**SESSION 1/PROGRAMMING FUNDAMENTALS**

**Dhoni's Marriage Year**

PROGRAM:

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

        Scanner sc=new Scanner(System.in);

        int n=sc.nextInt();

        if(n%4==0)

        {

            if(n%100==0)

            {

                if(n%400==0)

                {

                System.out.println("Yes");

                }

                else

                {

                    System.out.println("No");

                }

            }

            else

            {

                System.out.println("Yes");

            }

        }

        else

        {

            System.out.println("No");

        }

}

}

**Dhoni's First Meet**

PROGRAM:

import java.util.\*;

public class Main

 {

     public static void main(String[] args)

  {

        Scanner sc=new Scanner(System.in);

        int n=sc.nextInt();

        int flag=0;

        for(int i=2;i<100;i=i+3)

        {

            if(n==i)

            {

                flag=1;

            }

        }

        if(flag==1)

        {

            System.out.println("Yes");

        }

        else

        {

            System.out.println("No");

        }

   }

}

**Dhoni's Day Out**

PROGRAM:

import java.util.\*;

public class Main

{

    public static void main( String args[] )

    {

        Scanner sc =  new Scanner ( System.in );

        int count=0,x;

        int tot= sc.nextInt();

        int a= sc.nextInt();

        int b= sc.nextInt();

        int c= sc.nextInt();

        int mon= sc.nextInt();

        int r= mon-tot;

        if(r>=a+b)

        {

            x=r-a-b;

            if(x>=c)

            count=3;

            else

            count=2;

        }

        else if(r>=b+c)

        {

            x=r-b-c;

            if(x>=a)

            count=3;

            else

            count=2;

        }

        else if(r>=a+c)

        {

            x=r-a-c;

            if(x>=b)

            count=3;

            else

            count=2;

        }

        else if(( r<a+b || r<b+c || r<a+c) &&( r>=a||r>=b || r>=c))

        count=1;

        else

        count=0;

        System.out.println(count);

    }

}

**Player Selection for Cricket Team**

PROGRAM:

import java.util.\*;

public class Main

{

    public static void main(String args[])

    {

        Scanner sc=new Scanner(System.in);

        int n1=sc.nextInt();

        int n2=sc.nextInt();

        int n3=sc.nextInt();

        if((n1>=100)&&(n2>=100) ||

        (n2>=100)&&(n3>=100) ||

        (n3>=     100)&&(n1>=100 )||

        (n1>=50)&&(n2>=50)&&n3>=50))

        {

            System.out.println("Selected");

        }

        else if((n1>=100 )|| (n2>=100) || (n3>=100) || (n1>=50)&&(n2>=50) || (n2>=50)&&(n3>=50 )|| (n3>=50)&&(n1>=50))

        {

            System.out.println("Waitlisted");

        }

        else

        {

            System.out.println("Rejected");

        }

    }

}

**Valentine's Day Gift**

PROGRAM:

import java.util.\*;

class Main

{

    public static void main(String args[])

    {

        Scanner sc=new Scanner(System.in);

        int a=sc.nextInt();

        int b=sc.nextInt();

        int c=sc.nextInt();

        int d=sc.nextInt();

        int count=0;

        boolean e=sc.nextBoolean();

       int  max=( a>b ? (a>c?a:c) : (b>c ?b:c));

        int min=(a<b ? (a<c ? a:c ): (b<c?b:c));

        int mid= a+b+c - (min+max);

      if(e==true)

        {

            System.out.println(3);

        }

            else

            {

                int[] arr = new int[3];

                 arr[0]=min;

                 arr[1]=mid;

                 arr[2]=max;

                for(int i=0;i<3;i++)

                {

                    if(d>=arr[i])

                    {

                      d=d-arr[i];

                    count++;

                    }

                }

                System.out.println(count);

            }

    }

 }

**SESSION 2/PROGRAMMING FUNDAMENTALS**

**Dhoni's Lucky Number-1**

PROGRAM:

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner sc= new Scanner (System.in);

int a=sc.nextInt();

int b=sc.nextInt();

for(int i=a;i<=b;i++)

{

int n=i;

int count=0;

while(n>0)

{

int r=n%10;

if(r!=1)

{

if(r==2||r==3||r==5||r==7)

{

count++;

}

}

n=n/10;

}

if(count==2)

System.out.print(i+" ");

}

}

}

**Dhoni's Lucky Number-2**

PROGRAM**:**

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int a=sc.nextInt();

int b=sc.nextInt();

int n,c,k;

for(int i=a;i<=b;i++)

{

int rev=0;

n=i;

c=i;

while(n!=0)

{

k=n%10;

rev=rev\*10+k;

n=n/10;

}

if(rev==c)

{

System.out.print(rev+" ");

}

}

}

}

**Dhoni's Lucky Number-3**

PROGRAM:

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int a=sc.nextInt();

int b=sc.nextInt();

for(int i=a;i<=b;i++)

{

int sum=0;

for(int j=1;j<=i;j++)

{

if(i%j==0)

{

sum=sum+j;

}

}

if(sum<2\*i)

System.out.print(i+" ");

}

}

}

**Dhoni's Lucky Number-4**

PROGRAM:

import java.util.\*;

public class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int a=sc.nextInt();

int b=sc.nextInt();

for(int i=a;i<=b;i++)

{

if(i!=1)

{

int c=0;

for(int j=2;j<=i/2;j++)

{

if(i%j==0)

c++;

}

if(c==0)

System.out.print(i+" ");

}

}

}

}

**SESSION 1 /CLASSES AND OBJECTS 1A**

**Player**

PROGRAM:

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the player name");

String sname=sc.nextLine();

System.out.println("Enter the country name");

String scountry=sc.nextLine();

System.out.println("Enter the skill");

String sskill=sc.nextLine();

Player obj=new Player();

obj.name=sname;

obj.country=scountry;

obj.skill=sskill;

System.out.println("Player Details :");

System.out.println("Player Name : " +obj.name);

System.out.println("Country Name : "+obj.country);

System.out.println("Skill : " +obj.skill);

}

}

public class Player

{

String name;

String country;

String skill;

}

**Venue**

**PROGRAM:**

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the venue name");

String Vname=sc.nextLine();

System.out.println("Enter the city name");

String Vcity=sc.nextLine();

Venue obj=new Venue();

obj.name=Vname;

obj.city=Vcity;

System.out.println("Venue Details :");

System.out.println("Venue Name : " +obj.name);

System.out.println("City Name : " +obj.city);

}

}

public class Venue

{

String name;

String city;

}

**SESSION 2/CLASSES AND OBJECTS 1B**

**Venue**

**PROGRAM:**

import java.util.\*;

public class Main

{

public static void main (String args[])

{

Team obj=new Team();

Scanner x=new Scanner(System.in);

System.out.println("Enter the team name");

String name=x.nextLine();

System.out.println("Enter the coach name");

String coach=x.nextLine();

System.out.println("Enter the location name");

String location=x.nextLine();

System.out.println("Enter the players name");

String players=x.nextLine();

System.out.println("Enter the captain name");

String captain=x.nextLine();

obj.setName(name);

obj.setCoach(coach);

obj.setLocation(location);

obj.setPlayers(players);

obj.setCaptain(captain);

obj.displayTeamDetails();

}

}

class Team

{

String name;

String coach;

String location;

String players;

String captain;

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public String getCoach()

{

return coach;

}

public void setCoach(String coach)

{

this.coach = coach;

}

public String getLocation()

{

return location;

}

public void setLocation(String location)

{

this.location = location;

}

public String getPlayers()

{

return players;

}

public void setPlayers(String players)

{

this.players = players;

}

public String getCaptain()

{

return captain;

}

public void setCaptain(String captain)

{

this.captain = captain;

}

public void displayTeamDetails()

{

System.out.println("Team Details");

System.out.println("Team : "+this.getName());

System.out.println("Coach : "+this.getCoach());

System.out.println("Location : "+this.getLocation());

System.out.println("Players : "+this.getPlayers());

System.out.println("Captain : "+this.getCaptain());

}

}

**Update Player Details**

PROGRAM:

import java.util.\*;

public class Main{

public static void main(String[] args)

{

Player obj=new Player();

Scanner x=new Scanner(System.in);

System.out.println("Enter the player name");

obj.setName(x.nextLine());

System.out.println("Enter the country name");

obj.setCountry(x.nextLine());

System.out.println("Enter the skill");

obj.setSkill(x.nextLine());

System.out.println("Player Details");

System.out.println("Player Name :"+obj.getName());

System.out.println("Country Name :"+obj.getCountry());

System.out.println("Skill :"+obj.getSkill());

System.out.println("Verify and Update Player Details");

int a=0;

while(a==0)

{

System.out.println("Menu\n1.Update Player Name\n2.Update Country Name\n3.Update Skill\n4.All informations Correct/Exit\nType 1 or 2 or 3 or 4");

int n=x.nextInt();

switch(n)

{

case 1:

System.out.println("The current player name is "+obj.getName());

System.out.println("Enter the player name");

x.nextLine();

String z=x.nextLine();

obj.setName(z);

break;

case 2:

System.out.println("The current country name is "+obj.getCountry());

System.out.println("Enter the country name");

x.nextLine();

z=x.nextLine();

obj.setCountry(z);

break;

case 3:

System.out.println("The current skill is "+obj.getSkill());

System.out.println("Enter the skill");

x.nextLine();

z=x.nextLine();

obj.setSkill(z);

break;

case 4:

System.out.println("Player Details");

System.out.println("Player Name :"+obj.getName());

System.out.println("Country Name :"+obj.getCountry());

System.out.println("Skill :"+obj.getSkill());

a=1;

}

}

x.close();

}

}

class Player

{

String name;

String country;

String Skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getSkill() {

return Skill;

}

public void setSkill(String skill) {

Skill = skill;

}

}

**SESSION 1/CLASSES AND OBJECTS II**

**Venue Details**

PROGRAM:

import java.io.\*;

public class Main{

public static void main(String args[])throws IOException{

BufferedReader sc=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter the number of venues\n");

int n=Integer.parseInt(sc.readLine());

Venue mo[]=new Venue[34];

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

{

System.out.print("Enter the details of venue "+(i+1)+"\n");

String a=sc.readLine();

String b[]=a.split(",");

mo[i]=new Venue(b[0],b[1]);

}

System.out.println("Venue Details");

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

{

System.out.println(mo[i].toString());

}

}

}

import java.util.\*;

class Venue{

private String name;

private String city;

public void setName(String name)

{

this.name=name;

}

public String getName()

{

return name;

}

public void setCity(String city)

{

this.city=city;

}

public String getCity()

{

return city;

}

public Venue()

{

}

public Venue(String name,String city)

{

this.name=name;

this.city=city;

}

public String toString()

{

return "Venue Name : "+name+"\nCity Name : "+city;

}

}

**SESSION 2/CLASSES AND OBJECTS II**

**Innings Details**

PROGRAM:

import java.util.\*;

public class Main {

public static void main(String[] args)throws NullPointerException

{

Scanner x=new Scanner(System.in);

System.out.println("Enter the number of innings");

int n=x.nextInt();

Innings obj[]=new Innings[n];

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

{

obj[i]=new Innings();

System.out.println("Enter the values for Innings "+(i+1));

System.out.println("Enter the BattingTeam");

x.nextLine();

obj[i].setBattingTeam(x.nextLine());

System.out.println("Enter the runs scored");

// x.nextLine();

obj[i].setRuns(x.nextLong());

}

System.out.println("Innings Details");

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

{

System.out.println("Innings "+(i+1)+"\n"+obj[i].toString());

}

x.close();

}

}

public class Innings

{

String battingTeam;

Long runs;

public String getBattingTeam()

{

return battingTeam;

}

public void setBattingTeam(String battingTeam)

{

this.battingTeam = battingTeam;

}

public Long getRuns()

{

return runs;

}

public void setRuns(Long runs)

{

this.runs = runs;

}

public Innings()

{

}

public Innings(String battingTeam,long runs)

{

this.battingTeam=battingTeam;

this.runs=runs;

}

public String toString()

{

return battingTeam+" -- "+runs;

}

}

**Check Player Details**

PROGRAM:

import java.util.\*;

public class Main {

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the player 1 details");

System.out.println("Enter the player name");

String pname=sc.nextLine();

System.out.println("Enter the country name");

String pcity=sc.nextLine();

System.out.println("Enter the skill");

String pskill=sc.nextLine();

Player p=new Player();

p.setName(pname);

p.setCountry(pcity);

p.setSkill(pskill);

System.out.println(p.getName()+" --- "+p.getCountry()+" --- "+p.getSkill());

System.out.println("Enter the player 2 details");

System.out.println("Enter the player name");

String p1name=sc.nextLine();

System.out.println("Enter the country name");

String p1city=sc.nextLine();

System.out.println("Enter the skill");

String p1skill=sc.nextLine();

Player p1=new Player();

p1.setName(p1name);

p1.setCountry(p1city);

p1.setSkill(p1skill);

System.out.println(p1.getName()+" --- "+ p1.getCountry()+" --- "+ p1.getSkill());

if(p1.equals(p))

{

System.out.println("Both the player details are same.");

}

else

{

System.out.println("Both the player details are not same.");

}

}

}

public class Player {

private String name;

private String country;

private String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public boolean equals(Object o )

