import java.io.\*;

class Chess

{

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

{

InputStreamReader I= new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(I);

int ffd,tfd,fsd,tsd;

int valid,validm,validp;

int GameSwitch=1;

int WayCheaker;

int GameType;

int from,to;

int count=0;

char CharToMove;

char Turn='W';

String BR=" BR9 ";

String BH1=" BH15 ";

String BH2=" BH25 ";

String BG1=" BG13 ";

String BG2=" BG23 ";

String BU1=" BU14 ";

String BU2=" BU24 ";

String BW=" BW8 ";

String BS1=" BS11 ";

String BS2=" BS21 ";

String BS3=" BS31 ";

String BS4=" BS41 ";

String BS5=" BS51 ";

String BS6=" BS61 ";

String BS7=" BS71 ";

String BS8=" BS81 ";

String WR=" WR9 ";

String WH1=" WH15 ";

String WH2=" WH25 ";

String WG1=" WG13 ";

String WG2=" WG23 ";

String WU1=" WU14 ";

String WU2=" WU24 ";

String WW=" WW8 ";

String WS1=" WS11 ";

String WS2=" WS21 ";

String WS3=" WS31 ";

String WS4=" WS41 ";

String WS5=" WS51 ";

String WS6=" WS61 ";

String WS7=" WS71 ";

String WS8=" WS81 ";

String[][] brd=new String[9][9];

brd[0][0]=" ";

brd[0][1]=" 1 ";

brd[0][2]=" 2 ";

brd[0][3]=" 3 ";

brd[0][4]=" 4 ";

brd[0][5]=" 5 ";

brd[0][6]=" 6 ";

brd[0][7]=" 7 ";

brd[0][8]=" 8 ";

brd[1][0]="1";

brd[2][0]="2";

brd[3][0]="3";

brd[4][0]="4";

brd[5][0]="5";

brd[6][0]="6";

brd[7][0]="7";

brd[8][0]="8";

brd[1][1]=BH1;

brd[1][2]=BG1;

brd[1][3]=BU1;

brd[1][4]=BW;

brd[1][5]=BR;

brd[1][6]=BU2;

brd[1][7]=BG2;

brd[1][8]=BH2;

brd[2][1]=BS1;

brd[2][2]=BS2;

brd[2][3]=BS3;

brd[2][4]=BS4;

brd[2][5]=BS5;

brd[2][6]=BS6;

brd[2][7]=BS7;

brd[2][8]=BS8;

brd[8][1]=WH1;

brd[8][2]=WG1;

brd[8][3]=WU1;

brd[8][4]=WW;

brd[8][5]=WR;

brd[8][6]=WU2;

brd[8][7]=WG2;

brd[8][8]=WH2;

brd[7][1]=WS1;

brd[7][2]=WS2;

brd[7][3]=WS3;

brd[7][4]=WS4;

brd[7][5]=WS5;

brd[7][6]=WS6;

brd[7][7]=WS7;

brd[7][8]=WS8;

for(int y=3;y<=6;y++)

{

for(int z=1;z<=8;z++)

{

brd[y][z]=" ";

}

}

System.out.println("Welcome to Shubham's Chess Game");

System.out.println("Enter Your Game Type.\n 1. For 2Players.\n 2. For SinglePlayer.");

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

if(GameType==1)

{

do

{

if(count%2==0)

{

Turn='W';

}

else

{

Turn='B';

}

for(int i=0;i<9;i++)

{

for(int j=0;j<=8;j++)

{

System.out.print(brd[i][j]);

System.out.print(":");

}

System.out.print("\n........................................................................\n");

}

valid=0;

validm=0;

validp=0;

do

{

System.out.println("Enter Your Move:");

System.out.print(" From- ");

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

System.out.print(" To- ");

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

ffd=from/10;

fsd=from%10;

tfd=to/10;

tsd=to%10;

if(ffd>-1&&ffd<9&&tfd>-1&&tfd<9&&fsd>-1&&fsd<9&&tsd>-1&&tsd<9)

{

validm=1;

}

else

{

System.out.println("\n \n \nLocations Out Of Range. Try Again:");

}

}while(validm!=1);

do

{

validp=0;

if(Turn==brd[ffd][fsd].charAt(2))

{

validp=1;

}

else if(brd[ffd][fsd].charAt(2)=='B')

{

System.out.println("It Is The Turn of White Player. Enter Move Again.");

do

{

validm=0;

System.out.println("Enter Your Move:");

System.out.print(" From- ");

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

System.out.print(" To- ");

