JAVA PROGRAMMING
THEORY DIGITAL ASSIGNMENT

Anamaya Vyas
19BCE0568

++

Arrays

Q1) government started vaccination process and made 10 age groups such as (0-10), (11-20)….(90-infinity). In a city the number of patients of every age group are stored in an array as A1,A2,A3….A10, Every day the authorities can vaccinate only k number of people starting from the last group, if the vaccination process of less than k people is completed the group then the authorities will vaccinate the next group. Raghu is in 'T' group and wants to get vaccinated, in how many days will he get vaccinated.

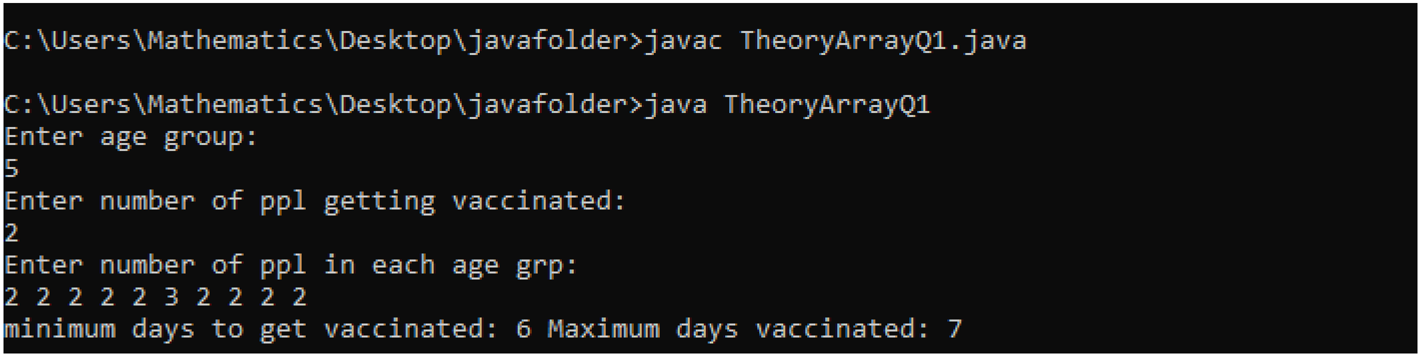
Code

```
import java.lang.*;
import java.io.*;
public class TheoryArrayQ1
{
   static class Reader {
      final private int BUFFER_SIZE = 1 << 16;
      private DataInputStream din;
      private byte[] buffer;
      private int bufferPointer, bytesRead;

      public Reader()
      {
         din = new DataInputStream(System.in);
         buffer = new byte[BUFFER_SIZE];
         bufferPointer = bytesRead = 0;
      }

      public Reader(String file_name) throws IOException
      {
         din = new DataInputStream(
               new FileInputStream(file_name));
         buffer = new byte[BUFFER_SIZE];
         bufferPointer = bytesRead = 0;
      }
```

```
public String readLine() throws IOException
{
    byte[] buf = new byte[64]; // line length
    int cnt = 0, c;
    while ((c = read()) != -1) {
        if (c == '\n') {
            if (cnt != 0) {
                break;
            }
            else {
                continue;
            }
        }
        buf[cnt++] = (byte)c;
    }
    return new String(buf, 0, cnt);
}

public int nextInt() throws IOException
{
    int ret = 0;
    byte c = read();
    while (c <= ' ') {
        c = read();
    }
    boolean neg = (c == '-');
    if (neg)
        c = read();
    do {
        ret = ret * 10 + c - '0';
    } while ((c = read()) >= '0' && c <= '9');

    if (neg)
        return -ret;
    return ret;
}

public long nextLong() throws IOException
{
    long ret = 0;
```

```
        byte c = read();
        while (c <= ' ')
            c = read();
        boolean neg = (c == '-');
        if (neg)
            c = read();
        do {
            ret = ret * 10 + c - '0';
        } while ((c = read()) >= '0' && c <= '9');
        if (neg)
            return -ret;
        return ret;
    }

    public double nextDouble() throws IOException
    {
        double ret = 0, div = 1;
        byte c = read();
        while (c <= ' ')
            c = read();
        boolean neg = (c == '-');
        if (neg)
            c = read();

        do {
            ret = ret * 10 + c - '0';
        } while ((c = read()) >= '0' && c <= '9');

        if (c == '.') {
            while ((c = read()) >= '0' && c <= '9') {
                ret += (c - '0') / (div *= 10);
            }
        }

        if (neg)
            return -ret;
        return ret;
    }

    private void fillBuffer() throws IOException
    {
        bytesRead = din.read(buffer, bufferPointer = 0,
                BUFFER_SIZE);
        if (bytesRead == -1)
            buffer[0] = -1;
    }

    private byte read() throws IOException
    {
        if (bufferPointer == bytesRead)
            fillBuffer();
        return buffer[bufferPointer++];
    }

    public void close() throws IOException
    {
        if (din == null)
            return;
        din.close();
    }
}

public static void main (String[] args) throws java.lang.Exception
{

    Reader sc = new Reader();
        System.out.println("Enter age group:");
        int G=sc.nextInt();
        System.out.println("Enter number of ppl getting vaccinated:");
        int P=sc.nextInt();
        int X[]=new int[10];
        G=G-1;
        int sum=0;
        System.out.println("Enter number of ppl in each age grp:");
        for(int i=0;i<10;i++) {
            X[i]=sc.nextInt();
            if(i>G) {
                sum+=X[i];
            }
        }
        int count=0;
        count=count+sum/P;
        sum=sum%P;


        sum=sum+X[G];
        int count1=0;
        if(sum>=P) {
           count1=count+1;
        }

        count=count+sum/P;
        sum=sum%P;

        if(sum<P && count1==0) {
           count++;
           System.out.println("minimum days :"+ count);
        }else if(sum==0) {
           System.out.println("minimum days : "+count1+" Maximum days: "+count);
        }else {
           count++;
           System.out.println("minimum days : "+count1+" Maximum days: "+count);
        }
    }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryArrayQ1.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryArrayQ1
Enter age group:
5
Enter number of ppl getting vaccinated:
2
Enter number of ppl in each age grp:
2 2 2 2 2 3 2 2 2 2
minimum days to get vaccinated: 6 Maximum days vaccinated: 7
```

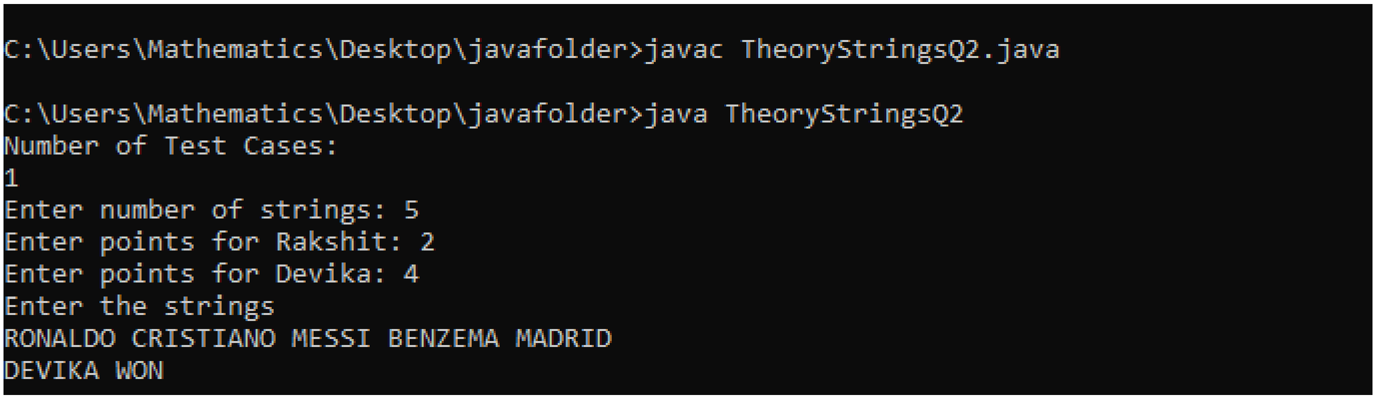
Strings

Q2)Rakshit and Devika played a game, in which they took n strings, if the string starts with a letter from one of the letter in the word 'UNITED' Rakshit gets A points and Devika gets B points. Find who won the game.

Code

```
import java.util.*;
import java.lang.*;
public class TheoryStringsQ2
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Number of Test Cases: ");
        if(sc.hasNext()){
            int t = sc.nextInt();
            while(t-->0){
                System.out.print("Enter number of strings: ");
                String check = "UNITED";
                int N = sc.nextInt();
                System.out.print("Enter points for Rakshit: ");
                long A = sc.nextLong();
                System.out.print("Enter points for Devika: ");
                long B = sc.nextLong();
                long rakshit=0,devika=0;
                System.out.println("Enter the strings ");
                for(int i=1;i<=N;i++){
                    String s = sc.next();
                    String ch =""+ s.charAt(0);
                    if(check.contains(ch))
                        rakshit = devika+A;
                    else
                        devika = devika+B;
                }
                if(rakshit>devika)
                    System.out.println("RAKSHIT WON");
                else if(devika>rakshit)
                    System.out.println("DEVIKA WON");
                else
                    System.out.println("DRAW");
            }
        }
    }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryStringsQ2.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryStringsQ2
Number of Test Cases:
1
Enter number of strings: 5
Enter points for Rakshit: 2
Enter points for Devika: 4
Enter the strings
RONALDO CRISTIANO MESSI BENZEMA MADRID
DEVIKA WON
```

