Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Penyelesaian *Word Search Puzzle* dengan Algoritma *Brute Force*

Disusun Oleh:

Ikmal Alfaozi 13520125

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2021/2022

Daftar Isi

A. Algoritma *Brute Force*

a. Algoritma Pencarian Kata pada Puzzle

1. Bentuk puzzle menjadi sebuah matriks berukuran $m \times n$.
2. Simpan kata ke dalam sebuah array yang mengandung informasi karakter, letak baris dan kolom huruf pada matriks, dan panjang katanya.
3. Telusuri matriks dari indeks paling kiri atas.
4. Jika karakter pada baris dan kolom matriks yang ditelusuri sesuai dengan huruf pertama kata, telusuri masing-masing delapan arah mata angin.
    a. Jika panjang karakter dari arah yang ditunjuk mata angin kurang dari panjang kata, lewati pemeriksaan arah mata angin tersebut.
    b. Jika panjang karakter dari arah yang ditunjuk mata angin lebih dari atau sama dengan panjang kata, bandingkan setiap karakter sampai:
        i. Semua karakter yang dibandingkan cocok dan simpan informasi baris dan kolom masing-masing karakter (pencarian berhasil).
        ii. Dijumpai karakter yang tidak sama (pencarian tidak berhasil).
5. Jika kata belum ditemukan dan indeks matriks bukan merupakan indeks paling akhir, periksa karakter pada indeks matriks berikutnya dan ulangi kembali langkah 3.

B. Kode Program

a. Modul bolean.h

Modul ini digunakan untuk mendefinisikan tipe boolean.

```
/* Definisi type boolean */

#ifndef _BOOLEAN_h
#define _BOOLEAN_h

#define boolean unsigned char
#define true 1
#define false 0

#endif
```

b. Modul main.h

Modul ini merupakan program utama yang melakukan proses *brute force*.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "boolean.h"

int main(int argc, char const *argv[])
{
// KAMUS
    typedef char ElType;
    typedef struct { // Matrix untuk menyimpan puzzle
    ElType contents[100][100];
    int rowEff; /* banyaknya ukuran baris yg terdefinisi */
    int colEff; /* banyaknya ukuran kolom yg terdefinisi */
    } Matrix;
    /* rowEff >= 1 dan colEff >= 1 */
    /* Indeks matriks yang digunakan: [0..99][0..99] */
    /* Memori matriks yang dipakai selalu di "ujung kiri atas" */

    typedef struct { // list untuk menyimpan kata yang dicari
        ElType contents[50];
        int row[50]; // indeks baris dari karakter yang dicari
        int col[50]; // indeks kolom dari karakter yang dicari
        int idxEff; // indeks effektif yang digunakan >= 1
    } Kata;

    char cc , file_name[25];
    Matrix m;
    Kata word;
    FILE *fp;
```

```
    double totalTime; // Variable untuk menyimpan waktu eksekusi program
    struct timespec begin, end; // Waktu awal dan akhir eksekusi
    long int totalComparison = 0; // Total jumlah perbandingan huruf


// ALGORITMA
    // Membaca input text
    printf("Masukkan nama file input: ");
    scanf("%s",file_name);
    fp = fopen(file_name, "r");

    // Mengecek file ada atau tidak
    if (fp == NULL) {
        perror("");
        exit(EXIT_FAILURE);
    } else {
        boolean flagPrec = false, flagAfter = false; // Penanda baris kosong
        int i = 0, j = 0; // index baris dan kolom matriks
        // Membaca Matriks huruf puzzle
        cc = fgetc(fp);
        while(!flagAfter) {
            if (cc != '\n') {
                flagPrec = false;
                if (cc != ' ') {
                    m.contents[i][j] = cc;
                    j++;
                }
            } else {
                if (!flagPrec) {
                    m.colEff = j + 1;
                    flagPrec = true;
                    j = 0;
                    i++;
                } else {
                    flagAfter = true;
                }
            }
            cc = fgetc(fp);
        }

        m.rowEff = i;

        // Membaca daftar kata yang dicari di dalam puzzle dan menampilkan
        // di mana kata tersebut berada di dalam puzzle
        i = 0;

        clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &begin);
        boolean flag = false;
        while (!flag) {
            if (cc != '\n' && cc != EOF) {
```

```
            if (cc != ' ') {
                word.contents[i] = cc;
                i++;
            }
        } else {
            if (cc == EOF) {
                flag = true;
            }
            word.idxEff = i;
            i = 0;

            // Menghitung waktu yang dibutuhkan untuk eksekusi program
            // begin = clock(); // waktu mulai

            if (word.idxEff != 0) { // Mengecek apakah string kosong
                // Mencari kata dalam puzzle
                int row = 0, col = 0, idx;
                boolean found = false;
                while (!found && row != m.rowEff && col != m.colEff) {
                    // printf("%d ",row);
                    idx = 0;
                    totalComparison++;
                    if (m.contents[row][col] == word.contents[idx]) {
                        // Mengecek vertikal ke atas
                        if (row + 1 - word.idxEff >= 0) {
                            int k = row;
                            while (idx < word.idxEff && m.contents[k][col] ==
                            word.contents[idx]) {
                                totalComparison++;
                                word.row[idx] = k;
                                word.col[idx] = col;
                                k--;
                                idx++;
                            }
                            if (idx == word.idxEff) {
                                found = true;
                            } else {
                                totalComparison++;
                                idx = 0;
                            }
                        }
                        // Mengecek Vertikal ke bawah
                        if (!found && row + word.idxEff <= m.rowEff) {
                            int k = row;
                            while (idx < word.idxEff && m.contents[k][col] ==
                            word.contents[idx]) {
                                totalComparison++;
                                word.row[idx] = k;
                                word.col[idx] = col;
                                k++;
```

```
                idx++;
            }
            if (idx == word.idxEff) {
                found = true;
            } else {
                totalComparison++;
                idx = 0;
            }
        }
        // Mengecek horizontal ke kanan
        if (!found && col + word.idxEff <= m.colEff) {
            int k = col;
            while (idx < word.idxEff && m.contents[row][k] ==
            word.contents[idx]) {
                totalComparison++;
                word.row[idx] = row;
                word.col[idx] = k;
                k++;
                idx++;
            }
            if (idx == word.idxEff) {
                found = true;
            } else {
                totalComparison++;
                idx = 0;
            }
        }
        // Mengecek horizontal ke kiri
        if (!found && col + 1 - word.idxEff >= 0) {
            int k = col;
            while (idx < word.idxEff && m.contents[row][k] ==
            word.contents[idx]) {
                totalComparison++;
                word.row[idx] = row;
                word.col[idx] = k;
                k--;
                idx++;
            }
            if (idx == word.idxEff) {
                found = true;
            } else {
                totalComparison++;
                idx = 0;
            }
        }
        // Mengecek diagonal ke kanan atas
        if (!found && (col + word.idxEff <= m.colEff) && (row + 1 –
        word.idxEff >= 0)) {
            int p = row, q = col;
            while (idx < word.idxEff && m.contents[p][q] ==
```

```
                word.contents[idx]) {
                    totalComparison++;
                    word.row[idx] = p;
                    word.col[idx] = q;
                    p--;
                    q++;
                    idx++;
                }
                if (idx == word.idxEff) {
                    found = true;
                } else {
                    totalComparison++;
                    idx = 0;
                }
            }
            // Mengecek diagonal ke kanan bawah
            if (!found && (col + word.idxEff <= m.colEff) && (row +
            word.idxEff <= m.rowEff)) {
                int p = row, q = col;
                while (idx < word.idxEff && m.contents[p][q] ==
                word.contents[idx]) {
                    totalComparison++;
                    word.row[idx] = p;
                    word.col[idx] = q;
                    p++;
                    q++;
                    idx++;
                }
                if (idx == word.idxEff) {
                    found = true;
                } else {
                    totalComparison++;
                    idx = 0;
                }
            }
            // Mengecek diagonal ke kiri atas
            if (!found && (col + 1 - word.idxEff >= 0) && (row + 1 –
            word.idxEff >= 0)) {
                int p = row, q = col;
                while (idx < word.idxEff && m.contents[p][q] ==
                word.contents[idx]) {
                    totalComparison++;
                    word.row[idx] = p;
                    word.col[idx] = q;
                    p--;
                    q--;
                    idx++;
                }
                if (idx == word.idxEff) {
                    found = true;
```

```
                    } else {
                        totalComparison++;
                        idx = 0;
                    }
                }
                // Mengecek diagonal ke kiri bawah
                if (!found && (col + 1 - word.idxEff >= 0) && (row +
                word.idxEff <= m.rowEff)) {
                    int p = row, q = col;
                    while (idx < word.idxEff && m.contents[p][q] ==
                    word.contents[idx]) {
                        totalComparison++;
                        word.row[idx] = p;
                        word.col[idx] = q;
                        p++;
                        q--;
                        idx++;
                    }
                    if (idx == word.idxEff) {
                        found = true;
                    } else {
                        totalComparison++;
                        idx = 0;
                    }
                }
            }

            if (!found) {
                if (col == m.colEff-1) {
                    col = 0;
                    row++;
                } else {
                    col++;
                }
            }
        }

        // end = clock(); // waktu akhir

        // Output ke layar
        if (found) {
            int r;
            for (int p = 0; p < m.rowEff; p++) {
                for (int q = 0; q < m.colEff; q++) {
                    found = false; r = 0;
                    while (!found && r < word.idxEff) {
                        if (p == word.row[r] && q == word.col[r]) {
                            found = true;
                        } else {
                            r++;
```

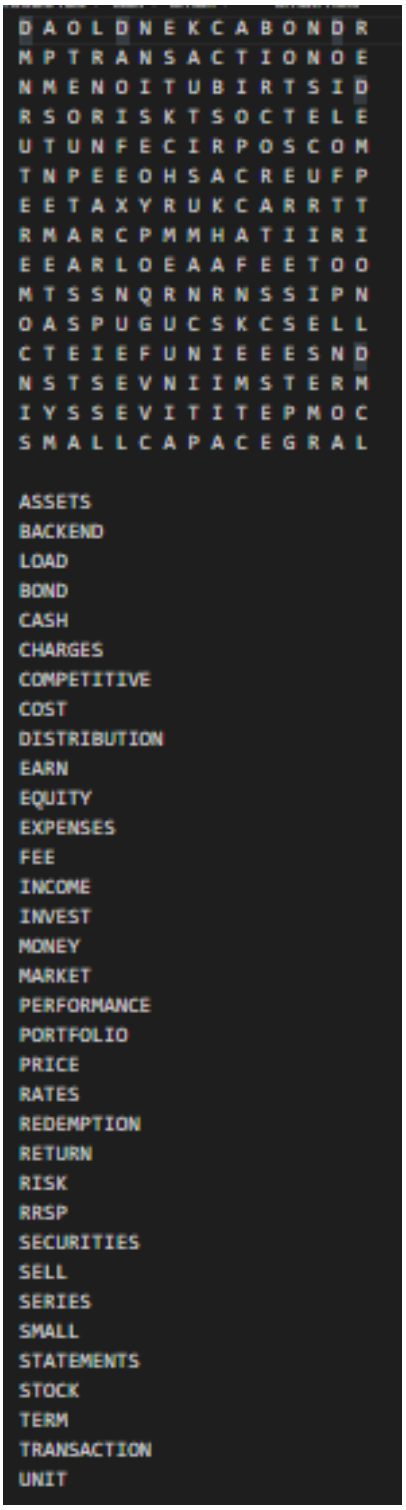
```
                            }
                        }
                        if (found) {
                            if (q != m.colEff - 1) {
                                printf("%c ", m.contents[p][q]);
                            } else {
                                printf("%c\n", m.contents[p][q]);
                            }
                            r++;
                        } else {
                            if (q != m.colEff - 1) {
                                printf("- ");
                            } else {
                                printf("\n");
                            }
                        }
                    }
                }
                printf("\n");
            }

            // totalTime += (end - begin) / CLOCKS_PER_SEC;
        }
    }
    cc = fgetc(fp);
  }
  clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &end);
  totalTime = (end.tv_sec - begin.tv_sec) + (end.tv_nsec - begin.tv_nsec) /
  1e9;
}

fclose(fp);

// Output total waktu eksekusi program
printf("Waktu eksekusi program: %.10lf seconds.\n", totalTime);
printf("Total perbandingan huruf: %d.\n", totalComparison);

return 0;
}
```

C. Screenshot Input dan Output Program

Karena keterbatasan ruang, output program dimasukkan ke dalam tabel.

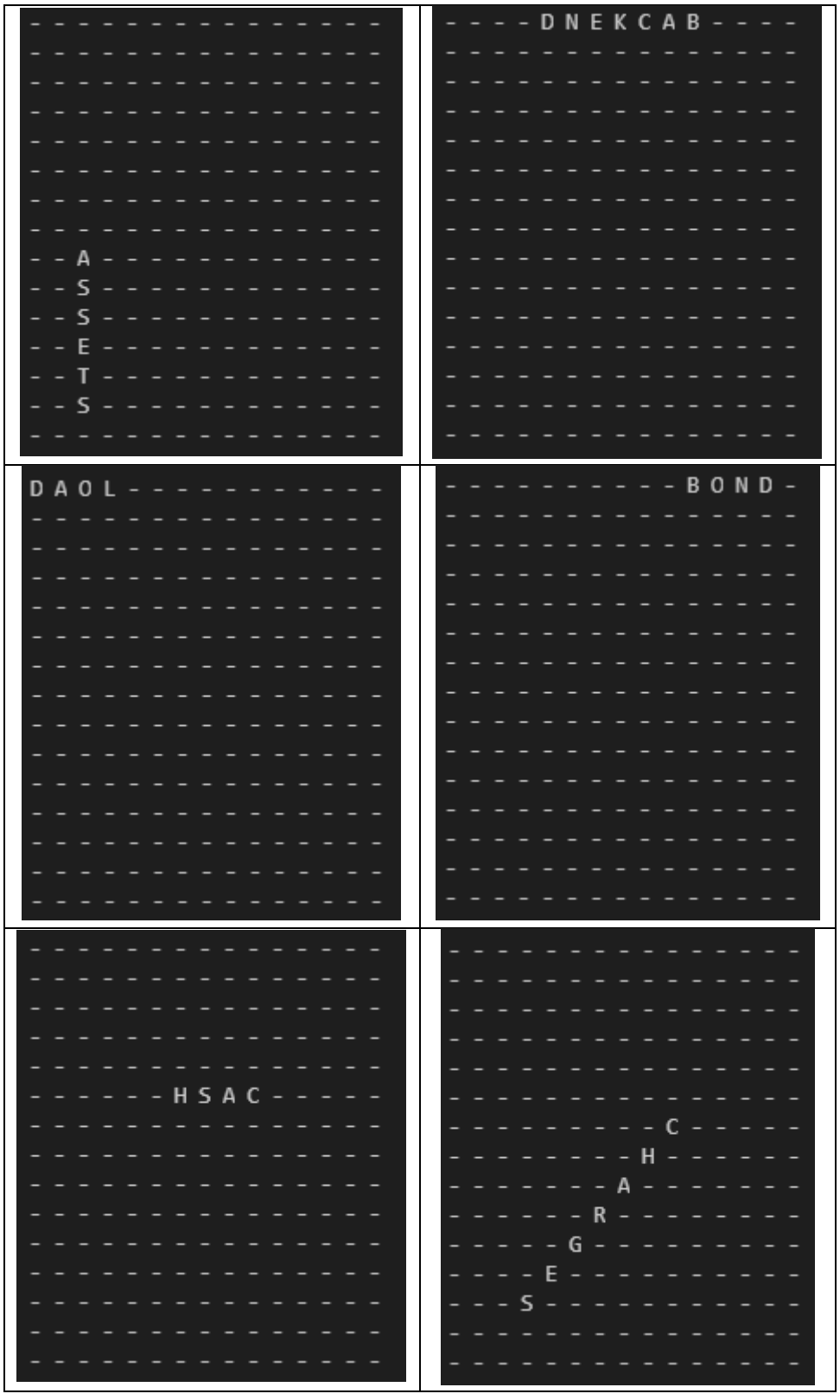
a. Puzzle berukuran small 1

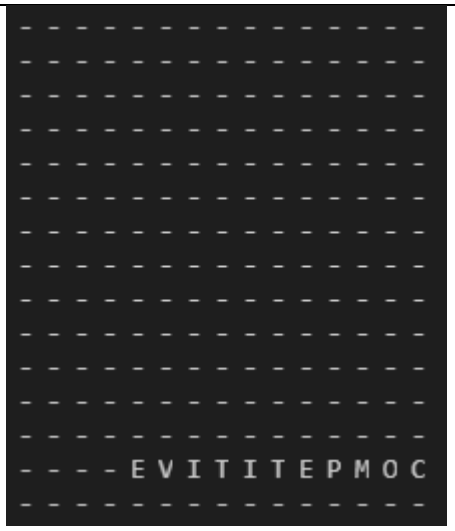
1. Input program

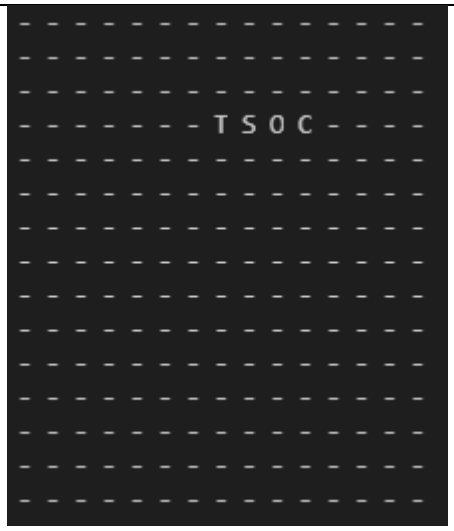


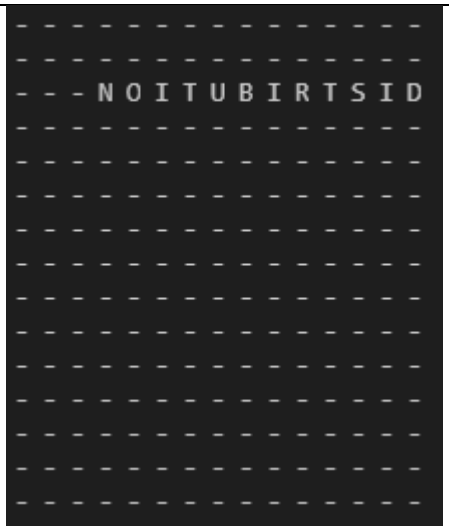
2. Output program

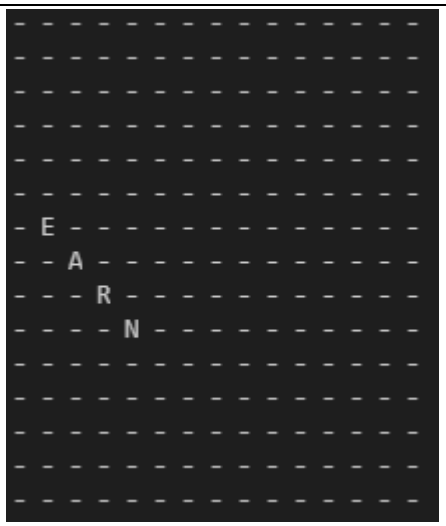
```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/small/small1.txt
```

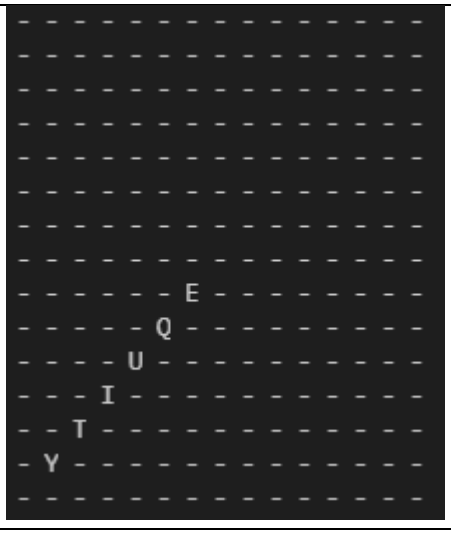


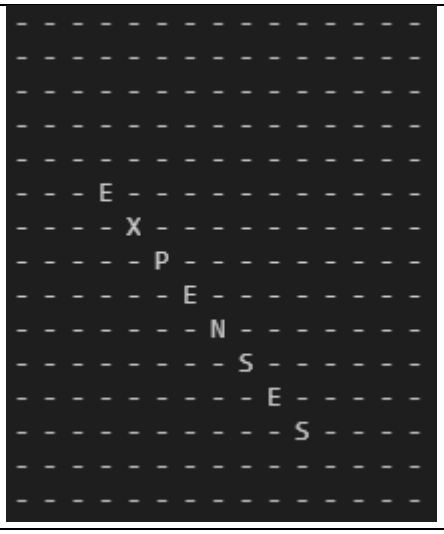

COMPETITIVE


COST

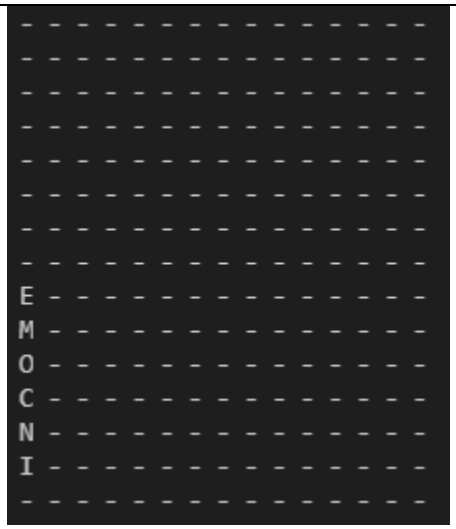

DISTRIBUTION


LEARN

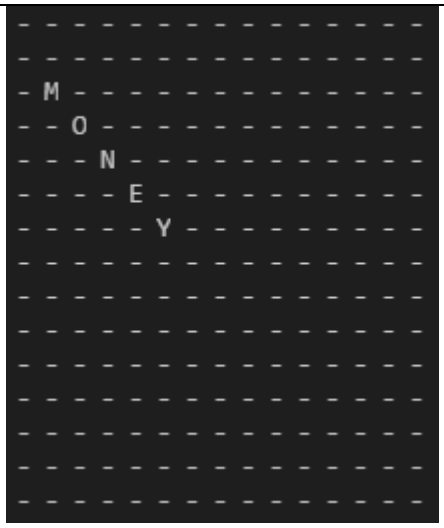

QUALITY


EXPENSES

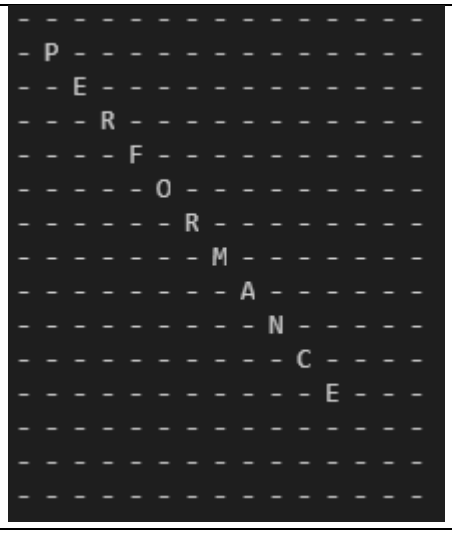

FEE

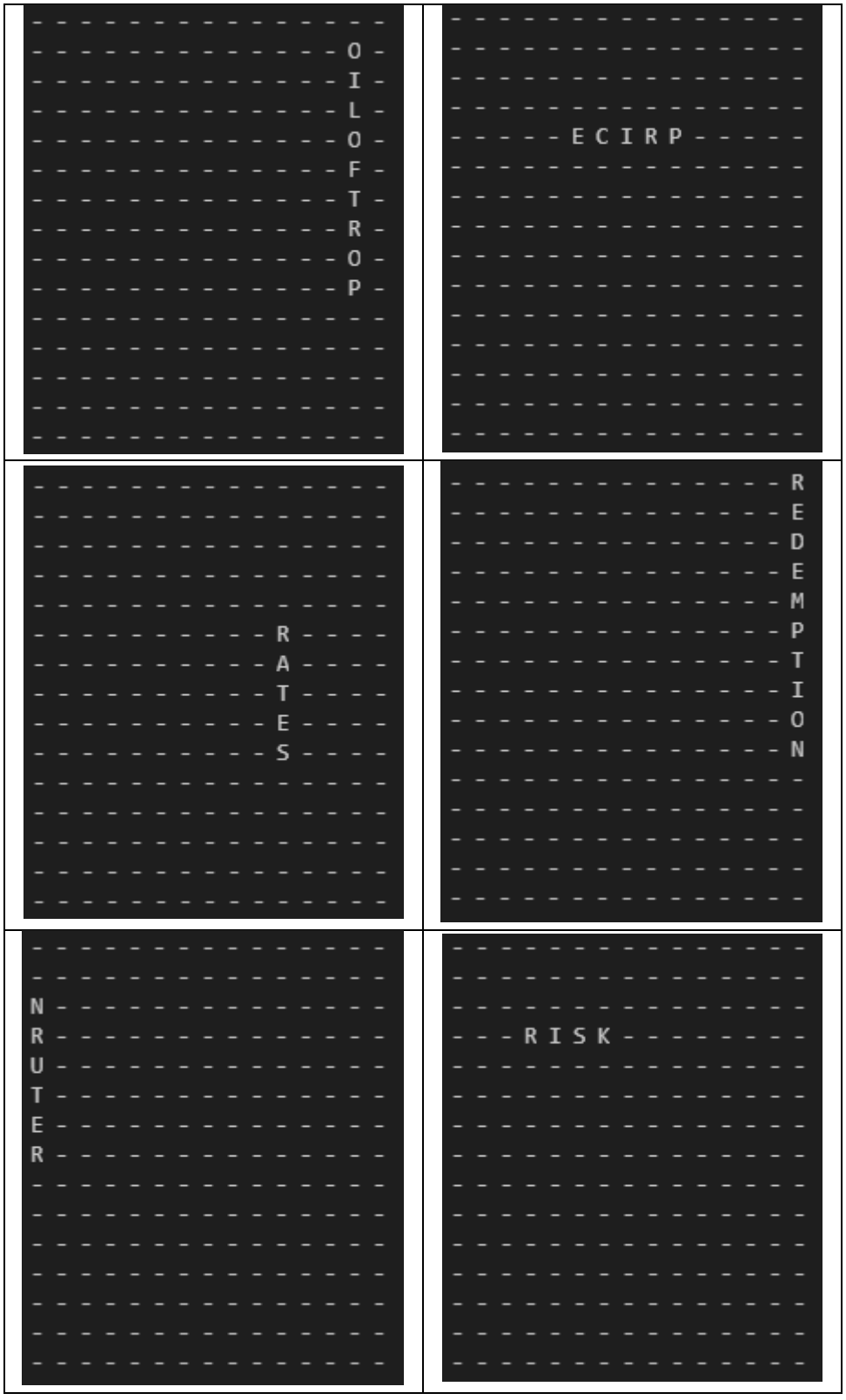

EMOCNI

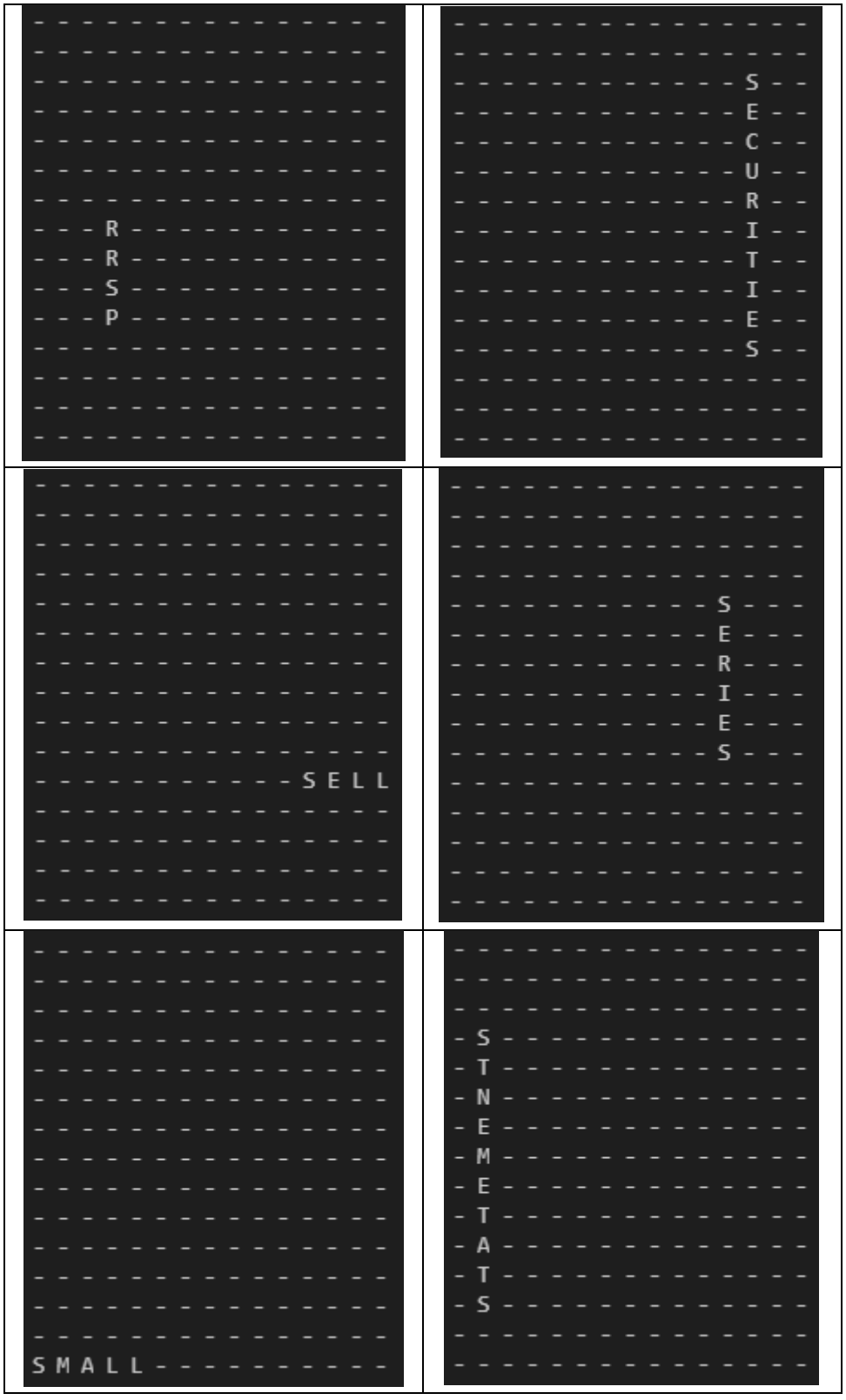

TSEVNI

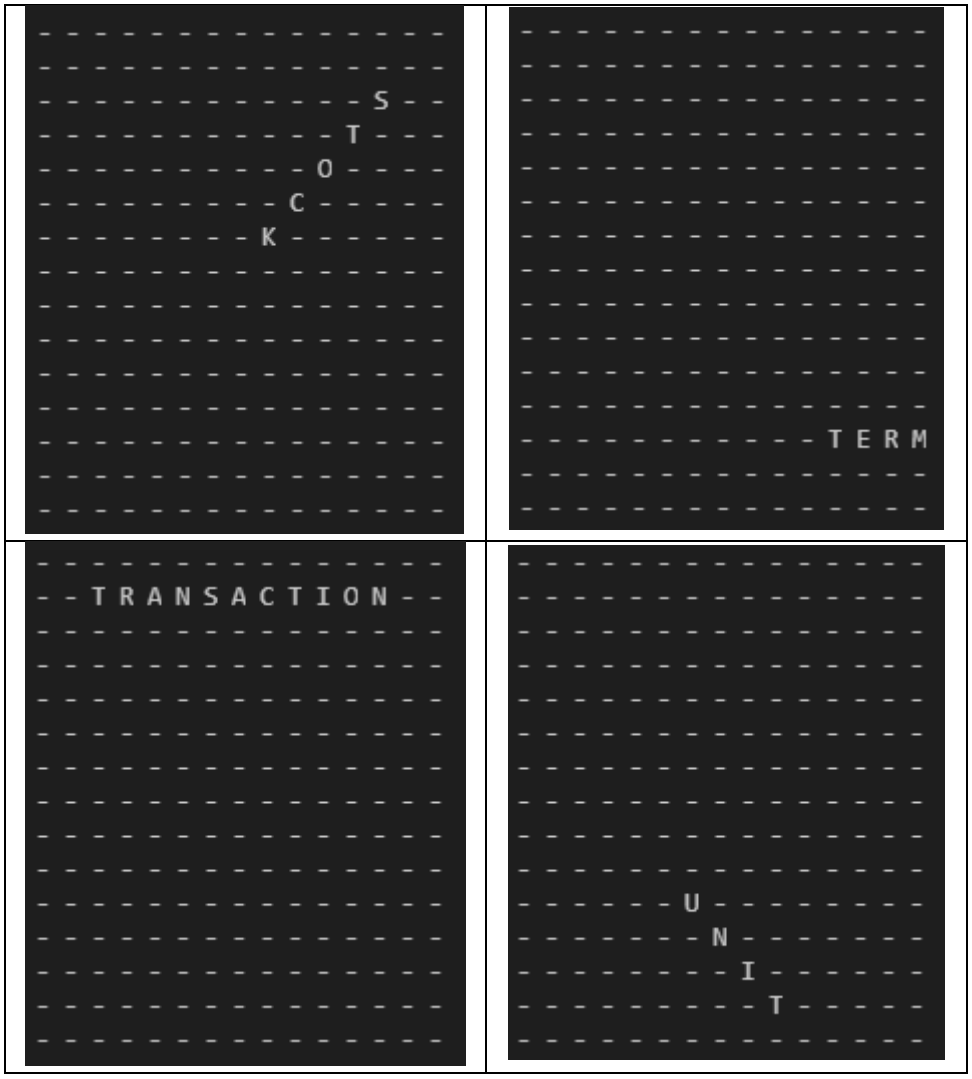

MONEY


MARKET


PERFORMANCE


OILOFTROP
PRICE
REDEMPTION
RATES
NRUTER
RISK


RRSP
SECURITIES
SELL
SERIES
SMALL
STNEMETATS




