

weak,

1. Glaxo needs to determine which of 2 machines is operating closest to the optimal temp of $100^{\circ} C$ using relational operator.

```
import java.util.Scanner;  
class temp{
```

```

public static void main (String [ ] args) {
    Scanner s = new Scanner (System.in);
    int a = s.nextInt ();
    int b = s.nextInt ();
    int c, d;
    if (a > 100) {
        c = a - 100;
    }
    else {
        c = 100 - a;
    }
    if (b > 100) {
        d = b - 100;
    }
    else {
        d = 100 - b;
    }
    if (c < d) {
        System.out.println (c);
    }
    else {
        System.out.println (d);
    }
}

```

```
System.out.println("The integer closer to 100 is  
+ a + " with a difference of "+c);  
g
```

6. Joey needs to write a program that takes an Int and the no of bits N as input & outputs the value of the lowest N bits of the Int

Soln:-

```
import java.util.Scanner;
class Joey{
    public static void main (String[] args){
        Scanner s = new Scanner (System. in);
        Int a = s.nextInt ();
        Int b = s.nextInt ();
        Int c = (1 << b) - 1;
        Int d = a & c;
        System.out.println ("Result: " + d);
    }
}
```

7. Miles is analyzing two integers. He wants to check if either one of the integers is both:

- Less than or equal to zero
- Odd

Soln:-

```
import java.util.Scanner;
class miles{
    public static void main (String[] args){
        Scanner s = new Scanner (System. in);
    }
}
```

```
Int a = s.nextInt ();
Int b = s.nextInt ();
if ((a <= 0 && Math.abs (a + 2) == 1) || (b <= 0 && Math.abs (b + 2) == 1)) {
    System.out.println ("true");
} else {
    System.out.println ("false");
}
}
```

8. The kingdom of finance, if the avg daily expense is greater than ~~two~~ of the categories, you need to calculate the avg of the three expenses & check if it is greater than any two categories.

Soln:-

```
import java.util.Scanner;
class miles{
    public static void main (String[] args){
        Scanner s = new Scanner (System. in);
        Int a = s.nextInt (), b, c;
        Int b = s.nextInt ();
        Int c = s.nextInt ();
        double avg = (a + b + c) / 3.0;
    }
}
```

else {

System.out.println ("The integer closer to 100 is
+ b + " with a difference of " + d);

}

}

2. Dave got two students. Each handout an int a
wants to find if one int +ve while the other is
not divisible by 3.

Soln:

```
import java.util.Scanner;  
class Dave {
```

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

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

```
int a = s.nextInt();
```

```
int b = s.nextInt();
```

```
if (a > 0) {
```

```
if (b % 3 != 0) {
```

```
System.out.println ("One of the integers is  
positive while the other is not divisible by 3.");
```

}

```
else {
```

```
if (b > 0) {
```

```
if (a % 3 != 0) {
```

```
System.out.println ("One of the integers  
is positive while the other is not  
divisible by 3.");
```

}

else {

```
System.out.println ("Neither of the integers  
meets the condition.");
```

}

else {

```
System.out.println ("Neither of the integers  
meets the condition.");
```

}

else {

```
if (b > 0) {
```

✓ if (a % 3 != 0) {

```
System.out.println ("One of the integers is  
positive while the other is not divisible  
by 3.");
```

}

else {

```
System.out.println ("Neither of the integers  
meets the condition.");
```

}

else {

```
System.out.println ("Neither of the integers meets  
the condition.");
```

}

```
System.out.println ("%.2f\n", avg);
```

```
if (a>b && a>c) {
```

```
    s1 = b;
```

```
    s2 = c;
```

```
}
```

```
else if (b>a && b>c) {
```

```
    s1 = a;
```

```
    s2 = c;
```

```
}
```

```
else {
```

```
    s1 = a;
```

```
    s2 = b;
```

```
}
```

```
if (avg>s1 && avg>s2) {
```

System.out.println ("Average is greater
than both "+s1+" and "+s2);

```
}
```

```
else {
```

System.out.println ("Average is not greater
than two smallest expenses");

```
}
```

```
}
```

```
}
```

- q. Phil needs to verify if a sensor reading at a mid station (s2) falls exactly halfway b/w the reading of the prev station (s1) & the next station (s3).