Inheritence

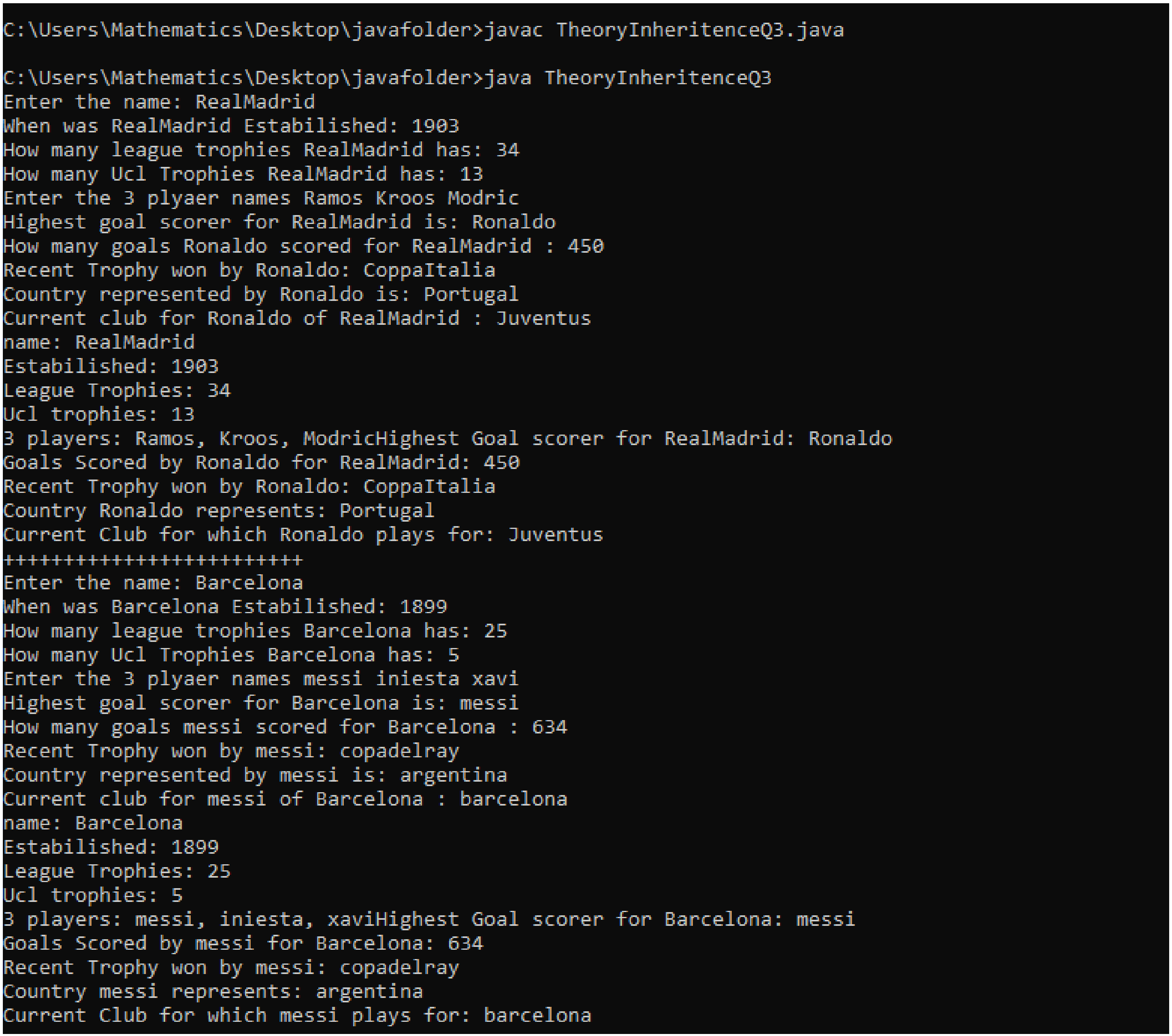
Q3) Write a java program and write define a class club with appropriate methods ‘estabilished’, ‘league_trophies’ and ‘ucl_trophies’ make two subclasses and include methods like ‘highest_goal_scorer’, ‘last_trophie_won’ etc. implement the above program using inheritance.

Code

```
import java.util.Scanner;

class club{
   static Scanner sc = new Scanner(System.in);
   String name;
   int estb;
   int leagueTrophy;
   int UclTrophy;
   void GetBasics()
   {
      System.out.print("Enter the name: ");
      name = sc.next();
      System.out.print("When was "+name+ " Estabilished: ");
      estb =sc.nextInt();
      System.out.print("How many league trophies "+name+" has: ");
      leagueTrophy = sc.nextInt();
      System.out.print("How many Ucl Trophies "+name+" has: ");
      UclTrophy = sc.nextInt();
   }
}
class Player extends club{
   String p1,p2,p3;
   String highest_goal_scorer;
   int no_of_goals;
   void Getplayer()
   {
      System.out.print("Enter the 3 plyaer names ");
      p1 = sc.next();
      p2 =sc.next();
      p3 = sc.next();
      System.out.print("Highest goal scorer for "+name+ " is: ");
      highest_goal_scorer =sc.next();
      System.out.print("How many goals "+highest_goal_scorer+" scored for "+name+" : ");
      no_of_goals = sc.nextInt();
   }
}

class Recent extends Player{
   String Recent_trophy;
   String Country;
   String Current_club;
   void Getdetails()
   {
      System.out.print("Recent Trophy won by "+highest_goal_scorer+": ");
      Recent_trophy = sc.next();
      System.out.print("Country represented by "+highest_goal_scorer+ " is: ");
      Country =sc.next();
      System.out.print("Current club for "+highest_goal_scorer+" of "+name+" : ");
      Current_club = sc.next();
   }

   void disp()
   {
      System.out.println("name: "+name);
      System.out.println("Estabilished: "+estb);
      System.out.println("League Trophies: "+leagueTrophy);
      System.out.println("Ucl trophies: "+UclTrophy);
      System.out.print("3 players: "+p1+", "+p2+", "+p3);
      System.out.println("Highest Goal scorer for "+name+": "+highest_goal_scorer);
      System.out.println("Goals Scored by "+highest_goal_scorer+" for "+name+":
"+no_of_goals);
      System.out.println("Recent Trophy won by "+highest_goal_scorer+": "+Recent_trophy);
      System.out.println("Country "+highest_goal_scorer+" represents: "+Country);
      System.out.println("Current Club for which "+highest_goal_scorer+" plays for:
"+Current_club);
```

```
    }
}


public class TheoryInheritenceQ3 {
    public static void main(String[] args) {
        Recent a1 = new Recent();
        Recent a2 = new Recent();
        a1.GetBasics();
        a1.Getplayer();
        a1.Getdetails();
        a1.disp();
        System.out.println("+++++++++++++++++++++++++");
        a2.GetBasics();
        a2.Getplayer();
        a2.Getdetails();
        a2.disp();
    }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryInheritenceQ3.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryInheritenceQ3
Enter the name: RealMadrid
When was RealMadrid Estabilished: 1903
How many league trophies RealMadrid has: 34
How many Ucl Trophies RealMadrid has: 13
Enter the 3 plyaer names Ramos Kroos Modric
Highest goal scorer for RealMadrid is: Ronaldo
How many goals Ronaldo scored for RealMadrid : 450
Recent Trophy won by Ronaldo: CoppaItalia
Country represented by Ronaldo is: Portugal
Current club for Ronaldo of RealMadrid : Juventus
name: RealMadrid
Estabilished: 1903
League Trophies: 34
Ucl trophies: 13
3 players: Ramos, Kroos, ModricHighest Goal scorer for RealMadrid: Ronaldo
Goals Scored by Ronaldo for RealMadrid: 450
Recent Trophy won by Ronaldo: CoppaItalia
Country Ronaldo represents: Portugal
Current Club for which Ronaldo plays for: Juventus
+++++++++++++++++++++++++
Enter the name: Barcelona
When was Barcelona Estabilished: 1899
How many league trophies Barcelona has: 25
How many Ucl Trophies Barcelona has: 5
Enter the 3 plyaer names messi iniesta xavi
Highest goal scorer for Barcelona is: messi
How many goals messi scored for Barcelona : 634
Recent Trophy won by messi: copadelray
Country represented by messi is: argentina
Current club for messi of Barcelona : barcelona
name: Barcelona
Estabilished: 1899
League Trophies: 25
Ucl trophies: 5
3 players: messi, iniesta, xaviHighest Goal scorer for Barcelona: messi
Goals Scored by messi for Barcelona: 634
Recent Trophy won by messi: copadelray
Country messi represents: argentina
Current Club for which messi plays for: barcelona
```

Interface and Abstract Class

Q4)Create an abstract class “CristianoRonaldo” and write appropriate methods for goals, freekicks, penalties and games in classes ManUtd, RealMadrid and Juventus.