```
Waktu eksekusi program: 0.2187500000 seconds.
Total perbandingan huruf: 5909.
```

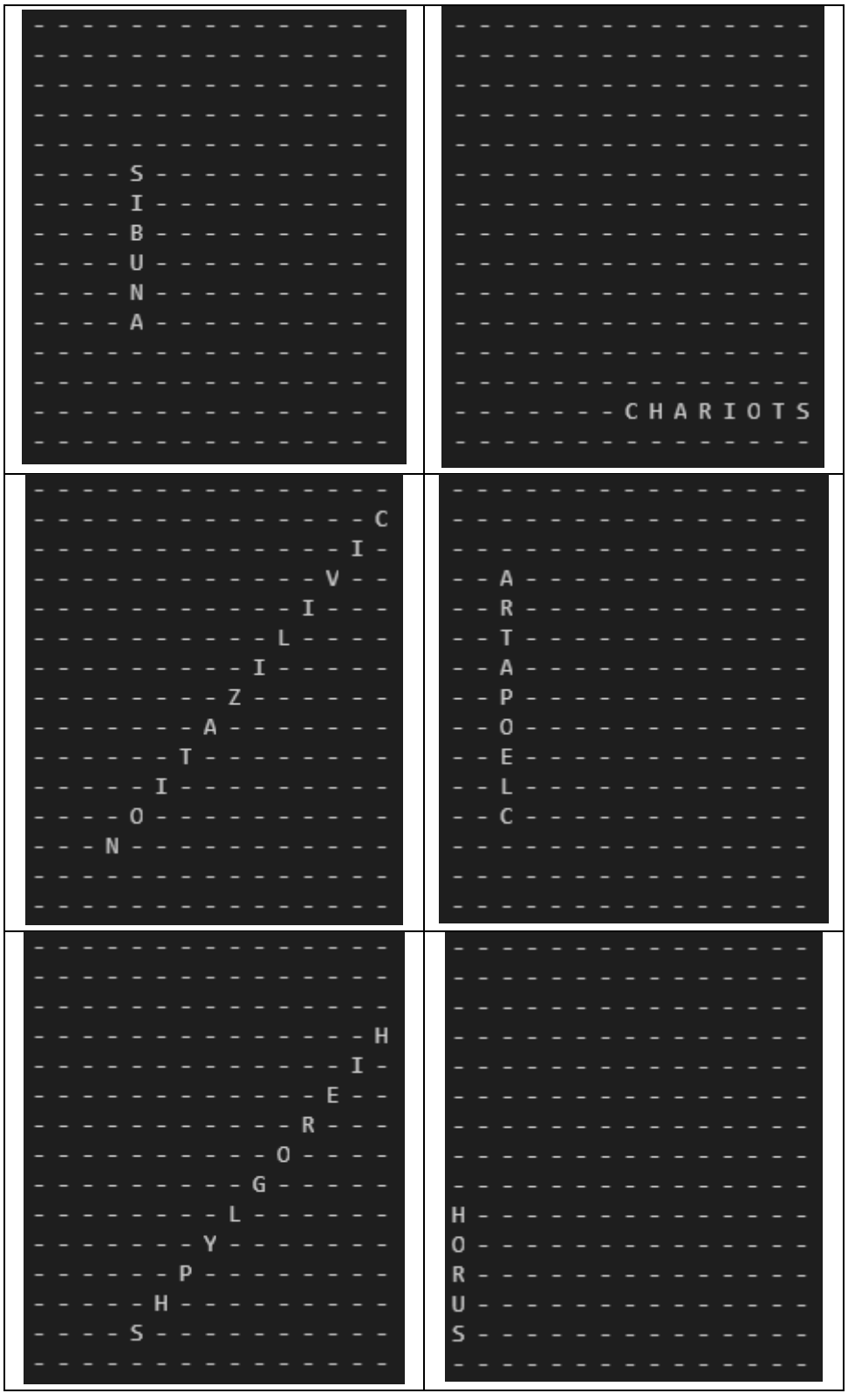
b. Puzzle berukuran small 2

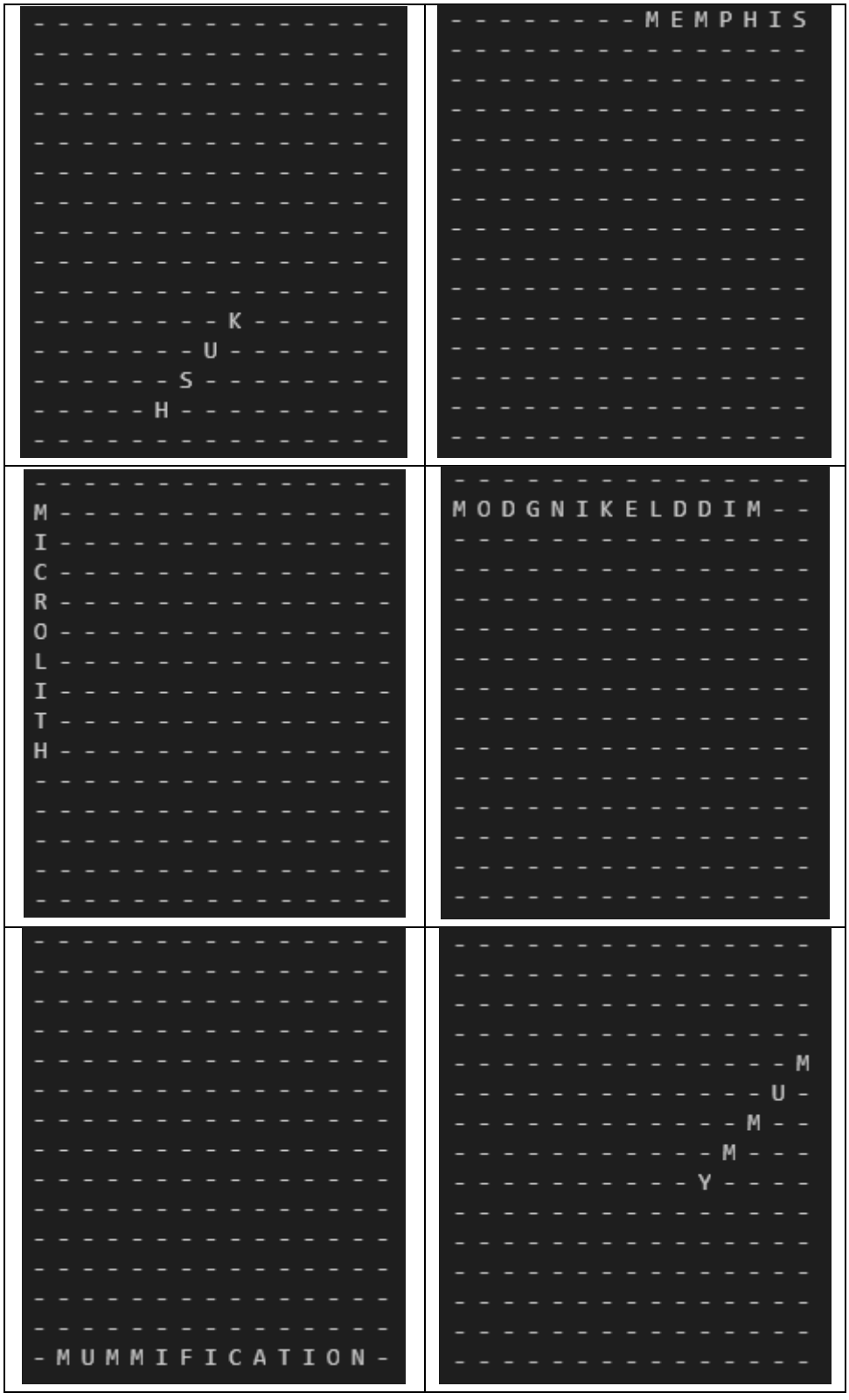
1. Input program

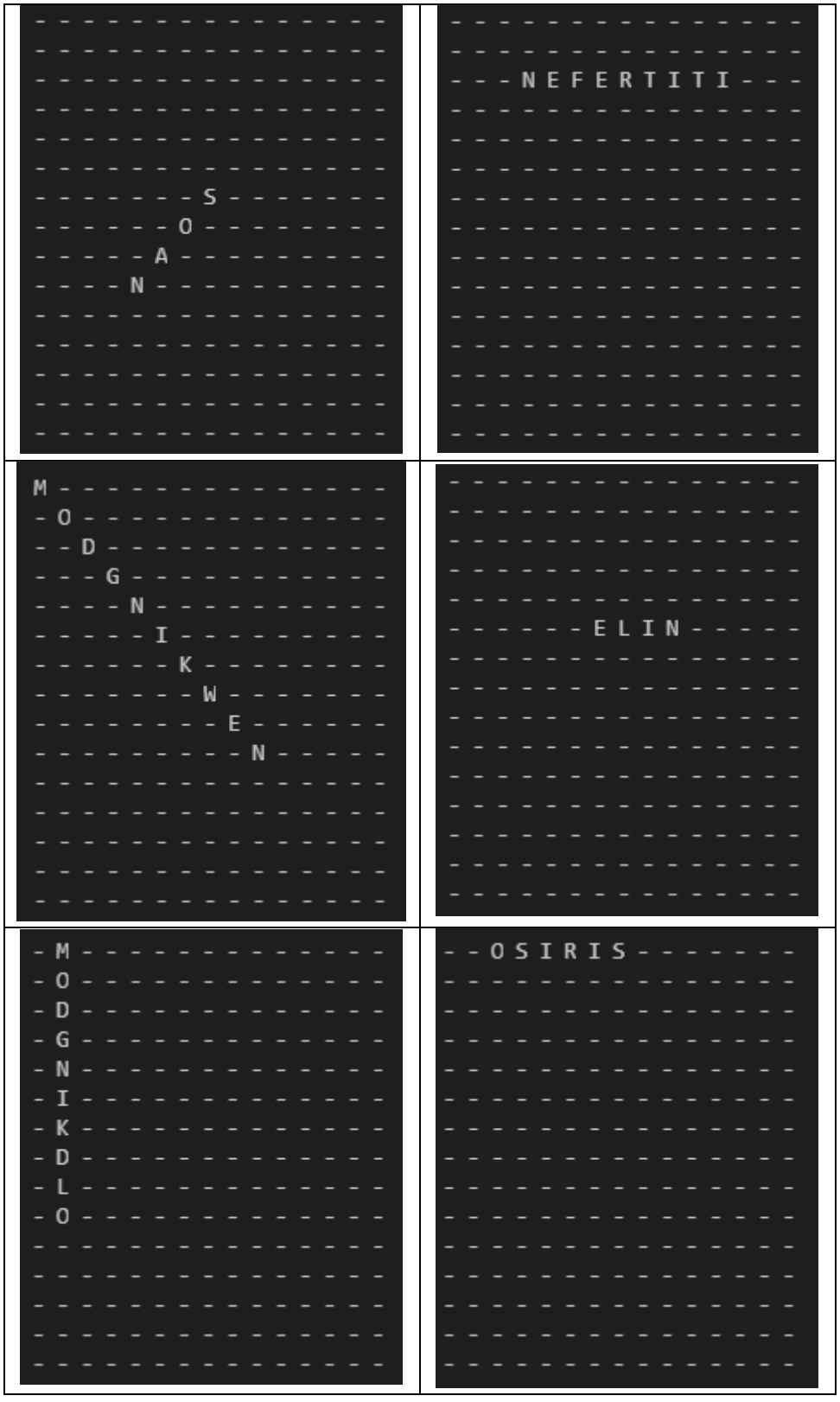
```
M M O S I R I S M E M P H I S
M O D G N I K E L D D I M S C
I D D N E F E R T I T I P I R
C G A G P A P Y R U S H V O H
R N R S N S H A B T I I M I M
O I T I S I E L I N L A E U P
L K A S I A K S X I N R M Y T
I D P T B N O W Z P O M R E P
T L O R U A S A E G Y A M H B
H O E U N L T R L N M P A S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

