

LAPORAN TUGAS KECIL I

PENYELESAIAN WORD SEARCH PUZZLE

DENGAN ALGORITMA BRUTE FORCE

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah

IF2211 Strategi Algoritma



Disusun oleh:

Kristo Abdi Wiguna 13520058

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

DAFTAR ISI

DAFTAR ISI	1
Algoritma Brute Force	2
Source Program	3
Screenshot Input dan Output	6
Link Drive Kode Program	34
Checklist	34

Algoritma Brute Force

Algoritma penyelesaian word puzzle pertama dilakukan dengan membaca file yang berisi puzzle dan daftar jawaban yang perlu dicari. Pada saat membaca file teks tersebut, program akan membaca puzzle menjadi sebuah *vector of vector of char* dan membaca daftar jawaban menjadi *vector of string*. Lalu, terdapat 3 nested loop untuk mengiterasi karakter satu per satu dalam matriks huruf puzzle tersebut dan dicocokkan dengan urutan daftar jawaban. Ketika mengiterasi sebuah karakter, akan dicek setiap arah dengan menggunakan perulangan perhitungan delta dr dan dc dengan alur sebagai berikut.

- Ketika dr bernilai -1 dan dc bernilai -1, maka akan memanggil prosedur check untuk mengecek arah diagonal kiri bawah dari sebuah karakter,
- Ketika dr bernilai -1 dan dc bernilai 0, maka akan memanggil prosedur check untuk mengecek arah horizontal kiri dari sebuah karakter,
- Ketika dr bernilai 0 dan dc bernilai 1, maka akan memanggil prosedur check untuk mengecek arah vertikal atas dari sebuah karakter, dan seterusnya.

Prosedur check akan mengecek kesamaan karakter puzzle dengan karakter jawaban ke-i dengan menggunakan perulangan. Jika out of bounds, maka perulangan akan terhenti dan akan kembali pada program utama untuk lanjut mengecek karakter selanjutnya. Jika tidak out of bounds dan sama, maka akan lanjut mengecek karakter selanjutnya sesuai arah yang sudah ditentukan oleh variabel dr dan dc serta mengubah karakter yang benar sementara menjadi ‘1’ untuk penanda.

Jika benar semua akan lanjut pada perulangan dimana karakter yang bukan ‘1’ akan diubah menjadi ‘-’ dan mengubah ‘1’ menggunakan karakter asli pada matriks huruf salinan dari matriks awal. Setelah itu, matriks yang hanya menampilkan huruf di puzzle yang sesuai jawaban akan ditampilkan di output.

Algoritma yang digunakan tidak menggunakan sifat heuristik. Langkah dimana komparasi tidak sesuai dan akan dilanjutkan, adalah sebuah *improvement* dimana dapat mengurangi waktu eksekusi dan tetap akurat serta bukan sebuah fungsi heuristik dimana ada tradeoff ketidakakuratan. Program sudah ditest di OS Windows dengan pemrograman C++14.

Source Program

Menggunakan bahasa pemrograman C++ dengan pertimbangan library yang sudah ada dan runtime yang cepat.

```
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
#include <vector>
#include <chrono>
using namespace std;
using namespace std::chrono;

typedef vector<vector<char>> matrix;

matrix loadMatrix(ifstream& file);
vector<string> loadAnswers(ifstream& file);
void printMatrix(matrix mat);
void check(matrix mat, string answer, int r, int c, int dr, int dc);
string clean(string s);

int counter;
```

```

int main() {
    string filename;
    char isPuzzleDone;
    do {
        cout << "======" << endl;
        cout << "Masukkan nama file : ";
        cin >> filename;
        string filedir = "../test/" + filename;
        ifstream file(filedir.c_str());
        matrix board;
        board = loadMatrix(file);
        printMatrix(board);
        vector<string> answers = loadAnswers(file);
        auto start = high_resolution_clock::now();
        for (const string& answer : answers) {
            for (int i = 0; i < board.size(); i++) {
                for (int j = 0; j < board[i].size(); j++) {
                    for (int dr = -1; dr <= 1; dr++) {
                        for (int dc = -1; dc <= 1; dc++) {
                            if (dr == 0 && dc == 0) continue;
                            check(board, answer, i, j, dr, dc);
                        }
                    }
                }
            }
        }
        auto stop = high_resolution_clock::now();
        auto duration = duration_cast<microseconds>(stop - start);
        cout << "Time taken by program: " << duration.count() << "seconds" << endl;
        cout << "Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: " << counter << " kali" << "\n\n";
        file.close();
    isPuzzleDone = 'n';
    cout << "Apakah anda ingin memecahkan puzzle lain? (y/n)" << endl;
    cin >> isPuzzleDone;
} while ((isPuzzleDone == 'Y') || (isPuzzleDone == 'y'));
cout << "======" << endl;
cout << "word search puzzle solver by Kristo Abdi 13520058." << endl;

```

```

matrix loadMatrix(ifstream& file) {
    string line;
    matrix mat;
    while (getline(file, line)) {
        line = clean(line);
        if (line.length() == 0) break; //EOF

        vector<char> row;
        for (const char& cc : line) {
            if (cc >= 'A' && cc <= 'Z') {
                row.push_back(cc);
            }
        }
        mat.push_back(row);
    }

    return mat;
}

vector<string> loadAnswers(ifstream& file) {
    vector<string> answers;
    string line;
    while (getline(file, line)) {
        if (line.length() == 0) break;
        answers.push_back(clean(line));
    }
    return answers;
};

```

```

void printMatrix(matrix mat) {
    for (int i = 0; i < mat.size(); i++) {
        for (int j = 0; j < mat[i].size(); j++) {
            cout << mat[i][j] << " ";
        }
        cout << endl;
    }
}

string clean(string s) {
    // sometimes the file can have CRLF ending, so we need to clean the string first before using it.
    if (s.length() == 0) return s;
    if (s[s.length()-1] != '\x0D') return s;
    return s.substr(0, s.length()-1);
}

void check(matrix mat, string answer, int r, int c, int dr, int dc) {
    bool correct = true;
    matrix res = mat;
    matrix temp = mat;
    for (int i = 0; i < answer.length(); i++) {
        int ri = r + dr * i;
        int ci = c + dc * i;
        if (
            ri < 0 || ri >= mat.size() || // overflow vertically
            ci < 0 || ci >= mat[ri].size() || // overflow horizontally
            mat[ri][ci] != answer[i]
        ) {
            correct = false;
            break;
        } else {
            temp[ri][ci] = '1';
        }
        counter++;
    }

    if (!correct) {
        return;
    }

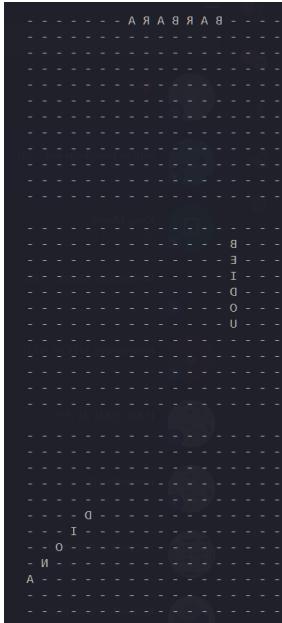
    for (int i = 0; i < mat.size(); i++) {
        for (int j = 0; j < mat[i].size(); j++) {
            if (temp[i][j] != '1') {
                res[i][j] = '-';
            }
        }
    }
    printMatrix(res);
    cout << endl;
}

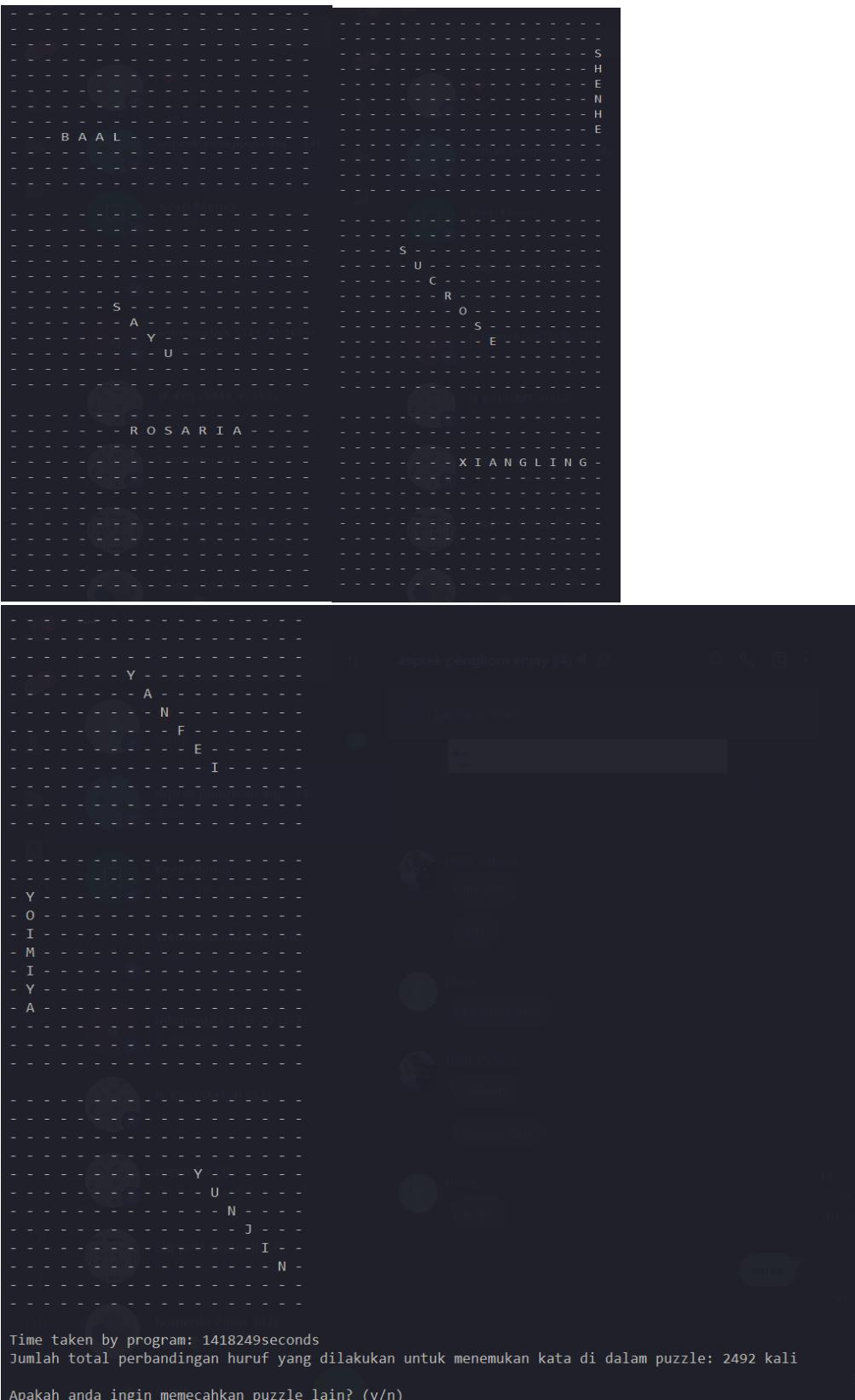
```