Code

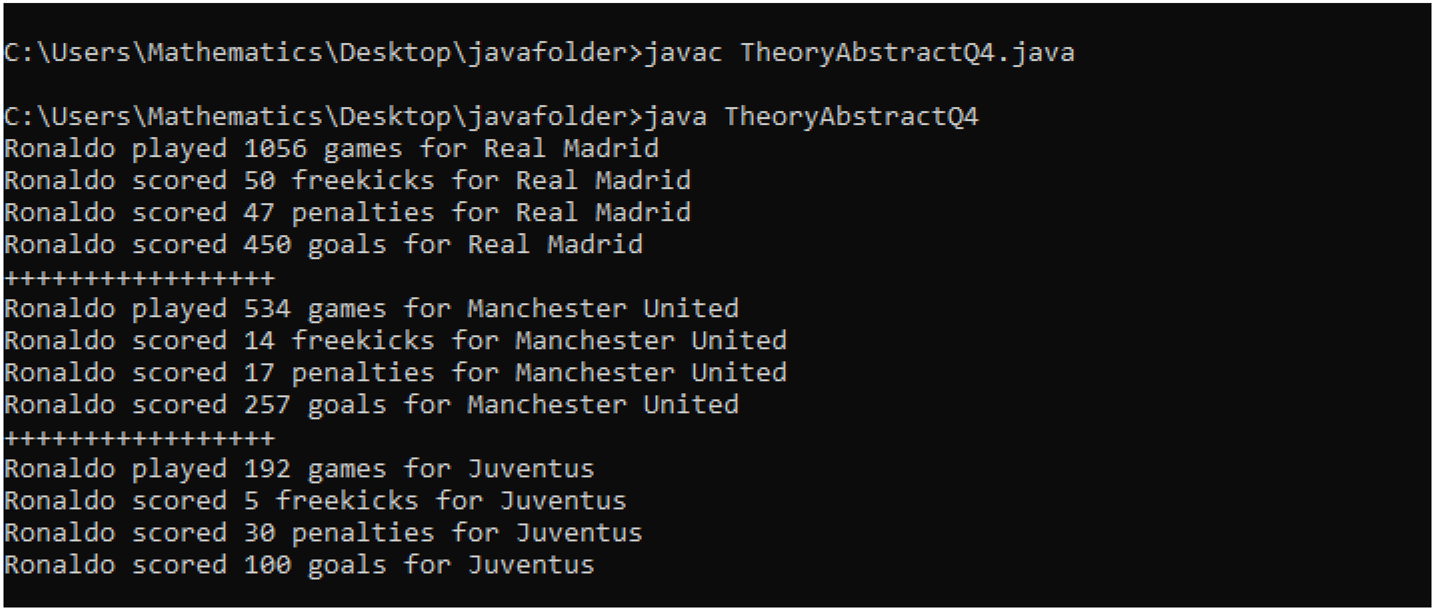
```
abstract class CristianoRonaldo{
   abstract void goals();
   abstract void freekicks();
   abstract void penalties();
   abstract void games();

}
```

```
class RealMadrid extends  CristianoRonaldo{
   int a,b,c,d;
   RealMadrid(int a, int b,int c,int d){
      this.a =a;
      this.b =b;
      this.c =c;
      this.d=d;
   }
   void goals(){
      System.out.println("Ronaldo scored "+a+" goals for Real Madrid");
   }
   void freekicks(){
      System.out.println("Ronaldo scored "+b+" freekicks for Real Madrid");
   }
   void penalties(){
      System.out.println("Ronaldo scored "+c+" penalties for Real Madrid");
   }
   void games(){
      System.out.println("Ronaldo played "+d+" games for Real Madrid");
   }
}
class ManUtd extends  CristianoRonaldo{
   int a,b,c,d;
  ManUtd(int a, int b,int c,int d){
      this.a =a;
      this.b =b;
      this.c =c;
      this.d=d;
   }
   void goals(){
      System.out.println("Ronaldo scored "+a+" goals for Manchester United");
   }
   void freekicks(){
      System.out.println("Ronaldo scored "+b+" freekicks for Manchester United");
   }
   void penalties(){
      System.out.println("Ronaldo scored "+c+" penalties for Manchester United");
   }
   void games(){
      System.out.println("Ronaldo played "+d+" games for Manchester United");
```

```
    }
}
class Juventus extends  CristianoRonaldo{
    int a,b,c,d;
    Juventus(int a, int b,int c,int d){
        this.a =a;
        this.b =b;
        this.c =c;
        this.d=d;
    }
    void goals(){
        System.out.println("Ronaldo scored "+a+" goals for Juventus");
    }
    void freekicks(){
        System.out.println("Ronaldo scored "+b+" freekicks for Juventus");
    }
    void penalties(){
        System.out.println("Ronaldo scored "+c+" penalties for Juventus");
    }
    void games(){
        System.out.println("Ronaldo played "+d+" games for Juventus");
    }
}


public class TheoryAbstractQ4 {
    public static void main(String[] args) {
        RealMadrid a1 =new RealMadrid(450, 50, 47, 1056);
        a1.games();
        a1.freekicks();
        a1.penalties();
        a1.goals();
        System.out.println("++++++++++++++++");
        ManUtd a2 =new ManUtd(257, 14, 17, 534);
        a2.games();
        a2.freekicks();
        a2.penalties();
        a2.goals();
        System.out.println("++++++++++++++++");
        Juventus a3 =new Juventus(100, 5, 30, 192);
        a3.games();
```

```
      a3.freekicks();
      a3.penalties();
      a3.goals();
   }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryAbstractQ4.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryAbstractQ4
Ronaldo played 1056 games for Real Madrid
Ronaldo scored 50 freekicks for Real Madrid
Ronaldo scored 47 penalties for Real Madrid
Ronaldo scored 450 goals for Real Madrid
+++++++++++++++++
Ronaldo played 534 games for Manchester United
Ronaldo scored 14 freekicks for Manchester United
Ronaldo scored 17 penalties for Manchester United
Ronaldo scored 257 goals for Manchester United
+++++++++++++++++
Ronaldo played 192 games for Juventus
Ronaldo scored 5 freekicks for Juventus
Ronaldo scored 30 penalties for Juventus
Ronaldo scored 100 goals for Juventus
```

Packages

Q5) Create a package “Real Madrid” which will check for the number of vowels in a string, create a subpackage “HalaMadrid” in “RealMadrid” to check if the given string is a palindrome or not. Execute this program which will import the suitable packages.

Code

//Vowel.java

```
package RM;
public class Vowel {
   public static int CountVowel(String a)
   {
      int len = a.length();
      int count =0;
      for(int i=0;i<len;i++)
      {
```

```
      if(a.charAt(i) == 'a' || a.charAt(i) == 'e' || a.charAt(i) == 'i' || a.charAt(i) == 'o' ||
a.charAt(i) == 'u')
      {
        count++;
      }
    }
    return count;
  }

  public static void main(String[] args) {

  }
}
```

//Pallindrome.java

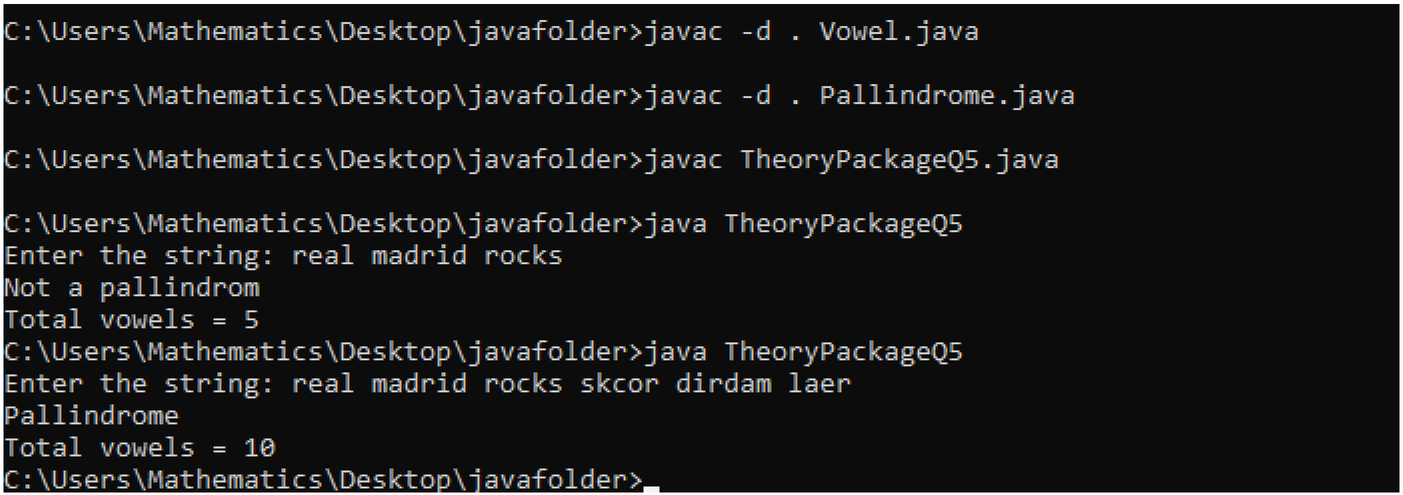
```
package RM.HalaMadrid;
public class Pallindrome {
 public static int Palicheck(String a){
    int len = a.length();
    int flag =0;
    for(int i=0;i<len/2;i++)
    {
      if(a.charAt(i) != a.charAt(len-1-i)) {
        flag = 1;
        return flag;
      }
    }
    return flag;
  }

  public static void main(String[] args) {

  }
}
```

//TheoryPackageQ5.java

```
import RM.Vowel;
import RM.HalaMadrid.Pallindrome;
import java.util.Scanner;

public class TheoryPackageQ5
```

```
{
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      System.out.print("Enter the string: ");
      String a = sc.nextLine();
      RM.Vowel a1 = new RM.Vowel();
      RM.HalaMadrid.Pallindrome a2 = new RM.HalaMadrid.Pallindrome();
      int l = a1.CountVowel(a);
      int m = a2.Palicheck(a);
      if(m == 1){
         System.out.println("Not a pallindrom");
      }
      else {
         System.out.println("Pallindrome");
      }
      System.out.print("Total vowels = "+l);


   }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac -d . Vowel.java

C:\Users\Mathematics\Desktop\javafolder>javac -d . Pallindrome.java

C:\Users\Mathematics\Desktop\javafolder>javac TheoryPackageQ5.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryPackageQ5
Enter the string: real madrid rocks
Not a pallindrom
Total vowels = 5
C:\Users\Mathematics\Desktop\javafolder>java TheoryPackageQ5
Enter the string: real madrid rocks skcor dirdam laer
Pallindrome
Total vowels = 10
C:\Users\Mathematics\Desktop\javafolder>_
```

User Defined Exception

Q6)Rahul prepared for campus placement and to get a job in Microsoft he must fulfil the following demands:-

- **a) Cgpa above 8.5**
- **b) More than 80% in competitive coding round**
- **c) More than 90% in Data structure and Algorithm round**

d) No Backs in any semester

e) Interview cleared

If Rahul does not meet the above conditions then the program should throw an exception with the reason why he cannot get the job.

Code

//EXC.java

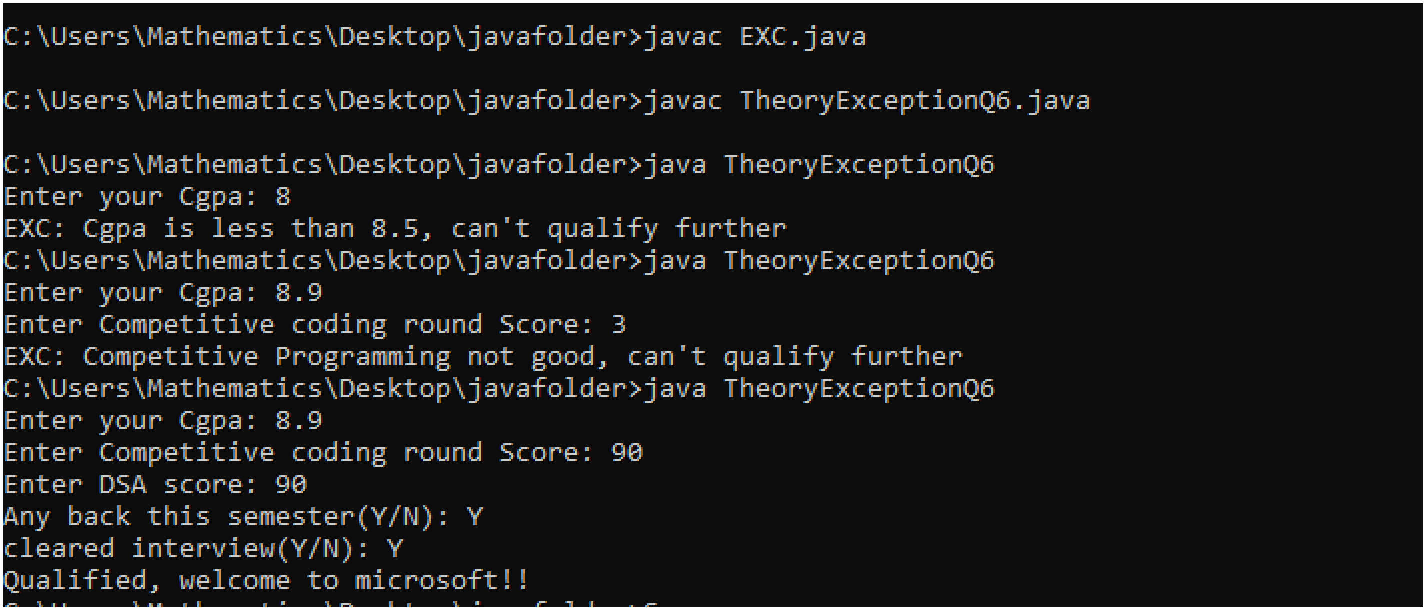
```
public class EXC extends Exception{
    public EXC(String s){
        super(s);
    }
}
```

