TUGAS KECIL IF2211
STRATEGI ALGORITMA

PENYELESAIAN WORD PUZZLE DENGAN ALGORITMA BRUTE FORCE

Disusun oleh
Farrel Farandieka Fibriyanto - 13520054

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG

Semester II Tahun 2021/2022

Daftar Isi

Algoritma Brute Force

Ketika diberikan masukan berupa file txt, program akan membaca isinya yang berupa word puzzle dan memasukkan puzzle ke dalam matrix char serta word yang perlu dicari dalam array. Kemudian akan dijalankan algoritma bruteforce sebagai berikut:

A. Dimulai dari huruf di kiri atas matrix, akan dimulai mengecek apakah terdapat suatu kata
    a. Mulai mengecek dari arah ke atas, kemudian sesuai arah jarum jam, cek arah lainnya. Berarti dari arah ke atas, arah ke atas-kanan, arah ke-kanan, dst. Cek kata pertama dalam array.
        i. Cek terlebih dahulu apakah kata tersebut muat, misalnya apabila kata yang ingin dicek memiliki panjang 3 huruf, tetapi huruf yang dicek sekarang hanya muat kata 2 huruf, maka tidak dilakukan pengecekan
        ii. Apabila kata yang dicek muat, cek satu per satu huruf dari kata yang dicek mulai dari huruf yang dalam pengecekkan sekarang ke arah pengecekkan apakah akan terus cocok atau tidak
            1. Apabila cocok, print kata tersebut ke arah yang cocok dengan huruf lain berupa '_' sehingga terlihat arah kata, dan hapus kata tersebut dalam array kata serta tidak akan dicek ulang.
            2. Apabila tidak cocok, batalkan pengecekan terhadap arah tersebut.
        iii. Lanjutkan pengecekkan ke arah berikutnya, kembali ke tahap i.
            1. Apabila semua arah sudah dicek, lanjut ke tahap b.
    b. Apabila masih terdapat kata lain yang belum dicek, kembali ke tahap a, tetapi lakukan pengecekan terhadap kata selanjutnya dalam array.
        i. Apabila semua kata sudah dicek, lanjut ke tahap B.
B. Apabila pengecekkan semua kata terhadap satu huruf sudah dilakukan, lanjut ke huruf berikutnya yaitu samping kanan huruf saat ini. Apabila tidak ada huruf di samping kanan, kembali ke huruf pertama baris berikutnya.

Source Program

1. Main.java

```
class Main {
	public static void main (String[] args){

	// Inisialisasi var
	Wordpuzzle wordpuzzle = new Wordpuzzle();

	// mengisi puzzle
	wordpuzzle.fillPuzzle();

	// mengeprint hasil pencarian
	wordpuzzle.bruteforceFindWords(true);

	// menghitung waktu pencarian (tanpa waktu mengeprint)
	// kalau ditambahkan bersama yg atas, bisa berdetik-detik waktu
tambahannya
	wordpuzzle.bruteforceFindWords(false);
	}
}
```

2. Wordpuzzle.java

```
import java.io.File;
import java.util.Scanner;
import java.io.FileNotFoundException;

public class Wordpuzzle {

	// Inisialisasi var
	char matrix[][] = new char[50][50];
	int iEff = 0; int jEff = 0;
	String[] words = new String[50];
	int wordsEff = 0;
	Statistics stats = new Statistics();
```

```
	void printWordpuzzle(){
	for (int i = 0; i < this.iEff; i++){
		for (int j = 0; j < this.jEff; j++){
			System.out.print(this.matrix[i][j]);
			System.out.print(" ");
		}
		System.out.println("");
	}

	System.out.println("");
	for (int i = 0; i < this.wordsEff; i++){
		System.out.println(words[i]);
	}
	}

	void fillPuzzle(){
	try {
	Scanner scanner = new Scanner(System.in);
	System.out.print("Input filename (with '.txt' on the end, with
file on the 'test' folder): ");
	String filename = scanner.next();
	scanner.close();
	File txtFile = new File("../test/" + filename);
	Scanner txtReader = new Scanner(txtFile);
	int n = 0; int m = 0;
	boolean hasReadPuzzle = false;

	//reads the puzzle
	while(txtReader.hasNextLine() && !hasReadPuzzle){
		String line = txtReader.nextLine();
		char[] lineChars = line.toCharArray();
		int b = 0;
		if (lineChars.length == 0){
			hasReadPuzzle = true;
			break;
		}
		else {
			for (int a = 0; a < lineChars.length; a++) {
```

```
                if(Character.isLetter(lineChars[a])){
                      this.matrix[n][b] = lineChars[a];
                      b++;
                }
                }
          }
          m = b;
          n++;
     }

     int c = 0;
     // reads the words that needs to be found in the puzzle
     while(txtReader.hasNextLine()){
          String line = txtReader.nextLine();
          this.words[c] = line;
          c++;
     }

     this.iEff = n;
     this.jEff = m;
     this.wordsEff = c;

     txtReader.close();
     }
     catch (FileNotFoundException e) {}
     }

     void printUpwards(int i, int j, int wordLen){
     for (int a = 0; a < this.iEff; a++){
          for (int b = 0; b < this.jEff; b++){
                if (((wordLen - i + a) > 0 && (a <= i)) && j == b)
System.out.print(this.matrix[a][b]);
                else System.out.print("_");
                System.out.print(" ");
          }
          System.out.println("");
     }
     }
```

```
      void printUpRight(int i, int j, int wordLen){
      int tempLen = wordLen - 1;

      for (int a = 0; a < this.iEff; a++){
            for (int b = 0; b < this.jEff; b++){
                  if (a == (i - tempLen) && b == (j + tempLen) &&
tempLen >= 0){
                  tempLen--;
                  System.out.print(this.matrix[a][b]);
                  }
                  else System.out.print("_");
                  System.out.print(" ");
            }
            System.out.println("");
      }
      }

      void printRight(int i, int j, int wordLen){
      for (int a = 0; a < this.iEff; a++){
            for (int b = 0; b < this.jEff; b++){
                  if (((b - j + 1) <= wordLen && (b >= j)) && i == a)
System.out.print(this.matrix[a][b]);
                  else System.out.print("_");
                  System.out.print(" ");
            }
            System.out.println("");
      }
      }

      void printDownRight(int i, int j, int wordLen){
      int tempLen = 0;

      for (int a = 0; a < this.iEff; a++){
            for (int b = 0; b < this.jEff; b++){
                  if (a == (i + tempLen) && b == (j + tempLen) &&
tempLen < wordLen){
                  tempLen++;
```

```
                System.out.print(this.matrix[a][b]);
                }
                else System.out.print("_");
                System.out.print(" ");
          }
          System.out.println("");
     }
     }

     void printDownwards(int i, int j, int wordLen){
     for (int a = 0; a < this.iEff; a++){
          for (int b = 0; b < this.jEff; b++){
                if (((a - i + 1) <= wordLen && (a >= i)) && j == b)
System.out.print(this.matrix[a][b]);
                else System.out.print("_");
                System.out.print(" ");
          }
          System.out.println("");
     }
     }

     void printDownLeft(int i, int j, int wordLen){
     int tempLen = 0;

     for (int a = 0; a < this.iEff; a++){
          for (int b = 0; b < this.jEff; b++){
                if (a == (i + tempLen) && b == (j - tempLen) &&
tempLen < wordLen){
                tempLen++;
                System.out.print(this.matrix[a][b]);
                }
                else System.out.print("_");
                System.out.print(" ");
          }
          System.out.println("");
     }
     }
```

```
      void printLeft(int i, int j, int wordLen){
      for (int a = 0; a < this.iEff; a++){
            for (int b = 0; b < this.jEff; b++){
                  if (((wordLen - j + b) > 0 && (b <= j)) && i == a)
System.out.print(this.matrix[a][b]);
                  else System.out.print("_");
                  System.out.print(" ");
            }
            System.out.println("");
      }
      }

      void printUpLeft(int i, int j, int wordLen){
      int tempLen = wordLen - 1;

      for (int a = 0; a < this.iEff; a++){
            for (int b = 0; b < this.jEff; b++){
                  if (a == (i - tempLen) && b == (j - tempLen) &&
tempLen >= 0){
                  tempLen--;
                  System.out.print(this.matrix[a][b]);
                  }
                  else System.out.print("_");
                  System.out.print(" ");
            }
            System.out.println("");
      }
      }

      void deleteWord(int idx){
      this.words[idx] = this.words[this.wordsEff - 1];
      this.wordsEff--;
      }

      void checkAllDirections(int i, int j, boolean print){
      int idx = 0;
      while(idx < this.wordsEff && this.wordsEff > 0){
            boolean found = false;
```

```
            int wordLen = this.words[idx].length();

            // checks up
            if (!found){
                  if (i + 1 - wordLen >= 0){
                  int traverseCheck = 0;
                  boolean matches = true;
                  while(traverseCheck < wordLen && matches){
                        this.stats.checkedALetter();
                        if (this.matrix[i - traverseCheck][j] !=
this.words[idx].charAt(traverseCheck)){
                              matches = false;
                        }
                        traverseCheck++;
                  }

                  if(matches) {
                        found = true;

                        // deals with printing
                        if (print) printUpwards(i, j, wordLen);

                        // deals with deleting word from word list
                        deleteWord(idx);
                  }
                  }
            }

            // checks up-right
            if (!found){
                  if (i + 1 - wordLen >= 0 && j + wordLen <= this.jEff){
                  int traverseCheck = 0;
                  boolean matches = true;
                  while(traverseCheck < wordLen && matches){
                        this.stats.checkedALetter();
                        if (this.matrix[i - traverseCheck][j +
traverseCheck] != this.words[idx].charAt(traverseCheck)){
                              matches = false;
```

```
                    }
                    traverseCheck++;
                }

                if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printUpRight(i, j, wordLen);

                    // deals with deleting word from word list
                    deleteWord(idx);
                }
                }
            }

            // checks right
            if (!found){
                if (j + wordLen <= this.jEff){
                int traverseCheck = 0;
                boolean matches = true;
                while(traverseCheck < wordLen && matches){
                    this.stats.checkedALetter();
                    if (this.matrix[i][j + traverseCheck] !=
this.words[idx].charAt(traverseCheck)){
                        matches = false;
                    }
                    traverseCheck++;
                }

                if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printRight(i, j, wordLen);

                    // deals with deleting word from word list
                    deleteWord(idx);
```

```
                }
                }
            }

            // checks down-right
            if (!found){
                if (i + wordLen <= this.iEff && j + wordLen <=
this.jEff){
                int traverseCheck = 0;
                boolean matches = true;
                while(traverseCheck < wordLen && matches){
                    this.stats.checkedALetter();
                    if (this.matrix[i + traverseCheck][j +
traverseCheck] != this.words[idx].charAt(traverseCheck)){
                        matches = false;
                    }
                    traverseCheck++;
                }

                if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printDownRight(i, j, wordLen);

                    // deals with deleting word from word list
                    deleteWord(idx);
                }
                }
            }

            // checks down
            if (!found){
                if (i + wordLen <= this.iEff){
                int traverseCheck = 0;
                boolean matches = true;
                while(traverseCheck < wordLen && matches){
                    this.stats.checkedALetter();
```

```
                    if (this.matrix[i + traverseCheck][j] !=
this.words[idx].charAt(traverseCheck)){
                         matches = false;
                    }
                    traverseCheck++;
               }

               if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printDownwards(i, j, wordLen);

                    // deals with deleting word from word list
                    deleteWord(idx);
               }
               }
          }

          // checks down-left
          if (!found){
               if (i + wordLen <= this.iEff && j + 1 - wordLen >= 0){
               int traverseCheck = 0;
               boolean matches = true;
               while(traverseCheck < wordLen && matches){
                    this.stats.checkedALetter();
                    if (this.matrix[i + traverseCheck][j -
traverseCheck] != this.words[idx].charAt(traverseCheck)){
                         matches = false;
                    }
                    traverseCheck++;
               }

               if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printDownLeft(i, j, wordLen);
```

```
                    // deals with deleting word from word list
                    deleteWord(idx);
               }
               }
          }

          // checks left
          if (!found){
               if (j + 1 - wordLen >= 0){
               int traverseCheck = 0;
               boolean matches = true;
               while(traverseCheck < wordLen && matches){
                    this.stats.checkedALetter();
                    if (this.matrix[i][j - traverseCheck] !=
this.words[idx].charAt(traverseCheck)){
                         matches = false;
                    }
                    traverseCheck++;
               }

               if(matches) {
                    found = true;

                    // deals with printing
                    if (print) printLeft(i, j, wordLen);

                    // deals with deleting word from word list
                    deleteWord(idx);
               }
               }
          }

          // checks up-left
          if (!found){
               if (i + 1 - wordLen >= 0 && j + 1 - wordLen >= 0){
               int traverseCheck = 0;
               boolean matches = true;
```

```
                while(traverseCheck < wordLen && matches){
                      this.stats.checkedALetter();
                      if (this.matrix[i - traverseCheck][j -
traverseCheck] != this.words[idx].charAt(traverseCheck)){
                            matches = false;
                      }
                      traverseCheck++;
                }

                if(matches) {
                      found = true;

                      // deals with printing
                      if (print) printUpLeft(i, j, wordLen);

                      // deals with deleting word from word list
                      deleteWord(idx);
                }
                }
          }

          if(found){
                System.out.println("");
          }
          else{
                idx++;
          }
     }

     }

     void bruteforceFindWords(boolean print){
     this.stats.startStopwatch();
     int i = 0; int j = 0;
     while (this.wordsEff > 0 && i < this.iEff){
          while(this.wordsEff > 0 && j < this.jEff){
                this.checkAllDirections(i, j, print);
                j++;
```

```
            }
            i++;
            j = 0;
      }
      this.stats.stopStopwatch();
      if (!(print)) {
            System.out.println("\nTime taken : " +
this.stats.getElapsedMilisecond() + " miliseconds");
            System.out.println("Checks done : " +
this.stats.getLetterCount());
      }
      }

}
```