{

Player p1=(Player)o;

if(name.equals(p1.name)&&country.equals(p1.country)&&skill.equals(p1.skill))

{

return true;

}

else

{

return false;

}

}

}

**SESSION 1/CLASSES AND OBJECTS III**

**Player Details**

PROGRAM:

import java.io.\*;

public class Main {

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

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the player name");

String pname=br.readLine();

System.out.println("Enter the country name");

String cname=br.readLine();

System.out.println("Enter the skill");

String pskill=br.readLine();

Player player=new Player(pname,cname,pskill);

PlayerBO bo=new PlayerBO();

bo.displayPlayerDetails(player);

}

}

public class Player {

private

String name;

String country;

String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public Player(String name, String country, String skill) {

this.name = name;

this.country = country;

this.skill = skill;

}

public String toString()

{

StringBuffer sb=new StringBuffer();

sb.append("player.getName()");

sb.append("player.getCountry()");

sb.append("player.getSkill()");

return String.format("%-15s %-15s %-15s",getName(),getCountry(),getSkill());

}

}

public class PlayerBO {

void displayPlayerDetails(Player player)

{

System.out.println("Player Details");

System.out.println(player.toString());

}

}

**Delivery Details**

PROGRAM:

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Main {

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

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

Long over;

Long ball;

String batsman;

String bowler;

String nonStriker;

Integer n;

System.out.println("Enter the number of deliveries");

n=Integer.parseInt(br.readLine());

Delivery [] delivery=new Delivery[n];

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

{

System.out.println("Enter the over");

over=Long.parseLong(br.readLine());

System.out.println("Enter the ball");

ball=Long.parseLong(br.readLine());

System.out.println("Enter the batsman");

batsman=br.readLine();

System.out.println("Enter the bowler");

bowler=br.readLine();

System.out.println("Enter the nonStriker");

nonStriker=br.readLine();

delivery[i]=new Delivery(over,ball,batsman,bowler,nonStriker);

}

DeliveryBO deliveryBO=new DeliveryBO();

deliveryBO.displayAllDeliveryDetails(delivery);

}

}

public class DeliveryBO {

void displayAllDeliveryDetails(Delivery[] deliveryList)

{

System.out.println("Delivery Details");

for(int i=0;i<deliveryList.length;i++)

{

System.out.println("Delivery--"+(i+1));

System.out.println(deliveryList[i].toString());

}

}

}

public class Delivery {

private

Long over;

Long ball;

String batsman;

String bowler;

String nonStriker;

public Long getOver() {

return over;

}

public void setOver(Long over) {

this.over = over;

}

public Long getBall() {

return ball;

}

public void setBall(Long ball) {

this.ball = ball;

}

public String getBatsman() {

return batsman;

}

public void setBatsman(String batsman) {

this.batsman = batsman;

}

public String getBowler() {

return bowler;

}

public void setBowler(String bowler) {

this.bowler = bowler;

}

public String getNonStriker() {

return nonStriker;

}

public void setNonStriker(String nonStriker) {

this.nonStriker = nonStriker;

}

public Delivery(Long over, Long ball, String batsman, String bowler,

String nonStriker) {

this.over = over;

this.ball = ball;

this.batsman = batsman;

this.bowler = bowler;

this.nonStriker = nonStriker;

}

public String toString()

{

return String.format("Over :"+getOver()+"\nBall :"+getBall()+"\nBatsman :"+getBatsman()+"\nBowler :"+getBowler()+"\nNonStriker :"+getNonStriker());

}

}

**Innings Details**

PROGRAM:

import java.util.\*;

import java.io.\*;

public class Main

{

public static void main(String[] args) throws Exception

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

Innings[] in=new Innings[2] ;

String s;

for(int i=0;i<2;i++)

{

if( i==0)

{

s="FirstInnings";

}

else

{

s="SecondInnings";

}

System.out.println("Enter the values for "+s);

System.out.println("Enter the BattingTeam");

String t=br.readLine();

System.out.println("Enter the runs scored");

Long r=Long.parseLong(br.readLine());

in[i]=new Innings(t,r);

}

InningsBO ibo=new InningsBO();

ibo.DisplayAllInningsDetails(in);

}

}

public class Innings

{

String \_battingTeam;

Long \_runs;

public String get\_battingTeam()

{

return \_battingTeam;

}

public void set\_battingTeam(String \_battingTeam)

{

this.\_battingTeam = \_battingTeam;

}

public Long get\_runs()

{

return \_runs;

}

public void set\_runs(Long \_runs)

{

this.\_runs = \_runs;

}

public Innings(String b,Long r)

{

\_battingTeam=b;

\_runs=r;

}

Innings()

{

}

public String toString()

{

String BattingTeam=\_battingTeam;

Long Runs=\_runs;

return (String.format("%-20s%-20s",BattingTeam,Runs)).trim();

}

}

public class InningsBO{

void DisplayAllInningsDetails(Innings[] inningsList)

{

System.out.println("Innings Details");

for(int i=0;i<2;i++)

{

System.out.println(inningsList[i]);

}

}

}

**SESSION 2/CLASSES AND OBJECTS III**

**Wicket details**

PROGRAM:

import java.util.\*;

class Main

{

public static void main(String args[])

{

Scanner scanner=new Scanner(System.in);

System.out.println("Enter the number of wickets");

int n=scanner.nextInt();

scanner.nextLine();

Wicket[] wicket = new Wicket[n];

WicketBO wb = new WicketBO();

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

{

System.out.println("Enter the details of wicket "+(i+1));

String Str=scanner.nextLine();

String[] array = Str.split(",");

Long t=Long.parseLong(array[0]);

Long t1=Long.parseLong(array[1]);

wicket[i] = new Wicket(t,t1,array[2],array[3],array[4]);

}

wb.displayAllWicketDetails(wicket);

System.out.println("Enter the wicket type to be searched");

String type=scanner.next();

wb.displaySpecificWicketDetails(wicket,type);

}

}

class WicketBO

{

public WicketBO(){}

void displayAllWicketDetails(Wicket[] wicketList)

{

int i;

System.out.println("Wicket Details");

for(i=0;i<wicketList.length;i++)

{

System.out.println("Wicket "+(i+1));

System.out.println(wicketList[i].toString());

}

}

void displaySpecificWicketDetails(Wicket[] wicketList,String Type)

{

for(int i=0;i<wicketList.length;i++)

{

if((wicketList[i].getWicketType()).equals(Type))

{

System.out.println("Wicket "+(i+1));

System.out.println(wicketList[i].toString());

}

}

}

}

class Wicket

{

private Long over,ball;

private String wicketType;

private String playerName;

private String bowlerName;

Wicket(){}

Wicket(Long over,Long ball,String wicketType,String playerName,String bowlerName)

{

this.over=over;

this.ball=ball;

this.wicketType=wicketType;

this.playerName=playerName;

this.bowlerName=bowlerName;

}

public void setOver(Long over) {

this.over = over;

}

public Long getOver() {

return over;

}

public void setBall(Long ball) {

this.ball = ball;

}

public Long getBall() {

return ball;

}

public void setWicketType(String wicketType) {

this.wicketType = wicketType;

}

public String getWicketType() {

return wicketType;

}

public void setPlayerName(String playerName) {

this.playerName = playerName;

}

public String getPlayerName() {

return playerName;

}

public void setBowlerName(String bowlerName) {

this.bowlerName = bowlerName;

}

public String getBowlerName() {

return bowlerName;

}

public String toString()

{

return String.format("Over:%s\nBall:%s\nWicket Type:%s\nPlayer Name:%s\nBowler Name:%s",over,ball,wicketType,playerName,bowlerName);

}

}

**Outcome Details**

**PROGRAM:**

import java.util.Scanner;

public class Main {

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

Scanner scanner = new Scanner(System.in);

String date ;

String status ;

String winnerTeam ;

String playerOfMatch ;

System.out.println("Enter the number of matches");

int n=scanner.nextInt();

scanner.nextLine();

Outcome[] outcome=new Outcome[n];

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

{

System.out.println("Enter match "+(i+1)+" details");

System.out.println("Enter the date");

date=scanner.nextLine();

System.out.println("Enter the status");

status=scanner.nextLine();

System.out.println("Enter the winner team");

winnerTeam=scanner.nextLine();

System.out.println("Enter the player of match");

playerOfMatch=scanner.nextLine();

outcome[i]=new Outcome(date,status,winnerTeam,playerOfMatch);

}

OutcomeBO oBo=new OutcomeBO();

oBo.displayAllOutcomeDetails(outcome);

System.out.println("Enter the date to be searhed");

String sd=scanner.next();

oBo.displaySpecificOutcomeDetails(outcome,sd);

}

}

class OutcomeBO

{

OutcomeBO(){}

void displayAllOutcomeDetails(Outcome[] outcomeList)

{

System.out.println("Outcome Details");

System.out.println("Status Winning Team Player Of The Match Date");

for(int i=0;i<outcomeList.length;i++)

{

System.out.println(outcomeList[i].toString());

}

}

void displaySpecificOutcomeDetails(Outcome[] outcomList, String date)

{

System.out.println("Outcome Details");

System.out.println("Status Winning Team Player Of The Match Date");

for(int i=0;i<outcomList.length;i++)

{

if((outcomList[i].getDate()).equals(date))

{

System.out.println(outcomList[i].toString());

}

}

}

}

class Outcome {

private

String status ;

String winnerTeam ;

String playerOfMatch ;

String date;

Outcome(){}

Outcome(String date,String status,String winnerTeam,String playerOfMatch)

{

this.date=date;

this.status=status;

this.winnerTeam=winnerTeam;

this.playerOfMatch=playerOfMatch;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public String getWinnerTeam() {

return winnerTeam;

}

public void setWinnerTeam(String winnerTeam) {

this.winnerTeam = winnerTeam;

}

public String getPlayerOfMatch() {

return playerOfMatch;

}

public void setPlayerOfMatch(String playerOfMatch) {

this.playerOfMatch = playerOfMatch;

}

void setDate(String date)

{

this.date=date;

}

String getDate()

{

return date;}

public String toString()

{

return String.format("%-20s %-20s %-20s %s",status,winnerTeam,playerOfMatch,date);

}

}

**SESSION 1/CLASSES AND OBJECTS IV**

**Team-Player(One to many)**

import java.io.\*;

import java.util.\*;

public class Main {

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

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

Integer n1,n2;

System.out.println("Enter the team count");

n1=Integer.parseInt(br.readLine());

Team [] teamList=new Team[n1];

TeamBO teamBO=new TeamBO();

String t,p;

for(int i=0;i<n1;i++)

{

System.out.println("Enter team "+(i+1)+" details");

t=br.readLine();

teamList[i]=teamBO.createTeam(t);

}

System.out.println("Enter the player count");

n2=Integer.parseInt(br.readLine());

Player [] playerList=new Player[n2];

PlayerBO playerBO=new PlayerBO();

for(int i=0;i<n2;i++)

{

System.out.println("Enter player "+(i+1)+" details");

p=br.readLine();

playerList[i]=playerBO.createPlayer(p, teamList);

}

System.out.println("Enter the player name for which you need to find the team name");

String name=br.readLine();

System.out.println(name+" belongs to "+playerBO.findTeamName(playerList,name));

System.out.println("Enter 2 player names");

String name1=br.readLine();

String name2=br.readLine();

if(playerBO.findWhetherPlayersAreInSameTeam(playerList,name1,name2))

System.out.println("The 2 player are in the same team");

else

System.out.println("The 2 player are in the different teams");

}

}

public class Player {

private

String name;

Team team;

public Player(String name, Team team) {

this.name = name;

this.team = team;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Team getTeam() {

return team;

}

public void setTeam(Team team) {

this.team = team;

}

}

public class Team {

private

String name;

String home;

public Team(String name, String home) {

this.name = name;

this.home = home;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getHome() {

return home;

}

public void setHome(String home) {

this.home = home;

}

}

public class PlayerBO {

public Player createPlayer(String data, Team[] teamList)

{

String[] arr = data.split(",");

String name=arr[0];

Team team=null;

for(Team t : teamList)

{

if(t.getName().equals(arr[1]))

team=t;

}

Player player=new Player(name, team);

return player;

}

public String findTeamName(Player[] playerList, String playername)

{

for(Player p:playerList)

{

if(p.getName().equals(playername))

return p.getTeam().getName();

}

return "";

}

public Boolean findWhetherPlayersAreInSameTeam (Player[] playerList, String playername1, String playername2)

{

String player1TeamName="";

String player2TeamName="";

for(Player p : playerList)

{

if(p.getName().equals(playername1))

player1TeamName=p.getTeam().getName();

if(p.getName().equals(playername2))

player2TeamName=p.getTeam().getName();

}

return player1TeamName.equals(player2TeamName);

}

}

public class TeamBO {

public Team createTeam(String data)

{

String arr[] = data.split(",");

Team t =new Team(arr[0], arr[1]);

return t;

}

}

**Match-Outcome(one to one)**

PROGRAM:

public class Match {

                String date;

                String teamOne;

                String teamTwo;

                String venue;

                Outcome outcome;

                public String getDate() {

                                return date;

                }

                public void setDate(String date) {

                                this.date = date;

                }

                public String getTeamOne() {

                                return teamOne;

