

# Laporan Tugas Kecil Strategi Algoritma 2021/2022 Word Search Puzzle



Disusun Oleh:

Andreas Indra Kurniawan 13520091/K-1

## Langkah-Langkah Algoritma Brute Force

1. Program menerima masukan *file text* yang berisi matrix segiempat berisi huruf dan juga kata yang akan dicari pada matrix tersebut(matrix dan kata dipisahkan satu enter kosong).
2. Program akan memproses *file* dan membuat *object* untuk menampung informasi di atas lalu akan dijalankan timer untuk mengetahui waktu awal ketika program dijalankan.
3. Program akan melangkah dari matrix kiri atas mengecek tiap baris terlebih dahulu.
4. Untuk setiap huruf yang sedang dicek, akan dilakukan cek untuk setiap kata yang ada. Jika huruf pertama matrix cocok dengan huruf pertama kata maka akan dilakukan pengecekan lebih lanjut yaitu ke arah kanan, kiri, atas, bawah, dan diagonal(kanan bawah, kiri bawah, kiri atas, dan kanan atas). Jika dari huruf pertama sudah tidak cocok, maka akan langsung dilanjutkan ke pengecekan kata selanjutnya.
5. Ketika kata sudah cocok 100%, program akan menandai posisi kata menggunakan matrix angka yang besarnya sudah disesuaikan dengan besar matrix puzzle. Angka 8 di matrix angka ini berarti warna netral, sedangkan 0-7 merupakan warna dengan kode ansi yang sudah dipilih.
6. Setelah menandai posisi kata, matrix akan menampilkan jumlah perbandingan yang digunakan untuk mencari satu kata dan waktu yang digunakan sehingga kata didapatkan.
7. Ketika seluruh kata berhasil ditemukan, maka program akan menambahkan kode ansi ke matrix puzzle sesuai dengan matrix angka yang ditandai.
8. Program akan menampilkan matrix puzzle yang telah diwarnai beserta total perbandingan, total waktu, dan total kata yang ditemukan.

## Source Program Dalam Bahasa Java

File mainProg.java

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

class mainProg{
    public static void printPuzzle(String[][] puzzle,int width,int length){
        int i,j;
        for(i=0;i<length;i++){
            for(j=0;j<width;j++){
                System.out.print(puzzle[i][j] + " ");
            }
            System.out.println();
        }
        System.out.print("\u001B[0m");
    }

    public static void printMatColor(puzzle objP){
        int i,j;
        for(i=0;i<objP.length;i++){
            for(j=0;j<objP.width;j++){
                System.out.print(objP.getIsiMatPC(i, j)+" ");
            }
            System.out.println();
        }
    }
}
```

```

        System.out.println();
    }
    public static void printWords(String[][] words, int length){
        int i,j;
        for(i=0;i<length;i++){
            int wordLen = words[i].length;
            for(j=0;j<wordLen;j++){
                System.out.print(words[i][j]);
            }
            System.out.println();
        }
    }

    public static void printWord(String[] word){
        int i;
        for(i=0;i<word.length;i++){
            System.out.print(word[i]);
        }
        System.out.println();
    }

    public static String arrToString(String[] word){
        int i;
        String x = "";
        for(i=0;i<word.length;i++){
            x += word[i];
        }
        return x;
    }

    public static void initDash(String[][] dash,int length, int width){
        int i,j;
        for(i=0;i<length;i++){
            for(j=0;j<width;j++){
                dash[i][j] = "-";
            }
        }
    }

    public static puzzle checkR(puzzle objP, String[] word ,int xAwal, int
yAwal, int width,int k){        //Mengecek kata di bagian horizontal
        boolean depanMatch = true;
        int i=1;
        int xUp = xAwal;
        int yUp = yAwal+1;
        objP.setStatus(false);
        while(depanMatch && i<word.length && yUp<width){
            objP.incrArrCount(k);

```

```

        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
        i++;
        yUp++;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle checkL(puzzle objP, String[] word ,int xAwal, int
yAwal, int width, int k){          //Mengecek kata di bagian horizontal ke kiri
    boolean depanMatch = true;
    int i=1;
    int xUp = xAwal;
    int yUp = yAwal-1;
    objP.setStatus(false);
    while(depanMatch && i<word.length && yUp>=0){
        objP.incrArrCount(k);
        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
        i++;
        yUp--;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle checkT(puzzle objP, String[] word ,int xAwal, int
yAwal, int width, int k){          //Mengecek kata di bagian vertikal ke atas
    boolean depanMatch = true;
    int i=1;
    int xUp = xAwal-1;
    int yUp = yAwal;
    objP.setStatus(false);
    while(depanMatch && i<word.length && xUp>=0){
        objP.incrArrCount(k);
        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
    }
}

```

```

        i++;
        xUp--;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

    public static puzzle checkB(puzzle objP, String[] word ,int xAwal, int
yAwal, int width,int k){        //Mengecek kata di bagian vertikal ke bawah
        boolean depanMatch = true;
        int i=1;
        int xUp = xAwal+1;
        int yUp = yAwal;
        objP.setStatus(false);
        while(depanMatch && i<word.length && xUp<objP.length){
            objP.incrArrCount(k);
            if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
                depanMatch = false;
                return objP;
            }
            i++;
            xUp++;
        }
        if(i==word.length){
            objP.setStatus(true);
        }
        return objP;
    }

    public static puzzle checkBR(puzzle objP, String[] word ,int xAwal, int
yAwal, int width, int k){        //Mengecek kata di bagian diagonal ke
tenggara
        boolean depanMatch = true;
        int i=1;
        int xUp = xAwal+1;
        int yUp = yAwal+1;
        objP.setStatus(false);
        while(depanMatch && i<word.length && xUp<objP.length &&
yUp<objP.width){
            if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
                objP.incrArrCount(k);
                depanMatch = false;
                return objP;
            }
            i++;
            xUp++;

```

```

        yUp++;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle checkBL(puzzle objP, String[] word ,int xAwal, int
yAwal, int width,int k){          //Mengecek kata di bagian diagonal ke barat
daya
    boolean depanMatch = true;
    int i=1;
    int xUp = xAwal+1;
    int yUp = yAwal-1;
    objP.setStatus(false);
    while(depanMatch && i<word.length && xUp<objP.length && yUp>-1){
        objP.incrArrCount(k);
        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
        i++;
        xUp++;
        yUp--;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle checkTL(puzzle objP, String[] word ,int xAwal, int
yAwal, int width, int k){          //Mengecek kata di bagian diagonal ke barat
laut
    boolean depanMatch = true;
    int i=1;
    int xUp = xAwal-1;
    int yUp = yAwal-1;
    objP.setStatus(false);
    while(depanMatch && i<word.length && xUp>-1 && yUp>-1){
        objP.incrArrCount(k);
        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
        i++;
        xUp--;
    }

```

```

        yUp--;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle checkTR(puzzle objP, String[] word ,int xAwal, int
yAwal, int width, int k){          //Mengecek kata di bagian diagonal ke timur
laut
    boolean depanMatch = true;
    int i=1;
    int xUp = xAwal-1;
    int yUp = yAwal+1;
    objP.setStatus(false);
    while(depanMatch && i<word.length && xUp>-1 && yUp<objP.width){
        objP.incrArrCount(k);
        if(!objP.getIsiMatP(xUp, yUp).equals(word[i])){
            depanMatch = false;
            return objP;
        }
        i++;
        xUp--;
        yUp++;
    }
    if(i==word.length){
        objP.setStatus(true);
    }
    return objP;
}

public static puzzle markR(puzzle objP, int xAwal, int yAwal, int
wordLen){          //Menandai matrix warna ke arah kanan
    int j;
    for(j=yAwal;j<wordLen+yAwal;j++){
        objP.setMatPC(xAwal, j, objP.getColorCode());
    }
    objP.updateColor();
    return objP;
}

public static puzzle markL(puzzle objP, int xAwal, int yAwal, int
wordLen){
    int j;
    for(j=yAwal;j>yAwal-wordLen;j--){
        objP.setMatPC(xAwal, j, objP.getColorCode());
    }
}

```

```

        objP.updateColor();
        return objP;
    }

    public static puzzle markT(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        for(i=xAwal;i>xAwal-wordLen;i--){
            objP.setMatPC(i, yAwal, objP.getColorCode());
        }
        objP.updateColor();
        return objP;
    }

    public static puzzle markB(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        for(i=xAwal;i<xAwal+wordLen;i++){
            objP.setMatPC(i, yAwal, objP.getColorCode());
        }
        objP.updateColor();
        return objP;
    }

    public static puzzle markBR(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        int j = yAwal;
        for(i=xAwal;i<xAwal+wordLen;i++){
            objP.setMatPC(i, j, objP.getColorCode());
            j++;
        }
        objP.updateColor();
        return objP;
    }

    public static puzzle markBL(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        int j = yAwal;
        for(i=xAwal;i<xAwal+wordLen;i++){
            objP.setMatPC(i, j, objP.getColorCode());
            j--;
        }
        objP.updateColor();
        return objP;
    }
}

```



```

    public static puzzle markTL(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        int j = yAwal;
        for(i=xAwal;i>xAwal-wordLen;i--){
            objP.setMatPC(i, j, objP.getColorCode());
            j--;
        }
        objP.updateColor();
        return objP;
    }

    public static puzzle markTR(puzzle objP, int xAwal, int yAwal, int
wordLen){
        int i;
        int j = yAwal;
        for(i=xAwal;i>xAwal-wordLen;i--){
            objP.setMatPC(i, j, objP.getColorCode());
            j++;
        }
        objP.updateColor();
        return objP;
    }

    public static puzzle color(puzzle
objP){
        //Mewarnain matrix puzzle sesuai
        warna dari matrix color
        int i,j;
        boolean color = false;
        for(i=0;i<objP.length;i++){
            for(j=0;j<objP.width;j++){
                if(objP.getIsiMatPC(i, j) != 8){
                    color = true;
                }else{
                    color = false;
                }
                if(color){
                    objP.setColorCode(objP.getIsiMatPC(i, j));
                    objP.setMatP(i, j, objP.getColor()+objP.getIsiMatP(i, j));
                }else{
                    objP.setMatP(i, j, objP.getResetC()+objP.getIsiMatP(i,
j));
                }
            }
        }
        return objP;
    }
}

```

```

    public static puzzle checkPuzzle(puzzle pObj, int k, int i, int j){
        if(!pObj.getArrFound(k) && pObj.getIsiMatP(i,
j).equals(pObj.getLetter(k, 0))){
            pObj.incrArrCount(k);
            pObj = checkR(pObj, pObj.getWord(k),i,j,pObj.width, k);
            if(pObj.getStatus()){
                // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
                System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") mendatar ke kanan dalam :");
                System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
                markR(pObj, i, j, pObj.getWord(k).length);
                pObj.incrWordGot();
                pObj.setArrFound(k, true);
            }
            pObj = checkL(pObj, pObj.getWord(k), i, j, pObj.width, k);
            if(pObj.getStatus()){
                // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
                System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") mendatar ke kiri dalam :");
                System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
                markL(pObj, i, j, pObj.getWord(k).length);
                pObj.incrWordGot();
                pObj.setArrFound(k, true);
            }
            pObj = checkT(pObj, pObj.getWord(k), i, j, pObj.width, k);
            if(pObj.getStatus()){
                // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
                System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") vertikal ke atas dalam :");
                System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
                markT(pObj, i, j, pObj.getWord(k).length);
                pObj.incrWordGot();
                pObj.setArrFound(k, true);
            }
            pObj = checkB(pObj, pObj.getWord(k), i, j, pObj.width, k);
            if(pObj.getStatus()){
                // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
                System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") vertikal ke bawah dalam :");
                System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");

```

