Twelve Man Morris

New Mexico Supercomputing Challenge
Final Report
April 4, 2007

Team 9
School: Albuquerque Academy

Team Members:
Brandon Oselio
Vlado Ovtcharov

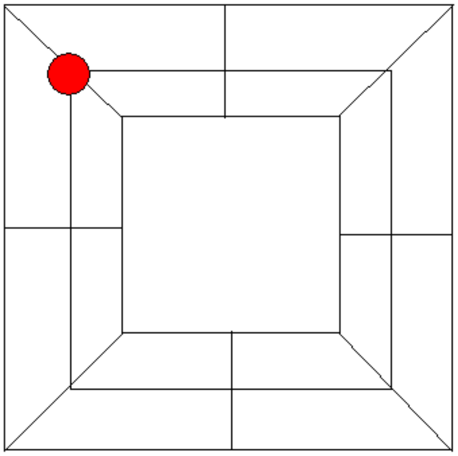
Sponsoring teacher: Jim Mims
Project Mentor: Jim Mims

Table of Contents

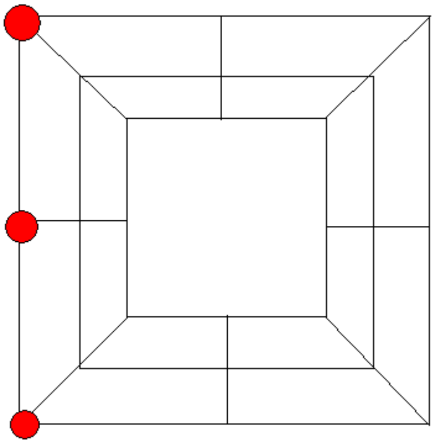
I. Introduction

A. Rules

Twelve Man Morris is a two player board game. An example of the board is shown below. A piece can be put on any location where two or more lines intersect. A red piece has been placed on one of these. This creates a total of 24 possible spots a piece can go. During the first stage of the game each player puts a piece on the board on an unoccupied space.



If they get three pieces in a row, vertically, horizontally or diagonally, as shown below, they can the remove a piece from the opponent off the board. After twelve turns of this the game advances to the second stage.



In the second stage of the game each player is allowed to move one piece to an adjacent open place. Each player takes turns doing this. If a line is formed with the newly moved piece then once again that player gets to remove a piece from the opponent. This continues until one of the players only has three pieces left. At this point the player with only three pieces on the board during his turn is allowed to move one of there pieces anywhere on the board, not only in adjacent places. Once again if a line is formed with a newly moved piece the player gets to remove a piece from the opponent. Once a player only has two pieces left or cannot move anywhere the game is over.

B. Purpose

The purpose of this project is to create a program where the computer can play against a human according to all the rules mentioned above. The computer should also be able to think strategically and make moves which are considered intelligent.

C. Significance

This is for entertainment. The significance is to let the player have fun playing against a challenging opponent.

II. How Our Program Works

A. Description

The computer makes decisions by looking ahead three moves and analyzing the situation to determine what outcome is best for it. It can play against a human player, and act as an intelligent opponent. The AI turns out to be the equivalent of a moderate or easy opponent, but the user can play the entire game with the computer and the computer will respond properly.

B. Materials

Visual C++

Previous Class experienced, including Visual C++ as well as specialized topics, with some limited AI.

C. Methods & Mathematical Modeling

The first thing we did was create constraints on where the computer could move so that it is not breaking any rules. In order to do this we analyzed the board and looked at all the empty spaces and dimmed the possible moves for the first stage and third stage. For the second stage we took these empty spaces and checked if the computer had any pieces adjacent to it and dimmed those as possible moves.

We decided to use a heuristic algorithm to solve the problem. Since we knew we could not look at all the possible board positions in a reasonable time we decided to only look at a few. The more computer power we have the more boards we could look at, but at this time we decided only to look ahead three moves. We do this by looking at all the possible moves the computer can make, and then on a fake board, make them. We then take this fake board and store it to an array. We then take each of those boards and do the same thing again. We do this a total of three times. The space required to do this grows linearly the more boards we look at, however the processing time grows exponentially. For example if we look ahead one move we have to store about 24 boards and analyze 24 boards. However if we were to look two moves ahead we would have two store 48 boards at one time but analyze about 576 (24^2). This number grows very fast so we can only analyze so many boards. How long we analyze each board comes into account. We can analyze one aspect much faster then ten aspects. This then allows us to look further ahead, but only looking at one aspect. We decided to only look at those aspects that are very important and not look at the others which come into effect very rarely.

D. Computational modeling

We use a 3X8 array to store the information on the board. For each move we look ahead we then create another array of size 24 (if stage one) or 48 (stage two). The computer looks ahead by checking which moves are valid and then pretends to go there. We do this by setting the board to one of the possible moves, then sending that board to get analyzed and then once analyzed we change the board back to the way it was. Simplified that looks like,

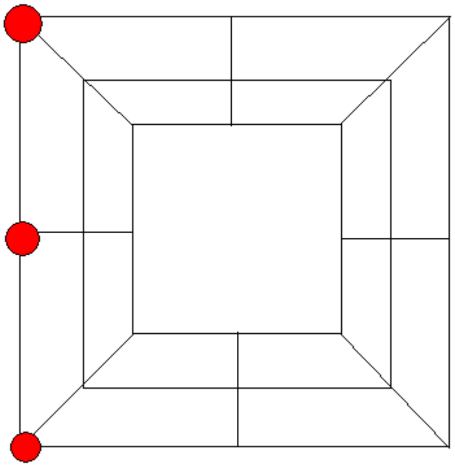
```
board[0][3] =1;
analyze(board);
board[0][3] =0;
```

We store computer pieces as 1's and player pieces a 2's in the board array. We also create an array for each move we look ahead to store the value, (how good of a move), that board is. We determine how good of a move is based on this criteria.

the player has made a line

the computer has made a line

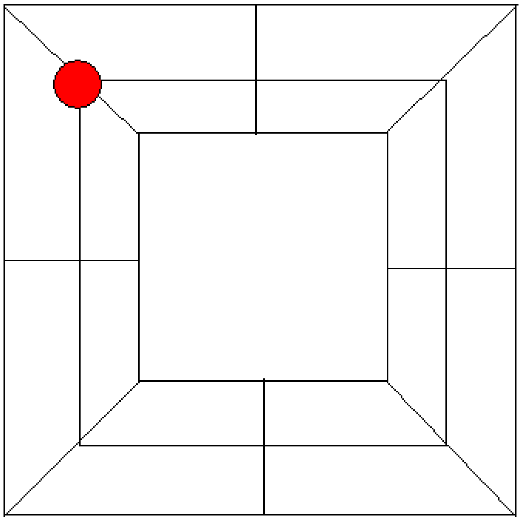
example below



how many spaces the computer can move to

how many spaces the player can move to

example below, the piece can move to 4 other places



the value of the spot the player's pieces are on

the value of the spot the computer's pieces are on example above, piece can make three lines

We then go through all of these values and determine which is the best for the computer by taking the highest valued board.

III.Results and Conclusion

We were able to create a program which obeys all the rules and plays as a fairly intelligent human. It is able to block and create simple strategies in the game. The computer can do very well during the first stage of the game but is a little weaker during the second. The second stage of the game is more complex and looking ahead only three moves is not always enough. If we had more time we would have added graphics and made the second stage of the game AI better. However, we have made significant accomplishments in the AI, even though it was not entirely complete. With looking ahead just three moves, we were able to create a realistic simulation for the player of what a real game of Twelve Man Morris would be like. From this project we will take valuable knowledge of AI and experience in implementing code. While you normally wouldn’t create a game in a classroom, we were able to experience it first hand, and we were able to see some of the difficulties that game programmers face. We feel confident that this process will help us throughout our scientific career and will give us valuable experience in high school where there usually is none.