//TheoryExceptionQ6.java

```
import java.util.Scanner;

public class TheoryExceptionQ6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try{
            System.out.print("Enter your Cgpa: ");
            double a = sc.nextDouble();
            if(a<8.5)
            {
                throw new EXC("Cgpa is less than 8.5, can't qualify further");
            }
            else {
                try{
                    System.out.print("Enter Competitive coding round Score: ");
                    double cp = sc.nextDouble();
                    if(cp<80){
                        throw new EXC("Competitive Programming not good, can't qualify further");
                    }
                    else {
                        try {
                            System.out.print("Enter DSA score: ");
                            double dsa = sc.nextDouble();
                            if (dsa < 90) {
                                throw new EXC("DSA not goo, can't qualify further");
                            }
                            else {
                                try {

                                    System.out.print("Any back this semester(Y/N): ");
                                    String ba = sc.next();
                                    if (ba == "Y" ) {
                                        throw new EXC("Cant accent students with back, can't qualify further");
                                    }
                                    else{
                                        try {

                                            System.out.print("cleared interview(Y/N): ");
                                            String in = sc.next();
                                            if (in == "N" ) {
                                                throw new EXC("Cant accent students who didn't clear interview,
can't qualify further");
                                            }
                                            else{
                                                System.out.print("Qualified, welcome to microsoft!!");
                                            }
                                        }
                                        catch (EXC e)
                                        {
                                            System.out.print(e);
                                        }
                                    }
                                }
                                catch (EXC e)
                                {
                                    System.out.print(e);
                                }
                            }
                        }
                        catch (EXC e)
                        {
                            System.out.print(e);
                        }
                    }
                }
                catch (EXC e)
                {
```

```
                System.out.print(e);
            }
        }
    }
    catch (EXC e)
    {
        System.out.print(e);
    }
  }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac EXC.java

C:\Users\Mathematics\Desktop\javafolder>javac TheoryExceptionQ6.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryExceptionQ6
Enter your Cgpa: 8
EXC: Cgpa is less than 8.5, can't qualify further
C:\Users\Mathematics\Desktop\javafolder>java TheoryExceptionQ6
Enter your Cgpa: 8.9
Enter Competitive coding round Score: 3
EXC: Competitive Programming not good, can't qualify further
C:\Users\Mathematics\Desktop\javafolder>java TheoryExceptionQ6
Enter your Cgpa: 8.9
Enter Competitive coding round Score: 90
Enter DSA score: 90
Any back this semester(Y/N): Y
cleared interview(Y/N): Y
Qualified, welcome to microsoft!!
```

Multithreading

Q7) Writa a java programme that created two threads to print even numbers till 24. After even numbers till 24 have been printed then create a new thread that prints odd numbers from 1 to 20. Print the numbers in such a way that it is visible that which thread is printing those numbers, give thread ID and thread name.

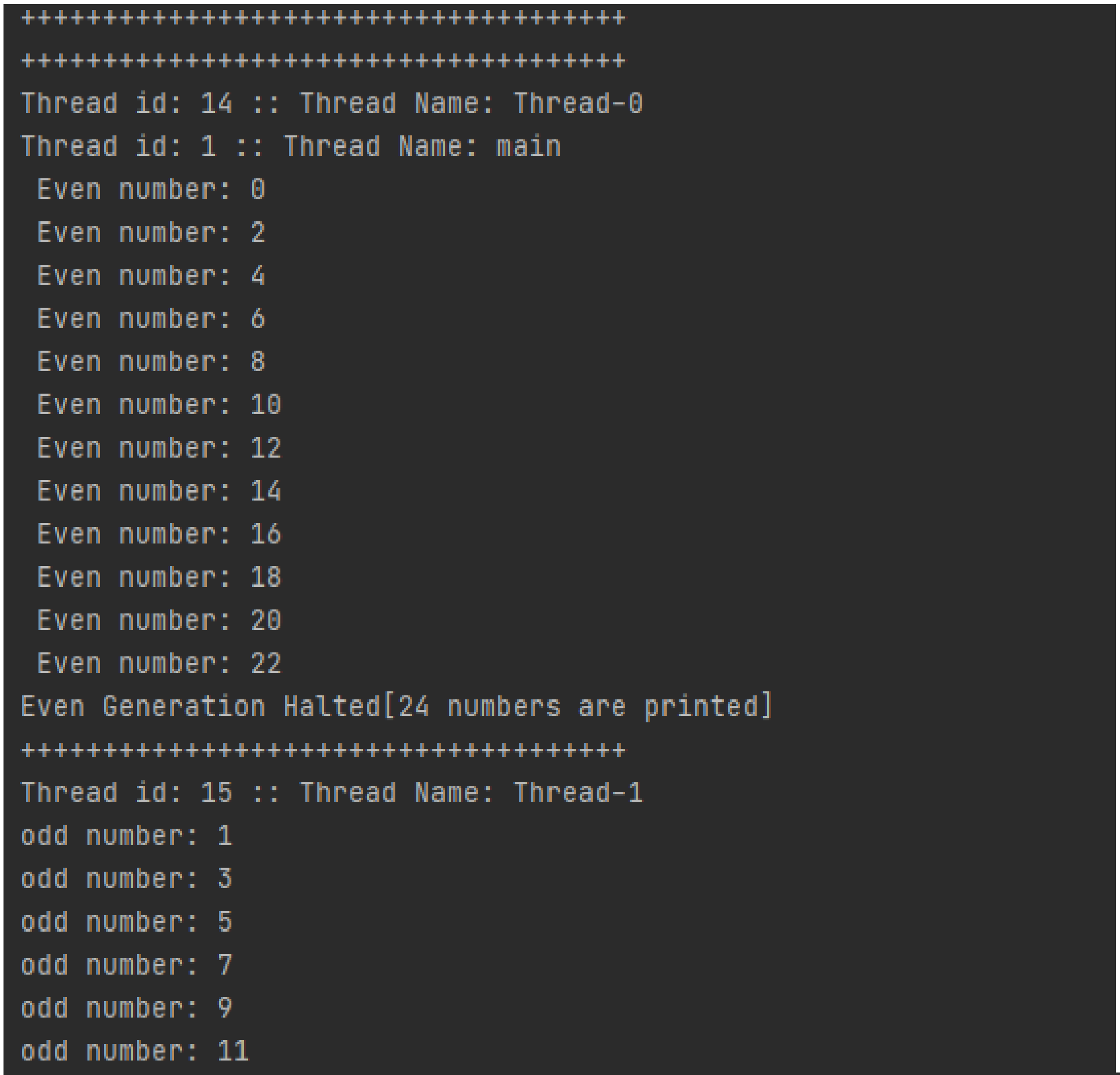
Code

```
class Even{
  private int c=0,n=0;
  synchronized void display(){
    for(int i=0;i<=24;i++){
      if(n==24)
        try{
          System.out.println("Even Generation Halted[24 numbers are printed]");
```

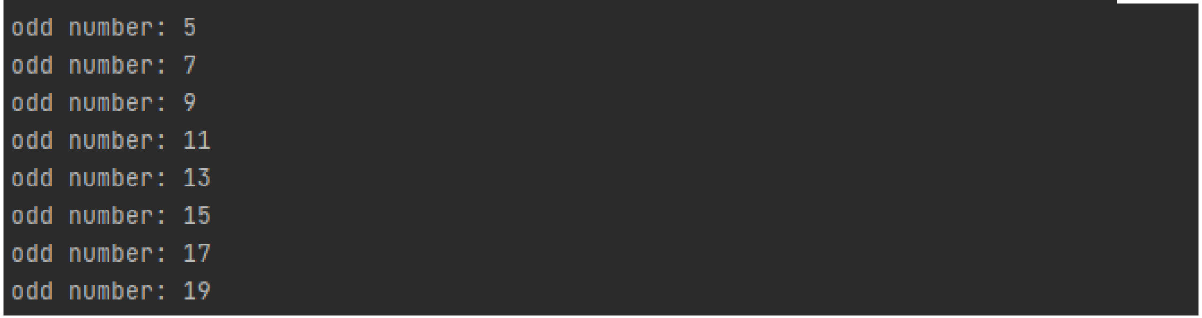
```
                Thread.sleep(5000);
            }catch(InterruptedException e){
                System.out.println("Caught interrupted exception");
            }
         System.out.println(" Even number: "+c);
         c = c+2;
         n= n+2;
      }
   }
}
class Odd{
   int n=1;
   synchronized void display(){
      for(int i=1;i<20;i= i+2)
      {
         System.out.println("odd number: "+i );

      }
   }
}
class OddThread implements Runnable{
   Thread t;
   Odd p1;
   OddThread(){
      t=new Thread(this);
      t.setPriority(Thread.NORM_PRIORITY);
      t.start();
   }
   static void dispt()
   {
      Thread t2 = Thread.currentThread();
      System.out.println("+++++++++++++++++++++++++++++++++++++");
      System.out.println("Thread id: "+t2.getId()+" :: Thread Name: "+t2.getName());
   }
   public void run(){
      p1=new Odd();
      dispt();
      p1.display();

   }
}
```

```
class EvenThread implements Runnable{
   Thread t2;
   Even f;

   EvenThread(){
      t2=new Thread(this);
      t2.setPriority(9);
      t2.start();
      dispt();

   }

   static void dispt()
   {
      Thread t2 = Thread.currentThread();
      System.out.println("++++++++++++++++++++++++++++++++++++++++");
      System.out.println("Thread id: "+t2.getId()+" :: Thread Name: "+t2.getName());
   }
   public void run(){
      f=new Even();
      dispt();
      f.display();

   }
}
class TheoryMultithreadingQ7{
   public static void main(String args[]){
      EvenThread ft=new EvenThread();
      OddThread pt=new OddThread();

   }
}
```

