public class WordSearch {

 private static ArrayList<String> board = new ArrayList<String>();

 private static Scanner scanner;

 public static void readBoard() {

 String line;

 while (true) {

 line = scanner.nextLine();

 if (line.equals(""))

 break;

 board.add(line.replaceAll(" ","").toUpperCase());

 }

 }

 public static void printBoard() {

 int rows = getRows();

 for (int row=0; row<rows; row++)

 System.out.println(board.get(row));

 }

 public static int getRows() {

 return board.size();

 }

 public static int getCols() {

 return board.get(0).length();

 }

 public static void createScanner() {

 try {

 scanner = new Scanner(new java.io.File("WordSearchInput.txt"));

 } catch (Exception e) {

 System.out.println();

 System.exit(1);

 }

 }

 public static void processWords() {

 while (scanner.hasNext()) {

 String word = scanner.next();

 findWord(word);

 }

 }

 public static void findWord(String word) {

 int rows = getRows();

 int cols = getCols();

 for (int row=0; row<rows; row++)

 for (int col=0; col<cols; col++)

 findWord(word,row,col);

 }

 public static void findWord(String word, int row, int col) {

 for (int drow=-1; drow<=1; drow++)

 for (int dcol=-1; dcol<=1; dcol++)

 findWord(word,row,col,drow,dcol);

 }

 public static void findWord(String word, int row, int col, int drow, int dcol) {

 int rows = getRows();

 int cols = getCols();

 for (int offset=0; offset<word.length(); offset++) {

 int targetRow = row + offset\*drow;

 int targetCol = col + offset\*dcol;

 if ((targetRow < 0) ||

 (targetRow >= rows) ||

 (targetCol < 0) ||

 (targetCol >= cols))

 return;

 char boardChar = board.get(targetRow).charAt(targetCol);

 char wordChar = word.charAt(offset);

 if (boardChar != wordChar)

 // mismatch, so we're done

 return;

 }

 System.out.printf("%s at %d,%d direction %d,%d\n",

 word, row, col, drow, dcol);

 }

 public static void main(String[] args) {

 createScanner();

 readBoard();

 printBoard();

 processWords();

 }

}