IV. Appendix

A. Code

```
#include<iostream.h>
#include <stdlib.h>
#include<fstream.h>
 ofstream examplefile ("c:example.txt");
void firststagegame(int board[3][8]);                  //setting down
all twelve pieces
 void computerlocationstageone(int board[3][8]);  //look ahead one move
 void computerlocationstagetwo(int board[3][8], int level, int
position, int & value); //look ahead two moves
 void computerlocationstagethree(int level, int position, int
board[3][8],int  i,int  j, int & value); //look ahead three moves
 void analyzecomputerlocationstagefour(int board[3][8], int ci, int cj,
int & cvalue); //analyze the board
    void computerlocation(int board[3][8], int firstlevel[3][8]);
//put down piece
                void checkcomputerlines(int board[3][8], int levelnew,
int positionnew); // checks if the computer has made a line
                        void computertakepiece(int board[3][8]);
//has the computer take piece
 void playerlocation(int board[3][8]);           //player move stage
one
    void checkplayerlines(int board[3][8], int levelnew, int
positionnew); //checks if player has made a line
      void playertakepiece(int board[3][8]);  //makes the plater take a
piece
```

```
void secondstagegame(int board[3][8]);    //second stage of the game
   void computermovestageone(int board[3][8]);  //looks ahead one move
stage two
   void computermovestagetwo(int holder[3][8], int & value, int
levelnewone, int positionnewone);  //looks ahead two move stage two
   void computermovestagethree(int holder[3][8], int & secondvalue, int
levelnewone, int positionnewone, int levelnewtwo, int positionnewtwo);
//looks ahead three move stage two
   void analyzecomputermovestagefour(int holder[3][8],int levelnewone,
int positionnewone, int levelnewtwo, int positionnewtwo, int
levelnewthree, int positionnewthree, int & thirdvalue);  //analyzes the
board stage two

   void playermove(int board[3][8]); //makes the player move

   void computerhop(int board[3][8]); //makes the computer fly, stage
three
   void playerhop(int board[3][8]); //makes the computer fly, stage
three


void clearboard(int board[3][8]);
//clears array
void printboard(int board[3][8]);
//outputs the board
void clearboard2(int firstlevel[3][8]);       //clears array
void hopperchecker(bool & playerhopcheck, bool & computerhopcheck,  int
board[3][8]);  //checks too see if the computer has entered the fly
stage
void checkforwinner(bool & playerwinner, bool & computerwinner,int
board[3][8]); //checks too see if the player has entered the fly stage
//checks to see if lines are made while looking ahead

void checkcomputerlineslookingahead(int board[3][8], int levelnew, int
positionnew, bool & lines); //checks to see if the computer has made a
line while looking ahead
void checkplayerlineslookingahead(int board[3][8], int levelnew, int
positionnew, bool & lines); //checks to see if the player has made a
line while looking ahead

void computerlocationstagetwolines(int board[3][8], int level, int
position, int & value); //checks to see if the computer has made a line
while looking ahead during second stage
void computerlocationstagethreelines(int level, int position, int
board[3][8],int  i,int  j, int & value); //checks to see if the player
has made a line while looking ahead during second stage

//board analyzers
void compthreeinarow(int board[3][8], int & cvalue); //adds points to
value if the computer makes a line while looking ahead
void playerthreeinarow(int board[3][8], int & cvalue); //subtracts
points to value if the player makes a line while looking ahead
void analyzecheckmoveability(int holder[3][8], int & thirdvalue);
//checks how many places the computer can move to
void analyzecheckmoveabilityplayer(int holder[3][8], int & thirdvalue);
//checks how many places the player can move to
```

```
void analyzecomputerboardlocation(int board[3][8], int & thirdvalue);
//checks to see what the values of the spots that the computer is on
are worth
void analyzeplayerboardlocation(int board[3][8], int & thirdvalue);
//checks to see what the values of the spots that the player is on are
worth
void analyzecheckcomputerlines(int holder[3][8],int levelnewone, int
positionnewone, int levelnewthree, int positionnewthree, int &
thirdvalue); //checks if the computer has made a line during the final
looking ahead stage
void analyzecheckplayerlines(int holder[3][8], int levelnewtwo, int
positionnewtwo, int & thirdvalue); //checks if the player has made a
line during the final looking ahead stage
int main()
{
int board[3][8];
clearboard(board);
firststagegame(board);
secondstagegame(board);

        return 0;
}

void firststagegame(int board[3][8])
{
        for (int i=0; i<4;i++)
        {
                computerlocationstageone(board);

                printboard(board);

                playerlocation(board);


        }
}

void secondstagegame(int board[3][8])
{
bool playerhopcheck = false;
bool computerhopcheck= false;
bool playerwinner = false;
bool computerwinner= false;
do
{
hopperchecker(playerhopcheck, computerhopcheck, board);
    checkforwinner(playerwinner, computerwinner, board);
    if(playerwinner == true)
    {
     cout<<"PLAYER HAS WON"<<endl;
    }
    if(computerwinner == true)
    {
     cout<<"COMPUTER HAS WON"<<endl;
    }
if( computerhopcheck == false)
{
```

```
computermovestageone(board);
}
if( computerhopcheck == true)
{
computerhop(board);
}

printboard(board);
hopperchecker(playerhopcheck, computerhopcheck, board);
    checkforwinner(playerwinner, computerwinner, board);
    if(playerwinner == true)
    {
     cout<<"PLAYER HAS WON!!!!!"<<endl;
    }
    if(computerwinner == true)
    {
     cout<<"COMPUTER HAS WON!!!!!"<<endl;
    }
if( playerhopcheck == false)
{
playermove(board);
}
if( playerhopcheck == true)
{
playerhop(board);
}

}
while(playerwinner ==false && computerwinner == false);
}
void checkforwinner(bool & playerwinner, bool & computerwinner,int
board[3][8])
{
int player=0;
int computer=0;
        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
     if(board[i][j] == 1)
     {
      computer++;
     }
     if(board[i][j] == 2)
     {
      player++;
     }
    }
  }
if (player == 2)
{
computerwinner = true;
}
if (computer == 2)
{
playerwinner = true;
}
```

```
}
void hopperchecker(bool & playerhopcheck, bool & computerhopcheck,int 
board[3][8])
{
int player=0;
int computer=0;
        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
     if(board[i][j] == 1)
     {
      computer++;
     }
     if(board[i][j] == 2)
     {
      player++;
     }
    }
  }
if (player == 3)
{
playerhopcheck = true;
}
if (computer == 3)
{
computerhopcheck = true;
}
}
void clearboard(int board[3][8])
{
        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
                        board[i][j] =0;

                }
        }
}

void clearboard2(int firstlevel[3][8])
{
        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
                        firstlevel[i][j] =0;

                }
        }
}
void printboard(int board[3][8])
{
cout<<          board[0][0]<<"     "
<<board[0][1]<<"     "
<<board[0][2]<<"     "<<endl;
```

```
cout<<"  "
<<board[1][0]<<"   "
<<board[1][1]<<"   "
<<board[1][2]<<"  "                                      <<endl;
cout<<"    "
<<board[2][0]<<" "      <<board[2][1]<<" "      <<board[2][2]<<" "
<<endl;
cout<<          board[0][3]<<" "        <<board[1][3]<<" "
<<board[2][3]<<"   "
<<board[2][4]<<" "      <<board[1][4]<<" "      <<board[0][4]<<" "
<<endl;
cout<<"    "
<<board[2][5]<<" "      <<board[2][6]<<" "      <<board[2][7]<<" "
<<endl;
cout<<"  "
<<board[1][5]<<"   "
<<board[1][6]<<"   "
<<board[1][7]<<"  "                                      <<endl;
cout<<          board[0][5]<<"     "
<<board[0][6]<<"     "
<<board[0][7]<<"     "<<endl;
}




void computerlocation(int board[3][8], int firstlevel[3][8])
{
int positionnew =0;
int levelnew=0;
int temp =-10000;
bool work =false;
bool worksecondstage = false;
        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
     if(board[i][j] ==0)
     {
      work = true;


                        if(temp <= firstlevel[i][j])
                        {
        worksecondstage = true;
        temp = firstlevel[i][j];
                                positionnew = j;
                                levelnew =i;

                        }
     }

                }
        }

if (work == false)
```

```
{
cout<<"GAME OVER"<<endl;
}

if ( worksecondstage == false)
{

        for (int i=0; i<3; i++)
        {
                for(int j=0; j<8; j++)
                {
                        if(board[i][j] == 0)
                        {
                                positionnew = j;
                                levelnew =i;

                        }
                }
        }

}

board[levelnew][positionnew]=1;
checkcomputerlines(board, levelnew, positionnew);
return;
}


void checkcomputerlines(int board[3][8], int levelnew, int positionnew)
{
bool lines = false;
if (levelnew ==0)
{
        if(positionnew == 0)
        {
                if(board[0][0] == 1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[0][0] ==1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[0][0] ==1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
}
if (levelnew ==1)
{
        if(positionnew == 0)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
                lines= true;
                }
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
}
```

```
if (levelnew ==2)
{
        if(positionnew == 0)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
                lines= true;
                }
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
}

        if (lines == true)
        {
                computertakepiece(board);
        }

}

void computertakepiece(int board[3][8])
{
int level;
int position;
int value =0;

bool lines[3][8];
int firstlevel[3][8];


for( int levelnew=0; levelnew<3; levelnew++)
{
 for(int positionnew=0; positionnew<8; positionnew++)
 {
  firstlevel[levelnew][positionnew] =0;
  lines[levelnew][positionnew] =true;
  if (board[levelnew][positionnew] ==2)
  {
   if (levelnew ==0)
   {
     if(positionnew == 0)
     {
       if(board[0][0] == 1 && board[0][1] ==2 && board[0][2] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
       if(board[0][0] ==2 && board[1][0] ==2 && board[2][0] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
       if(board[0][0] ==2 && board[0][3] ==2 && board[0][5] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
     }
     if(positionnew == 1)
```

```
{
  if(board[0][0] ==2 && board[0][1] ==2 && board[0][2] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][1] ==2 && board[1][1] ==2 && board[2][1] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew ==2)
{
  if(board[0][0] ==2 && board[0][1] ==2 && board[0][2] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][2] ==2 && board[1][2] ==2 && board[2][2] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][2] ==2 && board[0][4] ==2 && board[0][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 3)
{
  if(board[0][3] ==2 && board[1][3] ==2 && board[2][3] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][0] ==2 && board[0][3] ==2 && board[0][5] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 4)
{
  if(board[0][4] ==2 && board[1][4] ==2 && board[2][4] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][2] ==2 && board[0][4] ==2 && board[0][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 5)
{
  if(board[0][5] ==2 && board[0][6] ==2 && board[0][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][5] ==2 && board[1][5] ==2 && board[2][5] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
```

```
    if(board[0][0] ==2 && board[0][3] ==2 && board[0][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 6)
  {
    if(board[0][6] ==2 && board[1][6] ==2 && board[2][6] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][5] ==2 && board[0][6] ==2 && board[0][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 7)
  {
    if(board[0][5] ==2 && board[0][6] ==2 && board[0][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][7] ==2 && board[1][7] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][2] ==2 && board[0][4] ==2 && board[0][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
}
if (levelnew ==1)
{
  if(positionnew == 0)
  {
    if(board[1][0] ==2 && board[1][1] ==2 && board[1][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][0] ==2 && board[1][0] ==2 && board[2][0] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[1][0] ==2 && board[1][3] ==2 && board[1][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 1)
  {
    if(board[1][0] ==2 && board[1][1] ==2 && board[1][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][1] ==2 && board[1][1] ==2 && board[2][1] ==2)
    {
```

```
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew ==2)
{
  if(board[1][0] ==2 && board[1][1] ==2 && board[1][2] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][2] ==2 && board[1][2] ==2 && board[2][2] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[1][2] ==2 && board[1][4] ==2 && board[1][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 3)
{
  if(board[0][3] ==2 && board[1][3] ==2 && board[2][3] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[1][0] ==2 && board[1][3] ==2 && board[1][5] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 4)
{
  if(board[0][4] ==2 && board[1][4] ==2 && board[2][4] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[1][2] ==2 && board[1][4] ==2 && board[1][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 5)
{
  if(board[1][5] ==2 && board[1][6] ==2 && board[1][7] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[0][5] ==2 && board[1][5] ==2 && board[2][5] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
  if(board[1][0] ==2 && board[1][3] ==2 && board[1][5] ==2)
  {
  lines[levelnew][positionnew] = false;
  }
}
if(positionnew == 6)
{
    if(board[0][6] ==2 && board[1][6] ==2 && board[2][6] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[1][5] ==2 && board[1][6] ==2 && board[1][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 7)
  {
    if(board[1][5] ==2 && board[1][6] ==2 && board[1][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][7] ==2 && board[1][7] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[1][2] ==2 && board[1][4] ==2 && board[1][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
}
if (levelnew ==2)
{
  if(positionnew == 0)
  {
    if(board[2][0] ==2 && board[2][1] ==2 && board[2][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][0] ==2 && board[1][0] ==2 && board[2][0] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][0] ==2 && board[2][3] ==2 && board[2][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 1)
  {
    if(board[2][0] ==2 && board[2][1] ==2 && board[2][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][1] ==2 && board[1][1] ==2 && board[2][1] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew ==2)
  {
    if(board[2][0] ==2 && board[2][1] ==2 && board[2][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][2] ==2 && board[1][2] ==2 && board[2][2] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][2] ==2 && board[2][4] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 3)
  {
    if(board[0][3] ==2 && board[1][3] ==2 && board[2][3] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][0] ==2 && board[2][3] ==2 && board[2][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 4)
  {
    if(board[0][4] ==2 && board[1][4] ==2 && board[2][4] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][2] ==2 && board[2][4] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 5)
  {
    if(board[2][5] ==2 && board[2][6] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[0][5] ==2 && board[1][5] ==2 && board[2][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][0] ==2 && board[2][3] ==2 && board[2][5] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
  }
  if(positionnew == 6)
  {
    if(board[0][6] ==2 && board[1][6] ==2 && board[2][6] ==2)
    {
    lines[levelnew][positionnew] = false;
    }
    if(board[2][5] ==2 && board[2][6] ==2 && board[2][7] ==2)
    {
    lines[levelnew][positionnew] = false;
```

```
       }
     }
     if(positionnew == 7)
     {
       if(board[2][5] ==2 && board[2][6] ==2 && board[2][7] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
       if(board[0][7] ==2 && board[1][7] ==2 && board[2][7] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
       if(board[2][2] ==2 && board[2][4] ==2 && board[2][7] ==2)
       {
       lines[levelnew][positionnew] = false;
       }
     }
   }


  }

 }
}
for (int i =0; i<3; i++)
{
 for(int j=0; j<8; j++)
 {
  if(lines[i][j] == true && board[i][j] ==2)
  {
 board[i][j]=0;
value=0;
 analyzecheckmoveability(board, value);
 analyzecheckmoveabilityplayer(board, value);
 analyzeplayerboardlocation(board, value);
 analyzecomputerboardlocation(board, value);
   firstlevel[i][j] = value;
 board[i][j]=2;
  }
 }
}
int temp = -1000;
for ( i =0; i<3; i++)
{
 for(int j=0; j<8; j++)
 {
  if(firstlevel[i][j] >= temp && board[i][j] ==2)
  {
   temp = firstlevel[i][j];
   level =i;
   position =j;
  }
 }
}

if(board[level][position] ==2)
{
 board[level][position] =0;
 return;
}
else
{
 for ( i =0; i<3; i++)
 {
  for(int j=0; j<8; j++)
  {
   if(board[i][j] ==2)
   {
    board[i][j] =0;
    return;
   }
  }
 }
}
}



void playerlocation(int board[3][8])
{
int work =false;
int positionnew=0;
int levelnew=0;
do
{
cout<<"Which level would you like to play on?"<<endl;
cin>>levelnew;
cout<<"Which location would you like to play on?"<<endl;
cin>>positionnew;

        if( board[levelnew][positionnew] ==0)
        {
                board[levelnew][positionnew] =2;
                work = true;
        }
}
while(work == false);
checkplayerlines(board, levelnew, positionnew);

}

void checkplayerlines(int board[3][8], int levelnew, int positionnew)
{
bool lines = false;
if (levelnew ==0)
{
        if(positionnew == 0)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2]
==2)
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
}
if (levelnew ==1)
{
        if(positionnew == 0)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3]
==2)
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4]
==2)
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
}
if (levelnew ==2)
{

        if(positionnew == 0)
        {