ANUBIS
CHARIOTS
CIVILIZATION
CLEOPATRA
HIEROGLYPHS
HORUS
KUSH
MEMPHIS
MICROLITH
MIDDLE KINGDOM
MUMMIFICATION
MUMMY
NAOS
NEFERTITI
NEW KINGDOM
NILE
OLD KINGDOM
OSIRIS
PAPYRUS
PHARAOH
PYRAMIDS
ROMAN PERIOD
SCARAB
SHABTI
SISTRUM
SLAVERY
SOBEK
SPHINX
TEMPLES
TOMBS
```

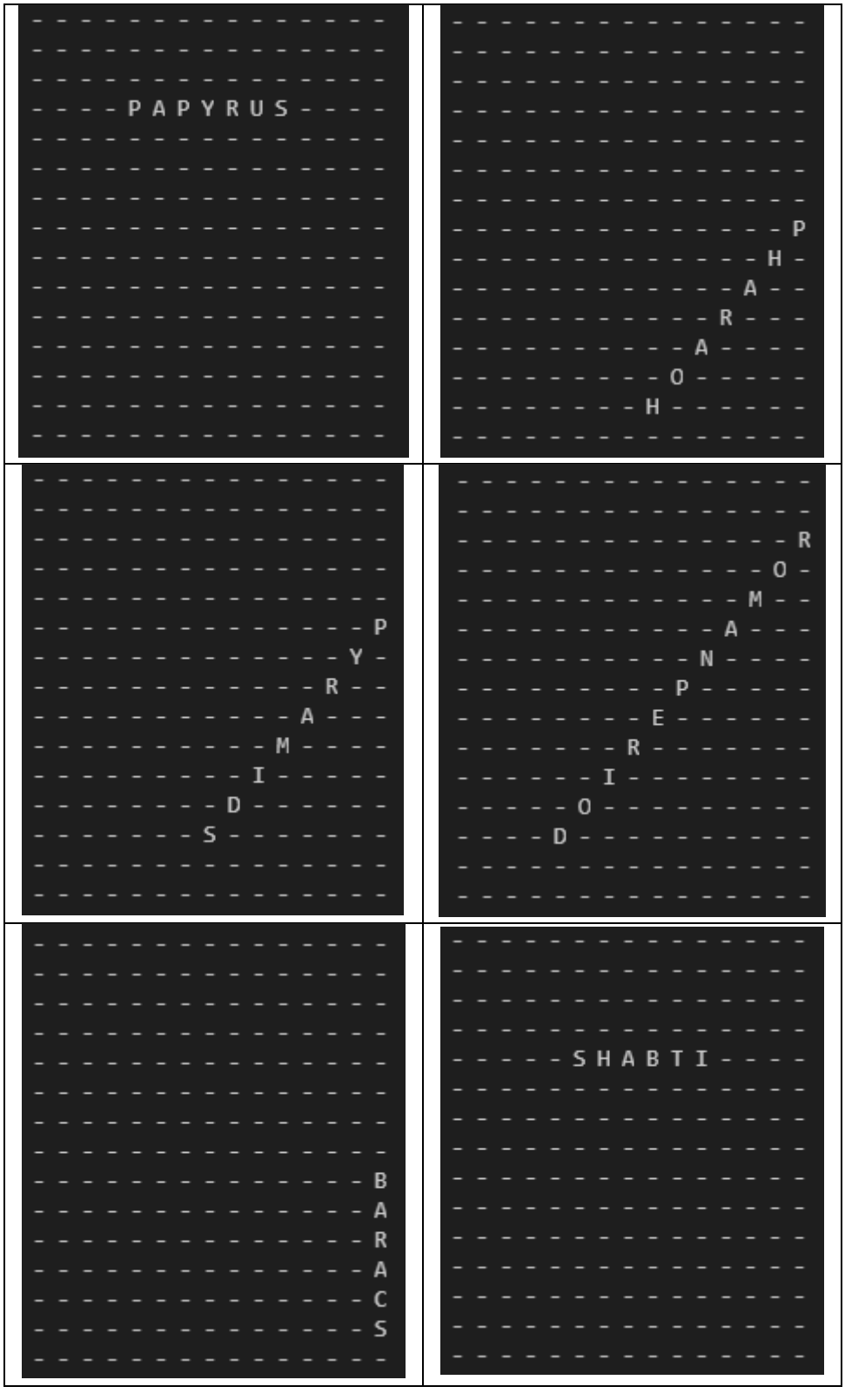
2. Output program

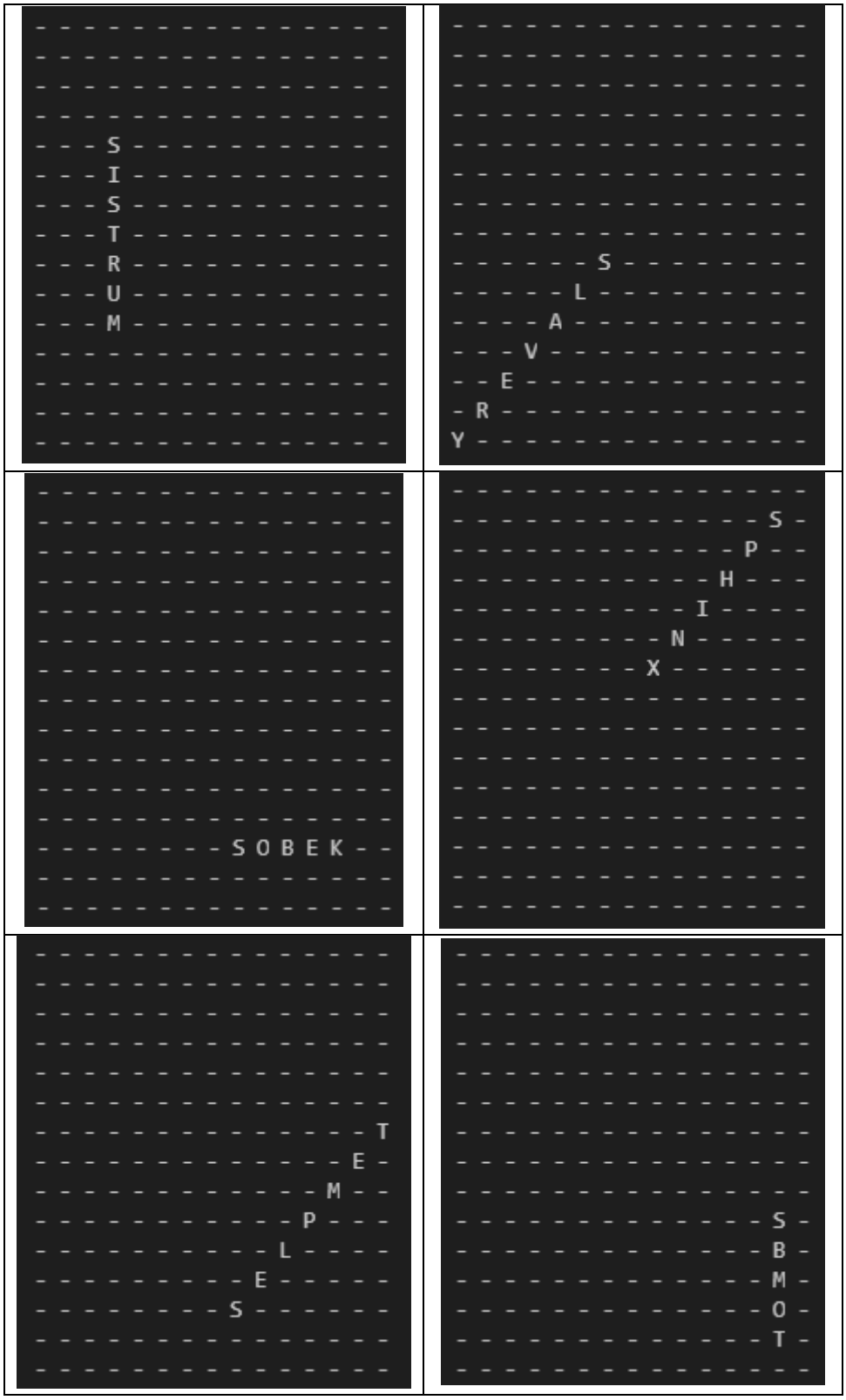
```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/small/small2.txt
```




K
U
S
H
MEMPHIS
MICROLITH
MODGNIKELDDIM
M
U
M
M
Y
MUMMIFICATION


SOAN
NEFERTITI
MODGNIKWEN
ELIN
MODGNIKDLO
OSIRIS

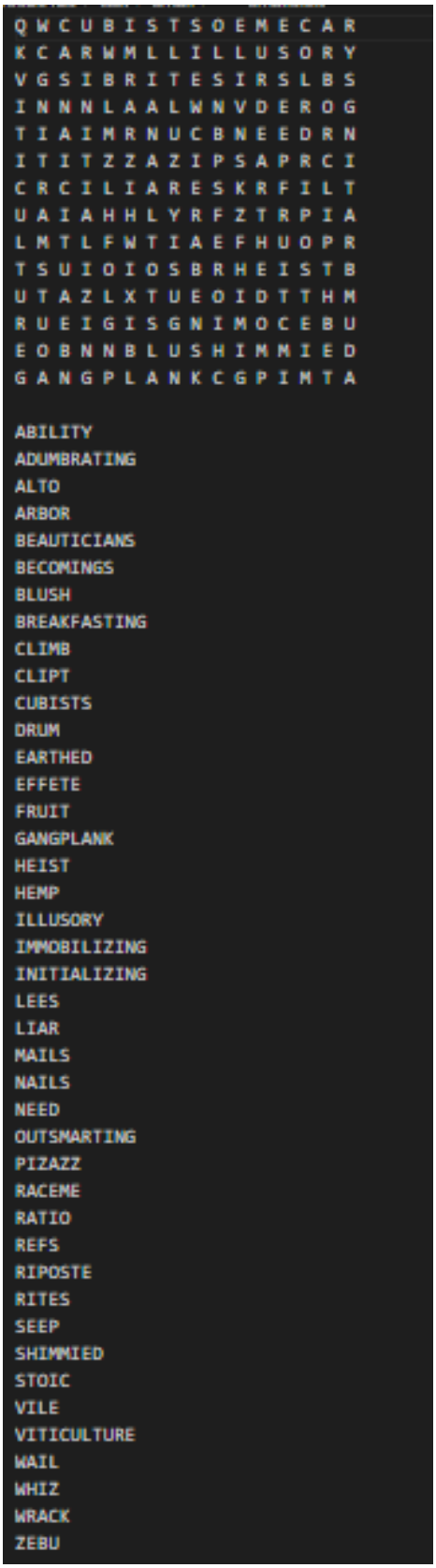

PAPYRUS
P
H
A
R
A
O
H
P
Y
R
A
M
I
D
S
R
O
M
A
N
P
E
R
I
O
D
SHABTI
BARACS




```
Waktu eksekusi program: 0.1093750000 seconds.
Total perbandingan huruf: 5029.
```

c. Puzzle berukuran small 3

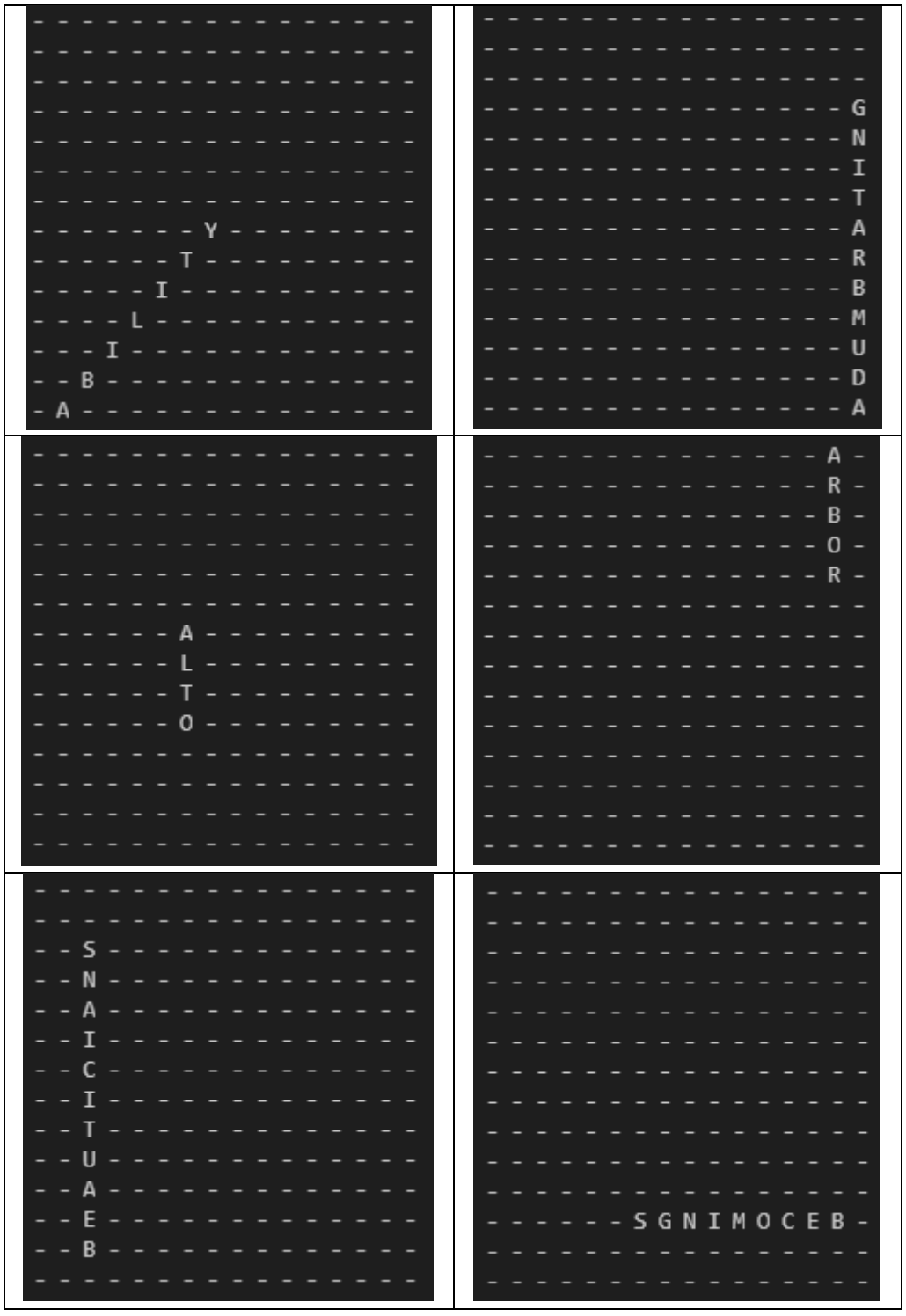
1. Input program

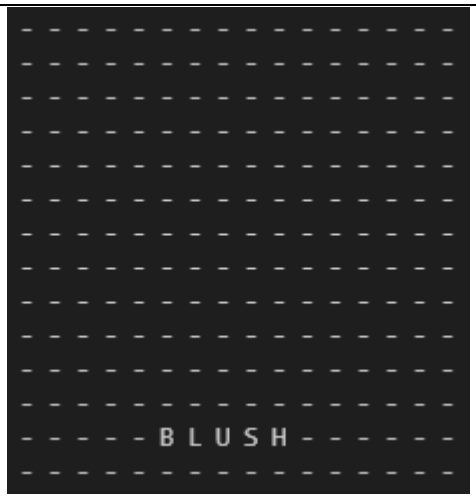


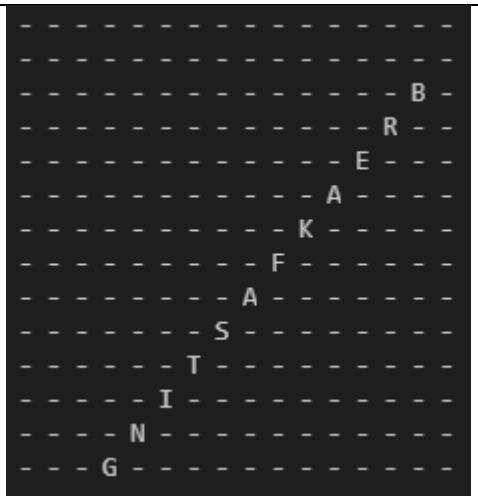
```
Q W C U B I S T S O E M E C A R
K C A R W M L L I L L U S O R Y
V G S I B R I T E S I R S L B S
I N N N L A A L W N V D E R O G
T I A I M R N U C B N E E D R N
I T I T Z Z A Z I P S A P R C I
C R C I L I A R E S K R F I L T
U A I A H H L Y R F Z T R P I A
L M T L F W T I A E F H U O P R
T S U I O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