                }

                public void setTeamOne(String teamOne) {

                                this.teamOne = teamOne;

                }

                public String getTeamTwo() {

                                return teamTwo;

                }

                public void setTeamTwo(String teamTwo) {

                                this.teamTwo = teamTwo;

                }

                public String getVenue() {

                                return venue;

                }

                public void setVenue(String venue) {

                                this.venue = venue;

                }

                public Outcome getOutcome() {

                                return outcome;

                }

public void setOutcome(Outcome outcome) {

                                this.outcome = outcome;

                }

public Match(String date, String teamOne, String teamTwo, String venue,

                                                Outcome outcome) {

                                this.date = date;

                                this.teamOne = teamOne;

                                this.teamTwo = teamTwo;

                                this.venue = venue;

                                this.outcome = outcome;

                }

              public Match()

                {

                }

                public String toString()

                {

                                return    String.format("%-15s %-15s %-15s %-15s %-30s\n",date,teamOne,teamTwo,venue,outcome);

                }

}

  public class Outcome {

                String status;

String winnerTeam;

                public String getStatus() {

                                return status;

                }

                public void setStatus(String status) {

                                this.status = status;

                }

                public String getWinnerTeam() {

                                return winnerTeam;

                }

                public void setWinnerTeam(String winnerTeam) {

                                this.winnerTeam = winnerTeam;

                }

                public Outcome(String status, String winnerTeam) {

                                this.status = status;

                                this.winnerTeam = winnerTeam;

                }

                public Outcome()

                {

                }

                public String toString()

                {

return String.format("%-15s %-15s",status,winnerTeam);

                }

}

  import java.io.\*;

import java.util.\*;

public class Main {

    public static void main(String[] args) throws IOException {

                                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

                                String date;

                                String teamOne;

                                String teamTwo;

                                String venue;

                                String status;

                                String winnerTeam;

                                System.out.println("Enter number of matches");

                                Integer n=Integer.parseInt(br.readLine());

                                Match match=new Match();

                                Outcome outcome=new Outcome();

                                ArrayList<Match> matchList = new ArrayList<Match>();

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

                                {

                                                System.out.println("Enter match "+(i+1)+" details:");

                                                System.out.println("Enter match date");

                                        date=br.readLine();

                                                System.out.println("Enter team one");

                                                teamOne=br.readLine();

                                                System.out.println("Enter team two");

                                                teamTwo=br.readLine();

                                                System.out.println("Enter venue");

                                                venue=br.readLine();

                                                System.out.println("Enter status");

                                                status=br.readLine();

                                                System.out.println("Enter winner Team");

                                                winnerTeam=br.readLine();

                                                outcome=new Outcome(status,winnerTeam);

                                                match=new Match(date,teamOne,teamTwo,venue,outcome);

                                                matchList.add(match);

                                }

                                MatchBO matchBO=new MatchBO();

                                while(true)

                                {

                                                System.out.println("Menu");

                                                System.out.println("1.View match details");

                                                System.out.println("2.Filter match details with outcome status");

                                                System.out.println("3.Filter match details with outcome winner team");

                                                System.out.println("4.Exit");

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

                                                Integer choice=Integer.parseInt(br.readLine());

                                                switch(choice)

                                                {

                                                case 1:

                                                                matchBO.printAllMatchDetails(matchList);

                                                                break;

                                                case 2:

                                                                System.out.println("Enter outcome status");

                                                                status=br.readLine();

                                                                matchBO.printMatchDetailsWithOutcomeStatus(matchList,status);

                                                                break;

                                                case 3:

                                                      System.out.println("Enter outcome winner team");

                                                     winnerTeam=br.readLine();

                                                                matchBO.printMatchDetailsWithOutcomeWinnerTeam(matchList,winnerTeam);

                                                                break;

                                                                default:

                                                                                System.exit(0);

                                                }

                                }

                }

}

  import java.util.\*;

class MatchBO {

                public void printAllMatchDetails(List<Match> matchList)

                {

                                System.out.println("Match Details");

        System.out.println(String.format("%-15s %-15s %-15s %-15s %-15s %s","Date","Team1","Team2","Venue","Status","Winner"));

       for(Match m:matchList){

        System.out.printf("%-15s %-15s %-15s %-15s %-15s %-15s\n",m.date,m.teamOne,m.teamTwo,m.venue,m.outcome.status,m.outcome.winnerTeam);

    }

                }

                public void printMatchDetailsWithOutcomeStatus(List<Match> matchList, String outcomeStatus) {

                                int c=0;

         for(Match m:matchList)

         {

         if((m.outcome.status).equals(outcomeStatus))

                {

                    System.out.println("Match Details");

                    System.out.println(""+String.format("%-15s %-15s %-15s %-15s %-15s %s","Date","Team1","Team2","Venue","Status","Winner"));

                      System.out.printf("%-15s %-15s %-15s %-15s %-15s %-15s\n",m.date,m.teamOne,m.teamTwo,m.venue,m.outcome.status,m.outcome.winnerTeam);

                 c++;

                }}

         if(c==0)

         {

             System.out.println("Status not found");

         }

                }

                public void printMatchDetailsWithOutcomeWinnerTeam(List<Match> matchList, String outcomeWinnerTeam)

                {int d=0;

          for(Match m:matchList)

         {

         if((m.outcome.winnerTeam).equals(outcomeWinnerTeam))

                {

                    System.out.println("Match Details");

                    System.out.println(""+String.format("%-15s %-15s %-15s %-15s %-15s %s","Date","Team1","Team2","Venue","Status","Winner"));

                            System.out.printf("%-15s %-15s %-15s %-15s %-15s %-15s\n",m.date,m.teamOne,m.teamTwo,m.venue,m.outcome.status,m.outcome.winnerTeam);

                 d++;

                }}

         if(d==0)

         {

             System.out.println("Winner Team not found");

         }

                }

}

**SESSION 2/CLASSES AND OBJECTS IV**

**Match Class , Venue Class, City Class and BO Classes**

PROGRAM:

class Venue {

    private String name;

    private City city;

    public Venue(String name, City city) {

        this.name = name;

        this.city = city;

    }

    public Venue() {

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public City getCity() {

        return city;

    }

    public void setCity(City city) {

        this.city = city;

    }

    @Override

    public String toString() {

        return "Venue{" + "name=" + name + ", city=" + city + '}';

    }

}

class City {

    private String name;

   public City(String name) {

        this.name = name;

    }

    public City() {

    }

    public boolean equals(City obj) {

        return this.name.equals(obj.name);

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

   @Override

    public String toString() {

        return "City{" + "name=" + name + '}';

    }

}

import java.util.\*;

public class Main {

                                public static void main(String args[]) {

                                                Scanner sc = new Scanner(System.in);

                                                System.out.println("Enter the city count");

                                                Integer ccount = sc.nextInt();

                                                sc.nextLine();

                                                City cityObj = new City();

                                                Venue venueObj = new Venue();

             Match matchObj = new Match();

                                                City [] cityList = new City[ccount];

                                             CityBO cityBO = new CityBO();

                                                Integer i;

                                                String data;

                                                String name;

                                                for(i=0;i<ccount;i++) {

                                                 System.out.println("Enter city "+(i+1)+" details");

                                                                data = sc.nextLine();

                                                                cityList[i] = cityBO.createCity(data);

                                                }

                                                System.out.println("Enter the venue count");

                                                Integer scount = sc.nextInt();

                                                sc.nextLine();

                                                Venue [] venueList = new Venue[scount];

                                                VenueBO venueBO = new VenueBO();

                                                for(i=0;i<scount;i++) {

                                              System.out.println("Enter venue "+(i+1)+" details");

                                               data = sc.nextLine();

                                               venueList[i] = venueBO.createVenue(data, cityList);

                                                }

                                                System.out.println("Enter the match count");

                                                Integer matchCount = sc.nextInt();

                                                sc.nextLine();

                                      Match [] matchList = new Match[matchCount];

                                       MatchBO matchBO = new MatchBO();

                                       for(i=0;i<matchCount;i++) {

                                      System.out.println("Enter match "+(i+1)+" details");

                                       data = sc.nextLine();

                                       matchList[i] = matchBO.createMatch(data, venueList);

                                                }

                                           System.out.println("Menu :");

                                          System.out.print("1)Find Venue\n2)Find All Matches In A Specific Venue\n");

                                                System.out.println("Type 1 or 2");

                                                do

                                                {

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

                                                Integer choice = sc.nextInt();

                                                sc.nextLine();

                                                switch(choice) {

                                                                case 1:

                                                                System.out.println("Enter Match Date");

                                                                name = sc.nextLine();

                                                                matchBO.findVenue(name, matchList);

                                                                break;

                                                                case 2:

                                                                System.out.println("Enter Venue Name");

                                                                name = sc.nextLine();

                                                                matchBO.findAllMatchesInGivenVenue(name, matchList);

                                                                break;

                                                                case 3:

                                                                break;

                                                }

                            System.out.println("Do you want to continue? Type Yes or No");

                                }while(sc.nextLine().equalsIgnoreCase("Yes"));

                                }

                }

class MatchBO {

    public Match createMatch(String data, Venue[] venueList) {

        String []b=data.split(",");

        Match m=null;

        for(Venue v : venueList)

        {

            if(v.getName().equals(b[3]))

            {

                m=new Match(b[0],b[1],b[2],v);

            }

        }

        return m;

    }

    public void findVenue(String date, Match[] matchList) {

        for(Match m : matchList)

        {

            if(m.getDate().equals(date))

            {

                System.out.println("Match was held at "+m.getVenue().getName());

            }

        }

    }

    public void findAllMatchesInGivenVenue(String sname, Match[] matchList) {

        System.out.println("Matches in venue "+sname+" are");

        System.out.println(String.format("%-15s%-15s%s","Date","TeamOne","TeamTwo"));

        for(Match m : matchList)

        {

            if(m.getVenue().getName().equals(sname))

            {

                System.out.println(String.format("%-15s%-15s%s",m.getDate(),m.getTeamone(),m.getTeamtwo()));

            }

        }

    }

}

class Match {

    private String date;

private String teamone;

private String teamtwo;

    private Venue venue;

    public Match(String date,String teamone,String teamtwo, Venue venue) {

        this.date = date;

        this.venue = venue;

this.teamone = teamone;

this.teamtwo = teamtwo;

    }

    @Override

    public boolean equals(Object obj) {

        final Match other = (Match) obj;

        return this.date.equals(other.getDate());

    }

    public Match() {

    }

    @Override

    public String toString() {

        return "Match{" + "date=" + date + ", venue=" + venue + '}';

    }

    public String getDate() {

        return date;

    }

    public void setDate(String date) {

        this.date = date;

    }

    public String getTeamone() {

        return teamone;

    }

    public void setTeamone(String teamone) {

        this.teamone = teamone;

    }

    public String getTeamtwo() {

        return teamtwo;

    }

    public void setTeamtwo(String teamtwo) {

        this.teamtwo = teamtwo;

    }

    public Venue getVenue() {

        return venue;

    }

    public void setVenue(Venue venue) {

        this.venue = venue;

    }

}

class CityBO  {

    public City createCity(String data) {

        City c=new City(data);

         return c;

    }

}

class VenueBO  {

    public Venue createVenue(String data, City[] cityList) {

        String[] a=data.split(",");

       Venue v=null;

       for(City c : cityList)

       {

           if(c.getName().equals(a[1]))

           v=new Venue(a[0],c);

       }

      return v;

    }

}

**STRINGS SESSION 1**

**charAt**

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner sc= new Scanner(System.in);

System.out.println("Enter team1");

String x=sc.nextLine();

System.out.println("Enter team2");

String y=sc.nextLine();

System.out.println("Enter third character");

char z=sc.next().charAt(0);

if(x.charAt(2)==z)

System.out.println("Winner Team : "+x);

else if(y.charAt(2)==z)

System.out.println("Winner Team : "+y);

}

}

**EqualsIgnoreCase**

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter venue1");

String s1=sc.nextLine();

System.out.println("Enter venue2");

String s2=sc.nextLine();

if(s2.equalsIgnoreCase(s1))

{

System.out.println("Both the venues are same.");

}

else

{

System.out.println("Both the venues are different.");

}

}

}

**IndexOf,LastIndexOf**

import java.io.\*;

public class Main

{

public static void main(String[] args) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of players");

int x=Integer.parseInt(br.readLine());

String s1[]=new String[x];

for(int i=0;i<x;i++)

{

s1[i]=br.readLine();

}

System.out.println("Player of the Match:");

for(int i=0;i<x;i++)

{

if(s1[i].indexOf("a")==s1[i].lastIndexOf("a") && s1[i].indexOf("a")>=1)

{

System.out.println(s1[i]);

}

}

}

}

**SubSequence**

import java.io.\*;

public class Main

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter team name");

String name=br.readLine();

System.out.println("Enter starting index of the sequence");

int n1=Integer.parseInt(br.readLine());

System.out.println("Enter ending index of the sequence");

int n2=Integer.parseInt(br.readLine());

System.out.println(name.subSequence(n1,n2));

}

}

**SubString**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter Player name");

String s=sc.nextLine();

System.out.println("Enter starting index");

int n=sc.nextInt();

String obj=new String();

System.out.println("Short name of "+s+": "+s.substring(n));