```

        markB(pObj, i, j, pObj.getWord(k).length);
        pObj.incrWordGot();
        pObj.setArrFound(k, true);
    }
    pObj = checkBR(pObj, pObj.getWord(k), i, j, pObj.width, k);
    if(pObj.getStatus()){
        // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
        System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") diagonal ke kanan bawah
dalam :");
        System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
        markBR(pObj, i, j, pObj.getWord(k).length);
        pObj.incrWordGot();
        pObj.setArrFound(k, true);
    }
    pObj = checkBL(pObj, pObj.getWord(k), i, j, pObj.width, k);
    if(pObj.getStatus()){
        // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
        System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") diagonal ke kiri bawah
dalam :");
        System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
        markBL(pObj, i, j, pObj.getWord(k).length);
        pObj.incrWordGot();
        pObj.setArrFound(k, true);
    }
    pObj = checkTL(pObj, pObj.getWord(k), i, j, pObj.width, k);
    if(pObj.getStatus()){
        // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);
        System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") diagonal ke kiri atas
dalam :");
        System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
        markTL(pObj, i, j, pObj.getWord(k).length);
        pObj.incrWordGot();
        pObj.setArrFound(k, true);
    }
    pObj = checkTR(pObj, pObj.getWord(k), i, j, pObj.width, k);
    if(pObj.getStatus()){
        // printR(pObj.getMatP(), matrixDash, i, j, pObj.getWord(k),
width, length);

```

```

        System.out.println("Kata " + arrToString(pObj.getWord(k)) + "
ditemukan pada koordinat (" + i + "," + j + ") diagonal ke kanan atas
dalam :");

        System.out.println(pObj.getArrCount(k) + " perbandingan dan
waktu " + pObj.getTime() + "ms\n");
        markTR(pObj, i, j, pObj.getWord(k).length);
        pObj.incrWordGot();
        pObj.setArrFound(k, true);
    }
}

return pObj;
};

public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    String filePath = sc.next();
    String path = System.getProperty("user.dir");
    path = path + "\\..\test\\" + filePath;
    System.out.println(path);
    puzzle pObj = new puzzle();

    int i,j;
    i = 0; j = 0;
    int length = 0;
    int width = 0;
    int countWords = 0;
    try{
        File file = new File(path);
        Scanner readData = new Scanner(file);
        Boolean puzzEnd = false;
        while(readData.hasNextLine()){
            String lineData = readData.nextLine();
            if(lineData.isEmpty()){
                puzzEnd = true;
                lineData = readData.nextLine();
            }
            if(!puzzEnd){
                String[] row = lineData.split(" ",0);
                for(j = 0;j<row.length;j++){
                    pObj.setMatP(i,j,row[j]);
                    width = j+1;
                }
                length++;
                i++;
            }else{
                pObj.setWord(countWords, lineData.split("",0)) ;
                countWords++;
            }
        }
    }
}

```

```

    }
    pObj.setCountW(countWords);
    pObj.setLength(length);
    pObj.setWidth(width);
    pObj.initMatPC();
    pObj.initArrCount();
    pObj.initArrFound();
    readData.close();
} catch (FileNotFoundException exception){
    System.out.println("File tidak ditemukan");
}

int k;
pObj.startTimer();
for(i=0;i<length;i++){
    for(j=0;j<width;j++){
        for(k=0;k<countWords;k++){
            pObj = checkPuzzle(pObj, k, i, j);
        }
    }
}
long endTime = pObj.getTime();
color(pObj);
printPuzzle(pObj.getMatP(), pObj.getWidth(), pObj.getLength());
System.out.println("\n" + "Total perbandingan yang dilakukan:");
System.out.println(pObj.getCountSum());
System.out.println("Total waktu yang digunakan:");
System.out.println(endTime + "ms");
System.out.println("Total kata yang ditemukan :");
System.out.println(pObj.getWordGot());
sc.close();
}
}

```

## File puzzle.java

```
public class puzzle {
    String[][] matrixPuzzle = new String[100][100];
    String[][] matrixWord = new String[100][100];
    int[][] matrixPuzzleC = new int[100][100];
    int[] arrayCount = new int[100];
    boolean[] arrayFound = new boolean[100];
    String[] arrayColor = new String[6];
    long startTime;
    int countWords = 0;
    int length;
    int width;
    int nthColor = 0;
    int getCountSum = 0;
    int wordGot = 0;
    boolean found = false;

    public static final String ANSI_RED = "\u001B[31m";
    public static final String ANSI_YELLOW = "\u001B[33m";
    public static final String ANSI_GREEN = "\u001B[32m";
    public static final String ANSI_BLUE = "\u001B[34m";
    public static final String ANSI_CYAN = "\u001B[36m";
    public static final String ANSI_MAGENTA = "\u001B[35m";
    public static final String ANSI_BRED = "\033[38;5;213m";
    public static final String ANSI_BBLUE = "\033[38;5;122m";
    public static final String ANSI_RESET = "\u001B[0m";

    // public void setArrC(){
    //     this.arrayColor[0] = "\u001B[31m"; //Red
    //     this.arrayColor[1] = "\u001B[33m"; //Yellow
    //     this.arrayColor[2] = "\u001B[32m"; //Green
    //     this.arrayColor[3] = "\u001B[34m"; //Blue
    //     this.arrayColor[4] = "\u001B[37m"; //White
    //     this.arrayColor[5] = "\u001B[0m"; //reset
    // }

    public void initArrFound(){
        int i;
        for(i=0;i<countWords;i++){
            this.arrayFound[i] = false;
        }
    }

    public void setArrFound(int i, boolean x){
        this.arrayFound[i] = x;
    }

    public boolean getArrFound(int i){
```

```

        return this.arrayFound[i];
    }

    public void incrWordGot(){
        this.wordGot++;
    }

    public int getWordGot(){
        return this.wordGot;
    }

    public void startTimer(){
        this.startTime = System.nanoTime();
    }

    public long getTime(){
        return (System.nanoTime()-this.startTime)/1000000;
    }

    public void setArrCount(int i, int x){
        this.arrayCount[i] = x;
    }

    public void incrGetCount(int x){
        this.getCountSum += x;
    }

    public int getArrCount(int i){
        return this.arrayCount[i];
    }

    public void incrArrCount(int i){
        this.arrayCount[i]++;
    }

    public int getCountSum(){
        int i;
        int sum = 0;
        for(i=0;i<this.arrayCount.length;i++){
            sum += this.arrayCount[i];
        }
        return sum;
    }

    public void initArrCount(){
        int i;
        for(i=0;i<this.countWords;i++){
            setArrCount(i, 0);
        }
    }
}

```

```

public String getColor(){
    if(this.nthColor == 0){
        return ANSI_RED;
    }else{
        if(this.nthColor == 1){
            return ANSI_YELLOW;
        }else{
            if(this.nthColor == 2){
                return ANSI_GREEN;
            }else{
                if(this.nthColor == 3){
                    return ANSI_BLUE;
                }else{
                    if(this.nthColor == 4){
                        return ANSI_CYAN;
                    }else{
                        if(this.nthColor == 5){
                            return ANSI_MAGENTA;
                        }else{
                            if(this.nthColor == 6){
                                return ANSI_BRED;
                            }else{
                                if(this.nthColor==7){
                                    return ANSI_BBLUE;
                                }else{
                                    return ANSI_RESET;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

public void updateColor(){
    this.nthColor++;
    this.nthColor %= 8;
}

public String getResetC(){
    return ANSI_RESET;
}

public void setMatP(int i, int j, String x){

```



```
        this.matrixPuzzle[i][j] = x;
    }
    public String[][] getMatP(){
        return this.matrixPuzzle;
    }

    public String getIsiMatP(int i, int j){
        return this.matrixPuzzle[i][j];
    }

    public void setMatW(int i, int j, String x){
        this.matrixWord[i][j] = x;
    }

    public String[][] getMatW(){
        return this.matrixWord;
    }

    public String[] getWord(int pos){
        return this.matrixWord[pos];
    }

    public void setWord(int pos, String[] S){
        this.matrixWord[pos] = S;
    }

    public String getLetter(int x, int y){
        return this.matrixWord[x][y];
    }

    public void setStatus(boolean status){
        this.found = status;
    }

    public boolean getStatus(){
        return this.found;
    }

    public void setCountW(int num){
        this.countWords = num;
    }

    public int getCountW(){
        return this.countWords;
    }

    public void setLength(int num){
        this.length = num;
    }
```

```

    }

    public int getLength(){
        return this.length;
    }

    public void setWidth(int num){
        this.width = num;
    }

    public int getWidth(){
        return this.width;
    }

    public void setMatPC(int i, int j, int x){
        this.matrixPuzzleC[i][j] = x;
    }

    public void initMatPC(){
        int i,j;
        for(i=0;i<this.length;i++){
            for(j=0;j<this.width;j++){
                setMatPC(i,j,8);
            }
        }
    }

    public int getIsiMatPC(int i, int j){
        return this.matrixPuzzleC[i][j];
    }

    public int[][] getMatPC(){
        return this.matrixPuzzleC;
    }

    public int getColorCode(){
        return this.nthColor;
    }

    public void setColorCode(int x){
        this.nthColor = x;
    }
}

```

## Hasil Screenshot

Input(small1.txt):