ABILITY
ADUMBRATING
ALTO
ARBOR
BEAUTICIANS
BECOMINGS
BLUSH
BREAKFASTING
CLIMB
CLIPT
CUBISTS
DRUM
EARTHED
EFFETE
FRUIT
GANGPLANK
HEIST
HEMP
ILLUSORY
IMMOBILIZING
INITIALIZING
LEES
LIAR
MAILS
NAILS
NEED
OUTSMARTING
PIZAZZ
RACEME
RATIO
REFS
RIPOSTE
RITES
SEEP
SHIMMIED
STOIC
VILE
VITICULTURE
WAIL
WHIZ
WRACK
ZEBU
```

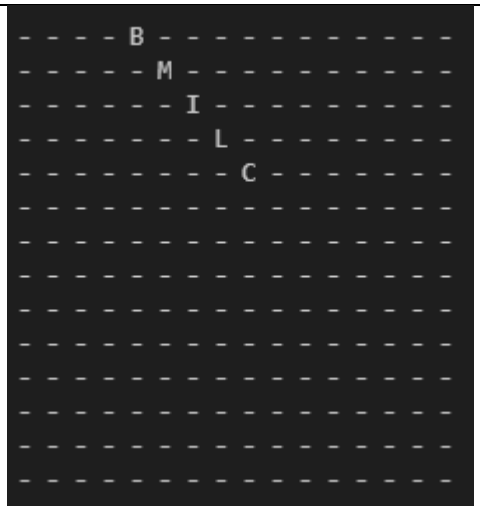
2. Output program

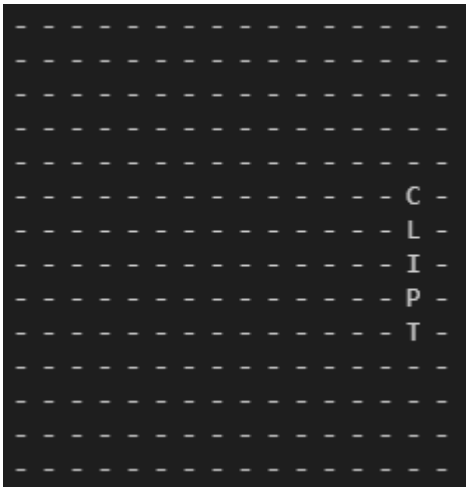
```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/small/small3.txt
```

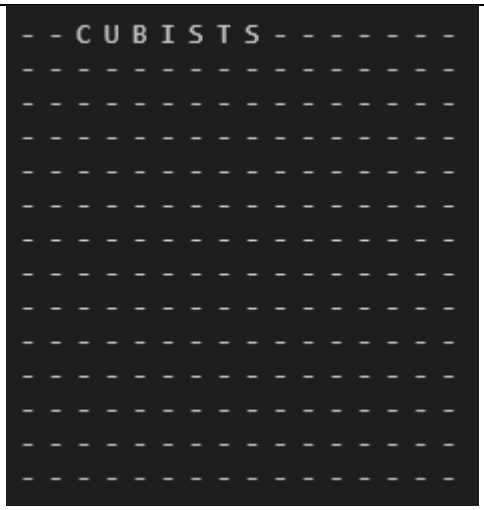


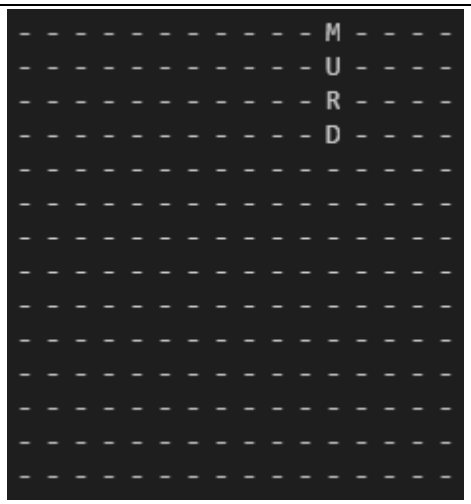

BLUSH

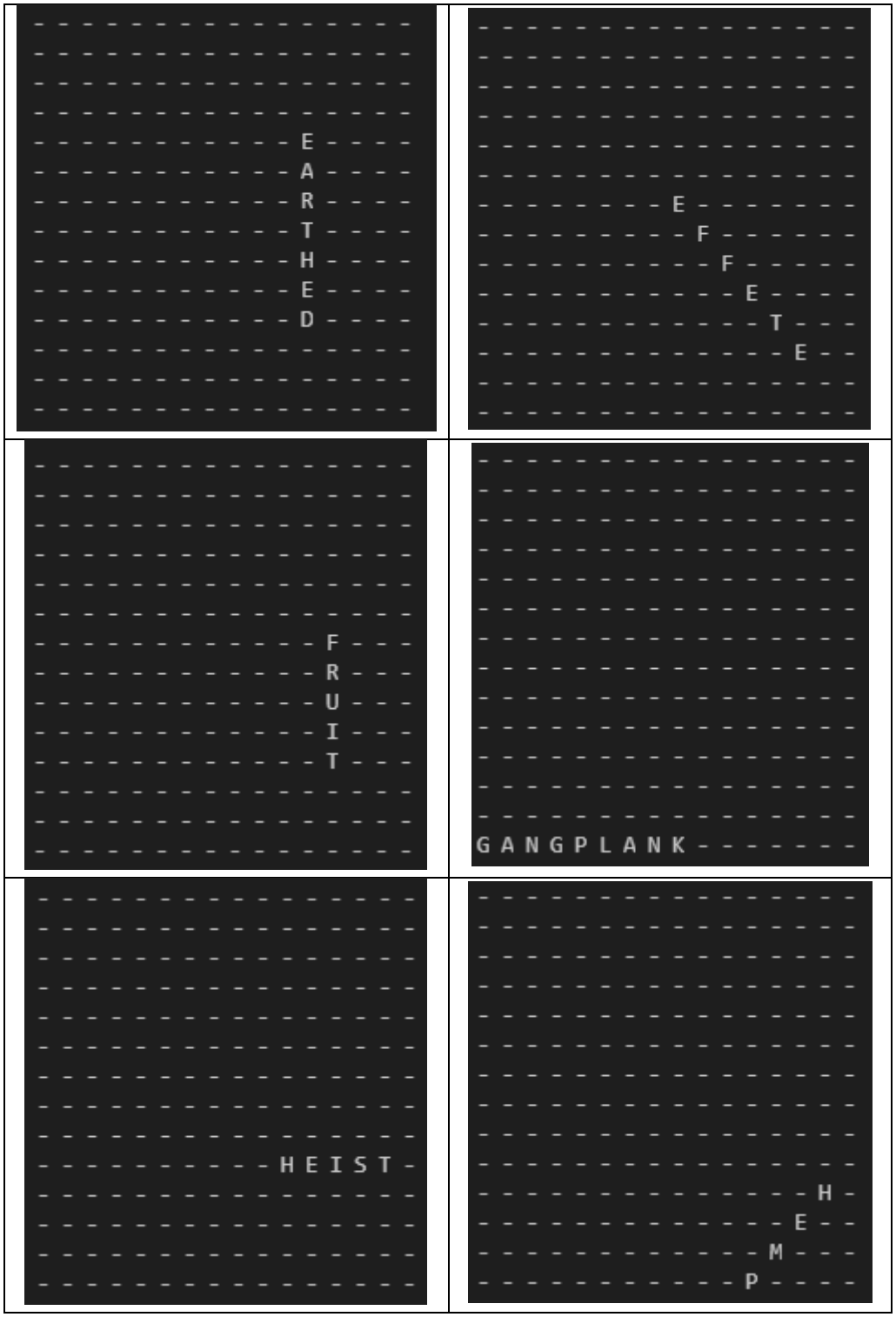

B
R
E
A
K
F
A
S
T
I
N
G


B
M
I
L
C

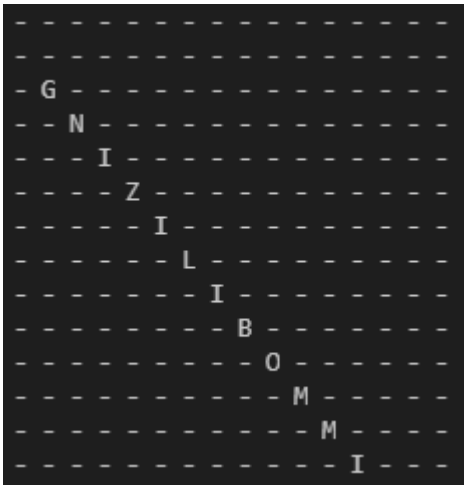

C
L
I
P
T

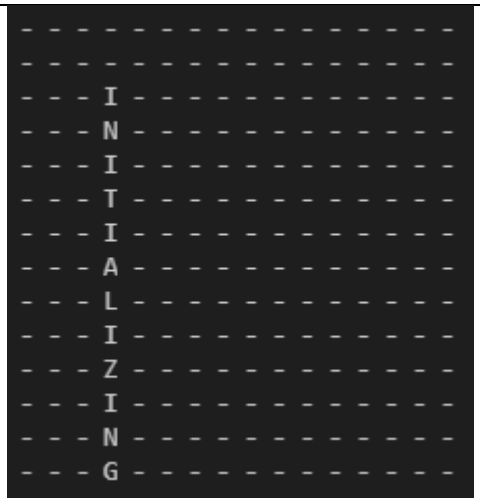

CUBISTS

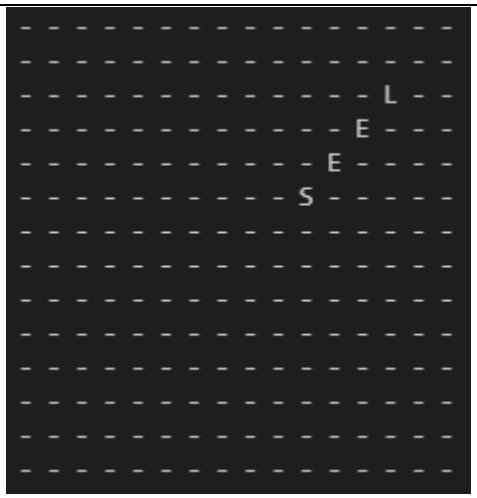

M
U
R
D


EARTHED
EFFETE
FRUIT
GANGPLANK
HEIST
HEMP

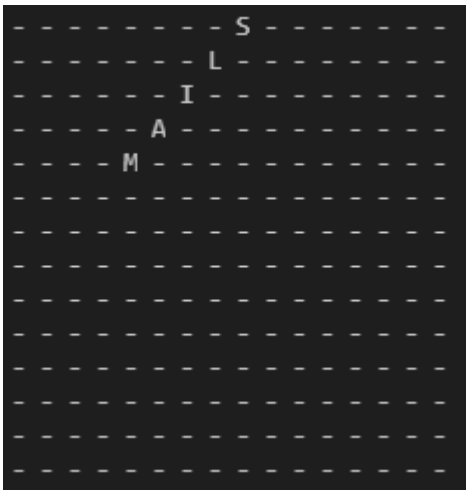

ILLUSORY

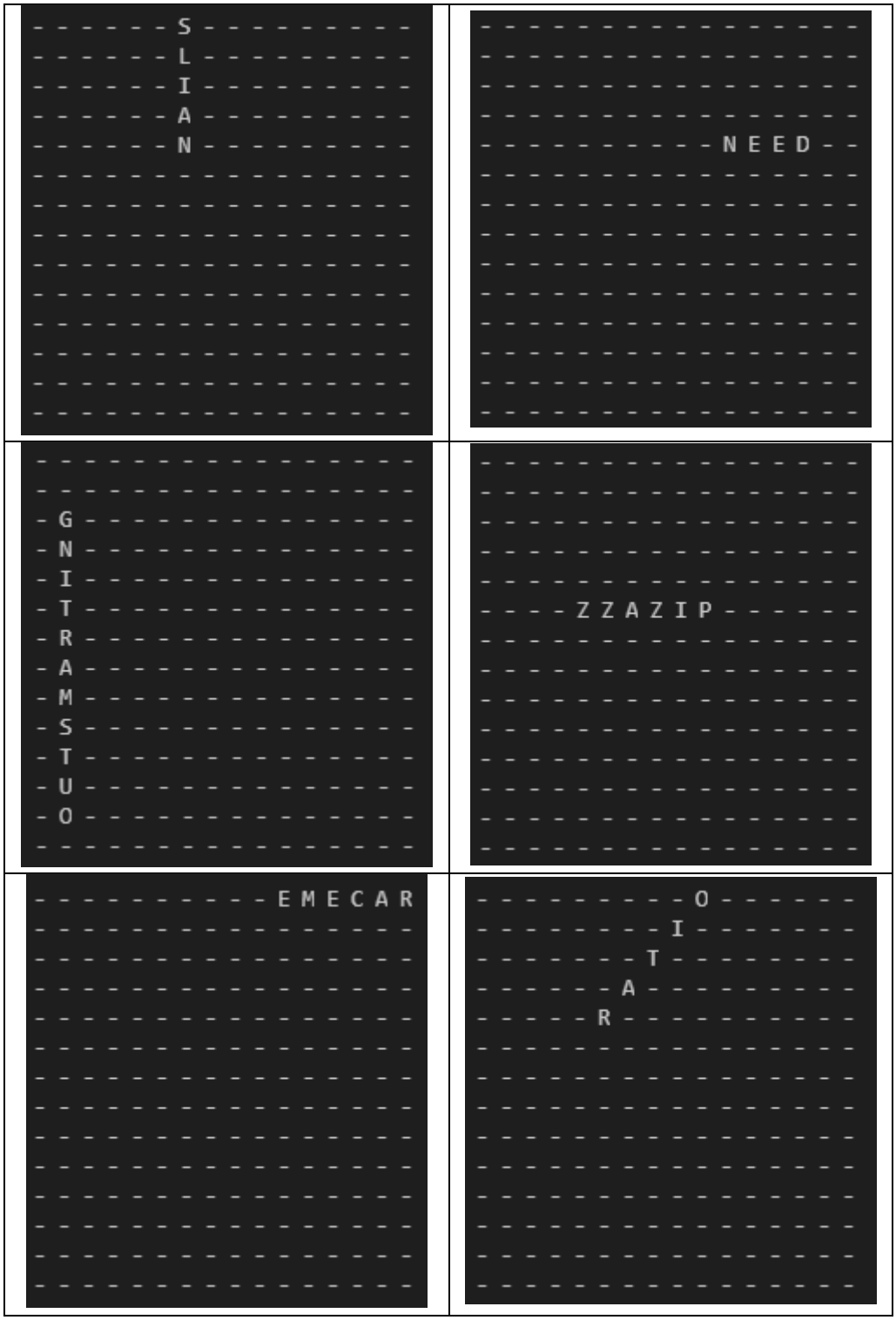

G
N
I
Z
I
L
I
B
O
M
M
I

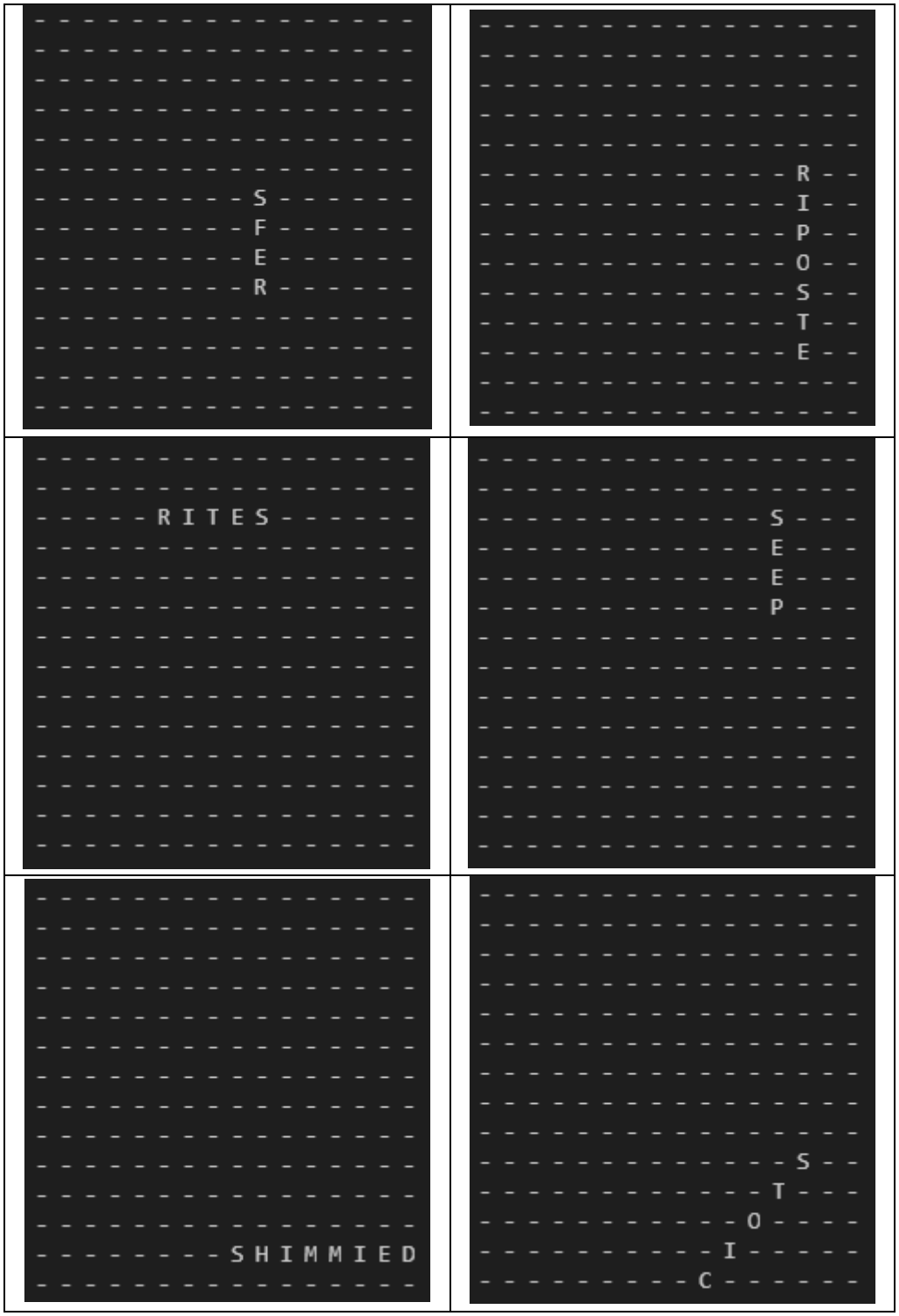

I
N
I
T
I
A
L
I
Z
I
N
G

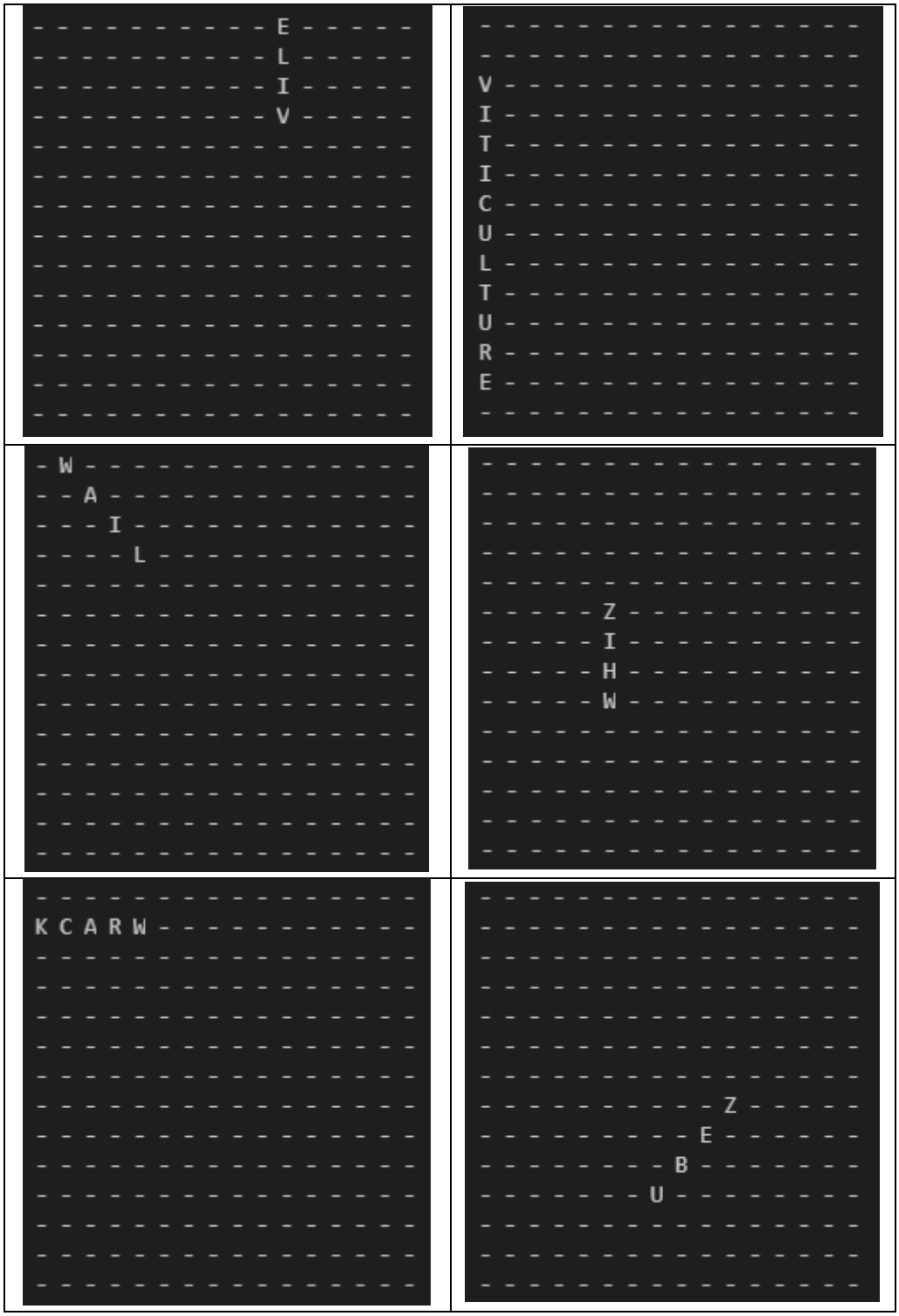

L
E
E
S


LIAR


S
L
I
A
M


SLIAN
NEED
GNITRAMSTUO
ZZAZIP
EMECAR
OITAR

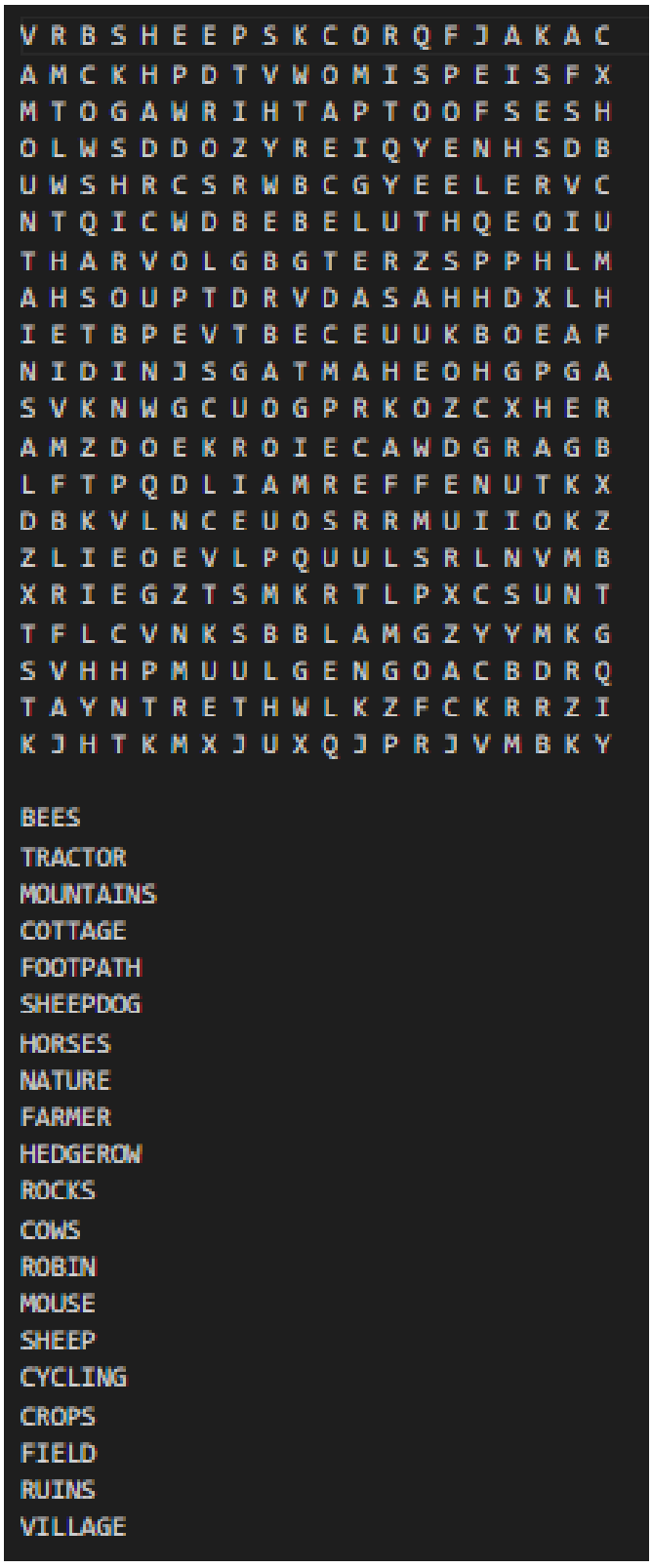

SFER
RIPOSTE
RITES
SEEP
SHIMMIED
STOIC




```
Waktu eksekusi program: 0.1562500000 seconds.
Total perbandingan huruf: 6707.
```

d. Puzzle berukuruan medium 1

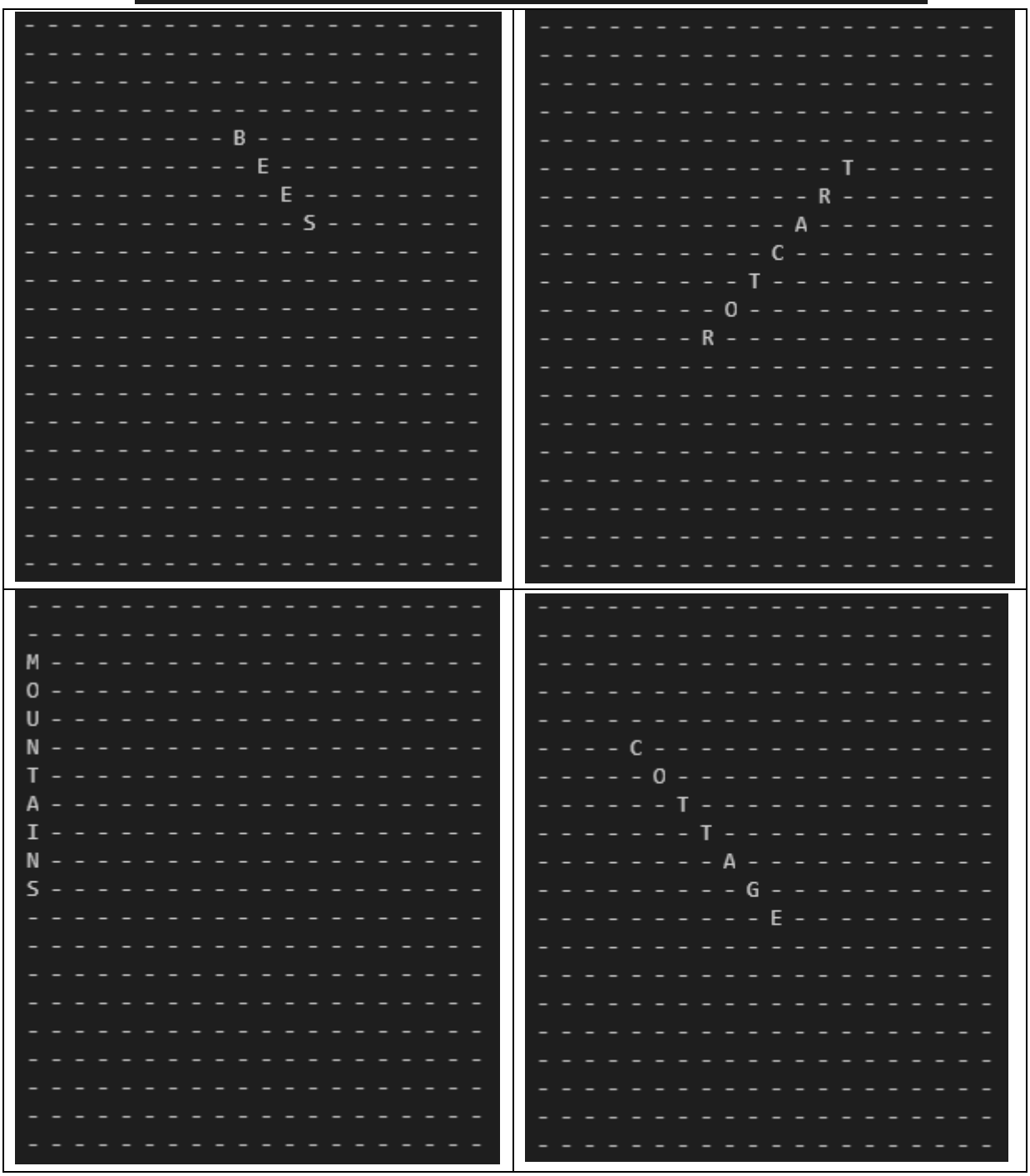
1. Input program

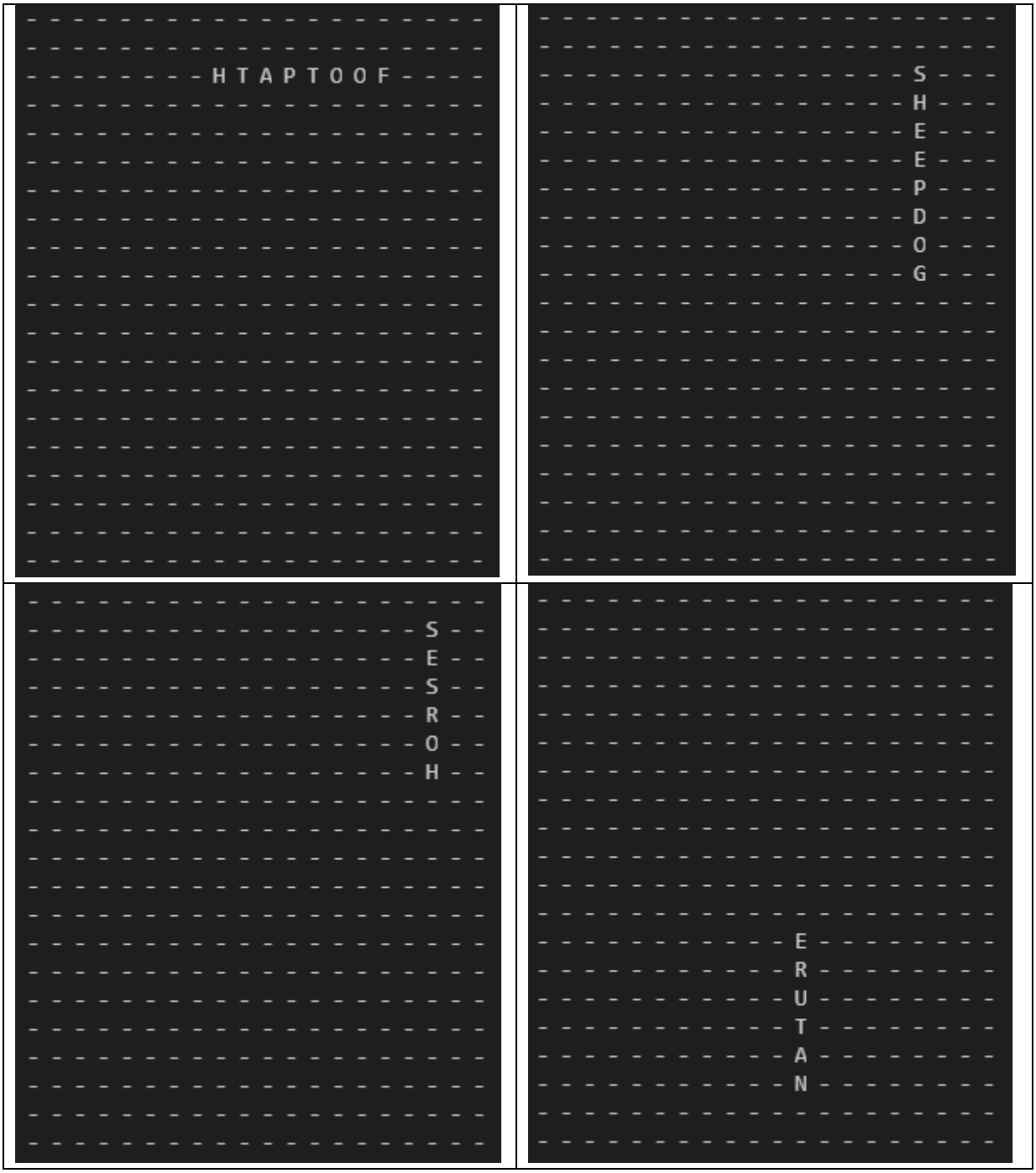


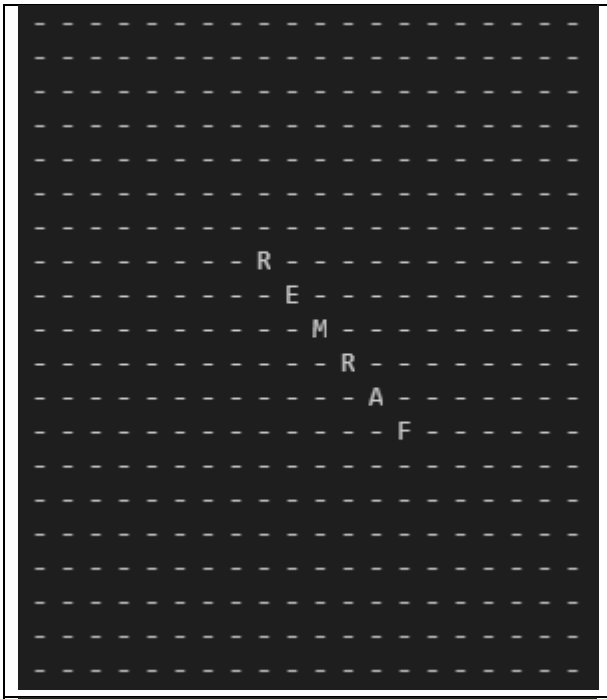
```
V R B S H E E P S K C O R Q F J A K A C
A M C K H P D T V W O M I S P E I S F X
M T O G A W R I H T A P T O O F S E S H
O L W S D D O Z Y R E I Q Y E N H S D B
U W S H R C S R W B C G Y E E L E R V C
N T Q I C W D B E B E L U T H Q E O I U
T H A R V O L G B G T E R Z S P P H L M
A H S O U P T D R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