                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {

                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
```

```
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
}

        if (lines == true)
        {
                playertakepiece(board);
        }

}

void playertakepiece(int board[3][8])
{
bool work = false;
        int level=0;
        int position=0;
do
{
                cout<<"Which level is the piece you want to take
on"<<endl;
                cin>>level;
                cout<<"What is the position of the piece you want to
take"<<endl;
                cin>>position;
if(board[level][position] == 1)
{
work = true;
}
```

```
}
while(work == false);
board[level][position] =0;
}

void computermovestageone(int board[3][8])
{

int levelnewone=0;
int positionnewone=0;
int source =0;
int destination=0;
int firststage[24][24];   //[destination][source]
bool work[24][24];
int holder[3][8];    //stores board data so board does not change
int value=0;
        for (int l=0; l<24; l++)
        {
         for(int m=0; m<24;m++)
         {
         firststage[l][m]=0;
         work[l][m] = false;
         }
        }
        for (l=0; l<3;l ++)
        {
          for(int m=0; m<8; m++)
          {
          holder[l][m] = board[l][m];
          }
        }
for (int i=0; i<3; i++)
{
        for(int j=0; j<8; j++)
        {
                if (holder[i][j] == 0)
                {
                        //corners
                        if(j == 0)
                        {
                                if(i==0)
                                {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][1] =1;
          firststage[i*8+j][1] = value;
          work[i*8+j][1] = true;


                                        }
                                        if(holder[1][0] ==1)
                        {
                        holder[i][j] =1;
                        holder[1][0] =0;
      levelnewone=i;
      positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
      holder[i][j] =0;
                        holder[1][0] =1;
      firststage[i*8+j][8]  =  value;
      work[i*8+j][8] = true;


                        }
                        if(holder[0][3] ==1)
                        {
                        holder[i][j] =1;
                        holder[0][3] =0;
      levelnewone=i;
      positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
      holder[i][j] =0;
                        holder[0][3] =1;
      firststage[i*8+j][3]  =  value;
      work[i*8+j][3] = true;


                        }
                   }
                   if(i==1)
                   {
                        if(holder[0][0] ==1)
                        {
                        holder[i][j] =1;
                        holder[0][0] =0;
      levelnewone=i;
      positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
      holder[i][j] =0;
                        holder[0][0] =1;
      firststage[i*8+j][0]  =  value;
      work[i*8+j][9] = true;


                        }
                        if(holder[1][1] ==1)
                        {
                        holder[i][j] =1;
                        holder[1][1] =0;
      levelnewone=i;
      positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
      holder[i][j] =0;
                        holder[1][1] =1;
      firststage[i*8+j][9]  =  value;
      work[i*8+j][9] = true;
```

```
                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][3] =1;
          firststage[i*8+j][11]  =  value;
          work[i*8+j][11] = true;


                                        }
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][0] =1;
          firststage[i*8+j][16]  =  value;
          work[i*8+j][16] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][1] =1;
          firststage[i*8+j][17]  =  value;
          work[i*8+j][17] = true;


                                        }
                                        if(holder[2][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][3] =1;
          firststage[i*8+j][19]  =  value;
          work[i*8+j][19] = true;
```

```
                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[1][0] =1;
             firststage[i*8+j][8]  =  value;
             work[i*8+j][18] = true;


                                        }
                              }
                    }
                    if(j == 2)
                    {
                              if(i==0)
                              {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[0][1] =1;
             firststage[i*8+j][1]  =  value;
             work[i*8+j][1] = true;


                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[1][2] =1;
             firststage[i*8+j][10]  =  value;
             work[i*8+j][10] = true;


                                        }
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
             levelnewone=i;
```

```
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][4] =1;
          firststage[i*8+j][4]  =  value;
          work[i*8+j][4] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][2] =1;
          firststage[i*8+j][2]  =  value;
          work[i*8+j][2] = true;


                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][1] =1;
          firststage[i*8+j][9]  =  value;
          work[i*8+j][9] = true;


                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][4] =1;
          firststage[i*8+j][12]  =  value;
          work[i*8+j][12] = true;


                                        }
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][2] =1;
          firststage[i*8+j][18]  =  value;
          work[i*8+j][18] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][1] =1;
          firststage[i*8+j][17]  =  value;
          work[i*8+j][17] = true;


                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][4] =1;
          firststage[i*8+j][20]  =  value;
          work[i*8+j][20] = true;


                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][2] =1;
          firststage[i*8+j][10]  =  value;
          work[i*8+j][10] = true;


                                        }
                                }
```

```
                        }
                        if(j == 5)
                        {
                                if(i==0)
                                {
                                        if(holder[0][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][3] =1;
          firststage[i*8+j][3]  =  value;
          work[i*8+j][3] = true;


                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][5] =1;
          firststage[i*8+j][13]  =  value;
          work[i*8+j][13] = true;


                                        }
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][6] =1;
          firststage[i*8+j][6]  =  value;
          work[i*8+j][6] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
```

```
          holder[i][j] =0;
                                        holder[0][5] =1;
          firststage[i*8+j][5]  =  value;
          work[i*8+j][5] = true;


                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][3] =1;
          firststage[i*8+j][11]  =  value;
          work[i*8+j][11] = true;


                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][6] =1;
          firststage[i*8+j][14]  =  value;
          work[i*8+j][14] = true;


                                        }
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][5] =1;
          firststage[i*8+j][21]  =  value;
          work[i*8+j][21] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][3] =1;
          firststage[i*8+j][19]  =  value;
          work[i*8+j][19] = true;


                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][5] =1;
          firststage[i*8+j][13]  =  value;
          work[i*8+j][13] = true;


                                        }
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][6] =1;
          firststage[i*8+j][22]  =  value;
          work[i*8+j][22] = true;


                                        }
                                }
                        }
                        if(j == 7)
                        {
                                if(i==0)
                                {
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][6] =1;
          firststage[i*8+j][6]  =  value;
          work[i*8+j][6] = true;


                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][7] =1;
          firststage[i*8+j][15]  =  value;
          work[i*8+j][15] = true;


                                        }
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][4] =1;
          firststage[i*8+j][4]  =  value;
          work[i*8+j][4] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][7] =1;
          firststage[i*8+j][7]  =  value;
          work[i*8+j][7] = true;


                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][6] =1;
          firststage[i*8+j][14]  =  value;
          work[i*8+j][14] = true;
```

```
                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][4] =1;
          firststage[i*8+j][12]  =  value;
          work[i*8+j][12] = true;


                                        }
                                        if(holder[2][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][7] =1;
          firststage[i*8+j][23]  =  value;
          work[i*8+j][23] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][6] =1;
          firststage[i*8+j][22]  =  value;
          work[i*8+j][22] = true;


                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][7] =1;
          firststage[i*8+j][15]  =  value;
          work[i*8+j][15] = true;


                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][4] =1;
          firststage[i*8+j][20]  =  value;
          work[i*8+j][20] = true;


                                        }
                                }
                        }
                        //sides
                        if(j == 1)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][0] =1;
          firststage[i*8+j][0]  =  value;
          work[i*8+j][0] = true;


                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][1] =1;
          firststage[i*8+j][9]  =  value;
          work[i*8+j][9] = true;


                                        }
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][2] =1;
          firststage[i*8+j][2]  =  value;
          work[i*8+j][2] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][1] =1;
          firststage[i*8+j][1]  =  value;
          work[i*8+j][1] = true;


                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][0] =1;
          firststage[i*8+j][8]  =  value;
          work[i*8+j][8] = true;


                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][2] =1;
          firststage[i*8+j][10]  =  value;
          work[i*8+j][10] = true;


                                        }
                                        if(holder[2][1] ==1)
                                        {
```

```
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][1] =1;
          firststage[i*8+j][17]  =  value;
          work[i*8+j][17] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][0] =1;
          firststage[i*8+j][16]  =  value;
          work[i*8+j][16] = true;


                                        }
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][2] =1;
          firststage[i*8+j][18]  =  value;
          work[i*8+j][18] = true;


                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][1] =1;
          firststage[i*8+j][9]  =  value;
          work[i*8+j][9] = true;


                                        }
                                }
                        }
                        if(j == 3)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][0] =1;
          firststage[i*8+j][0]  =  value;
          work[i*8+j][0] = true;


                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][3] =1;
          firststage[i*8+j][11]  =  value;
          work[i*8+j][11] = true;


                                        }
                                        if(holder[0][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][5] =1;
          firststage[i*8+j][5]  =  value;
          work[i*8+j][5] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][3] =1;
          firststage[i*8+j][3]  =  value;
          work[i*8+j][3] = true;


                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][0] =1;
          firststage[i*8+j][8]  =  value;
          work[i*8+j][8] = true;


                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][5] =1;
          firststage[i*8+j][13]  =  value;
          work[i*8+j][13] = true;


                                        }
                                        if(holder[2][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][3] =1;
          firststage[i*8+j][19]  =  value;
          work[i*8+j][19] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][0] =1;
          firststage[i*8+j][16]  =  value;
          work[i*8+j][16] = true;


                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][3] =1;
          firststage[i*8+j][11]  =  value;
          work[i*8+j][11] = true;


                                        }
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][5] =1;
          firststage[i*8+j][21]  =  value;
          work[i*8+j][21] = true;


                                        }
                                }
                        }
                        if(j == 4)
                        {
                                if(i==0)
                                {
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][2] =1;
          firststage[i*8+j][2]  =  value;
          work[i*8+j][2] = true;
```

```
                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][4] =1;
          firststage[i*8+j][12]  =  value;
          work[i*8+j][12] = true;


                                        }
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][7] =1;
          firststage[i*8+j][7]  =  value;
          work[i*8+j][7] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][4] =1;
          firststage[i*8+j][4]  =  value;
          work[i*8+j][4] = true;


                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][2] =1;
          firststage[i*8+j][10]  =  value;
          work[i*8+j][10] = true;


                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][7] =1;
          firststage[i*8+j][15]  =  value;
          work[i*8+j][15] = true;


                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][4] =1;
          firststage[i*8+j][20]  =  value;
          work[i*8+j][20] = true;


                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][2] =1;
          firststage[i*8+j][18]  =  value;
          work[i*8+j][18] = true;


                                        }

                                        if(holder[2][7] ==1)
                                        {

                                        holder[i][j] =1;
                                        holder[2][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[2][7] =1;
          firststage[i*8+j][23]  =  value;
          work[i*8+j][23] = true;



                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][4] =1;
          firststage[i*8+j][12]  =  value;
          work[i*8+j][12] = true;


                                        }
                                }
                        }
                        if(j == 6)
                        {
                                if(i==0)
                                {
                                        if(holder[0][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][5] =1;
          firststage[i*8+j][5]  =  value;
          work[i*8+j][5] = true;


                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][6] =1;
          firststage[i*8+j][14]  =  value;
          work[i*8+j][14] = true;
```

```
                                        }
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][7] =1;
          firststage[i*8+j][7]  =  value;
          work[i*8+j][7] = true;


                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[0][6] =1;
          firststage[i*8+j][6]  =  value;
          work[i*8+j][6] = true;


                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][5] =1;
          firststage[i*8+j][13]  =  value;
          work[i*8+j][13] = true;


                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewone=i;
          positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
          holder[i][j] =0;
                                        holder[1][7] =1;
          firststage[i*8+j][15]  =  value;
          work[i*8+j][15] = true;
```

```
                                        }
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[2][6] =1;
             firststage[i*8+j][22]  =  value;
             work[i*8+j][22] = true;


                                        }
                              }
                              if(i==2)
                              {
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[2][5] =1;
             firststage[i*8+j][21]  =  value;
             work[i*8+j][21] = true;


                                        }
                                        if(holder[2][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][7] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[2][7] =1;
             firststage[i*8+j][23]  =  value;
             work[i*8+j][23] = true;


                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
             levelnewone=i;
             positionnewone=j;computermovestagetwo(holder, value,
levelnewone, positionnewone);
             holder[i][j] =0;
                                        holder[1][6] =1;
```

```
          firststage[i*8+j][14]  =  value;
          work[i*8+j][14] = true;


                                        }
                                }
                        }
                }
        }
}

for(l=0; l<24; l++)
{
 for(int m=0; m<24; m++)
 {
  if(work[l][m] ==true)
  {
