```java
package com.ishank;

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

public class Main {

    private static int[] magicSquareArray = new int[]{8,3,4,1,5,9,6,7,2};      //reference magic square array
    private static String[] runningArray = new String[]{"_","_","_","_","_","_","_","_","_"}; //display array
    private static List<Integer> humanList = new ArrayList<>();     //list of human moves
    private static List<Integer> computerList = new ArrayList<>();  //list of computer moves


    public static void main(String[] args) {

        System.out.println("\n\nReference magic Square\n");
        printMagicSquare(magicSquareArray);     //function to print the magic square

        System.out.print("\nPress h to start human first\nPress c to start computer first\n");

        Scanner input = new Scanner(System.in);
        String choice = input.nextLine();   //input choice between human and computer from the user


        while (!(choice.equals("c") || choice.equals("h"))){    //input the choice again in case of wrong input
            System.out.println("please enter the correct input");
            choice = input.nextLine();
        }

        if (choice.equals("c")) Go("computer"); //start with computer if input is c
        else Go("human");   //start with human if input is h
    }

    private static void Go(String player) {     //Go method for starting the game
        for (int i=0;i<9;i++){
            if (player.equals("computer")){     //if computer starts the game
                if (i==0) {     //in first move, machine always plays in the middle
                    System.out.println("Move 1: Machine\n");
                    runningArray[4] = "x";
                    computerList.add(magicSquareArray[4]);
                    printRunningSquare(runningArray);           //printing the board
                    printLists();
                }

                else if (i%2 != 0){     //odd counts corresponds to human input
                    inputHuman(i);      //calling human input method and passing the move number
                    checkDraw();        //checks if the match is draw
```

```
            }

            else if(i==2) {     //in third move, machine tries to occupy the corners first
                System.out.println("\nMove 3: Machine\n");
                GotoCorners();      //calling the corners occupying method
                printRunningSquare(runningArray);
                printLists();
            }

            else if (i>3 && (i%2==0)){      //after fifth move machine tries to check the winning possibilities for both computer and human
                System.out.println("\nMove" + (i+1) + ": Machine\n");
                if (possWin("c")) {     //checking the winning possibility of the computer
                    printRunningSquare(runningArray);
                    printLists();
                    System.out.println("\nMachine wins !\n");
                    break;      //move out of the loop if machine wins
                }
                else if (possWin("h")){         //checking the winning possibility of the computer if machine can't win
                    printRunningSquare(runningArray);
                    printLists();
                    checkDraw();
                }
                else {
                    GotoCorners();      //machine plays randomly if both possibilities return false
                    printRunningSquare(runningArray);
                    printLists();
                    checkDraw();
                }
            }
        }
        else if (player.equals("human")){       //if human starts the game
            if (i%2 == 0){      //even counts corresponds to human input
                inputHuman(i);
                checkDraw();
            }
            else if (i==1) {
                System.out.println("Move 2: Machine\n");
                if(runningArray[0].equals("o") || runningArray[2].equals("o")) {    //if human plays in the corners, machine will not play in the middle
                    runningArray[1]="x";
                    computerList.add(magicSquareArray[1]);
                }
                else if(runningArray[6].equals("o") || runningArray[8].equals("o")) {
                    runningArray[7]="x";
                    computerList.add(magicSquareArray[7]);
                }
                else if (runningArray[4].equals("_")) {     //else machine will play in the middle
                    runningArray[4] = "x";
                    computerList.add(magicSquareArray[4]);
                }
```

```
                else GotoCorners();     //if human has played in the middle, machine will go to
corners
                printRunningSquare(runningArray);
                printLists();
            }

            else if(i==3) {     //from fourth move, machine will start checking winning possibilities
                System.out.println("\nMove 4: Machine\n");