Screenshot Input dan Output

Test 1. (Kecil 12x18)

```
D:\Tucil\word-search-puzzle-solver\src (master -> origin)
$ main
=====
Masukkan nama file : small1.txt
Y T I K B A R B A R A N O E L L E Q
Y I J B S S M R O S A R I A D Q M F
B Y V E S N K R O N Y H U T A O A S
I O Q I L U I Y X I A N G L I N G H
P I K D I N C N A B U Y R E U L A E
K M S O S B A R G N Q T U D U Z K N
E I G U A U S M O G F C H N I D M H
Q Y A W V Z H A B S U E Q K J O R E
I A N B A A L C Y E E A I I L I N S
N P Y N L J Y K Q U R O N K Q E N A
G I U S J O Z F I S C H L G C I E F
J E A N S K K O K O M I A Y A K A Z
-
-
-
-
-
-
A
M
B
E
R
-
AYAKA-
```





Test 2. (Kecil 14x14)

```
Apakah anda ingin memecahkan puzzle lain? (y/n)
y
=====
Masukkan nama file : small2.txt
W H K P G S D G U C S P S S
T H G I M U M W L O M S I U
H V Z W O I L U U A E R Q R
H S A B A S M R G R N A Q U
T R H H E L C Z P Q I C M A
H W A R P E D M F U S S E S
R S U Q L C O G L T Q O W E
L A A E Z C W A R M R G L H
H D S L E F O O M V G E H T
F S Q D F U P U T H F N A L
D E S P I S I N G T T U D A
I A Y C S X W U Y L X S L U
O N F A T Q R E J Y G C A G
O G P S L I M E N R Q H Q H
```

H S A B A

A
M
H
T
S
A

S
U
I
S
L
E
C
-
C
E
D

S
S
E
R
P
M
O
-
C
E
D

DESPISING

T H G I M
- - -
H -
- S
- A -
- L -
- F -
- S -
- W -
- E -
- N -
- - - R
- - - A
- - - C
- - - S
- - - O

T
R
O
P
S
A
P - S L I M E -

S
M
U
G

S
O
U
R
C
E
L
E
S
T

- W A R P E D -

Time taken by program: 751485seconds

Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 10859 kali

Test 3. (Kecil 16x16)

```
Apakah anda ingin memecahkan puzzle lain? (y/n)
y
=====
Masukkan nama file : small13.txt
H Z D E N Q N R E F U J E Y K R
U C Y P H E R A N U T Z Y N E F
V R A T W L M W O U A D I B F S
K Q O E T X Q O T R W F M R D J
Q G U Y R D R A S B G A B K D Q
R E Y N A B N M M E H U M C W K
O S R E P I V G I C P R H M I X
C K T J M Q W D R F B X D L P I
J Y E I G O M A B Q K V L U C N
F E T Y D R L V C T Y J I C R E
E S T G Q B S N V S O H J R T O
Q K A T X O K X O Y C N Q A J H
H A Z G V D J N O E N Q U A L P
B Y Y A E D V H P F H X O W V O
S O F S M M W O C N Z G F C D C
J Z A R T S A I I Q P E U Q Y
```

H
C
A
E
R
B

E
N
O
T
S
M
I
R
B

R
E
B
M
A
H
C

The image shows a black rectangular background. At the top left, the word "CYPHER" is printed in a white, sans-serif font. The rest of the image is filled with a series of thin, white dashed horizontal lines spaced evenly across the entire area.

J E T T

K
I
L
L
J
O
Y

A black background featuring a grid of white dashed horizontal lines spaced evenly apart. The word "NOEN" is centered in the lower half of the grid, written in white capital letters.

N
E
M
O

- X I N E O H P

E		
Z		
A		
R		
	R E Y N A	
		S
		A
		G
		E

		R E P I V
S		
K		
Y		
E		
	S	
	O	
	V	
	A	

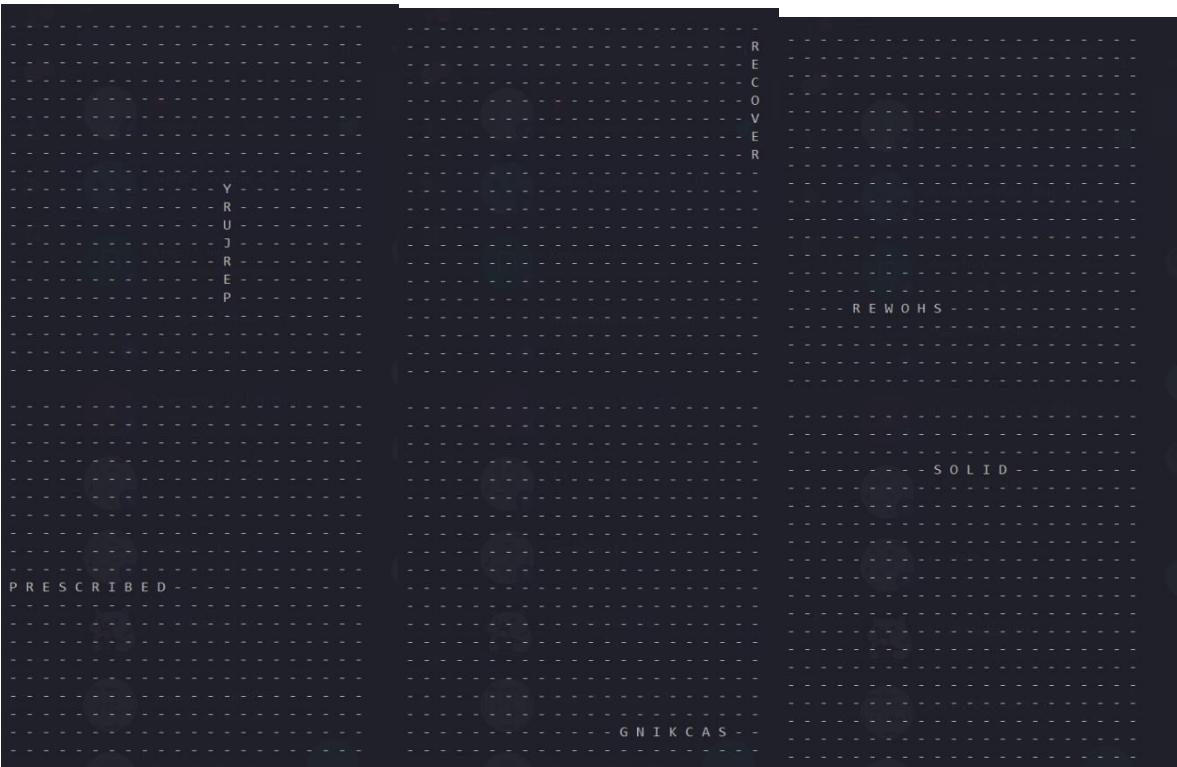
U	
R	
O	
Y	

Time taken by program: 1191985seconds

Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 12533 kali

Test 4. (Sedang 20x22)