BEES
TRACTOR
MOUNTAINS
COTTAGE
FOOTPATH
SHEEPDOG
HORSES
NATURE
FARMER
HEDGEROW
ROCKS
COWS
ROBIN
MOUSE
SHEEP
CYCLING
CROPS
FIELD
RUINS
VILLAGE
```

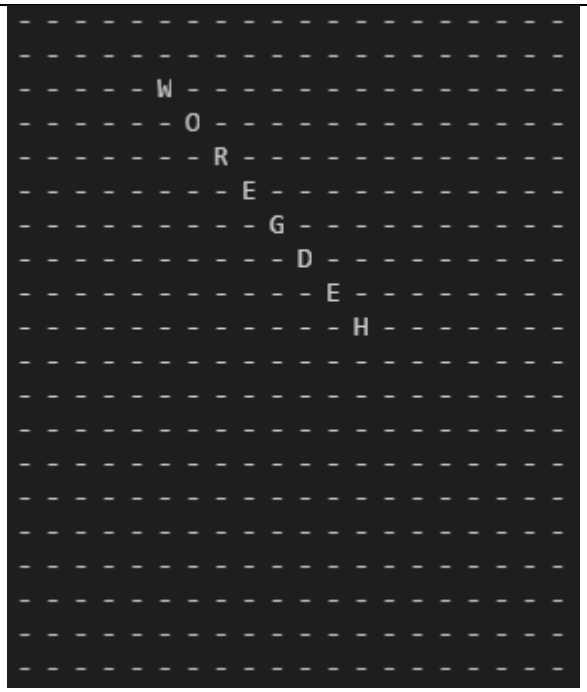
2. Output program

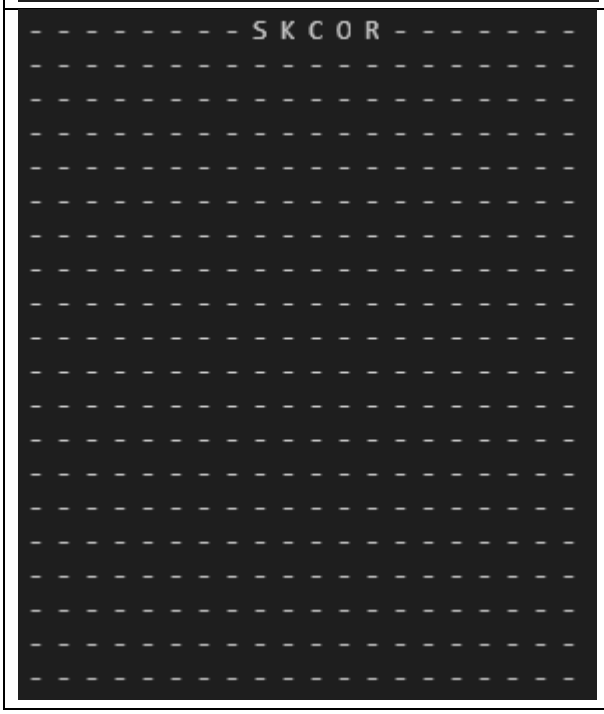
```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/medium/medium1.txt
```



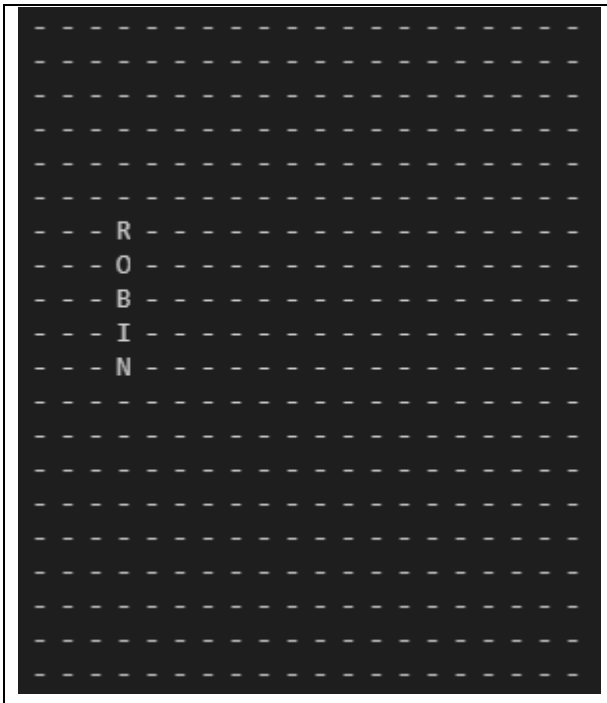

HTAPTOOF
SHEEPDOG
SEESROH
ERUTAN

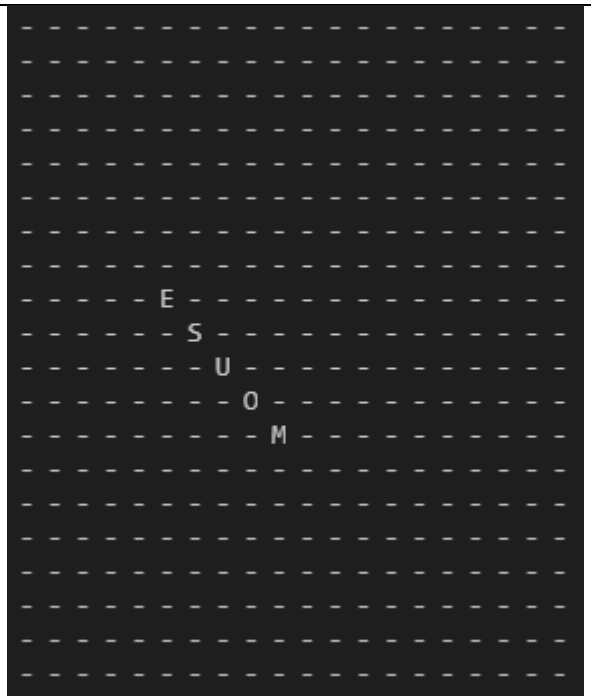

R
E
M
R
A
F

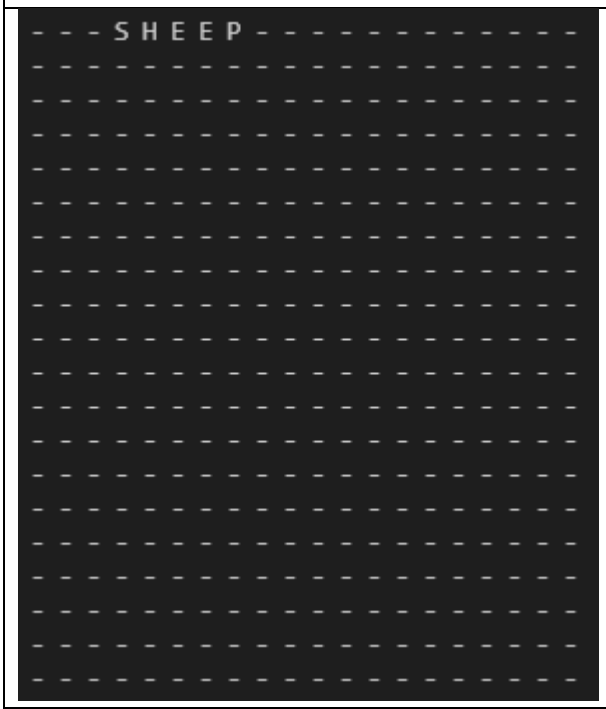

W
O
R
E
G
D
E
H

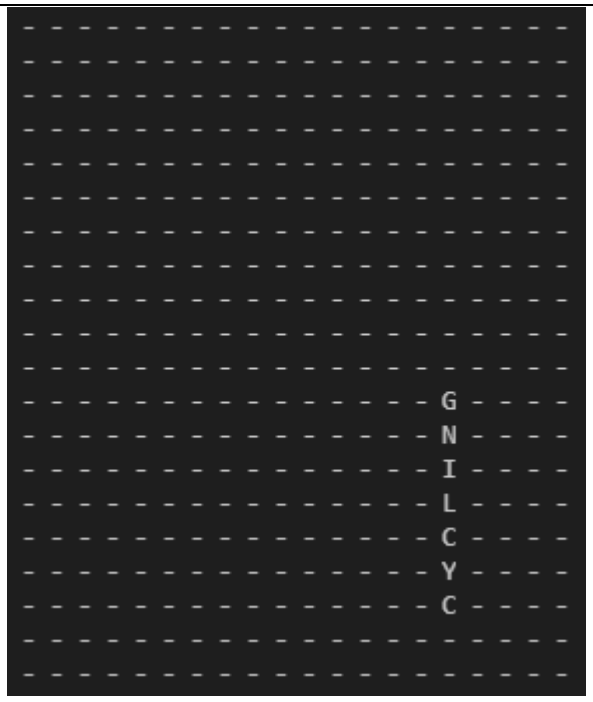

SKCOR

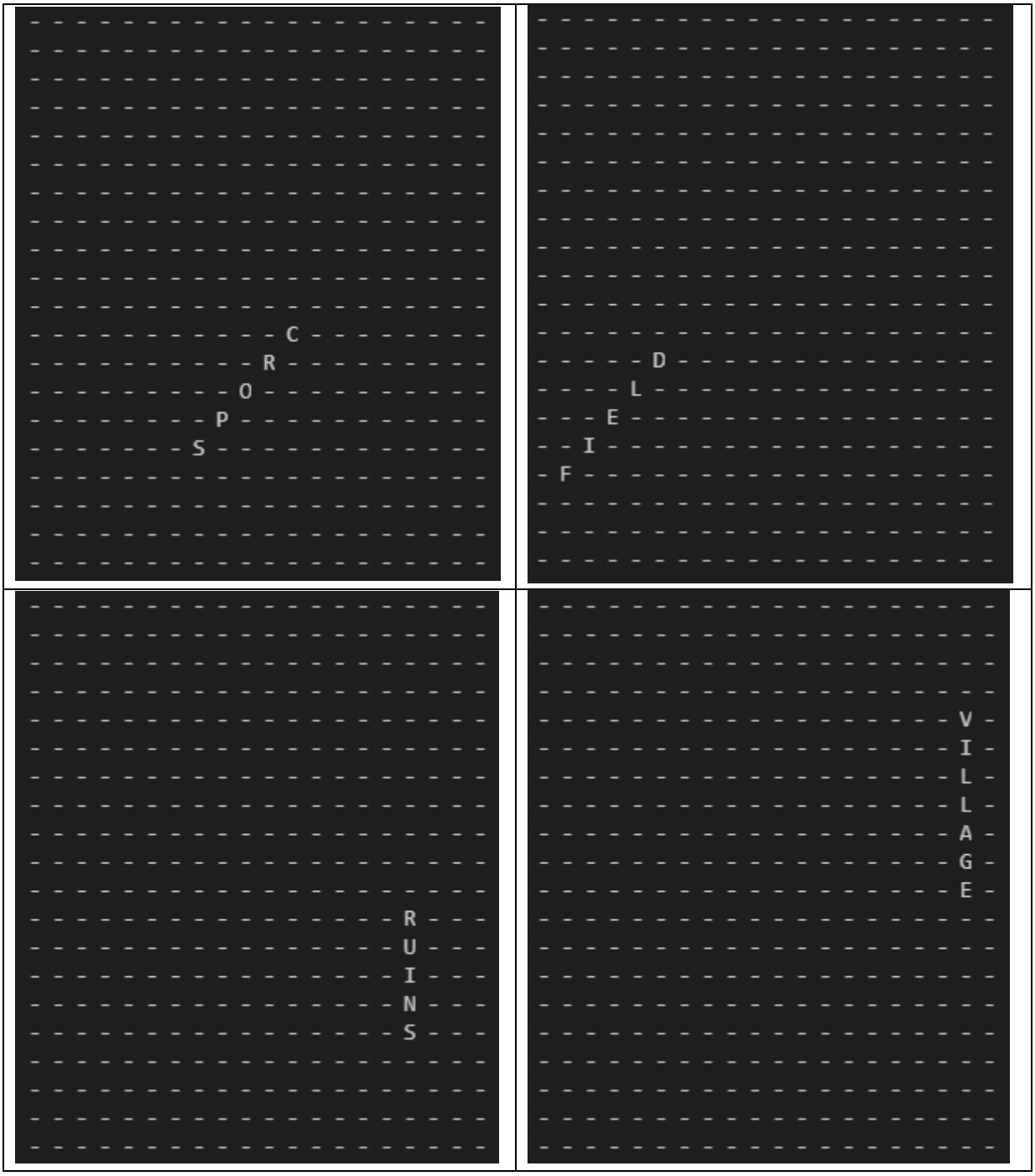

C
O
W
S


ROBIN


ESUOM


SHEEP

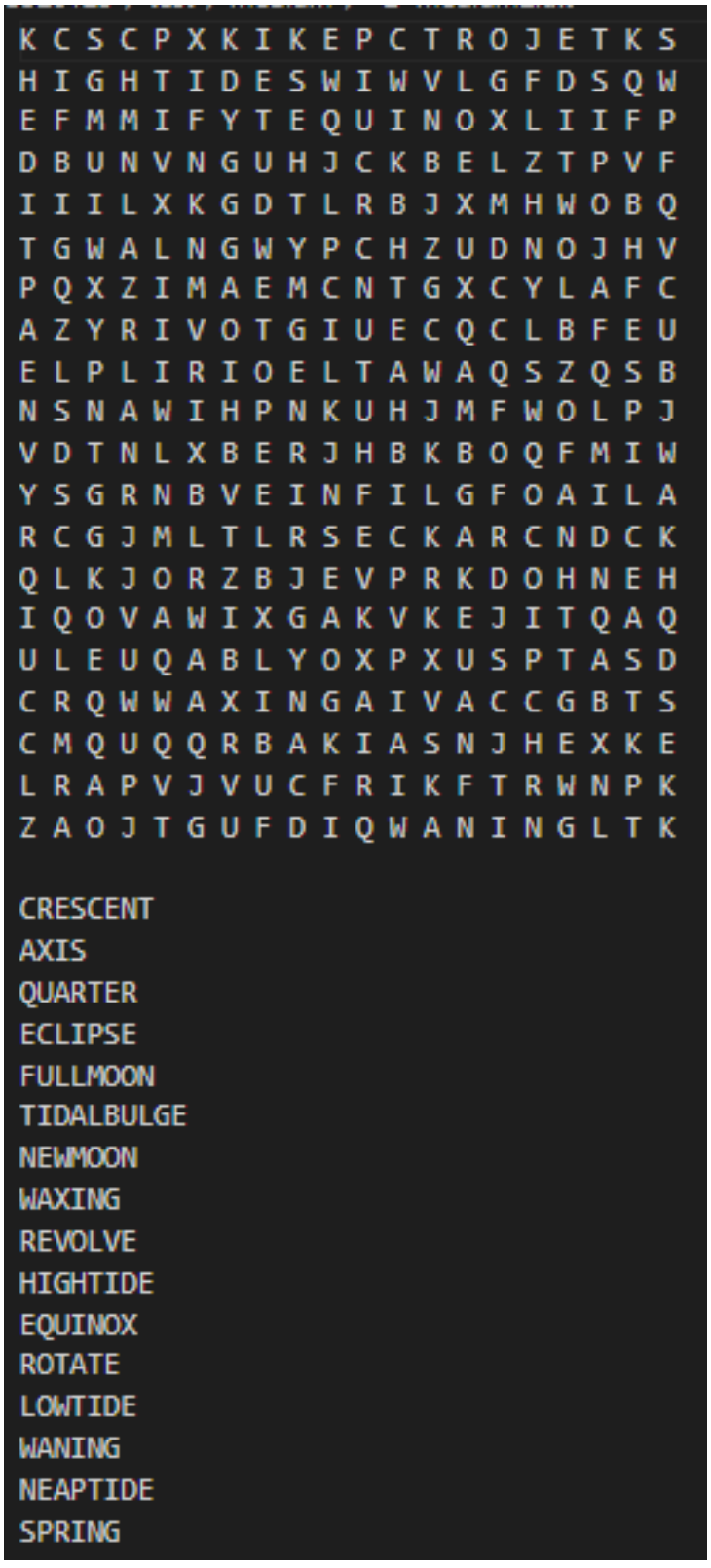

GNILCYC




```
Waktu eksekusi program: 0.2343750000 seconds.
Total perbandingan huruf: 4629.
```

e. Puzzle berukuran medium 2

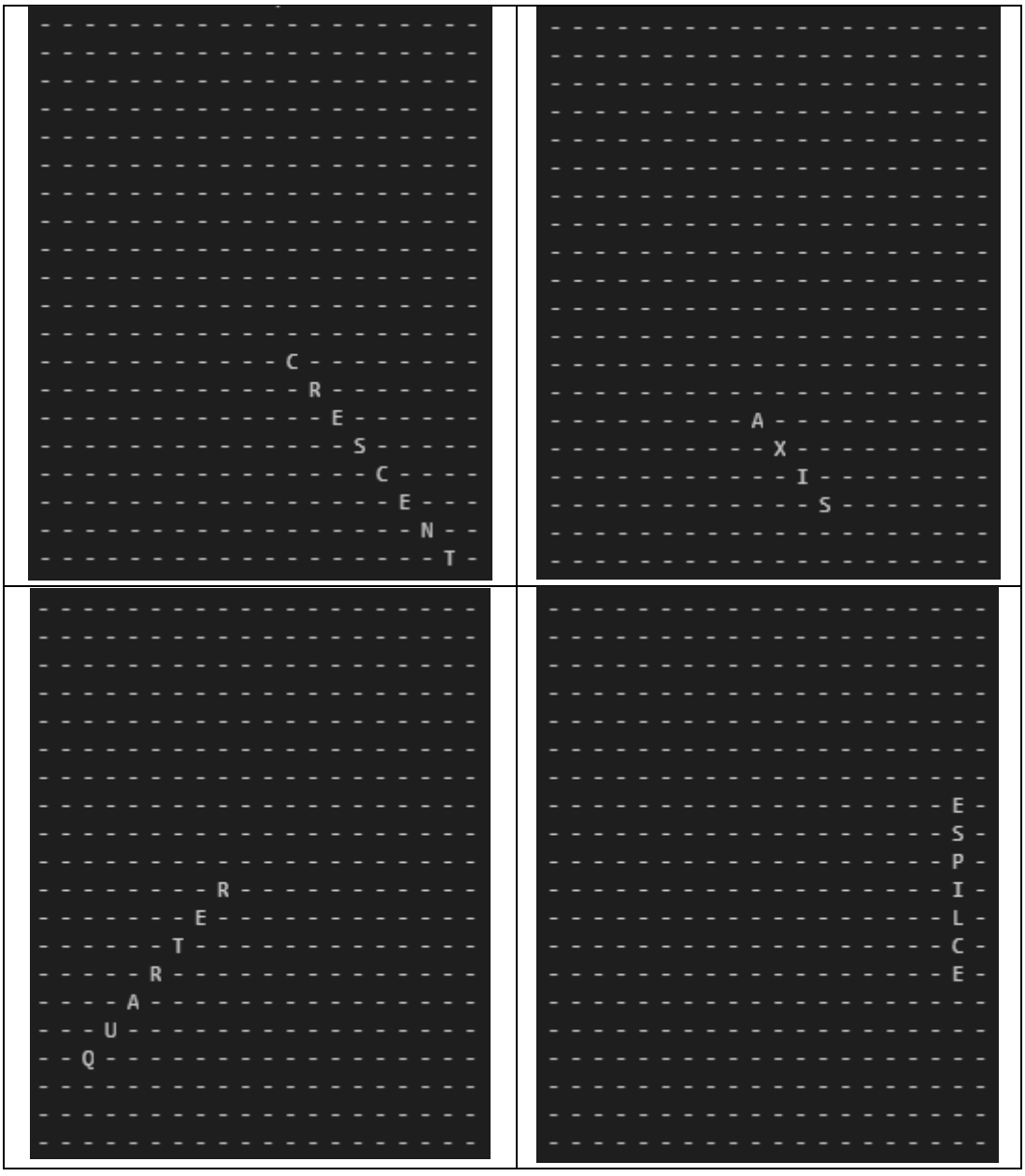
1. Input program

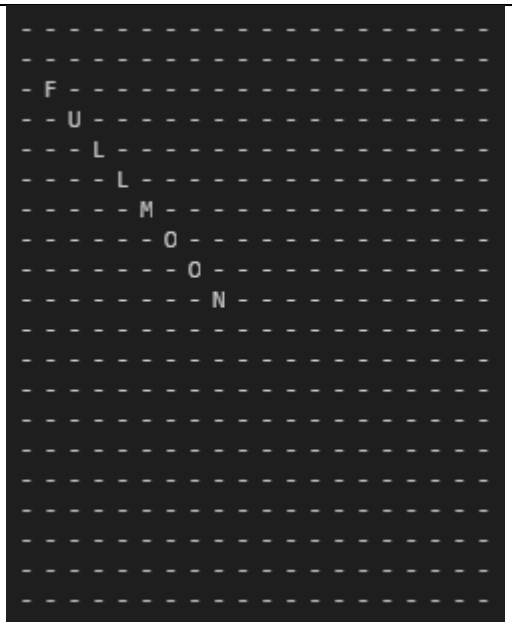


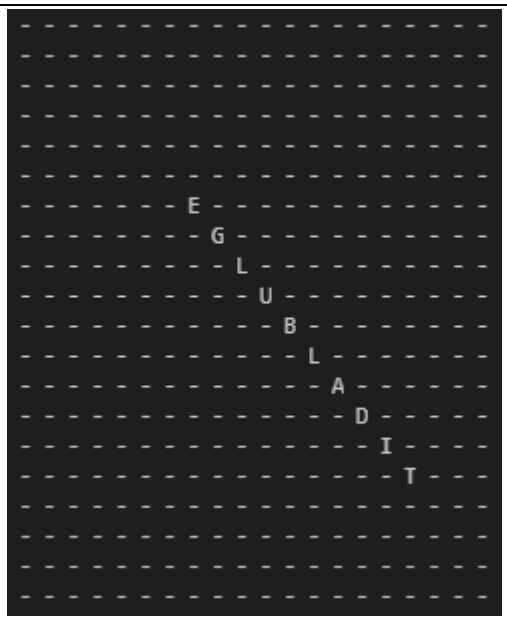
```
K C S C P X K I K E P C T R O J E T K S
H I G H T I D E S W I W V L G F D S Q W
E F M M I F Y T E Q U I N O X L I I F P
D B U N V N G U H J C K B E L Z T P V F
I I I L X K G D T L R B J X M H W O B Q
T G W A L N G W Y P C H Z U D N O J H V
P Q X Z I M A E M C N T G X C Y L A F C
A Z Y R I V O T G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