    //cout<<"source "<<m/8<<"   "<<m%8<<"      destination "<<l/8<<"
"<<l%8<<" value "<<firststage[l][m]<<endl;
    if(firststage[l][m] >= value)
    {
    value = firststage[l][m];
     destination = l;
     source =m;
    }
  }
 }
}
//cout<<source/8<<"  "<<source%8<<endl;
//cout<<destination/8<<"  "<<destination%8<<endl;
int levelnew=destination/8;
int positionnew=destination%8;
board[source/8][source%8] =0;
board[destination/8][destination%8] =1;
checkcomputerlines(board, levelnew, positionnew);
}
void playermove(int board[3][8])
{
int ipos ;
int ilev;
int levelnew;
int positionnew;
bool work = false;
do
{
cout<<"Which level is the piece you want to move on?"<<endl;
cin>> ilev;
cout<<"Which position is the piece you want to move on?"<<endl;
cin>> ipos;
cout<<"Which level would you like to move to"<<endl;
cin>> levelnew;
cout<<"Which position would you like to move to"<<endl;
cin>> positionnew;
if(board[ilev][ipos] == 2 && board[levelnew][positionnew] == 0)
{
 work = true;
 board[ilev][ipos]=0;
 board[levelnew][positionnew] = 2;
}
if(work == false && board[ilev][ipos] != 2)
{
cout<<"you don't have a piece there"<<endl;
}
if(work == false && board[levelnew][positionnew] != 0)
{
cout<<"There is a piece already there"<<endl;
}
}
while(work == false);
checkplayerlines(board, levelnew, positionnew);

}

void computerhop(int board[3][8])
{
int levelnew=0;
int positionnew=0;

for (int i=0; i<3; i++)
{
        for(int j=0; j<8; j++)
        {
                if(board[i][j] ==1)
                {
                        for (int m=0; m<3; m++)
                        {
                                for(int k=0; k<8; k++)
                                {
                                        if(board[m][k] == 0)
                                        {
                                        board[m][k] = 1;
                                        board[i][j] = 0;
          levelnew=m;
          positionnew= k;
          checkcomputerlines(board, levelnew, positionnew);
                                        return;
                                        }
                                }
                        }
                }
        }
}
}
void playerhop(int board[3][8])
{
int ipos ;
int ilev;
int levelnew;
int positionnew;
cout<<"Which level is the piece you want to move on?"<<endl;
cin>> ilev;
cout<<"Which position is the piece you want to move on?"<<endl;
cin>> ipos;
cout<<"Which level would you like to move to"<<endl;
```

```
cin>> levelnew;
cout<<"Which level would you like to move to"<<endl;
cin>> positionnew;

board[ilev][ipos]=0;
board[levelnew][positionnew] = 2;

checkplayerlines(board, levelnew, positionnew);
}

void computerlocationstageone(int board[3][8])
{
int level;
int position;
int value =0;
int firstlevel[3][8];
clearboard2(firstlevel);
for(level=0; level<3; level++)
{
 for(position=0; position<8; position++)
 {
  value = 0;
  if (board[level][position] == 0)
  {
  board[level][position]=1;
   computerlocationstagetwo(board, level, position, value);
  board[level][position]=0;
  firstlevel[level][position] = value;

  }
 }
}

computerlocation(board, firstlevel);
}



void computerlocationstagetwo(int board[3][8], int level, int position,
int & value)
{
bool lines = false;
int secondlevel[24];
   for (int z=0; z<24; z++)
   {
      secondlevel[z]=0;
   }
//check lines
checkcomputerlineslookingahead(board, level, position, lines);
if (lines == true)
{
computerlocationstagetwolines(board, level, position, value);
}
if (lines == false)
{
int i=0;
int j=0;
```

```
for (i=0; i<3; i++)
{
 for(j=0; j<8; j++)
 {
   if(board[i][j] ==0)
   {
    computerlocationstagethree(level, position, board, i, j, value);
    secondlevel[(i*8)+j] =value;
   }
 }
}
for (i=0; i<3; i++)
{
 for(j=0; j<8; j++)
 {
  if(board[i][j] ==0)
  {
     if(secondlevel[(i*8)+j] <=value)
     {
      value=secondlevel[(i*8)+j];
     }
     secondlevel[(i*8)+j]=0;
  }
 }
}
}
}
void computerlocationstagetwolines(int board[3][8], int level, int
position, int & value)
{
int i=0;
int j=0;
int secondlevel[24];
  for (int z=0; z<24; z++)
  {
     secondlevel[z]=0;
  }
for(int x=0; x<3; x++)
{
 for(int y=0; y<8; y++)
 {
  if(board[x][y] == 2)
  {
  board[x][y] =0;
  for (i=0; i<3; i++)
  {
   for(j=0; j<8; j++)
   {
     if(board[i][j] ==0)
     {
      computerlocationstagethree(level, position, board, i, j, value);
      secondlevel[(i*8)+j] =value;
     }
   }
  }
  for (i=0; i<3; i++)
  {
   for(j=0; j<8; j++)
   {
    if(board[i][j] ==0)
    {
       if(secondlevel[(i*8)+j] <=value)
       {
        value=secondlevel[(i*8)+j];
       }
       secondlevel[(i*8)+j]=0;
    }
   }
  }
  board[x][y] =2;
  }
 }
}
}
void computerlocationstagethree(int level, int position, int
board[3][8],int  i,int  j, int & value)
{
bool lines =false;
int thirdlevel[24];
  for (int z=0; z<24; z++)
  {
     thirdlevel[z]=0;
  }
int cvalue=0;
int ci=0;
int cj=0;
board[i][j]=2;
checkplayerlineslookingahead(board, level, position, lines);
if (lines == true)
{
computerlocationstagethreelines(level, position, board, i, j, value);
}
if (lines == false)
{
 for(ci=0; ci<3; ci++)
 {
  for(cj=0; cj<8; cj++)
  {
   if(board[ci][cj]==0)
   {
   analyzecomputerlocationstagefour(board, ci, cj, cvalue);
   thirdlevel[(ci*8)+cj]=cvalue;
   }

  }

 }

 for (int l=0; l<3; l++)
 {
  for(int m=0; m<8; m++)
  {
   if(thirdlevel[(l*8)+m] >=cvalue && board[l][m]  ==0)
   {
    cvalue =thirdlevel[(l*8)+m];
   }
  thirdlevel[(l*8)+m] =0;
  }
 }

}
value = cvalue;
board[i][j]=0;
}
void computerlocationstagethreelines(int level, int position, int
board[3][8],int  i,int  j, int & value)
{
int thirdlevel[24];
  for (int z=0; z<24; z++)
  {
     thirdlevel[z]=0;
  }
 int cvalue=0;
 int ci=0;
 int cj=0;
for(int x=0; x<3; x++)
{
 for(int y=0; y<8; y++)
 {
  if(board[x][y] == 1)
  {
  board[x][y] =0;
    for(ci=0; ci<3; ci++)
    {
     for(cj=0; cj<8; cj++)
     {
      if(board[ci][cj]==0)
      {
      analyzecomputerlocationstagefour(board, ci, cj, cvalue);
      thirdlevel[(ci*8)+cj]=cvalue;
      }

     }

    }

    for (int l=0; l<3; l++)
    {
     for(int m=0; m<8; m++)
     {
      if(thirdlevel[(l*8)+m] >=cvalue && board[l][m]  ==0)
      {
       cvalue =thirdlevel[(l*8)+m];
      }
     thirdlevel[(l*8)+m] =0;
     }
    }
  board[x][y] =1;
  }
 }
```

```
}
}
void analyzecomputerlocationstagefour(int board[3][8], int ci, int cj,
int & cvalue)
{

 board[ci][cj]=1;

  cvalue=0;
 compthreeinarow(board, cvalue);
 playerthreeinarow(board, cvalue);
 analyzecheckmoveability(board, cvalue);
 analyzecheckmoveabilityplayer(board, cvalue);
 analyzecomputerboardlocation(board, cvalue);
 analyzeplayerboardlocation(board, cvalue);
  board[ci][cj]=0;
}

void compthreeinarow(int board[3][8], int & cvalue)
{
 if(board[0][0] ==1 && board[0][1] ==1 && board[0][2]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]==1)
 {
cvalue= cvalue+ 100;
 }
 if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]==1)
 {
cvalue= cvalue+ 100;
 }

}
void playerthreeinarow(int board[3][8], int & cvalue)
{
 if(board[0][0] ==2 && board[0][1] ==2 && board[0][2]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][0] ==2 && board[0][3] ==2 && board[0][5]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][2] ==2 && board[0][4] ==2 && board[0][7]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][5] ==2 && board[0][6] ==2 && board[0][7]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[1][0] ==2 && board[1][1] ==2 && board[1][2]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[1][0] ==2 && board[1][3] ==2 && board[1][5]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[1][2] ==2 && board[1][4] ==2 && board[1][7]==2)
 {
```

```
cvalue= cvalue- 150;
 }
 if(board[1][5] ==2 && board[1][6] ==2 && board[1][7]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[2][0] ==2 && board[2][1] ==2 && board[2][2]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[2][0] ==2 && board[2][3] ==2 && board[2][5]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[2][2] ==2 && board[2][4] ==2 && board[2][7]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[2][5] ==2 && board[2][6] ==2 && board[2][7]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][0] ==2 && board[1][0] ==2 && board[2][0]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][2] ==2 && board[1][2] ==2 && board[2][2]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][5] ==2 && board[1][5] ==2 && board[2][5]==2)
 {
cvalue= cvalue- 150;
 }
 if(board[0][7] ==2 && board[1][7] ==2 && board[2][7]==2)
 {
cvalue= cvalue- 150;
 }

}

void checkcomputerlineslookingahead(int board[3][8], int levelnew, int
positionnew, bool & lines)
{
lines = false;
if (levelnew ==0)
{
        if(positionnew == 0)
        {
                if(board[0][0] == 1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
```

```
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[0][0] ==1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[0][0] ==1 && board[0][1] ==1 && board[0][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
```

```
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[0][3] ==1 && board[0][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[0][5] ==1 && board[0][6] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[0][4] ==1 && board[0][7]
==1)
                {
                lines= true;
                }
        }
```

```
}
if (levelnew ==1)
{
        if(positionnew == 0)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[1][0] ==1 && board[1][1] ==1 && board[1][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
```

```
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[1][0] ==1 && board[1][3] ==1 && board[1][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
                lines= true;
                }
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[1][5] ==1 && board[1][6] ==1 && board[1][7]
==1)
                {
```

```
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[1][2] ==1 && board[1][4] ==1 && board[1][7]
==1)
                {
                lines= true;
                }
        }
}
if (levelnew ==2)
{
        if(positionnew == 0)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][0] ==1 && board[1][0] ==1 && board[2][0]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][1] ==1 && board[1][1] ==1 && board[2][1]
==1)
                {
                lines= true;
                }
        }
        if(positionnew ==2)
        {
                if(board[2][0] ==1 && board[2][1] ==1 && board[2][2]
==1)
                {
                lines= true;
                }
                if(board[0][2] ==1 && board[1][2] ==1 && board[2][2]
==1)
                {
```

```
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] ==1 && board[1][3] ==1 && board[2][3]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] ==1 && board[1][4] ==1 && board[2][4]
==1)
                {
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][5] ==1 && board[1][5] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
                if(board[2][0] ==1 && board[2][3] ==1 && board[2][5]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] ==1 && board[1][6] ==1 && board[2][6]
==1)
                {
```

```
                lines= true;
                }
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[2][5] ==1 && board[2][6] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[0][7] ==1 && board[1][7] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
                if(board[2][2] ==1 && board[2][4] ==1 && board[2][7]
==1)
                {
                lines= true;
                }
        }
}
}

void checkplayerlineslookingahead(int board[3][8], int levelnew, int
positionnew, bool & lines)
{
lines = false;
if (levelnew ==0)
{
        if(positionnew == 0)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2]
==2)
```

```
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1] 
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[0][0] == 2 && board[0][1] ==2 && board[0][2] 
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2] 
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7] 
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3] 
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5] 
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4] 
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7] 
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7] 
==2)
```

```
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[0][3] ==2 && board[0][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[0][5] == 2 && board[0][6] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[0][4] ==2 && board[0][7]
==2)
                {
                lines= true;
                }
        }
}
if (levelnew ==1)
{
        if(positionnew == 0)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
```

```
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[1][0] == 2 && board[1][1] ==2 && board[1][2]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3]
==2)
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4]
==2)
```

```
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[1][0] == 2 && board[1][3] ==2 && board[1][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[1][5] == 2 && board[1][6] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[1][2] == 2 && board[1][4] ==2 && board[1][7]
==2)
                {
                lines= true;
                }
        }
}
if (levelnew ==2)
{

        if(positionnew == 0)
        {
                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][0] == 2 && board[1][0] ==2 && board[2][0]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 1)
        {

                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][1] == 2 && board[1][1] ==2 && board[2][1]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 2)
        {
                if(board[2][0] == 2 && board[2][1] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[0][2] == 2 && board[1][2] ==2 && board[2][2]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 3)
        {
                if(board[0][3] == 2 && board[1][3] ==2 && board[2][3]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 4)
        {
                if(board[0][4] == 2 && board[1][4] ==2 && board[2][4]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 5)
        {
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][5] == 2 && board[1][5] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
                if(board[2][0] == 2 && board[2][3] ==2 && board[2][5]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 6)
        {
                if(board[0][6] == 2 && board[1][6] ==2 && board[2][6]
==2)
                {
                lines= true;
                }
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
        if(positionnew == 7)
        {
                if(board[2][5] == 2 && board[2][6] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[0][7] == 2 && board[1][7] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
                if(board[2][2] == 2 && board[2][4] ==2 && board[2][7]
==2)
                {
                lines= true;
                }
        }
}
}

void computermovestagetwo(int holder[3][8], int & value, int
levelnewone, int positionnewone)
{
value =0;
bool lines = false;
int levelnewtwo=0;
int positionnewtwo=0;
int secondvalue =0;
int secondstage[24][24];
bool work[24][24];
        for (int l=0; l<24; l++)
        {
         for(int m=0; m<24;m++)
         {
         secondstage[l][m]=0;
         work[l][m] = false;
         }
        }
for (int i=0; i<3; i++)
{
        for(int j=0; j<8; j++)
        {
                if (holder[i][j] == 0)
                {
                        //corners
                        if(j == 0)
                        {
                                if(i==0)
                                {
                                        if(holder[0][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][1] =2;
          secondstage[i*8+j][1] = secondvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[1][0] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[1][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[1][0] =2;
          secondstage[i*8+j][8]  =  secondvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[0][3] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[0][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[0][3] =2;
          secondstage[i*8+j][3]  =  secondvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                 }
                                 if(i==1)
                                 {
                                         if(holder[0][0] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[0][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[0][0] =2;
          secondstage[i*8+j][0]  =  secondvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[1][1] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[1][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
```

```
          holder[i][j] =0;
                                        holder[1][1] =2;
          secondstage[i*8+j][9]  =  secondvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][3] =2;
          secondstage[i*8+j][11]  =  secondvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][0] =2;
          secondstage[i*8+j][16]  =  secondvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][1] =2;
          secondstage[i*8+j][17]  =  secondvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                       holder[2][3] =2;
          secondstage[i*8+j][19]  =  secondvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                       }
                                       if(holder[1][0] ==2)
                                       {
                                       holder[i][j] =2;
                                       holder[1][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                       holder[1][0] =2;
          secondstage[i*8+j][8]  =  secondvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                       }
                               }
                       }
                       if(j == 2)
                       {
                               if(i==0)
                               {
                                       if(holder[0][1] ==2)
                                       {
                                       holder[i][j] =2;