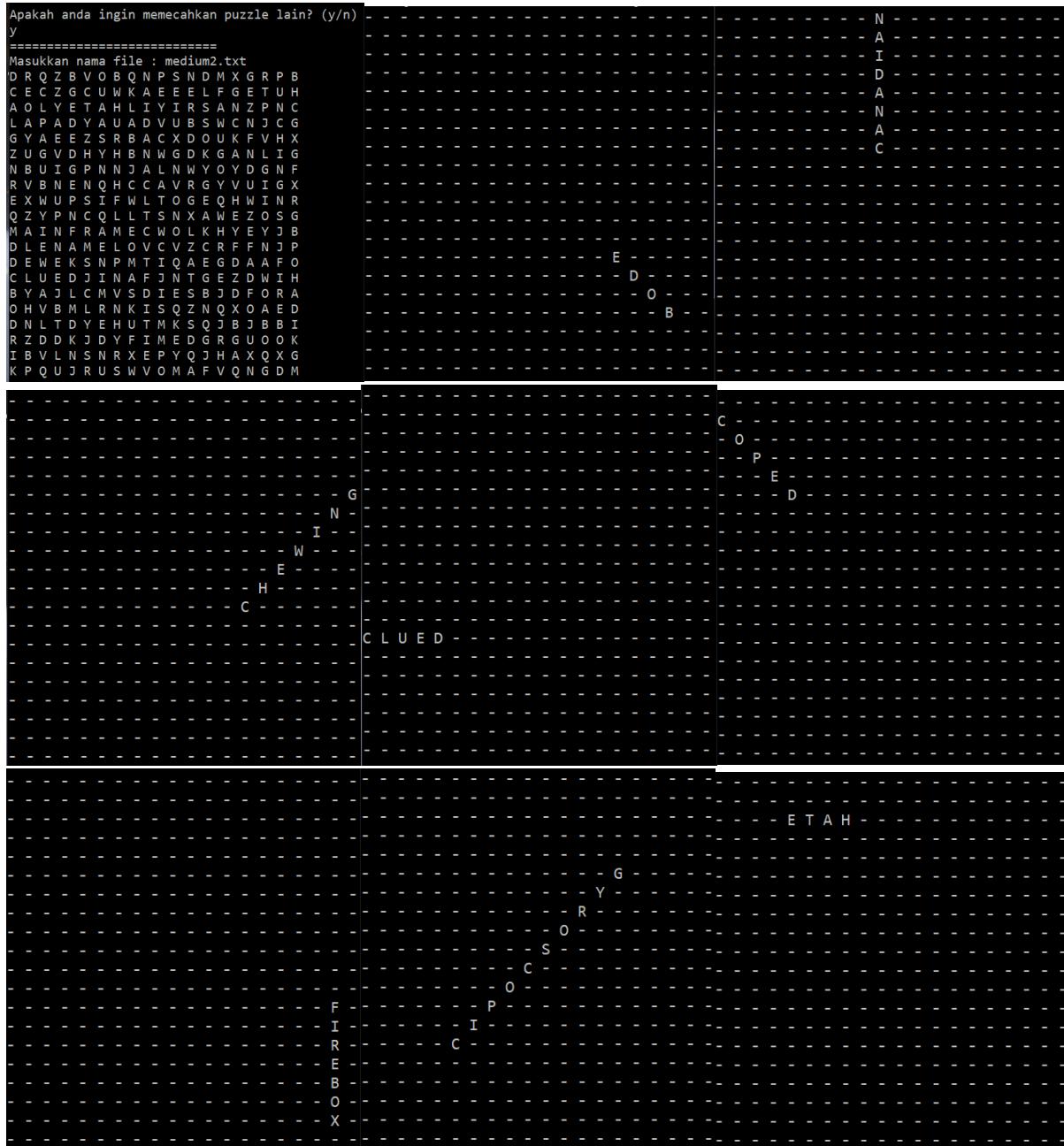


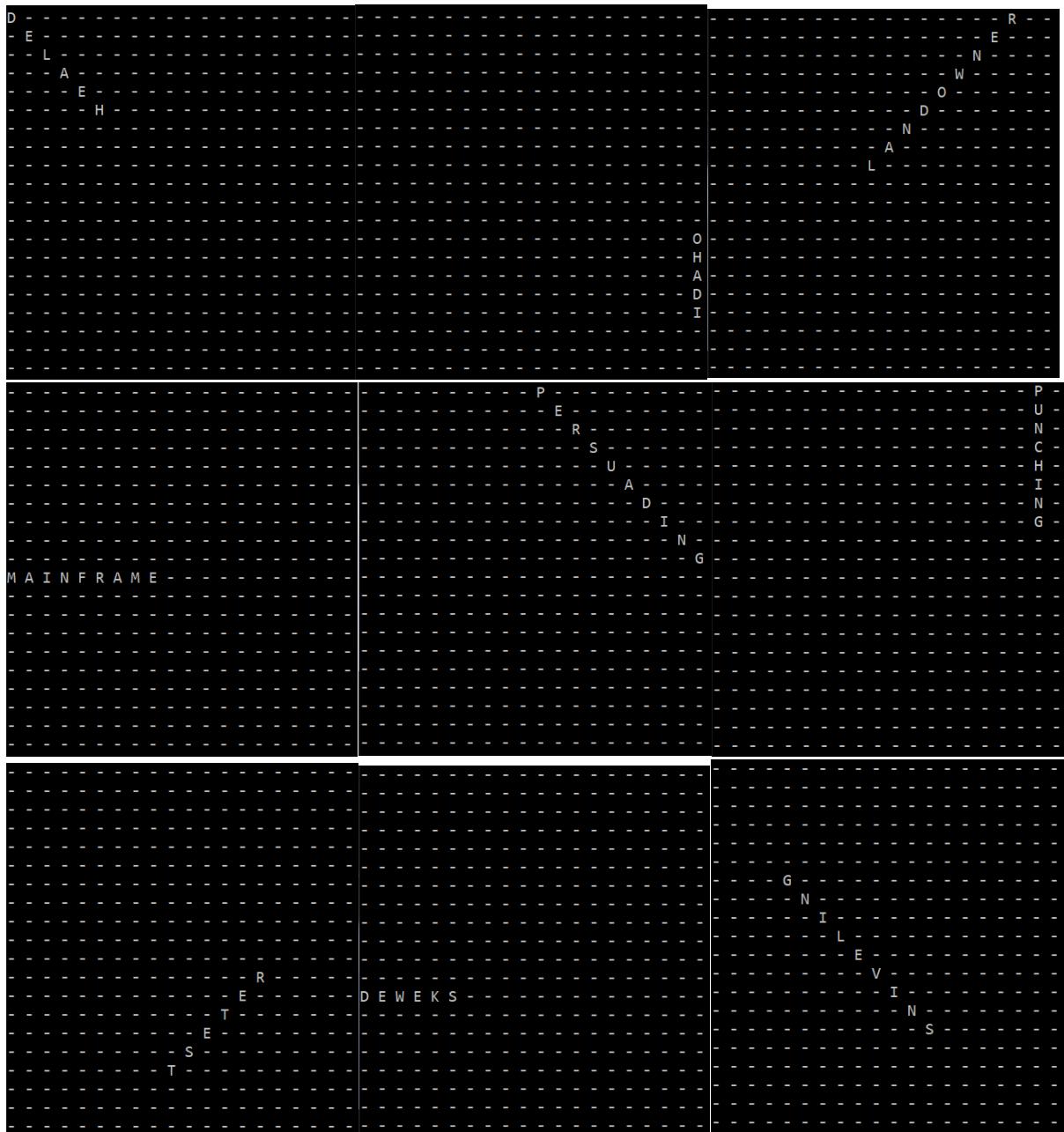
```
A  
L  
U  
T  
A  
P  
S  
  
- U R U G U A Y -  
  
Time taken by program: 2406829seconds  
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 3505 kali  
Apakah anda ingin memecahkan puzzle lain? (y/n)  
n  
=====
```

TOTALLY

Word search puzzle solver by Kristo Abdi 13520058.