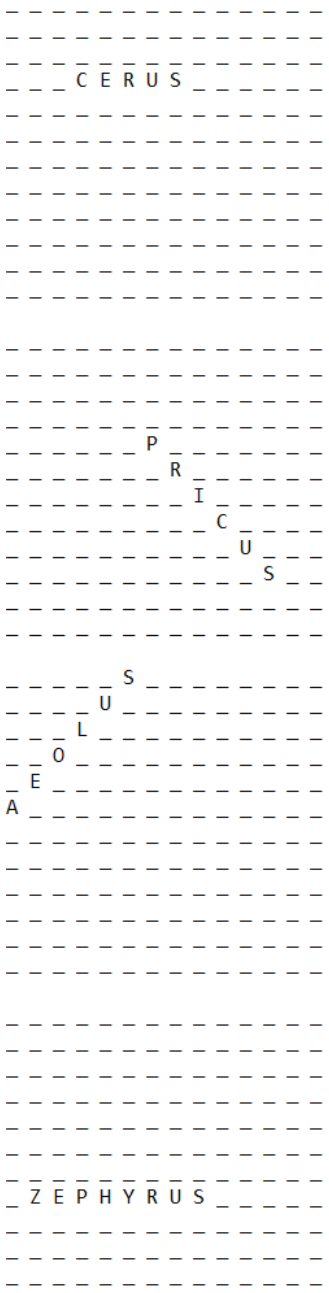
3. Statistics.java

```
public class Statistics {

	// Inisialisasi var
	private long stopwatchStart = 0;
	private long stopwatchEnd = 0;
	private long checkLetterCounter = 0;

	public void checkedALetter(){
	this.checkLetterCounter++;
	}

	public void startStopwatch(){
	this.stopwatchStart = System.currentTimeMillis();
	}

	public void stopStopwatch(){
	this.stopwatchEnd = System.currentTimeMillis();
	}

	public long getElapsedMilisecond(){
	return (long) this.stopwatchEnd - this.stopwatchStart;
	}

	public long getLetterCount(){
	return (long) checkLetterCounter;
	}
}
```

Screenshot

small1.txt

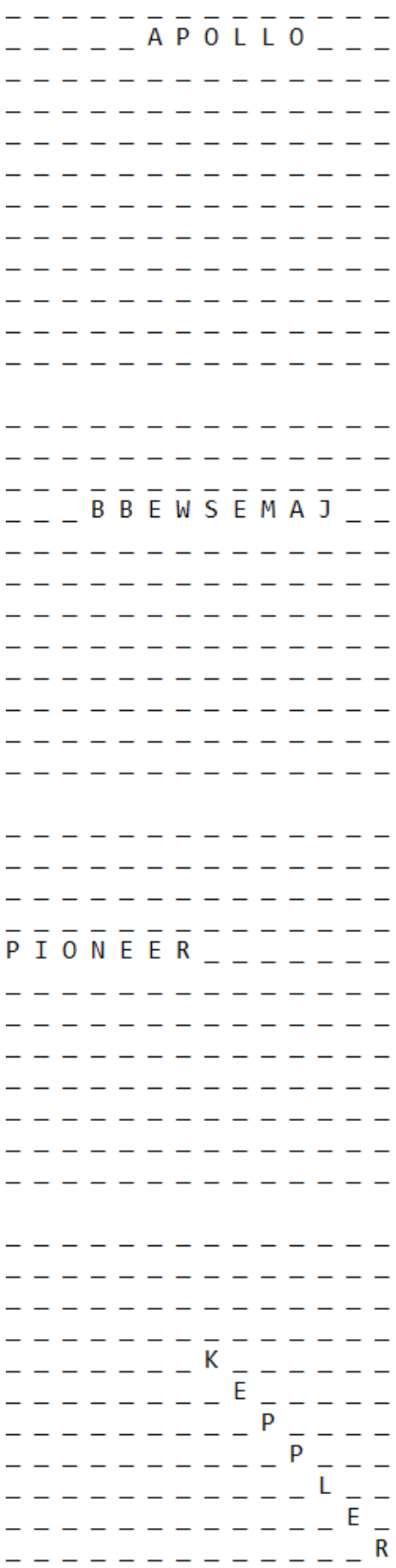
```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): small1.txt
_ _ _ _ _ _ C _ _ _ _ _ _ _
_ _ _ _ _ _ _ R _ _ _ _ _ _
_ _ _ _ _ _ _ _ O _ _ _ _ _
_ _ _ _ _ _ _ _ _ N _ _ _ _
_ _ _ _ _ _ _ _ _ _ U _ _ _
_ _ _ _ _ _ _ _ _ _ _ S _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ E _ _ _
_ _ _ _ _ _ _ _ _ _ R _ _ _
_ _ _ _ _ _ _ _ _ _ E _ _ _
_ _ _ _ _ _ _ _ _ _ B _ _ _
_ _ _ _ _ _ _ _ _ _ U _ _ _
_ _ _ _ _ _ _ _ _ _ S _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ K
_ _ _ _ _ _ _ _ _ _ _ _ _ R
_ _ _ _ _ _ _ _ _ _ _ _ _ A
_ _ _ _ _ _ _ _ _ _ _ _ _ T
_ _ _ _ _ _ _ _ _ _ _ _ _ O
_ _ _ _ _ _ _ _ _ _ _ _ _ S
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
```

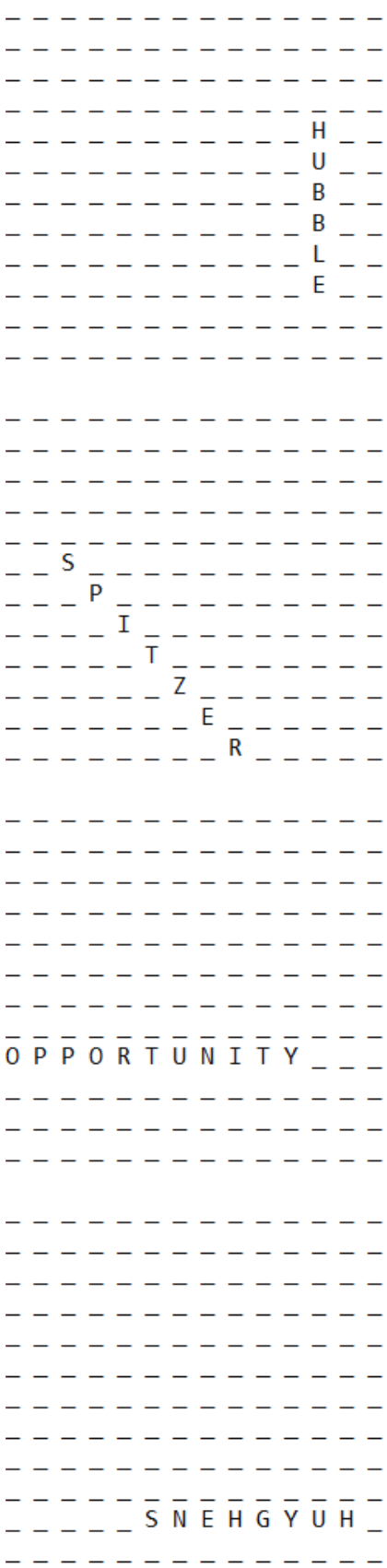

CERUS
PRICUS
SULOEA
ZEPHYRUS

```
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
N O D I E S O P _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ S _
_ _ _ _ _ _ _ _ _ _ _ _ _ U _
_ _ _ _ _ _ _ _ _ _ _ _ _ E _
_ _ _ _ _ _ _ _ _ _ _ _ _ H _
_ _ _ _ _ _ _ _ _ _ _ _ _ T _
_ _ _ _ _ _ _ _ _ _ _ _ _ E _
_ _ _ _ _ _ _ _ _ _ _ _ _ M _
_ _ _ _ _ _ _ _ _ _ _ _ _ O _
_ _ _ _ _ _ _ _ _ _ _ _ _ R _
_ _ _ _ _ _ _ _ _ _ _ _ _ P _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _


Time taken : 0 miliseconds
Checks done : 2543

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

small2.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): small2.txt
_ _ _ _ _ S _ _ _ _ _ _ _ _
_ _ _ _ P _ _ _ _ _ _ _ _ _
_ _ _ I _ _ _ _ _ _ _ _ _ _
_ _ R _ _ _ _ _ _ _ _ _ _ _
_ I _ _ _ _ _ _ _ _ _ _ _ _
T _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ G N I K I V _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ C
_ _ _ _ _ _ _ _ _ _ _ _ _ A
_ _ _ _ _ _ _ _ _ _ _ _ _ S
_ _ _ _ _ _ _ _ _ _ _ _ _ S
_ _ _ _ _ _ _ _ _ _ _ _ _ I
_ _ _ _ _ _ _ _ _ _ _ _ _ N
_ _ _ _ _ _ _ _ _ _ _ _ _ I
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
```

