++
++
++

/*Code 15
ArrayIndexOutOfBoundsException:
The prominent exception which you will see is ArrayIndexOutOfBoundsException. It occurs when the program try to access the array beyond its size. As we know arrays have fixed size. So when you try to use array beyond its size it throws this exception.
Let's try to handle this exception.
Get an Array of size N and get an index, then print the Array[index]. If the index is greater or equal to array size(N), then print the Exception*/

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

class Array {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    try {
      int size = sc.nextInt();
      int[] intArray = new int[size];
      for (int i = 0; i < size; i++) {
        intArray[i] = sc.nextInt();
      }
      int index = sc.nextInt();
      System.out.println(intArray[index]);
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("Array index out of bound.");
    }
    try {
      int a = sc.nextInt();
      int b = sc.nextInt();
      int c = a / b;
      System.out.println(c);
    } catch (ArithmeticException e) {
      System.out.println(e);
    }
    try {
      String str = null;
      int index = sc.nextInt();
      System.out.println(str);
      System.out.println(str.charAt(index));
    } catch (NullPointerException n) {
      System.out.println(n);
    }
  }
}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* Code 5
Create two classes a Box class and a Main class, create an object for the Box class in the Main
class, and calculate the volume of the box.*/

import java.util.Scanner;

class Box {
  double width;
  double height;
  double depth;
}

class Main {
  public static void main(String args[]) {
    Box mybox = new Box();
    double vol;
    Scanner sc = new Scanner(System.in);
    mybox.width = sc.nextDouble();
    mybox.height = sc.nextDouble();
    mybox.depth = sc.nextDouble();
    if (mybox.width > 0 && mybox.height > 0 && mybox.depth > 0) {
      vol = mybox.width * mybox.height * mybox.depth;
      System.out.format("%.2f", vol);
    } else {
      System.out.print("Invalid");
    }
  }
}

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/*Code 19
Gokul is given a list of N integers, help him find out all the even numbers exist in the list using
Stream functions*/
```

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

class evennumbers {
  public static void main(String args[]) {
    List<Integer> arList = new ArrayList<Integer>();
    Scanner sc = new Scanner(System.in);
    int num = sc.nextInt();
    for (int i = 0; i < num; i++) {
      arList.add(sc.nextInt());
    }
    arList.stream()
        .filter(n -> n % 2 == 0)
        .forEach(System.out::println);
  }
}
```

++
++
++

```
/*Code 1
Write a program that will get the input of the event details like name of the event,type of the
event, number of people expected, a string value (Y/N) telling whether the event is going to be a
paid entry and the projected expenses (in lakhs) for the event.The program should then display
the input values as a formatted output.
*/
import java.util.*;
import java.io.*;

class Eventdetails {
  public static void main(String[] args) {
    String name, type;
    double expenses;
    int number;
    char entry;
    Scanner sc = new Scanner(System.in);
    name = sc.nextLine();
    type = sc.nextLine();
    number = sc.nextInt();
    entry = sc.next().charAt(0);
    expenses = sc.nextDouble();
    System.out.println("Event Name : " + name);
    System.out.println("Event Type : " + type);
    System.out.println("Expected Count : " + number);
    System.out.println("Paid Entry : " + entry);
    System.out.println("Projected Expense : " + expenses + "L");
  }
}
```