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

ffd=from/10;

fsd=from%10;

tfd=to/10;

tsd=to%10;

if(ffd>-1&&ffd<9&&tfd>-1&&tfd<9&&fsd>-1&&fsd<9&&tsd>-1&&tsd<9)

{

validm=1;

}

else

{

System.out.println("\n \n \nLocations Out Of Range. Try Again:");

}

}while(validm!=1);

}

else if(brd[ffd][fsd].charAt(2)=='W')

{

System.out.println("It Is The Turn of Black Player(NO RACISM). Enter Move Again.");

do

{

validm=0;

System.out.println("Enter Your Move:");

System.out.print(" From- ");

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

System.out.print(" To- ");

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

ffd=from/10;

fsd=from%10;

tfd=to/10;

tsd=to%10;

if(ffd>-1&&ffd<9&&tfd>-1&&tfd<9&&fsd>-1&&fsd<9&&tsd>-1&&tsd<9)

{

validm=1;

}

else

{

System.out.println("\n \n \nLocations Out Of Range. Try Again:");

}

}while(validm!=1);

}

}while(validp!=1);

do

{

valid=0;

CharToMove=brd[ffd][fsd].charAt(3);

if(CharToMove=='S')

{

if(brd[ffd][fsd].charAt(2)=='B')

{

if(ffd==2)

{

if(tfd==ffd+2&&tsd==fsd||tfd==ffd+1&&tsd==fsd)

{

if(brd[tfd][tsd].charAt(2)!='W')

{

valid=1;

}

else

{

System.out.println("Pawns Cannot Kill Going Straight. Enter Another Move");

}

}

}

else

{

if(tfd==ffd+1&&tsd==fsd)

{

if(brd[tfd][tsd].charAt(2)!='W')

{

valid=1;

}

else

{

System.out.println("Pawns Cannot Kill Going Straight. Enter Another Move");

}

}

}

if(tfd==ffd+1&&tsd==fsd+1||tfd==ffd+1&&tsd==fsd-1)

{

if(brd[tfd][tsd].charAt(2)=='W')

{

valid=1;

}

}

}

else

{

if(ffd==7)

{

if(tfd==ffd-2&&tsd==fsd||tfd==ffd-1&&tsd==fsd)

{

if(brd[tfd][tsd].charAt(2)!='B')

{

valid=1;

}

else

{

System.out.println("Pawns Cannot Kill Going Straight. Enter Another Move");

}

}

}

else

{

if(tfd==ffd+1&&tsd==fsd)

{

if(brd[tfd][tsd].charAt(2)!='B')

{

valid=1;

}

else

{

System.out.println("Pawns Cannot Kill Going Straight. Enter Another Move");

}

}

}

if(tfd==ffd-1&&tsd==fsd-1||tfd==ffd-1&&tsd==fsd+1)

{

if(brd[tfd][tsd].charAt(2)=='B')

{

valid=1;

}

}

}

}

if(CharToMove=='G')

{

if(tfd==ffd-2&&tsd==fsd+1||tfd==ffd-2&&tsd==fsd-1||tfd==ffd-1&&tsd==fsd+2||tfd==ffd-1&&tsd==fsd-2||tfd==ffd+2&&tsd==fsd+1||tfd==ffd+2&&tsd==fsd-1||tfd==ffd+1&&tsd==fsd+2||tfd==ffd+1&&tsd==fsd-2)

{

valid=1;

}

}

if(CharToMove=='W')

{

if(tfd==ffd+1&&tsd==fsd+1||tfd==ffd-1&&tsd==fsd-1||tfd==ffd+1&&tsd==fsd-1||tfd==ffd-1&&tsd==fsd+1||tfd==ffd+1&&tsd==fsd||tfd==ffd&&tsd==fsd+1||tfd==ffd-1&&tsd==fsd||tfd==ffd&&tsd==fsd-1)

{

valid=1;

}

if(valid==0)

{

for(int y=2;y<=7;y++)

{

if(tfd==ffd+y&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd+WayCheaker].charAt(2)=='B'||brd[ffd+WayCheaker][fsd+WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd-WayCheaker].charAt(2)=='W'||brd[ffd-WayCheaker][fsd-WayCheaker].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd+y&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd-WayCheaker].charAt(2)=='B'||brd[ffd+WayCheaker][fsd-WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd+WayCheaker].charAt(2)=='W'||brd[ffd-WayCheaker][fsd+WayCheaker].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd+y&&tsd==fsd)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd].charAt(2)=='W'||brd[ffd+WayCheaker][fsd].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd][fsd+WayCheaker].charAt(2)=='W'||brd[ffd][fsd+WayCheaker].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd].charAt(2)=='W'||brd[ffd-WayCheaker][fsd].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd][fsd-WayCheaker].charAt(2)=='B'||brd[ffd][fsd-WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