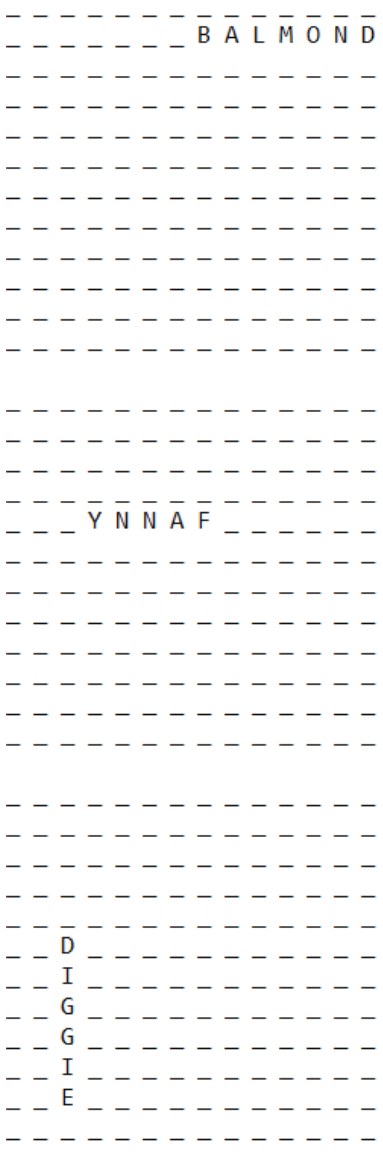

APOLLO
BBEWSEMAJ
PIONEER
K
E
P
P
L
E
R



```
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
C H A N D R A _ _ _ _ _ _ _


Time taken : 1 miliseconds
Checks done : 3519

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

small3.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): small3.txt
_ C _ _ _ _ _ _ _ _ _ _ _ _
_ _ H _ _ _ _ _ _ _ _ _ _ _
_ _ _ O _ _ _ _ _ _ _ _ _ _
_ _ _ _ U _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ E _ _ _ _ _ _ _ _ _
_ _ _ S _ _ _ _ _ _ _ _ _ _
_ _ T _ _ _ _ _ _ _ _ _ _ _
_ E _ _ _ _ _ _ _ _ _ _ _ _
S _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ K _ _ _ _ _ _ _ _
_ _ _ _ _ _ A _ _ _ _ _ _ _
_ _ _ _ _ _ _ G _ _ _ _ _ _
_ _ _ _ _ _ _ _ U _ _ _ _ _
_ _ _ _ _ _ _ _ _ R _ _ _ _
_ _ _ _ _ _ _ _ _ _ A _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _
```


BALMOND
YNNAF
D
I
G
G
I
E

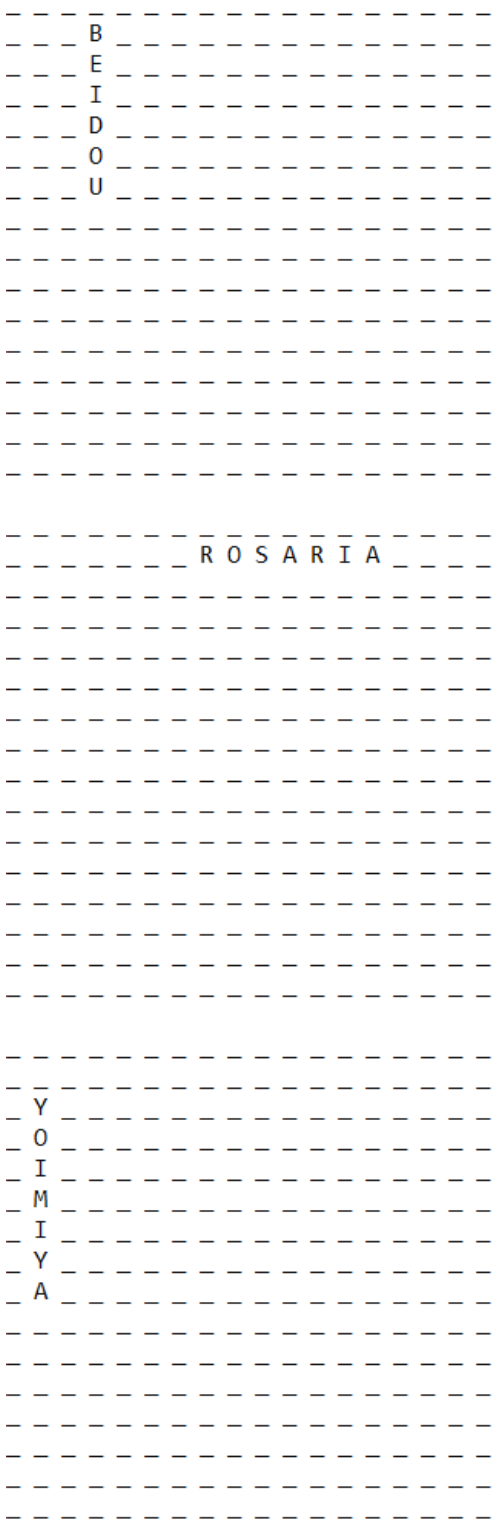

D
R
A
C
U
L
A
L
E
H
T
I
R
I
D
R
O
G
J O H N S O N

```
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ Y _ _ _ _ _ _
_ _ _ _ _ _ _ _ E _ _ _ _ _ _
_ _ _ _ _ _ _ _ L _ _ _ _ _ _
_ _ _ _ _ _ _ _ R _ _ _ _ _ _
_ _ _ _ _ _ _ _ A _ _ _ _ _ _
_ _ _ _ _ _ _ _ H _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

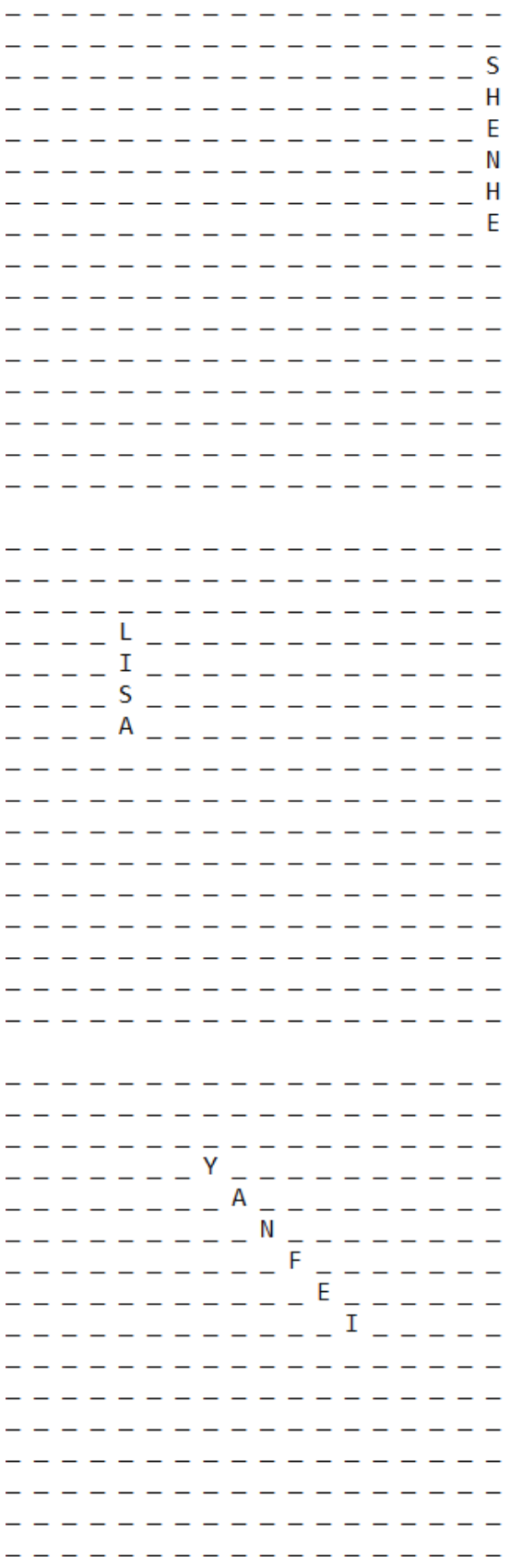
```
Time taken : 0 miliseconds
Checks done : 3382

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

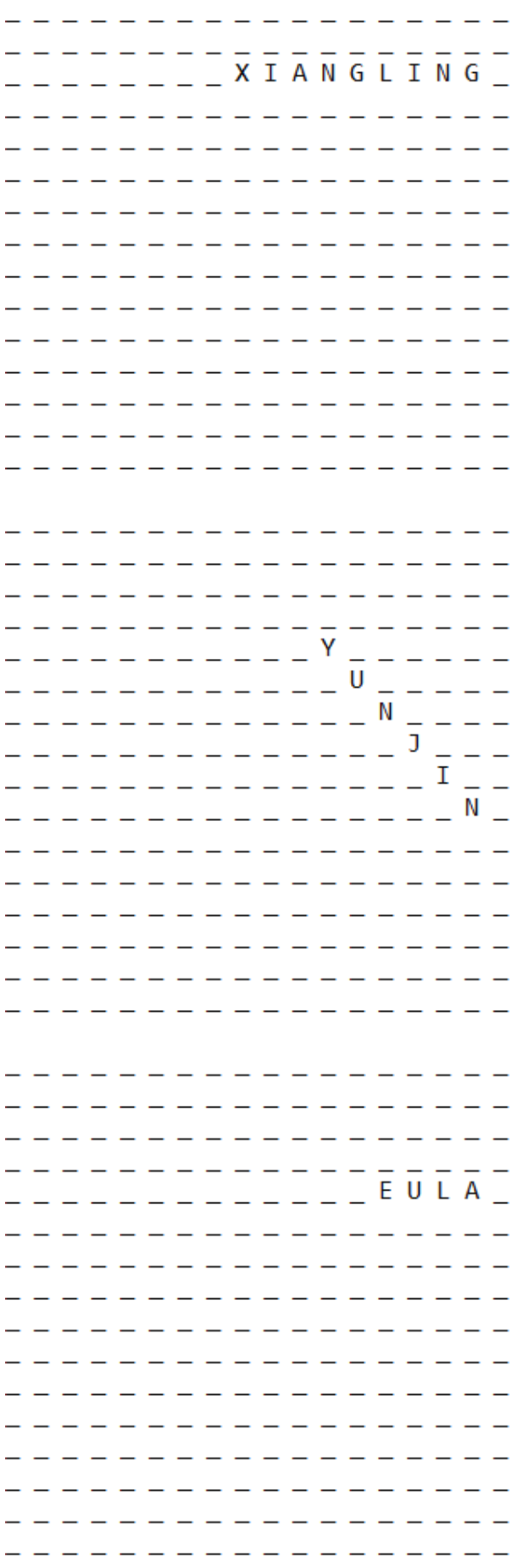
medium1.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): medium1.txt
_ _ _ _ B A R B A R A _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ N O E L L E _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