                                       holder[0][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                       holder[0][1] =2;
          secondstage[i*8+j][1]  =  secondvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                       }
                                       if(holder[1][2] ==2)
                                       {
                                       holder[i][j] =2;
                                       holder[1][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                       holder[1][2] =2;
          secondstage[i*8+j][10]  =  secondvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                       }
```

```
                                        if(holder[0][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][4] =2;
          secondstage[i*8+j][4]  =  secondvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][2] =2;
          secondstage[i*8+j][2]  =  secondvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][1] =2;
          secondstage[i*8+j][9]  =  secondvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][4] =2;
          secondstage[i*8+j][12]  =  secondvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);
```

```
                                        }
                                        if(holder[2][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][2] =2;
          secondstage[i*8+j][18]  =  secondvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][1] =2;
          secondstage[i*8+j][17]  =  secondvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][4] =2;
          secondstage[i*8+j][20]  =  secondvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][2] =2;
          secondstage[i*8+j][10]  =  secondvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        if(j == 5)
                        {
                                if(i==0)
                                {
                                        if(holder[0][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][3] =2;
          secondstage[i*8+j][3]  =  secondvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][5] =2;
          secondstage[i*8+j][13]  =  secondvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[0][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][6] =2;
          secondstage[i*8+j][6]  =  secondvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][5] ==2)
                                        {
```

```
                                        holder[i][j] =2;
                                        holder[0][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][5] =2;
          secondstage[i*8+j][5]  =  secondvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][3] =2;
          secondstage[i*8+j][11]  =  secondvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][6] =2;
          secondstage[i*8+j][14]  =  secondvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][5] =2;
          secondstage[i*8+j][21]  =  secondvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][3] =2;
          secondstage[i*8+j][19]  =  secondvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][5] =2;
          secondstage[i*8+j][13]  =  secondvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][6] =2;
          secondstage[i*8+j][22]  =  secondvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        if(j == 7)
                        {
                                if(i==0)
                                {
                                        if(holder[0][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][6] =2;
          secondstage[i*8+j][6]  =  secondvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[1][7] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[1][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[1][7] =2;
          secondstage[i*8+j][15]  =  secondvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[0][4] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[0][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[0][4] =2;
          secondstage[i*8+j][4]  =  secondvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                 }
                                 if(i==1)
                                 {
                                         if(holder[0][7] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[0][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                         holder[0][7] =2;
          secondstage[i*8+j][7]  =  secondvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                         }
                                         if(holder[1][6] ==2)
                                         {
                                         holder[i][j] =2;
                                         holder[1][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
```

```
          holder[i][j] =0;
                                        holder[1][6] =2;
          secondstage[i*8+j][14]  =  secondvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][4] =2;
          secondstage[i*8+j][12]  =  secondvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][7] =2;
          secondstage[i*8+j][23]  =  secondvalue;
          work[i*8+j][23] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][6] =2;
          secondstage[i*8+j][22]  =  secondvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][7] =2;
          secondstage[i*8+j][15]  =  secondvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][4] =2;
          secondstage[i*8+j][20]  =  secondvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        //sides
                        if(j == 1)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][0] =2;
          secondstage[i*8+j][0]  =  secondvalue;
          work[i*8+j][0] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][1] =2;
          secondstage[i*8+j][9]  =  secondvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);
```

```
                                        }
                                        if(holder[0][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][2] =2;
          secondstage[i*8+j][2]  =  secondvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][1] =2;
          secondstage[i*8+j][1]  =  secondvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][0] =2;
          secondstage[i*8+j][8]  =  secondvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][2] =2;
          secondstage[i*8+j][10]  =  secondvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][1] =2;
          secondstage[i*8+j][17]  =  secondvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][0] =2;
          secondstage[i*8+j][16]  =  secondvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][2] =2;
          secondstage[i*8+j][18]  =  secondvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][1] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][1] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][1] =2;
          secondstage[i*8+j][9]  =  secondvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        if(j == 3)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][0] =2;
          secondstage[i*8+j][0]  =  secondvalue;
          work[i*8+j][0] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][3] =2;
          secondstage[i*8+j][11]  =  secondvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[0][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][5] =2;
          secondstage[i*8+j][5]  =  secondvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][3] =2;
          secondstage[i*8+j][3]  =  secondvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][0] =2;
          secondstage[i*8+j][8]  =  secondvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][5] =2;
          secondstage[i*8+j][13]  =  secondvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][3] =2;
          secondstage[i*8+j][19]  =  secondvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][0] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][0] =2;
          secondstage[i*8+j][16]  =  secondvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][3] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][3] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][3] =2;
          secondstage[i*8+j][11]  =  secondvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][5] =2;
          secondstage[i*8+j][21]  =  secondvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        if(j == 4)
                        {
                                if(i==0)
                                {
                                        if(holder[0][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
```

```
                                        holder[0][2] =2;
          secondstage[i*8+j][2]  =  secondvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][4] =2;
          secondstage[i*8+j][12]  =  secondvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[0][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][7] =2;
          secondstage[i*8+j][7]  =  secondvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][4] =2;
          secondstage[i*8+j][4]  =  secondvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][2] =2;
          secondstage[i*8+j][10]  =  secondvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][7] =2;
          secondstage[i*8+j][15]  =  secondvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][4] =2;
          secondstage[i*8+j][20]  =  secondvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][2] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][2] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][2] =2;
          secondstage[i*8+j][18]  =  secondvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }

                                        if(holder[2][7] ==2)
                                        {
```

```
                                        holder[i][j] =2;
                                        holder[2][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][7] =2;
          secondstage[i*8+j][23]  =  secondvalue;
          work[i*8+j][23] = true;

          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][4] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][4] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][4] =2;
          secondstage[i*8+j][12]  =  secondvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                        if(j == 6)
                        {
                                if(i==0)
                                {
                                        if(holder[0][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][5] =2;
          secondstage[i*8+j][5]   =  secondvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][6] =2;
          secondstage[i*8+j][14]  =  secondvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[0][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][7] =2;
          secondstage[i*8+j][7]  =  secondvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[0][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[0][6] =2;
          secondstage[i*8+j][6]  =  secondvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][5] =2;
          secondstage[i*8+j][13]  =  secondvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
```

```
          holder[i][j] =0;
                                        holder[1][7] =2;
          secondstage[i*8+j][15]  =  secondvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][6] =2;
          secondstage[i*8+j][22]  =  secondvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][5] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][5] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][5] =2;
          secondstage[i*8+j][21]  =  secondvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[2][7] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[2][7] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[2][7] =2;
          secondstage[i*8+j][23]  =  secondvalue;
          work[i*8+j][23] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                        if(holder[1][6] ==2)
                                        {
                                        holder[i][j] =2;
                                        holder[1][6] =0;
          levelnewtwo=i;
          positionnewtwo=j;computermovestagethree(holder, secondvalue,
levelnewone, positionnewone, levelnewtwo, positionnewtwo);
          holder[i][j] =0;
                                        holder[1][6] =2;
          secondstage[i*8+j][14]  =  secondvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewtwo, positionnewtwo);

                                        }
                                }
                        }
                }
        }
}
int temp = 1000;
for(l=0; l<24; l++)
{
 for(int m=0; m<24; m++)
 {

  if(secondstage[l][m] <= temp && work [l][m]==true)
  {
   temp = secondstage[l][m];

  }
 }
}
value =temp;

}
void computermovestagethree(int holder[3][8], int & secondvalue, int
levelnewone, int positionnewone, int levelnewtwo, int positionnewtwo)
{
secondvalue =0;
int thirdvalue =0;
int levelnewthree=0;
int positionnewthree=0;
int thirdstage[24][24];
bool work[24][24];
        for (int l=0; l<24; l++)
        {
         for(int m=0; m<24;m++)
         {
         thirdstage[l][m]=0;
         work[l][m] = false;
         }
        }
for (int i=0; i<3; i++)
{
        for(int j=0; j<8; j++)
        {
                if (holder[i][j] == 0)
                {
                        //corners
                        if(j == 0)
                        {
                                if(i==0)
                                {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][1] =1;
          thirdstage[i*8+j][1] = thirdvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][0] =1;
          thirdstage[i*8+j][8]  =  thirdvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][3] =1;
          thirdstage[i*8+j][3]  =  thirdvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][0] =1;
          thirdstage[i*8+j][0]  =  thirdvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][1] =1;
          thirdstage[i*8+j][9]  =  thirdvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][3] =1;
          thirdstage[i*8+j][11]  =  thirdvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][0] =1;
          thirdstage[i*8+j][16]  =  thirdvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);
```

```
                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][1] =1;
          thirdstage[i*8+j][17]  =  thirdvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][3] =1;
          thirdstage[i*8+j][19]  =  thirdvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][0] =1;
          thirdstage[i*8+j][8]  =  thirdvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                        }
                        if(j == 2)
                        {
                                if(i==0)
                                {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][1] =1;
          thirdstage[i*8+j][1]  =  thirdvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][2] =1;
          thirdstage[i*8+j][10]  =  thirdvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][4] =1;
          thirdstage[i*8+j][4]  =  thirdvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][2] =1;
          thirdstage[i*8+j][2]  =  thirdvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][1] =1;
          thirdstage[i*8+j][9]  =  thirdvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][4] =1;
          thirdstage[i*8+j][12]  =  thirdvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][2] =1;
          thirdstage[i*8+j][18]  =  thirdvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][1] =1;
          thirdstage[i*8+j][17]  =  thirdvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][4] =1;
          thirdstage[i*8+j][20]  =  thirdvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][2] =1;
          thirdstage[i*8+j][10]  =  thirdvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                }
                        }
                        if(j == 5)
                        {
                                if(i==0)
                                {
                                        if(holder[0][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][3] =1;
          thirdstage[i*8+j][3]  =  thirdvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][5] =1;
          thirdstage[i*8+j][13]  =  thirdvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][6] =1;
          thirdstage[i*8+j][6]  =  thirdvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][5] ==1)
                                        {
```

```
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][5] =1;
          thirdstage[i*8+j][5]  =  thirdvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][3] =1;
          thirdstage[i*8+j][11]  =  thirdvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][6] =1;
          thirdstage[i*8+j][14]  =  thirdvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][5] =1;
          thirdstage[i*8+j][21]  =  thirdvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][3] =1;
          thirdstage[i*8+j][19]  =  thirdvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][5] =1;
          thirdstage[i*8+j][13]  =  thirdvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][6] =1;
          thirdstage[i*8+j][22]  =  thirdvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
```

```
                        }
                        if(j == 7)
                        {
                                if(i==0)
                                {
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][6] =1;
          thirdstage[i*8+j][6]  =  thirdvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][7] =1;
          thirdstage[i*8+j][15]  =  thirdvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][4] =1;
          thirdstage[i*8+j][4]  =  thirdvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][7] =1;
          thirdstage[i*8+j][7]  =  thirdvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][6] =1;
          thirdstage[i*8+j][14]  =  thirdvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][4] =1;
          thirdstage[i*8+j][12]  =  thirdvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][7] =1;
          thirdstage[i*8+j][23]  =  thirdvalue;
          work[i*8+j][23] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][6] =1;