CRESCENT
AXIS
QUARTER
ECLIPSE
FULLMOON
TIDALBULGE
NEWMOON
WAXING
REVOLVE
HIGHTIDE
EQUINOX
ROTATE
LOWTIDE
WANING
NEAPTIDE
SPRING
```

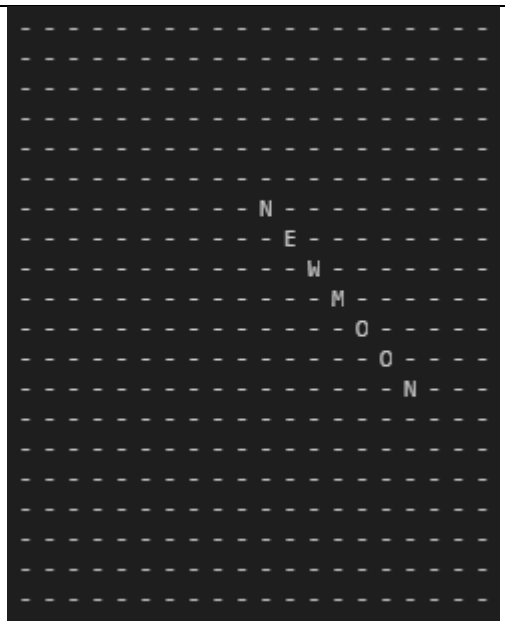
2. Output program

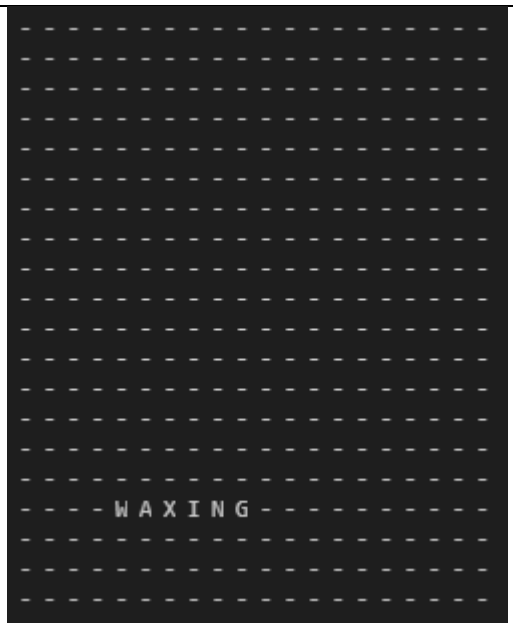
```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/medium/medium2.txt
```

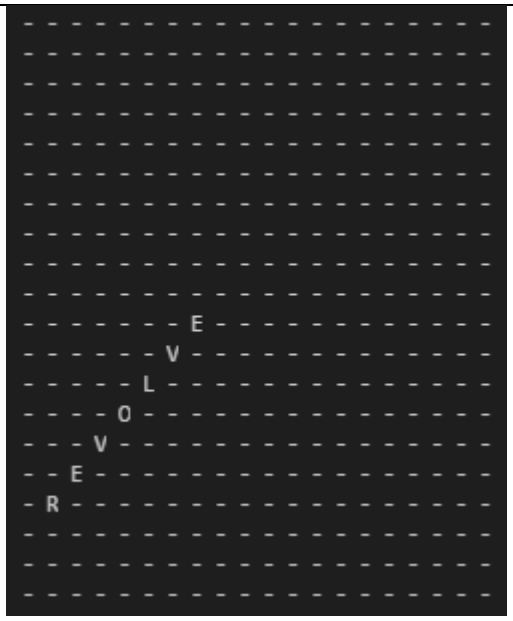


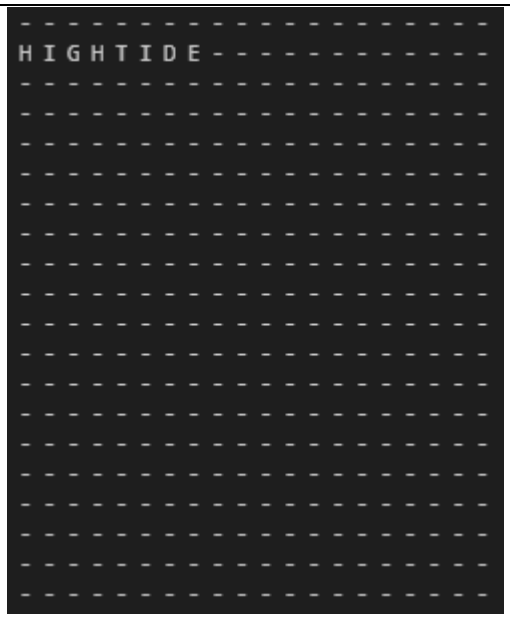

FULL
MOON

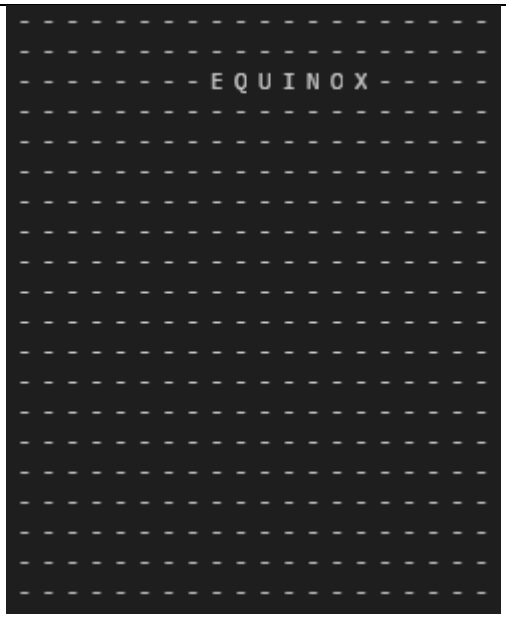

EGLUBLADIT

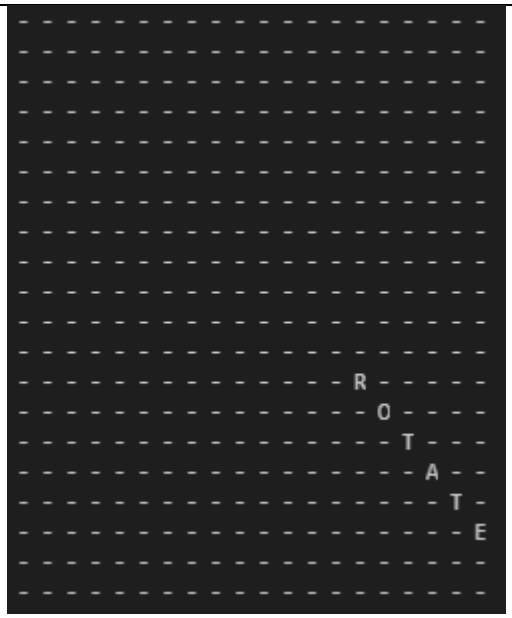

NEW
MOON

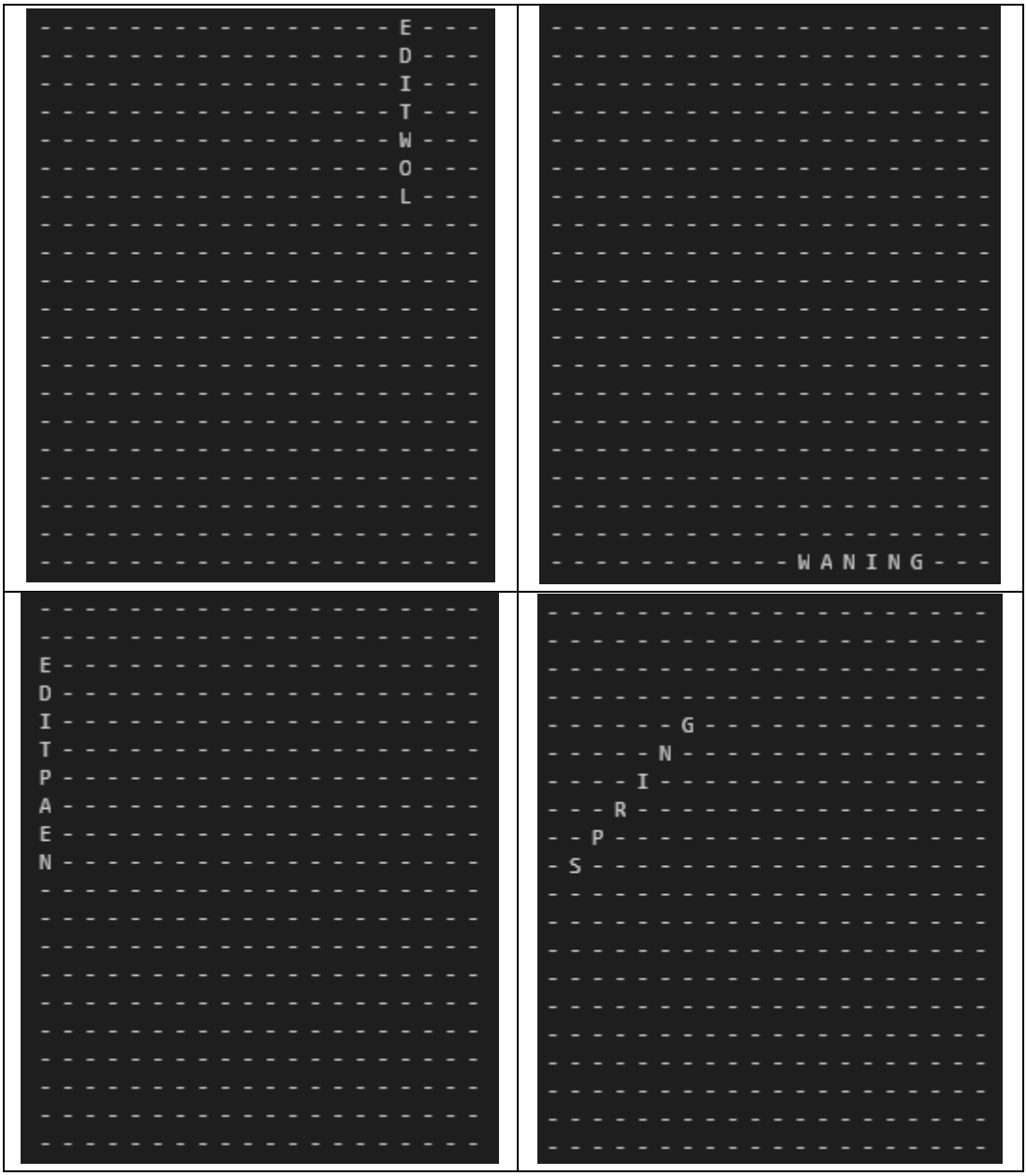

WAXING


REVOLVE


HIGHTIDE


EQUINOX


ROTATE




```
Waktu eksekusi program: 0.2343750000 seconds.
Total perbandingan huruf: 5269.
```

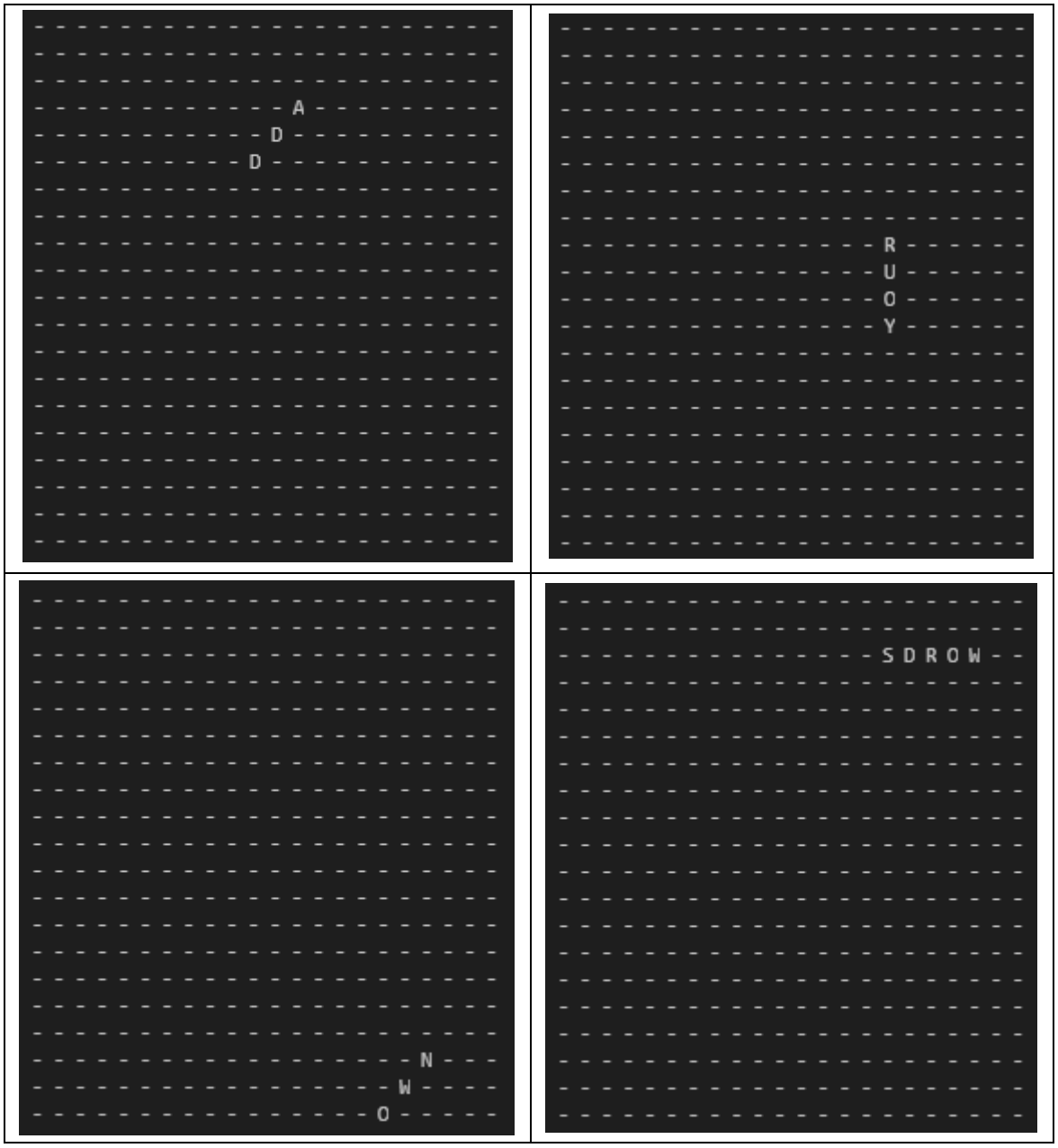
f. Puzzle berukuruan medium 3

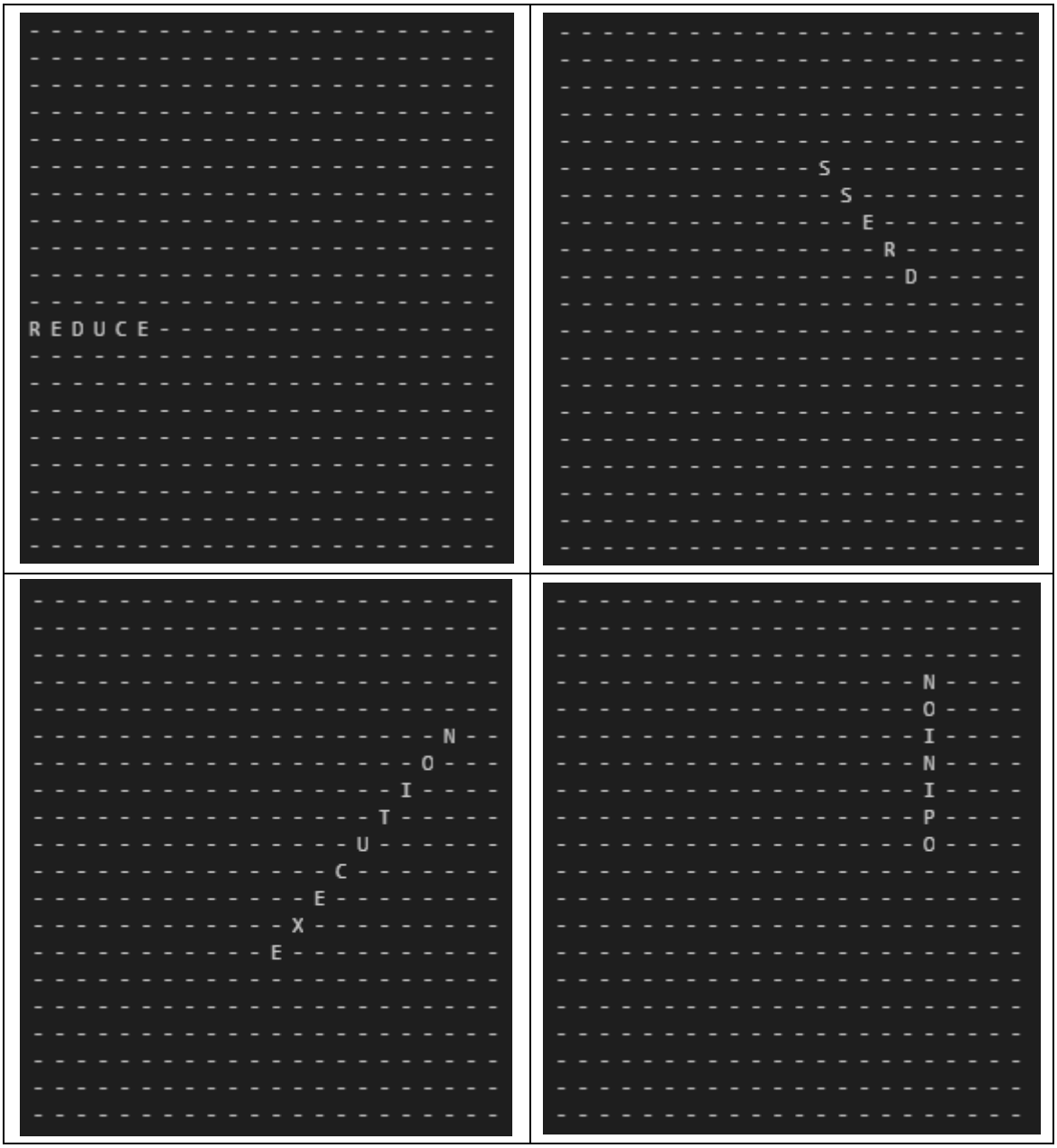
1. Input program

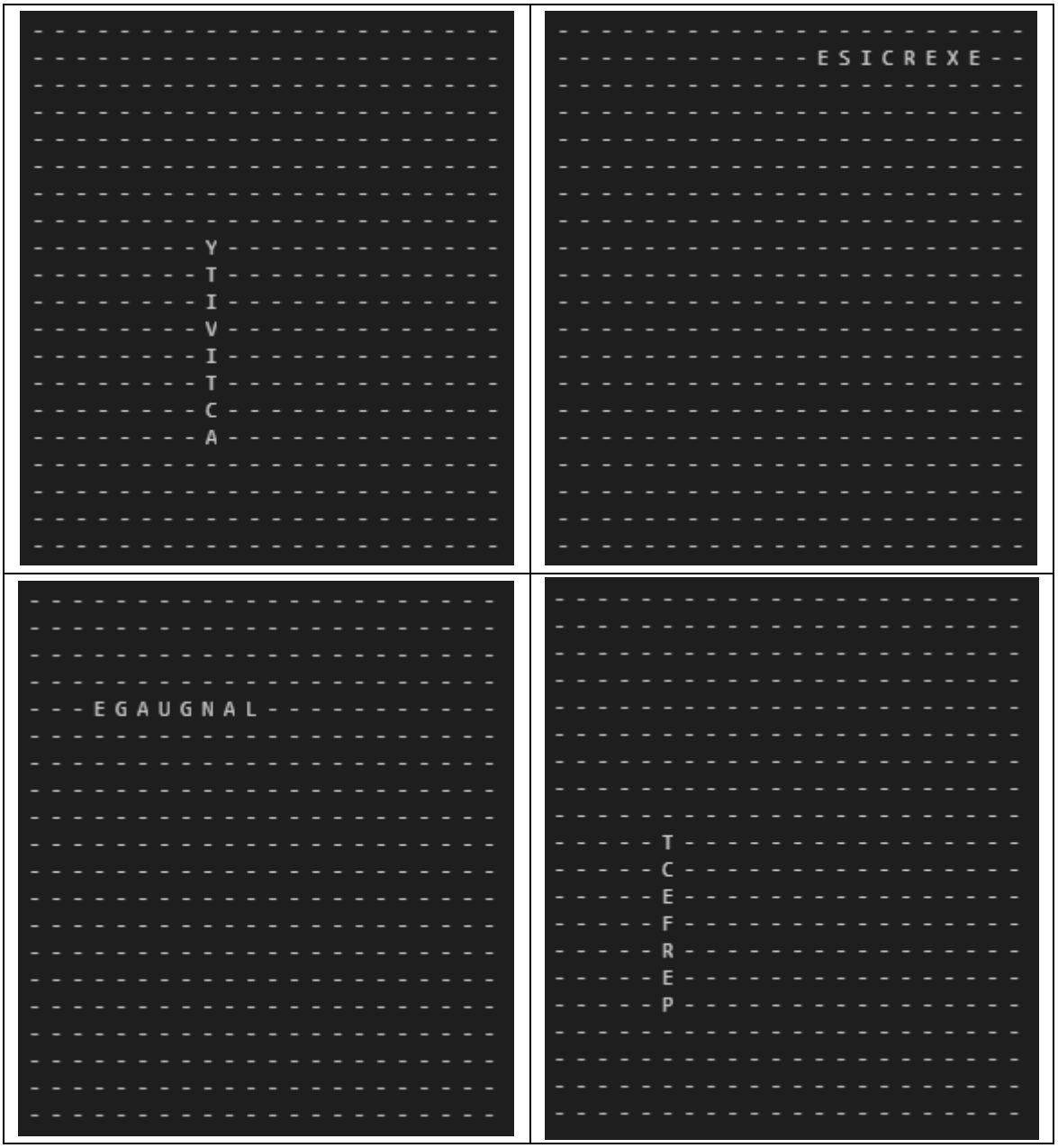
```
E F D E H D A T T S Y F C L F W O L L O F R
R D H P E N S G C G E R E S I C R E X E I Y
I E E N T S I A N L F O N C I S D R O W E G
L E U T O T D E T C E D A O O S T N X R R C
D D N E G A U G N A L D R Y U A H O T R F P
Y P E O I S A S E Y D E S G T P V I I N I O
E V P E I C H E T G I M E S X G S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

ADD
YOUR
OWN
WORDS
REDUCE
DRESS
EXECUTION
OPINION
ACTIVITY
EXERCISE
LANGUAGE
PERFECT
COMPLETE
GUARANTEED
INDEPENDENT
FREELANCE
ANYTHING
PATIENT
DEMOLISH
FOLLOW
```

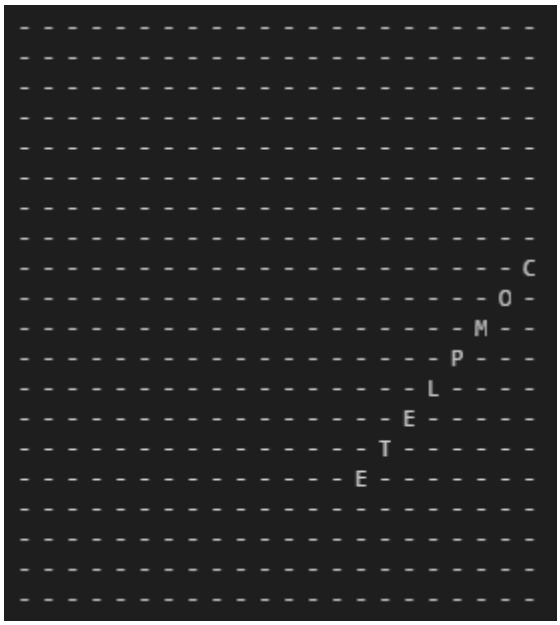
2. Output program