}

}

**startswith,endswith**

import java.io.\*;

class Main

{

public static void main(String A[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int i;

System.out.println("Enter the number of players");

int n=Integer.parseInt(br.readLine());

String a[]=new String[n];

System.out.println("Enter the player name");

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

a[i]=br.readLine();

System.out.println("Player name starting with 'M' or Ending with 'a'");

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

{

if(a[i].startsWith("M")||a[i].endsWith("a"))

System.out.println(a[i]);

}

}

}

**RegionMatches**

import java.io.\*;

public class Main

{

public static void main(String[] args) throws Exception

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter player names");

String s1=br.readLine();

String s2=br.readLine();

if(s1.regionMatches(0,s2,0,7))

{

System.out.println("Both the players names starts with "+s1.substring(0,7));

}

else

System.out.println("Both the players names does not starts with "+s1.substring(0, 7));

}

}

**Replace**

import java.io.\*;

public class Main

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter team details");

String s=br.readLine();

System.out.println("After replacement");

System.out.println(s.replaceAll("Captain","Skipper"));

}

}

**Contains**

import java.io.\*;

public class Main

{

public static void main(String args[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter number of players");

int n=Integer.parseInt(br.readLine());

String a[]=new String[n];

System.out.println("Enter player names");

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

{

a[i]=br.readLine();

}

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

{

if(a[i].contains("Sharma"))

{

System.out.println(a[i]);

}

}

}

}

**Transformer Optimus Prime’s Quest**

import java.io.\*;

public class Main {

public static void main(String[] args)throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int a=Integer.parseInt(br.readLine());

String s=br.readLine();

int x=0,y=0;

char arr[]=s.toCharArray();

for(int i=0;i<a;i++)

{

if(arr[i]=='D')

y=y-1;

else if(arr[i]=='R')

x=x+1;

else if(arr[i]=='L')

x=x-1;

else if(arr[i]=='U')

y=y+1;

}

System.out.println(x+" "+y);

}

}

**STRINGS SESSION 2**

**Validation -- I**

import java.text.ParseException;

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

String s1=s.nextLine();

if(UserMainCode.validateDate(s1))

System.out.println("Valid");

else

System.out.println("Invalid");

}

}

import java.util.\*;

import java.text.\*;

public class UserMainCode

{

public static boolean validateDate(String s)

{

SimpleDateFormat d=new SimpleDateFormat("dd-MM-yyyy");

d.setLenient(false);

try

{

Date dt=d.parse(s);

}

catch(Exception e)

{

return false;

}

return true;

}

}

**Validation -- IV**

import java.util.\*;

public class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

if(UserMainCode.validateCity(s))

System.out.println("Valid");

else

System.out.println("Invalid");

}

}

import java.util.\*;

public class UserMainCode

{

public static boolean validateCity(String s)

{

if((s.matches("[A-Z]{1}[a-z]{1}(.\*)")) && (s.matches("(.\*)[a-z]{2}")))

return true;

else

return false;

}

}

**Validation -- VII**

import java.io.\*;

public class Main

{

public static void main(String[] args) throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String s=br.readLine();

if(UserMainCode.validatePlayer(s))

System.out.println("Valid");

else

System.out.println("Invalid");

}

}

import java.util.\*;

public class UserMainCode

{

public static boolean validatePlayer(String s)

{

int count=0;

char arr[]=s.toCharArray();

if(s.contains("a"))

{

for(int i=0;i<s.length();i++)

{

if((i+1)%2==0)

{

if(arr[i]=='a')

{

count++;

break;

}

}

}

}

if(count==0)

return true;

else return false;

}

}

**Clara and her monthly budget**

import java.util.\*;

public class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

int x=0,y=3;

int count=0;

for(int i=0;i<s.length()-2;i++)

{

if(y<s.length()+1)

{

int t=Integer.parseInt(s.substring(x,y));

if(t%4==0)

count ++;

}

else

{

int t=Integer.parseInt(s.substring(x));

if(t%4==0)

count++;

}

x++;

y++;

}

System.out.println(count);

}

}

**DATE API/SESSION 1**

**Display Date**

import java.text.ParseException;

import java.util.\*;

public class Main1

{

public static void main ( String args[] ) throws ParseException

{

Scanner sc = new Scanner( System.in );

String s = sc.nextLine();

UserMainCode1.displayDate(s);

}

}

import java.util.\*;

import java.text.\*;

public class UserMainCode1

{

public static void displayDate( String s) throws ParseException

{

SimpleDateFormat sdf = new SimpleDateFormat("MMMM dd,yyyy");

Date d = sdf.parse(s);

SimpleDateFormat sdf1 = new SimpleDateFormat("yyyy-MM-dd");

String s1 = sdf1.format(d);

System.out.println(s1);

}

}

**Extract Date and time**

import java.text.ParseException;

import java.util.Scanner;

public class Main

{

public static void main(String arg[]) throws ParseException

{

System.out.println("Enter String in this format(yyyy-MM-DD HH:mm:ss)");

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

UserMainCode.displayDateTime(s);

}

}

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.\*;

public class UserMainCode

{

public static void displayDateTime(String s) throws ParseException

{

SimpleDateFormat obj=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date d=obj.parse(s);

String ob=new SimpleDateFormat("dd/MM/yyyy, H:mm:ss").format(d);

System.out.println(ob);

}

}

**Day Of The Year**

import java.text.ParseException;

import java.util.Scanner;

public class Main

{

public static void main(String arg[]) throws ParseException

{

UserMainCode.displayDay();

}

}

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.Scanner;

public class UserMainCode

{

public static void displayDay() throws ParseException

{

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

SimpleDateFormat ob=new SimpleDateFormat("yyyy-MM-dd");

Date d=ob.parse(s);

Calendar c=Calendar.getInstance();

c.setTime(d);

System.out.println("Day of year : "+c.get(Calendar.DAY\_OF\_YEAR));

}

}

**Difference between two years**

import java.io.\*;

import java.text.\*;

public class Main

{

public static void main ( String args[] ) throws IOException,ParseException

{

BufferedReader br = new BufferedReader( new InputStreamReader (System.in ));

String s1 = br.readLine();

String s2 = br.readLine();

UserMainCode.displayDetails(s1, s2);

}

}

import java.util.\*;

import java.text.\*;

public class UserMainCode

{

public static void displayDetails( String s1, String s2) throws ParseException

{

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

Date d1 = sdf.parse(s1);

Date d2 = sdf.parse(s2);

Calendar c1 = Calendar.getInstance();

Calendar c2 = Calendar.getInstance();

c1.setTime(d1);

c2.setTime(d2);

int year = Math.abs(c2.get(Calendar.YEAR) - c1.get(Calendar.YEAR));

int month = Math.abs(c2.get(Calendar.MONTH) - c1.get(Calendar.MONTH));

if((c1.get(Calendar.DAY\_OF\_MONTH)) > c2.get(Calendar.DAY\_OF\_MONTH) )

month = month -1;

System.out.println("Difference between "+s1+" and "+s2+": "+year+" Years and "+month+" Months");

}

}

**Name Of the Day**

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.\*;

public class Main

{

public static void main(String arg[]) throws ParseException

{

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

UserMainCode.displayDay(s);

}

}

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class UserMainCode

{

public static void displayDay(String s) throws ParseException

{

SimpleDateFormat ob=new SimpleDateFormat("yyyy-MM-dd");

Date d=ob.parse(s);

SimpleDateFormat objj=new SimpleDateFormat("EEEE");

String ss=objj.format(d);

System.out.println(ss);

}

}

**DATE API/SESSION 2**

**Leap Year**

**import java.text.ParseException;**

import java.util.Scanner;

public class Main

{

public static void main(String[] args) throws ParseException

{

Scanner x=new Scanner(System.in);

System.out.println("Enter the year");

int s=x.nextInt();

UserMainCode.displayAge(s);

}

}

public class UserMainCode

{

public static void displayAge(int s)

{

if(s%4==0 && s%100!=0 || s%400==0)

{

System.out.println(s+" is leap year");

}

else

{

System.out.println(s+" is not leap year");

}

}

}

**Date before months**

import java.text.ParseException;

import java.util.Scanner;

public class Main

{

                public static void main(String[] args) throws ParseException

                {

                                Scanner x=new Scanner(System.in);

                                String s=x.nextLine();

                                UserMainCode.displayAge(s);

                }

}

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

public class UserMainCode

{

public static void displayAge(String s) throws ParseException

{

                SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

                Date d1=(Date) sdf.parse(s);

                Calendar c=Calendar.getInstance();

                c.setTime(d1);

                c.add(Calendar.MONTH,-20);

                Date d=c.getTime();

                String ss=sdf.format(d);

                System.out.println("20 months before "+s+" will be "+ss);

}

}

**Calculate Age**

import java.util.\*;

import java.text.\*;

import java.time.\*;

class Main

{

public static void main(String[] args)throws Exception

{

Scanner ip =new Scanner(System.in);

String string1 = ip.nextLine();

String string2 = ip.nextLine();

UserMainCode.displayAge(string1,string2);

}

}

class UserMainCode

{

public static void displayAge(String string1,String string2)throws Exception

{

LocalDate date = LocalDate.parse(string1);

LocalDate today = LocalDate.parse(string2);

Period diff = Period.between(date,today);

long days = diff.getDays();

long years = diff.getYears();

long months = diff.getMonths();

System.out.println("I am "+years+" years, "+months+" months and "+days +" days old.");

}

}

**INHERITANCE AND ABSTRACT CLASSES**

**Abstract Class I – Shape**

class Circle extends Shape

{

    private int radius;

                public int getRadius() {

                                return radius;

                }

                public void setRadius(int radius) {

                                this.radius = radius;

                }

                Circle(String name,int radius)

                {

                                super(name);

                                this.name=name;

                                this.radius=radius;

                }

public float calculateArea()

{

                float y=(float)(3.14\*radius\*radius);

                return y;

}

}

abstract class Shape

{

    protected String name;

                public String getName() {

                                return name;

                }

                public void setName(String name) {

                                this.name = name;

                }

                Shape(String name)

                {

                                this.name=name;

                }

                public abstract float calculateArea();

}

import java.text.DecimalFormat;

import java.util.\*;

public class Main {

    public static void main(String[] args)

                {

                                Scanner sc=new Scanner(System.in);

                                System.out.println("Circle");

                                System.out.println("Square");

                                System.out.println("Rectangle");

                                System.out.println("Enter the shape name");

                                String s1=sc.next();

                                if(s1.equals("Circle"))

                                {

                                System.out.println("Enter the radius");

            int r=sc.nextInt();

           Shape cl=new Circle(s1,r);

            float p=cl.calculateArea();

            DecimalFormat df=new DecimalFormat("0.00");

          System.out.println("Area of Circle is "+df.format(p));

                                }

                                if(s1.equals("Square"))

                                {

                                                System.out.println("Enter the side");

                                                int n1=sc.nextInt();

                                                Shape sq=new Square(s1,n1);

                                                float p1=sq.calculateArea();

                                    System.out.printf("Area of Square is %.2f",p1);

                                }

                                if(s1.equals("Rectangle"))

                                {

                                                System.out.println("Enter the length");

                                                int n2=sc.nextInt();

                                                System.out.println("Enter the breadth");

                                                int n3=sc.nextInt();

                                                Shape r=new Rectangle(s1,n2,n3);

                                                float p2=r.calculateArea();

                                                DecimalFormat df=new DecimalFormat("0.00");

                                                System.out.println("Area of Rectangle is "+df.format(p2));

                                }

                }

}

public class Rectangle extends Shape

{

    private int length;

                private int breadth;

                public int getLength() {

                                return length;

                }

                public void setLength(int length) {

                                this.length = length;

                }

                public int getBreadth() {

                                return breadth;

                }

                public void setBreadth(int breadth) {

                                this.breadth = breadth;

                }

                Rectangle(String name,int length,int breadth)

                {

                                super(name);

                                this.length=length;

                                this.breadth=breadth;

                }

                public float calculateArea()

                {

                                float rec=(length\*breadth);

                                return rec;

                }

}

public class Square extends Shape

{

  private int side;

public int getSide() {

    return side;

}

public void setSide(int side) {

                this.side = side;

}

Square(String name,int side)

{

                super(name);

                this.name=name;

                this.side=side;

}

public float calculateArea()

{

                float sqr=(side\*side);

                return sqr;

}

}

**Abstract Class II – Card**

public abstract class Card

{

  protected String holderName;

  protected String cardNumber;

  protected String expiryDate;

public String getHolderName() {

    return holderName;

}

public void setHolderName(String holderName) {

                this.holderName = holderName;

}

public String getCardNumber() {

                return cardNumber;

}

public void setCardNumber(String cardNumber) {

                this.cardNumber = cardNumber;

}

public String getExpiryDate() {

                return expiryDate;

}

public void setExpiryDate(String expiryDate) {

                this.expiryDate = expiryDate;

}

Card(String holderName,String cardNumber,String expiryDate)

{

                this.holderName=holderName;

                this.cardNumber=cardNumber;

                this.expiryDate=expiryDate;

}

}

public class PaybackCard extends Card

{

    private Integer pointsEarned;

                public Integer getPointsEarned() {

                                return pointsEarned;