Output



```
+++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++
Thread id: 14 :: Thread Name: Thread-0
Thread id: 1 :: Thread Name: main
 Even number: 0
 Even number: 2
 Even number: 4
 Even number: 6
 Even number: 8
 Even number: 10
 Even number: 12
 Even number: 14
 Even number: 16
 Even number: 18
 Even number: 20
 Even number: 22
Even Generation Halted[24 numbers are printed]
+++++++++++++++++++++++++++++++++++++
Thread id: 15 :: Thread Name: Thread-1
odd number: 1
odd number: 3
odd number: 5
odd number: 7
odd number: 9
odd number: 11
```



```
odd number: 5
odd number: 7
odd number: 9
odd number: 11
odd number: 13
odd number: 15
odd number: 17
odd number: 19
```

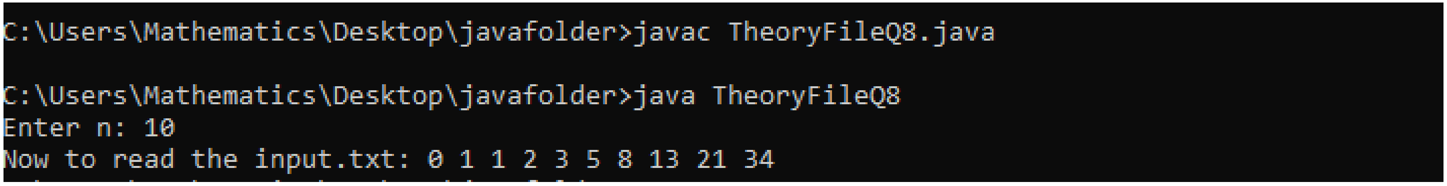
File Handling

Q8) Create a file that stores the Fibonacci series upto n number taken from user.

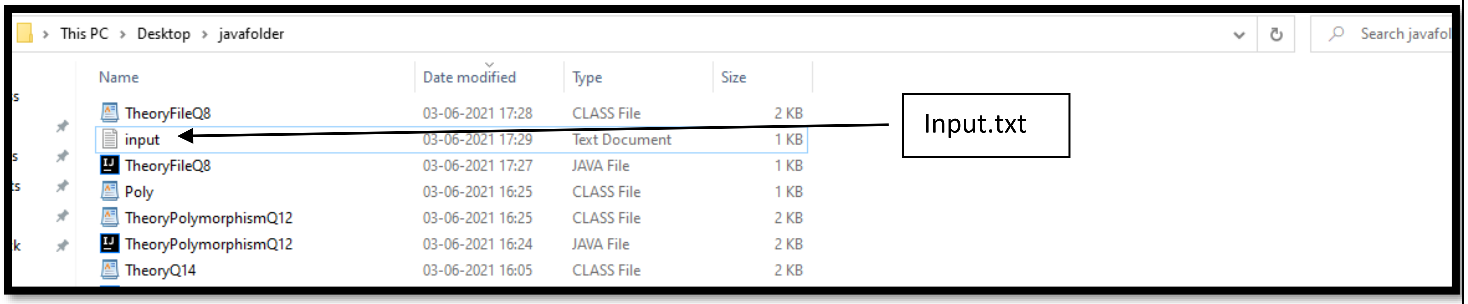
Code

```
import java.util.Scanner;
import java.io.FileWriter;
import java.io.FileReader;

public class TheoryFileQ8 {
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) throws Exception {
        System.out.print("Enter n: ");
        int n = sc.nextInt();
        int [] arr = new int[n];
        arr[0] = 0;
        arr[1] = 1;
        FileWriter writer = new FileWriter("input.txt");
        for(int i=2;i<n;i++){
            int temp = arr[i] = arr[i-1]+arr[i-2];
        }
        for(int i=0;i<n;i++)
        {
            writer.write(arr[i]+ " ");
        }
        writer.close();

        System.out.print("Now to read the input.txt: ");
        FileReader fr=new FileReader("input.txt");
        int i;
        while((i=fr.read())!=-1)
            System.out.print((char)i);
        fr.close();
    }
}
```

Output

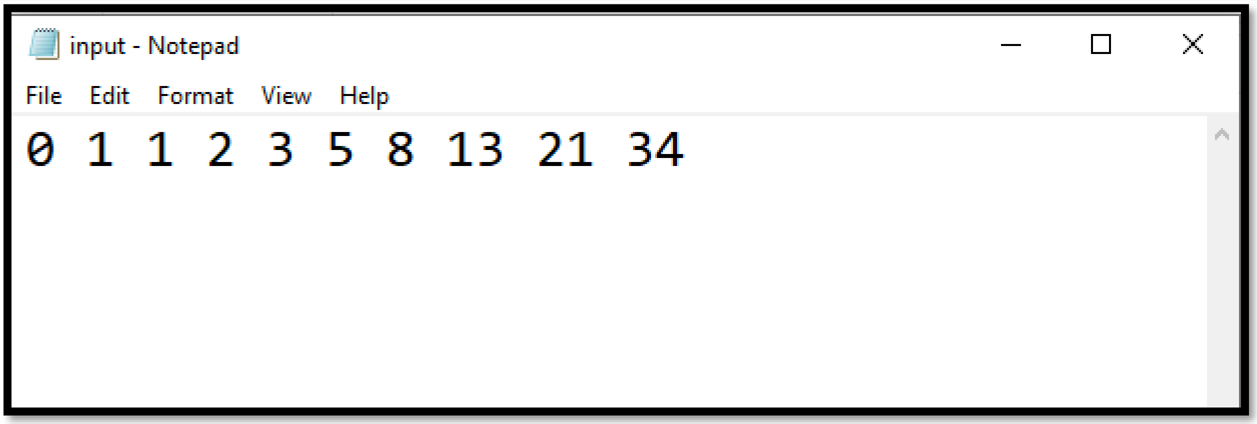


```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryFileQ8.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryFileQ8
Enter n: 10
Now to read the input.txt: 0 1 1 2 3 5 8 13 21 34
```



Name	Date modified	Type	Size
TheoryFileQ8	03-06-2021 17:28	CLASS File	2 KB
input	03-06-2021 17:29	Text Document	1 KB
TheoryFileQ8	03-06-2021 17:27	JAVA File	1 KB
Poly	03-06-2021 16:25	CLASS File	1 KB
TheoryPolymorphismQ12	03-06-2021 16:25	CLASS File	2 KB
TheoryPolymorphismQ12	03-06-2021 16:24	JAVA File	2 KB
TheoryQ14	03-06-2021 16:05	CLASS File	2 KB

Input.txt



```
0 1 1 2 3 5 8 13 21 34
```

List/Map/Set

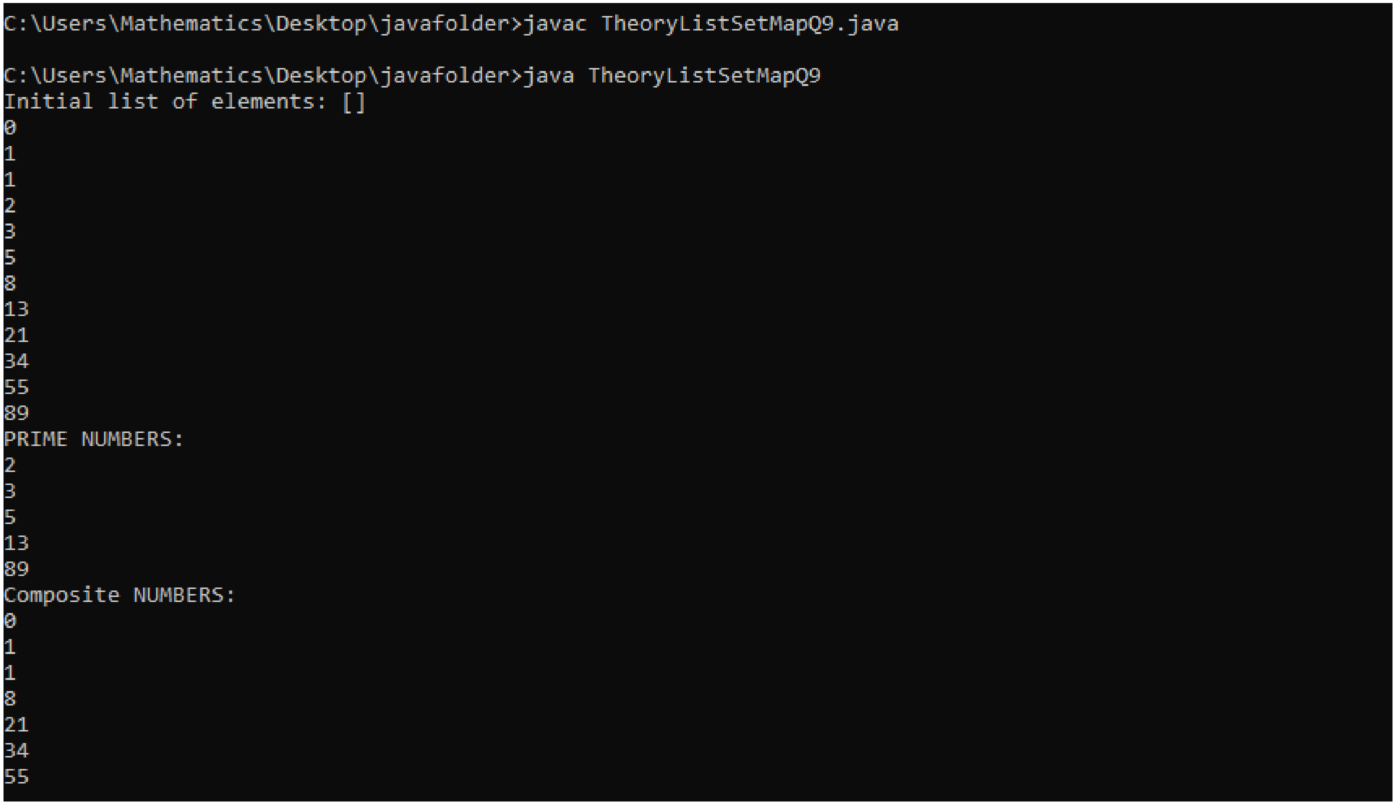
Q9) Ayush is a number enthusiast, he has a linked list that contains the list of Fibonacci series till 100. Ayush comes with an idea to separate the numbers of the linked list in a way where prime numbers and composite numbers are separated. Help Ayush by writing a function to identify the prime numbers from the linked list and store the prime numbers in one array list and composite numbers in another array list. Display the elements in both the array list.