                if (possWin("h")){
                    printRunningSquare(runningArray);
                    printLists();
                }

                else {
                    GotoCorners();
                    printRunningSquare(runningArray);
                    printLists();
                }
            }

            else if (i>4 && (i%2!=0)){
                System.out.println("\nMove" + (i+1) + ": Machine\n");
                if (possWin("c")) {     //winning possibility of computer
                    printRunningSquare(runningArray);
                    printLists();
                    System.out.println("\nMachine wins !\n");
                    break;
                }
                else if (possWin("h")){     //winning possibility of human
                    printRunningSquare(runningArray);
                    printLists();
                    checkDraw();
                }
                else {      //if both doesn't persist, machine will go to corners
                    GotoCorners();
                    printRunningSquare(runningArray);
                    printLists();
                    checkDraw();
                }
            }
        }
    }
  }

  private static boolean possWin(String winCheck) {      //initializing boolean function possWin

     if (winCheck.equals("c")){      //winning possibility of computer
        if (computerList.size()==2){       //checking for the case when computerList contains 2
elements
            int sum = computerList.get(0)+computerList.get(1);      //sum of 2 numbers in
computerList
            int diff = 15-sum;
```

```
            if (diff<0 || diff>9) return false;
            else {
                for (int x=0; x<9; x++){
                    if (magicSquareArray[x]==diff && runningArray[x].equals("_")) {     //if
computer finds empty position then it will play in that position and win
                        runningArray[x] = "x";
                        computerList.add(magicSquareArray[x]);
                        return true;
                    }
                }
                return false;
            }
        }
        else if (computerList.size()==3){       //checking for the case when computerList contains 3
elements
            int[] sum = new int[]{0,0,0};       //sum array
            int[] diff = new int[]{0,0,0};      //difference array

            sum[0] = computerList.get(0)+computerList.get(1);
            sum[1] = computerList.get(1)+computerList.get(2);
            sum[2] = computerList.get(2)+computerList.get(0);

            for (int i=0; i<3; i++){        //first loop compute difference array
                diff[i] = 15-sum[i];
                for (int x=0; x<9; x++){        //second loop will check the winning possibilities
                    if (magicSquareArray[x]==diff[i] && runningArray[x].equals("_")) {
                        runningArray[x] = "x";
                        computerList.add(magicSquareArray[x]);
                        return true;
                    }
                }
            }
            return false;

        }
        else if (computerList.size()==4){       //checking for the case when computerList contains 4
elements
            int[] sum = new int[]{0,0,0,0,0,0};       //sum array
            int[] diff = new int[]{0,0,0,0,0,0};      //difference array

            sum[0] = computerList.get(0)+computerList.get(1);
            sum[1] = computerList.get(1)+computerList.get(2);
            sum[2] = computerList.get(2)+computerList.get(3);
            sum[3] = computerList.get(0)+computerList.get(2);
            sum[4] = computerList.get(0)+computerList.get(3);
            sum[5] = computerList.get(1)+computerList.get(3);

            for (int i=0; i<6; i++){        //first loop compute difference array
                diff[i] = 15-sum[i];
                for (int x=0; x<9; x++){        //second loop will check the winning possibilities
                    if (magicSquareArray[x]==diff[i] && runningArray[x].equals("_")) {
                        runningArray[x] = "x";
```

```
                    computerList.add(magicSquareArray[x]);
                    return true;
                }
            }
        }
        return false;

    }
}
else if (winCheck.equals("h")){     //winning possibility of human

    if (humanList.size()==2){
        int sumHuman = humanList.get(0)+humanList.get(1);
        int diffHuman = 15-sumHuman;

        if (runningArray[5].equals("o") && runningArray[7].equals("o")) {       //IMPORTANT
exception case
            runningArray[8] = "x";
            computerList.add(magicSquareArray[8]);
            return true;
        }
        else if (diffHuman<0 || diffHuman>9) return false;

        else {      //blocking the winning possibility of human
            for (int x=0; x<9; x++){
                if (magicSquareArray[x]==diffHuman && runningArray[x].equals("_")) {
