**Özgür Gürcan 2016510032**

**Definition:**

There are 3 classes which `GUI`, `Main` and `Word`.

1. ` Word` Class

There are 2 variables which `name` and `distance`. name is a string which store the word name. Distance is an int that describes how many operations are necessary.

This class implements Comparable because we need to sort this class’s objects according to objects' distance.

1. `GUI` class

Just have Text Area for giving output and Text Field for taking input.

1. `Main` Class

The most important feature of this program is it runs simultaneity. This means it calculates med as you write or delete anything.

GUI gui = **new** GUI();

String[] tokens = *readFileAsString*(System.*getProperty*("user.dir") + "/tr\_dictionary.txt").split("\\n");

// ### When input changed ### //

*list*(tokens,gui.myWord.getText(),gui);

And list function;

list(String[] tokens, String userWord, GUI gui) {

// ### Start Timer here ### //

Word[] differentWords = *differentWords*(tokens, userWord);

Arrays.*sort*(differentWords);

// clear text box here //

**for** (**int** i = 0; i < 5; i++) {

gui.text.append(differentWords[i].toString() + "\n");

}

// ### END Timer here ### //

}

**The general idea for med algorithm:**

1-Read file and split into array `tokens`.

2-Take user input `userWord`

3-find distance between `userWord` and tokens array `differentWords`

4-sort array `differentWords`

5-list top 5 item

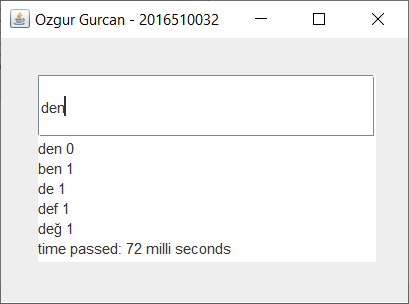
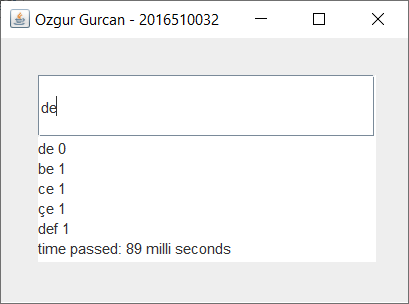
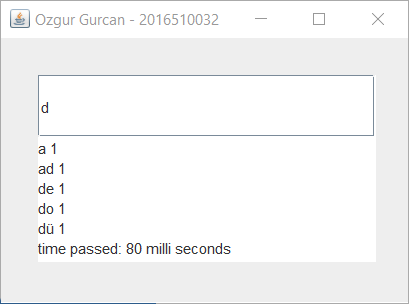
**Input:** any word from the user.

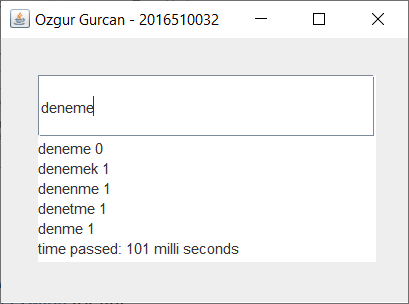
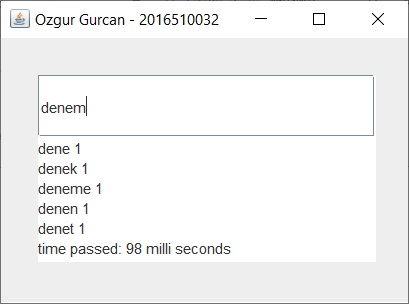
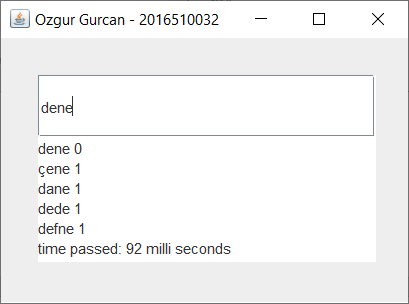
**Output:** best 5 words according to med algorithm and calculation time. Shown in Figure 1.

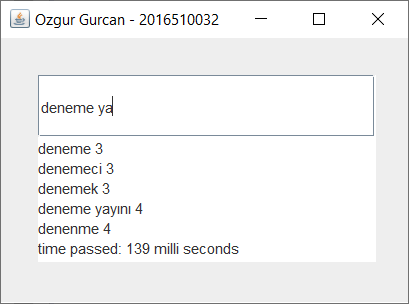
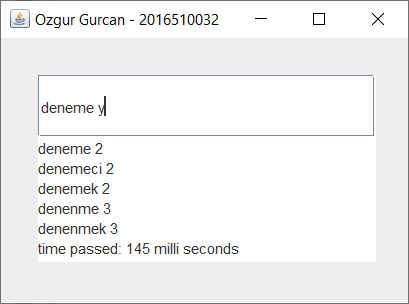
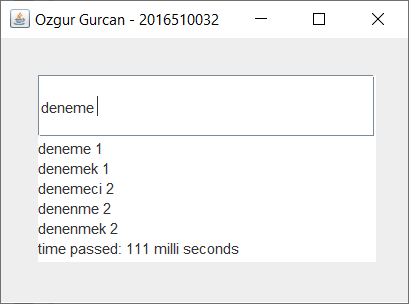
**Total running time:**

The average running time depends on CPU speed. On test system (windows 10, 16gb ram and 8750h CPU and runs on SSD, macOS 14.05, 16gb ram and 8257u CPU runs on SSD) minimum calculation time was 50milli seconds. As the input word length increases, the working time increases

**Promram output:**







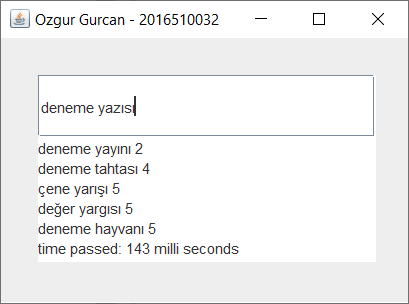
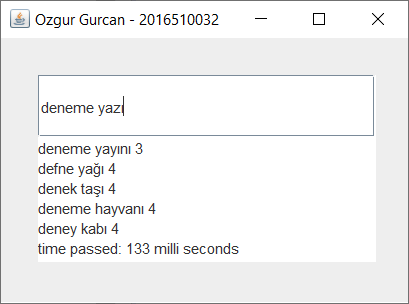
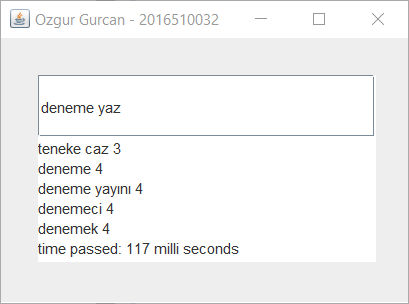


Figure 1. Program output

**References:**

1. <https://www.geeksforgeeks.org/edit-distance-dp-5> for distance calculation
2. <https://www.javatpoint.com/java-swing> for gui