```
test > ≡ small1.txt
 1  E G A Q Y T A C V J L R
 2  I N G R O M I O F H A J
 3  Q F I W U B H W J R P Y
 4  R M A R O G S B U K L B
 5  R A A Z A I U J U F U Q
 6  A T M V A M R N X X S B
 7  O S P X D I R O Y O K A
 8  K U W E R C W A T A M E
 9  I R P O K O X U J A O R
10  M I H T I O V N H E U I
11  U A C Q W Y R C H Q B I
12  A L E N W P A A K K Y K
13  U M M J E A M O N I N U
14  C J E S H I M E M O R I
15
16  AME
17  GURA
18  HAACHAMA
19  HIMEMORI
20  IROHA
21  KOYORI
22  LAPLUS
23  MARINE
24  MATSURI
25  MIKO
26  NINOMAE
27  PEKORA
28  RUSHIA
29  TOWA
30  WATAME
```

Output:

Kata TOWA ditemukan pada koordinat (0,5) diagonal ke kiri bawah dalam :  
8 perbandingan dan waktu 8ms

Kata LAPLUS ditemukan pada koordinat (0,10) vertikal ke bawah dalam :  
8 perbandingan dan waktu 13ms

Kata MATSURI ditemukan pada koordinat (3,1) vertikal ke bawah dalam :  
20 perbandingan dan waktu 14ms

Kata GURA ditemukan pada koordinat (3,5) diagonal ke kiri atas dalam :  
25 perbandingan dan waktu 15ms

Kata MARINE ditemukan pada koordinat (5,5) diagonal ke kiri atas dalam :  
45 perbandingan dan waktu 16ms

Kata RUSHIA ditemukan pada koordinat (5,6) vertikal ke atas dalam :  
53 perbandingan dan waktu 17ms

Kata PEKORA ditemukan pada koordinat (6,2) diagonal ke kanan bawah dalam :  
14 perbandingan dan waktu 18ms

Kata IROHA ditemukan pada koordinat (6,5) diagonal ke kiri bawah dalam :  
46 perbandingan dan waktu 19ms

Kata KOYORI ditemukan pada koordinat (6,10) mendatar ke kiri dalam :  
16 perbandingan dan waktu 20ms

Kata WATAME ditemukan pada koordinat (7,6) mendatar ke kanan dalam :  
34 perbandingan dan waktu 21ms

Kata AME ditemukan pada koordinat (7,9) mendatar ke kanan dalam :  
97 perbandingan dan waktu 22ms

Kata MIKO ditemukan pada koordinat (9,0) vertikal ke atas dalam :  
55 perbandingan dan waktu 23ms

Kata NINOMAE ditemukan pada koordinat (12,10) mendatar ke kiri dalam :  
55 perbandingan dan waktu 24ms

Kata HAACHAMA ditemukan pada koordinat (13,4) diagonal ke kanan atas dalam :  
61 perbandingan dan waktu 25ms

Kata HIMEMORI ditemukan pada koordinat (13,4) mendatar ke kanan dalam :  
55 perbandingan dan waktu 26ms

```

E G A Q Y T A C V J L R
I N G R O M I O F H A J
Q F I W U B H W J R P Y
R M A R O G S B U K L B
R A A Z A I U J U F U Q
A T M V A M R N X X S B
O S P X D I R O Y O K A
K U W E R C W A T A M E
I R P O K O X U J A O R
M I H T I O V N H E U I
U A C Q W Y R C H Q B I
A L E N W P A A K K Y K
U M M J E A M O N I N U
C J E S H I M E M O R I

```

```

Total perbandingan yang dilakukan:
643
Total waktu yang digunakan:
27ms
Total kata yang ditemukan :
15

```

Input(small2.txt):

```

test > small2.txt
1  D B H L D T Y Q R Z A H Z J
2  M L V T T N J Q E B V S H P
3  D E R G D I W A S M R J X T
4  E T T O D R K T V E I P O D
5  C N D H W P R O T A L I P K
6  O E U P O A Q N R O L L E H
7  R M S L C D I P V S C O D E
8  A E P T R O O N M G L R X B
9  T R Y F P O E L B A I R A V
10 O C T Y L C L A S S P U K Y
11 R N H G D T Q G Q X G R O M
12 S I O I J T S K I F F H T Q
13 U R N L V I S C V Z V U K E
14 O Z R Z W N E O E J M E O Q
15 W L B D W Z V V O X S Y D O
16
17 ABSTRACT
18 CLASS
19 DECORATORS
20 HELLO
21 INCREMENT
22 JAVA
23 LOOP
24 METHOD
25 POINTERS
26 PRINT
27 PYTHON
28 VARIABLE
29 VSCODE
30 WORLD

```

## Output:

Kata ABSTRACT ditemukan pada koordinat (0,10) diagonal ke kiri bawah dalam :  
12 perbandingan dan waktu 8ms

Kata METHOD ditemukan pada koordinat (1,0) diagonal ke kanan bawah dalam :  
4 perbandingan dan waktu 13ms

Kata JAVA ditemukan pada koordinat (1,6) diagonal ke kanan bawah dalam :  
9 perbandingan dan waktu 14ms

Kata DECORATORS ditemukan pada koordinat (2,0) vertikal ke bawah dalam :  
23 perbandingan dan waktu 15ms

Kata WORLD ditemukan pada koordinat (4,4) diagonal ke kiri atas dalam :  
21 perbandingan dan waktu 16ms

Kata PRINT ditemukan pada koordinat (4,5) vertikal ke atas dalam :  
24 perbandingan dan waktu 17ms

Kata HELLO ditemukan pada koordinat (5,13) mendatar ke kiri dalam :  
35 perbandingan dan waktu 19ms

Kata VSCODE ditemukan pada koordinat (6,8) mendatar ke kanan dalam :  
35 perbandingan dan waktu 20ms

Kata PYTHON ditemukan pada koordinat (7,2) vertikal ke bawah dalam :  
60 perbandingan dan waktu 21ms

Kata POINTERS ditemukan pada koordinat (8,4) diagonal ke kanan atas dalam :  
81 perbandingan dan waktu 22ms

Kata VARIABLE ditemukan pada koordinat (8,13) mendatar ke kiri dalam :  
45 perbandingan dan waktu 23ms

Kata LOOP ditemukan pada koordinat (9,4) diagonal ke kanan atas dalam :  
86 perbandingan dan waktu 24ms

Kata CLASS ditemukan pada koordinat (9,5) mendatar ke kanan dalam :  
42 perbandingan dan waktu 25ms

Kata INCREMENT ditemukan pada koordinat (11,1) vertikal ke atas dalam :  
58 perbandingan dan waktu 26ms

```
DBHLD TYQRZ AHZJ
MLVTT NJQE BVSH P
DERGDIWAS MRJXT
ET TODRK TVE IPOD
CNDHW PROTALIPK
OEUP O AQN ROLLEH
RMSLC DI PVSCODE
AEPTR OONMGLRXB
TRYF POELBAIRAV
OCTYLCLASSPUKY
RNHGD TQGQXGROM
SIOIJTSKIFFHTQ
URNLVISCVZVUKE
OZRZWNEOEJMEQQ
WLB DWZVV OXS YDO

Total perbandingan yang dilakukan:
580
Total waktu yang digunakan:
27ms
Total kata yang ditemukan :
14
```

Input(small3.txt):

```
test > small3.txt
1 CSHRUKATGNWQEOV
2 MGHYERGNILGNAIX
3 JB AEABICFZBWEWF
4 ISXBNQM IKLEE QND
5 ZQRZEHSABEIDOU L
6 EAIKHCEILBSEONI
7 BIWQH PNH AUIMISS
8 GEELJIKVANUNVUA
9 XUKFJWB OBLGVOCY
10 LLONNWWAKGOVCRI
11 SAUAHARKUODCZOM
12 ZYVNTSYANWMIXSI
13 BIBYBUNYAHNIOEO
14 UYNAGGHAELLEONY
15 ROSARIADJZBJLVA
16
17 AMBER
18 AYAKA
19 BARBARA
20 BEIDOU
21 DIONA
22 EULA
23 FISCHL
24 GANYU
25 HUTAO
26 JEAN
27 KEQING
28 KLEE
```

29	KOKOMI
30	LISA
31	NINGGUANG
32	NOELLE
33	QIQI
34	BAAL
35	SAYU
36	ROSARIA
37	SHENHE
38	SUCROSE
39	XIANGLING
40	YANFEI
41	YOIMIYA
42	YUNJIN

**Output:**

Kata SHENHE ditemukan pada koordinat (0,1) diagonal ke kanan bawah dalam :  
5 perbandingan dan waktu 7ms

Kata XIANGLING ditemukan pada koordinat (1,14) mendatar ke kiri dalam :  
9 perbandingan dan waktu 13ms

Kata FISCHL ditemukan pada koordinat (2,8) diagonal ke kiri bawah dalam :  
12 perbandingan dan waktu 13ms

Kata SAYU ditemukan pada koordinat (3,1) diagonal ke kanan atas dalam :  
17 perbandingan dan waktu 14ms

Kata KLEE ditemukan pada koordinat (3,8) mendatar ke kanan dalam :  
10 perbandingan dan waktu 15ms

Kata AMBER ditemukan pada koordinat (4,7) diagonal ke kiri atas dalam :  
44 perbandingan dan waktu 16ms

Kata BEIDOU ditemukan pada koordinat (4,8) mendatar ke kanan dalam :  
46 perbandingan dan waktu 17ms

Kata LISA ditemukan pada koordinat (4,14) vertikal ke bawah dalam :  
25 perbandingan dan waktu 18ms

Kata KEQING ditemukan pada koordinat (5,3) diagonal ke kanan atas dalam :  
29 perbandingan dan waktu 19ms

Kata NINGGUANG ditemukan pada koordinat (5,13) diagonal ke kiri bawah dalam :  
59 perbandingan dan waktu 20ms

Kata BARBARA ditemukan pada koordinat (6,0) diagonal ke kanan atas dalam :  
72 perbandingan dan waktu 20ms



Kata QIQI ditemukan pada koordinat (6,3) diagonal ke kiri atas dalam :  
44 perbandingan dan waktu 21ms

Kata SUCROSE ditemukan pada koordinat (6,13) vertikal ke bawah dalam :  
44 perbandingan dan waktu 21ms

Kata EULA ditemukan pada koordinat (7,1) vertikal ke bawah dalam :  
100 perbandingan dan waktu 22ms

Kata KOKOMI ditemukan pada koordinat (7,6) diagonal ke kanan bawah dalam :  
29 perbandingan dan waktu 22ms

Kata BAAL ditemukan pada koordinat (8,8) vertikal ke atas dalam :  
82 perbandingan dan waktu 23ms

Kata DIONA ditemukan pada koordinat (10,10) diagonal ke kanan bawah dalam :  
21 perbandingan dan waktu 26ms

Kata YUNJIN ditemukan pada koordinat (11,1) diagonal ke kanan atas dalam :  
29 perbandingan dan waktu 27ms

Kata YANFEI ditemukan pada koordinat (11,6) diagonal ke kiri atas dalam :  
41 perbandingan dan waktu 28ms

Kata GANYU ditemukan pada koordinat (13,4) mendatar ke kiri dalam :  
63 perbandingan dan waktu 29ms

Kata HUTAO ditemukan pada koordinat (13,6) diagonal ke kiri atas dalam :  
80 perbandingan dan waktu 30ms

Kata AYAKA ditemukan pada koordinat (13,7) vertikal ke atas dalam :  
154 perbandingan dan waktu 30ms

Kata NOELLE ditemukan pada koordinat (13,13) mendatar ke kiri dalam :  
156 perbandingan dan waktu 31ms

Kata YOIMIYA ditemukan pada koordinat (13,14) vertikal ke atas dalam :  
68 perbandingan dan waktu 32ms

Kata ROSARIA ditemukan pada koordinat (14,0) mendatar ke kanan dalam :  
52 perbandingan dan waktu 32ms

Kata JEAN ditemukan pada koordinat (14,8) vertikal ke atas dalam :  
30 perbandingan dan waktu 33ms