Soln:

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

```
class phil {
```

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

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

```
        int s1 = s.nextInt();
```

```
        int s2 = s.nextInt();
```

```
        int s3 = s.nextInt();
```

```
        boolean mid = s2 == (s1+s3)/2;
```

```
        System.out.println (mid);
```

```
    if (mid) {
```

System.out.println ("The second integer is
 halfway b/w the between the first and third
 integers.");

```
}
```

```
else {
```

System.out.println ("The second integer is
 not halfway between the first and third
 integers.");

```
}
```

```
}
```

```
}
```

```
}
```

3. Manoj needs to create a program that takes an int input, converts it into a double, & display both the original int, and the converted double value.

Soln:-

```
import java.util.Scanner;
class manoj{
    public static void main (String [] args){
        Scanner s=new Scanner (System.in);
        int a = s.nextInt();
        System.out.println ("Original Integer: " + a);
        System.out.println ("Converted Double: " + (double)a);
    }
}
```

4. Vishal gives Arun two int. He asks Arun to check if the sum of these two numbers is a multiple of their product.

Soln:-

```
import java.util.Scanner;
class vishal{
    public static void main (String [] args){
        Scanner s = new Scanner (System.in);
        int a = s.nextInt();
        int b = s.nextInt();
        int c = a+b;
        int d = a*b;
        if (c+d == 0){
            System.out.println ("Sum is Multiple of Product");
        }
    }
}
```

5. 5
System.out.println ("Sum is Not Multiple of Product");
}

6. Emily has a beautiful circular garden in her backyard. To calculate the circumference and area,

$$C = 2\pi r$$

$$A = \pi r^2$$

Soln:-

```
import java.util.Scanner;
class emily{

```

```
    public static void main (String [] args){
        Scanner s = new Scanner (System.in);
        double r = s.nextDouble();
        double C = 2 * 3.14159 * r;
        double A = 3.14159 * r * r;
        System.out.println ("Circumference: " + C);
        System.out.println ("Area: " + A);
    }
}
```

```

class sum {
    public static void main (String [ ] args) {
        Scanner s = new Scanner (System.in);
        int n = s.nextInt ();
        if ((n % 5 == 0) && (n % 7 == 0)) {
            System.out.println (n + " is a multiple of 5");
        } else if ((n % 5 != 0) && (n % 7 == 0)) {
            System.out.println (n + " is a multiple of 7");
        } else if ((n % 5 == 0) && (n % 7 != 0)) {
            System.out.println (n + " is neither multiple
                                of 5 nor 7");
    }
}

```

3. John wants to use BMI to calculate the body mass index of his clients
- BMI < 18.5 - Underweight
 - BMI $\geq 18.6 \wedge$ BMI ≤ 24.9 - Normal weight
 - BMI $\geq 25.0 \wedge$ BMI ≤ 29.9 - Overweight
 - BMI > 30.0 - obese.

Soln:
Import java.util.Scanner;

class John {

```

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

```

19

```

    double h = s.nextDouble ();
    double w = s.nextDouble ();
    double bmi = w / (h * h);
    System.out.println ("BMI: " + bmi);
    if (bmi < 18.5)
        System.out.println ("Classification: Underweight");
    else if ((bmi > 18.6) && (bmi < 24.9))
        System.out.println ("Classification: Normal
                                weight");
    else if ((bmi >= 25.0) && (bmi < 29.9))
        System.out.println ("Classification: Overweight");
    else if (bmi >= 30.0)
        System.out.println ("Classification: obese");
}

```

4. Arun wants to evaluate the depreciation of his car once
time to understand its current value and categorize it
based on that value.