Test 5. (Sedang 20x20)





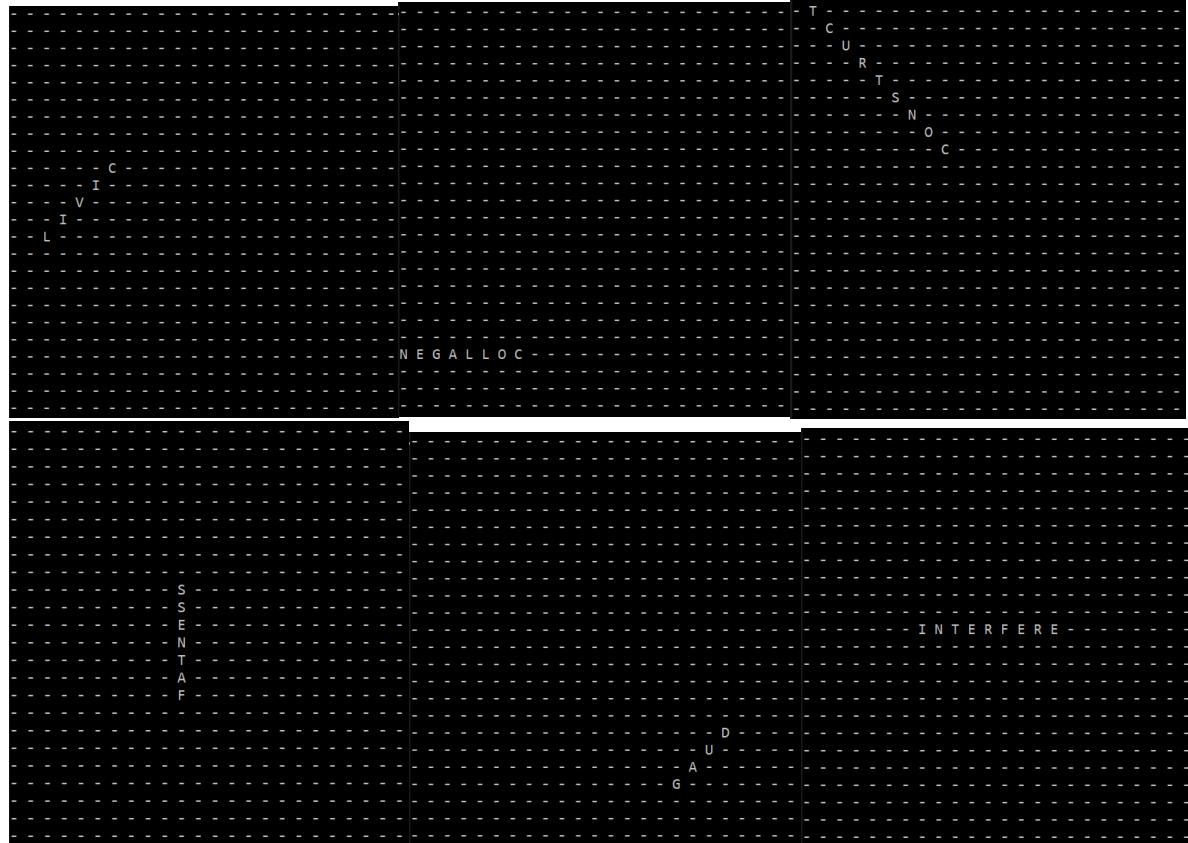
		D	
		E	
		I	
		V	
- E -			
- N -			
- I -			
- M -			
- R -			
- E -			
- D -			
- N -			
- U -			

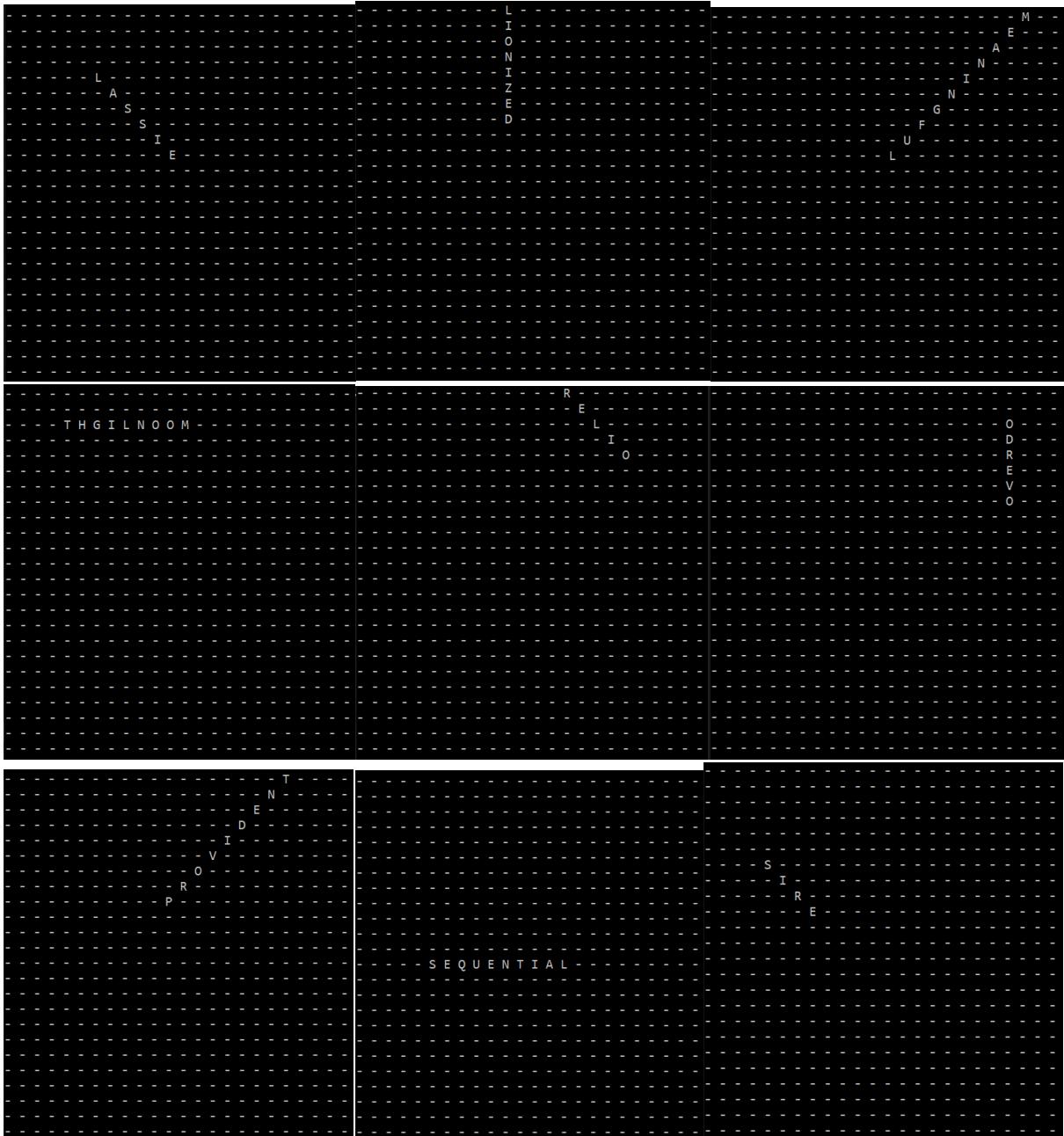
- E N A M E L -

Time taken by program: 1954958seconds
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 15162 kali