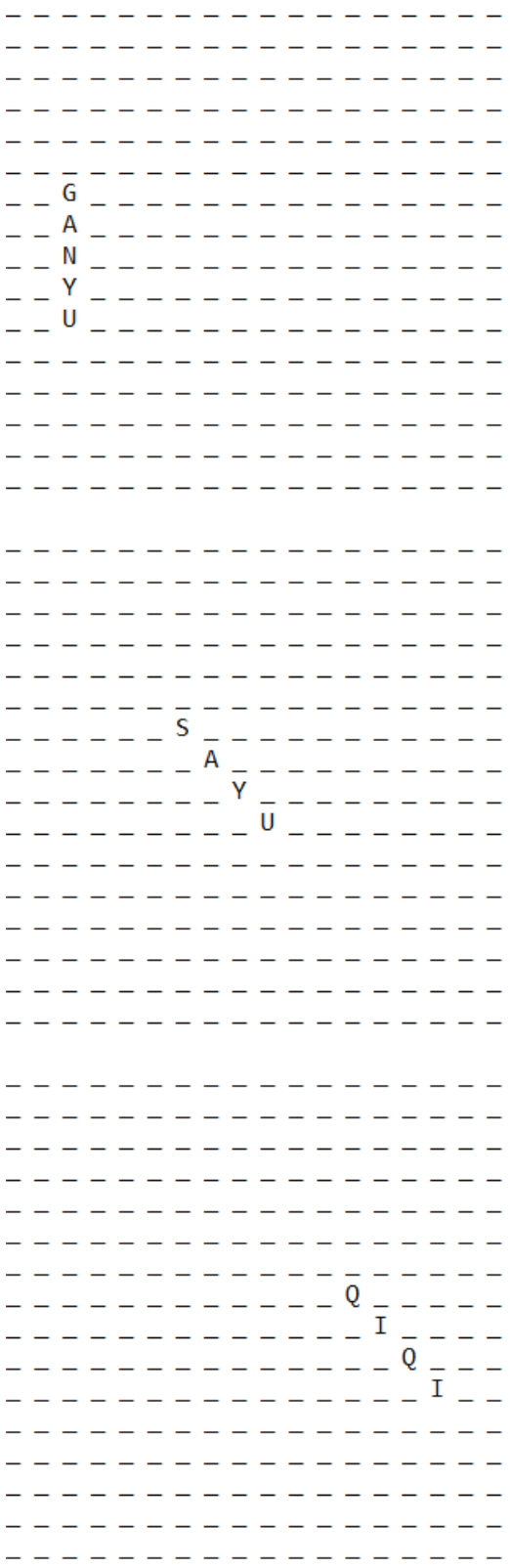

B
E
I
D
O
U
ROSARIA
Y
O
I
M
I
Y
A

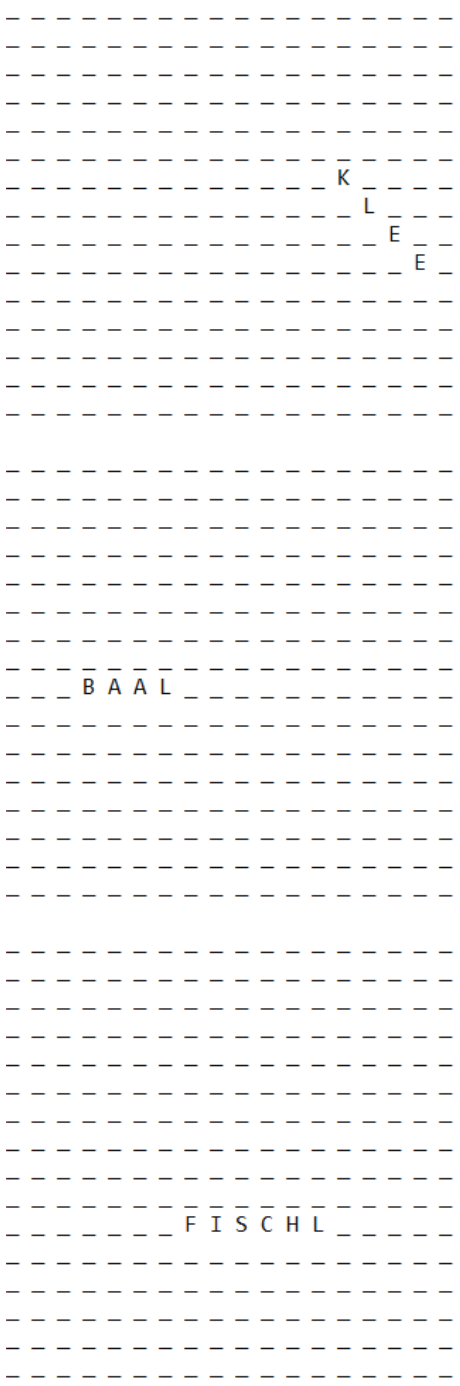
```
S
 U
  C
   R
    O
     S
      E


    N
     I
      N
       G
        G
         U
          A
           N
            G


                  H U T A O
```