- High : current value $> 10,000$
- Medium: current value is b/w 5,000 & 10,000
- Low : current value < 5000

Soln:
Import java.util.Scanner;

6. Maya, wants to create a pattern using stars (*) in a specific format.

Soln:

```
import java.util.Scanner;  
class maya{  
    public static void main(String[] args){  
        Scanner s = new Scanner (System.in);  
        int n = s.nextInt();  
        for (int i=1; i<=n; i++){  
            for (int j=1; j<=i; j++){  
                System.out.print("* ");  
            }  
            System.out.println();  
        }  
        for (int i=n-1; i>=1; i--){  
            for (int j=1; j<=i; j++){  
                System.out.print("* ");  
            }  
            System.out.println();  
        }  
    }  
}
```

7. You are creating a numerical pyramid pattern. The enchanting pattern is fashioned using a for loop and is created based on user input.

Soln:

```
import java.util.Scanner;  
class num{  
    public static void main (String[] args){  
        Scanner s = new Scanner (System.in);  
        int n = s.nextInt();  
        for (int i=1; i<=n; i++){  
            for (int j=1; j<=n-i; j++){  
                System.out.print(" ");  
            }  
            for (int k=1; k<=(2*i-1); k++){  
                System.out.print(k);  
            }  
            System.out.println();  
        }  
    }  
}
```

```

class anit {
    public static void main (String[] args) {
        Scanner s = new Scanner (System.in);
        int a = s.nextInt();
        int b = s.nextInt();
        double c = 0.15;
        double d = a;
        for (int i=0; i < b; i++) {
            d = d * c;
        }
        System.out.println ("Current Value : %.2f\n", d);
        if (d > 10000) {
            System.out.println ("Category: High");
        } else if ((d > 5000) && (d < 10000)) {
            System.out.println ("Category: Medium");
        } else if (d < 5000) {
            System.out.println ("Category: Low");
        }
    }
}

```

5. Ted has accepted the challenge of writing a program that checks if the number of digits in an integer matches the sum of its digits
~~but~~

import java.util.Scanner;

```

class ted {
    public static void main (String[] args) {
        Scanner s = new Scanner (System.in);
        int n = s.nextInt();
        int temp = n;
        int sum = 0;
        int count = 0;
        do {
            int a = temp % 10;
            sum = sum + a;
            count++;
            temp = temp / 10;
        } while (temp > 0);
        if (sum == count) {
            System.out.println ("The number of digits in " +
                n + " matches the sum of its digits.");
        } else {
            System.out.println ("The number of digits in " +
                n + " does not match the sum of its digits.");
        }
    }
}

```

10. Ashu knows how many days the each workers would take the time to complete.

Soln:

```

import java.util.Scanner;
class Ashu {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int x = s.nextInt ();
        int y = s.nextInt ();
        int z = s.nextInt ();
        int d1 = s.nextInt ();
        int d2 = s.nextInt ();
        double a = 1.0/x;
        double b = 1.0/y;
        double c = 1.0/z;
        double w1 = (a+b+c)*d1;
        double w2 = (a+b)*d2;
        double rem = 1.0 - (d1+d2);
        System.out.printf ("Work done in first d1 days
(A+B+C): %.2f\n", w1);
        System.out.printf ("Work done in next d2 days
(A+B): %.2f\n", w2);
        System.out.printf ("Remaining work: %.2f\n");
    }
}

```

11. Aun is working on a project to automate the process of determining whether a student has passed or failed based on their subject marks.

Soln:

```

import java.util.Scanner;
class Student {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int a = s.nextInt ();
        int b = s.nextInt ();
        int c = s.nextInt ();
        int d = s.nextInt ();
        int e = s.nextInt ();
        int f = (a+b+c+d+e)/5;
        System.out.println ("Average score: " + f);
        if (f >= 50) {
            System.out.println ("The student has passed");
        } else {
            System.out.println ("The student has failed");
        }
    }
}

```

12. Samantha, her teacher gives her an problem to solve, which takes a no as input & checks whether it is a multiple of 5 or 7.