Test 6. (Sedang 24x24)

```
Apakah anda ingin memecahkan puzzle lain? (y/n)  
y  
=====  
Masukkan nama file : medium3.txt  
A T I D W Z M C T B L U Z W R F C W W T I M U T  
U L C A Q X G Z C F I F Y O Y E E Z N B E S E P  
X Q I U T H G I L N O O M T W E L E U A O B E L  
J E V E R S X Y S Z N J G C K T D I N G D H B J  
D I T R N T L G O A I A J D P I R I O D R P I A  
E D G M U A S A E E Z H A Z V K N A Y S E G D S S  
Q M N N S J T N S F E Y U O L G D L A Q V X U E  
D K O R T I D I O S D Y R M F A V T Q K O W R H  
D K S J R N O P G U B T X E A J K A N T R  
Z K N E B A C E Q N S E L L Y R X R C D H P E F  
W T C T U I M H N N S Y Z I D M N A B F C H E N  
G F L K V H W I N T E R F E R E B T Y B W N Y K  
B J P I E S E Q U E N T I A L Q R I J I W W F C  
Z M L I Q U N T A L T U U B L V T O A B G J L V  
Q C D I W R N M E P A R N O K R W N W M E H D O  
S B B J Y Z W O V C F N I U K M I L T A B H Z X  
C J H H E W M J M B K H F D D X M P I H W L K U  
B W F I A R D V O G Y S Y H B N J C Y D X E N Q  
Q X F O U D H H D O I X I U T Q B M U K X J H Z  
N C E W F F P Z N D M M M N L M E F A O R O G N R  
N E G A L L C C T M B G G O Y G Z E G J R O E  
G X T Y V X C S W X X S O P X O X N X S Y Y I I  
D G F V C H N T J N B A M U D N D W P T V A K C  
T J O Y U S M G F N S K K Q T H Y N O G G W V A
```