```

C S H R U K A T G N W Q E O V
M G H Y E R G N I L G N A I X
J B A E A B I C F Z B W E W F
I S X B N Q M I K L E E Q N D
Z Q R Z E H S A B E I D O U L
E A I K H C E I L B S E O N I
B I W Q H P N H A U I M I S S
G E E L J I K V A N U N V U A
X U K F J W B O B L G V O C Y
L L O N N W W A K G O V C R I
S A U A H A R K U O D C Z O M
Z Y V N T S Y A N W M I X S I
B I B Y B U N Y A H N I O E O
U Y N A G G H A E L L E O N Y
R O S A R I A D J Z B J L V A

```

```

Total perbandingan yang dilakukan:
1395
Total waktu yang digunakan:
33ms
Total kata yang ditemukan :
26

```

Input(medium1.txt):

```

1  I Z T B I O G E D A D B J N P X L S H M
2  B E W I M Z P N L B V Y Z E O N A T O M
3  V C A S U N M X I I G Y A T K P M T T F
4  E O R C L A W G B W P G G E K E Z U O G
5  C P F U L H Z U S H I E E R L P K A G R
6  R Y T I I K E I N K J Z Z O E U U V C Z
7  F V O T N T N O L Z Z U D I Z L N G M O
8  E R T P I E B O A F O Y R I L P K M Q A
9  I H D K I U M F V P Q E H I R M Z G P H
10 T N F K N X I P J O D S K L E R O M O T
11 A X H A R P K R A N L A H S S C X M T K
12 N D G Y O K N U C K L E S F L T G P U N
13 S A H T E I G U J N P J N C B C A R L O
14 D A R I D K N S M V Z H G Z Y R A Q L U
15 P O D J H M I N C M M A I N I P N L A Z
16 K E P A L C G M E R U E M S I C O O K V
17 D N C S S Y N F I X E W T K O R Y O G E
18 C T M E W O G E W X I O A O H K Y U C U
19 G P U S C H A K M M N V P C U Y A X G W
20 A L L U K A O I R O E L I H C A M O K T

```

22	ALLUKA
23	BISCUIT
24	CHROLLO
25	FEITAN
26	GING
27	GON
28	GOTOH
29	HANZO
30	HISOKA
31	ILLUMI
32	KALLUTO
33	KILLUA
34	KITE
35	KNOV
36	KNUCKLE
37	KORTOPI
38	KURAPIKA
39	LEORIO
40	MACHI
41	MELODY
42	MENCHI
43	MERUEM
44	MOREL
45	NETERO
46	NOBUNAGA
47	PARISTON
48	POKKLE
49	SADASO
50	SHALNARK
51	SHIZUKU
52	WING
53	YOPI
54	ZEPILE
55	ZUSHI

## Output:

Kata BISCUIT ditemukan pada koordinat (0,3) vertikal ke bawah dalam :  
10 perbandingan dan waktu 7ms

Kata NETERO ditemukan pada koordinat (0,13) vertikal ke bawah dalam :  
8 perbandingan dan waktu 12ms

Kata POKKLE ditemukan pada koordinat (0,14) vertikal ke bawah dalam :  
8 perbandingan dan waktu 13ms

Kata MELODY ditemukan pada koordinat (2,16) diagonal ke kiri bawah dalam :  
39 perbandingan dan waktu 14ms

Kata WING ditemukan pada koordinat (3,9) diagonal ke kiri atas dalam :  
30 perbandingan dan waktu 16ms

Kata HANZO ditemukan pada koordinat (4,5) vertikal ke atas dalam :  
13 perbandingan dan waktu 17ms

Kata ZUSHI ditemukan pada koordinat (4,6) mendatar ke kanan dalam :  
40 perbandingan dan waktu 18ms

Kata GOTOH ditemukan pada koordinat (4,18) vertikal ke atas dalam :  
57 perbandingan dan waktu 20ms

Kata YOPI ditemukan pada koordinat (5,1) diagonal ke kanan bawah dalam :  
23 perbandingan dan waktu 21ms

Kata ILLUMI ditemukan pada koordinat (5,4) vertikal ke atas dalam :  
66 perbandingan dan waktu 22ms

Kata KNOV ditemukan pada koordinat (5,5) diagonal ke kanan bawah dalam :  
33 perbandingan dan waktu 23ms

Kata NOBUNAGA ditemukan pada koordinat (5,8) diagonal ke kiri bawah dalam :  
47 perbandingan dan waktu 24ms

Kata ZEPILE ditemukan pada koordinat (5,12) diagonal ke kiri atas dalam :  
70 perbandingan dan waktu 25ms

Kata FEITAN ditemukan pada koordinat (6,0) vertikal ke bawah dalam :  
23 perbandingan dan waktu 26ms

Kata KITE ditemukan pada koordinat (8,3) diagonal ke kanan atas dalam :  
68 perbandingan dan waktu 27ms

Kata SHIZUKU ditemukan pada koordinat (9,11) diagonal ke kanan atas dalam :  
41 perbandingan dan waktu 29ms

Kata KILLUA ditemukan pada koordinat (9,12) diagonal ke kanan atas dalam :  
90 perbandingan dan waktu 29ms

Kata MOREL ditemukan pada koordinat (9,17) mendatar ke kiri dalam :  
83 perbandingan dan waktu 31ms

Kata SHALNARK ditemukan pada koordinat (10,13) mendatar ke kiri dalam :  
45 perbandingan dan waktu 32ms

Kata KURAPIKA ditemukan pada koordinat (10,19) diagonal ke kiri bawah dalam :  
104 perbandingan dan waktu 33ms

Kata KNUCKLE ditemukan pada koordinat (11,5) mendatar ke kanan dalam :  
111 perbandingan dan waktu 34ms

Kata PARISTON ditemukan pada koordinat (11,17) diagonal ke kiri bawah dalam :  
120 perbandingan dan waktu 35ms

Kata SADASO ditemukan pada koordinat (12,0) diagonal ke kanan bawah dalam :  
70 perbandingan dan waktu 36ms

Kata HISOKA ditemukan pada koordinat (13,11) diagonal ke kanan bawah dalam :  
84 perbandingan dan waktu 37ms

Kata KORTOPI ditemukan pada koordinat (15,0) diagonal ke kanan atas dalam :  
137 perbandingan dan waktu 38ms

Kata GING ditemukan pada koordinat (15,6) vertikal ke atas dalam :  
122 perbandingan dan waktu 39ms

Kata MERUEM ditemukan pada koordinat (15,7) mendatar ke kanan dalam :  
137 perbandingan dan waktu 39ms

Kata KALLUTO ditemukan pada koordinat (15,18) vertikal ke atas dalam :  
143 perbandingan dan waktu 40ms

Kata GON ditemukan pada koordinat (16,18) diagonal ke kiri atas dalam :  
134 perbandingan dan waktu 41ms

Kata MENCHI ditemukan pada koordinat (18,8) diagonal ke kiri atas dalam :  
174 perbandingan dan waktu 42ms

Kata CHROLLO ditemukan pada koordinat (18,13) diagonal ke kanan atas dalam :  
145 perbandingan dan waktu 43ms

Kata ALLUKA ditemukan pada koordinat (19,0) mendatar ke kanan dalam :  
206 perbandingan dan waktu 44ms

Kata LEORIO ditemukan pada koordinat (19,11) mendatar ke kiri dalam :  
166 perbandingan dan waktu 44ms

Kata MACHI ditemukan pada koordinat (19,16) mendatar ke kiri dalam :  
181 perbandingan dan waktu 45ms

```

I Z T B I O G E D A D B J N P X L S H M
B E W I M Z P N L B V Y Z E O N A T O M
V C A S U N M X I I G Y A T K P M T T F
E O R C L A W G B W P G G E K E Z U O G
C P F U L H Z U S H I E E R L P K A G R
R Y T I I K E I N K J Z Z O E U U V C Z
F V O T N T N O L Z Z U D I Z L N G M O
E R T P I E B O A F O Y R I L P K M Q A
I H D K I U M F V P Q E H I R M Z G P H
T N F K N X I P J O D S K L E R O M O T
A X H A R P K R A N L A H S S C X M T K
N D G Y O K N U C K L E S F L T G P U N
S A H T E I G U J N P J N C B C A R L O
D A R I D K N S M V Z H G Z Y R A Q L U
P O D J H M I N C M M A I N I P N L A Z
K E P A L C G M E R U E M S I C O O K V
D N C S S Y N F I X E W T K O R Y O G E
C T M E W O G E W X I O A O H K Y U C U
G P U S C H A K M M N V P C U Y A X G W
A L L U K A O I R O E L I H C A M O K T

Total perbandingan yang dilakukan:
2925
Total waktu yang digunakan:
45ms
Total kata yang ditemukan :
34

```


Input(medium2.txt):

```

test > ≡ medium2.txt
1  A T I M P I G N I G N I R B P U W M K V B T
2  Q G B E W H I M S I C A L L E X V G X R G G
3  H I J G F W X V V L Z D E U G R E Z O D N U
4  H E A O S A Y Y H S I U G N A C C T X I L L
5  N N F B I M U N U D S M W E Z Z H A H Y K I
6  F I Y D L K T Q J N X R Y P I E Y D S Y V V
7  X V K E I N R D E J Q L Y P R J L V A S K R
8  R I G U Q E C N A H C R E P V L T N L R A T
9  A D R G R D Y W U Y Q J B X W B N N B K N M
10 C F C O R D L C S R A A D U Q Q I Y N Q P S
11 M E X L T Q E U Q A T Y A Z W C A G V P T X
12 I B Q A S C M R L I Z X F I U B U H G D G D
13 G R U T R S O E W N V V U B Z P Q S T J O T
14 F K Q A W U H D I U R O T A L U M I T S S W
15 P Q Q C R N Q E B C D S T P F O P P D M Q E
16 T P U X F B O J I E X K I X I L N R P C K W
17 N D E T C E J E R P E Y E T P L Y N V M I V
18 T X Z D Z A Y A A B K Q C O N F A F I I B Y
19 G N I C A M I R G W B E E L Y M P H O M A D
20 X T C B V I J O H F O L D G C T O T E C A F

```





22	ANGUISH
23	BROTHER
24	CATALOGUED
25	CURED
26	DARNS
27	DECEIT
28	DIVINE
29	FACET
30	GRIMACING
31	HOMELY
32	LYMPHOMA
33	MASSACRE
34	PECUNIARY
35	PERCHANCE
36	QUAINTLY
37	REJECTED
38	STIMULATOR
39	SUNBEAM
40	UPBRINGING
41	WHIMSICAL

**Output:**

Kata UPBRINGING ditemukan pada koordinat (0,15) mendatar ke kiri dalam :  
11 perbandingan dan waktu 9ms

Kata BROTHER ditemukan pada koordinat (0,20) diagonal ke kiri bawah dalam :  
18 perbandingan dan waktu 15ms

Kata WHIMSICAL ditemukan pada koordinat (1,4) mendatar ke kanan dalam :  
15 perbandingan dan waktu 16ms

Kata ANGUISH ditemukan pada koordinat (3,14) mendatar ke kiri dalam :  
40 perbandingan dan waktu 16ms

Kata DARNS ditemukan pada koordinat (5,17) diagonal ke kanan bawah dalam :  
43 perbandingan dan waktu 17ms

Kata PERCHANCE ditemukan pada koordinat (7,13) mendatar ke kiri dalam :  
44 perbandingan dan waktu 18ms

Kata DIVINE ditemukan pada koordinat (8,1) vertikal ke atas dalam :  
66 perbandingan dan waktu 19ms

Kata MASSACRE ditemukan pada koordinat (8,21) diagonal ke kiri atas dalam :  
52 perbandingan dan waktu 20ms

Kata CURED ditemukan pada koordinat (9,7) vertikal ke bawah dalam :  
68 perbandingan dan waktu 22ms

Kata SUNBEAM ditemukan pada koordinat (12,5) vertikal ke bawah dalam :  
92 perbandingan dan waktu 23ms

Kata QUAINLY ditemukan pada koordinat (12,16) vertikal ke atas dalam :  
113 perbandingan dan waktu 23ms

Kata HOMELY ditemukan pada koordinat (13,6) vertikal ke atas dalam :  
74 perbandingan dan waktu 24ms

Kata STIMULATOR ditemukan pada koordinat (13,19) mendatar ke kiri dalam :  
111 perbandingan dan waktu 25ms

Kata CATALOGUED ditemukan pada koordinat (14,3) vertikal ke atas dalam :  
103 perbandingan dan waktu 25ms

Kata REJECTED ditemukan pada koordinat (16,8) mendatar ke kiri dalam :  
188 perbandingan dan waktu 26ms

Kata PECUNIARY ditemukan pada koordinat (16,9) vertikal ke atas dalam :  
134 perbandingan dan waktu 27ms

Kata GRIMACING ditemukan pada koordinat (18,8) mendatar ke kiri dalam :  
145 perbandingan dan waktu 27ms

Kata LYMPHOMA ditemukan pada koordinat (18,13) mendatar ke kanan dalam :  
155 perbandingan dan waktu 28ms

Kata DECEIT ditemukan pada koordinat (19,12) vertikal ke atas dalam :  
173 perbandingan dan waktu 29ms

Kata FACET ditemukan pada koordinat (19,21) mendatar ke kiri dalam :  
100 perbandingan dan waktu 29ms