          thirdstage[i*8+j][22]  =  thirdvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][7] =1;
          thirdstage[i*8+j][15]  =  thirdvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][4] =1;
          thirdstage[i*8+j][20]  =  thirdvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                        }
                        //sides
                        if(j == 1)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][0] =1;
          thirdstage[i*8+j][0]  =  thirdvalue;
          work[i*8+j][0] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][1] =1;
          thirdstage[i*8+j][9]  =  thirdvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][2] =1;
          thirdstage[i*8+j][2]  =  thirdvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][1] =1;
          thirdstage[i*8+j][1]  =  thirdvalue;
          work[i*8+j][1] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][0] =1;
          thirdstage[i*8+j][8]  =  thirdvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][2] =1;
          thirdstage[i*8+j][10]  =  thirdvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
```

```
          holder[i][j] =0;
                                        holder[2][1] =1;
          thirdstage[i*8+j][17]  =  thirdvalue;
          work[i*8+j][17] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][0] =1;
          thirdstage[i*8+j][16]  =  thirdvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][2] =1;
          thirdstage[i*8+j][18]  =  thirdvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][1] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][1] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][1] =1;
          thirdstage[i*8+j][9]  =  thirdvalue;
          work[i*8+j][9] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);
```

```
                                        }
                                }
                        }
                        if(j == 3)
                        {
                                if(i==0)
                                {
                                        if(holder[0][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][0] =1;
          thirdstage[i*8+j][0]  =  thirdvalue;
          work[i*8+j][0] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][3] =1;
          thirdstage[i*8+j][11]  =  thirdvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][5] =1;
          thirdstage[i*8+j][5]  =  thirdvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
```

```
                                 if(i==1)
                                 {
                                         if(holder[0][3] ==1)
                                         {
                                         holder[i][j] =1;
                                         holder[0][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                         holder[0][3] =1;
          thirdstage[i*8+j][3]  =  thirdvalue;
          work[i*8+j][3] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                         }
                                         if(holder[1][0] ==1)
                                         {
                                         holder[i][j] =1;
                                         holder[1][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                         holder[1][0] =1;
          thirdstage[i*8+j][8]  =  thirdvalue;
          work[i*8+j][8] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                         }
                                         if(holder[1][5] ==1)
                                         {
                                         holder[i][j] =1;
                                         holder[1][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                         holder[1][5] =1;
          thirdstage[i*8+j][13]  =  thirdvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                         }
                                         if(holder[2][3] ==1)
                                         {
                                         holder[i][j] =1;
                                         holder[2][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][3] =1;
          thirdstage[i*8+j][19]  =  thirdvalue;
          work[i*8+j][19] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][0] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][0] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][0] =1;
          thirdstage[i*8+j][16]  =  thirdvalue;
          work[i*8+j][16] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][3] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][3] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][3] =1;
          thirdstage[i*8+j][11]  =  thirdvalue;
          work[i*8+j][11] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][5] =1;
          thirdstage[i*8+j][21]  =  thirdvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                }
                        }
                        if(j == 4)
                        {
                                if(i==0)
                                {
                                        if(holder[0][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][2] =1;
          thirdstage[i*8+j][2]  =  thirdvalue;
          work[i*8+j][2] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][4] =1;
          thirdstage[i*8+j][12]  =  thirdvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][7] =1;
          thirdstage[i*8+j][7]  =  thirdvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);
```

```
                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][4] =1;
          thirdstage[i*8+j][4]  =  thirdvalue;
          work[i*8+j][4] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][2] =1;
          thirdstage[i*8+j][10]  =  thirdvalue;
          work[i*8+j][10] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][7] =1;
          thirdstage[i*8+j][15]  =  thirdvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[2][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][4] =1;
          thirdstage[i*8+j][20]  =  thirdvalue;
          work[i*8+j][20] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][2] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][2] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][2] =1;
          thirdstage[i*8+j][18]  =  thirdvalue;
          work[i*8+j][18] = true;
          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }

                                        if(holder[2][7] ==1)
                                        {

                                        holder[i][j] =1;
                                        holder[2][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder, 
levelnewone, positionnewone, levelnewtwo, positionnewtwo, 
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][7] =1;
          thirdstage[i*8+j][23]  =  thirdvalue;
          work[i*8+j][23] = true;

          //checkcomputerlines(holder, levelnewthree, 
positionnewthree);

                                        }
                                        if(holder[1][4] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][4] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][4] =1;
          thirdstage[i*8+j][12]  =  thirdvalue;
          work[i*8+j][12] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                        }
                        if(j == 6)
                        {
                                if(i==0)
                                {
                                        if(holder[0][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][5] =1;
          thirdstage[i*8+j][5]  =  thirdvalue;
          work[i*8+j][5] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][6] =1;
          thirdstage[i*8+j][14]  =  thirdvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[0][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
```

```
          holder[i][j] =0;
                                        holder[0][7] =1;
          thirdstage[i*8+j][7]  =  thirdvalue;
          work[i*8+j][7] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==1)
                                {
                                        if(holder[0][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[0][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[0][6] =1;
          thirdstage[i*8+j][6]  =  thirdvalue;
          work[i*8+j][6] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][5] =1;
          thirdstage[i*8+j][13]  =  thirdvalue;
          work[i*8+j][13] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][7] =1;
          thirdstage[i*8+j][15]  =  thirdvalue;
          work[i*8+j][15] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);
```

```
                                        }
                                        if(holder[2][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][6] =1;
          thirdstage[i*8+j][22]  =  thirdvalue;
          work[i*8+j][22] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                                if(i==2)
                                {
                                        if(holder[2][5] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][5] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][5] =1;
          thirdstage[i*8+j][21]  =  thirdvalue;
          work[i*8+j][21] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[2][7] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[2][7] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[2][7] =1;
          thirdstage[i*8+j][23]  =  thirdvalue;
          work[i*8+j][23] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                        if(holder[1][6] ==1)
                                        {
                                        holder[i][j] =1;
                                        holder[1][6] =0;
          levelnewthree=i;
          positionnewthree=j;analyzecomputermovestagefour(holder,
levelnewone, positionnewone, levelnewtwo, positionnewtwo,
levelnewthree, positionnewthree, thirdvalue);
          holder[i][j] =0;
                                        holder[1][6] =1;
          thirdstage[i*8+j][14]  =  thirdvalue;
          work[i*8+j][14] = true;
          //checkcomputerlines(holder, levelnewthree,
positionnewthree);

                                        }
                                }
                        }
                }
        }
}

for(l=0; l<24; l++)
{
 for(int m=0; m<24; m++)
 {

  if(thirdstage[l][m] >= thirdvalue && work[l][m] == true)
  {
   thirdvalue = thirdstage[l][m];
   secondvalue = thirdstage[l][m];
  }
 }
}


}

void analyzecomputermovestagefour(int holder[3][8],int levelnewone, int
positionnewone, int levelnewtwo, int positionnewtwo, int levelnewthree,
int positionnewthree, int & thirdvalue)
{
 thirdvalue=0;
 analyzecheckcomputerlines(holder, levelnewone, positionnewone,
levelnewthree, positionnewthree, thirdvalue);
 analyzecheckplayerlines(holder, levelnewtwo, positionnewtwo,
thirdvalue);
analyzecheckmoveability(holder, thirdvalue);
analyzecheckmoveabilityplayer(holder, thirdvalue);
 analyzecomputerboardlocation(holder, thirdvalue);
 analyzeplayerboardlocation(holder, thirdvalue);
}

void analyzecheckcomputerlines(int holder[3][8],int levelnewone, int
positionnewone, int levelnewthree, int positionnewthree, int &
thirdvalue)
{

if (levelnewone ==0 || levelnewthree ==0)
{
        if(positionnewone == 0 || positionnewthree  == 0)
        {
                if(holder[0][0] == 1 && holder[0][1] ==1 &&
holder[0][2] ==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][0] ==1 && holder[1][0] ==1 && holder[2][0]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][0] ==1 && holder[0][3] ==1 && holder[0][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 1 || positionnewthree  == 1)
        {
                if(holder[0][0] ==1 && holder[0][1] ==1 && holder[0][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][1] ==1 && holder[1][1] ==1 && holder[2][1]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone ==2 || positionnewthree ==2)
        {
                if(holder[0][0] ==1 && holder[0][1] ==1 && holder[0][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[1][2] ==1 && holder[2][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[0][4] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 3 || positionnewthree  == 3)
        {
                if(holder[0][3] ==1 && holder[1][3] ==1 && holder[2][3]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][0] ==1 && holder[0][3] ==1 && holder[0][5]
==1)
```

```
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 4 || positionnewthree  == 4)
        {
                if(holder[0][4] ==1 && holder[1][4] ==1 && holder[2][4]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[0][4] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 5 || positionnewthree  == 5)
        {
                if(holder[0][5] ==1 && holder[0][6] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][5] ==1 && holder[1][5] ==1 && holder[2][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][0] ==1 && holder[0][3] ==1 && holder[0][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 6 || positionnewthree  == 6)
        {
                if(holder[0][6] ==1 && holder[1][6] ==1 && holder[2][6]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][5] ==1 && holder[0][6] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 7 || positionnewthree  == 7)
        {
                if(holder[0][5] ==1 && holder[0][6] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][7] ==1 && holder[1][7] ==1 && holder[2][7]
==1)
```

```
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[0][4] ==1 && holder[0][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
}
if (levelnewone ==1 || levelnewthree ==1)
{
        if(positionnewone == 0 || positionnewthree  == 0)
        {
                if(holder[1][0] ==1 && holder[1][1] ==1 && holder[1][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][0] ==1 && holder[1][0] ==1 && holder[2][0]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][0] ==1 && holder[1][3] ==1 && holder[1][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 1 || positionnewthree  == 1)
        {
                if(holder[1][0] ==1 && holder[1][1] ==1 && holder[1][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][1] ==1 && holder[1][1] ==1 && holder[2][1]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone ==2 || positionnewthree ==2)
        {
                if(holder[1][0] ==1 && holder[1][1] ==1 && holder[1][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[1][2] ==1 && holder[2][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][2] ==1 && holder[1][4] ==1 && holder[1][7]
==1)
```

```
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 3 || positionnewthree  == 3)
        {
                if(holder[0][3] ==1 && holder[1][3] ==1 && holder[2][3]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][0] ==1 && holder[1][3] ==1 && holder[1][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 4 || positionnewthree  == 4)
        {
                if(holder[0][4] ==1 && holder[1][4] ==1 && holder[2][4]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][2] ==1 && holder[1][4] ==1 && holder[1][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 5 || positionnewthree  == 5)
        {
                if(holder[1][5] ==1 && holder[1][6] ==1 && holder[1][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][5] ==1 && holder[1][5] ==1 && holder[2][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][0] ==1 && holder[1][3] ==1 && holder[1][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 6 || positionnewthree  == 6)
        {
                if(holder[0][6] ==1 && holder[1][6] ==1 && holder[2][6]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[1][5] ==1 && holder[1][6] ==1 && holder[1][7]
==1)
```

```
				{
				thirdvalue = thirdvalue + 100; return;
				}
		}
		if(positionnewone == 7 || positionnewthree  == 7)
		{
				if(holder[1][5] ==1 && holder[1][6] ==1 && holder[1][7]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
				if(holder[0][7] ==1 && holder[1][7] ==1 && holder[2][7]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
				if(holder[1][2] ==1 && holder[1][4] ==1 && holder[1][7]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
		}
}
if (levelnewone ==2 || levelnewthree ==2)
{
		if(positionnewone == 0 || positionnewthree  == 0)
		{
				if(holder[2][0] ==1 && holder[2][1] ==1 && holder[2][2]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
				if(holder[0][0] ==1 && holder[1][0] ==1 && holder[2][0]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
				if(holder[2][0] ==1 && holder[2][3] ==1 && holder[2][5]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
		}
		if(positionnewone == 1 || positionnewthree  == 1)
		{
				if(holder[2][0] ==1 && holder[2][1] ==1 && holder[2][2]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
				if(holder[0][1] ==1 && holder[1][1] ==1 && holder[2][1]
==1)
				{
				thirdvalue = thirdvalue + 100; return;
				}
		}
		if(positionnewone ==2 || positionnewthree ==2)
```

```
        {
                if(holder[2][0] ==1 && holder[2][1] ==1 && holder[2][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][2] ==1 && holder[1][2] ==1 && holder[2][2]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][2] ==1 && holder[2][4] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 3 || positionnewthree  == 3)
        {
                if(holder[0][3] ==1 && holder[1][3] ==1 && holder[2][3]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][0] ==1 && holder[2][3] ==1 && holder[2][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 4 || positionnewthree  == 4)
        {
                if(holder[0][4] ==1 && holder[1][4] ==1 && holder[2][4]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][2] ==1 && holder[2][4] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 5 || positionnewthree  == 5)
        {
                if(holder[2][5] ==1 && holder[2][6] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][5] ==1 && holder[1][5] ==1 && holder[2][5]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][0] ==1 && holder[2][3] ==1 && holder[2][5]
==1)
```

```
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 6 || positionnewthree  == 6)
        {