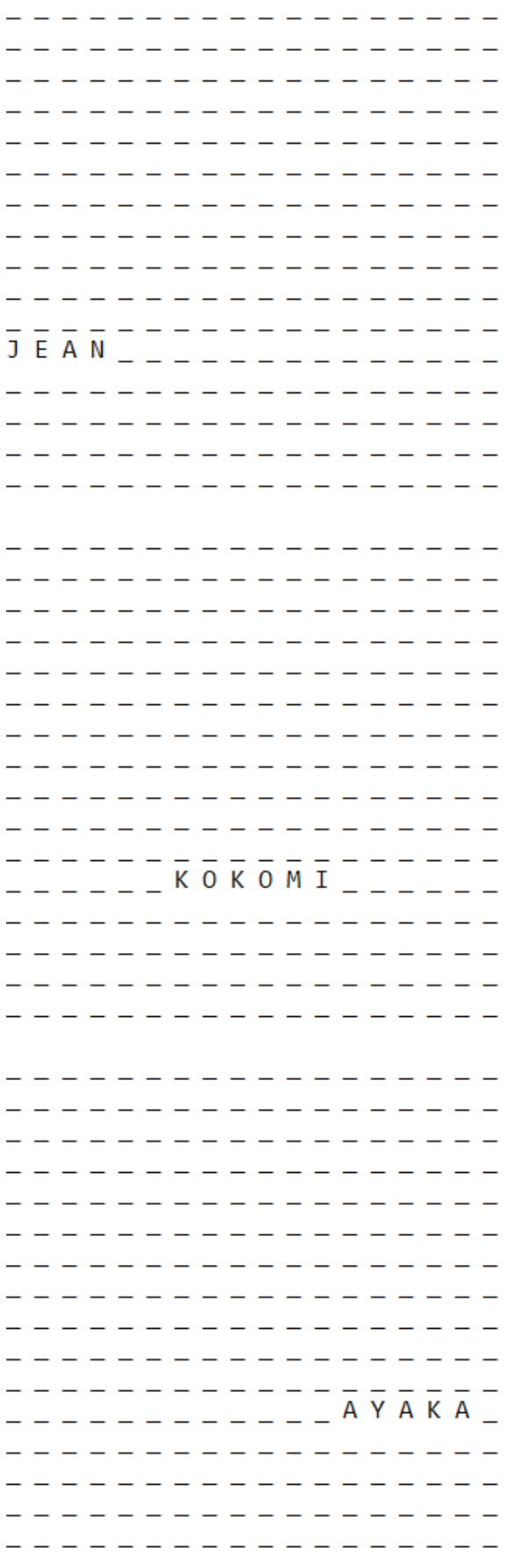

S
H
E
N
H
E
L
I
S
A
Y
A
N
F
E
I


XIANGLING
YUNJIN
EULA


KEQING
AMBER
DIONA


GANYU
SAYU
QIQI


K
L
E
E
B A A L
F I S C H L

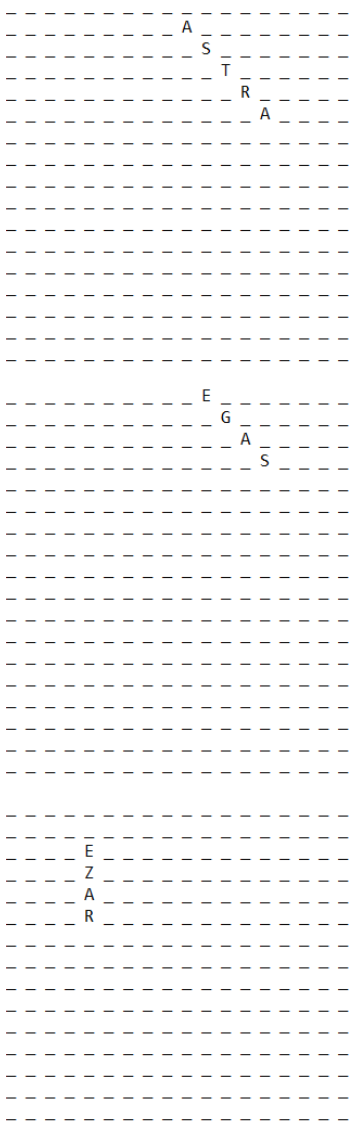

JEAN
KOKOMI
AYAKA

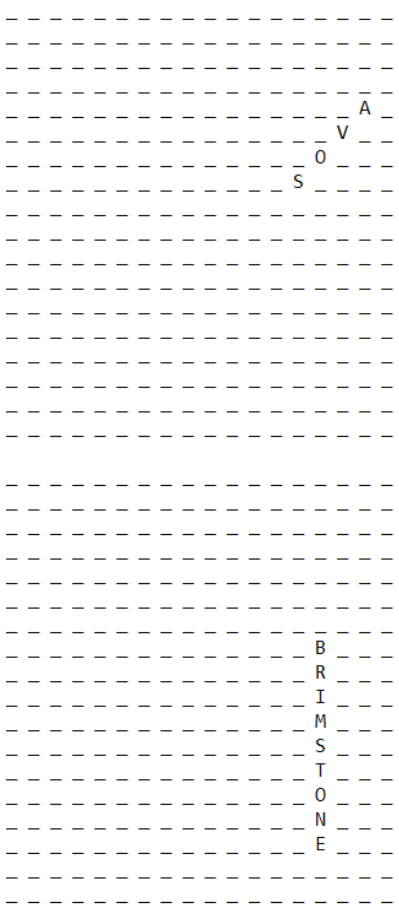
```
Time taken : 0 miliseconds
Checks done : 12820

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

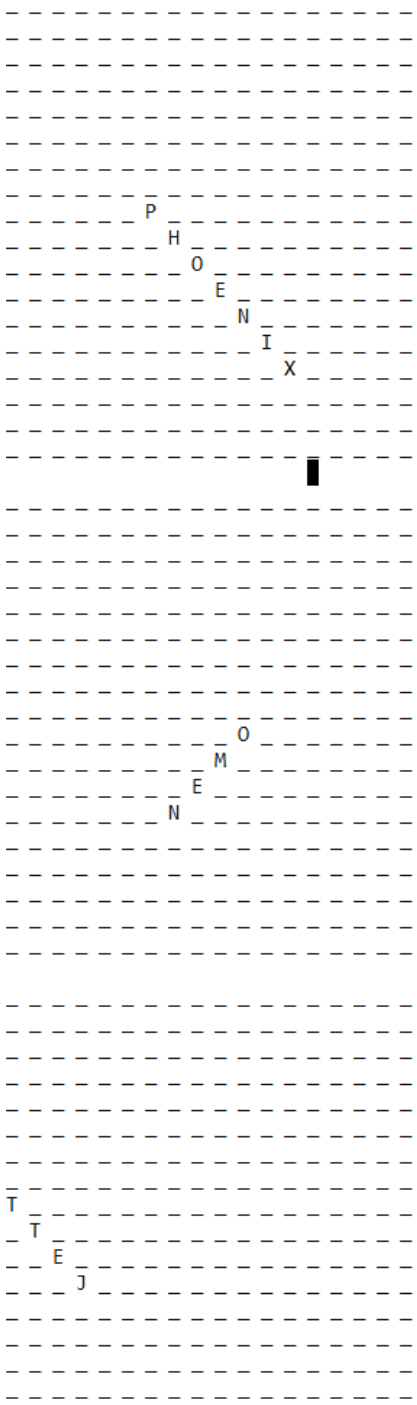
medium2.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): medium2.txt
_ _ S _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ K _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ Y _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ E _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ H C A E R B _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```


C
Y
P
H
E
R
YOJLLIK


A
S
T
R
A
E
G
A
S
E
Z
A
R

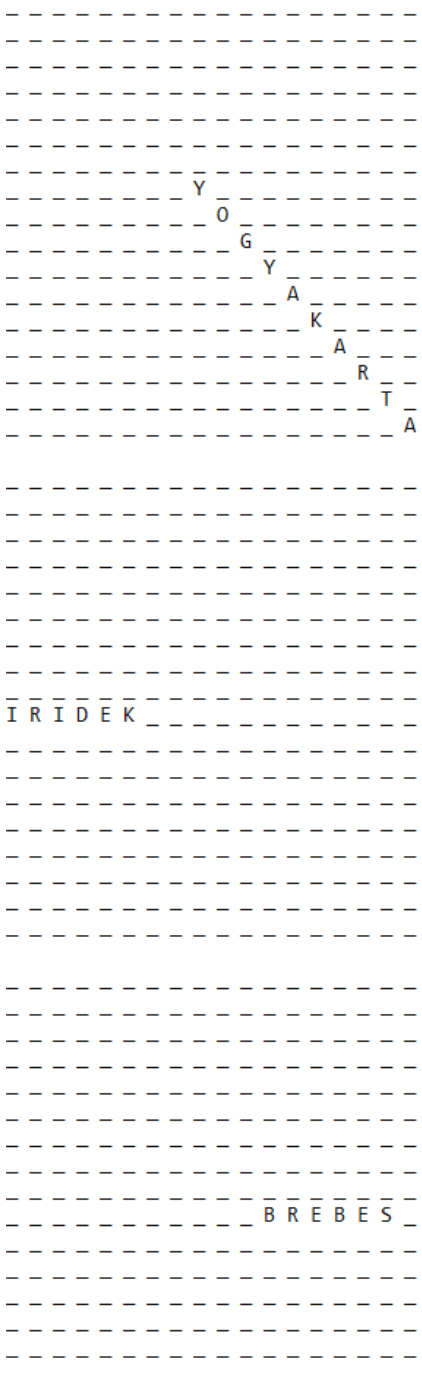

A
V
O
S
B
R
I
M
S
T
O
N
E

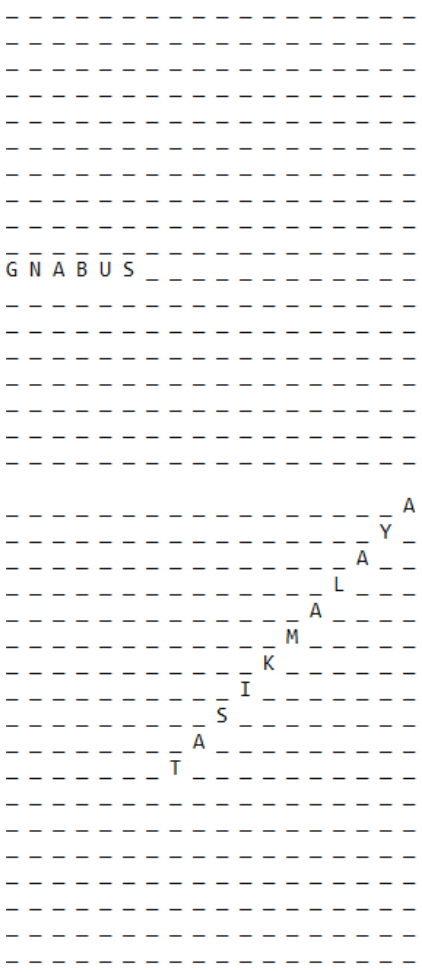


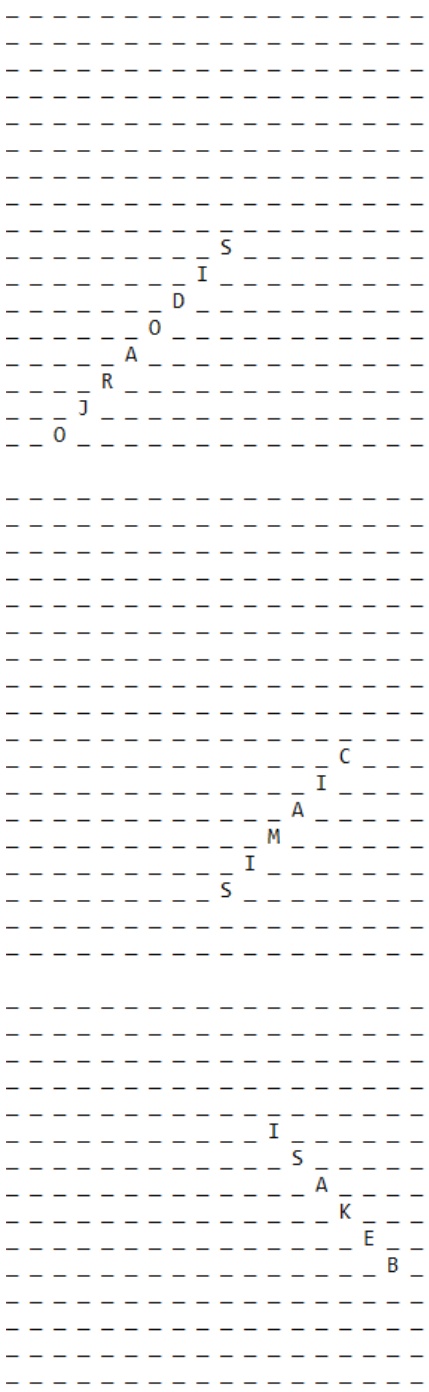
```
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ R E Y N A _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Time taken : 0 miliseconds
Checks done : 7465

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

medium3.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): medium3.txt
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ C _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ I _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ L _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ E _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ G _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ O _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ N _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ S _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ U _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ R _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ A _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ B _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ A _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ Y _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ A _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ N _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ U _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ I _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ D _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ A _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ M _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```


YOGYAKARTA
IRIDEK
BREBES


GNABUS
A
Y
A
L
A
M
K
I
S
A
T



Time taken : 1 miliseconds
Checks done : 13299

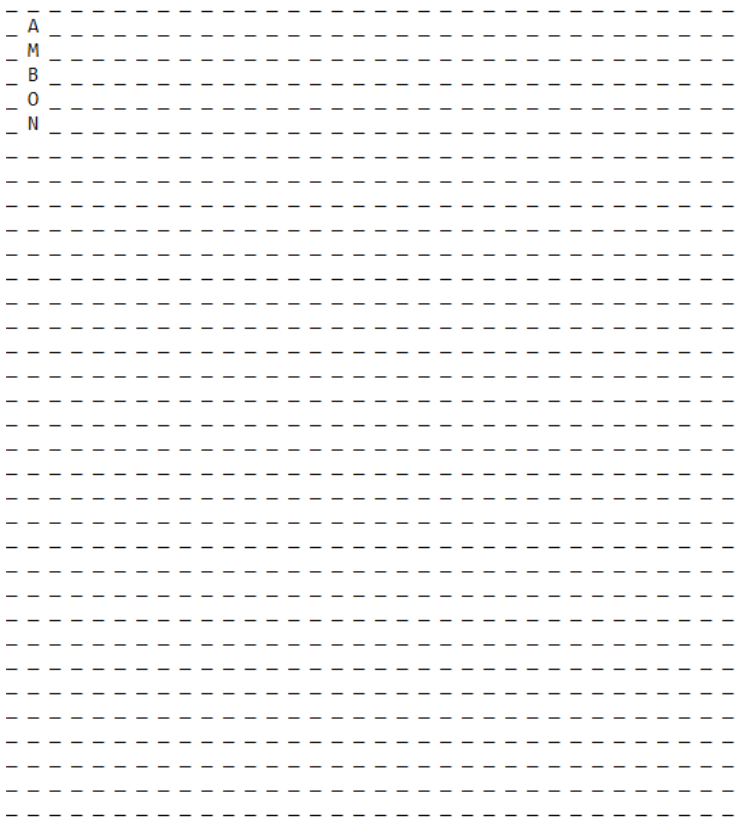
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>