Code

```
package HalaMadrid;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.*;
public class TheoryListSetMapQ9 {
   static boolean checkprime(int n){
      int i,flag=0;
      if(n==0||n==1){
         return false;
      }else{
         for(i=2;i*i<=n;i++){
            if(n%i==0){
               flag=1;
               break;
```

```
                }
            }
            if(flag==0) { return true; }
            else return false;
        }
    }
    static void FibonacciNumbers(int n,LinkedList<Integer> ll)
    {
        int f1 = 0, f2 = 1;
        if (n < 1)
            return;
        ll.add(f1);
        for (int i = 1; f2 < n; i++) {
            ll.add(f2);
            int next = f1 + f2;
            f1 = f2;
            f2 = next;
        }
    }
    public static void main(String args[]){
        LinkedList<Integer> ll=new LinkedList<Integer>();
        System.out.println("Initial list of elements: "+ll);
        FibonacciNumbers(100,ll);
        Iterator<Integer> itr=ll.iterator();
        ArrayList<Integer> primelist=new ArrayList<Integer>();
        ArrayList<Integer> compositelist=new ArrayList<Integer>();
        while(itr.hasNext()){
            int k=itr.next();
            System.out.println(k);
            if(checkprime(k))primelist.add(k);
            else compositelist.add(k);
        }
        Iterator<Integer> itrp=primelist.iterator();
        Iterator<Integer> itrc=compositelist.iterator();
        System.out.println("PRIME NUMBERS:");
        while(itrp.hasNext()){
            System.out.println(itrp.next()+ " ");
        }
        System.out.println("Composite NUMBERS:");
        while(itrc.hasNext()){
            System.out.println(itrc.next()+ " ");
        }
    }
}
```

Output



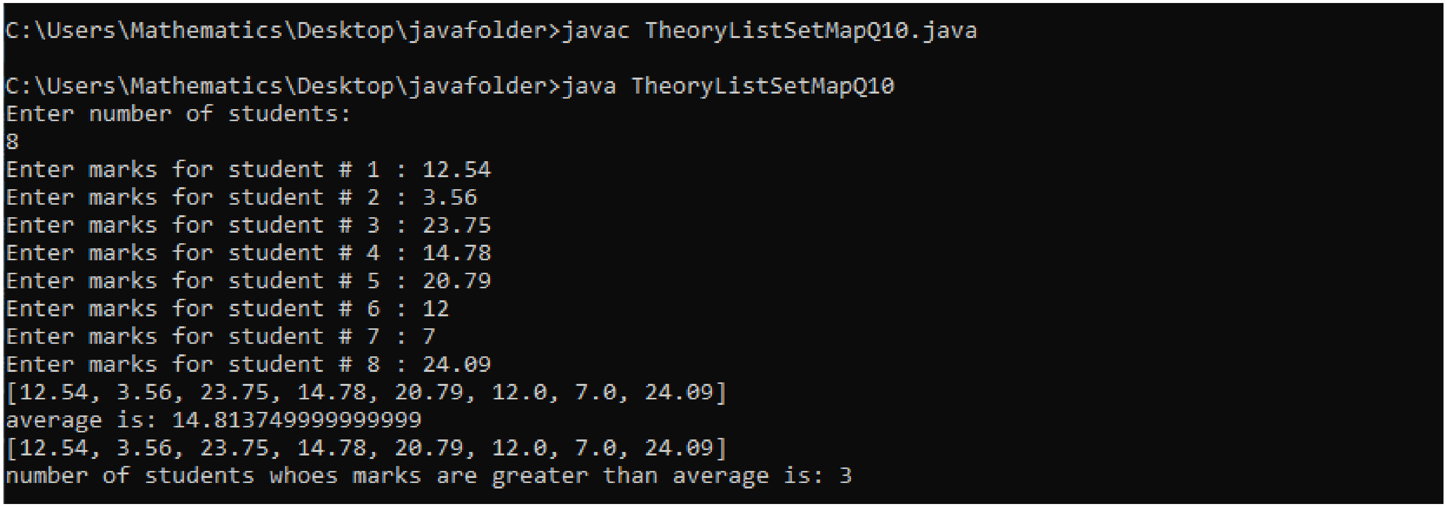
```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryListSetMapQ9.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryListSetMapQ9
Initial list of elements: []
0
1
1
2
3
5
8
13
21
34
55
89
PRIME NUMBERS:
2
3
5
13
89
Composite NUMBERS:
0
1
1
8
21
34
55
```

Q10) Take a list that gives the marks of the student scored in CAT1 in java programming. Store these values in a queue. Find the average marks and display the count of the number of students whose marks are more than the average.

Code

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

class TheoryListSetMapQ10 {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        Queue<Double> q = new LinkedList<>();
        int z;
        System.out.println("Enter number of students: ");
        z =sc.nextInt();
        for(int i=0;i<z;i++)
        {
            double mark;
            int l =i+1;
            System.out.print("Enter marks for student # "+l+" : ");
            mark = sc.nextDouble();
            q.add(mark);
        }
        Queue<Double> temp=new LinkedList<>();
        int size=q.size();
        System.out.println(q);
        double sum = 0;
        while (!q.isEmpty()) {
            Double n = q.remove();
            sum += n;
            temp.add(n);
        }
        double avg=sum/size;
        System.out.println("average is: "+avg+" ");
        System.out.println(temp);
        int count=0;
        for (Double item: temp) {
            if(item>avg){count++;}
        } System.out.println("number of students whoes marks are greater than average is: 
"+count);
    }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryListSetMapQ10.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryListSetMapQ10
Enter number of students:
8
Enter marks for student # 1 : 12.54
Enter marks for student # 2 : 3.56
Enter marks for student # 3 : 23.75
Enter marks for student # 4 : 14.78
Enter marks for student # 5 : 20.79
Enter marks for student # 6 : 12
Enter marks for student # 7 : 7
Enter marks for student # 8 : 24.09
[12.54, 3.56, 23.75, 14.78, 20.79, 12.0, 7.0, 24.09]
average is: 14.813749999999999
[12.54, 3.56, 23.75, 14.78, 20.79, 12.0, 7.0, 24.09]
number of students whoes marks are greater than average is: 3
```

Generic Java class

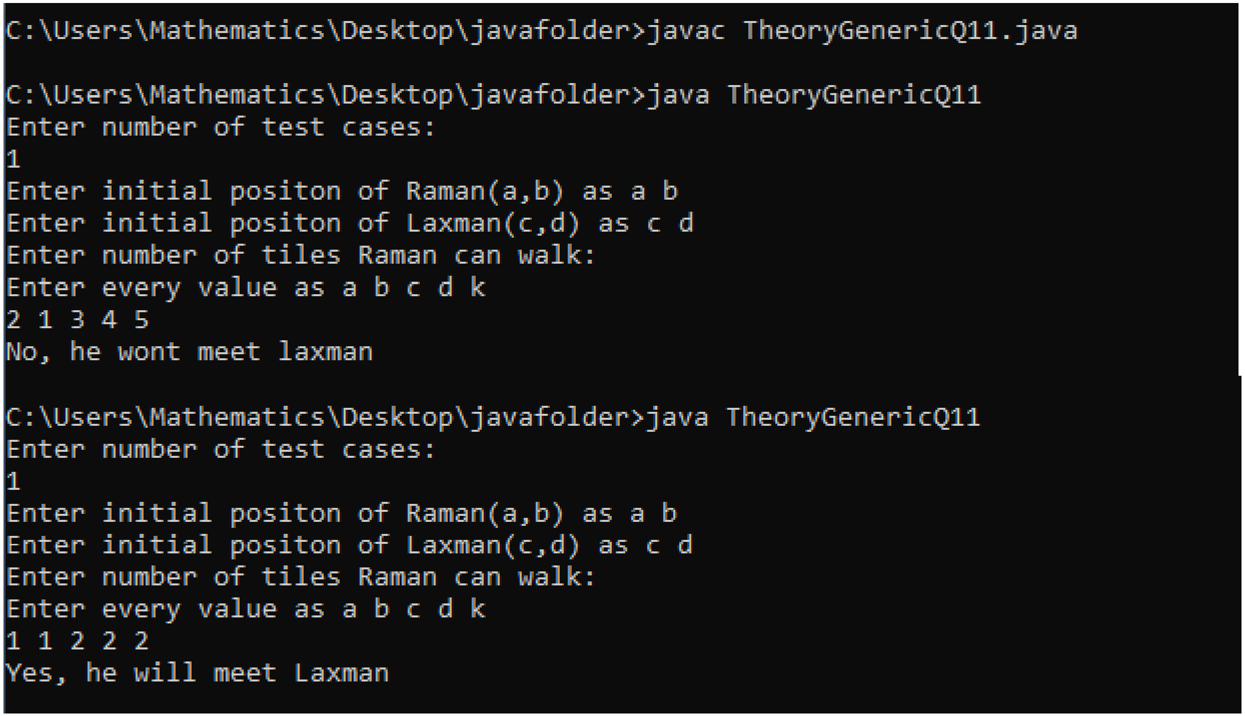
Q11) Raman is on an infinite chess board and he wants to meet Laxman who is on position (c,d) and Raman is on position (a,b). Raman can move only on blocks whose edges are shared with the current block he is standing on, or in other word he can move only on adjacent blocks, but he can only move k blocks, now find whether he will be able to meet Laxman?

Code