Soln:

```

import java.util.Scanner;

```

```

int[] a = new int[n];
for (int i=0; i<n; i++){
    a[i] = s.nextInt();
}
int num = 0;
for (int i=0; i<n; i++){
    num = a[0] + a[n-1];
}
System.out.println ("Sum of the first and last
elements: " + num);
}

```

4. Lisha ~~oo~~ wants to add temp data from two diff weather stations

Soln:

Import java.util.*;

```

class lisha {
    public static void main (String[] args){
        Scanner s = new Scanner (System.in);
        int n = s.nextInt();
        int m = s.nextInt();
        int[][] a = new int [n][m];
        int[][] b = new int [n][m];
        for (int i=0; i<n; i++){

```

```

        for (int j=0; j<m; j++){
            a[i][j] = s.nextInt();
        }
        for (int i=0; i<n; i++){
            for (int j=0; j<m; j++){
                b[i][j] = s.nextInt();
            }
        }
        for (int i=0; i<n; i++){
            for (int j=0; j<m; j++){
                System.out.printf ("%.2f ", a[i][j] + b[i][j]);
            }
        }
        System.out.println ("\n");
    }
}

```

5. Sharon wants to finds the first repeated element in ~~put~~ array.

Soln:

Import java.util.*;

```

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

```

1. Roh wants to compute the sum of the third largest and second-smallest elements from a list of integers.

Soln:

Import java.util.*;

class roh {

```
public static void main (String[] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt ();
    int [] a = new int [n];
    for (int i = 0; i < n; i++) {
        a[i] = s.nextInt ();
    }
    Arrays.sort (a);
    int sum = a[1] + a[n-3];
    System.out.println (sum);
}
```

2. Monica wants sum of main diagonal ele and secondary diagonal elements.

Soln:

Import java.util.*;

class monica {

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

```
int [] [] a = new int [n] [n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        a[i][j] = s.nextInt ();
}
```

int num = 0, dnum = 0;

```
for (int i = 0; i < n; i++) {
    num = num + a[i][i];
    dnum = a[i][n - 1 - i];
}
```

System.out.println ("sum of the main diagonal " + num);

System.out.println ("sum of the secondary diagonal " + dnum);

}

3. Task: sum of the weight of 10 last packages in the list.

Soln:

Import java.util.*;

class ware {

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

Week 5

1. Each customer booking has a Booking ID (Integer)
 - * customer name (String)
 - * distance travelled in km

The fare calculation rules are

- * Base fare = 50 units
- * Per km charge = 10 units/km
- * If distance > 20 , 10% discount is applied on total fare.

soln:

```
import java.util.*;
class citybank {
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        int n = s.nextInt();
        s.nextLine();
        for (int i = 0; i < n; i++)
        {
            int an = s.nextInt();
            s.nextLine();
            String cn = s.nextLine();
            double fb = s.nextDouble();
            double da = s.nextDouble();
            double ws = s.nextDouble();
            fb += da;
            System.out.println ("Account Number: " + an);
            System.out.println ("Customer Name: " + cn);
```

if (fb >= 50)

System.out.println ("Final Balance: " + fb);

else

fb = fb * 0.9;

System.out.println ("Final Balance: " + fb);

}

3

2. Each stu enrollment has an enrollment ID (Int)

- * stu name (String)
- * The no of sub (Int)

En fee calculation rules are:

- * Reg fee = 1000 units
- * Per sub fee = 500 units
- * If stu enrolls in more than 3 sub, 20% scholarship is applied.