```
PS D:\Program\C\Tucil Stigma\Tugas 1\Tucil1_13520125> bin/main.exe
Masukkan nama file input: test/medium/medium3.txt
```

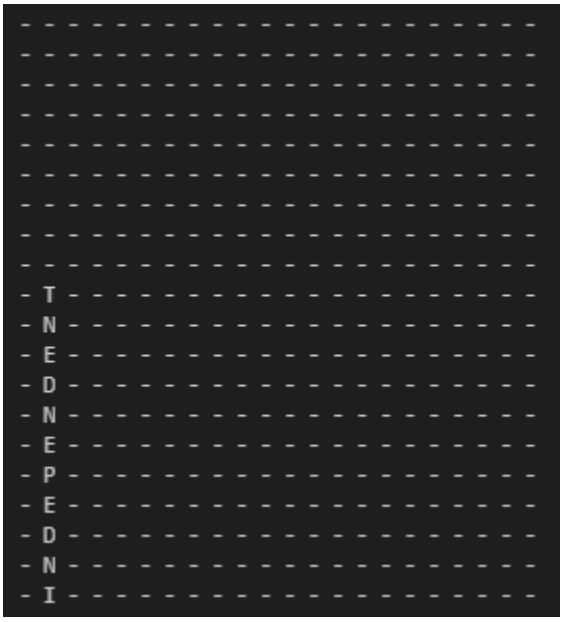


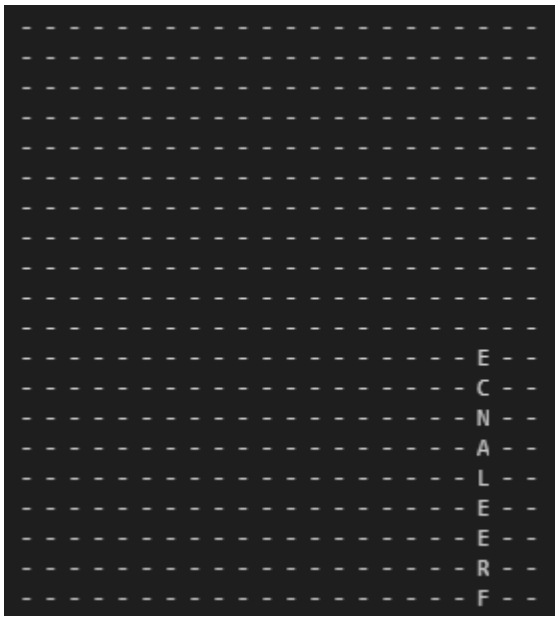

REDUCE
SSERD
EXECUTION
NOITPO

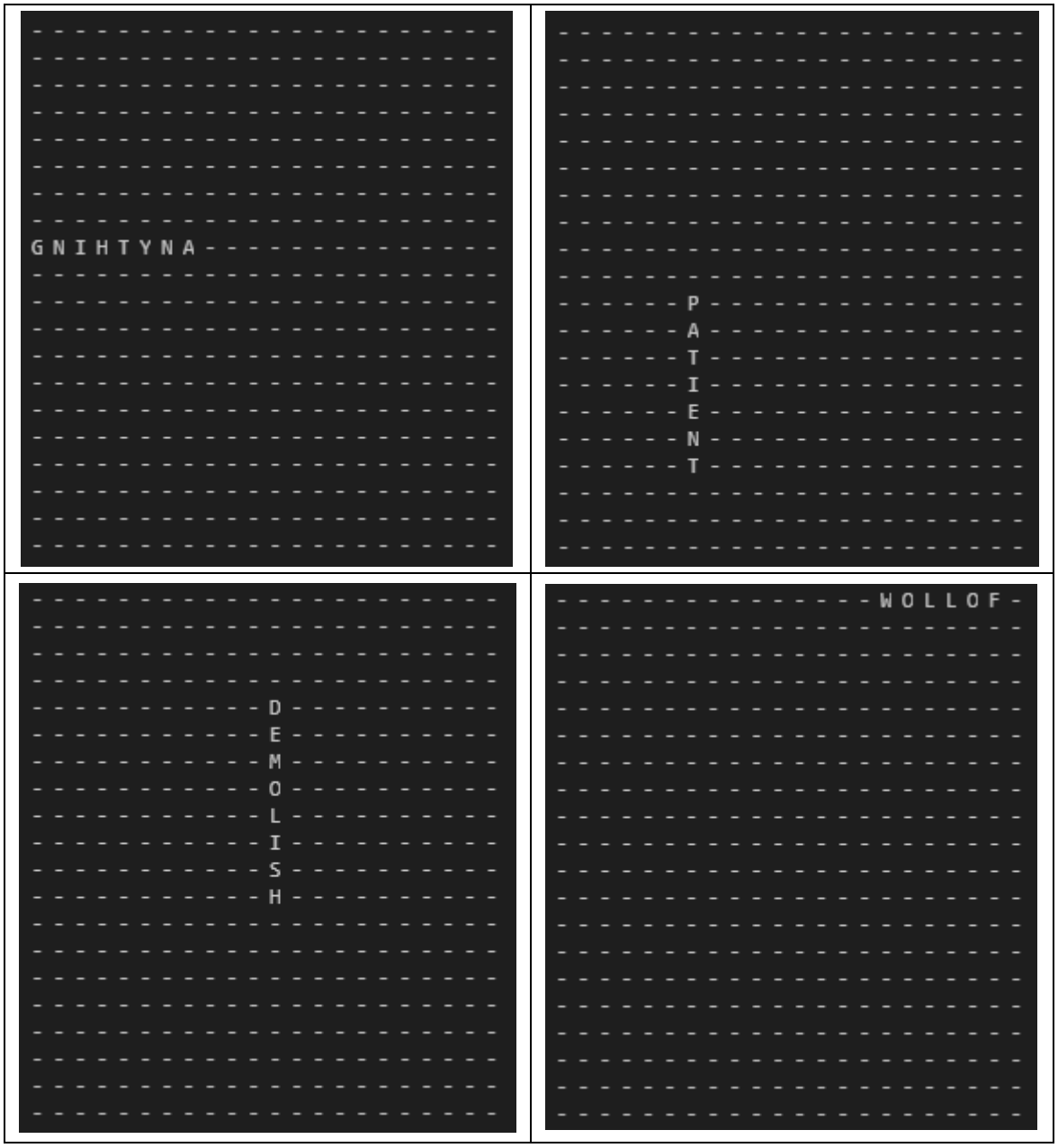

YTIVITCA
ESICREXE
EGAUGNAL
TCEFREP


COMPLETE


DEETNARAUG


TNEDNEPEDNI

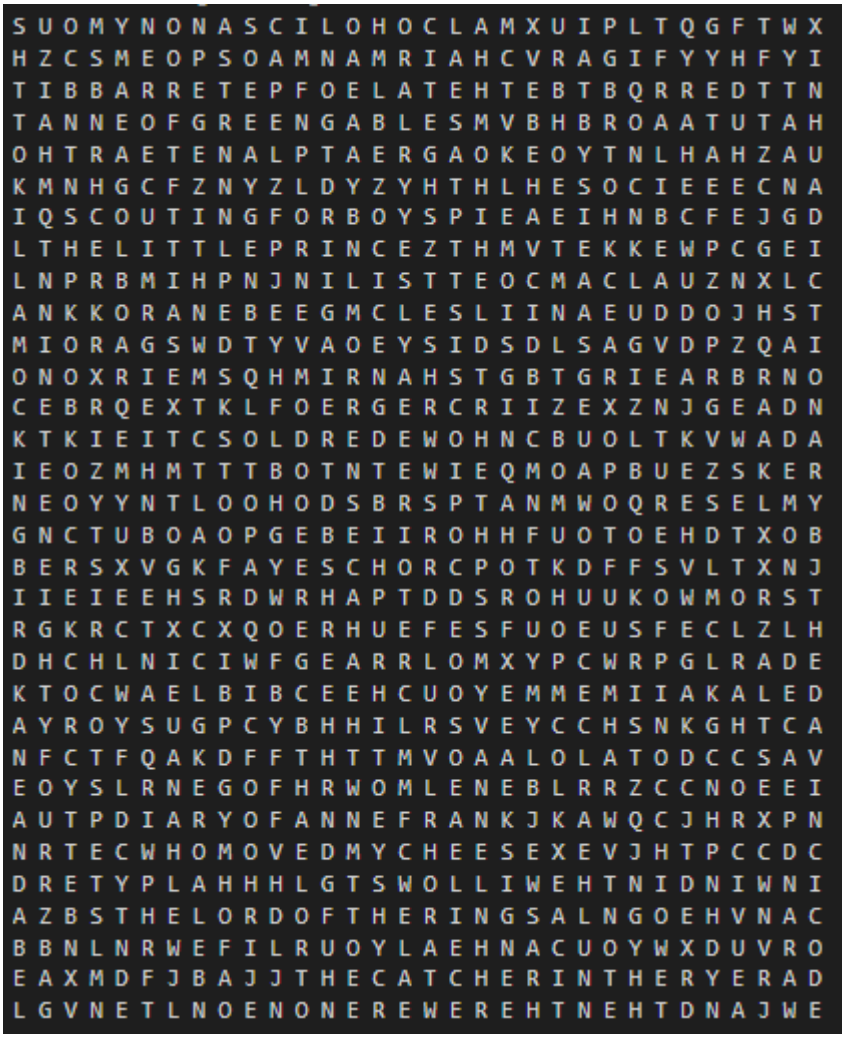

ECNALEERF




```
Waktu eksekusi program: 0.2500000000 seconds.
Total perbandingan huruf: 7528.
```

g. Puzzle berukuran large 1

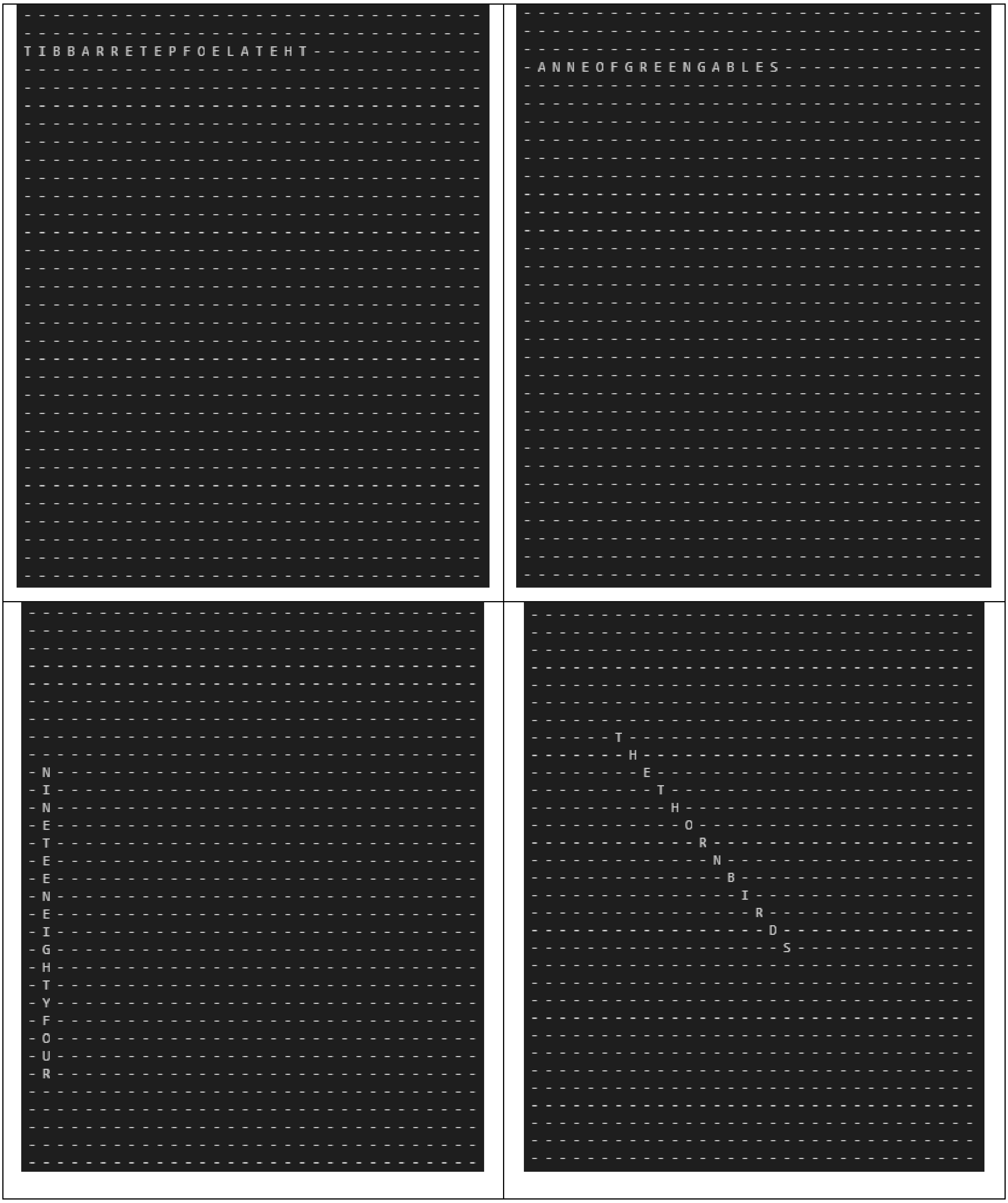
1. Input program

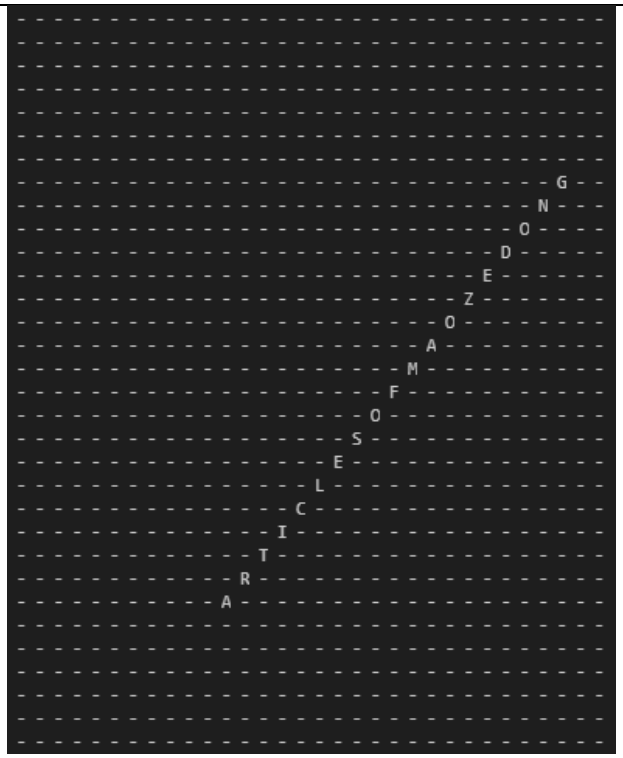


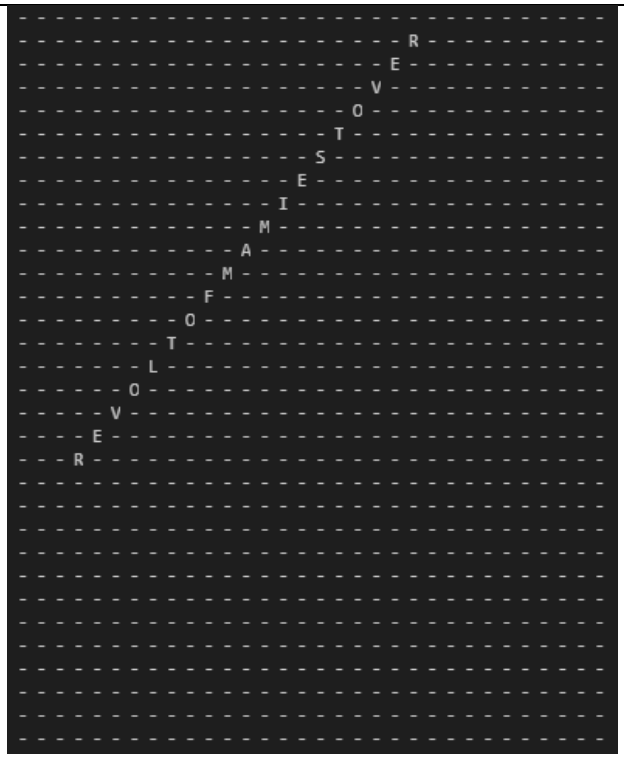
```
S U O M Y N O N A S C I L O H O C L A M X U I P L T Q G F T W X
H Z C S M E O P S O A M N A M R I A H C V R A G I F Y Y H F Y I
T I B B A R R E T E P F O E L A T E H T E B T B Q R R E D T T N
T A N N E O F G R E E N G A B L E S M V B H B R O A A T U T A H
O H T R A E T E N A L P T A E R G A O K E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

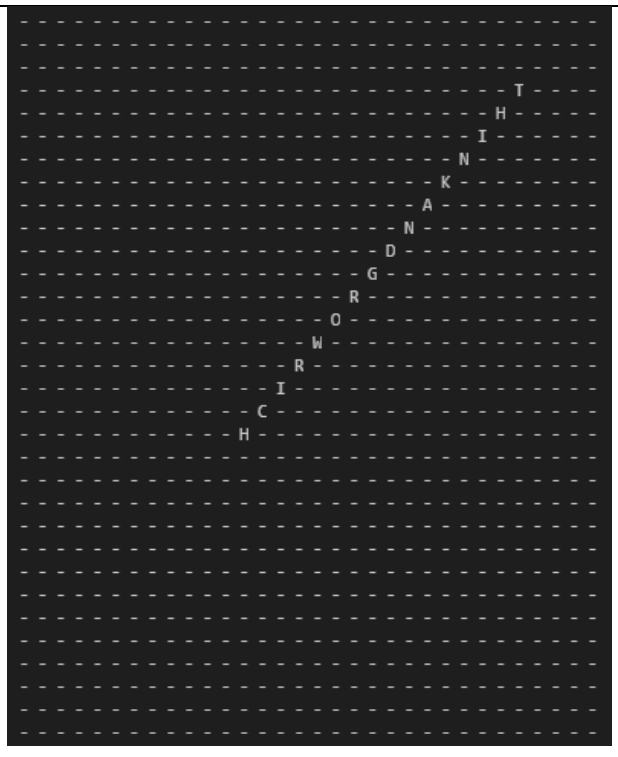
```
A MESSAGE TO GARCIA
GREAT PLANET EARTH
THE HOBBIT
A TALE OF TWO CITIES
HARRY POTTER
THE LITTLE PRINCE
ADVENTURES OF PINOCCHIO
KANE AND ABEL
THE LITTLE RED BOOK
ALCOHOLICS ANONYMOUS
KORAN
THE LORD OF THE RINGS
AND THEN THERE WERE NONE
LOVE STORY
THE PURPOSE DRIVEN LIFE
ANGELS AND DEMONS
LOVE YOU FOREVER
THE TALE OF PETER RABBIT
ANNE OF GREEN GABLES
NINETEEN EIGHTY FOUR
THE THORN BIRDS
ARTICLES OF MAO ZEDONG
REVOLT OF MAMIE STOVER
THINK AND GROW RICH
BETTY CROCKER COOKBOOK
ROGETS THESAURUS
TO KILL A MOCKINGBIRD
BIBLE
SCOUTING FOR BOYS
VALLEY OF THE DOLLS
BLACK BEAUTY
SHE
WAR AND PEACE
BOOK OF MORMON
STEPS TO CHRIST
WEBSTERS DICTIONARY
CHAIRMAN MAOS POEMS
THE ALCHEMIST
WHO MOVED MY CHEESE
CHARLOTTES WEB
THE CATCHER IN THE RYE
WIND IN THE WILLOWS
DIARY OF ANNE FRANK
THE CELESTINE PROPHECY
XINHUA DICTIONARY
DREAM OF THE RED CHAMBER
THE DA VINCI CODE
YOU CAN HEAL YOUR LIFE
```

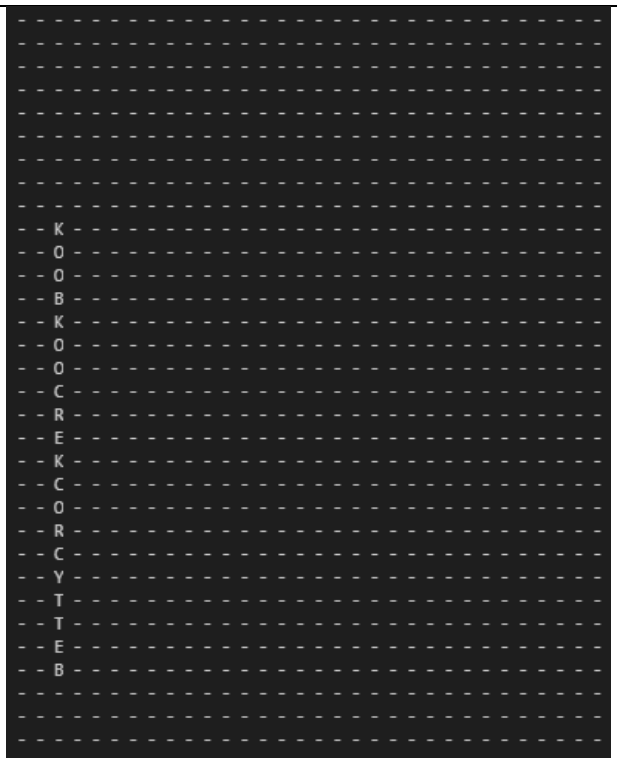
2. Output program
   Karena keterbatasan ruang pada terminal, hanya beberapa kata saja yang dapat dilihat hasil outputnya.