```
import java.util.*;
import java.lang.*;
import java.io.*;
class TheoryGenericQ11
{
    public static void main (String[] args) throws java.lang.Exception
    {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(System.out));
       System.out.println("Enter number of test cases:");
        int t = Integer.parseInt(reader.readLine());
        System.out.println("Enter initial positon of Raman(a,b) as a b");
        System.out.println("Enter initial positon of Laxman(c,d) as c d");
        System.out.println("Enter number of tiles Raman can walk: ");
        System.out.println("Enter every value as a b c d k");
        for (int p = 0; p < t; p++){
            String[] s = reader.readLine().split(" ");
            int a = Integer.parseInt(s[0]);
            int b = Integer.parseInt(s[1]);
            int c = Integer.parseInt(s[2]);
            int d = Integer.parseInt(s[3]);
            int k = Integer.parseInt(s[4]);

            int row = Math.abs(a - c);
            int col = Math.abs(b - d);
            int sum  = row + col;
            int temp = 0;
            if (k == sum) temp = 1;
            else if (k < sum) {
                temp = 0;
            }
            else if ((k % 2 == 0 && sum % 2 == 0) || (k % 2 == 1 && sum % 2 == 1)) temp = 1;
```

```
      String answer;
      if (temp == 1) answer = "Yes, he will meet Laxman";
      else answer = "No, he wont meet laxman";
      writer.write(answer + "\n");
      writer.flush();
    }
  }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryGenericQ11.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryGenericQ11
Enter number of test cases:
1
Enter initial positon of Raman(a,b) as a b
Enter initial positon of Laxman(c,d) as c d
Enter number of tiles Raman can walk:
Enter every value as a b c d k
2 1 3 4 5
No, he wont meet laxman

C:\Users\Mathematics\Desktop\javafolder>java TheoryGenericQ11
Enter number of test cases:
1
Enter initial positon of Raman(a,b) as a b
Enter initial positon of Laxman(c,d) as c d
Enter number of tiles Raman can walk:
Enter every value as a b c d k
1 1 2 2 2
Yes, he will meet Laxman
```

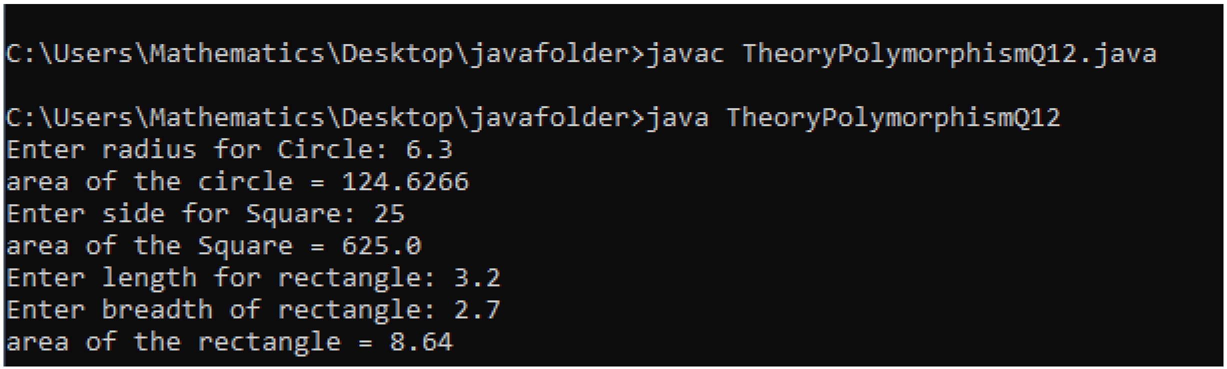
Polymorphism

Q12) Rakshita is computer engineer, and she is fascinated by programming, and she loves the word calculate, so she defines a method calculate and uses it find the area of a square, a rectangle and a circle. Help her in writing this program.

Code

```
import java.util.*;
class Poly{
  static double Area(double r)
  {
    return 3.14*r*r;
  }
  static int Area(int r){
    return r*r;
  }
  static double Area(double l , double b)
  {
    return l*b;
  }
}
public class TheoryPolymorphismQ12 {
  static Scanner sc =new Scanner(System.in);
  static void display(double r, String a, String b)
  {
    System.out.println(a+ " of the "+b+" = "+r);
  }

  public static void main(String[] args) {
    Poly a1 = new Poly();
    Poly a2 = new Poly();
    Poly a3 = new Poly();
    System.out.print("Enter radius for Circle: ");
    double r = sc.nextDouble();
    double a;
    a = a1.Area(r);
    display(a, "area", "circle");
    System.out.print("Enter side for Square: ");
    int s = sc.nextInt();
    int A = a2.Area(s);
    display(A, "area", "Square");
    System.out.print("Enter length for rectangle: ");
    double l = sc.nextDouble();
    System.out.print("Enter breadth of rectangle: ");
    double b = sc.nextDouble();
    a = a3.Area(l,b);
    display(a, "area", "rectangle");
  }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryPolymorphismQ12.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryPolymorphismQ12
Enter radius for Circle: 6.3
area of the circle = 124.6266
Enter side for Square: 25
area of the Square = 625.0
Enter length for rectangle: 3.2
Enter breadth of rectangle: 2.7
area of the rectangle = 8.64
```

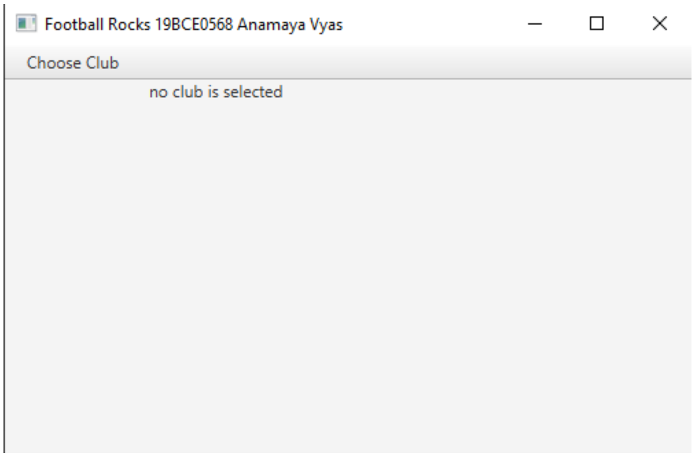
JavaFX

Q13)Write a program in javaFX with a UI and a menu Bar with 7 menu options of club names which will display the facts of the club on the user interface once chosen by the user.

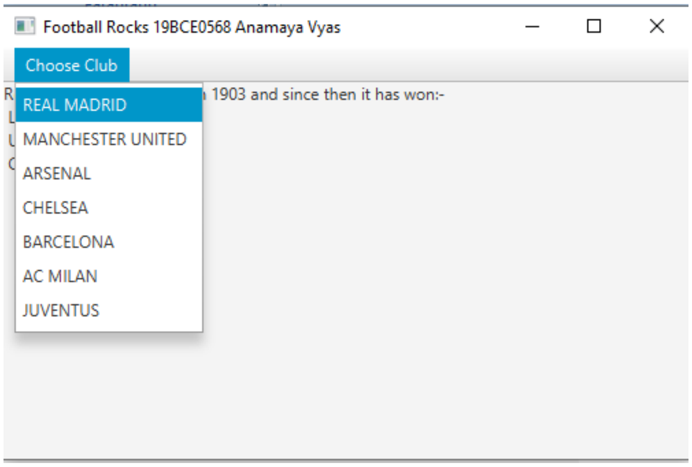
Code

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.*;
import javafx.stage.Stage;
import javafx.scene.control.Alert.AlertType;
import java.time.LocalDate;
public class TheoryJavaFXQ13 extends Application {

    public void start(Stage s)
    {
        s.setTitle("Football Rocks 19BCE0568 Anamaya Vyas");
        Menu m = new Menu("Choose Club");
        MenuItem m1 = new MenuItem("REAL MADRID");
        MenuItem m2 = new MenuItem("MANCHESTER UNITED");
        MenuItem m3 = new MenuItem("ARSENAL");
        MenuItem m4 = new MenuItem("CHELSEA");
        MenuItem m5 = new MenuItem("BARCELONA");
        MenuItem m6 = new MenuItem("AC MILAN");
        MenuItem m7 = new MenuItem("JUVENTUS");
        m.getItems().add(m1);
        m.getItems().add(m2);
        m.getItems().add(m3);
        m.getItems().add(m4);
        m.getItems().add(m5);
        m.getItems().add(m6);
        m.getItems().add(m7);
        Label l = new Label("\t\t\t\t"+ "no club is selected");
        String RM = "Real Madrid was founded in 1903 and since then it has won:-\n La Liga 34 time\n UCL 13 times \n Copa Del Ray 19 times";
        String MU = "Manchester United was founded in 1878 and since then it has won:-\n English Premier League 20 times\n UCL 3 times \n FA Cup 12 times";
        String AS = "Arsenal FC was founded in 1886 and since then it has won:-\n English Premier League 13 times \n UCL 0 times \nFA Cup 19 times";
        String CHE = "Chelsea FC was founded in 1905 and since then it has won:-\n English Premier League 7 times \n UCL 2 times \n FA cup 8 times";
        String FCB = "FC Barcelona was founded in 1899 and since then it has won:- \n La Liga 25 times \n UCL 5 times \n Copa Del Ray 30 times ";
        String ACM = "AC Milan was founded in 1899 and since then it has won:- \n Serie A 18 times \n UCL 7 times \n Coppa Italia 7 times";
        String JUVE = "Juventus was founded in 1897 and since then it has won:- \n Serie A 36 times \nUCL 2 times \n Coppa Italia 14 times ";

        EventHandler<ActionEvent> event = new EventHandler<ActionEvent>() {
           public void handle(ActionEvent e)
           {
              if(((MenuItem)e.getSource()).getText() == "REAL MADRID")
              l.setText(RM);
              else if(((MenuItem)e.getSource()).getText() == "MANCHESTER UNITED")
                 l.setText(MU);
              else if(((MenuItem)e.getSource()).getText() == "ARSENAL")
                 l.setText(AS);
              else if(((MenuItem)e.getSource()).getText() == "CHELSEA")
                 l.setText(CHE);
              else if(((MenuItem)e.getSource()).getText() == "BARCELONA")
                 l.setText(FCB);
              else if(((MenuItem)e.getSource()).getText() == "AC MILAN")
                 l.setText(ACM);
              else if(((MenuItem)e.getSource()).getText() == "JUVENTUS")
                 l.setText(JUVE);
           }
        };
        m1.setOnAction(event);
        m2.setOnAction(event);
        m3.setOnAction(event);
        m3.setOnAction(event);
        m4.setOnAction(event);
        m5.setOnAction(event);
        m6.setOnAction(event);
        m7.setOnAction(event);
        MenuBar mb = new MenuBar();
        mb.getMenus().add(m);
        VBox vb = new VBox(mb, l);
        Scene sc = new Scene(vb, 500, 300);
        s.setScene(sc);
        s.show();
    }

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

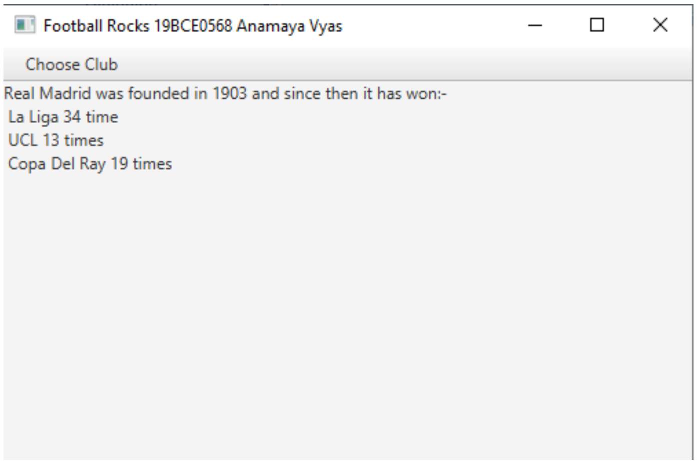
Output



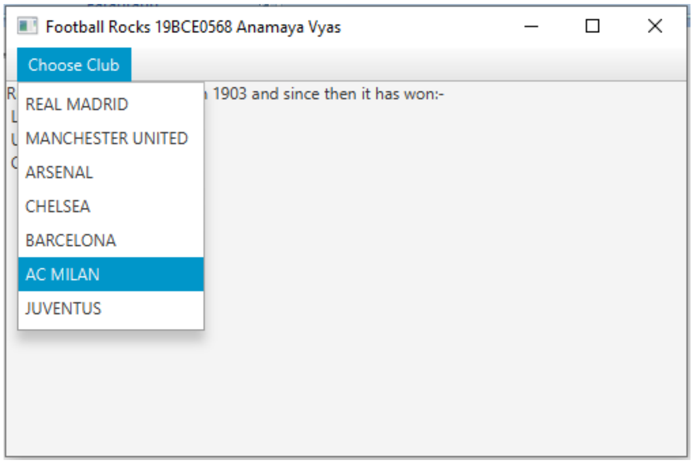
no club is selected



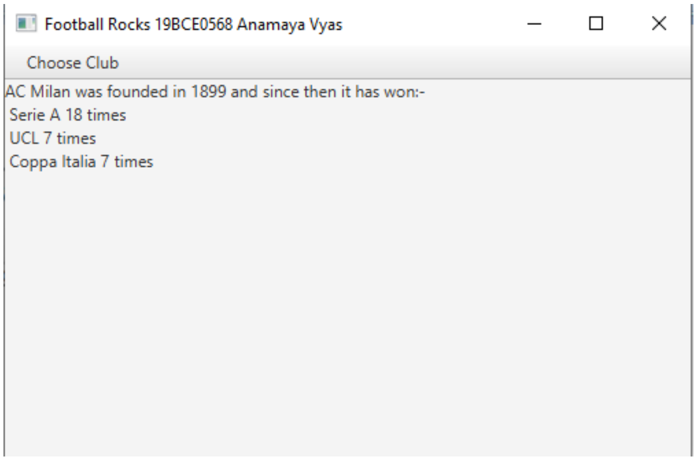
R
1903 and since then it has won:-
L
U
C