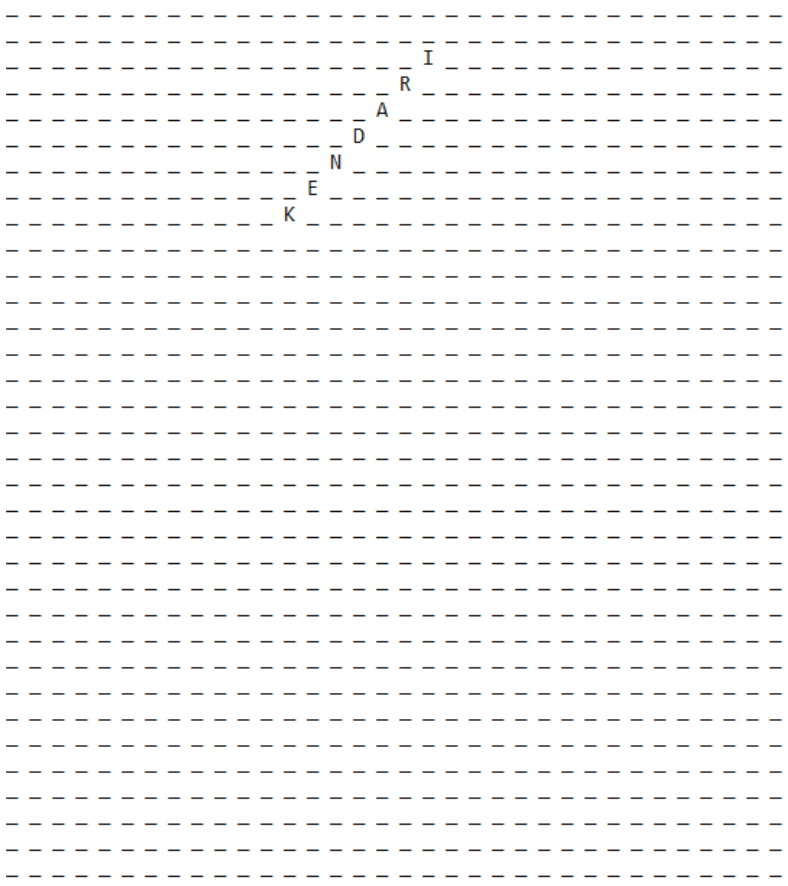
large1.txt

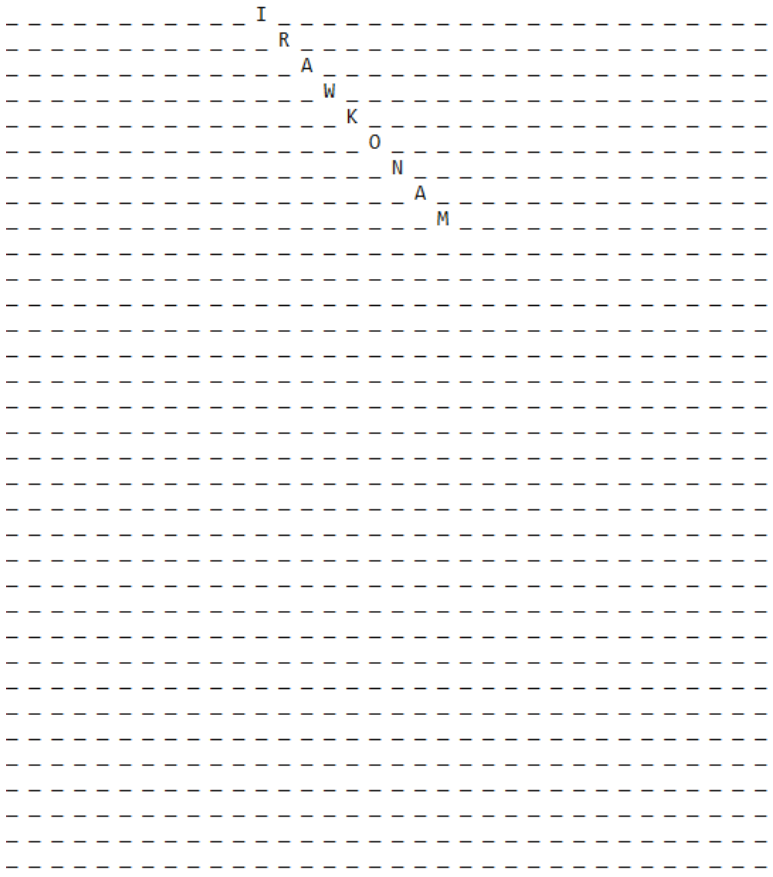
```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): large1.txt
_ B _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ A _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ N _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ J _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ A _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ R _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ M _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ A _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ S _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ I _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ N _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

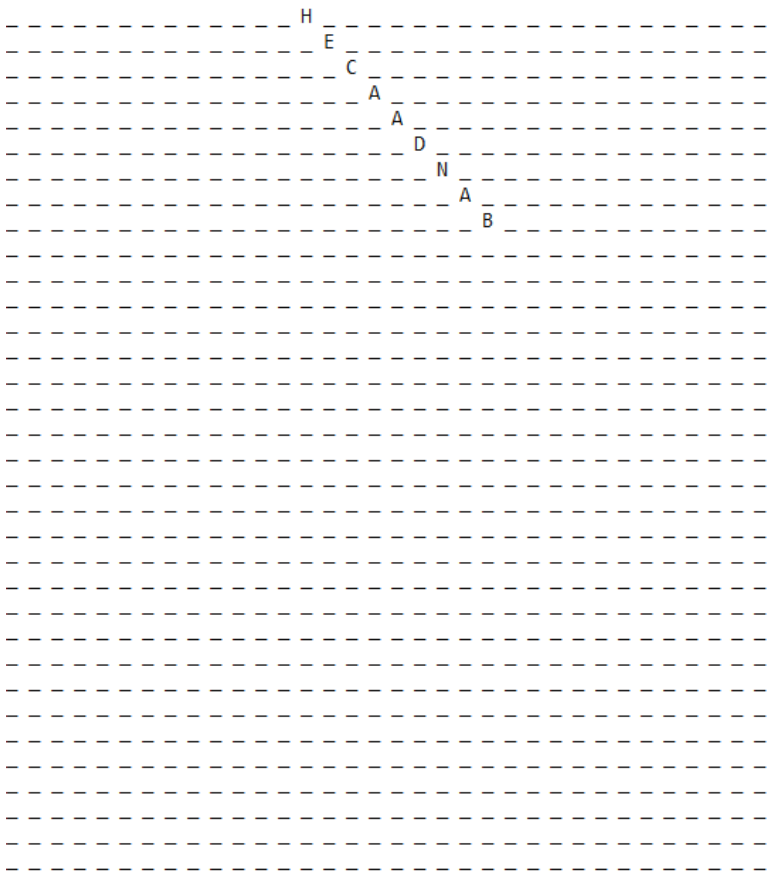

S
O
F
I
F
I


PALEMBANG


A
M
B
O
N