```

A T I M P I G N I G N I R B P U W M K V B T
Q G B E W H I M S I C A L L E X V G X R G G
H I J G F W X V V L Z D E U G R E Z O D N U
H E A O S A Y Y H S I U G N A C C T X I L L
N N F B I M U N U D S M W E Z Z H A H Y K I
F I Y D L K T Q J N X R Y P I E Y D S Y V V
X V K E I N R D E J Q L Y P R J L V A S K R
R I G U Q E C N A H C R E P V L T N L R A T
A D R G R D Y W U Y Q J B X W B N N B K N M
C F C O R D L C S R A A D U Q Q I Y N Q P S
M E X L T Q E U Q A T Y A Z W C A G V P T X
I B Q A S C M R L I Z X F I U B U H G D G D
G R U T R S O E W N V V U B Z P Q S T J O T
F K Q A W U H D I U R O T A L U M I T S S W
P Q Q C R N Q E B C D S T P F O P P D M Q E
T P U X F B O J I E X K I X I L N R P C K W
N D E T C E J E R P E Y E T P L Y N V M I V
T X Z D Z A Y A A B K Q C O N F A F I I B Y
G N I C A M I R G W B E E L Y M P H O M A D
X T C B V I J O H F O L D G C T O T E C A F

```

Total perbandingan yang dilakukan:  
1838  
Total waktu yang digunakan:  
30ms  
Total kata yang ditemukan :  
20



Input(medium3.txt):

```
test > medium3.txt
1  A T I M P I G N I G N I R B P U W M K V B T
2  Q G B E W H I M S I C A L L E X V G X R G G
3  H I J G F W X V V L Z D E U G R E Z O D N U
4  H E A O S A Y Y H S I U G N A C C T X I L L
5  N N F B I M U N U D S M W E Z Z H A H Y K I
6  F I Y D L K T Q J N X R Y P I E Y D S Y V V
7  X V K E I N R D E J Q L Y P R J L V A S K R
8  R I G U Q E C N A H C R E P V L T N L R A T
9  A D R G R D Y W U Y Q J B X W B N N B K N M
10 C F C O R D L C S R A A D U Q Q I Y N Q P S
11 M E X L T Q E U Q A T Y A Z W C A G V P T X
12 I B Q A S C M R L I Z X F I U B U H G D G D
13 G R U T R S O E W N V V U B Z P Q S T J O T
14 F K Q A W U H D I U R O T A L U M I T S S W
15 P Q Q C R N Q E B C D S T P F O P P D M Q E
16 T P U X F B O J I E X K I X I L N R P C K W
17 N D E T C E J E R P E Y E T P L Y N V M I V
18 T X Z D Z A Y A A B K Q C O N F A F I I B Y
19 G N I C A M I R G W B E E L Y M P H O M A D
20 X T C B V I J O H F O L D G C T O T E C A F
```

```
22  ANGUISH
23  BROTHER
24  CATALOGUED
25  CURED
26  DARNS
27  DECEIT
28  DIVINE
29  FACET
30  GRIMACING
31  HOMELY
32  LYMPHOMA
33  MASSACRE
34  PECUNIARY
35  PERCHANCE
36  QUAINLY
37  REJECTED
38  STIMULATOR
39  SUNBEAM
40  UPBRINGING
41  WHIMSICAL
```

## Output:

Kata UPBRINGING ditemukan pada koordinat (0,15) mendatar ke kiri dalam :  
11 perbandingan dan waktu 9ms

Kata BROTHER ditemukan pada koordinat (0,20) diagonal ke kiri bawah dalam :  
18 perbandingan dan waktu 13ms

Kata WHIMSICAL ditemukan pada koordinat (1,4) mendatar ke kanan dalam :  
15 perbandingan dan waktu 14ms

Kata ANGUISH ditemukan pada koordinat (3,14) mendatar ke kiri dalam :  
40 perbandingan dan waktu 15ms

Kata DARNs ditemukan pada koordinat (5,17) diagonal ke kanan bawah dalam :  
43 perbandingan dan waktu 16ms

Kata PERCHANCE ditemukan pada koordinat (7,13) mendatar ke kiri dalam :  
44 perbandingan dan waktu 17ms

Kata DIVINE ditemukan pada koordinat (8,1) vertikal ke atas dalam :  
66 perbandingan dan waktu 18ms

Kata MASSACRE ditemukan pada koordinat (8,21) diagonal ke kiri atas dalam :  
52 perbandingan dan waktu 19ms

Kata CURED ditemukan pada koordinat (9,7) vertikal ke bawah dalam :  
68 perbandingan dan waktu 20ms

Kata SUNBEAM ditemukan pada koordinat (12,5) vertikal ke bawah dalam :  
92 perbandingan dan waktu 21ms

Kata QUAINtLY ditemukan pada koordinat (12,16) vertikal ke atas dalam :  
113 perbandingan dan waktu 21ms

Kata HOMELY ditemukan pada koordinat (13,6) vertikal ke atas dalam :  
74 perbandingan dan waktu 22ms

Kata STIMULATOR ditemukan pada koordinat (13,19) mendatar ke kiri dalam :  
111 perbandingan dan waktu 22ms

Kata CATALOGUED ditemukan pada koordinat (14,3) vertikal ke atas dalam :  
103 perbandingan dan waktu 23ms

Kata REJECTED ditemukan pada koordinat (16,8) mendatar ke kiri dalam :  
188 perbandingan dan waktu 23ms

Kata PECUNIARY ditemukan pada koordinat (16,9) vertikal ke atas dalam :  
134 perbandingan dan waktu 24ms

Kata GRIMACING ditemukan pada koordinat (18,8) mendatar ke kiri dalam :  
145 perbandingan dan waktu 25ms

Kata LYMPHOMA ditemukan pada koordinat (18,13) mendatar ke kanan dalam :  
155 perbandingan dan waktu 26ms

Kata DECEIT ditemukan pada koordinat (19,12) vertikal ke atas dalam :  
173 perbandingan dan waktu 27ms

Kata FACET ditemukan pada koordinat (19,21) mendatar ke kiri dalam :  
100 perbandingan dan waktu 27ms

```
A T I M P I G N I G N I R B P U W M K V B T
Q G B E W H I M S I C A L L E X V G X R G G
H I J G F W X V V L Z D E U G R E Z O D N U
H E A O S A Y Y H S I U G N A C C T X I L L
N N F B I M U N U D S M W E Z Z H A H Y K I
F I Y D L K T Q J N X R Y P I E Y D S Y V V
X V K E I N R D E J Q L Y P R J L V A S K R
R I G U Q E C N A H C R E P V L T N L R A T
A D R G R D Y W U Y Q J B X W B N N B K N M
C F C O R D L C S R A A D U Q Q I Y N Q P S
M E X L T Q E U Q A T Y A Z W C A G V P T X
I B Q A S C M R L I Z X F I U B U H G D G D
G R U T R S O E W N V V U B Z P Q S T J O T
F K Q A W U H D I U R O T A L U M I T S S W
P Q Q C R N Q E B C D S T P F O P P D M Q E
T P U X F B O J I E X K I X I L N R P C K W
N D E T C E J E R P E Y E T P L Y N V M I V
T X Z D Z A Y A A B K Q C O N F A F I I B Y
G N I C A M I R G W B E E L Y M P H O M A D
X T C B V I J O H F O L D G C T O T E C A F
```

Total perbandingan yang dilakukan:

1838

Total waktu yang digunakan:

28ms

Total kata yang ditemukan :

20

Input(large1.txt):

test > ≡ large1.txt