+++
+++
+++

```
/* Code 2
Write a program to display the grade of a student. If the mark is > 85 then display 'A', if the mark
is > 75 then display 'B', if the mark is > 65 then display 'C', for others display 'D'.
Note: Use if else statement*/

import java.util.Scanner;

class Grade {
  public static void main(String args[]) {
    int mark;
    Scanner in = new Scanner(System.in);
    mark = in.nextInt();
    if (mark >= 85) {
      System.out.print("A");
    } else if (mark >= 75) {
      System.out.print("B");
    } else if (mark >= 65) {
      System.out.print("C");
    } else if (mark >= 0) {
      System.out.print("D");
    } else {
      System.out.print("Invalid");
    }
  }
}
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
/*Code 3
Write a java program to program for half-diamond pattern printing using numbers and stars.*/
import java.util.*;
class Halfdiamond {
  public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int i,j,k,N,count=0;
    N = sc.nextInt();
    for(i=1;i<=N;i++){
      k=1;
      for(j=0;j<i;j++){
        System.out.print(i);
        if(k<i){
          System.out.print("*");
          k=k+1;
        }
      }
      System.out.println();
    }
    for(i=N;i>0;i--){
      k=1;
      for(j=0;j<i;j++){
        System.out.print(i);
        if(k<i){
          System.out.print("*");
          k=k+1;
        }
      }
      System.out.println();
    }
  }
}
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* Code 12
METHOD OVERLOADING
Create a class named 'Hello'. Define a method 'sayHello' 1. Create an object obj.
2. Call method 'sayHello' without argument, Output should display 'Hello'.
3. Call method 'sayHello' with one argument, Output should display 'Hello'argumentvalue'' (Ex: If
the argument passed is 'John' Output should display 'Hello John')*/

import java.util.*;

class Hello {
  public String sayHello() {
    return "Hello";
  }

  public String sayHello(String s) {
    return ("Hello " + s);
  }

  public static void main(String args[]) {
    Scanner in = new Scanner(System.in);
    String str = in.nextLine();
    Hello s = new Hello();
    System.out.println(s.sayHello());
    System.out.println(s.sayHello(str));
  }
}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/*Code 19
Gokul is given a list of N integers, help him find out all the even numbers exist in the list using
Stream functions*/

import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
class Integers {
 public static void main(String args[]) {
```

```
 List<Integer> arList = new ArrayList<Integer>();
 Scanner sc=new Scanner(System.in);
 int num=sc.nextInt();
 for(int i=0;i<num;i++)
 {
 arList.add(sc.nextInt());
 }
 arList.stream()
 .filter(n -> n%2 == 0)
 .forEach(System.out::println);
 }
}
```

+++
+++
+++

```
/* Code 4
Write a program to print the Lucas sequence for the given number.*/
import java.util.Scanner;
class Lucas {
  public static void main(String[] args) {
    int num;
    Scanner s = new Scanner(System.in);
    num = s.nextInt();
    int thirdLast = 0;
    int secondLast = 0;
    int last = 1;
    int current = 0;
    System.out.print(thirdLast + " " + secondLast + " " + last + " ");
    for (int i = 3; i < num; i++) {
      current = last + secondLast + thirdLast;
      System.out.print(current + " ");
      int tmp = last;
      last = current;
      thirdLast = secondLast;
      secondLast = tmp;
    }
  }
}
```

+++
+++
+++

```
//Code 8
Create abstract class marks with the following method
void getPercentage();
Create a class A that extends marks and has 3 attributes marks1, marks2, and marks3,
and a method getPercentage that calculates and prints the percentage of the student.
Create a class B that extends marks and has 4 attributes marks1, marks2, marks3, and
marks4, and a method getPercentage that calculates and prints the percentage of the
student.
Round off the output to two decimal places*/
import java.io.*;
import java.util.*;
import java.text.*;
abstract class marks {
 abstract public void getPercentage();
}
class A extends marks {
 DecimalFormat d = new DecimalFormat("0.00");
 public int marks1;
 public int marks2;
 public int marks3;
 A() {
 this.marks1=0;
 this.marks2=0;
 this.marks3=0;
 }
 A(int m1,int m2,int m3) {
 this.marks1 = m1;
 this.marks2 = m2;
 this.marks3 = m3;
 }
 public void getPercentage() {
 int total = marks1+marks2+marks3;
 double percent = (total/300.0)*100.0;
 System.out.println(d.format(percent));
 }
}
class B extends marks {
 DecimalFormat d = new DecimalFormat("0.00");
 public int marks1;
 public int marks2;
 public int marks3;
 public int marks4;
 B() {
```

```
 this.marks1=0;
 this.marks2=0;
 this.marks3=0;
 this.marks4=0;
 }
 B(int m1,int m2,int m3,int m4) {
 this.marks1 = m1;
 this.marks2 = m2;
 this.marks3 = m3;
 this.marks4 = m4;
 }
 public void getPercentage() {
 int total = marks1+marks2+marks3+marks4;
 double percent = (total/400.0)*100.0;
 System.out.println(d.format(percent));
 }
}
class Main {
 public static void main(String args[]) {
 A a = new A();
 Scanner sc = new Scanner(System.in);
 a.marks1 = sc.nextInt();
 a.marks2 = sc.nextInt();
 a.marks3 = sc.nextInt();
 a.getPercentage();
 B b = new B();
 b.marks1 = sc.nextInt();
 b.marks2 = sc.nextInt();
 b.marks3 = sc.nextInt();
 b.marks4 = sc.nextInt();
 b.getPercentage();
 }
}
```