                }

                public void setPointsEarned(Integer pointsEarned) {

                                this.pointsEarned = pointsEarned;

                }

                public Double getTotalAmount() {

                                return totalAmount;

                }

                public void setTotalAmount(Double totalAmount) {

                                this.totalAmount = totalAmount;

                }

                private Double totalAmount;

                PaybackCard(String holderName, String cardNumber,String expiryDate,int pointsEarned,double totalAmount)

                {

                                super(holderName,cardNumber,expiryDate);

                                this.pointsEarned=pointsEarned;

                                this.totalAmount=totalAmount;

                }

}

public class MembershipCard extends Card

{

    private Integer rating;

                public Integer getRating() {

                                return rating;

                }

                public void setRating(Integer rating) {

                                this.rating = rating;

                }

        MembershipCard(String holderName,String cardNumber, String expiryDate,int rating)

        {

               super(holderName,cardNumber,expiryDate);

               this.holderName=holderName;

               this.cardNumber=cardNumber;

               this.expiryDate=expiryDate;

               this.rating=rating;

        }}

import java.io.\*;

public class Main {

    public static void main(String[] args) throws IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Select the Card");

        System.out.println("1.Payback Card");

        System.out.println("2.Membership Card");

        Integer selCard = Integer.parseInt(br.readLine());

        System.out.println("Enter the Card Details:");

        String[] dataList1 = br.readLine().split("\\|");

        String holderName = dataList1[0];

        String cardNumber = dataList1[1];

        String expiryDate = dataList1[2];

        if(selCard == 1){

            System.out.println("Enter points in card");

            Integer points = Integer.parseInt(br.readLine());

            System.out.println("Enter Amount");

            Double amt = Double.parseDouble(br.readLine());

            PaybackCard payIns = new PaybackCard(holderName, cardNumber, expiryDate, points, amt);

            System.out.println(payIns.getHolderName()+"'s Payback Card Details:");

            System.out.println("Card Number "+payIns.getCardNumber());

            System.out.println("Points Earned "+payIns.getPointsEarned());

            System.out.println("Total Amount "+payIns.getTotalAmount());

        }else if(selCard == 2){

            System.out.println("Enter rating in card");

            Integer rating = Integer.parseInt(br.readLine());

            MembershipCard memIns = new MembershipCard(holderName, cardNumber, expiryDate, rating);

            System.out.println(memIns.getHolderName()+"'s Membership Card Details:");

            System.out.println("Card Number "+memIns.getCardNumber());

            System.out.println("Rating "+memIns.getRating());

        }else{

            System.out.println("Invalid card");

        }

        }

}

**METHOD OVERRIDING , METHOD OVERLOADING,INTERFACES**

**Method Overloading / Illustration(Match)**

import java.io.\*;

import java.text.\*;

import java.util.\*;

public class Main

{

public static void main( String args[] ) throws IOException,ParseException

{

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

String s1,s2,s3;

int n;

long r;

System.out.println("Menu\n1.Match Date\n2.Match Venue\n3.Match Outcome");

n= Integer.parseInt( br.readLine());

Match m = new Match();

switch(n)

{

case 1:

System.out.println("Enter the date of the match");

s1 = br.readLine();

Date dt = sdf.parse(s1);

m.displayMatchDetails(dt);

break;

case 2:

System.out.println("Enter venue of the match");

s2 = br.readLine();

m.displayMatchDetails(s2);

break;

case 3:

System.out.println("Enter the winner team of the match");

s3 = br.readLine();

System.out.println("Enter the number of runs");

r = Long.parseLong( br.readLine());

m.displayMatchDetails(s3,r);

}

br.close();

}

}

import java.util.\*;

import java.text.\*;

public class Match

{

public void displayMatchDetails(Date matchDate) throws ParseException

{

SimpleDateFormat obj = new SimpleDateFormat("MM-dd-yyyy");

System.out.println("Match Date : "+obj.format(matchDate));

}

public void displayMatchDetails(String venue)

{

String s[] = venue.split(",");

System.out.println("Match Venue :");

System.out.print("Stadium : "+s[0]+"\nCity : "+s[1]);

}

public void displayMatchDetails(String winnerTeam,long runs)

{

System.out.println("Match Outcome :");

System.out.print(winnerTeam+" won by "+runs+" runs");

}

}

**Method Overriding  
Player-International Player**

import java.util.\*;

public class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter player name ");

String name=sc.nextLine();

System.out.println("Enter player country");

String country=sc.nextLine();

System.out.println("Enter the Cap number");

String capNumber=sc.nextLine();

System.out.println("Enter the number of test appearnace");

Long noOfTestAppearnace=sc.nextLong();

System.out.println("Enter the number of ODI appearnace");

Long noOfODIAppearnace=sc.nextLong();

InternationalPlayer obj=new InternationalPlayer(name,country,capNumber,noOfTestAppearnace,noOfODIAppearnace);

obj.displayDetails();

}

}

  class Player

{

protected

String name;

String country;

public Player(String name,String country)

{

this.name=name;

this.country=country;

}

public void displayDetails()

{

System.out.println("Player Details:");

System.out.println("Player name : "+name);

System.out.println("Country : "+country);

}

}

class InternationalPlayer extends Player

{

private

String capNumber;

Long noOfTestAppearnace;

Long noOfODIAppearnace;

public InternationalPlayer(String name,String country,String capNumber,Long noOfTestAppearnace,Long noOfODIAppearnace)

{

super(name,country);

this.name=name;

this.country=country;

this.capNumber=capNumber;

this.noOfTestAppearnace=noOfTestAppearnace;

this.noOfODIAppearnace=noOfODIAppearnace;

}

public void displayDetails()

{

super.displayDetails();

System.out.println("Cap number : "+capNumber);

System.out.println("Number of test appearnace : "+noOfTestAppearnace);

System.out.println("Number of ODI appearnace : "+noOfODIAppearnace);

}

}

**INTERFACES /SESSION 2**

**Simple Interface**

import java.util.\*;

interface IPlayerStatistics

{

public void displayPlayerStatistics();

}

class Player implements IPlayerStatistics

{

private String name;

private String teamName;

private Integer noOfMatches;

private Long totalRunsScored;

private Integer noOfWicketsTaken;

public Player(String name,String teamName,Integer noOfMatches,Long totalRunsScored,Integer noOfWicketsTaken )

{

this.name=name;

this.teamName=teamName;

this.noOfMatches=noOfMatches;

this.totalRunsScored=totalRunsScored;

this.noOfWicketsTaken=noOfWicketsTaken;

}

public void displayPlayerStatistics()

{

System.out.println("Player Details");

System.out.println("Player name : "+this.name);

System.out.println("Team name : "+this.teamName);

System.out.println("No of matches : "+this.noOfMatches);

System.out.println("Total runsscored : "+this.totalRunsScored);

System.out.println("No of wickets taken : "+this.noOfWicketsTaken);

}

}

class Main

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println(" Enter player name");

String name=sc.nextLine();

System.out.println("Enter team name");

String team=sc.nextLine();

System.out.println("Enter number of matches played");

int m=sc.nextInt();

System.out.println("Enter total runs scored");

long r=sc.nextInt();

System.out.println("Enter number of wickets taken");

int n=sc.nextInt();

IPlayerStatistics obj=new Player(name,team,m,r,n);

obj.displayPlayerStatistics();

}

}

**Interface III**

public class Batsman extends CricketPlayer implements IPlayerStatistics

{

                private String runs;

                public Batsman(String name, String teamName, String noOfMatches, String runs) {

                                super(name, teamName, noOfMatches);

                                this.runs = runs;

                }

                public Batsman() {

                                super();

                                // TODO Auto-generated constructor stub

                }

                public Batsman(String runs) {

                                super();

                                this.runs = runs;

                }

                @Override

                public void displayPlayerStatistics() {

                                // TODO Auto-generated method stub

                                System.out.println("Player name : "+super.getName());

                                System.out.println("Team name : "+super.getTeamName());

                                System.out.println("No of matches : "+super.getNoOfMatches());

                                System.out.println("Runs scored : "+this.runs);

                }

                public String getRuns() {

                                return runs;

                }

                public void setRuns(String runs) {

                                this.runs = runs;

                }

}

import java.util.Scanner;

public class Main

{

                public static void main(String[] args) {

                                Scanner sc=new Scanner(System.in);

                                String s="yes";

                                while (s.equalsIgnoreCase("yes"))

                                {

                                                System.out.print("Menu\n1.Bowler\n2.Batsman\n3.WicketKeeper\n4.AllRounder\nEnter your choice\n");

                                                Integer choice=Integer.parseInt(sc.nextLine());

                                switch(choice)

                                {

                                case 1:System.out.println("Enter the Bowler details");System.out.println("Enter player name");String name=sc.nextLine();

                                System.out.println("Enter team name");String teamName=sc.nextLine();System.out.println("Enter number of matches played");

                                String noOfMatches=sc.nextLine();System.out.println("Enter number of wickets taken");String noOfWickets=sc.nextLine();

                                Bowler b=new Bowler(name, teamName, noOfMatches, noOfWickets);

                                b.displayPlayerStatistics();

                                break;

                                case 2:System.out.println("Enter the Batsman details");System.out.println("Enter player name"); name=sc.nextLine();

                                System.out.println("Enter team name"); teamName=sc.nextLine();System.out.println("Enter number of matches played");

                                noOfMatches=sc.nextLine();System.out.println("Enter the runs scored"); String runs=sc.nextLine();

                                Batsman bt=new Batsman(name, teamName, noOfMatches, runs);

                                bt.displayPlayerStatistics();

                                break;

                                case 3:

                                                System.out.println("Enter the WicketKeeper details");System.out.println("Enter player name"); name=sc.nextLine();

                                                System.out.println("Enter team name"); teamName=sc.nextLine();System.out.println("Enter number of matches played");

                                                noOfMatches=sc.nextLine();System.out.println("Enter number of catches taken"); String noOfCatches=sc.nextLine();

                                                System.out.println("Enter number of stumpings");String noOfStumpings=sc.nextLine();System.out.println("Enter number of dismissals");

                                                String noOfDismissals=sc.nextLine();System.out.println("Enter the runs scored");runs=sc.nextLine();

                         WicketKeeper w=new WicketKeeper(name, teamName, noOfMatches, noOfCatches, noOfStumpings, runs, noOfDismissals);w.displayPlayerStatistics();break;

                                case 4:System.out.println("Enter the AllRounder details");System.out.println("Enter player name"); name=sc.nextLine();

                                System.out.println("Enter team name"); teamName=sc.nextLine();System.out.println("Enter number of matches played");

                                noOfMatches=sc.nextLine();System.out.println("Enter the runs scored");  runs=sc.nextLine();

                                System.out.println("Enter number of wickets taken ");noOfWickets=sc.nextLine();

                                AllRounder ar=new AllRounder(name, teamName, noOfMatches, runs, noOfWickets); ar.displayPlayerStatistics();

                                                break;

                                default: System.out.println("Please Enter a Valid Input");continue;

                                }

                                System.out.println("Do you want to continue?");

                                s=sc.nextLine();

                                }

                }

}

public class AllRounder extends CricketPlayer implements IPlayerStatistics

{

                private String runs;

                private String noOfWickets;

                public AllRounder(String name, String teamName, String noOfMatches, String runs, String noOfWickets) {

                                super(name, teamName, noOfMatches);

                                this.runs = runs;

                                this.noOfWickets = noOfWickets;

                }

                public AllRounder(String runs, String noOfWickets) {

                                super();

                                this.runs = runs;

                                this.noOfWickets = noOfWickets;

                }

                public AllRounder() {

                                super();

                                // TODO Auto-generated constructor stub

                }

                @Override

                public void displayPlayerStatistics()

                {

                                // TODO Auto-generated method stub

                                System.out.println("Player name : "+super.getName());

                                System.out.println("Team name : "+super.getTeamName());

                                System.out.println("No of matches : "+super.getNoOfMatches());

                                System.out.println("Runs scored : "+this.runs);

                                System.out.println("No of wickets taken : "+this.noOfWickets);

                }

                public String getRuns() {

                                return runs;

                }

                public void setRuns(String runs) {

                                this.runs = runs;

                }

                public String getNoOfWickets() {

                                return noOfWickets;

                }

                public void setNoOfWickets(String noOfWickets) {

                                this.noOfWickets = noOfWickets;

                }

 }

public class CricketPlayer

{

                private String name;

                private String teamName;

                private String noOfMatches;

                public CricketPlayer() {

                                super();

                                // TODO Auto-generated constructor stub

                }

                public CricketPlayer(String name, String teamName, String noOfMatches) {

                                super();

                                this.name = name;

                                this.teamName = teamName;

                                this.noOfMatches = noOfMatches;

                }

                public String getName() {

                                return name;

                }

                public void setName(String name) {

                                this.name = name;

                }

                public String getTeamName() {

                                return teamName;

                }

                public void setTeamName(String teamName) {

                                this.teamName = teamName;

                }

                public String getNoOfMatches() {

                                return noOfMatches;

                }

                public void setNoOfMatches(String noOfMatches) {

                                this.noOfMatches = noOfMatches;

                }

}

public class WicketKeeper extends CricketPlayer implements IPlayerStatistics

{

                private String noOfCatches;

                private String noOfStumpings;

                private String runs;

                private String noOfDismissals;