                if(holder[0][6] ==1 && holder[1][6] ==1 && holder[2][6]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][5] ==1 && holder[2][6] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
        if(positionnewone == 7 || positionnewthree  == 7)
        {
                if(holder[2][5] ==1 && holder[2][6] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[0][7] ==1 && holder[1][7] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
                if(holder[2][2] ==1 && holder[2][4] ==1 && holder[2][7]
==1)
                {
                thirdvalue = thirdvalue + 100; return;
                }
        }
}

}


void analyzecheckplayerlines(int holder[3][8], int levelnewtwo, int
positionnewtwo, int & thirdvalue)
{

if (levelnewtwo ==0)
{
        if(positionnewtwo == 0)
        {
                if(holder[0][0] == 1 && holder[0][1] ==2 &&
holder[0][2] ==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[1][0] ==2 && holder[2][0]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[0][3] ==2 && holder[0][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 1)
        {
                if(holder[0][0] ==2 && holder[0][1] ==2 && holder[0][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][1] ==2 && holder[1][1] ==2 && holder[2][1]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo ==2)
        {
                if(holder[0][0] ==2 && holder[0][1] ==2 && holder[0][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[1][2] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[0][4] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 3)
        {
                if(holder[0][3] ==2 && holder[1][3] ==2 && holder[2][3]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[0][3] ==2 && holder[0][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 4)
        {
                if(holder[0][4] ==2 && holder[1][4] ==2 && holder[2][4]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[0][4] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 5)
        {
                if(holder[0][5] ==2 && holder[0][6] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][5] ==2 && holder[1][5] ==2 && holder[2][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[0][3] ==2 && holder[0][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 6)
        {
                if(holder[0][6] ==2 && holder[1][6] ==2 && holder[2][6]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][5] ==2 && holder[0][6] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 7)
        {
                if(holder[0][5] ==2 && holder[0][6] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][7] ==2 && holder[1][7] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[0][4] ==2 && holder[0][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
}
```

```
if (levelnewtwo ==1)
{
        if(positionnewtwo == 0)
        {
                if(holder[1][0] ==2 && holder[1][1] ==2 && holder[1][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[1][0] ==2 && holder[2][0]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[1][0] ==2 && holder[1][3] ==2 && holder[1][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 1)
        {
                if(holder[1][0] ==2 && holder[1][1] ==2 && holder[1][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][1] ==2 && holder[1][1] ==2 && holder[2][1]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo ==2)
        {
                if(holder[1][0] ==2 && holder[1][1] ==2 && holder[1][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[1][2] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[1][2] ==2 && holder[1][4] ==2 && holder[1][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 3)
        {
                if(holder[0][3] ==2 && holder[1][3] ==2 && holder[2][3]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
```

```
                 }
                 if(holder[1][0] ==2 && holder[1][3] ==2 && holder[1][5]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
        }
        if(positionnewtwo == 4)
        {
                 if(holder[0][4] ==2 && holder[1][4] ==2 && holder[2][4]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
                 if(holder[1][2] ==2 && holder[1][4] ==2 && holder[1][7]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
        }
        if(positionnewtwo == 5)
        {
                 if(holder[1][5] ==2 && holder[1][6] ==2 && holder[1][7]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
                 if(holder[0][5] ==2 && holder[1][5] ==2 && holder[2][5]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
                 if(holder[1][0] ==2 && holder[1][3] ==2 && holder[1][5]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
        }
        if(positionnewtwo == 6)
        {
                 if(holder[0][6] ==2 && holder[1][6] ==2 && holder[2][6]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
                 if(holder[1][5] ==2 && holder[1][6] ==2 && holder[1][7]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
                 }
        }
        if(positionnewtwo == 7)
        {
                 if(holder[1][5] ==2 && holder[1][6] ==2 && holder[1][7]
==2)
                 {
                 thirdvalue = thirdvalue - 100; return;
```

```
                }
                if(holder[0][7] ==2 && holder[1][7] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[1][2] ==2 && holder[1][4] ==2 && holder[1][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
}
if (levelnewtwo ==2)
{
        if(positionnewtwo == 0)
        {
                if(holder[2][0] ==2 && holder[2][1] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][0] ==2 && holder[1][0] ==2 && holder[2][0]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][0] ==2 && holder[2][3] ==2 && holder[2][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 1)
        {
                if(holder[2][0] ==2 && holder[2][1] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][1] ==2 && holder[1][1] ==2 && holder[2][1]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo ==2)
        {
                if(holder[2][0] ==2 && holder[2][1] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][2] ==2 && holder[1][2] ==2 && holder[2][2]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][2] ==2 && holder[2][4] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 3)
        {
                if(holder[0][3] ==2 && holder[1][3] ==2 && holder[2][3]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][0] ==2 && holder[2][3] ==2 && holder[2][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 4)
        {
                if(holder[0][4] ==2 && holder[1][4] ==2 && holder[2][4]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][2] ==2 && holder[2][4] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 5)
        {
                if(holder[2][5] ==2 && holder[2][6] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][5] ==2 && holder[1][5] ==2 && holder[2][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][0] ==2 && holder[2][3] ==2 && holder[2][5]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 6)
        {
                if(holder[0][6] ==2 && holder[1][6] ==2 && holder[2][6]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][5] ==2 && holder[2][6] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
        if(positionnewtwo == 7)
        {
                if(holder[2][5] ==2 && holder[2][6] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[0][7] ==2 && holder[1][7] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
                if(holder[2][2] ==2 && holder[2][4] ==2 && holder[2][7]
==2)
                {
                thirdvalue = thirdvalue - 100; return;
                }
        }
}

}

void analyzecheckmoveability(int holder[3][8], int & thirdvalue)
{
bool work[23][23];
for(int k=0; k<23; k++)
{
 for(int l=0; l<23; l++)
 {
  work[k][l] =false;
 }
}
  for (int i=0; i<3; i++)
  {
          for(int j=0; j<8; j++)
          {
                  if (holder[i][j] == 0)
                  {
                          //corners
                          if(j == 0)
                          {
                                  if(i==0)
                                  {
                                          if(holder[0][1] ==1)
                                          {
```

```
work[i*8+j][1] = true;

                              }
                              if(holder[1][0] ==1)
                              {

work[i*8+j][8] = true;

                              }
                              if(holder[0][3] ==1)
                              {

work[i*8+j][3] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][0] ==1)
                              {

work[i*8+j][9] = true;

                              }
                              if(holder[1][1] ==1)
                              {
```

```
work[i*8+j][9] = true;


                              }
                              if(holder[1][3] ==1)
                              {




work[i*8+j][11] = true;


                              }
                              if(holder[2][0] ==1)
                              {




work[i*8+j][16] = true;


                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][1] ==1)
                              {




work[i*8+j][17] = true;


                              }
                              if(holder[2][3] ==1)
                              {




work[i*8+j][19] = true;
```

```
                              }
                              if(holder[1][0] ==1)
                              {

work[i*8+j][18] = true;

                              }
                      }
              }
              if(j == 2)
              {
                      if(i==0)
                      {
                              if(holder[0][1] ==1)
                              {

work[i*8+j][1] = true;

                              }
                              if(holder[1][2] ==1)
                              {

work[i*8+j][10] = true;

                              }
                              if(holder[0][4] ==1)
                              {

work[i*8+j][4] = true;
```

```
                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][2] ==1)
                              {

work[i*8+j][2] = true;

                              }
                              if(holder[1][1] ==1)
                              {

work[i*8+j][9] = true;

                              }
                              if(holder[1][4] ==1)
                              {

work[i*8+j][12] = true;

                              }
                              if(holder[2][2] ==1)
                              {

work[i*8+j][18] = true;
```

```
                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][1] ==1)
                              {



work[i*8+j][17] = true;

                              }
                              if(holder[2][4] ==1)
                              {



work[i*8+j][20] = true;

                              }
                              if(holder[1][2] ==1)
                              {



work[i*8+j][10] = true;

                              }
                      }
              }
              if(j == 5)
              {
                      if(i==0)
                      {
                              if(holder[0][3] ==1)
                              {
```

```
work[i*8+j][3] = true;


                              }
                              if(holder[1][5] ==1)
                              {







work[i*8+j][13] = true;


                              }
                              if(holder[0][6] ==1)
                              {







work[i*8+j][6] = true;


                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][5] ==1)
                              {







work[i*8+j][5] = true;


                              }
                              if(holder[1][3] ==1)
                              {







work[i*8+j][11] = true;
```

```
                              }
                              if(holder[1][6] ==1)
                              {

work[i*8+j][14] = true;

                              }
                              if(holder[2][5] ==1)
                              {

work[i*8+j][21] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][3] ==1)
                              {

work[i*8+j][19] = true;

                              }
                              if(holder[1][5] ==1)
                              {

work[i*8+j][13] = true;
```

```
                              }
                              if(holder[2][6] ==1)
                              {



work[i*8+j][22] = true;


                              }
                      }
              }
              if(j == 7)
              {
                      if(i==0)
                      {
                              if(holder[0][6] ==1)
                              {



work[i*8+j][6] = true;


                              }
                              if(holder[1][7] ==1)
                              {



work[i*8+j][15] = true;


                              }
                              if(holder[0][4] ==1)
                              {



work[i*8+j][4] = true;
```

```
                               }
                       }
                       if(i==1)
                       {
                               if(holder[0][7] ==1)
                               {

work[i*8+j][7] = true;

                               }
                               if(holder[1][6] ==1)
                               {

work[i*8+j][14] = true;

                               }
                               if(holder[1][4] ==1)
                               {

work[i*8+j][12] = true;

                               }
                               if(holder[2][7] ==1)
                               {

work[i*8+j][23] = true;

                               }
```

```
                      }
                      if(i==2)
                      {
                              if(holder[2][6] ==1)
                              {



work[i*8+j][22] = true;


                              }
                              if(holder[1][7] ==1)
                              {



work[i*8+j][15] = true;


                              }
                              if(holder[2][4] ==1)
                              {



work[i*8+j][20] = true;


                              }
                      }
              }
              //sides
              if(j == 1)
              {
                      if(i==0)
                      {
                              if(holder[0][0] ==1)
                              {
```

```
work[i*8+j][0] = true;

                              }
                              if(holder[1][1] ==1)
                              {



work[i*8+j][9] = true;

                              }
                              if(holder[0][2] ==1)
                              {



work[i*8+j][2] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][1] ==1)
                              {



work[i*8+j][1] = true;

                              }
                              if(holder[1][0] ==1)
                              {



work[i*8+j][8] = true;
```

```
                              }
                              if(holder[1][2] ==1)
                              {

work[i*8+j][10] = true;

                              }
                              if(holder[2][1] ==1)
                              {

work[i*8+j][17] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][0] ==1)
                              {

work[i*8+j][16] = true;

                              }
                              if(holder[2][2] ==1)
                              {

work[i*8+j][18] = true;
```

```
                              }
                              if(holder[1][1] ==1)
                              {

work[i*8+j][9] = true;

                              }
                      }
              }
              if(j == 3)
              {
                      if(i==0)
                      {
                              if(holder[0][0] ==1)
                              {

work[i*8+j][0] = true;

                              }
                              if(holder[1][3] ==1)
                              {

work[i*8+j][11] = true;

                              }
                              if(holder[0][5] ==1)
                              {

work[i*8+j][5] = true;
```

```
                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][3] ==1)
                              {



work[i*8+j][3] = true;

                              }
                              if(holder[1][0] ==1)
                              {



work[i*8+j][8] = true;

                              }
                              if(holder[1][5] ==1)
                              {



work[i*8+j][13] = true;

                              }
                              if(holder[2][3] ==1)
                              {



work[i*8+j][19] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][0] ==1)
                              {







work[i*8+j][16] = true;


                              }
                              if(holder[1][3] ==1)
                              {







work[i*8+j][11] = true;


                              }
                              if(holder[2][5] ==1)
                              {







work[i*8+j][21] = true;


                              }
                      }
              }
              if(j == 4)
              {
                      if(i==0)
                      {
                              if(holder[0][2] ==1)
                              {
```

```
work[i*8+j][2] = true;

                              }
                              if(holder[1][4] ==1)
                              {

work[i*8+j][12] = true;

                              }
                              if(holder[0][7] ==1)
                              {

work[i*8+j][7] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][4] ==1)
                              {

work[i*8+j][4] = true;

                              }
                              if(holder[1][2] ==1)
                              {

work[i*8+j][10] = true;
```

```
                               }
                               if(holder[1][7] ==1)
                               {

work[i*8+j][15] = true;