I
R
A
D
N
E
K

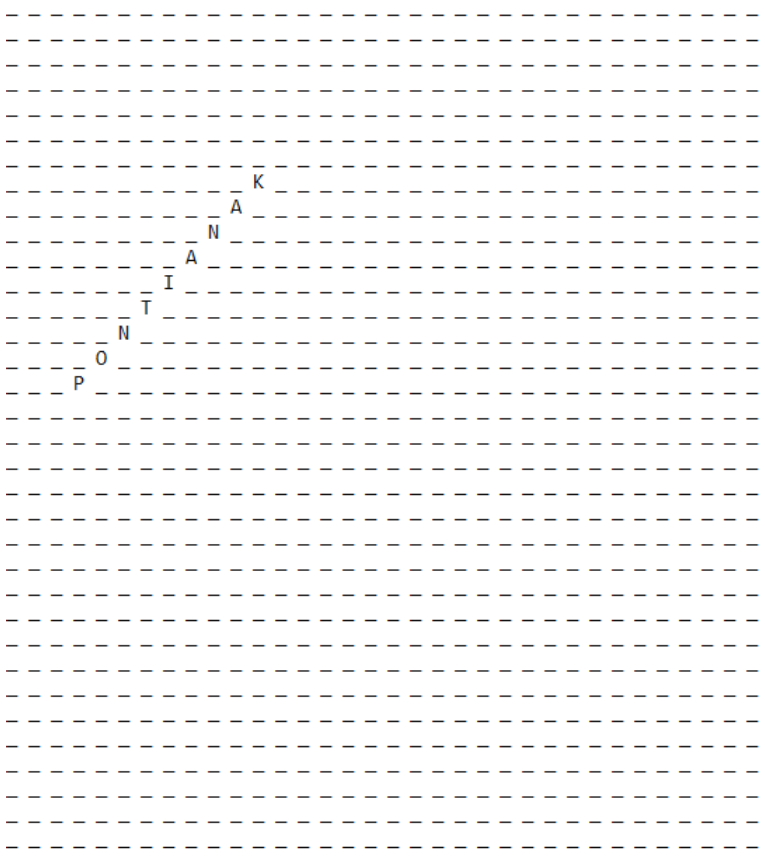

I
R
A
W
K
O
N
A
M


H
E
C
A
A
D
N
A
B

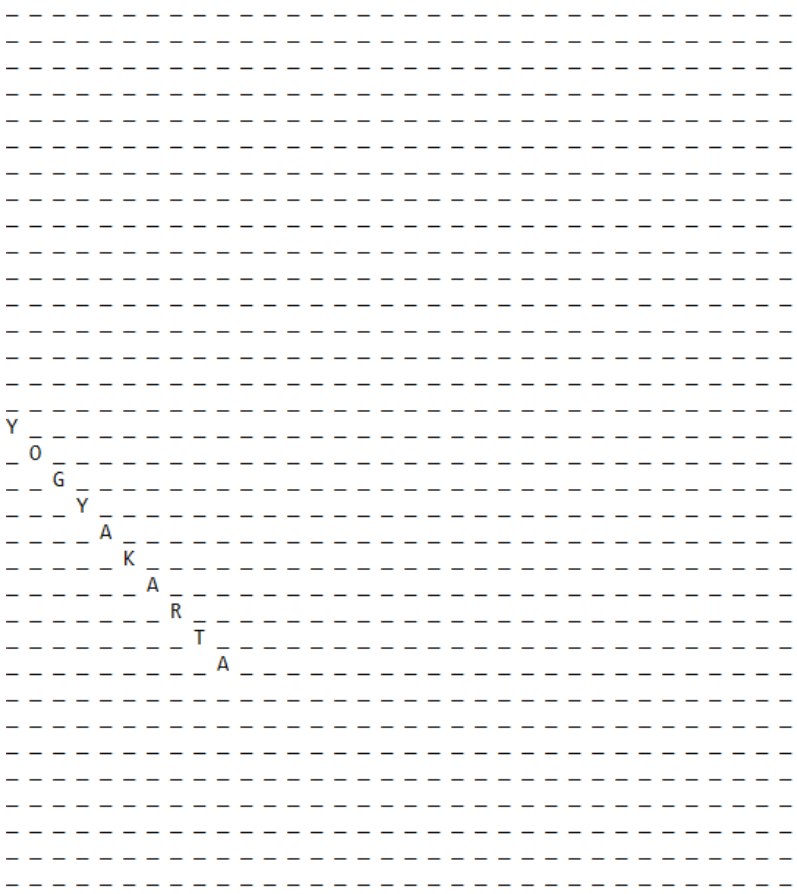
PEKANBARU


R
A
S
S
A
K
A
M


G
N
A
N
I
P
L
A
K
G
N
A
P

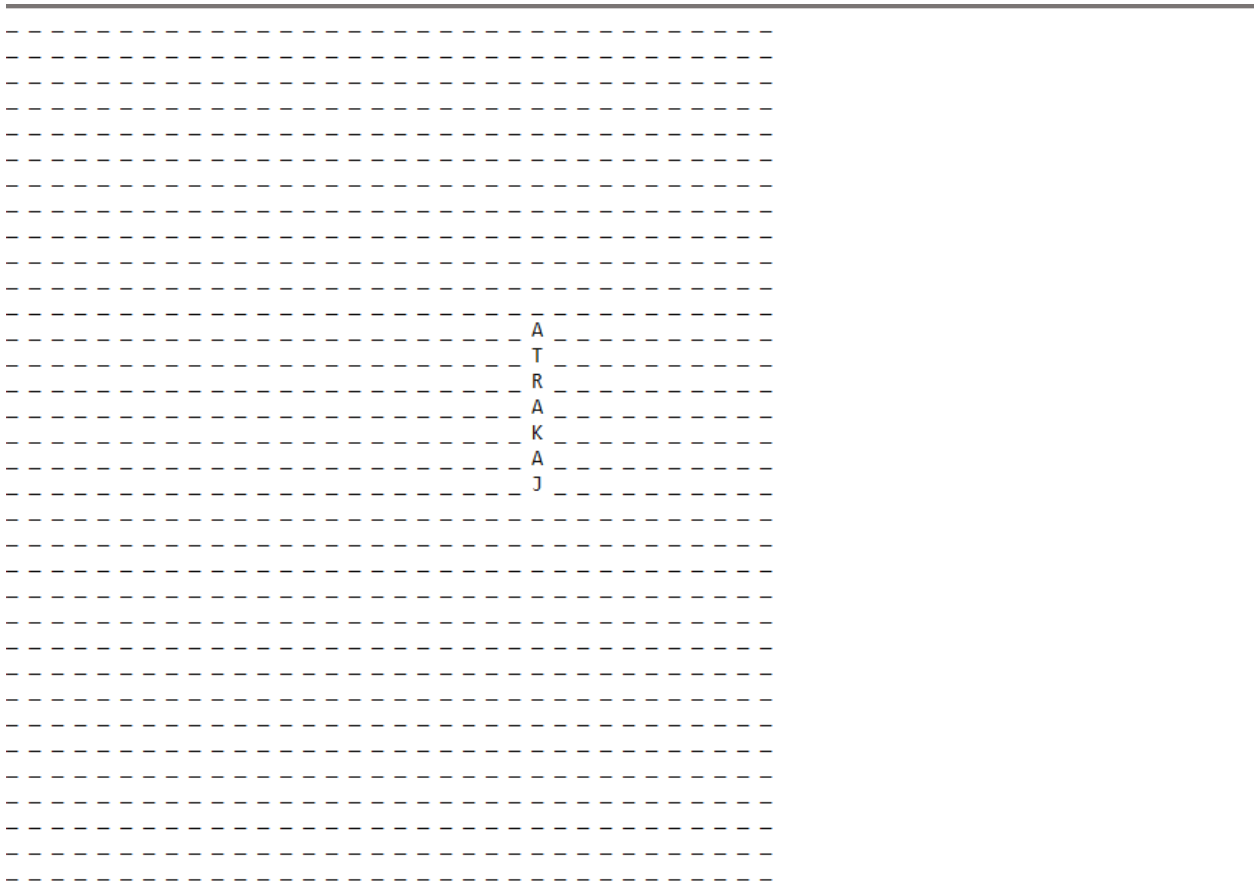

K
A
N
A
I
T
N
O
P

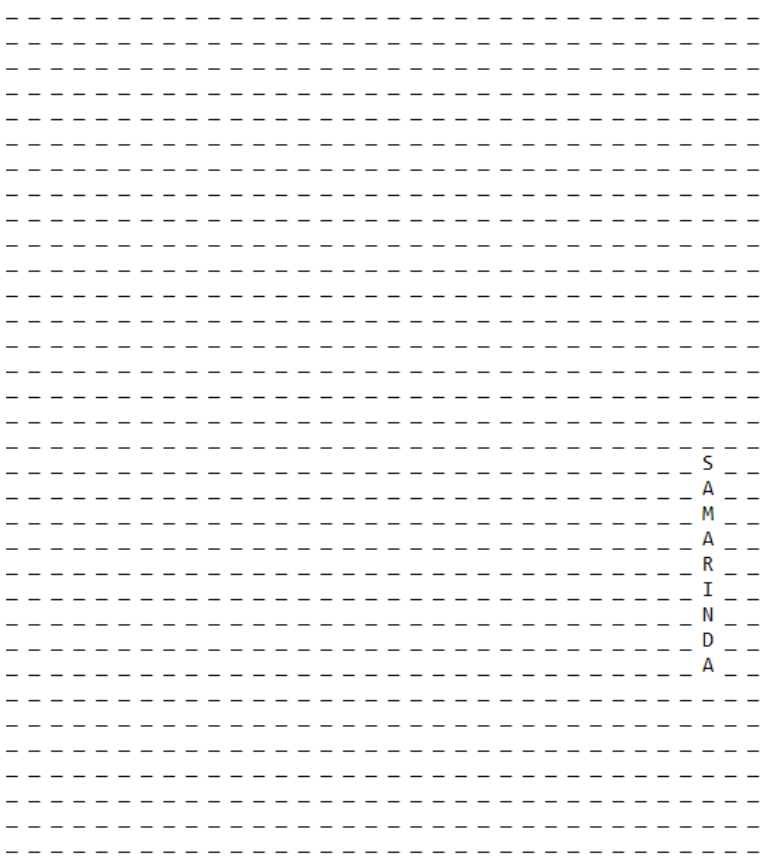

G
N
A
D
A
P


Y
O
G
Y
A
K
A
R
T
A

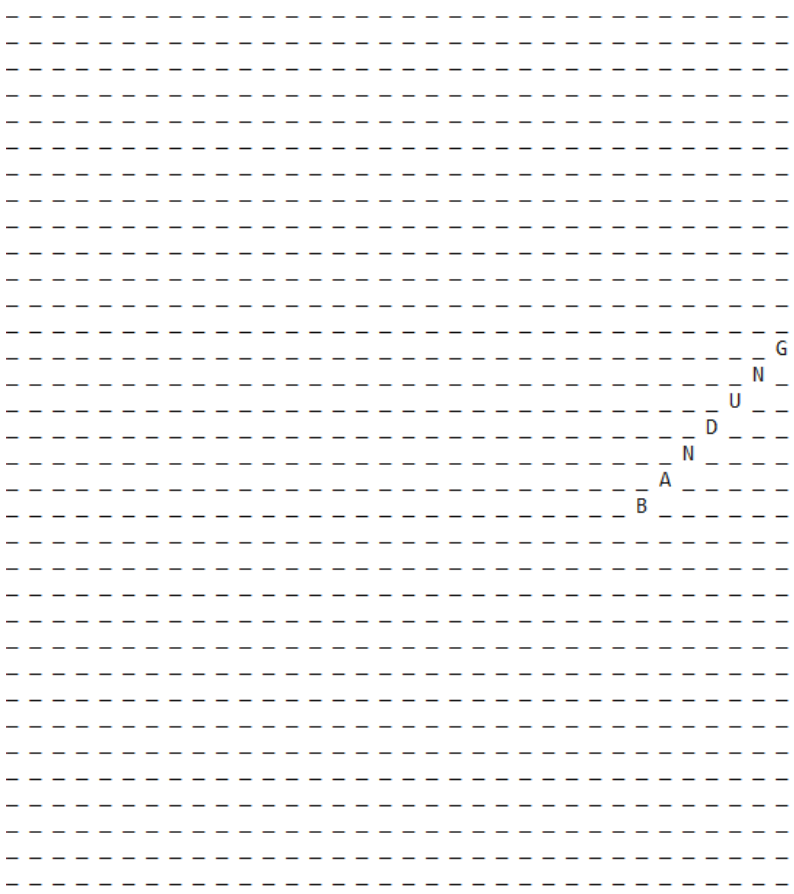

GORONTALO

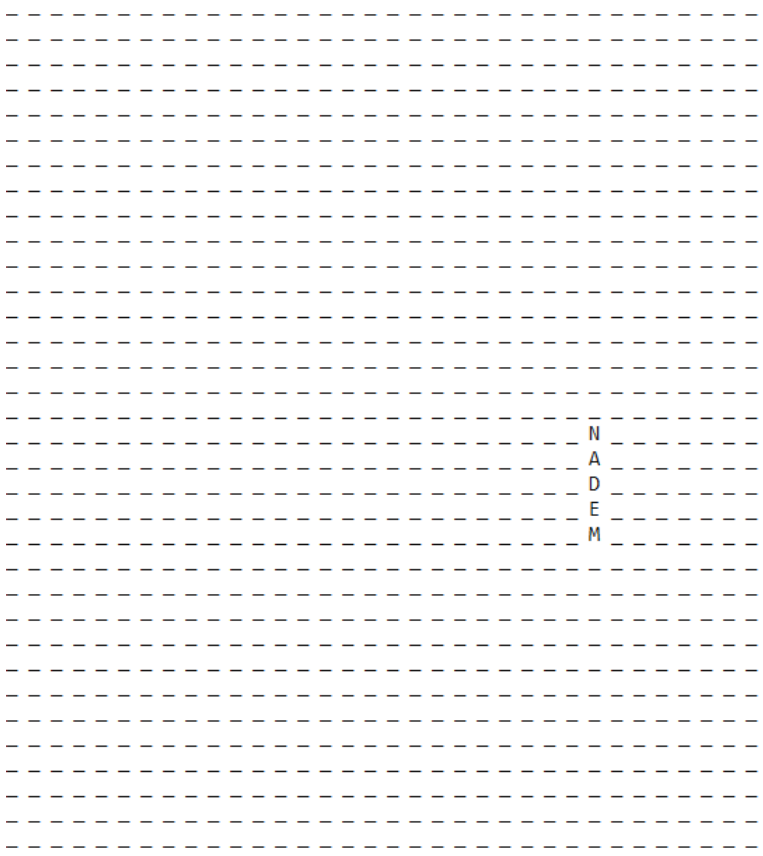

G
N
A
P
U
K


ATRAKAJ


SAMARINDA


SURABAYA

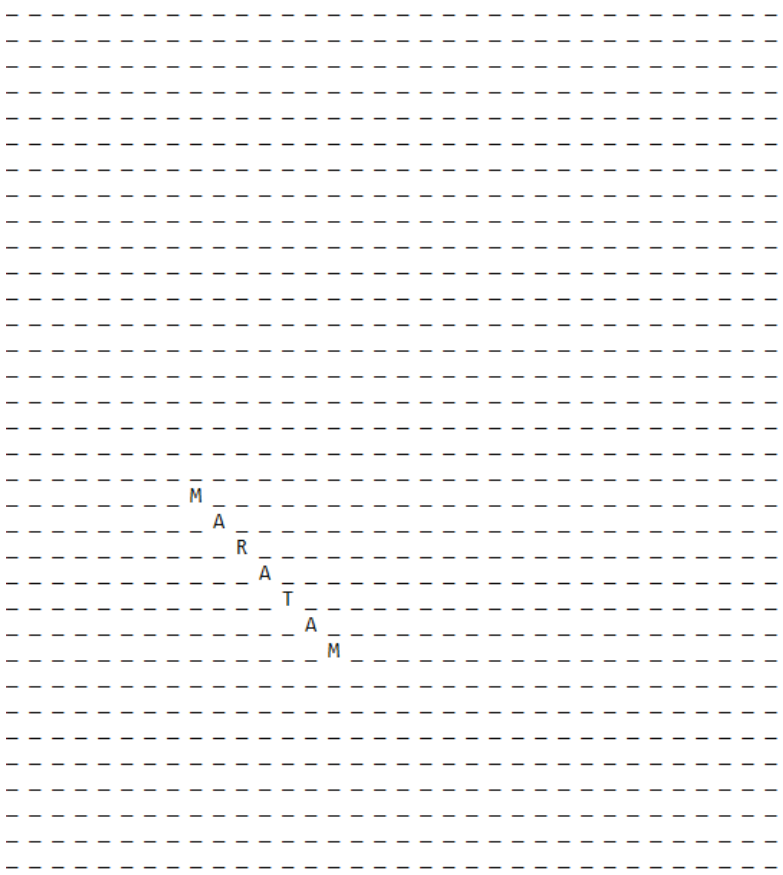

G
N
U
D
N
A
B


NADEM


T
A
N
J
U
N
G
S
E
L
O
R

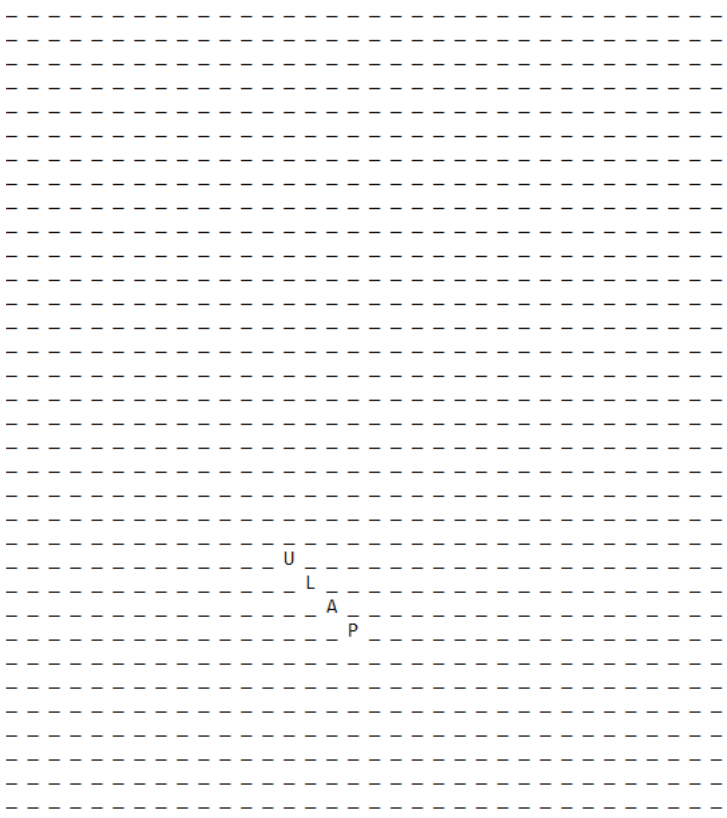

J
A
M
B
I

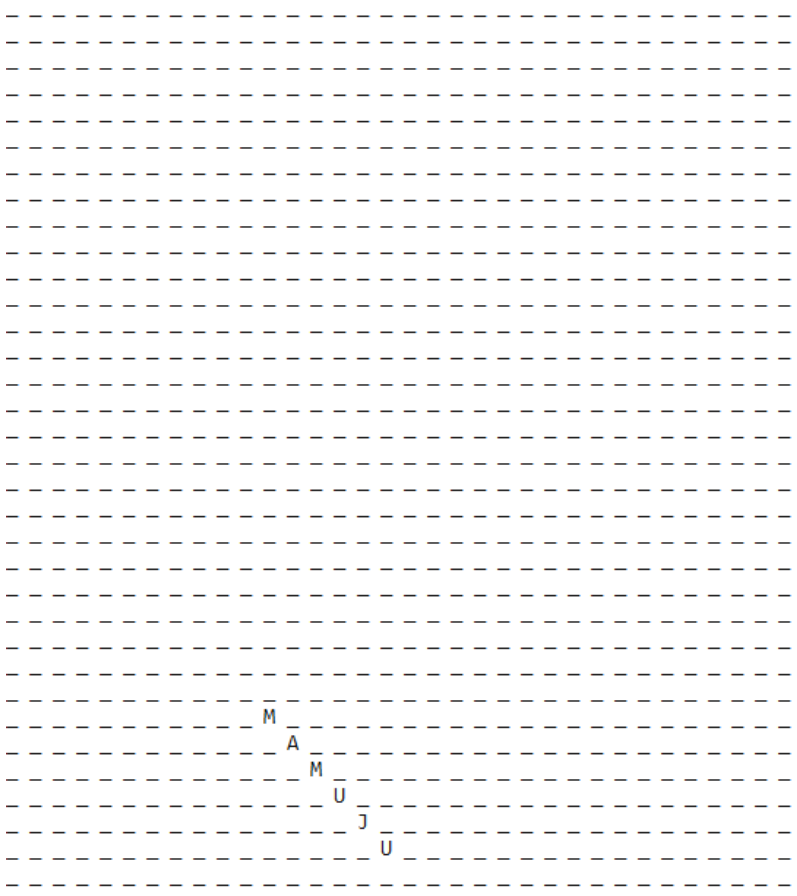

RASAPNED

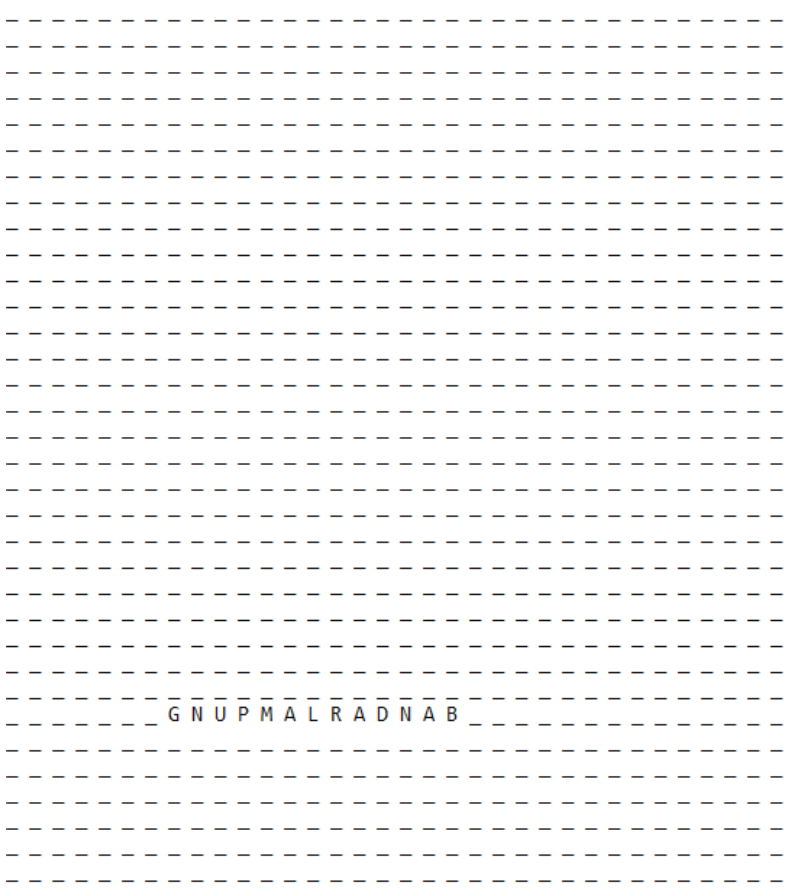

M
A
R
A
T
A
M

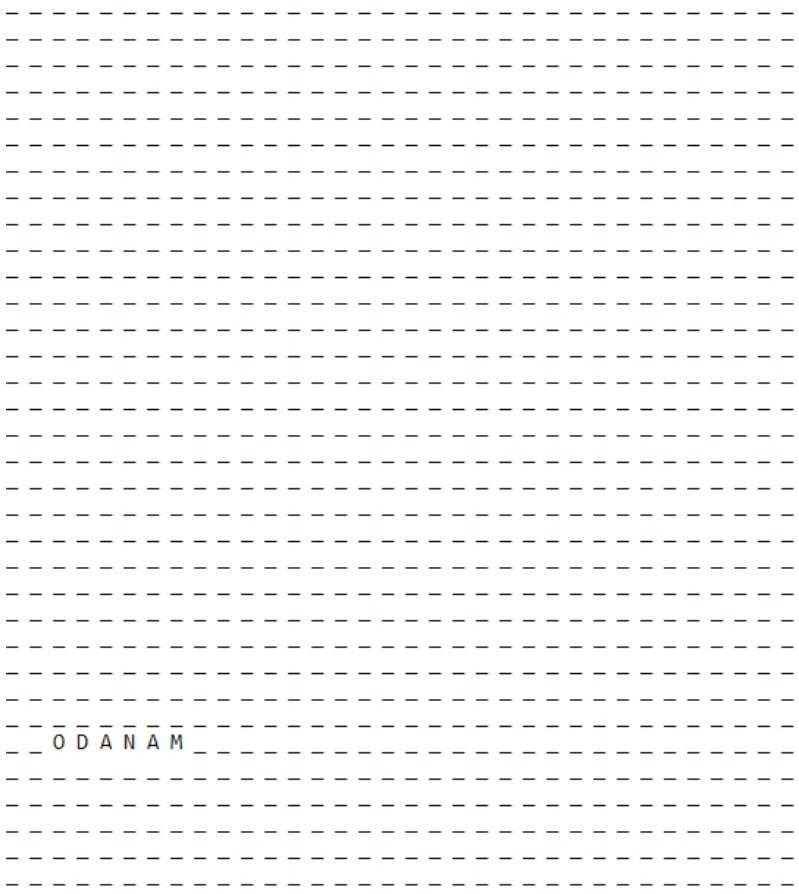

S
E
R
A
N
G


B
E
N
G
K
U
L
U


U
L
A
P


M
A
M
U
J
U


GNUPMALRADNAB


ODANAM


PALANGKARAYA


SEMARANG