+++
+++
+++

/*Code 9

Write a program to count a minimum number of front moves to sort an array.
Note: Create an interface and declare a method, the class should implement the interface.*/

```
import java.io.*;
import java.util.Scanner;

interface Move {
  int minmoves(int arr[], int n);
}

class Main implements Move {
  public int minmoves(int arr[], int n) {
            int expectedItem = n;
            for (int i = n - 1; i >= 0; i--) {
            if (arr[i] == expectedItem)
            expectedItem--;
            }
            System.out.print(expectedItem);
            return expectedItem;
  }

  public static void main(String[] args) {
            Scanner in = new Scanner(System.in);
            int n;
            n = in.nextInt();
            int arr[] = new int[n];
            for (int i = 0; i < n; i++) {
            arr[i] = in.nextInt();
            }
            Main obj = new Main();
            obj.minmoves(arr, n);
  }
}
```

++
++
++

/* Code 16
Write a program to implement multithreading that extends the Object class and implements the Runnable interface.*/

```
import java.util.Scanner;
class MultithreadingTest implements Runnable {
 public void run()
 {
 try {
```

```
 System.out.println(
 "Thread " + Thread.currentThread().getId()
 + " is running");
 }
 catch (Exception e) {
 System.out.println("Exception has occurred and is caught");
 }
 }
}
class Multithreading {
 public static void main(String[] args)
 {

 Scanner sc =new Scanner(System.in);
int n=Integer.parseInt(sc.nextLine());
 for (int i = 0; i < n; i++) {
 Thread obj = new Thread(new MultithreadingTest());
 obj.start();
 }
 }
}
```

+++
+++
+++

/*Code 17
Write a program to implement multithreading using the join(), isAlive() and sleep() methods.*/

```
import java.util.Scanner;

class A implements Runnable {
  String name;
  Thread t;

  A(String tname) {
          name = tname;
          t = new Thread(this, name);
          t.start();
  }

  public void run() {
          try {
          Scanner sc = new Scanner(System.in);
          int n = Integer.parseInt(sc.nextLine());
          for (int i = 1; i <= n; i++) {
          System.out.println(name + " = " + i);
          Thread.sleep(300);
          }
          System.out.println("END OF = " + name);
          } catch (Exception e) {
          }
  }
}

class multithreading {
  public static void main(String args[]) {
          A nt1 = new A("Thread");
          System.out.println("Thread is alive :: " + nt1.t.isAlive());
          try {
          nt1.t.join();
          } catch (Exception e) {
          }
          System.out.println("Thread is alive = " + nt1.t.isAlive());
          System.out.println("End of main.");
  }
}
```

++
++
++

/*Code 14
NullPointerException
Another prominent exception is NullPointerException. It occurs when you try to access a null value. Assign null value to a string and obtain an index position and try to access it. Print the exception.*/

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

class NullPointer {
  public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          try {
          String str = null;
          int index = Integer.parseInt(sc.nextLine());
          System.out.println(str);
```

```
        System.out.println(str.charAt(index));
        } catch (NullPointerException n) {
        System.out.println(n);
        }
  }
}
```

++
++
++

/*Code 18
Athiyan is having a list of integers, your task is to help him find out all the numbers starting with 1 using Stream functions*/

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

class number {
  public static void main(String[] args) {
        List<String> arList = new ArrayList<String>();
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        for (int i = 0; i < num; i++) {
        arList.add(sc.next());
        }
        int n = sc.nextInt();
        int ans = arList.stream()
        .filter(s -> s.length() > n)
        .mapToInt(s -> s.length())
        .sum();
        System.out.println(ans);
  }
}
```