W
E
E
K
D
A
Y

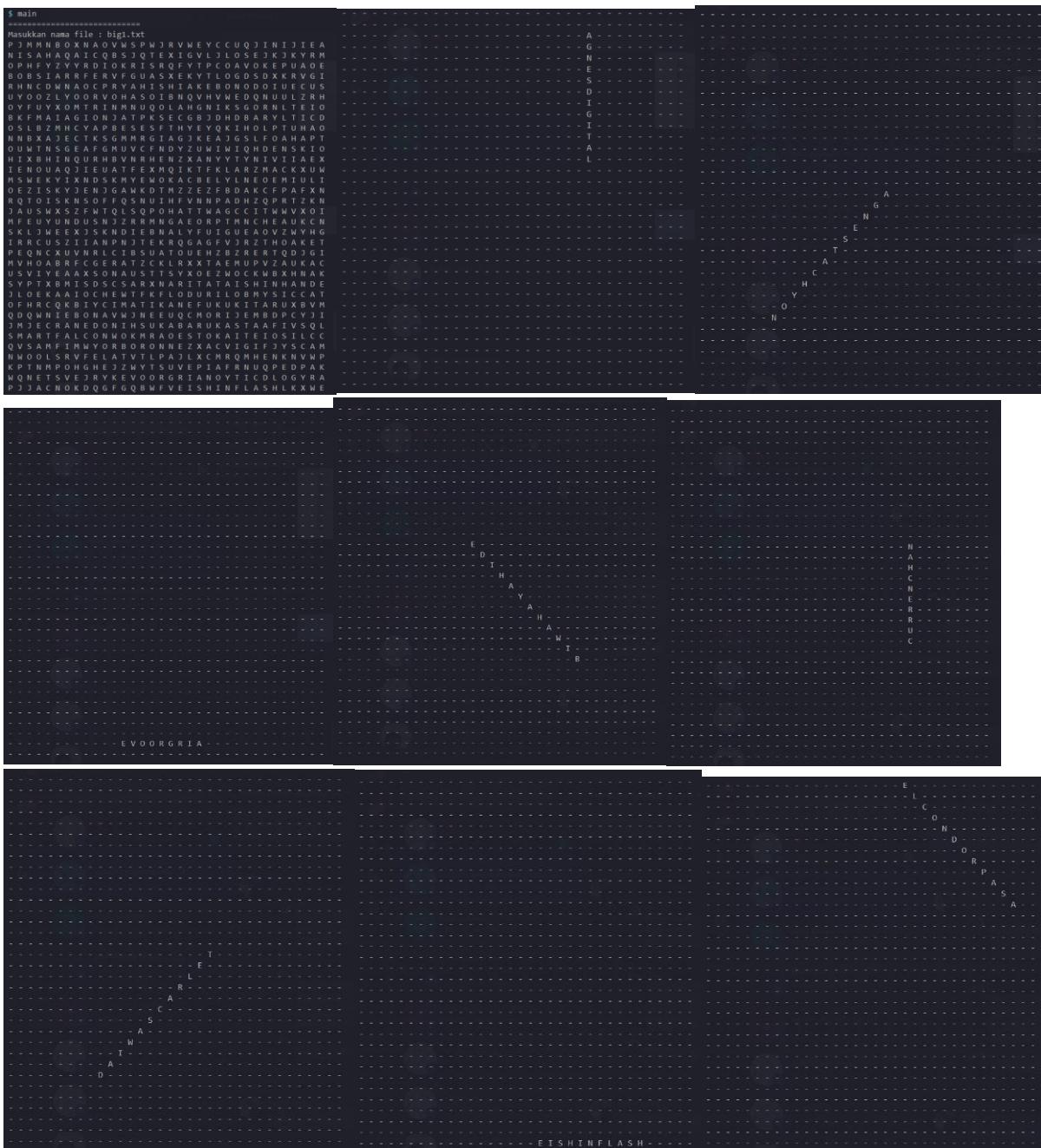
U
N
I
F
Y
I
N
G

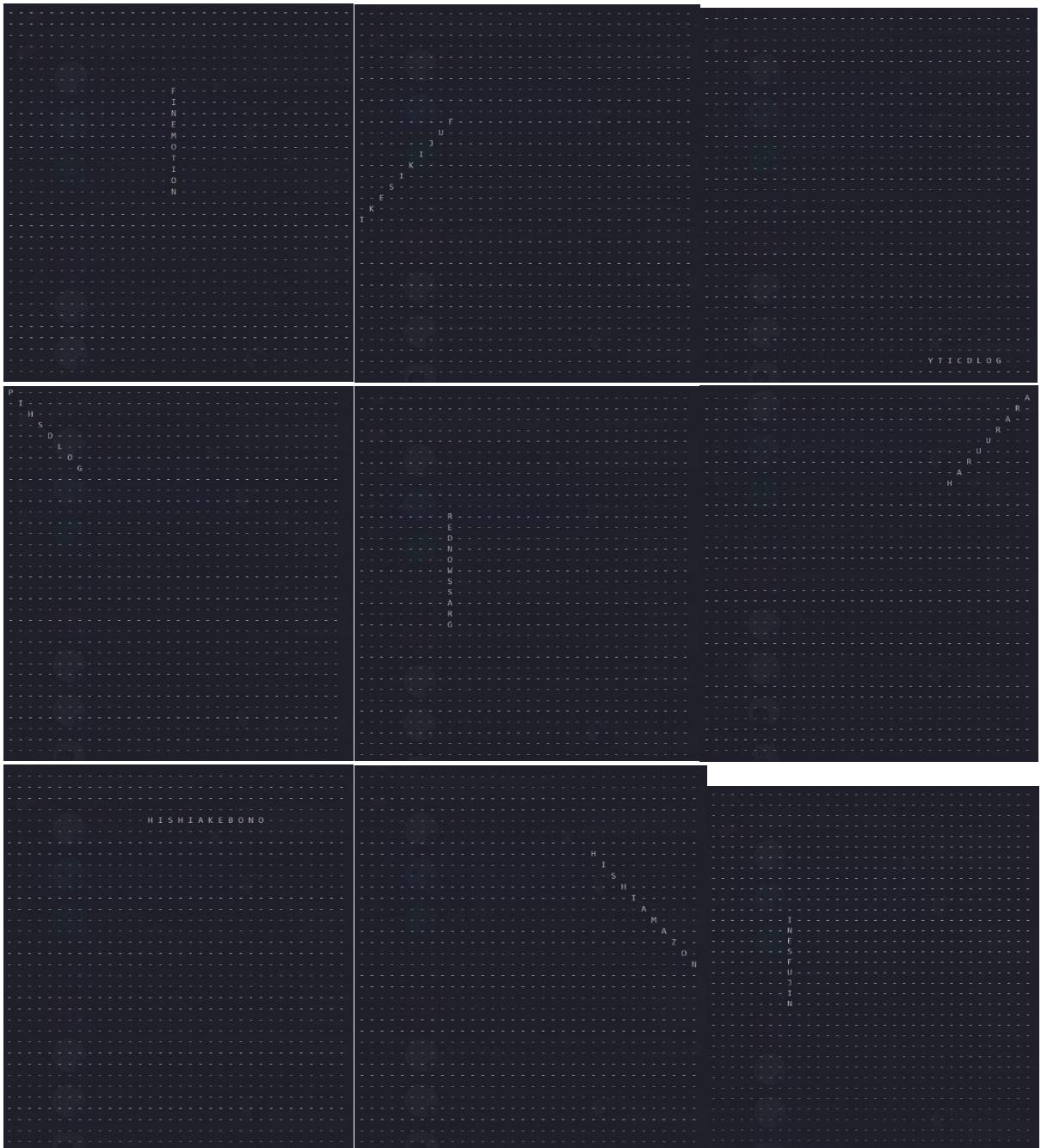
P
I
H
W

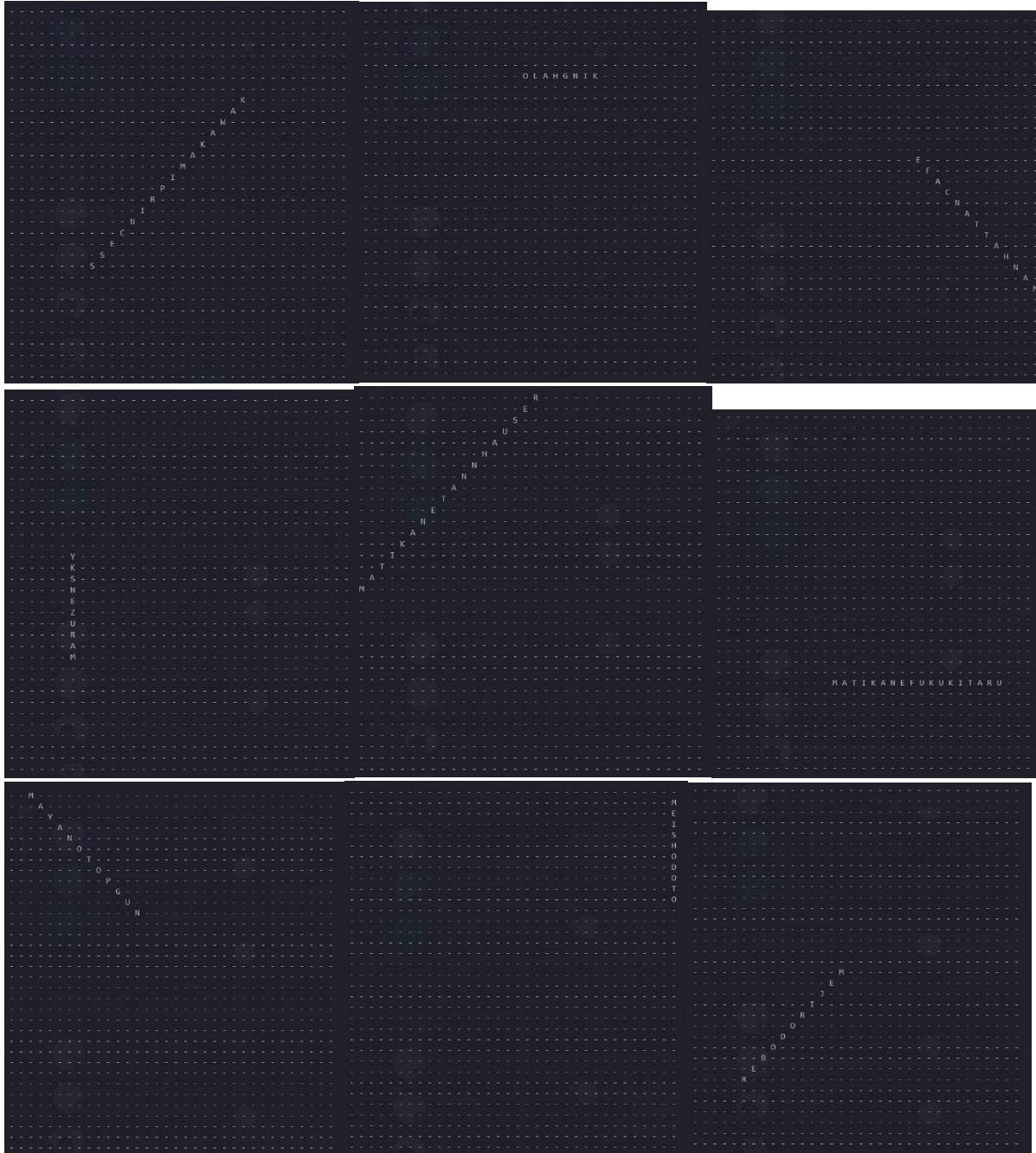
Time taken by program: 3153324seconds

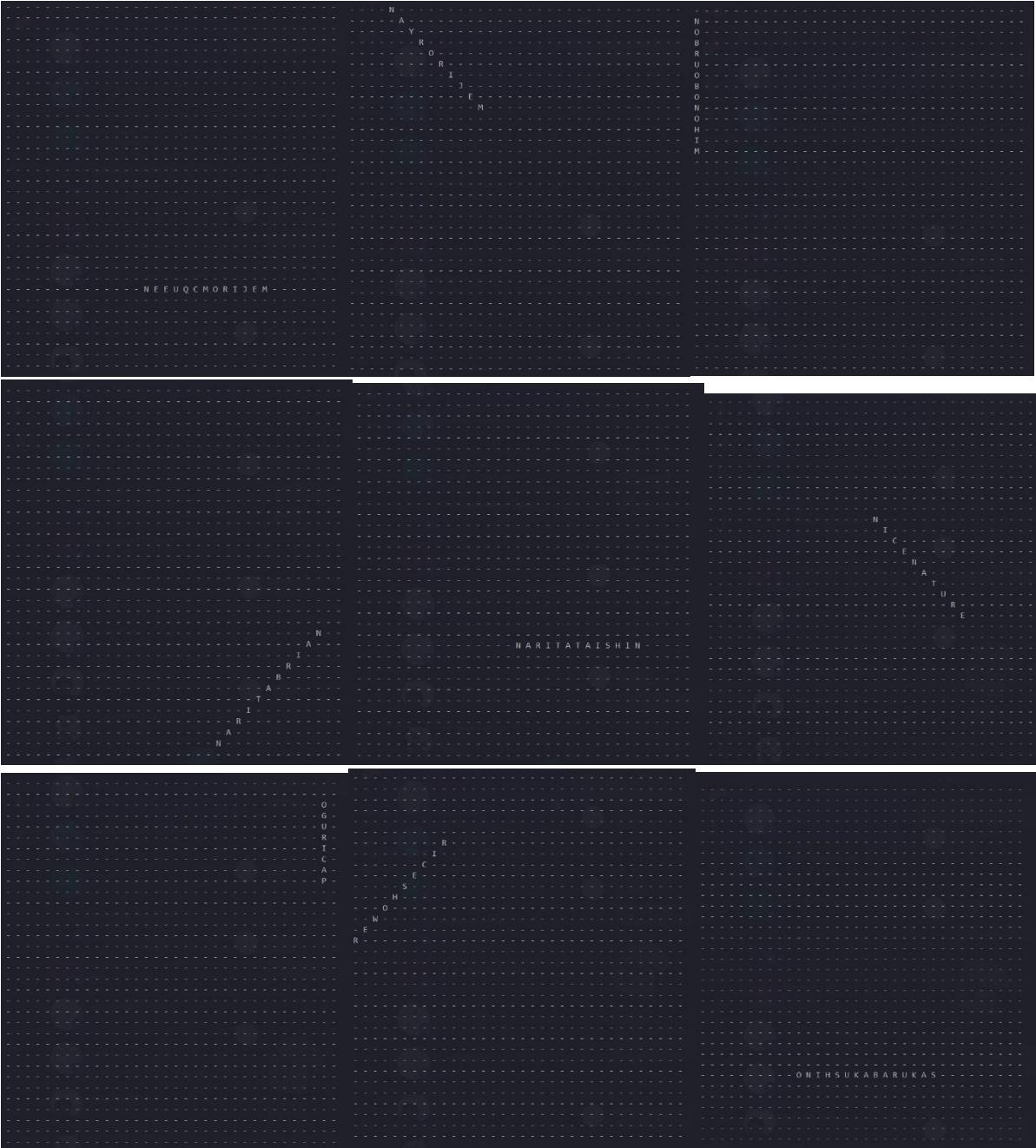
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 23353 kali