```
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ A R U P A Y A J _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Time taken : 0 miliseconds
Checks done : 122352

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

spek.txt

```
C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>java Main
Input filename (with '.txt' on the end, with file on the 'test' folder): spek.txt
J _ _ _ _ _ _ _
_ U _ _ _ _ _ _
_ _ P _ _ _ _ _
_ _ _ I _ _ _ _
_ _ _ _ T _ _ _
_ _ _ _ _ E _ _
_ _ _ _ _ _ R _

_ _ _ _ _ _ _ S
_ _ _ _ _ _ _ A
_ _ _ _ _ _ _ T
_ _ _ _ _ _ _ U
_ _ _ _ _ _ _ R
_ _ _ _ _ _ _ N
_ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _
S U N A R U _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
N E P T U N E _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _

_ _ _ _ _ _ _ S
_ _ _ _ _ _ U _
_ _ _ _ _ N _ _
_ _ _ _ E _ _ _
_ _ _ V _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
```

```
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ H T R A E _ _
_ _ _ _ _ _ _ _

_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
S _ _ _ _ _ _ _
R _ _ _ _ _ _ _
A _ _ _ _ _ _ _
M _ _ _ _ _ _ _

_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ M E R C U R Y


Time taken : 0 miliseconds
Checks done : 586

C:\Lenovo Temp\IF Files\Sem4\Stima\Tucil1\wordsearchpuzzle\src>
```

Checklist

No.	Poin	Keberhasilan Poin
1	Program berhasil dikompilasi tanpa kesalahan (no syntax error)	☑
2	Program berhasil running	☑
3	Program dapat membaca file masukan dan menuliskan luaran.	☑
4	Program berhasil menemukan semua kata di dalam puzzle.	☑

Alamat drive kode program

https://drive.google.com/drive/folders/1BU6r3guKEJEUzpIcqJDAIchF1aRs1sw6?usp=sharing

https://github.com/Noxira/wordsearchpuzzle