                @Override

                public void displayPlayerStatistics() {

                                // TODO Auto-generated method stub

                                System.out.println("Player name : "+super.getName());

                                System.out.println("Team name : "+super.getTeamName());

                                System.out.println("No of matches : "+super.getNoOfMatches());

                                System.out.println("No of catches taken : "+this.noOfCatches);

                                System.out.println("No of stumpings : "+this.noOfStumpings);

                                System.out.println("No of dismissals : "+this.noOfDismissals);

                                System.out.println("Runs scored : "+this.runs);

                }

                public WicketKeeper(String noOfCatches, String noOfStumpings, String runs, String noOfDismissals) {

                                super();

                                this.noOfCatches = noOfCatches;

                                this.noOfStumpings = noOfStumpings;

                                this.runs = runs;

                                this.noOfDismissals = noOfDismissals;

                }

                public WicketKeeper(String name, String teamName, String noOfMatches, String noOfCatches, String noOfStumpings,

                                                String runs, String noOfDismissals) {

                                super(name, teamName, noOfMatches);

                                this.noOfCatches = noOfCatches;

                                this.noOfStumpings = noOfStumpings;

                                this.runs = runs;

                                this.noOfDismissals = noOfDismissals;

                }

                public WicketKeeper() {

                                super();

                                // TODO Auto-generated constructor stub

                }

}

public class Bowler extends CricketPlayer implements IPlayerStatistics

{

                private String noOfWickets;

                public Bowler(String noOfWickets) {

                                super();

                                this.noOfWickets = noOfWickets;

                }

                public Bowler() {

                                super();

                                // TODO Auto-generated constructor stub

                }

                public Bowler(String name, String teamName, String noOfMatches, String noOfWickets) {

                                super(name, teamName, noOfMatches);

                                this.noOfWickets = noOfWickets;

                }

                @Override

                public void displayPlayerStatistics() {

                                System.out.println("Player name : "+super.getName());

                                System.out.println("Team name : "+super.getTeamName());

                                System.out.println("No of matches : "+super.getNoOfMatches());

                                System.out.println("No of wickets taken : "+this.noOfWickets);

                }

}

public interface IPlayerStatistics

{

                public void displayPlayerStatistics();

}

**EXCEPTION HANDLING /SESSION 1**

**Exception Handling -1(Arithmetic exception and Number format exception)**

import java.io.\*;

public class Main {

public static void main(String[] args)

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

float n,m;

float k;

try

{

System.out.println("Enter the total runs scored");

n=Float.parseFloat(br.readLine());

System.out.println("Enter the total overs faced");

m=Float.parseFloat(br.readLine());

if(m==0)

{

throw new ArithmeticException();

}

k=n/m;

System.out.printf("Current Run Rate : %.2f",k);

}

catch(Exception e )

{

System.out.println(e.getClass().getName());

}

}

}

**Exception 2(ArrayIndexOutOfBoundsException And NegativeArraySizeException)**

import java.util.\*;

public class Main{

public static void main(String args[])

{

try

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number of overs");

int n=sc.nextInt();

if(n<0)

{

throw new NegativeArraySizeException();

}

System.out.println("Enter the number of runs for each over");

int arr[]=new int[n];

for(int i=0;i<arr.length;i++)

{

arr[i]=sc.nextInt();

}

System.out.println("Enter the over number");

int m=sc.nextInt();

if(m>n)

{

throw new ArrayIndexOutOfBoundsException();

}

System.out.println(arr[m-1]);

}

catch(Exception e)

{

System.out.println(e.getClass().getName());

}

}

}

**Exception Handling -3(NullPointerException)**

import java.util.\*;

class Main {

public static void main(String args[]) {

int num;

String a,o="Yes",n="No",n1,c1,sset,x,y,z;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of players");

num=sc.nextInt();

sc.nextLine();

Player p[] = new Player[num+1];

System.out.println("Do you know the details of the captain? Type Yes / No");

a=sc.nextLine();

//sc.nextLine();

if(a.equals(o))

{

System.out.println("Enter name of the captain");

n1=sc.nextLine();

System.out.println("Enter country of the captain");

c1=sc.nextLine();

System.out.println("Enter skillset of the captain");

sset=sc.nextLine();

p[0] = new Player(n1,c1,sset);

}

if(a.equals(n))

{

//throw new NullPointerException();

p[0] = new Player(null,null,null);

}

for(int i=0;i<num;i++)

{

System.out.println("Enter name of player "+(i+1));

x=sc.nextLine();

System.out.println("Enter country of player "+(i+1));

y=sc.nextLine();

System.out.println("Enter skillset of player "+(i+1));

z=sc.nextLine();

p[i+1] = new Player(x,y,z);

}

PlayerBO pb = new PlayerBO();

pb.displayPlayerDetails(p);

//fill your code;

}

}

class Player

{

private String name,country,skill;

Player()

{

}

Player(String name, String country, String skill)

{

this.name = name;

this.country = country;

this.skill = skill;

}

public void setName(String nn)

{

name=nn;

}

public void setCountry(String cc)

{

country=cc;

}

public void setSkill(String ss)

{

skill=ss;

}

public String getName()

{

return name;

}

public String getCountry()

{

return country;

}

public String getSkill()

{

return skill;

}

}

class PlayerBO

{

public void displayPlayerDetails(Player[] p)

{

String s = null;

try

{

if(p[0].getName().equals(s) || p[0].getCountry().equals(s) || p[0].getSkill().equals(s) ||

p[0].getName().equals(s) && p[0].getCountry().equals(s) && p[0].getSkill().equals(s))

throw new NullPointerException();

else

System.out.println(p[0].getName() + ", " + p[0].getCountry() + ", " + p[0].getSkill());

}

catch(NullPointerException e)

{

System.out.println("Exception Occured : java.lang.NullPointerException");

System.out.println("Captain details not available");

}

System.out.println("Player Details");

for(int i=1;i<p.length;i++)

{

System.out.println(p[i].getName() + ", " + p[i].getCountry() + ", " + p[i].getSkill());

}

}

}

**Exception Handling -4 (InstantiationException)**

import java.util.\*;

public class Main {

public static void main(String args[])

{

try

{

Player obj=Player.class.newInstance();

}

catch(InstantiationException e)

{

System.out.println("Trying to invoke a no-argument constructor (that is not available) using newInstance method");

System.out.println("Exception Occured : "+e.getClass().getName());

Scanner sc=new Scanner(System.in);

System.out.println("Enter name of the player");

String s=sc.nextLine();

System.out.println("Enter country of the player");

String s1=sc.nextLine();

System.out.println("Enter skillset of the player");

String s2=sc.nextLine();

Player obj1=new Player(s,s1,s2);

System.out.println(obj1.getName()+", "+obj1.getCountry()+", "+obj1.getSkill());

}

catch (IllegalAccessException e)

{

e.printStackTrace();

}

}

}

public class Player

{

private String name;

private String country;

private String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public Player(String name, String country, String skill) {

super();

this.name = name;

this.country = country;

this.skill = skill;

}

}

**EXCEPTION HANDLING /SESSION 2**

**Custom Exceptions [Age]**

import java.io.\*;

import java.util.\*;

class Main {

public static void main(String[] args)throws IOException {

System.out.println("Enter the player name");

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

System.out.println("Enter the player age");

int n = sc.nextInt();

try

{

if(n<18)

throw new CustomException("InvalidAgeRangeException");

else

System.out.println("Player name : " + s);

System.out.println("Player age : " +n);

}

catch(Exception m)

{

System.out.println( m);

}

}

}

class CustomException extends Exception

{

public CustomException(String s)

{

super(s);

}

}

**TeamNameNotFound Exception**

**import java.util.\*;**

**public class Main**

**{**

**public static void main(String ar[])**

**{**

**try**

**{**

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

**String arr[]={"Chennai Super Kings","Deccan Chargers","Delhi Daredevils","Kings XI Punjab","Kolkata Knight Riders","Mumbai Indians","Rajasthan Royals","Royal Challengers Bangalore"};**

**System.out.println("Enter the expected winner team of IPL Season 4");**

**String s1=sc.nextLine();**

**int f=0,f1=0;**

**for(int i=0;i<arr.length;i++)**

**{**

**if(s1.equals(arr[i]))**

**f=1;**

**}**

**if(f==0)**

**{**

**throw new TeamNameNotFoundException();**

**}**

**else**

**{**

**System.out.println("Enter the expected runner Team of IPL Season 4");**

**String s2=sc.nextLine();**

**for(int i=0;i<arr.length;i++)**

**{**

**if(s2.equals(arr[i]))**

**f1= 1;**

**}**

**if(f1==0)**

**{**

**throw new TeamNameNotFoundException();**

**}**

**if (f==1 && f1==1)**

**{**

**System.out.println("Expected IPL Season 4 winner: "+s1);**

**System.out.println("Expected IPL Season 4 runner: "+s2);**

**}**

**}**

**}**

**catch(Exception e)**

**{**

**System.out.println(e.getClass().getName()+": Entered team is not a part of IPL Season 4");**

**}**

**}**

**}**

public class TeamNameNotFoundException extends Exception

{

public TeamNameNotFoundException()

{

super();

}

}

**Custom Exception (OverRange Exception)**

import java.io.\*;

import java.text.DecimalFormat;

class Main

{

public static void main(String[] arr)throws IOException

{

DecimalFormat d= new DecimalFormat("0.00");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the total runs scored");

float runs=Float.parseFloat(br.readLine());

System.out.println("Enter the total overs faced");

float overs=Float.parseFloat(br.readLine());

try

{

if(overs>20)

{

throw new OverRangeException();

}

float runrate=runs/overs;

System.out.println("Current Run Rate :"+(d.format(runrate)));

}

catch(Exception e)

{

System.out.println(e.getClass().getName()+": Over is not in the specified range");

}

}

}

class OverRangeException extends Exception

{

OverRangeException()

{

super();

}

}

**Duplicate Id Exception**

import java.io.\*;

import java.util.\*;

public class Main

{

public static void main(String[] args)throws IOException

{

int count=0;

HashMap<Integer,String> hm=new HashMap();

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the team name");

String teamname=br.readLine();

System.out.println("Enter the number of players suggested");

int players=Integer.parseInt(br.readLine());

for(int i=1;i<=players;i++)

{

System.out.println("Enter player "+i+" details");

int id=Integer.parseInt(br.readLine());

String name=br.readLine();

if(hm.containsKey(id))

{

try

{

count++;

throw new DuplicateIdException();

}

catch(DuplicateIdException e)

{

}

}

else

{

hm.put(id,name);

}

}

if(count==0)

{

Set s=hm.keySet();

Iterator<Integer> itr=s.iterator();

while(itr.hasNext())

{

Object o=itr.next();

Integer in=(Integer)o;

System.out.println(o+" "+hm.get(o));

}

}

}

}

class DuplicateIdException extends Exception{

public DuplicateIdException()

{

System.out.println("DuplicateIdException: Player Id must be unique");

}

}

**COLLECTIONS /GENERICS,LISTS/SESSION 1**

**List 1**

import java.util.\*;

public class Main

{

public static void main(String[] args)

    {

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

float avg;

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

{

int m=sc.nextInt();

arr.add(m);

}

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

{

Object o=arr.get(i);

Integer in=(Integer)o;

sum=sum+in;

}

avg=(float)sum/n;

System.out.println(sum);

System.out.println(avg);

}

}

**List 2**

import java.util.\*;

public class Main {

public static void main(String[] args) {

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

{

int m=sc.nextInt();

arr.add(m);

}

Collections.sort(arr);

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

{

System.out.println(arr.get(i));

}

}

}

**List 3**

import java.util.\*;

public class Main {

public static void main(String[] args) {

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

sc.nextLine();

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

{

String s=sc.nextLine();

arr.add(s);

}

String s1=sc.nextLine();

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

{

if(s1.equals(arr.get(i)))

{

count++;

}

}

System.out.println(count);

}

}

**List 4**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the team name");

String s=sc.nextLine();

System.out.println("Enter the number of matches played in home ground");

int n=sc.nextInt();

System.out.println("Enter the runs scored");

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

{

int m=sc.nextInt();

arr.add(m);

}

System.out.println("Enter the number of matches played in other ground");

int n1=sc.nextInt();

System.out.println("Enter the runs scored");

for(int i=0;i<n1;i++)

{

int m1=sc.nextInt();

arr.add(m1);

}

System.out.println("Runs scored by "+s);

for(int i=0;i<(n+n1);i++)

{

System.out.println(arr.get(i));

}

System.out.println("Run scored by "+s+" more than 300");

for(int i=0;i<(n+n1);i++)

{

    Object o=arr.get(i);

    Integer in=(Integer)o;

    if(in>300)

    {

System.out.println(in);

    }

}

}

}

**List 5**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

ArrayList arr1=new ArrayList();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the top 5 scorers of IPL Season 8");

for(int i=0;i<5;i++)

{

String s=sc.nextLine();

arr.add(s);

}

System.out.println("Enter the top 5 scorers of IPL Season 9");

for(int j=0;j<5;j++)

{

String s1=sc.nextLine();

arr1.add(s1);

}

System.out.println("Consistent run scorers");

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

if(arr.get(i).equals(arr1.get(j)))

{

System.out.println(arr.get(i));