++
++
++

```
/*Code 20
Write a program such that it should have three classes, Person and Student and the Main class
where Person is the base class and Student is the derived class.
Student class should have the following:
A Student class constructor, which has parameters:
A string, firstName.
A string, lastName.
An integer, id.
An integer array of test scores.
A char calculate() method that calculates a Student object's average and returns the grade
character representative of their calculated average:*/

import java.io.*;
import java.util.*;
import java.util.Scanner;

class Person {
  protected String firstName;
  protected String lastName;
  protected int idNumber;

  Person(String firstName, String lastName, int identification) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.idNumber = identification;
  }

  public void printPerson() {
        System.out.println(
        "Name: " + lastName + " " + firstName
        + "\nID: " + idNumber);
  }
}

class Student extends Person {
  private int[] testScores;

  public Student(String firstName, String lastName, int id, int[] testScores) {
        super(firstName, lastName, id);
        this.testScores = testScores;
  }

  char calculate() {
        int i, sum = 0, avg;
        char grade;
        for (i = 0; i < testScores.length; i++)
        sum += testScores[i];
        avg = sum / testScores.length;
        if (avg <= 100 && avg >= 90)
        grade = 'O';
        else if (avg >= 80 && avg < 90)
        grade = 'E';
        else if (avg >= 70 && avg < 80)
        grade = 'A';
        else if (avg >= 55 && avg < 70)
        grade = 'P';
        else if (avg >= 40 && avg < 55)
        grade = 'D';
        else
        grade = 'T';
        return grade;
  }
}

class Main {
  public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String firstName = scan.next();
        String lastName = scan.next();
        int id = scan.nextInt();
        int numScores = scan.nextInt();
        int[] testScores = new int[numScores];
        for (int i = 0; i < numScores; i++) {
        testScores[i] = scan.nextInt();
        }
        scan.close();
        Student s = new Student(firstName, lastName, id, testScores);
        s.printPerson();
        System.out.println("Grade: " + s.calculate());
  }
}
```

++
++
++

/*Code 7
Create an abstract class Shape with the following methods
abstract void rectangleArea();
abstract void squareArea();
abstract void circleArea();
Create a class Area that extends Shape that calculates and prints all the area.
Create a Main class, get the inputs, and pass it to the methods.*/

```
import java.io.*;
import java.lang.Math.*;
import java.util.*;
import java.text.*;

abstract class Shape {
  abstract void rectangleArea(int l, int b);

  abstract void squareArea(int s);

  abstract void circleArea(int r);
}

class Area extends Shape {
  public void rectangleArea(int l, int b) {
          System.out.println(l * b);
  }

  public void squareArea(int s) {
          System.out.println(s * s);
  }

  public void circleArea(int r) {
          DecimalFormat d = new DecimalFormat("0.00");
          double res = Math.PI * r * r;
          System.out.println(d.format(res));
  }
}

class Main {
  public static void main(String[] args) {
          int length, breadth, radius, side;
          Scanner sc = new Scanner(System.in);
          length = sc.nextInt();
          breadth = sc.nextInt();
          side = sc.nextInt();
          radius = sc.nextInt();
          Area a = new Area();
          a.rectangleArea(length, breadth);
          a.squareArea(side);
          a.circleArea(radius);
  }
}
```

+++
+++
+++

```
/*Code 10
Create an interface ShapeCalculator with the following method
void calc(int n)
Create two classes Square and Circle that implement the interface and implement the method in
it. Calculate the area and perimeter of both squares and circles.*/

import java.util.*;
import java.math.*;

interface ShapeCalculator {
  void calc(int n);
}

class Square implements ShapeCalculator {
  int ar, pr;

  public void calc(int s) {
          ar = s * s;
          pr = 4 * s;
          System.out.print(ar + " " + pr + "\n");
  }
}

class Circle implements ShapeCalculator {
  double ar, pr;
  double pi = 3.14;

  public void calc(int r) {
          ar = Math.round(pi * r * r * 100.0) / 100.0;
          pr = Math.round(2 * pi * r * 100.0) / 100.0;
          System.out.println(ar + " " + pr);
  }
}

class CalcMain {
  public static void main(String args[]) {
          int n;
          Scanner in = new Scanner(System.in);
          n = in.nextInt();
          Square sq = new Square();
          sq.calc(n);
          Circle cr = new Circle();
          cr.calc(n);
  }
}
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

/*Code 13
Write a Java program to implement constructor overloading.
Create a student class with properties like name, dept, and college.
Create a parametrized constructor such that it takes all 3 values to create a student object or just name and dept to create a student object.
Use appropriate constructors for object creation based on the data provided and print the student details.*/