soln:

```
import java.util.*;
class enroll {
    private int enrollId;
    private String stuName;
    private int sub;
    private double fe;
    public Enroll (int enrollId, String stuName, int sub)
    {
        this.enrollId = enrollId;
        this.stuName = stuName;
        this.sub = sub;
        this.fe = 1000;
    }
    public void calculateFee()
    {
        if (sub > 3)
            fe = fe - fe * 0.2;
        else
            fe = fe + sub * 500;
    }
    public void display()
    {
        System.out.println ("Enrollment ID: " + enrollId);
        System.out.println ("Student Name: " + stuName);
        System.out.println ("Number of Subjects: " + sub);
        System.out.println ("Fee: " + fe);
    }
}
```

```

else {
    for (int j=0; j < d.length(); j++) {
        char ch = d.charAt(j);
        if (((ch >='a' && ch <='z') || (ch >='A' &
            ch <='Z'))){
            e = false;
            break;
        }
    }
    if (e){
        c.append(d).append(" ");
    }
}
if (c.length() > 0){
    System.out.println(c.toString().trim());
}
else {
    System.out.println("No valid words.");
}
}
}

```

5. PIN code 1. exactly 4 digits 2. numeric (0-9)
 3. identical digits X.

Soln:

```

import java.util.*;
class customer{
    public static void main (String[] args){

```

57

```

Scanner s = new Scanner (System.in);
int n = Integer.parseInt (s.nextLine());
for (int i=0; i < n; i++){
    String a = s.nextLine();
    if (a.length() == 4 && a.matches ("^d{4}$") &&
        !(a.charAt(0) == a.charAt(1) && a.charAt(1) ==
            a.charAt(2) && a.charAt(2) == a.charAt(3))) {
        System.out.println ("YES");
    }
    else {
        System.out.println ("NO");
    }
}

```

2. Write a program that accepts at least five keywords as input arguments & outputs them in sorted alphabetical order.

Soln:

```
import java.util.*;
class arru {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int n = Integer.parseInt (s.nextLine ());
        String [] a = s.nextLine ().split (" ");
        Arrays.sort (a);
        for (int i = 0; i < n; i++)
            System.out.println (" " + a[i]);
    }
}
```

3. Bechar chacha is seeking for a valid mobile no.
- * exactly 10 digits.
 - * numeric values (0-9)
 - * does not begin with 0.

Soln:

```
import java.util.*;
class bechar {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int n = Integer.parseInt (s.nextLine ());
        for (int i = 0; i < n; i++)
```

```
String a = s.nextLine ().trim ();
if (a.length () == 10 & a.matcher ("^\\d{10}$").matches () &&
    a.charAt (0) != '0') {
    System.out.println ("YES");
} else {
    System.out.println ("NO");
}
}
}
```

4. Arjun wants to identify valid words in a sentence.

Soln:

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

```
class arjun
```

```
public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    String a = s.nextLine ().trim ();
    String [] b = a.split (" ");
    String Builder c = new String Builder ();
    for (int i = 0; i < b.length; i++) {
        String d = b[i];
        if
        boolean e = true;
        if (d.length () < 2) {
            e = false;
        }
    }
}
```

3

3. Neka *
- * customerID (int)
 - * customerName (string)
 - * UnitsConsumed (double)

↳ Solution:

```

import java.util.*;
class Neka {
    public static void main (String args[]){
        Scanner s = new Scanner (System.in);
        int n = s.nextInt();
        s.nextLine();
        for (int i = 0; i < n; i++) {
            int cID = s.nextInt();
            s.nextLine();
            String cName = s.nextLine();
            double un = s.nextDouble();
            System.out.println ("Customer ID: " + cID);
            System.out.println ("Customer Name: " + cName);
            if (un <= 100) {
                un *= 5;
            }
            else if (un > 100 & un <= 200) {
                un = 100 * 5 + (un - 100) * 7;
            }
            else {
                un = (100 * 5) + (100 * 7) + (un - 200) * 9;
            }
        }
    }
}

```

If (un > 200) {

un = un - (un * 0.05);

System.out.println ("Final Bill: " + un);

1. Elsa subscribes to a premium service with a base monthly cost, a service tax and an extra feature. Assist her in writing an inheritance program that takes input for these values & calculates the total monthly cost.