}

}

}

if(CharToMove=='H')

{

if(tfd==ffd+1&&tsd==fsd||tfd==ffd&&tsd==fsd+1||tfd==ffd-1&&tsd==fsd||tfd==ffd&&tsd==fsd-1)

{

valid=1;

}

if(valid==0)

{

for(int y=2;y<=8;y++)

{

if(tfd==ffd+y&&tsd==fsd)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd].charAt(2)=='W'||brd[ffd+WayCheaker][fsd].charAt(2)=='B')

{

break;

}

valid=1;

}

}

if(tfd==ffd&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd][fsd+WayCheaker].charAt(2)=='B'||brd[ffd][fsd+WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd].charAt(2)=='B'||brd[ffd-WayCheaker][fsd].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd][fsd-WayCheaker].charAt(2)=='B'||brd[ffd][fsd-WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

}

}

}

if(CharToMove=='U')

{

if(tfd==ffd+1&&tsd==fsd+1||tfd==ffd-1&&tsd==fsd-1||tfd==ffd+1&&tsd==fsd-1||tfd==ffd-1&&tsd==fsd+1)

{

valid=1;

}

if(valid==0)

{

for(int y=2;y<=8;y++)

{

if(tfd==ffd+y&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd+WayCheaker].charAt(2)=='B'||brd[ffd+WayCheaker][fsd+WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd-WayCheaker].charAt(2)=='B'||brd[ffd-WayCheaker][fsd-WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd+y&&tsd==fsd-y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd+WayCheaker][fsd-WayCheaker].charAt(2)=='B'||brd[ffd+WayCheaker][fsd-WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

if(tfd==ffd-y&&tsd==fsd+y)

{

for(WayCheaker=1;WayCheaker<y;WayCheaker++)

{

if(brd[ffd-WayCheaker][fsd+WayCheaker].charAt(2)=='B'||brd[ffd-WayCheaker][fsd+WayCheaker].charAt(2)=='W')

{

break;

}

valid=1;

}

}

}

}

}

if(CharToMove=='R')

{

if(tfd==ffd+1&&tsd==fsd||tfd==ffd-1&&tsd==fsd||tfd==ffd&&tsd==fsd+1||tfd==ffd&&tsd==fsd-1||tfd==ffd+1&&tsd==fsd+1||tfd==ffd-1&&tsd==fsd-1||tfd==ffd-1&&tsd==fsd+1||tfd==ffd+1&&tsd==fsd-1)

{

valid=1;

}

}

if(valid==1)

{

do

{

do

{

if(brd[tfd][tsd].charAt(2)==brd[ffd][fsd].charAt(2))

{

System.out.println("Please Do Not Try To Kill Yourself. Enter Move Again");

System.out.print(" From- ");

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

System.out.print(" To- ");

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

ffd=from/10;

fsd=from%10;

tfd=to/10;

tsd=to%10;

if(Turn==brd[ffd][fsd].charAt(2))

{

validp=1;

}

else

{

System.out.println("Not Your Turn. Another Player's Turn");

validp=0;

}

}

}while(validp!=1);

}while(brd[tfd][tsd].charAt(2)==brd[ffd][fsd].charAt(2));

if(brd[tfd][tsd].charAt(3)=='R')

{

GameSwitch=0;

if(brd[(to/10)][(to%10)].charAt(2)=='W')

{

System.out.println("Congratulations To Black Player(No Racism). You Won :)");

}

else

{

System.out.println("Congratulations To White Player. You Won :)");

}

break;

}

else

{

brd[tfd][tsd]=brd[ffd][fsd];

brd[ffd][fsd]=" ";

}

}

else

{

do

{

System.out.println("\n \n \nYour Last Move Was Not a Valid One. Try Again:");

System.out.println("Enter Your Move Again:");

System.out.print(" From- ");

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

System.out.print(" To- ");

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

ffd=from/10;

fsd=from%10;

tfd=to/10;

tsd=to%10;

if(Turn==brd[ffd][fsd].charAt(2))

{

validp=1;

}

else

{

System.out.println("Not Your Turn. Another Player's Turn");

validp=0;

}

}while(validp!=1);

}

}while(valid==0);

if(GameSwitch==0)

{

break;

}

System.out.println("\n\n\n");

count++;

}while(1>0);

}

}

}