                               }
                               if(holder[2][4] ==1)
                               {

work[i*8+j][20] = true;

                               }
                       }
                       if(i==2)
                       {
                               if(holder[2][2] ==1)
                               {

work[i*8+j][18] = true;

                               }

                               if(holder[2][7] ==1)
                               {

work[i*8+j][23] = true;
```

```
                              }
                              if(holder[1][4] ==1)
                              {

work[i*8+j][12] = true;

                              }
                      }
              }
              if(j == 6)
              {
                      if(i==0)
                      {
                              if(holder[0][5] ==1)
                              {

work[i*8+j][5] = true;

                              }
                              if(holder[1][6] ==1)
                              {

work[i*8+j][14] = true;

                              }
                              if(holder[0][7] ==1)
                              {
```

```
work[i*8+j][7] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][6] ==1)
                              {

work[i*8+j][6] = true;

                              }
                              if(holder[1][5] ==1)
                              {

work[i*8+j][13] = true;

                              }
                              if(holder[1][7] ==1)
                              {

work[i*8+j][15] = true;

                              }
                              if(holder[2][6] ==1)
                              {

work[i*8+j][22] = true;
```

```
                                          }
                                  }
                                  if(i==2)
                                  {
                                          if(holder[2][5] ==1)
                                          {



            work[i*8+j][21] = true;


                                          }
                                          if(holder[2][7] ==1)
                                          {



            work[i*8+j][23] = true;


                                          }
                                          if(holder[1][6] ==1)
                                          {

          work[i*8+j][14] = true;


                                          }
                                  }
                          }
                  }
          }
  }

for(i =0; i<24; i++)
{
for(int j=0; j<24; j++)
{
 if(work[i][j] == true)
 {
  thirdvalue = thirdvalue +2;
 }
}

}
```

```
}

void analyzecheckmoveabilityplayer(int holder[3][8], int & thirdvalue)
{
bool work[23][23];
for(int k=0; k<23; k++)
{
 for(int l=0; l<23; l++)
 {
  work[k][l] =false;
 }
}
  for (int i=0; i<3; i++)
  {
          for(int j=0; j<8; j++)
          {
                  if (holder[i][j] == 0)
                  {
                          //corners
                          if(j == 0)
                          {
                                  if(i==0)
                                  {
                                          if(holder[0][1] ==2)
                                          {



            work[i*8+j][1] = true;


                                          }
                                          if(holder[1][0] ==2)
                                          {



            work[i*8+j][8] = true;


                                          }
                                          if(holder[0][3] ==2)
                                          {
```

```
work[i*8+j][3] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][0] ==2)
                              {

work[i*8+j][9] = true;

                              }
                              if(holder[1][1] ==2)
                              {

work[i*8+j][9] = true;

                              }
                              if(holder[1][3] ==2)
                              {

work[i*8+j][11] = true;

                              }
                              if(holder[2][0] ==2)
                              {

work[i*8+j][16] = true;
```

```
                                }
                        }
                        if(i==2)
                        {
                                if(holder[2][1] ==2)
                                {

work[i*8+j][17] = true;

                                }
                                if(holder[2][3] ==2)
                                {

work[i*8+j][19] = true;

                                }
                                if(holder[1][0] ==2)
                                {

work[i*8+j][18] = true;

                                }
                        }
                }
                if(j == 2)
                {
                        if(i==0)
                        {
                                if(holder[0][1] ==2)
                                {
```

```
work[i*8+j][1] = true;


                              }
                              if(holder[1][2] ==2)
                              {






work[i*8+j][10] = true;


                              }
                              if(holder[0][4] ==2)
                              {






work[i*8+j][4] = true;


                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][2] ==2)
                              {






work[i*8+j][2] = true;


                              }
                              if(holder[1][1] ==2)
                              {
```

```
work[i*8+j][9] = true;

                              }
                              if(holder[1][4] ==2)
                              {

work[i*8+j][12] = true;

                              }
                              if(holder[2][2] ==2)
                              {

work[i*8+j][18] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][1] ==2)
                              {

work[i*8+j][17] = true;

                              }
                              if(holder[2][4] ==2)
                              {

work[i*8+j][20] = true;
```

```
                              }
                              if(holder[1][2] ==2)
                              {



work[i*8+j][10] = true;

                              }
                      }
              }
              if(j == 5)
              {
                      if(i==0)
                      {
                              if(holder[0][3] ==2)
                              {



work[i*8+j][3] = true;

                              }
                              if(holder[1][5] ==2)
                              {



work[i*8+j][13] = true;

                              }
                              if(holder[0][6] ==2)
                              {
```

```
work[i*8+j][6] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][5] ==2)
                              {

work[i*8+j][5] = true;

                              }
                              if(holder[1][3] ==2)
                              {

work[i*8+j][11] = true;

                              }
                              if(holder[1][6] ==2)
                              {

work[i*8+j][14] = true;

                              }
                              if(holder[2][5] ==2)
                              {

work[i*8+j][21] = true;
```

```
                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][3] ==2)
                              {

work[i*8+j][19] = true;

                              }
                              if(holder[1][5] ==2)
                              {

work[i*8+j][13] = true;

                              }
                              if(holder[2][6] ==2)
                              {

work[i*8+j][22] = true;

                              }
                      }
              }
              if(j == 7)
              {
                      if(i==0)
                      {
                              if(holder[0][6] ==2)
                              {
```

```
work[i*8+j][6] = true;

                              }
                              if(holder[1][7] ==2)
                              {

work[i*8+j][15] = true;

                              }
                              if(holder[0][4] ==2)
                              {

work[i*8+j][4] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][7] ==2)
                              {

work[i*8+j][7] = true;

                              }
                              if(holder[1][6] ==2)
                              {
```

```
work[i*8+j][14] = true;

                              }
                              if(holder[1][4] ==2)
                              {


work[i*8+j][12] = true;

                              }
                              if(holder[2][7] ==2)
                              {


work[i*8+j][23] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][6] ==2)
                              {


work[i*8+j][22] = true;

                              }
                              if(holder[1][7] ==2)
                              {


work[i*8+j][15] = true;
```

```
                              }
                              if(holder[2][4] ==2)
                              {

work[i*8+j][20] = true;

                              }
                      }
              }
              //sides
              if(j == 1)
              {
                      if(i==0)
                      {
                              if(holder[0][0] ==2)
                              {

work[i*8+j][0] = true;

                              }
                              if(holder[1][1] ==2)
                              {

work[i*8+j][9] = true;

                              }
                              if(holder[0][2] ==2)
                              {
```

```
work[i*8+j][2] = true;


                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][1] ==2)
                              {







work[i*8+j][1] = true;


                              }
                              if(holder[1][0] ==2)
                              {







work[i*8+j][8] = true;


                              }
                              if(holder[1][2] ==2)
                              {







work[i*8+j][10] = true;


                              }
                              if(holder[2][1] ==2)
                              {







work[i*8+j][17] = true;
```

```
                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][0] ==2)
                              {



work[i*8+j][16] = true;

                              }
                              if(holder[2][2] ==2)
                              {



work[i*8+j][18] = true;

                              }
                              if(holder[1][1] ==2)
                              {



work[i*8+j][9] = true;

                              }
                      }
              }
              if(j == 3)
              {
                      if(i==0)
                      {
                              if(holder[0][0] ==2)
                              {
```

```
work[i*8+j][0] = true;

                              }
                              if(holder[1][3] ==2)
                              {

work[i*8+j][11] = true;

                              }
                              if(holder[0][5] ==2)
                              {

work[i*8+j][5] = true;

                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][3] ==2)
                              {

work[i*8+j][3] = true;

                              }
                              if(holder[1][0] ==2)
                              {
```

```
work[i*8+j][8] = true;

                              }
                              if(holder[1][5] ==2)
                              {



work[i*8+j][13] = true;

                              }
                              if(holder[2][3] ==2)
                              {



work[i*8+j][19] = true;

                              }
                      }
                      if(i==2)
                      {
                              if(holder[2][0] ==2)
                              {



work[i*8+j][16] = true;

                              }
                              if(holder[1][3] ==2)
                              {



work[i*8+j][11] = true;
```

```
                              }
                              if(holder[2][5] ==2)
                              {
work[i*8+j][21] = true;
                              }
                      }
              }
              if(j == 4)
              {
                      if(i==0)
                      {
                              if(holder[0][2] ==2)
                              {
work[i*8+j][2] = true;
                              }
                              if(holder[1][4] ==2)
                              {
work[i*8+j][12] = true;
                              }
                              if(holder[0][7] ==2)
                              {
work[i*8+j][7] = true;
```

```
                               }
                       }
                       if(i==1)
                       {
                               if(holder[0][4] ==2)
                               {



work[i*8+j][4] = true;

                               }
                               if(holder[1][2] ==2)
                               {



work[i*8+j][10] = true;

                               }
                               if(holder[1][7] ==2)
                               {



work[i*8+j][15] = true;

                               }
                               if(holder[2][4] ==2)
                               {



work[i*8+j][20] = true;
```

```
                                }
                        }
                        if(i==2)
                        {
                                if(holder[2][2] ==2)
                                {







work[i*8+j][18] = true;


                                }

                                if(holder[2][7] ==2)
                                {








work[i*8+j][23] = true;



                                }
                                if(holder[1][4] ==2)
                                {







work[i*8+j][12] = true;


                                }
                        }
                }
                if(j == 6)
                {
                        if(i==0)
                        {
                                if(holder[0][5] ==2)
                                {
```

```
work[i*8+j][5] = true;


                              }
                              if(holder[1][6] ==2)
                              {







work[i*8+j][14] = true;


                              }
                              if(holder[0][7] ==2)
                              {







work[i*8+j][7] = true;


                              }
                      }
                      if(i==1)
                      {
                              if(holder[0][6] ==2)
                              {







work[i*8+j][6] = true;


                              }
                              if(holder[1][5] ==2)
                              {
```

```
work[i*8+j][13] = true;

                               }
                               if(holder[1][7] ==2)
                               {

work[i*8+j][15] = true;

                               }
                               if(holder[2][6] ==2)
                               {

work[i*8+j][22] = true;

                               }
                       }
                       if(i==2)
                       {
                               if(holder[2][5] ==2)
                               {

work[i*8+j][21] = true;

                               }
                               if(holder[2][7] ==2)
                               {
```

```
            work[i*8+j][23] = true;


                                          }
                                          if(holder[1][6] ==2)
                                          {


          work[i*8+j][14] = true;


                                          }
                                  }
                          }
                  }
          }
  }

for(i =0; i<24; i++)
{
for(int j=0; j<24; j++)
{
 if(work[i][j] == true)
 {
  thirdvalue = thirdvalue -2;
 }
}

}
}

void analyzecomputerboardlocation(int board[3][8], int & thirdvalue)
{
if(board[0][0]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[0][2]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[0][5]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[0][7]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[2][0]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[2][2]==1)
{
thirdvalue = thirdvalue + 2;
}
```

```
if(board[2][5]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[2][7]==1)
{
thirdvalue = thirdvalue + 2;
}
if(board[1][0]==1)
{
thirdvalue = thirdvalue + 7;
}
if(board[1][2]==1)
{
thirdvalue = thirdvalue + 7;
}
if(board[1][5]==1)
{
thirdvalue = thirdvalue + 7;
}
if(board[1][7]==1)
{
thirdvalue = thirdvalue + 7;
}
if(board[1][1]==1)
{
thirdvalue = thirdvalue + 4;
}
if(board[1][3]==1)
{
thirdvalue = thirdvalue + 4;
}
if(board[1][4]==1)
{
thirdvalue = thirdvalue + 4;
}
if(board[1][6] ==1)
{
thirdvalue = thirdvalue + 4;
}

}
void analyzeplayerboardlocation(int board[3][8], int & thirdvalue)
{

if(board[0][0]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[0][2]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[0][5]==2)
{
thirdvalue = thirdvalue - 2;
}
```

```
if(board[0][7]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[2][0]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[2][2]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[2][5]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[2][7]==2)
{
thirdvalue = thirdvalue - 2;
}
if(board[1][0]==2)
{
thirdvalue = thirdvalue - 7;
}
if(board[1][2]==2)
{
thirdvalue = thirdvalue - 7;
}
if(board[1][5]==2)
{
thirdvalue = thirdvalue - 7;
}
if(board[1][7]==2)
{
thirdvalue = thirdvalue - 7;
}
if(board[1][1]==2)
{
thirdvalue = thirdvalue - 4;
}
if(board[1][3]==2)
{
thirdvalue = thirdvalue - 4;
}
if(board[1][4]==2)
{
thirdvalue = thirdvalue - 4;
}
if(board[1][6] ==2)
{
thirdvalue = thirdvalue - 4;
}

}
```