Sample test case:

Input 1 :

10.0
2.5
5.0

Output 1 :

Rs. 17.50

```
→ import java.util.Scanner;
class Subscription {
    protected double baseMonthlyCost;
    public Subscription(double baseMonthlyCost) {
        this.baseMonthlyCost = baseMonthlyCost;
    }
}
class PremiumSubscription extends Subscription {
    private double serviceTax;
    private double extraFeatureCost;
    public PremiumSubscription(double baseMonthlyCost, double serviceTax, double extraFeatureCost) {
        super(baseMonthlyCost);
        this.serviceTax = serviceTax;
        this.extraFeatureCost = extraFeatureCost;
    }
}
```

g

```
35
public double calculateMonthlyCost() {
    return baseMonthlyCost + serviceTax + extraFeatureCost;
}
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double baseMonthlyCost = scanner.nextDouble();
        double serviceTax = scanner.nextDouble();
        double extraFeatureCost = scanner.nextDouble();
        PremiumSubscription premiumSubscription = new PremiumSubscription(baseMonthlyCost, serviceTax, extraFeatureCost);
        double totalMonthlyCost = premiumSubscription.calculateMonthlyCost();
        System.out.print("Rs. " + totalMonthlyCost);
    }
}
```

```

    this . stuName = stuName;
    this . sub = sub;
    calculateFee ();
}

public void setEnrollId (int enrollId){
    this . enrollId = enrollId;
}

public void setStuName (String stuName){
    this . stuName = stuName;
}

public void setSub (int sub){
    this . sub = sub;
    calculateFee ();
}

public int getEnrollId (){
    return enrollId ;
}

public String getStuName (){
    return stuName ;
}

public int getSub (){
    return sub ;
}

public double getFee (){
    return fee;
}

```

```

private void calculateFee (){
    fee = 1000 + sub * 200;
    if (sub > 5){
        fee = fee - (sub * 25);
    }
}

class main {
    public static void main (String[] args){
        Scanner s = new Scanner (System.in);
        int n = Integer . parseInt (s.nextLine ());
        for (int i = 0; i < n; i++){
            int Id = Integer . parseInt (s.nextLine ());
            String name = s.nextLine ();
            int sub = Integer . parseInt (s.nextLine ());
            Enroll enroll = new Enroll (Id, name, sub);
            System . out . println ("Enrollment ID: " + enroll .
                getEnrollId ());
            System . out . println ("Student Name: " + enroll .
                getStuName ());
            System . out . println ("Final Fee: " + enroll .
                getFee ());
        }
    }
}

```

3. Neethi is working on a project to automate sales tax calculations for items in a store. She wants to create a program that takes the price of an item & the sales tax rate as input & calculates the final price of the item after applying the sales tax.

Formula used : Final price = price + (price * sales tax rate) / 100

⇒ Import java.util.Scanner;

class SalesTaxCalculator {

public static int calculateFinalPrice (int price,
int taxRate) {

return price + (price * taxRate / 100);

}

public static double calculateFinalPrice (double price,
double taxRate) {

return price + (price * taxRate / 100);

}

class Main {

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

int intPrice = scanner.nextInt();

int intTaxRate = scanner.nextInt();

double doublePrice = scanner.nextDouble();

double doubleTaxRate = scanner.nextDouble();

int finalPriceInt = SalesTaxCalculator.calculateFinalPrice (intPrice, intTaxRate);

double finalPriceDouble = SalesTaxCalculator.

calculateFinalPrice (doublePrice,
double TaxRate);

System.out.println (finalPriceInt);

System.out.format ("%1.2f", finalPriceDouble);

}

}

4. Mr. Kapoor wants to create a program to calculate the volume of a cuboid & cube using method overriding. Sample test case :

Input 1 :

60.0 60.0 60.0

50.0

Output 1 :