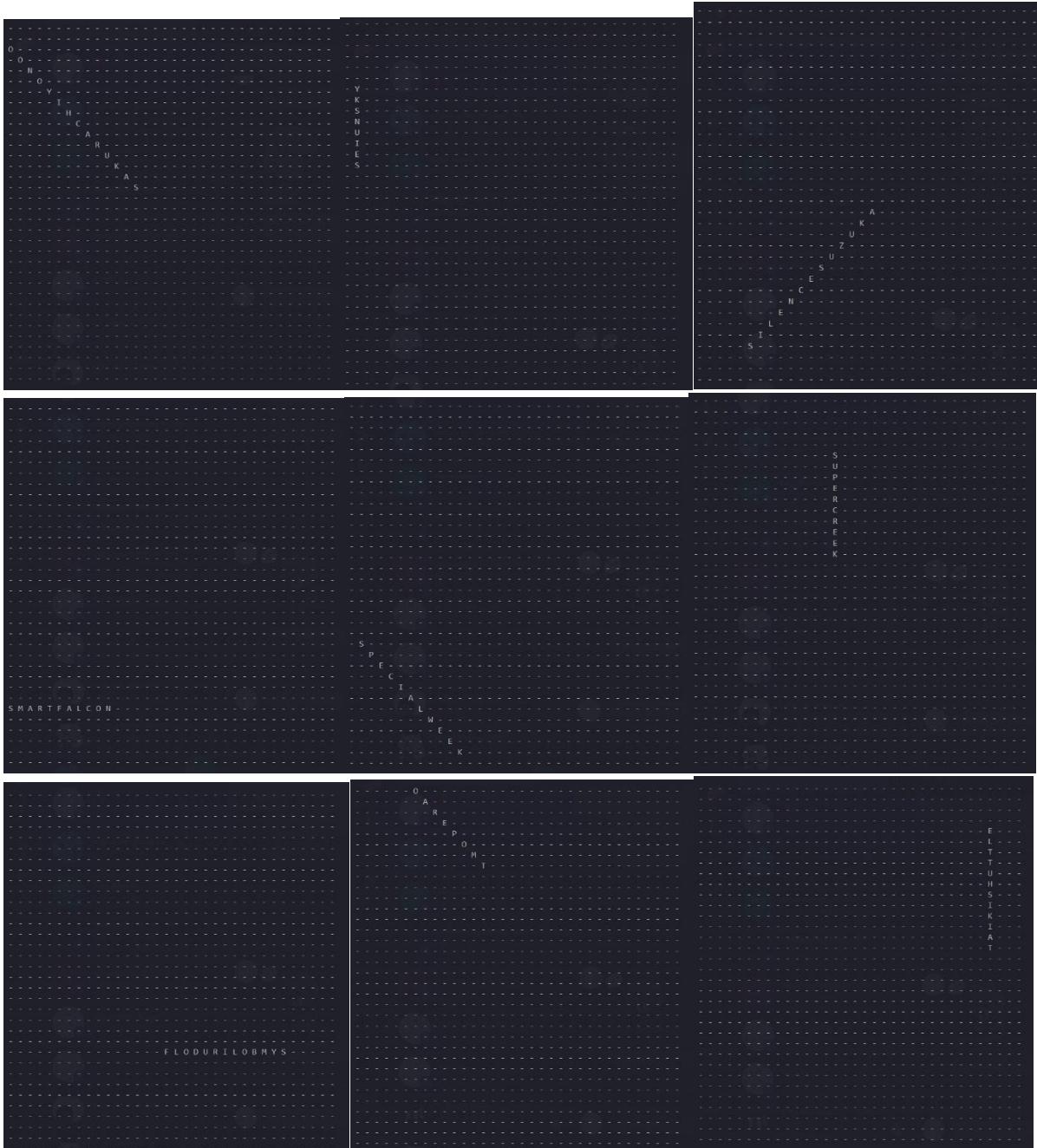
Test 7. (Besar 34x34)









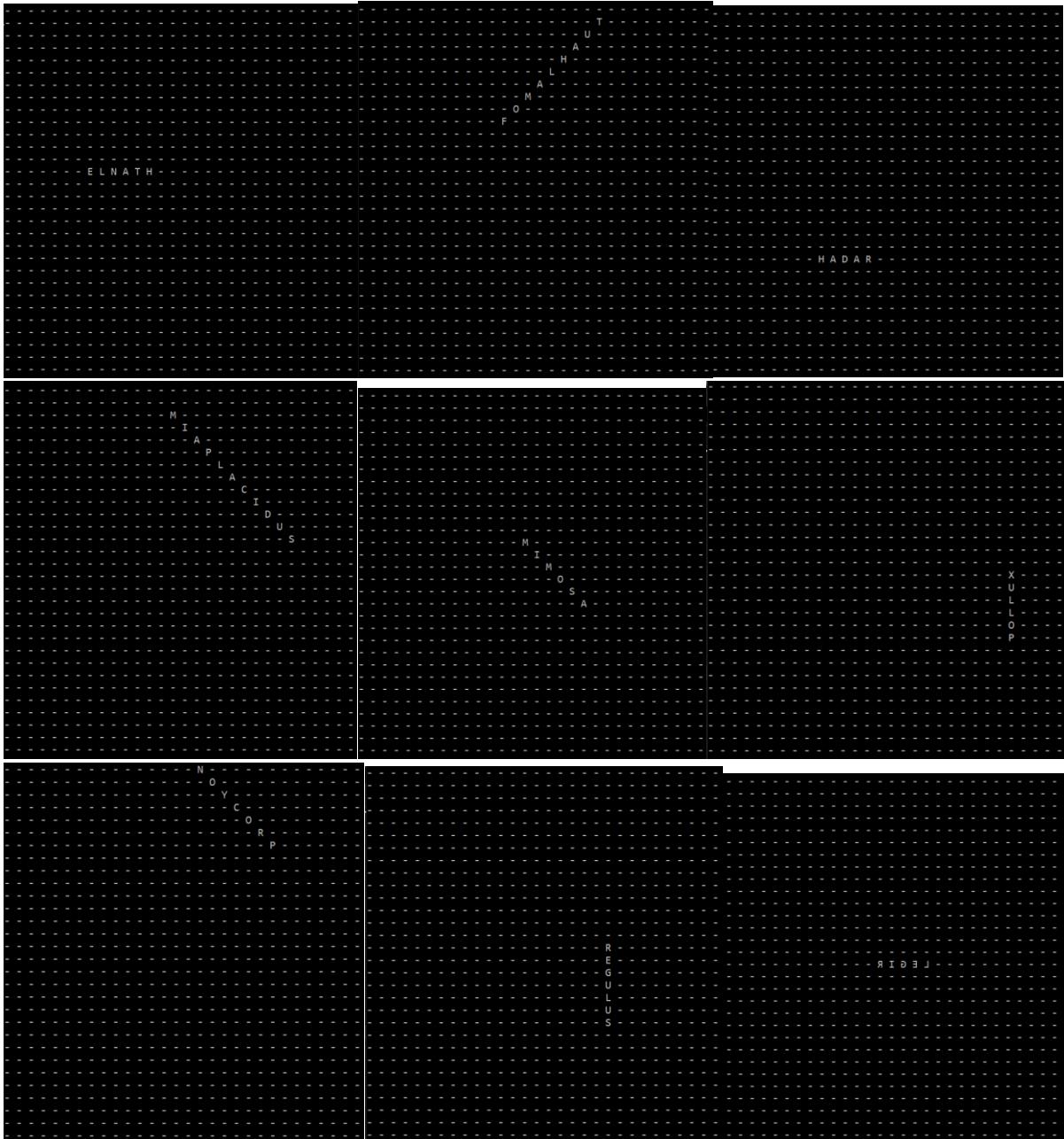




Y O R B O R O N N E Z

Time taken by program: 22176437seconds
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 21259 kali

Test 8. (Besar 30x30)



RIGILKENT

S
U
I
R
I
S
-

A L U A H S

V
E
G
A

A
C
I
P
S

Time taken by program: 16894724seconds
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 9237 kali

Test 9. (Besar 32x32)

