```
// Node to represent each element in 2x2 matrix
struct Node:
    Char ch
    Bool sol

// WordSearch class
class WordSearch {
    public:
    WordSearch(int, int, vector<vector<Node>>);
    void solve(int, char**);
    private:
    int, x,y;
    void findWord(char*);
    void drawWord(char*, int, int,int,int);
    bool checkWord(char*, int,int,int,int);
    void printMatrix();
}

// main method
int main(int argc, char **argv):
    Int row, col
    Input row,col
    Vector<Vector<Node> matrix
    matrix.resize(row)
     for i in range(row)
        matrix[i].resize(col) f
        for j in range(col)
            Node n
            Input n.ch
             n.sol = false
            matrix[i][j] = n
    WordSearch wordsearch(row, col, matrix)
    solve(argc, argv)
    return 0
// solve
void solve(int len, char**words) {
    For i in range(len)
        findWord(*(words+i))
    printMatrix()
}

// find word in matrix
void findWord(char* word):
    for i in range(matrix.size())
```

```
                for j in range(matrix[0].size())
                if checkWord(matrix,word,i,0,j,1)
                        drawWord(matrix,word,i,0,j,1)
                if checkWord(matrix,word,i,0,j,-1)
                        drawWord(matrix,word,i,0,j,-1)
                if checkWord(matrix,word,i,1,j,0)
                        drawWord(matrix,word,i,1,j,0)
                if checkWord(matrix,word,i,-1,j,0)
                        drawWord(matrix,word,i,-1,j,0)
                if checkWord(matrix,word,i,1,j,1)
                        drawWord(matrix,word,i,1,j,1)
                if checkWord(matrix,word,i,1,j,-1)
                        drawWord(matrix,word,i,1,j,-1)
                if checkWord(matrix,word,i,-1,j,-1)
                        drawWord(matrix,word,i,-1,j,-1)
                if checkWord(matrix,word,i,-1,j,1)
                        drawWord(matrix,word,i,-1,j,1)



// draw wod
void drawWord(char* word, int x, int delX, int y, int delY):
        int count = 0
        while count < length(word)
                 matrix[x][y].sol = true
                count++
                x += delX
                y += delY

bool checkWord(char* word, int x, int delX, int y, int delY):
        int count = 0
        while count < length(word)
                 if x<0 or x>= matrix.size() or y<0 or y>=matrix[0].size()
                        return false
                if matrix[x][y].chr != *(word + count)
                        return false
                count++
                x += delX
                y += delY
        return true
void printMatrix(char* word, int x, int delX, int y, int delY):
        for i in range(matrix.size())
                for j in range(matrix[0].size())
                        if matrix[i][j].solution
```

```
                    print (matrix[i][j].chr, “ “)
          else
                    print(“* “)
print(“”) //new line
```