Welcome");
                Thread.sleep(3000);
            }
        }
    }
}

```

```

    }
    catch(InterruptedException e)
    {
        System.out.println(e.toString());
    }
    System.out.println("Thread3 exit");
}
}
public class Week7a
{
    public static void main(String args[])
    {
        Thread t1= new Thread(new Thread1());
        Thread t2= new Thread(new Thread2());
        Thread t3= new Thread(new Thread3());
        t1.start();
        t2.start();
        t3.start();
    }
}

```

Output:

```
D:\20481A05J9 (JAVA)>javac Week7a.java

D:\20481A05J9 (JAVA)>java Week7a
Hello
Good Morning
Welcome
Good Morning
Hello
Good Morning
Welcome
Good Morning
Hello
Good Morning
Thread1 exit
Hello
Welcome
Hello
Welcome
Thread2 exit
Welcome
Thread3 exit
```

14. *Create three threads- with different priorities – MAX, MIN, NORM- and start the threads at the same time. Observe the completion of the threads.*

Program:

```
import java.lang.*;

import java.util.*;

class Thread1 extends Thread
{
    public void run()
```

```

{
    try

        {
            System.out.println("Hello I am"+Thread.currentThread().getName());
            Thread.sleep(1000);
        }

        catch(InterruptedException e)
        {
            System.out.println(e.toString());
        }

    }
}

public class Week7b
{
    public static void main(String args[])
    {
        Thread t1= new Thread(new Thread1(),"First");
        Thread t2= new Thread(new Thread1(),"Second");
        Thread t3= new Thread(new Thread1(),"Third");
        t3.setPriority(Thread.MAX_PRIORITY);
        t2.setPriority(Thread.MIN_PRIORITY);
    }
}

```

```
        t1.setPriority(Thread.NORM_PRIORITY);  
        t1.start();  
        t2.start();  
        t3.start();  
    }  
}
```

Output:

```
D:\20481A05J9 (JAVA)>javac Week7b.java  
  
D:\20481A05J9 (JAVA)>java Week7b  
Hello I amThird  
Hello I amSecond  
Hello I amFirst
```