```
import java.util.*;

class Student {
  String name;
  String dept;
  String college;

  public Student(String name, String dept, String college) {
          this.name = name;
          this.dept = dept;
          this.college = college;
  }

  public Student(String name, String dept) {
          this.name = name;
          this.dept = dept;
  }

  public void printStudentDetails() {
          System.out.println("Name : " + this.name);
          System.out.println("Dept : " + this.dept);
          System.out.println("College : " + this.college);
  }
}

class Test {
  public static void main(String args[]) {
          Scanner scan = new Scanner(System.in);
          int n = scan.nextInt();
          scan.nextLine();
          Student s1;
          if (n == 2) {
          String name = scan.nextLine();
          String dept = scan.nextLine();
          s1 = new Student(name, dept);
          s1.printStudentDetails();
          } else if (n == 3) {
          String name = scan.nextLine();
          String dept = scan.nextLine();
          String college = scan.nextLine();
          s1 = new Student(name, dept, college);
          s1.printStudentDetails();
          }

          scan.close();
  }
}
```

++
++
++

/*Code 6
Write a program to check whether the given character is a vowel or consonant. Create two methods namely main method and alph. Create an object in the main method and access the alph method, that performs the above operation.*/

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

```
class vowelcons {
  void alph(char ch) {
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
        System.out.println(ch + " :vowel");
        else
        System.out.println(ch + " :consonant");
  }

  public static void main(String[] args) {
        char ch;
        Scanner in = new Scanner(System.in);
        ch = in.next().charAt(0);
        Main obj = new Main();
        obj.alph(ch);
  }
}
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

/*Code 11
Create multilevel inheritance with the below classes.
Create data members and method as mentioned below.
WorkerDetail
1. employee code and basic salary as integer
2. name as string
3. methods to display above details and to calculate HRA(60% of basic salary)OfficerSal
1. method to calculate DA(98% of basic salary)ManagerSal
1. method to calculate City allowance (20% of basic salary)
2. method to calculate Gross pay (basic salary + HRA + DA + City allowance)
Create Main class to get the employee ID , Name and Basic salary as input. Calculate HRA, DA, CA and Gross Salary.
*/

```
import java.util.Scanner;

class WorkerDetail {
  int c, s;
  String n;
  float h;

  void setSalary(int x, String y, int z) {
        c = x;
```

```
          n = y;
          s = z;
  }

  void showDetail() {
          System.out.println("Code :" + c);
          System.out.println("Name : " + n);
          System.out.println("Salary " + s);
  }

  void getHra() {
          h = (float) s * 60 / 100;
          System.out.println("HRA :" + h);
  }
}

class OfficerSal extends WorkerDetail {
  float d;

  void getDA() {
          d = (float) s * 98 / 100;
          System.out.println("DA :" + d);
  }
}

class ManagerSal extends OfficerSal {
  float ca, g;

  void getCA() {
          ca = (float) s * 20 / 100;
          System.out.println("City Allowance :" + ca);
  }

  void getgross() {
          g = s + h + d + ca;
          System.out.println("Gross Salary :" + g);
  }
}

class MultilevelMain {
  public static void main(String args[]){
          ManagerSal m=new ManagerSal();
          Scanner myObj = new Scanner(System.in);
          int code = Integer.parseInt(myObj.nextLine());
```

```
        String name =myObj.nextLine();
        int sal =Integer.parseInt(myObj.nextLine());
        m.setSalary(code,name,sal);
        m.showDetail();
        m.getHra();
        m.getDA();
        m.getCA();
        m.getgross();
  }
}
```

++
++
++