```
1  C V L U X G E L L B Z Z R D S W E N D T I E M Z G Z Q X I U Y X
2  P E Q G Y N L I O A Y V X K R I H M Z W N T N H V B M R A Y I E
3  X M I S T A C Q L C A L L I M J V Z A I Z K D U Z A O S T S D Z
4  C D H S T U K A G L O I R U S T A M E T S L S G N Y B O O P M E
5  S E J Y F I A O K C O A L P Y U L R D I A H K U O H C Y C S Y N
6  U B G I F I K Z M I Y U L K B F L S Y G D W L K L O R W N C Z Q
7  B K T U H X D U K K E X B F E I A P Y N S P W O B R Z E E S Y G
8  J Y G O E K A S T I F C H M M V U F A R I Q X O N H B S E D T F
9  W C F E B B E X L O D K F F S M U E A L I B R O I C W W S Y X P
10 W V K I X U F J J Q P A A E W B J N U A A X E A L N J M Y Y S G
11 D N S E M Y A K Y B D E C M U T U O O M M L N X F B H A Q M T U
12 L O Y M F R V Y Z H E A T K A N X R R E A M K J H N M Z I R S X
13 A X K U O C K V A I M H I Z R R Q O F P H K N O A M Y V P E Y A
14 G S Q K R W R L S M D O C P O S I K H Z C M C K N G U H U F X N
15 T Z E A R I X E N D E R J C J B K N B X A G H T E Z D M L N X A
16 E P I W Z S R O A Z R I P N X N N B E H A V N I N A S X E Z E S
17 Y S Q O W E T Z K I L U A F X Y M O G U H Q F C E N I D V I Q H
18 O S O T C K R O N I I L A R J A G J Y X L F P W Z P V E W R Y F
19 Y R C R O J M Z M S G J G J O N O Z K O S Q O G F U K I V P S L
20 Y D A H I M E A R A I K L I W S K W R Z A H K M S I S H K U P A
21 R N K Y V K L J X C P K C T H M E V Z H A U N E F P A J B A Z R
22 A T A N A K A L U I O X V Y W Q V A H A R O I Y L U B A W G U E
23 L L B E B Y S H I O N S D C Z E D O J R J L N Q C A R A K L O P
24 M Q Z D W L A V A Q Z F P Q I O S U I S E I U J H U M M T M L O
25 G T L V M U P B G A K Z Z X X F F R T U Y M Z S O W Y Y O R P T
26 Q A T I Q E U R B K J N L E G S I H I O T V G X C U R O D D A G
27 S Z P A Q I U C X H F F C O C O C N G U L Y I U O K N B V Q M S
28 M R N C L G X X F B O A M U R N B H K T G N G U R A C O N M D H
29 B I E X Q I Z Z P G W R J F R I I X L P A G O N W A K Y U K V A
30 B S A Z J D A V V O R U S H I A B R Q O B I L H C B O A D Y C Y
31 A U L M B K E A K K H J X H O O K S R G E H P O O F M I E F D N
32 E W B K E N X C C I T K U P T N O I M P D V X M F D U E A F I A
33 L L D J C X C K Y M E V N A G J U B G N X V L H R Z I F O I D T
34 Z I M A Z J E T J Q C I N S H I A K A P C I F O D K I C S T M Z
```

36	AKIROSE	68	MEL
37	AME	69	MIKO
38	ANYA	70	MIO
39	AQUA	71	MOONA
40	AYAME	72	MUMEI
41	AZKI	73	NENE
42	BAELS	74	NOEL
43	BOTAN	75	OKAYU
44	CALLI	76	OLLIE
45	CERES	77	PEKORA
46	CHLOE	78	POLKA
47	CHOCO	79	REINE
48	COCO	80	RISU
49	FLARE	81	ROBOCO
50	FUBUKI	82	RUSHIA
51	GURA	83	SANA
52	HAACHAMA	84	SHION
53	INA	85	SORA
54	IOFI	86	SUBARU
55	IROHA	87	SUISEI
56	IRYS	88	TOWA
57	KANATA	89	WATAME
58	KIARA		
59	KORONE		
60	KOYORI		
61	KRONII		
62	LAMY		
63	LAPLUS		
64	LUI		
65	LUNA		
66	MARINE		
67	MATSURI		



## Output:

Kata CALLI ditemukan pada koordinat (2,9) mendatar ke kanan dalam :  
19 perbandingan dan waktu 10ms

Kata AME ditemukan pada koordinat (2,18) diagonal ke kiri atas dalam :  
46 perbandingan dan waktu 16ms

Kata MATSURI ditemukan pada koordinat (3,17) mendatar ke kiri dalam :  
51 perbandingan dan waktu 18ms

Kata AZKI ditemukan pada koordinat (4,6) diagonal ke kanan bawah dalam :  
81 perbandingan dan waktu 20ms

Kata OKAYU ditemukan pada koordinat (4,7) diagonal ke kiri atas dalam :  
58 perbandingan dan waktu 21ms

Kata OLLIE ditemukan pada koordinat (4,10) diagonal ke kiri atas dalam :  
68 perbandingan dan waktu 22ms

Kata REINE ditemukan pada koordinat (4,17) diagonal ke kanan atas dalam :  
45 perbandingan dan waktu 23ms

Kata WATAME ditemukan pada koordinat (5,21) diagonal ke kiri atas dalam :  
28 perbandingan dan waktu 24ms

Kata LUNA ditemukan pada koordinat (5,22) diagonal ke kanan atas dalam :  
125 perbandingan dan waktu 25ms

Kata KOYORI ditemukan pada koordinat (5,23) diagonal ke kanan atas dalam :  
79 perbandingan dan waktu 26ms

Kata FUBUKI ditemukan pada koordinat (7,17) diagonal ke kiri bawah dalam :  
60 perbandingan dan waktu 27ms

Kata NOEL ditemukan pada koordinat (7,24) diagonal ke kiri bawah dalam :  
77 perbandingan dan waktu 28ms

Kata LAPLUS ditemukan pada koordinat (8,19) diagonal ke kiri atas dalam :  
159 perbandingan dan waktu 28ms

Kata IRYS ditemukan pada koordinat (8,20) diagonal ke kiri atas dalam :  
163 perbandingan dan waktu 29ms

Kata ROBOCO ditemukan pada koordinat (8,22) diagonal ke kanan atas dalam :  
86 perbandingan dan waktu 30ms

Kata AYAME ditemukan pada koordinat (10,6) diagonal ke kanan bawah dalam :  
193 perbandingan dan waktu 31ms

Kata MARINE ditemukan pada koordinat (10,13) diagonal ke kanan bawah dalam :  
123 perbandingan dan waktu 32ms

Kata MUMEI ditemukan pada koordinat (12,25) diagonal ke kanan bawah dalam :  
199 perbandingan dan waktu 32ms

Kata KORONE ditemukan pada koordinat (13,17) vertikal ke atas dalam :  
204 perbandingan dan waktu 33ms

Kata NENE ditemukan pada koordinat (13,24) vertikal ke bawah dalam :  
151 perbandingan dan waktu 35ms

Kata PEKORA ditemukan pada koordinat (15,1) diagonal ke kanan atas dalam :  
100 perbandingan dan waktu 36ms

Kata IROHA ditemukan pada koordinat (15,11) vertikal ke atas dalam :  
249 perbandingan dan waktu 37ms

Kata INA ditemukan pada koordinat (15,23) mendatar ke kanan dalam :  
244 perbandingan dan waktu 38ms

Kata SANA ditemukan pada koordinat (15,31) vertikal ke atas dalam :  
231 perbandingan dan waktu 39ms

Kata HAACHAMA ditemukan pada koordinat (16,20) vertikal ke atas dalam :  
171 perbandingan dan waktu 40ms

Kata TOWA ditemukan pada koordinat (17,3) vertikal ke atas dalam :  
148 perbandingan dan waktu 40ms

Kata CERES ditemukan pada koordinat (17,4) diagonal ke kanan atas dalam :  
171 perbandingan dan waktu 41ms

Kata KRONII ditemukan pada koordinat (17,5) mendatar ke kanan dalam :  
235 perbandingan dan waktu 42ms

Kata LUI ditemukan pada koordinat (17,11) vertikal ke atas dalam :  
211 perbandingan dan waktu 42ms

Kata FLARE ditemukan pada koordinat (17,31) vertikal ke bawah dalam :  
173 perbandingan dan waktu 43ms

Kata MEL ditemukan pada koordinat (18,6) vertikal ke bawah dalam :  
248 perbandingan dan waktu 44ms

Kata SUBARU ditemukan pada koordinat (18,30) diagonal ke kiri bawah dalam :  
276 perbandingan dan waktu 45ms

Kata KIARA ditemukan pada koordinat (19,11) mendatar ke kiri dalam :  
272 perbandingan dan waktu 45ms

Kata SORA ditemukan pada koordinat (19,15) diagonal ke kiri atas dalam :  
288 perbandingan dan waktu 46ms

Kata KANATA ditemukan pada koordinat (21,5) mendatar ke kiri dalam :  
342 perbandingan dan waktu 47ms

Kata AKIROSE ditemukan pada koordinat (21,6) diagonal ke kiri atas dalam :  
457 perbandingan dan waktu 47ms

Kata LAMY ditemukan pada koordinat (21,24) diagonal ke kanan bawah dalam :  
277 perbandingan dan waktu 48ms

Kata SHION ditemukan pada koordinat (22,6) mendatar ke kanan dalam :  
308 perbandingan dan waktu 49ms

Kata CHOCO ditemukan pada koordinat (22,24) vertikal ke bawah dalam :  
214 perbandingan dan waktu 50ms

Kata POLKA ditemukan pada koordinat (22,31) mendatar ke kiri dalam :  
166 perbandingan dan waktu 51ms

Kata SUISEI ditemukan pada koordinat (23,16) mendatar ke kanan dalam :  
327 perbandingan dan waktu 51ms

Kata MOONA ditemukan pada koordinat (23,29) diagonal ke kiri bawah dalam :  
319 perbandingan dan waktu 52ms

Kata GURA ditemukan pada koordinat (25,22) diagonal ke kanan bawah dalam :  
192 perbandingan dan waktu 53ms

Kata AQUA ditemukan pada koordinat (26,3) diagonal ke kanan atas dalam :  
544 perbandingan dan waktu 54ms

Kata COCO ditemukan pada koordinat (26,12) mendatar ke kanan dalam :  
246 perbandingan dan waktu 55ms

Kata CHLOE ditemukan pada koordinat (26,16) diagonal ke kanan bawah dalam :  
256 perbandingan dan waktu 55ms

Kata RISU ditemukan pada koordinat (27,1) vertikal ke bawah dalam :  
347 perbandingan dan waktu 56ms

Kata RUSHIA ditemukan pada koordinat (29,10) mendatar ke kanan dalam :  
392 perbandingan dan waktu 57ms

Kata BOTAN ditemukan pada koordinat (29,16) diagonal ke kiri bawah dalam :  
273 perbandingan dan waktu 58ms

Kata MIO ditemukan pada koordinat (31,18) mendatar ke kiri dalam :  
396 perbandingan dan waktu 59ms

Kata ANYA ditemukan pada koordinat (31,31) vertikal ke atas dalam :  
664 perbandingan dan waktu 59ms

Kata MIKO ditemukan pada koordinat (32,9) vertikal ke atas dalam :  
417 perbandingan dan waktu 60ms

Kata IOFI ditemukan pada koordinat (32,29) mendatar ke kiri dalam :  
562 perbandingan dan waktu 61ms



C V L U X G E L L B Z Z R D S W E N D T I E M Z G Z Q X I U Y X  
 P E Q G Y N L I O A Y V X K R I H M Z W N T N H V B M R A Y I E  
 X M I S T A C Q L C A L L I M J V Z A I Z K D U Z A O S T S D Z  
 C D H S T U K A G L O I R U S T A M E T S L S G N Y B O O P M E  
 S E J Y F I A O K C O A L P Y U L R D I A H K U O H C Y C S Y N  
 U B G I F I K Z M I Y U L K B F L S Y G D W L K L O R W N C Z Q  
 B K T U H X D U K K E X B F E I A P Y N S P W O B R Z E E S Y G  
 J Y G O E K A S T I F C H M M V U F A R I Q X O N H B S E D T F  
 W C F E B B E X L O D K F F S M U E A L I B R O I C W W S Y X P  
 W V K I X U F J J Q P A A E W B J N U A A X E A L N J M Y Y S G  
 D N S E M Y A K Y B D E C M U T U O O M M L N X F B H A Q M T U  
 L O Y M F R V Y Z H E A T K A N X R R E A M K J H N M Z I R S X  
 A X K U O C K V A I M H I Z R R Q O F P H K N O A M Y V P E Y A  
 G S Q K R W R L S M D O C P O S I K H Z C M C K N G U H U F X N  
 T Z E A R I X E N D E R J C J B K N B X A G H T E Z D M L N X A  
 E P I W Z S R O A Z R I P N X N N B E H A V N I N A S X E Z E S  
 Y S Q O W E T Z K I L U A F X Y M O G U H Q F C E N I D V I Q H  
 O S O T C K R O N I I L A R J A G J Y X L F P W Z P V E W R Y F  
 Y R C R O J M Z M S G J G J O N O Z K O S Q O G F U K I V P S L  
 Y D A H I M E A R A I K L I W S K W R Z A H K M S I S H K U P A  
 R N K Y V K L J X C P K C T H M E V Z H A U N E F P A J B A Z R  
 A T A N A K A L U I O X V Y W Q V A H A R O I Y L U B A W G U E  
 L L B E B Y S H I O N S D C Z E D O J R J L N Q C A R A K L O P  
 M Q Z D W L A V A Q Z F P Q I O S U I S E I U J H U M M T M L O  
 G T L V M U P B G A K Z Z X X F F R T U Y M Z S O W Y Y O R P T  
 Q A T I Q E U R B K J N L E G S I H I O T V G X C U R O D D A G  
 S Z P A Q I U C X H F F C O C O C N G U L Y I U O K N B V Q M S  
 M R N C L G X X F B O A M U R N B H K T G N G U R A C O N M D H  
 B I E X Q I Z Z P G W R J F R I I X L P A G O N W A K Y U K V A  
 B S A Z J D A V V O R U S H I A B R Q O B I L H C B O A D Y C Y  
 A U L M B K E A K K H J X H O O K S R G E H P O O F M I E F D N  
 E W B K E N X C C I T K U P T N O I M P D V X M F D U E A F I A  
 L L D J C X C K Y M E V N A G J U B G N X V L H R Z I F O I D T  
 Z I M A Z J E T J Q C I N S H I A K A P C I F O D K I C S T M Z

Total perbandingan yang dilakukan:

12239

Total waktu yang digunakan:

61ms

Total kata yang ditemukan :

53

Input(large2.txt):