}

}

}

}

}

**List 6**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the teams in ranking table");

for(int i=0;i<5;i++)

{

String s=sc.nextLine();

arr.add(s);

}

System.out.println("Enter the rank to be searched");

int n=sc.nextInt();

for(int i=0;i<5;i++)

{

if(n==i+1)

{

System.out.println(arr.get(i));

}

}

}

}

**List 7**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

for(int i=0;i<5;i++)

{

String s=sc.nextLine();

arr.add(s);

}

System.out.println("Enter swap positons");

int n=sc.nextInt();

int n1=sc.nextInt();

Collections.swap(arr,n,n1);

for(int i=0;i<5;i++)

{

System.out.println(arr.get(i));

}

}

}

**List 8**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the player details");

System.out.println("Enter player name");

String s=sc.nextLine();

System.out.println("Enter age");

int n=sc.nextInt();

sc.nextLine();

System.out.println("Enter Country");

String s1=sc.nextLine();

arr.add(s);

arr.add(n);

arr.add(s1);

System.out.println("Player Details");

for(int i=0;i<arr.size();i++)

{

System.out.println(arr.get(i));

}

System.out.println("Enter Skill");

String s2=sc.nextLine();

arr.add(s2);

System.out.println("Enter the position to add the skill");

int n1=sc.nextInt();

Collections.swap(arr,n1,3);

arr.remove(3);

System.out.println("Player Details");

for(int i=0;i<arr.size();i++)

{

System.out.println(arr.get(i));

}

System.out.println("Enter the position of the detail to be removed");

int n2=sc.nextInt();

arr.remove(n2);

System.out.println("Player Details");

for(int i=0;i<arr.size();i++)

{

System.out.println(arr.get(i));

}

}

}

**COLLECTIONS /GENERICS,LISTS/SESSION 2**

**Sum Scores in Even Position**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

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

{

int m=sc.nextInt();

arr.add(m);

}

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

{

if(i%2!=0)

{

Object o=arr.get(i);

Integer in=(Integer)o;

sum=sum+in;

}

}

System.out.println(sum);

}

}

**Odd-Even Index**

import java.util.\*;

public class Main

{

public static void main(String[] args)

    {

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

int sum1=0;

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

{

int m=sc.nextInt();

arr.add(m);

}

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

{

if(i%2!=0)

{

Object o=arr.get(i);

Integer in=(Integer)o;

if(in%2!=0)

{

sum=sum+in;

}

}

else

{

Object o=arr.get(i);

Integer in=(Integer)o;

if(in%2==0)

{

sum1=sum1+in;

}

}

}

System.out.println(sum+sum1);

}

}

**Duck - batsman's dismissal**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Main m=new Main();

ArrayList arr=new ArrayList();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

        int count=0;

sc.nextLine();

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

{

String s=sc.nextLine();

arr.add(s);

}

for(int i=0;i<arr.size();i++)

{

Object obj=arr.get(i);

String str=(String)obj;

String a[]=str.split("-");

            if(a[1].equals("0")&& a[2].equals("0"))

            {

                count++;

                System.out.println(a[0]);

            }

}

        if(count==0)

        {

           System.out.println("No player has scored a duck");

        }

}

}

**COLLECTIONS SETS/SESSION 1**

**HashSet 1**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

HashSet hs=new HashSet();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

sc.nextLine();

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

{

String s=sc.nextLine();

hs.add(s);

}

System.out.println(hs.size());

}

}

**Player of the Match**

import java.util.\*;

public class Main {

public static void main(String[] args) {

TreeSet hs=new TreeSet();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

sc.nextLine();

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

{

String s=sc.nextLine();

hs.add(s);

}

Iterator itr=hs.iterator();

while(itr.hasNext())

{

System.out.println(itr.next());

}

}

}

**Set - Revenue Manager**

import java.util.\*;

public class Main {

public static void main(String[] args) {

TreeSet hs=new TreeSet();

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

sc.nextLine();

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

{

String s=sc.nextLine();

hs.add(s);

}

Iterator itr=hs.iterator();

while(itr.hasNext())

{

System.out.println(itr.next());

}

}

}

**Set - Player List Index Builder**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

HashSet<Player> playerSet = new HashSet<Player>();

String c = "";

do

{

System.out.println("Enter Player Name:");

String name = scan.nextLine();

System.out.println("Enter Skill:");

String skill = scan.nextLine();

Player player = new Player(name , skill);

if(playerSet.contains(player))

{

System.out.println("Player "+name+" already exist");

}

System.out.println("Do you want to continue(yes/no):");

c = scan.nextLine();

playerSet.add(player);

}while(c.equals("yes"));

IndexBuilder indBuild = new IndexBuilder();

TreeSet<Index> indexSet = indBuild.buildIndex(playerSet);

indBuild.displayIndex(indexSet);

scan.close();

}

}

import java.util.\*;

public class IndexBuilder {

public TreeSet<Index> buildIndex(HashSet<Player> playerSet)

{

TreeSet<Index> indexSet = new TreeSet<Index>();

for(Player player : playerSet)

{

char firstChar = player.getName().charAt(0);

Index foundIndex = findIndex(indexSet , firstChar);

if(foundIndex == null)

{

Index ind = new Index(firstChar , 1);

indexSet.add(ind);

}

else

{

int count = foundIndex.getCount();

count++;

foundIndex.setCount(count);

}

}

return indexSet;

}

private Index findIndex(TreeSet<Index> indexSet , char ch )

{

for(Index index : indexSet)

{

if(index.getCh() == ch)

{

return index;

}

}

return null;

}

public void displayIndex(TreeSet<Index> indexSet)

{

System.out.println(String.format("%-14s%-15s" , "Player" , "Index"));

for(Index i : indexSet)

{

System.out.println(String.format("%-15s%-15d" , i.getCh() , i.getCount()));

}

}

}

public class Index implements Comparable<Index>{

Character ch;

Integer count;

public char getCh() {

return ch;

}

public void setCh(char ch) {

this.ch = ch;

}

public int getCount() {

return count;

}

public void setCount(int count) {

this.count = count;

}

Index()

{

}

public Index(char ch, int count) {

super();

this.ch = ch;

this.count = count;

}

@Override

public int compareTo(Index arg0) {

return this.getCh() - arg0.getCh();

}

}

public class Player {

String name;

String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

Player()

{

}

public Player(String name, String skill) {

super();

this.name = name;

this.skill = skill;

}

@Override

public boolean equals(Object arg0) {

Player arg = (Player) arg0;

return this.getName().equals(arg.getName());

}

@Override

public int hashCode() {

return this.getName().hashCode();

}

}

**Difference**

**import java.util.\*;**

**public class Main {**

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

**{**

**HashSet hs=new HashSet();**

**HashSet hs1=new HashSet();**

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

**System.out.println("Enter the number of top run scorers in season 4");**

**int n=sc.nextInt();**

**sc.nextLine();**

**System.out.println("Enter the name of players");**

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

**{**

**String s=sc.nextLine();**

**hs.add(s);**

**}**

**System.out.println("Enter the number of top run scorers in season 5");**

**int n1=sc.nextInt();**

**sc.nextLine();**

**System.out.println("Enter the name of players");**

**for(int i=0;i<n1;i++)**

**{**

**String s=sc.nextLine();**

**hs1.add(s);**

**}**

**System.out.println("Player Set 1");**

**Iterator itr=hs.iterator();**

**while(itr.hasNext())**

**{**

**System.out.println(itr.next());**

**}**

**System.out.println("Player Set 2");**

**Iterator itr1=hs1.iterator();**

**while(itr1.hasNext())**

**{**

**System.out.println(itr1.next());**

**}**

**hs.removeAll(hs1);**

**System.out.println("Difference");**

**Iterator itr2=hs.iterator();**

**while(itr2.hasNext())**

**{**

**System.out.println(itr2.next());**

**}**

**}**

**}**

**Union**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

HashSet hs=new HashSet();

HashSet hs1=new HashSet();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number of top run scorers in season 4");

int n=sc.nextInt();

sc.nextLine();

System.out.println("Enter the name of players");

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

{

String s=sc.nextLine();

hs.add(s);

}

System.out.println("Enter the number of top run scorers in season 5");

int n1=sc.nextInt();

sc.nextLine();

System.out.println("Enter the name of players");

for(int i=0;i<n1;i++)

{

String s=sc.nextLine();

hs1.add(s);

}

System.out.println("Player Set 1");

Iterator itr=hs.iterator();

while(itr.hasNext())

{

System.out.println(itr.next());

}

System.out.println("Player Set 2");

Iterator itr1=hs1.iterator();

while(itr1.hasNext())

{

System.out.println(itr1.next());

}

hs.addAll(hs1);

System.out.println("Union");

Iterator itr2=hs.iterator();

while(itr2.hasNext())

{

System.out.println(itr2.next());

}

}

}

**Intersection**

import java.util.\*;

public class Main {

public static void main(String[] args)

{

HashSet hs=new HashSet();

HashSet hs1=new HashSet();

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number of top run scorers in season 4");

int n=sc.nextInt();

sc.nextLine();

System.out.println("Enter the name of players");

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

{

String s=sc.nextLine();

hs.add(s);

}

System.out.println("Enter the number of top run scorers in season 5");

int n1=sc.nextInt();

sc.nextLine();

System.out.println("Enter the name of players");

for(int i=0;i<n1;i++)

{

String s=sc.nextLine();

hs1.add(s);

}

System.out.println("Player Set 1");

Iterator itr=hs.iterator();

while(itr.hasNext())

{

System.out.println(itr.next());

}

System.out.println("Player Set 2");

Iterator itr1=hs1.iterator();

while(itr1.hasNext())

{

System.out.println(itr1.next());

}

hs.retainAll(hs1);

System.out.println("Intersection");

Iterator itr2=hs.iterator();

while(itr2.hasNext())

{

System.out.println(itr2.next());

}

}

}

**COLLECTIONS SETS/SESSION 2**

**Comparable - List Match By Date**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Iterator;

import java.util.TreeSet;

public class Main

{

public static void main(String args[]) throws NumberFormatException, IOException, ParseException

{

TreeSet<Match> ts=new TreeSet<Match>();

SimpleDateFormat s=new SimpleDateFormat("MM-dd-yyyy");

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of matches");

int n=Integer.parseInt(br.readLine());

for(int i=1;i<=n;i++)

{

System.out.println("Enter match date in (MM-dd-yyyy)");

String d=br.readLine();

Date dd= s.parse(d);

System.out.println("Enter Team 1");

String t1=br.readLine();

System.out.println("Enter Team 2");

String t2=br.readLine();

Match mat=new Match(dd,t1,t2);

ts.add(mat);

}

System.out.println("Match Details");

Iterator<Match> itr=ts.iterator();

while(itr.hasNext())

{

Match m=itr.next();

String sss=s.format(m.getMatchDate());

System.out.println("Team 1 "+m.getTeamOne());

System.out.println("Team 2 "+m.getTeamTwo());

System.out.println("Match held on "+sss);

}

}

}

import java.util.Date;

public class Match implements Comparable<Match>

{

Date matchDate=new Date();

String teamOne;

String teamTwo;

public Date getMatchDate() {

return matchDate;

}

public void setMatchDate(Date matchDate) {

this.matchDate = matchDate;

}

public String getTeamOne() {

return teamOne;

}

public void setTeamOne(String teamOne) {

this.teamOne = teamOne;

}

public String getTeamTwo() {

return teamTwo;

}

public void setTeamTwo(String teamTwo) {

this.teamTwo = teamTwo;

}

public Match(Date matchDate, String teamOne, String teamTwo) {

super();

this.matchDate = matchDate;

this.teamOne = teamOne;

this.teamTwo = teamTwo;

}

@Override

public int compareTo(Match a)

{

if(a.matchDate==matchDate)

return 0;

else if(a.matchDate.after(matchDate))

return 1;

else

return -1;

}

}

**Comparable - Player Information Based on Player Cap Number**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.\*;

public class Main

{

public static void main(String[] args) throws NumberFormatException, IOException

{

TreeSet<Player> ts=new TreeSet<Player>();

BufferedReader sc=new BufferedReader (new InputStreamReader(System.in));

System.out.println("Enter number of players:");

int n=Integer.parseInt(sc.readLine());

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

{

System.out.println("Enter player "+(i+1)+" detail");

System.out.println("Enter Name");

String name=sc.readLine();

System.out.println("Enter Skill");

String skill=sc.readLine();

System.out.println("Enter Cap Number");

long num=Long.parseLong(sc.readLine());

Player p=new Player(name,skill,num);

ts.add(p);

}

java.util.Iterator<Player> itr=ts.iterator();

System.out.println("Player list after sorting by cap number in descending order");

while(itr.hasNext())

{

Player obj=itr.next();

System.out.println(obj.getPlayerName()+"-"+obj.getCapNumber());

}

}

}

public class Player implements Comparable<Player>

{

private String PlayerName;

public Player(String playerName, String skill, Long capNumber) {

super();

PlayerName = playerName;

this.skill = skill;

this.capNumber = capNumber;

}

private String skill;

private Long capNumber;

public String getPlayerName() {

return PlayerName;

}

public void setPlayerName(String playerName) {

PlayerName = playerName;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public Long getCapNumber() {

return capNumber;