                    runningArray[x] = "x";
                    computerList.add(magicSquareArray[x]);
                    return true;
                }
            }
            return false;
        }
    }
    else if (humanList.size()==3){      //checking for the case when humanList contains 3
elements
        int[] sum = new int[]{0,0,0};
        int[] diff = new int[]{0,0,0};

        sum[0] = humanList.get(0)+humanList.get(1);
        sum[1] = humanList.get(1)+humanList.get(2);
        sum[2] = humanList.get(2)+humanList.get(0);

        for (int i=0; i<3; i++){
            diff[i] = 15-sum[i];
            for (int x=0; x<9; x++){
                if (magicSquareArray[x]==diff[i] && runningArray[x].equals("_")) {
                    runningArray[x] = "x";
                    computerList.add(magicSquareArray[x]);
                    return true;
                }
            }
```

```
            }
            return false;
        }
        else if (humanList.size()==4){      //checking for the case when humanList contains 4 elements
            int[] sum = new int[]{0,0,0,0,0,0};
            int[] diff = new int[]{0,0,0,0,0,0};

            sum[0] = humanList.get(0)+humanList.get(1);
            sum[1] = humanList.get(1)+humanList.get(2);
            sum[2] = humanList.get(2)+humanList.get(3);
            sum[3] = humanList.get(0)+humanList.get(2);
            sum[4] = humanList.get(0)+humanList.get(3);
            sum[5] = humanList.get(1)+humanList.get(3);

            for (int i=0; i<6; i++){
                diff[i] = 15-sum[i];
                for (int x=0; x<9; x++){
                    if (magicSquareArray[x]==diff[i] && runningArray[x].equals("_")) {
                        runningArray[x] = "x";
                        computerList.add(magicSquareArray[x]);
                        return true;
                    }
                }
            }
            return false;
        }
    }
    return false;
  }

  private static void inputHuman(int i) {     //method declaration for human input
    System.out.println("\nMove " + (i+1) + ": Human, Input index from 1 to 9\n");
    Scanner input  = new Scanner(System.in);
    int humanInput = input.nextInt();

    while (humanInput<1 || humanInput>9){      //human input should be between 1 to 9
        System.out.println("please enter the correct input");
        humanInput = input.nextInt();
    }

    while (!runningArray[humanInput-1].equals("_")){       //human input should not override an already taken input
        System.out.println("Input already taken, please try again");
        humanInput = input.nextInt();
    }

    runningArray[humanInput-1] = "o";
    printRunningSquare(runningArray);
    humanList.add(magicSquareArray[humanInput-1]);      //adding elements of humanInput to humanList
    printLists();
```

```
}

private static void GotoCorners() {     //GotoCorners method call
    if (runningArray[0].equals("_")) {  //will try to occupy corners first, then the centres
        runningArray[0] = "x";
        computerList.add(magicSquareArray[0]);
    }
    else if (runningArray[2].equals("_")) {
        runningArray[2] = "x";
        computerList.add(magicSquareArray[2]);
    }
    else if (runningArray[6].equals("_")) {
        runningArray[6] = "x";
        computerList.add(magicSquareArray[6]);
    }
    else if (runningArray[8].equals("_")) {
        runningArray[8] = "x";
        computerList.add(magicSquareArray[8]);
    }
    else if (runningArray[1].equals("_")) {
        runningArray[1] = "x";
        computerList.add(magicSquareArray[1]);
    }
    else if (runningArray[3].equals("_")) {
        runningArray[3] = "x";
        computerList.add(magicSquareArray[3]);
    }
    else if (runningArray[5].equals("_")) {
        runningArray[5] = "x";
        computerList.add(magicSquareArray[5]);
    }
    else if (runningArray[7].equals("_")) {
        runningArray[7] = "x";
        computerList.add(magicSquareArray[7]);
    }
}

private static void printRunningSquare(String[] array) {    //prints the tic tac toe square board
    for (int i=0;i<array.length; i++){
        if ((i+1)%3 == 0) System.out.print(array[i] + " \n");
        else System.out.print(array[i] + " ");
    }
}

private static void printMagicSquare(int[] array) {     //prints the magicSquareArray
    for (int i=0;i<array.length; i++){
        if ((i+1)%3 == 0) System.out.print(array[i] + " \n");
        else System.out.print(array[i] + " ");
    }
}

private static void printLists() {      //prints the list of human and computer
```

```
        System.out.println("\n" + "computer list : " + computerList);
        System.out.println("human list : " + humanList + "\n");
    }

    private static void checkDraw() {
        int count = 0;
        for (int i=0; i<9; i++){
            if (!runningArray[i].equals("_")) ++count;
        }
        if (count==9) System.out.println("\nMatch Draw !\n");
    }
}
```