```
test > ≡ large2.txt
1  Z B E Y D A J N Y S P E A A E F N H E E I F S E B S N C O R
2  L V Z B M J C Z L R S Q C L N A N O S S T V F C U X N N R C
3  R I G I L K E N T U M S R N M B Y R Y U R E Q P N R P R Y K
4  K T Z A G S Y T E C C F U I E I E Y A C H U O A M R F N C I
5  X U Q T H I P G U X Z A X L H F A H B M O N W U H Q O Y C D
6  W L Q J Y Q L T I S D Y I A R F L P F P A R Z Q V P I T P Q
7  H G X Y G E T R F V G Q V M G A D E L C W X P F Z G V Y S T
8  K P A E T Q T O K Q F A N R M F N G T A T Q N D Q K M C L A
9  S J O E N A V B Z C D Q T O H T K Y O L C W E Z G A U J A P
10 E Z B C L B N F H Q M N F L W R O N W S W I A X J B N Y E V
11 J H D L G N Y F O G T H B K T B R O V I O U D R B E X L S J
12 G X E V B C U J J T X Q Z J A B M A R H C G F U Z Q L U Z L
13 B B S R W D T E B U O O R A M D F E X T W Y D D S A V H D K
14 S E R A T N A E L N A T H M K I G Y M A Y E K J L N Y L Z W
15 K Y S G U U M J C C S X D Y H G M S O N R S O Y N P P I Y S
16 M V E Q B W F F U G T V L E G I R O F W E N U R E X V U X S
17 U M R L P H Y R B A Z E C M O C R U S I G E Q O K U M T V H
18 V L P G Q D F A U U L G J Y T R A M I A U F A N U L C P G B
19 S A M Y D Q X H A K N A P T Q T N C T U L E Y Y S L Z N W U
20 U U R T Q F X L K X H D L Z K F U X I L U D Q B J O P E R M
21 I C Q C P W D S T H A D A R D B Q N Y P S A Q T V P T J U N
22 R S R C T E Q P O V Z N G G J S Q T R H S X L A L U A H S G
23 I Y B G B U A F H W H A T G A J M R G R P E X T H V V S N L
24 S Q S A Z A R Z V P M N Q S M R S L N X P W D P A D P X H W
25 C H R J L I J U T Y O J Q L S R M F P A V P L D R I L U Z R
26 W A W N P L R D S N N C C K R M E O D D K K K O F T R J B X
27 N G A U W F Y U W Q L C S W D B A H C P R E J V P S L B A R
28 U I L D M U V X S V V Q I Z P A A Z W K J Y Q Q X H A D E N
29 R D H U K R T S F U B M Y R Q R R K D Z Y R G U M M U U Y G
30 K M F H Z F W Z Q F W Z F W A H G E P O P X T O I O H D G G
```



32	ACRUX
33	ADHARA
34	ALDEBARAN
35	ALNAIR
36	ALNILAM
37	ALTAIR
38	ANTARES
39	ARCTURUS
40	BELLATRIX
41	BETELGEUSE
42	CANOPUS
43	ELNATH
44	FOMALHAUT
45	HADAR
46	MIAPLACIDUS
47	MIMOSA
48	POLLUX
49	PROCYON
50	REGULUS
51	RIGEL
52	RIGILKENT
53	SHAULA
54	SIRIUS
55	SPICA
56	VEGA

#### Output:

```
Kata ACRUX ditemukan pada koordinat (0,12) vertikal ke bawah dalam :  
13 perbandingan dan waktu 9ms  
  
Kata ALNILAM ditemukan pada koordinat (0,13) vertikal ke bawah dalam :  
21 perbandingan dan waktu 15ms  
  
Kata RIGILKENT ditemukan pada koordinat (2,0) mendatar ke kanan dalam :  
31 perbandingan dan waktu 15ms  
  
Kata MIAPLACIDUS ditemukan pada koordinat (2,14) diagonal ke kanan bawah dalam :  
24 perbandingan dan waktu 16ms  
  
Kata CANOPUS ditemukan pada koordinat (6,19) diagonal ke kanan atas dalam :  
100 perbandingan dan waktu 17ms  
  
Kata PROCYON ditemukan pada koordinat (6,22) diagonal ke kiri atas dalam :  
86 perbandingan dan waktu 18ms  
  
Kata BETELGEUSE ditemukan pada koordinat (9,2) diagonal ke kanan atas dalam :  
68 perbandingan dan waktu 20ms
```

Kata FOMALHAUT ditemukan pada koordinat (9,12) diagonal ke kanan atas dalam :  
129 perbandingan dan waktu 21ms

Kata BELLATRIX ditemukan pada koordinat (12,1) diagonal ke kanan atas dalam :  
150 perbandingan dan waktu 22ms

Kata MIMOSA ditemukan pada koordinat (12,14) diagonal ke kanan bawah dalam :  
98 perbandingan dan waktu 23ms

Kata ANTARES ditemukan pada koordinat (13,6) mendatar ke kiri dalam :  
228 perbandingan dan waktu 24ms

Kata ELNATH ditemukan pada koordinat (13,7) mendatar ke kanan dalam :  
199 perbandingan dan waktu 25ms

Kata REGULUS ditemukan pada koordinat (14,20) vertikal ke bawah dalam :  
197 perbandingan dan waktu 25ms

Kata VEGA ditemukan pada koordinat (15,11) vertikal ke bawah dalam :  
116 perbandingan dan waktu 26ms

Kata RIGEL ditemukan pada koordinat (15,16) mendatar ke kiri dalam :  
200 perbandingan dan waktu 27ms

Kata ARCTURUS ditemukan pada koordinat (18,1) diagonal ke kanan bawah dalam :  
302 perbandingan dan waktu 28ms

Kata ALDEBARAN ditemukan pada koordinat (18,8) diagonal ke kiri bawah dalam :  
322 perbandingan dan waktu 29ms

Kata HADAR ditemukan pada koordinat (20,9) mendatar ke kanan dalam :  
177 perbandingan dan waktu 30ms

Kata ALTAIR ditemukan pada koordinat (20,21) diagonal ke kanan bawah dalam :  
351 perbandingan dan waktu 31ms

Kata POLLUX ditemukan pada koordinat (20,25) vertikal ke atas dalam :  
186 perbandingan dan waktu 32ms

Kata SPICA ditemukan pada koordinat (21,20) diagonal ke kiri atas dalam :  
254 perbandingan dan waktu 33ms

Kata SHAULA ditemukan pada koordinat (21,28) mendatar ke kiri dalam :  
257 perbandingan dan waktu 34ms

Kata SIRIUS ditemukan pada koordinat (23,0) vertikal ke atas dalam :  
276 perbandingan dan waktu 34ms

Kata ALNAIR ditemukan pada koordinat (23,5) diagonal ke kiri bawah dalam :  
423 perbandingan dan waktu 35ms

Kata ADHARA ditemukan pada koordinat (24,19) diagonal ke kiri bawah dalam :  
437 perbandingan dan waktu 36ms

Z B E Y D A J N Y S P E A A E F N H E E I F S E B S N C O R  
L V Z B M J C Z L R S Q C L N A N O S S T V F C U X N N R C  
R I G I L K E N T U M S R N M B Y R Y U R E Q P N R P R Y K  
K T Z A G S Y T E C C F U I E I E Y A C H U O A M R F N C I  
X U Q T H I P G U X Z A X L H F A H B M O N W U H Q O Y C D  
W L Q J Y Q L T I S D Y I A R F L P F P A R Z Q V P I T P Q  
H G X Y G E T R F V G Q V M G A D E L C W X P F Z G V Y S T  
K P A E T Q T O K Q F A N R M F N G T A T Q N D Q K M C L A  
S J O E N A V B Z C D Q T O H T K Y O L C W E Z G A U J A P  
E Z B C L B N F H Q M N F L W R O N W S W I A X J B N Y E V  
J H D L G N Y F O G T H B K T B R O V I O U D R B E X L S J  
G X E V B C U J J T X Q Z J A B M A R H C G F U Z Q L U Z L  
B B S R W D T E B U O O R A M D F E X T W Y D D S A V H D K  
S E R A T N A E L N A T H M K I G Y M A Y E K J L N Y L Z W  
K Y S G U U M J C C S X D Y H G M S O N R S O Y N P P I Y S  
M V E Q B W F F U G T V L E G I R O F W E N U R E X V U X S  
U M R L P H Y R B A Z E C M O C R U S I G E Q O K U M T V H  
V L P G Q D F A U U L G J Y T R A M I A U F A N U L C P G B  
S A M Y D Q X H A K N A P T Q T N C T U L E Y Y S L Z N W U  
U U R T Q F X L K X H D L Z K F U X I L U D Q B J O P E R M  
I C Q C P W D S T H A D A R D B Q N Y P S A Q T V P T J U N  
R S R C T E Q P O V Z N G G J S Q T R H S X L A L U A H S G  
I Y B G B U A F H W H A T G A J M R G R P E X T H V V S N L  
S Q S A Z A R Z V P M N Q S M R S L N X P W D P A D P X H W  
C H R J L I J U T Y O J Q L S R M F P A V P L D R I L U Z R  
W A W N P L R D S N N C C K R M E O D D K K K O F T R J B X  
N G A U W F Y U W Q L C S W D B A H C P R E J V P S L B A R  
U I L D M U V X S V V Q I Z P A A Z W K J Y Q Q X H A D E N  
R D H U K R T S F U B M Y R Q R R K D Z Y R G U M M U U Y G  
K M F H Z F W Z Q F W Z F W A H G E P O P X T O I O H D G G

Total perbandingan yang dilakukan:

4723

Total waktu yang digunakan:

37ms


Total kata yang ditemukan :

25



Input(large3.txt):