```
Real Madrid was founded in 1903 and since then it has won:-
 La Liga 34 time
 UCL 13 times
 Copa Del Ray 19 times
```



```
1903 and since then it has won:-
```



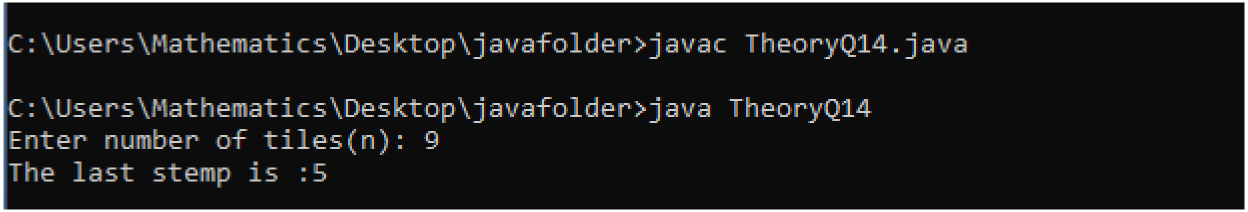
```
AC Milan was founded in 1899 and since then it has won:-
 Serie A 18 times
 UCL 7 times
 Coppa Italia 7 times
```

Q14) There are N tiles in a room, and Aarav and Aditya are standing on either ends of the tiles and they start coming towards each other, in the following way:- Aarav will step on tile 1, then Aditya on tile N, then Aarav on tile 2, and Aditya on tile N-1, such that they come face to face with each other. Tell the last tile they landed on before coming face to face to each other.

Code

```
import java.util.*;
import java.lang.*;
public class TheoryQ14
{
   public static void main (String[] args) {
      Scanner scn = new Scanner(System.in);
      System.out.print("Enter number of tiles(n): ");
      int n = scn.nextInt();
      int ans = n / 2;
      ans = ans + 1;
      System.out.println("The last stemp is :" + ans);
   }
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryQ14.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryQ14
Enter number of tiles(n): 9
The last stemp is :5
```

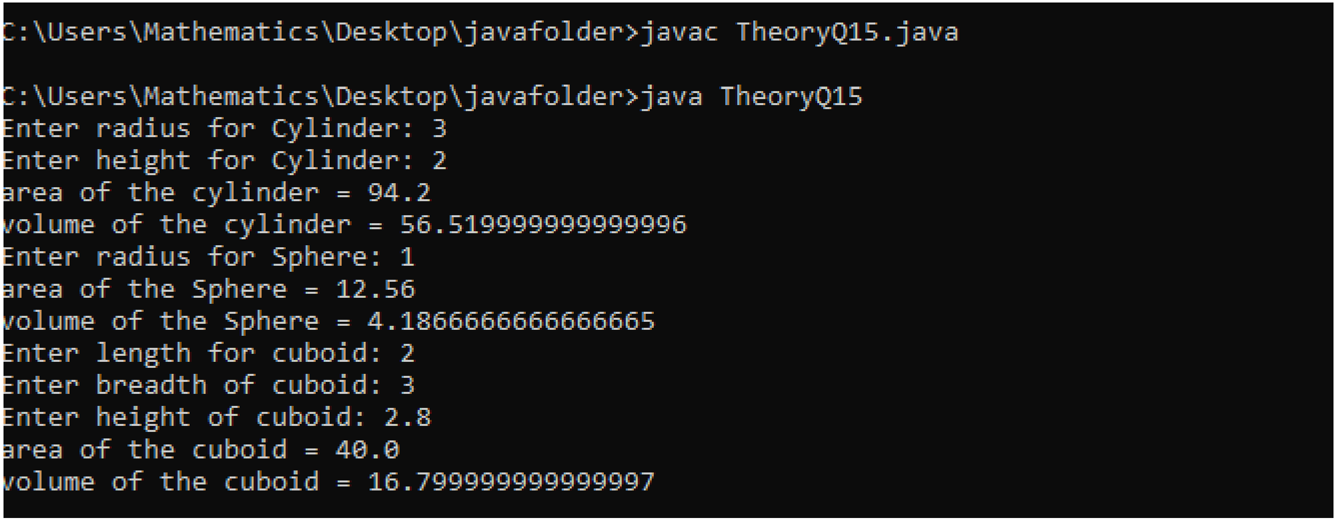
Q15) Using method overloading write a function that calculates the area as well as volume of :-
i) Cylinder
ii) Sphere
iii) Cuboid
Create classes for each of the geometric shape which will store its attributes.

Code

```
import java.lang.invoke.VolatileCallSite;
import java.util.Scanner;

class Polymorphism{
   static double Area(double r , double h)
   {
      return 3.14*2*r*(r+h);
   }
   static double Area(double r){
      return 4*3.14*r*r;
   }
   static double Area(double l , double b, double h)
   {
      return 2*(l*b+b*h+h*l);
   }
   static double Volume(double r, double h){
      return 3.14*r*r*h;
   }
   static double Volume(double r)
   {
      return 4*3.14*r*r*r/3;
   }
   static double Volume(double l, double b, double h){
      return l*b*h;
   }
}

public class TheoryQ15 {
      static Scanner sc =new Scanner(System.in);
      static void display(double r, String a, String b)
      {
         System.out.println(a+ " of the "+b+" = "+r);
      }

   public static void main(String[] args) {
      Polymorphism a1 = new Polymorphism();
      Polymorphism a2 = new Polymorphism();
      Polymorphism a3 = new Polymorphism();
      System.out.print("Enter radius for Cylinder: ");
      double r = sc.nextDouble();
      System.out.print("Enter height for Cylinder: ");
      double h = sc.nextDouble();
      double a;
      a = a1.Area(r,h);
      double v = a1.Volume(r, h);
      display(a, "area", "cylinder");
      display(v, "volume", "cylinder");
      System.out.print("Enter radius for Sphere: ");
      r = sc.nextDouble();
      a = a2.Area(r);
      v = a2.Volume(r);
      display(a, "area", "Sphere");
      display(v, "volume", "Sphere");
      System.out.print("Enter length for cuboid: ");
      double l = sc.nextDouble();
      System.out.print("Enter breadth of cuboid: ");
      double b = sc.nextDouble();
      System.out.print("Enter height of cuboid: ");
      h = sc.nextDouble();
      a = a2.Area(l,b,h);
      v = a2.Volume(l,b,h);
      display(a, "area", "cuboid");
      display(v, "volume", "cuboid");

   }
```

```
}
```

Output



```
C:\Users\Mathematics\Desktop\javafolder>javac TheoryQ15.java

C:\Users\Mathematics\Desktop\javafolder>java TheoryQ15
Enter radius for Cylinder: 3
Enter height for Cylinder: 2
area of the cylinder = 94.2
volume of the cylinder = 56.519999999999996
Enter radius for Sphere: 1
area of the Sphere = 12.56
volume of the Sphere = 4.1866666666666665
Enter length for cuboid: 2
Enter breadth of cuboid: 3
Enter height of cuboid: 2.8
area of the cuboid = 40.0
volume of the cuboid = 16.799999999999997
```