}

public void setCapNumber(Long capNumber) {

this.capNumber = capNumber;

}

@Override

public int compareTo(Player p) {

if(this.capNumber==p.capNumber)

return 0;

else if(this.capNumber>p.capNumber)

return -1;

else

return 1;

}

}

**COLLECTIONS MAP/SESSION 1**

**Comparator - Team name and Number of matches**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Set;

import java.util.TreeMap;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException

{

TreeMap<Integer,Team> tm=new TreeMap();

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

System.out.println("Enter number of teams:");

int n= Integer.parseInt(br.readLine());

for(int i=1;i<=n;i++)

{

System.out.println("Enter team "+i+" detail");

System.out.println("Enter Name");

String name=br.readLine();

System.out.println("Enter number of matches");

Long match=Long.parseLong(br.readLine());

Team tt=new Team(name,match);

tm.put(i, tt);

}

Set s=tm.keySet();

java.util.Iterator itr= s.iterator();

ArrayList<Team> arr=new ArrayList<Team>();

while(itr.hasNext())

{

Integer i=(Integer) itr.next();

Team t=tm.get(i);

arr.add(t);

}

Collections.sort(arr, new TeamComparator());

System.out.println("Team list after sort by number of matches");

java.util.Iterator<Team> itr1=arr.iterator();

while(itr1.hasNext())

{

Team obj=(Team) itr1.next();

System.out.println(obj.getName()+"-"+obj.getNumberOfMatches());

}

}

}

import java.util.Comparator;

public class TeamComparator implements Comparator<Team>

{

@Override

public int compare(Team t1,Team t2)

{

returnt1.getNumberOfMatches().compareTo(t2.getNumberOfMatches());

}

}

public class Team

{

String name;

Long numberOfMatches;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Long getNumberOfMatches() {

return numberOfMatches;

}

public void setNumberOfMatches(Long numberOfMatches) {

this.numberOfMatches = numberOfMatches;

}

public Team(String name, Long numberOfMatches) {

super();

this.name = name;

this.numberOfMatches = numberOfMatches;

}

}

**Comparator - Player List Based on Name and Skill**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.\*;

public class Main

{

public static void main(String[] args) throws NumberFormatException, IOException{

TreeMap<Integer,Player> tm=new TreeMap();

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

System.out.println("Please provide the number of players to be registered");

int n=Integer.parseInt(br.readLine());

for(int i=1;i<=n;i++)

{

System.out.println("Please enter player name");

String s=br.readLine();

System.out.println("Please select the skill of the player");

System.out.println("1.All Rounder");

System.out.println("2.Batsman");

System.out.println("3.Bowler");

int n1=Integer.parseInt(br.readLine());

String sk=" ";

switch(n1)

{

case 1: sk="All Rounder";

break;

case 2: sk="Batsman";

break;

case 3: sk="Bowler";

break;

}

Player p=new Player(s,sk);

tm.put(i,p);

}

Set s=tm.keySet();

Iterator itr= s.iterator();

ArrayList<Player> arr=new ArrayList<Player>();

while(itr.hasNext())

{

Integer i=(Integer) itr.next();

Player t=tm.get(i);

arr.add(t);

}

Collections.sort(arr, new PlayerComparator());

System.out.println("Player Details");

Iterator<Player> itr1=arr.iterator();

while(itr1.hasNext())

{

Player obj=(Player) itr1.next();

System.out.println("Player : "+obj.getName()+" Skill : "+obj.getSkill()); }

}

}

import java.util.Comparator;

public class PlayerComparator implements Comparator<Player>

{

@Override

public int compare(Player p1, Player p2)

{

int skill=(p1.getSkill().compareTo(p2.getSkill()));

if(skill==0)

{

return p1.getName().compareTo(p2.getName());

}

return skill;

}

}

public class Player

{

String name;

String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public Player(String name, String skill) {

super();

this.name = name;

this.skill = skill;

}

}

**HashMap – Player**

import java.util.\*;

import java.io.\*;

public class Main

{

public static void main( String args[] ) throws IOException

{

BufferedReader br = new BufferedReader( new InputStreamReader ( System.in ));

HashMap<Integer,Player> hm = new HashMap();

ArrayList al = new ArrayList();

String opt = "no";

int i=1;

while(true)

{

System.out.println("Enter the player name");

String name = br.readLine();

al.add(name);

char ch ='|';

System.out.println("Enter wickets - seperated by \"|\" symbol.");

String wickets = br.readLine();

StringTokenizer s = new StringTokenizer(wickets, "|");

int wick = s.countTokens();

Player p = new Player(name,wick);

hm.put(i,p);

i++;

System.out.println("Do you want to add another player (yes/no)");

String add = br.readLine();

if(add.equals("yes"))

continue;

else

break;

}

do

{

System.out.println("Enter the player name to search");

String pname = br.readLine();

if(al.contains(pname))

{

Iterator itr = hm.keySet().iterator();

while(itr.hasNext())

{

Integer o = (Integer)itr.next();

Player pl = hm.get(o);

if(pname.equals(pl.getBowlerName()))

{

System.out.println("Player name : "+pl.getBowlerName());

System.out.println("Wicket Count : "+pl.getWicketCount());

}

}

}

else

{

System.out.println("No player found with the name "+pname);

}

System.out.println("Do you want to search another player (yes/no)");

opt = br.readLine();

}

while(opt.equals("yes"));

}

}

public class Player

{

private

String bowlerName;

Integer wicketCount;

public Player( String bowlerName, Integer wicketCount)

{

this.bowlerName = bowlerName;

this.wicketCount = wicketCount;

}

public String getBowlerName() {

return bowlerName;

}

public void setBowlerName(String bowlerName) {

this.bowlerName = bowlerName;

}

public Integer getWicketCount() {

return wicketCount;

}

public void setWicketCount(Integer wicketCount) {

this.wicketCount = wicketCount;

}

}

**TreeMap-Player Details**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.\*;

public class Main {

public static void main(String[] args) throws NumberFormatException, IOException

{

TreeMap<String,Player> tm=new TreeMap();

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

System.out.println("Enter the number of players");

int n=Integer.parseInt(br.readLine());

for(int i=1;i<=n;i++)

{

System.out.println("Enter the details of the player "+i);

String capNumber=(br.readLine());

String name=br.readLine();

String team=br.readLine();

String skill=br.readLine();

Player p=new Player(name,team,skill);

tm.put(capNumber,p);

}

System.out.println("Player Details");

Iterator itr = tm.keySet().iterator();

while(itr.hasNext())

{

String s = ( String )itr.next();

Player pl = tm.get(s);

System.out.println(s +"--"+pl.getName()+"--"+pl.getTeam()+"--"+pl.getSkill());

}

}

}

public class Player

{

private String name;

private String team;

private String skill;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTeam() {

return team;

}

public void setTeam(String team)

{

this.team=team;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public Player(String name,String team, String skill) {

super();

this.name = name;

this.team=team;

this.skill = skill;

}

}

**COLLECTIONS MAP/SESSION 2**

**HashMap – Wicket Details**

import java.util.\*;

import java.io.\*;

public class Main

{

public static void main( String args[] ) throws IOException

{

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

HashMap<String,Wicket> hm = new HashMap();

ArrayList al = new ArrayList();

String opt = "no";

while(true)

{

System.out.println("Enter the player name");

String name = br.readLine();

al.add(name);

System.out.println("Enter wickets - seperated by \"|\" symbol");

String wickets = br.readLine();

System.out.println("Do you want to add another player (yes/no)");

String opt1 = br.readLine();

Bowler b = new Bowler(name);

Wicket w = new Wicket(wickets,name,b);

hm.put(name, w);

if( opt1.equals("yes"))

continue;

else

break;

}

do

{

System.out.println("Enter the player name to search");

String pname = br.readLine();

if(al.contains(pname))

{

Iterator itr = hm.keySet().iterator();

while(itr.hasNext())

{

String o = (String ) itr.next();

Wicket w1 = hm.get(o);

if(pname.equals(w1.getName()))

{

System.out.println("Player Name : "+o);

System.out.println("Wickets :");

StringTokenizer st = new StringTokenizer(w1.playerName, "|");

while( st.hasMoreTokens())

{

String arr = st.nextToken();

System.out.println(arr);

}

}

}

}

else

System.out.println("No player found with the name "+pname);

System.out.println("Do you want to search another player (yes/no)");

opt = br.readLine();

} while(opt.equals("yes"));

}

}public class Bowler

{

private

String name;

public Bowler(String name) {

super();

this.name = name;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

public class Wicket extends Bowler

{

String playerName;

Bowler Bowler = new Bowler(playerName);

public Wicket(String pname,String name,Bowler Bowler)

{

super(name);

this.Bowler = Bowler;

this.playerName = pname;

}

public String getPlayerName()

{

return playerName;

}

public void setPlayerName(String playerName)

{

this.playerName = playerName;

}

public Bowler getBowler()

{

return Bowler;

}

public void setBowler(Bowler bowler)

{

Bowler = bowler;

}

}

**HashMap - Scores to Bin**

import java.util.\*;

class Main

{

public static void main(String args[])

{

Histogram h=new Histogram();

h.addScore();

h.displayHistogram();

}

}

import java.util.\*;

class Histogram

{

int i,j,n;

private HashMap<Integer,Integer> map=new HashMap<>();

Scanner s=new Scanner(System.in);

public Histogram()

{

n=Integer.parseInt(s.nextLine());

}

public void addScore()

{

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

{

map.put(i+1,s.nextInt());

}

}

public void displayHistogram()

{

int[] a={0,0,0,0};

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

{

if(map.get(i+1)<=10)

a[0]++;

else if(map.get(i+1)<=20)

a[1]++;

else if(map.get(i+1)<=30)

a[2]++;

else

a[3]++;

}

System.out.println("Histogram");

for(i=0;i<4;i++)

{

System.out.printf(((i+1)\*10)+" : ");

for(j=0;j<a[i];j++)

System.out.printf("\*");

System.out.println("");

}

}

}

**TreeMap-Player Details**

import java.io.\*;

import java.util.\*;

public class Main

{

public static void main ( String args[]) throws IOException

{

BufferedReader br = new BufferedReader ( new InputStreamReader ( System.in ));

TreeMap<Integer,Player> tm = new TreeMap();

ArrayList al = new ArrayList();

System.out.println("Enter the number of players");

int n = Integer.parseInt( br.readLine());

for( int i=1;i<=n;i++)

{

System.out.println("Enter the details of the player "+i);

int capNumber = Integer.parseInt(br.readLine());

al.add(capNumber);

String name = br.readLine();

String team = br.readLine();

String skill = br.readLine();

Player p = new Player (name,team,skill);

tm.put(capNumber,p);

}

System.out.println("Player Details");

Iterator itr = tm.keySet().iterator();

while(itr.hasNext())

{

Integer o = ( Integer ) itr.next();

Player p1 = tm.get(o);

System.out.println(o+"--"+p1.getName()+"--"+p1.getTeam()+"--"+p1.getSkill());

}

System.out.println("Enter the cap number of the player to be searched");

int se = Integer.parseInt( br.readLine() );

if(al.contains(se))

{

Iterator itr1 = tm.keySet().iterator();

while(itr1.hasNext())

{

Integer o1 = (Integer) itr1.next();

int i = o1.intValue();

Player p2 = tm.get(o1);

if(se == i)

{

System.out.println("Player Details");

System.out.println(p2.getName()+"--"+p2.getTeam()+"--"+p2.getSkill());

}

}

}

else

{

System.out.println("Player not found");

}

}

}

public class Player

{

private

String name,team,skill;

public Player(String name, String team, String skill) {

super();

this.name = name;

this.team = team;

this.skill = skill;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTeam() {

return team;

}

public void setTeam(String team) {

this.team = team;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

}

**TreeMap - Letter Frequency**

import java.util.\*;

import java.io.\*;

public class Main

{

public static void main(String[] args) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader ( System.in ));

System.out.println("Enter the input string");

String sentence = br.readLine();

LetterSequence obj = new LetterSequence(sentence);

TreeMap tm1 = obj.computeFrequency();

obj.displayLetterFrequency(tm1);

}

}

import java.util.ArrayList;

import java.util.Collections;

import java.util.Iterator;

import java.util.TreeMap;

public class LetterSequence

{

private String sentence;

public LetterSequence(String sentence) {

super();

this.sentence = sentence;//Kohli is the man of the match

}

public TreeMap<Character,Integer> computeFrequency()

{

TreeMap<Character,Integer> tm = new TreeMap<Character,Integer>();

ArrayList<Character> al = new ArrayList<Character>();

for(int i=0;i<sentence.length();i++)

{

al.add(sentence.charAt(i));

}

for(int i=0;i<al.size();i++)

{

tm.put(al.get(i),Collections.frequency(al,al.get(i)));

}

return tm;

}

public void displayLetterFrequency(TreeMap<Character,Integer> frequencyMap)

{

Iterator<Character> itr = frequencyMap.keySet().iterator();

while(itr.hasNext())

{

Character o = (Character) itr.next();

if(!o.toString().equals(" "))

{

Integer val = (Integer) frequencyMap.get(o);

System.out.print(o+" : ");

for(int i=0;i<val.intValue();i++)

{

System.out.print("\*");

}

System.out.println();

}

}

}

}