```
test > ≡ large3.txt
1  B M Y W P X K M Z H J J T B I N Z B M S D K Z N J W D G J W
2  R E Y B T Q D H A Y Y N K R D M C V I S R A C H V C G W S W
3  M B H U A O S T N D F L A T D Y Z Y N Z E K Z J Z K L L L N
4  S V U M L A B S L L A W U H H D Y A A E I U Z W Y Z I Q K C
5  S K R E L N K A Z T A R H H N T Q Z T P Y Z A V U A V F O Y
6  I A H G F E X T U M D A A B O R G Q O T X U R R S N S Z Z U
7  C H S O K D E A I L Q F S G X H S V T Y E K D V D Y L Z Q O
8  T S F U B X O H C Z F J O Y G V M M B S Y U S U C L Q J O L
9  T Q H L K V O E E J J J X S Y P J M T O E D Q I N K Z Z E J
10 X I Q Z G E W U N U I U A T H L O Z R X F U F W A B I K H J
11 D D N Q N Z B O L M P R W L R X U U P L V T K M L J Y K I V
12 O U R G Z A S U G S U M B Z O L I L E Z G V M X R F Q B A O
13 D T L U F S N U C K X V V P B R H M C C M T J G U K U S S H
14 Q S I R C Y V O A U I M V M A O B J U H B J G C P U N I U K
15 O Y A B Y F C S K U U M X U O F R P Z K C L E E Y R U I C S
16 G Y G K O U F M J U Y R P G W V P U T V L N J K A V K C D D
17 O J U V T P C G R R P A K A M A R U T N K Q P M L C C A F S
18 H I N A T A U B B A G T S O E Y J J Y O H P Q T P E I N C J
19 C Q A A F H P L C M C C Y Q V X J S A B J X E H Z Y A I I X
20 D K S A S O R I B I L L K P G C N H Z N N M Y X E H W H R Y
21 E J A A O I V H I H W W E X Y P D I N T A H W D T B T S X J
22 I F W K B P L X F C U K A H V I G K K R C X X M H E Q U J M
23 D C N Q A A P F S O O B I N S U M A I A D U N W T T A K X Z
24 A J A S F S H S V R Z B A B Y K N M N U O H G A S Q S N S L
25 R O D B F V H H J O M R D G A I F A P L E K H O S Z V F K A
26 A F I Z O N B I K S U Z K N J N D R U Y J W G H W C X V S A
27 F P H V J U P A E T Q O P H E L A U I Y Y M P M V Q J O J S
28 F P W N F I R R O I J E N J A C F H Q C G R R U Y C F A D E
29 H W I Q V I V O O P J C Y D I E Z H F T A M W T Z G D H J W
30 W U Q J N U A H V Z H O F O P T P H C N G D E V W C H B O G
31 W B S H V H F S T U H S V J U R G B X Y N K T T B Z J J H A
32 T X O L S M C R Q U P G X Z Q Q E I K K Z H V K C T Y I B W
```



```
34 AKAMARU
35 BORUTO
36 DEIDARA
37 GUY
38 HAKU
39 HANABI
40 HIDAN
41 HIMAWARI
42 HINATA
43 KAKASHI
44 KAKUZU
45 KARIN
46 KIBA
47 KONAN
48 KUSHINA
49 KYUUBI
50 LEE
51 MADARA
52 MINATO
53 NARUTO
54 NEJI
55 OBITO
56 OROCHIMARU
57 SAKURA
58 SASORI
59 SASUKE
60 SHIKAMARU
61 TEMARI
```

**Output:**

```
Kata MADARA ditemukan pada koordinat (0,7) diagonal ke kanan bawah dalam :
10 perbandingan dan waktu 8ms

Kata MINATO ditemukan pada koordinat (0,18) vertikal ke bawah dalam :
20 perbandingan dan waktu 14ms

Kata KAKUZU ditemukan pada koordinat (0,21) vertikal ke bawah dalam :
14 perbandingan dan waktu 15ms

Kata SASUKE ditemukan pada koordinat (4,0) diagonal ke kanan bawah dalam :
54 perbandingan dan waktu 16ms

Kata LEE ditemukan pada koordinat (4,4) diagonal ke kanan bawah dalam :
70 perbandingan dan waktu 17ms

Kata HIMAWARI ditemukan pada koordinat (7,7) diagonal ke kanan atas dalam :
111 perbandingan dan waktu 19ms

Kata KIBA ditemukan pada koordinat (9,27) mendatar ke kiri dalam :
108 perbandingan dan waktu 21ms

Kata BORUTO ditemukan pada koordinat (12,14) diagonal ke kanan bawah dalam :
118 perbandingan dan waktu 22ms
```

Kata SAKURA ditemukan pada koordinat (14,7) diagonal ke kanan atas dalam :  
214 perbandingan dan waktu 23ms

Kata KONAN ditemukan pada koordinat (14,8) diagonal ke kiri atas dalam :  
166 perbandingan dan waktu 24ms

Kata OBITO ditemukan pada koordinat (15,4) diagonal ke kiri atas dalam :  
185 perbandingan dan waktu 25ms

Kata KYUUBI ditemukan pada koordinat (15,23) diagonal ke kanan atas dalam :  
197 perbandingan dan waktu 25ms

Kata AKAMARU ditemukan pada koordinat (16,11) mendatar ke kanan dalam :  
232 perbandingan dan waktu 27ms

Kata HINATA ditemukan pada koordinat (17,0) mendatar ke kanan dalam :  
166 perbandingan dan waktu 28ms

Kata TEMARI ditemukan pada koordinat (17,23) diagonal ke kiri bawah dalam :  
202 perbandingan dan waktu 30ms

Kata SHIKAMARU ditemukan pada koordinat (18,17) vertikal ke bawah dalam :  
236 perbandingan dan waktu 32ms

Kata DEIDARA ditemukan pada koordinat (19,0) vertikal ke bawah dalam :  
150 perbandingan dan waktu 34ms

Kata KAKASHI ditemukan pada koordinat (19,1) diagonal ke kanan bawah dalam :  
235 perbandingan dan waktu 35ms

Kata SASORI ditemukan pada koordinat (19,2) mendatar ke kanan dalam :  
250 perbandingan dan waktu 35ms

Kata HAKU ditemukan pada koordinat (21,13) mendatar ke kiri dalam :  
265 perbandingan dan waktu 37ms

Kata GUY ditemukan pada koordinat (21,16) diagonal ke kiri bawah dalam :  
171 perbandingan dan waktu 38ms

Kata NARUTO ditemukan pada koordinat (22,13) diagonal ke kiri bawah dalam :  
232 perbandingan dan waktu 39ms

Kata KUSHINA ditemukan pada koordinat (22,27) vertikal ke atas dalam :  
300 perbandingan dan waktu 41ms

Kata OROCHIMARU ditemukan pada koordinat (24,9) vertikal ke atas dalam :  
275 perbandingan dan waktu 43ms

Kata KARIN ditemukan pada koordinat (25,8) diagonal ke kiri bawah dalam :  
336 perbandingan dan waktu 44ms

Kata HIDAN ditemukan pada koordinat (26,2) vertikal ke atas dalam :  
336 perbandingan dan waktu 45ms

Kata NEJI ditemukan pada koordinat (27,12) mendatar ke kiri dalam :  
300 perbandingan dan waktu 46ms

Kata HANABI ditemukan pada koordinat (27,17) diagonal ke kiri atas dalam :  
365 perbandingan dan waktu 47ms



B M Y W P X K M Z H J J T B I N Z B M S D K Z N J W D G J W  
 R E Y B T Q D H A Y Y N K R D M C V I S R A C H V C G W S W  
 M B H U A O S T N D F L A T D Y Z Y N Z E K Z J Z K L L L N  
 S V U M L A B S L L A W U H H D Y A A E I U Z W Y Z I Q K C  
 S K R E L N K A Z T A R H H N T Q Z T P Y Z A V U A V F O Y  
 I A H G F E X T U M D A A B O R G Q O T X U R R S N S Z Z U  
 C H S O K D E A I L Q F S G X H S V T Y E K D V D Y L Z Q O  
 T S F U B X O H C Z F J O Y G V M M B S Y U S U C L Q J O L  
 T Q H L K V O E E J J J X S Y P J M T O E D Q I N K Z Z E J  
 X I Q Z G E W U N U I U A T H L O Z R X F U F W A B I K H J  
 D D N Q N Z B O L M P R W L R X U U P L V T K M L J Y K I V  
 O U R G Z A S U G S U M B Z O L I L E Z G V M X R F Q B A O  
 D T L U F S N U C K X V V P B R H M C C M T J G U K U S S H  
 Q S I R C Y V O A U I M V M A O B J U H B J G C P U N I U K  
 O Y A B Y F C S K U U M X U O F R P Z K C L E E Y R U I C S  
 G Y G K O U F M J U Y R P G W V P U T V L N J K A V K C D D  
 O J U V T P C G R R P A K A M A R U T N K Q P M L C C A F S  
 H I N A T A U B B A G T S O E Y J J Y O H P Q T P E I N C J  
 C Q A A F H P L C M C C Y Q V X J S A B J X E H Z Y A I I X  
 D K S A S O R I B I L L K P G C N H Z N N M Y X E H W H R Y  
 E J A A O I V H I H W W E X Y P D I N T A H W D T B T S X J  
 I F W K B P L X F C U K A H V I G K K R C X X M H E Q U J M  
 D C N Q A A P F S O O B I N S U M A I A D U N W T T A K X Z  
 A J A S F S H S V R Z B A B Y K N M N U O H G A S Q S N S L  
 R O D B F V H H J O M R D G A I F A P L E K H O S Z V F K A  
 A F I Z O N B I K S U Z K N J N D R U Y J W G H W C X V S A  
 F P H V J U P A E T Q O P H E L A U I Y Y M P M V Q J O J S  
 F P W N F I R R O I J E N J A C F H Q C G R R U Y C F A D E  
 H W I Q V I V O O P J C Y D I E Z H F T A M W T Z G D H J W  
 W U Q J N U A H V Z H O F O P T P H C N G D E V W C H B O G  
 W B S H V H F S T U H S V J U R G B X Y N K T T B Z J J H A  
 T X O L S M C R Q U P G X Z Q Q E I K K Z H V K C T Y I B W

Total perbandingan yang dilakukan:

5406

Total waktu yang digunakan:

48ms

Total kata yang ditemukan :

28

Link

<https://github.com/IMYELI/Tucil-Word-Puzzle.git>

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