Volume of cuboid : 216000.00

Volume of cube : 125000.00

⇒ Import java.util.Scanner;

class Cuboid {

protected double length;

protected double width;

protected double height;

public Cuboid (double length, double width, double height) {

this.length = length;

```
this. width = width;  
this. height = height;
```

```
}
```

```
public double calculateVolume(){  
    return length * width * height;
```

```
}
```

```
}
```

```
class cube extends cuboid{
```

```
public cube(double side){  
    super(side, side, side);
```

```
}
```

```
public double calculateVolume(){  
    return length * length * length;
```

```
}
```

```
public class Main{
```

```
public static void main (String[] args){  
    Scanner scanner = new Scanner (System.in);  
    double cuboidLength = scanner.nextDouble();  
    double cuboidWidth = scanner.nextDouble();  
    double cuboidHeight = scanner.nextDouble();  
    cuboid cuboid = new cuboid (cuboidLength, cuboidWidth,  
        cuboidHeight);
```

```
System.out.print ("Volume of cuboid: " + cuboid.  
    calculateVolume());  
double cubeSide = scanner.nextDouble();  
cube cube = new cube (cubeSide);  
System.out.print ("Volume of cube: " + cube.  
    calculateVolume());
```

```
}
```

2. Alice is managing an online store & wants to implement a program using inheritance to calculate the selling price of products after applying discounts. Guide her by following the instructions:

1. Create a base class called Product with a public double attribute price.

2. Create a subclass called DiscountedProduct, which extends Product & includes a private double attribute discount rate. This subclass has a method called calculateSellingPrice() to determine the final selling price after applying the discount.

Formula: discounted selling price = price * (1 - discount rate)

Sample test cases:

Input 1: 50.00
Output 1: 40.00
0.20

import java.util.Scanner;

```
class Product{  
    public double initialPrice;  
    public Product(double initialPrice){  
        this.initialPrice = initialPrice;  
    }  
}
```

```
class DiscountedProduct extends Product{  
    private double discountRate;  
    public DiscountedProduct(double initialPrice, double  
    discountRate){  
        super(initialPrice);  
        this.discountRate = discountRate;  
    }  
}
```

```
3  
public double calculateSellingPrice(){  
    return initialPrice * (1 - discountRate);  
}
```

3

```
g  
class ProductTesting{
```

```
    public static void main(String[] args){  
        Scanner scanner = new Scanner(System.in);  
        double initialPrice = scanner.nextDouble();  
        double discountRate = scanner.nextDouble();  
        DiscountedProduct discountedProduct = new  
        DiscountedProduct(initialPrice, discountRate);  
        double sellingPrice = discountedProduct.calculateSellingPrice();
```

```
        if(sellingPrice > 0){  
            System.out.print("Rs. " + sellingPrice);  
        } else{  
            System.out.print("Not applicable");  
        }  
    }  
}
```

3

3

```

int [] a = new int [n];
for (int i=0; i<n; i++) {
    a[i] = s.nextInt();
}
HashSet<Integer> set = new HashSet<>();
boolean d = false;
for (int i=0; i<n; i++) {
    if (!set.add (a[i])) {
        d = true;
        System.out.print (a[i]);
        break;
    }
}
if (!d) {
    System.out.println ("No repeated element found
    in the array");
}
}
}

```

- Week 4
1. punctuation marks of interest are (comma (,)) & period (.)
 2. Question marks (?)

Soln:

```

import java.util.*;
class editors {
    public static void main (String [] args) {
        Scanner s = new Scanner (System.in);
        int n = Integer.parseInt (s.nextInt ());
        for (int i=0; i<n; i++) {
            String a = s.nextLine ();
            int c=0;
            int p=0;
            int q=0;
            for (int j=0; j<a.length(); j++) {
                char ch = a.charAt (j);
                if (ch == ',') c++;
                else if (ch == '.') p++;
                else if (ch == '?') q++;
            }
            System.out.println (c + " " + p + " " + q);
        }
    }
}

```