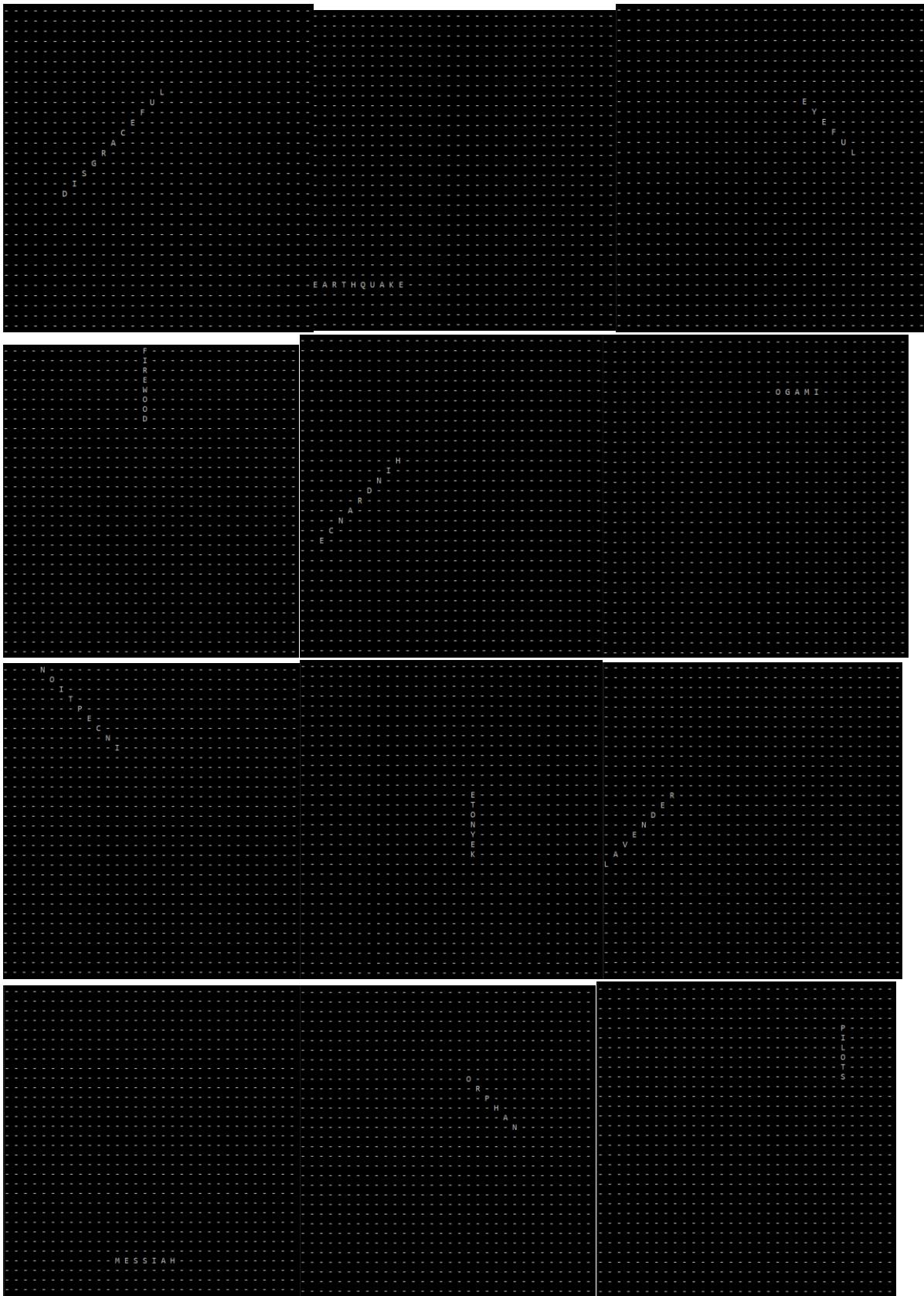
- ASSOCIATED

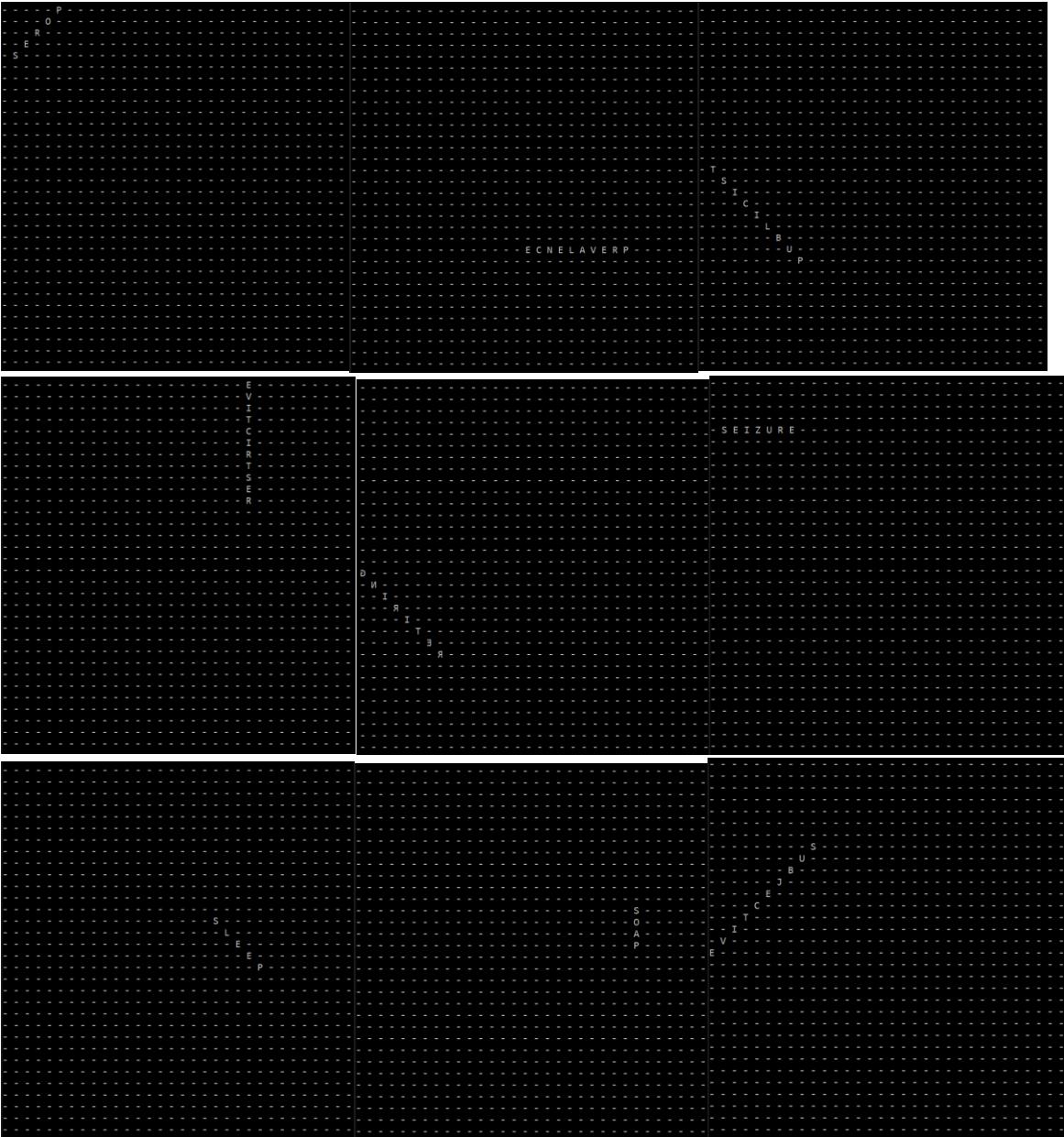
A
T
T
A
C
H
M
E
N
T

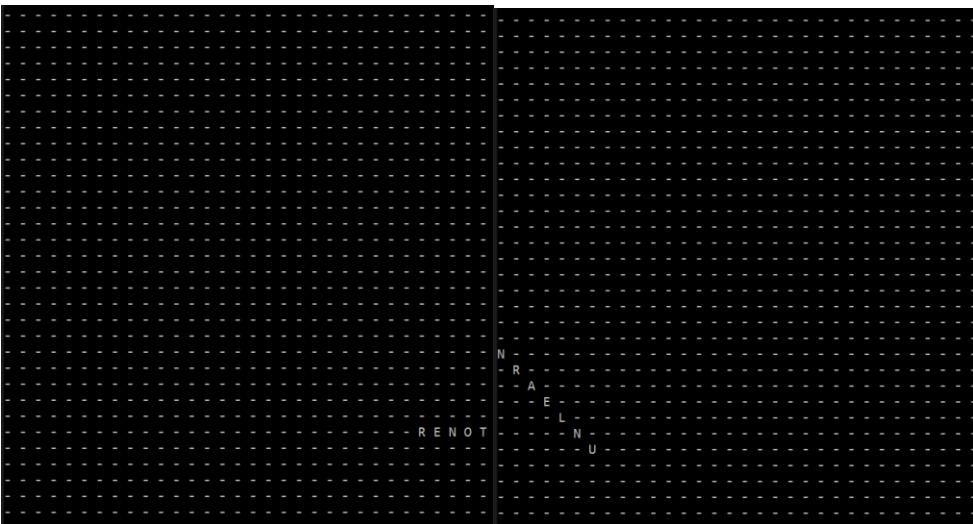
BANGKOK

U
T
I
O
U
S

YRELEC







Time taken by program: 10336717seconds

Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 11867 kali

Mohon maaf apabila ukuran tangkapan layar yang diperkecil, hal ini dilakukan agar dapat memangkas jumlah halaman yang ada agar tidak berlebihan.

Link Drive Kode Program

https://drive.google.com/drive/folders/1Z3QIil6msTq7ytK-AV4v5LDk-feMmBO_?usp=sharing

Checklist

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	