import java.util.ArrayList;

import java.util.List;

/\* \*

\* Es lo que pude hacer de la sopa de letras

\* Ricardo Gabriel Rodriguez Gonzalez

\* 17001433

\*/

public class sopa\_Letras {

private char[][] matriz = {{'a', 'b', 'c', 'd', 'e', 'a','f'},

{'g', 'h', 'h', 'j', 'r', 'l', 'u'},

{'u', 'm', 'o', 'r', 'p', 'q', 'e'},

{'a', 's', 'l', 'u', 'v', 'w', 'g'},

{'x', 'i', 'a', 'a', 'b', 'c', 'l'},

{'t', 'f', 'u', 'e', 'g', 'o', 's'}};

public char[][] getMatriz() {

return matriz;

}

public void setMatriz(char[][] nuevaMatriz) {

matriz = nuevaMatriz;

}

public void mostrarMatriz(){

System.out.println("==== SOPA DE LETRAS ====");

for(int i=0; i < matriz.length; i++){

for(int j=0; j < matriz[i].length; j++){

System.out.print(" " + matriz[i][j] + " ");

}

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

}

}

public String resolver(String palabra){

for( int[] pos : posiblesSolucionesDe(palabra) ){

// Buscar horizontalmente hacia derecha.

String palabraEncontrada = palabraEnMatriz(pos, palabra.length(), 0, 1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n horizontal derecha";

// Buscar horizontalmente hacia izquierda.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), 0, -1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n horizontal izquierda";

// Buscar verticalmente hacia abajo.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), 1, 0);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n vertical hacia abajo";

// Buscar verticalmente hacia arriba.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), -1, 0);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n vertical hacia arriba";

// Buscar diagonal superior derecha.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), -1, 1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n diagonal superior derecha";

// Buscar diagonal superior izquierda.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), -1, -1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n diagonal superior izquierda";

// Buscar diagonal inferior derecha.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), 1, 1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n diagonal inferior derecha";

// Buscar diagonal inferior izquierda.

palabraEncontrada = palabraEnMatriz(pos, palabra.length(), 1, -1);

if(palabraEncontrada.equals(palabra))

return "palabra '"+ palabra +"' encontrada a partir de la posición [" + pos[0] + "," + pos[1] + "] de la matriz con orientaci—n diagonal inferior izquierda";

}

return "La palabra '" +palabra+ "' no fue encontrada en la sopa de letras";

}

/\*

\* Retorna indice invertido de las posiciones donde puede

\* resolverse una palabra buscada.

\*/

public int[][] posiblesSolucionesDe(String palabra) {

char primeraLetra = palabra.charAt(0);

List<int[]> indiceInvertido = new ArrayList<int[]>();

for(int i=0; i < matriz.length; i++){

for(int j=0; j < matriz[i].length; j++){

if(matriz[i][j] == primeraLetra){

indiceInvertido.add(new int[]{i, j}); // Guardar la posicion de la letra en la matriz.

}

}

}

return toArrayInt(indiceInvertido);

}

private int[][] toArrayInt( List<int[]> list ){

return (int[][]) list.toArray( new int[list.size()][list.get(0).length]);

}

public String palabraEnMatriz(int[] posInicial, int numeroCaracteres, int moverEnFila, int moverEnColumna) {

String palabra = "";

int recorrido = 0, fila = posInicial[0], columna = posInicial[1];

while((recorrido < numeroCaracteres) &&

(fila < matriz.length && columna < matriz.length) &&

(fila > -1 && columna > -1)) {

palabra += matriz[fila][columna];

fila = fila + moverEnFila;

columna = columna + moverEnColumna;

recorrido++;

}

return palabra;

}

public static void main(String[] args){

sopa\_Letras sopaDeLetras = new sopa\_Letras();

sopaDeLetras.mostrarMatriz();

System.out.println(sopaDeLetras.resolver("agua"));

System.out.println(sopaDeLetras.resolver("tierra"));

System.out.println(sopaDeLetras.resolver("fuego"));

System.out.println(sopaDeLetras.resolver("hola"));

}

}