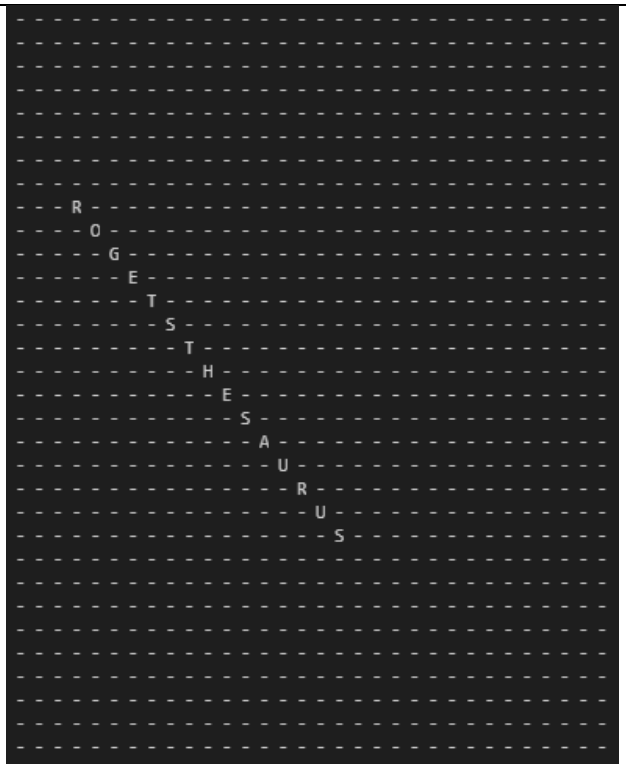


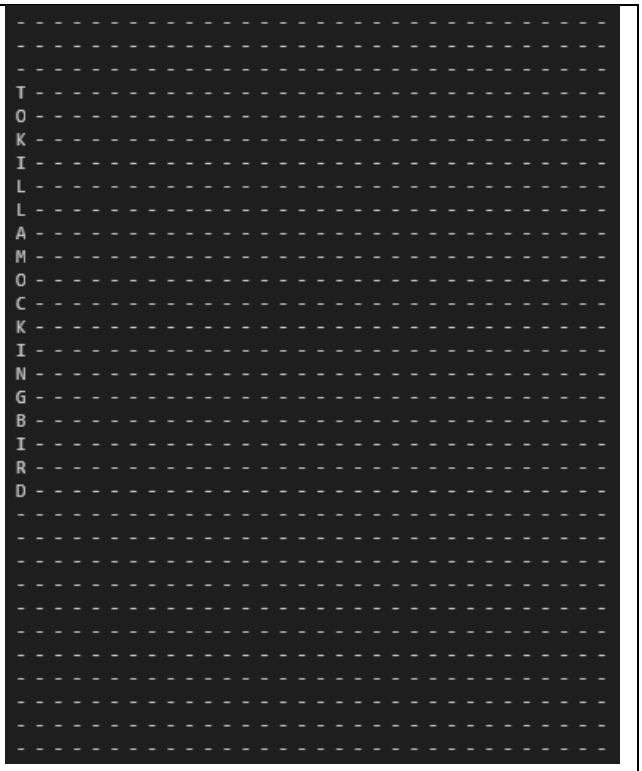


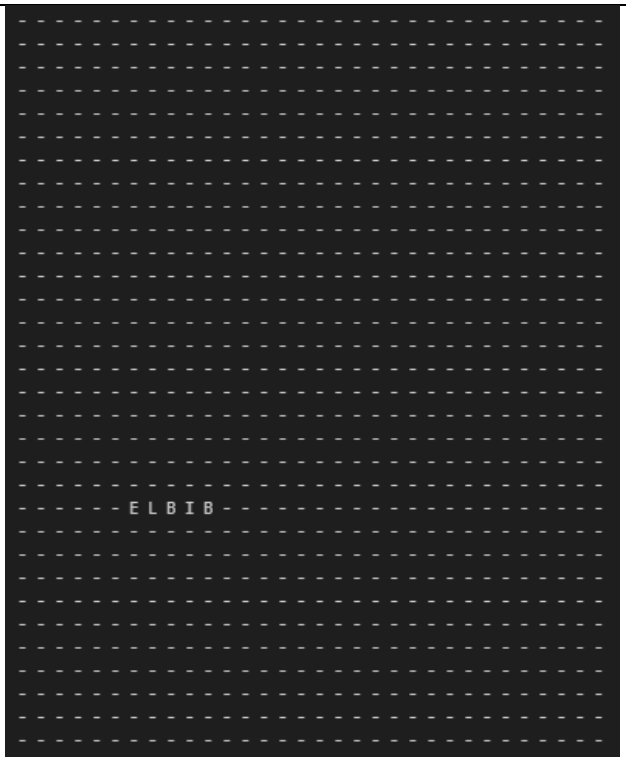


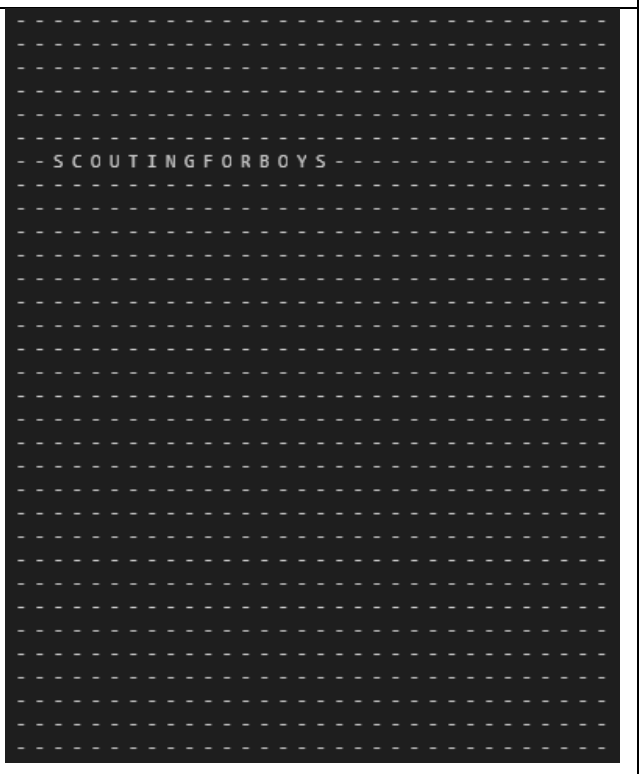


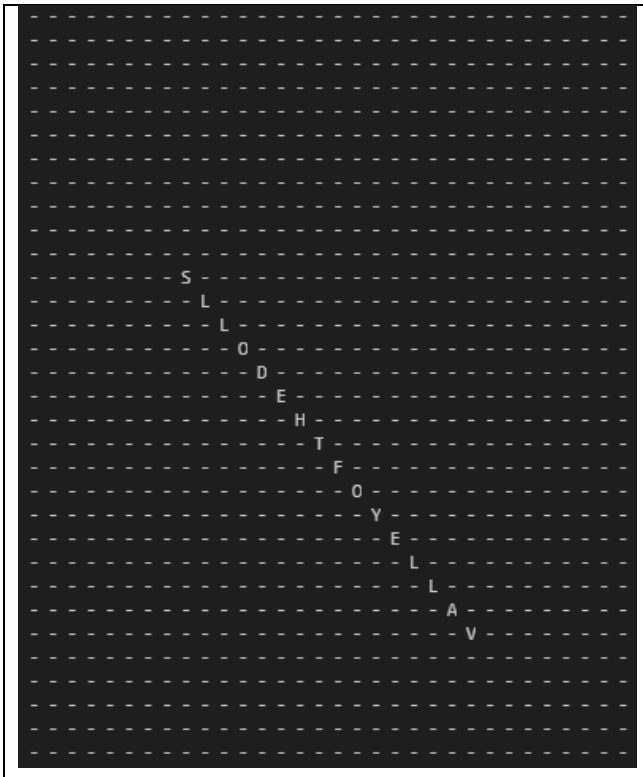


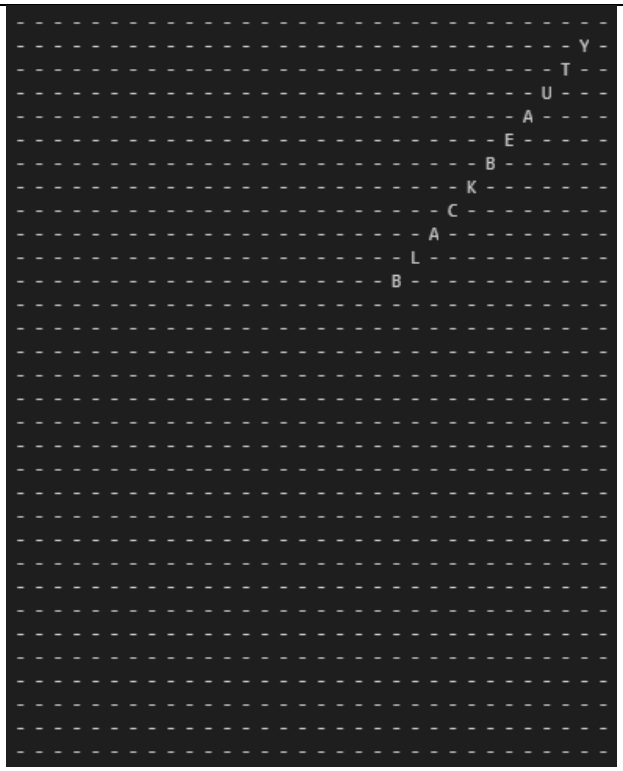

R
O
G
E
T
S
T
H
E
S
A
U
R
U
S

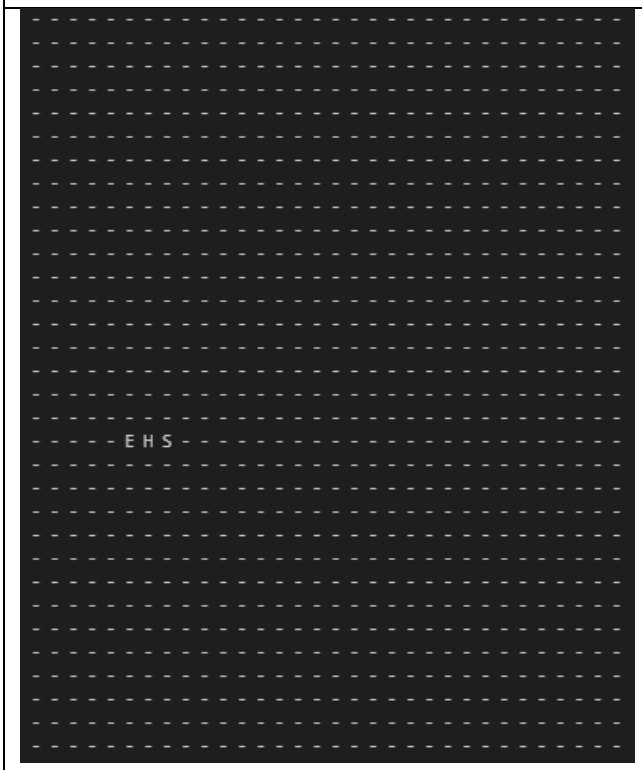

TOKILLAMOCKINGBIRD

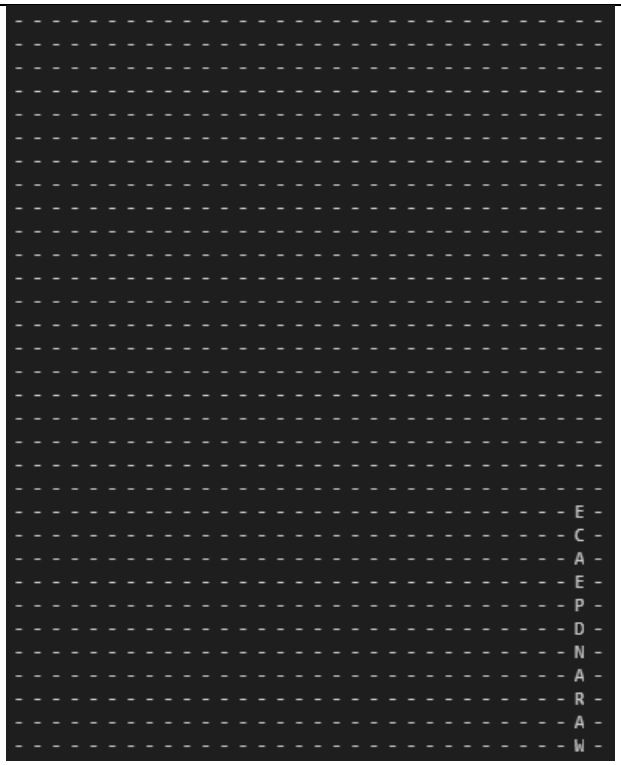

ELBIB

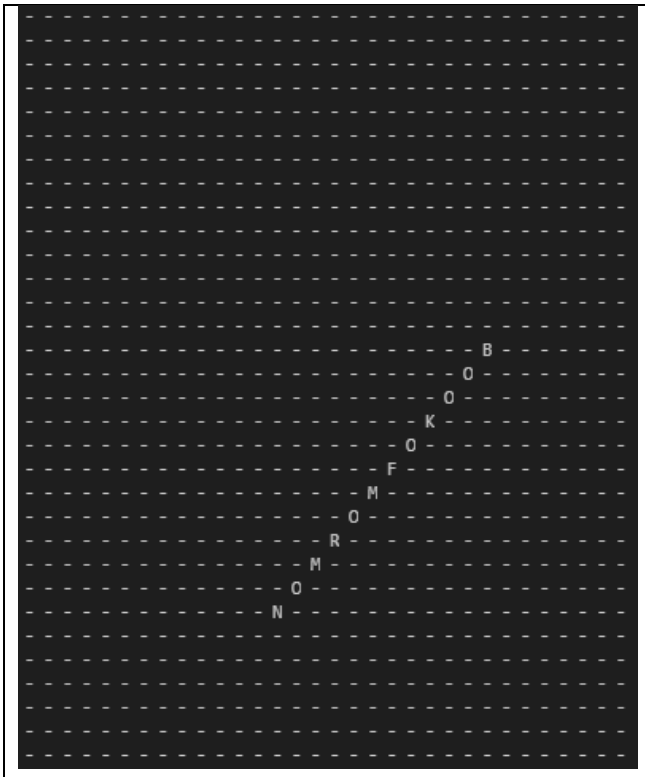

SCOUTINGFORBOYS

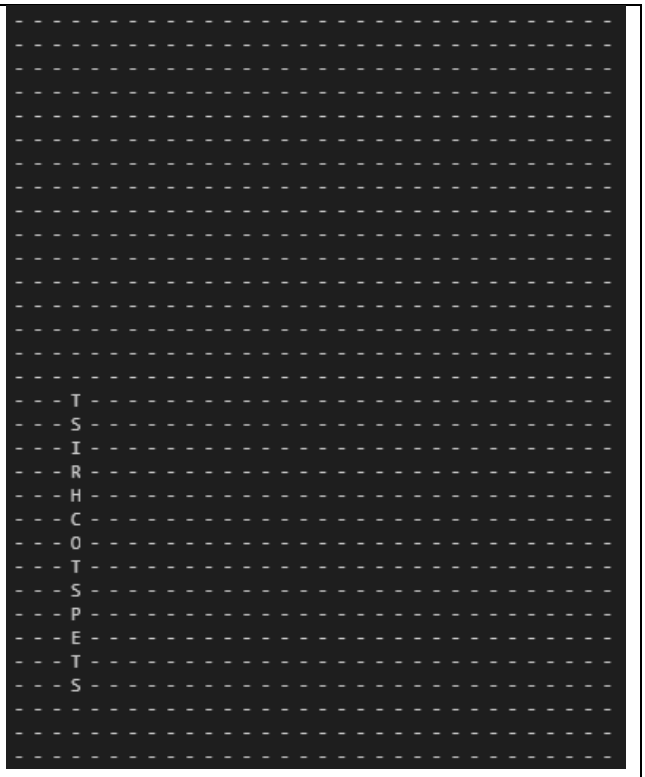

S
L
L
O
D
E
H
T
F
O
Y
E
L
L
A
V

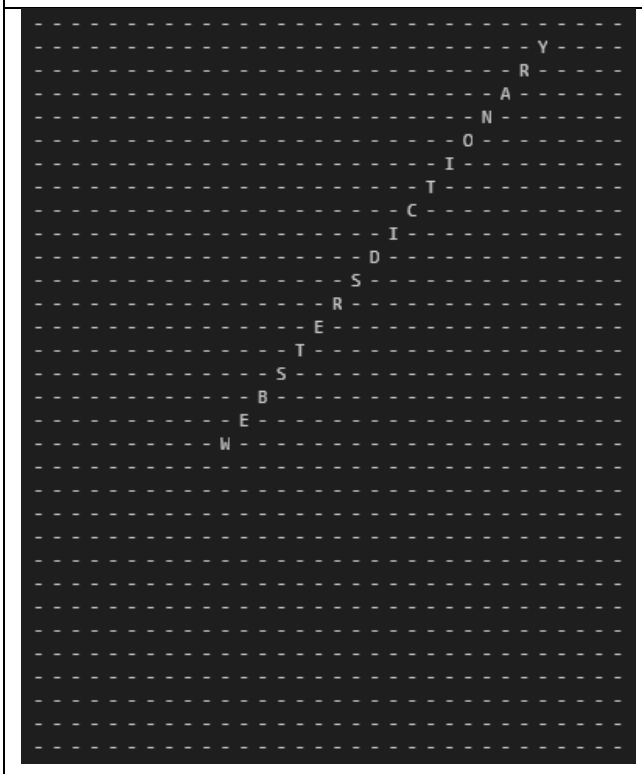

Y
T
U
A
E
B
K
C
A
L
B

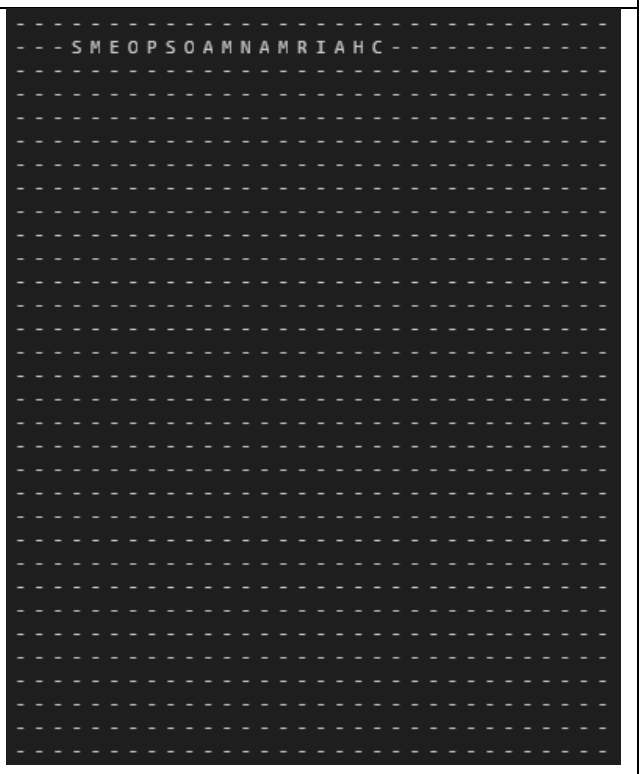

E H S

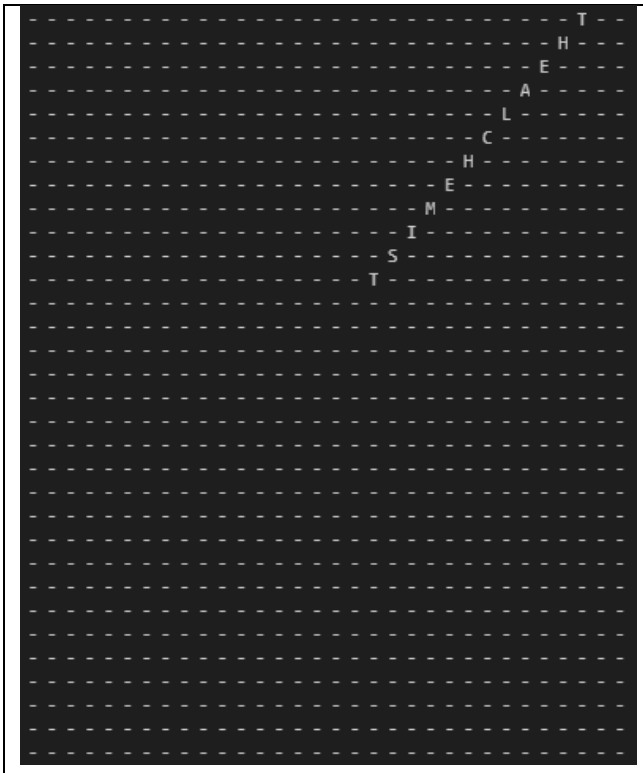

E
C
A
E
P
D
N
A
R
A
W

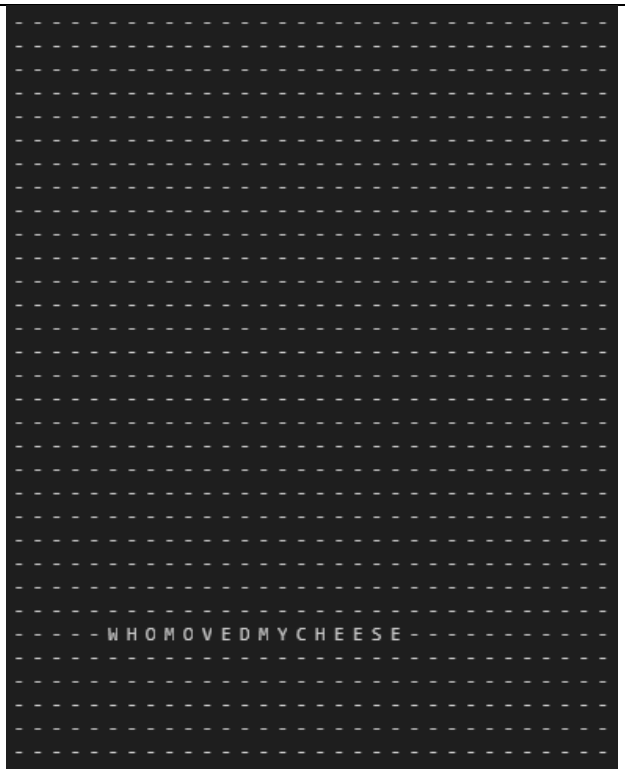

B
O
O
K
O
F
M
O
R
M
O
N

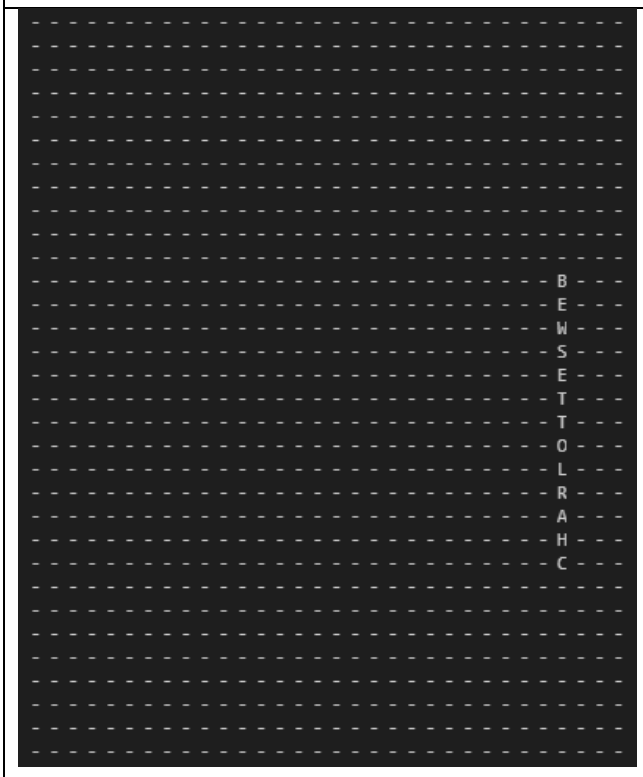

T
S
I
R
H
C
O
T
S
P
E
T
S

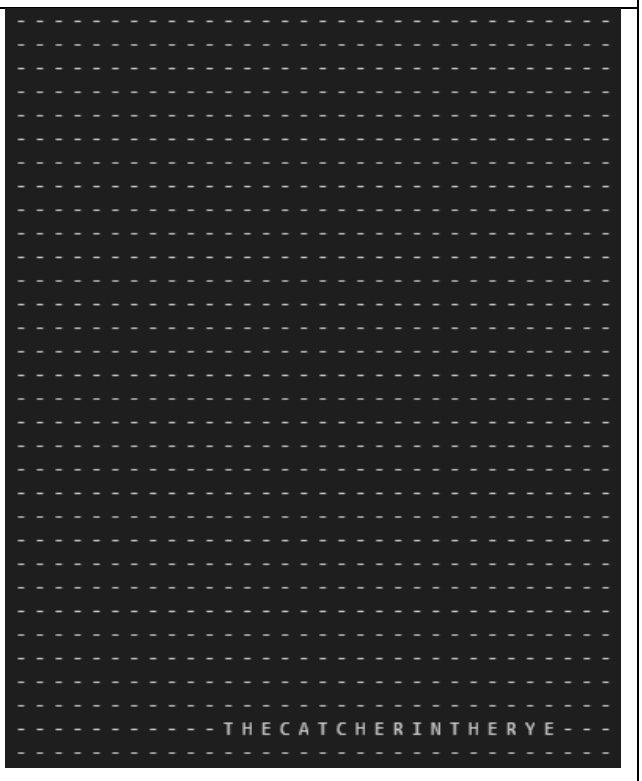

Y
R
A
N
O
I
T
C
I
D
S
R
E
T
S
B
E
W

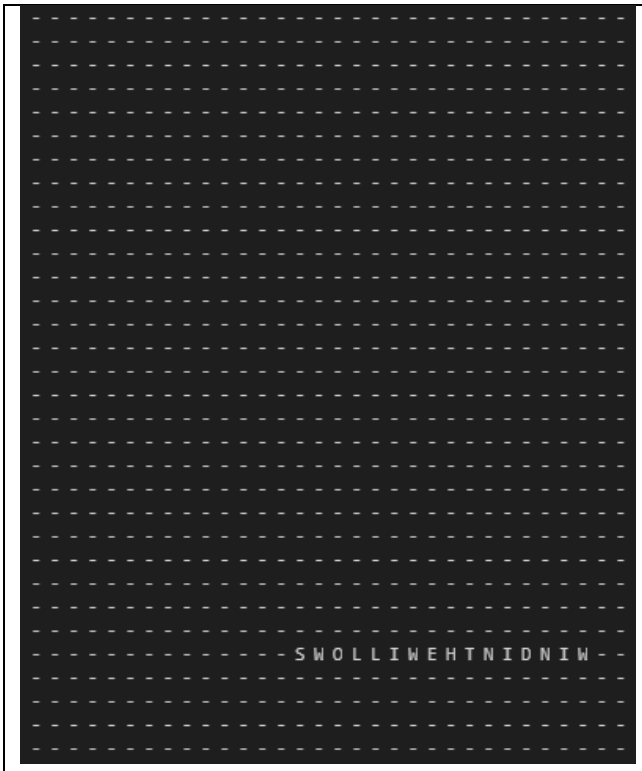

SMEOPSOAMNAMRIAHC

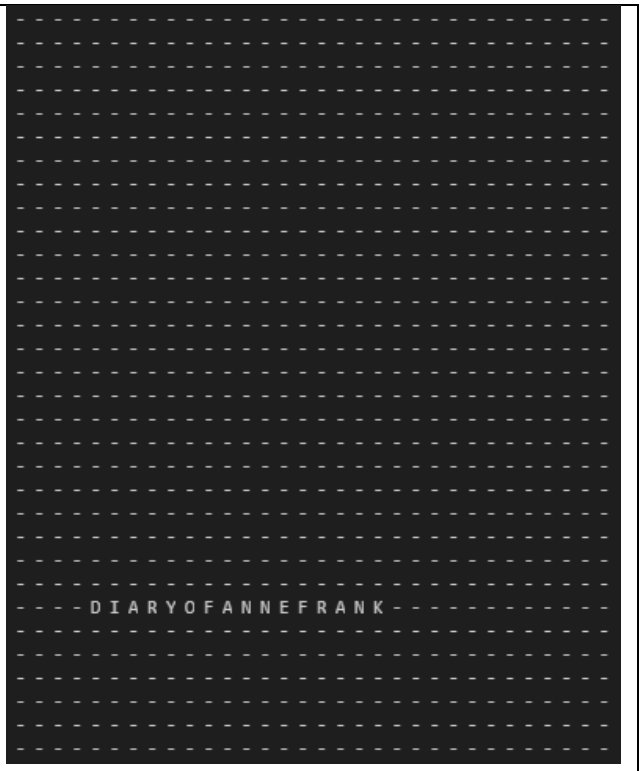

THECHEESEMOVED

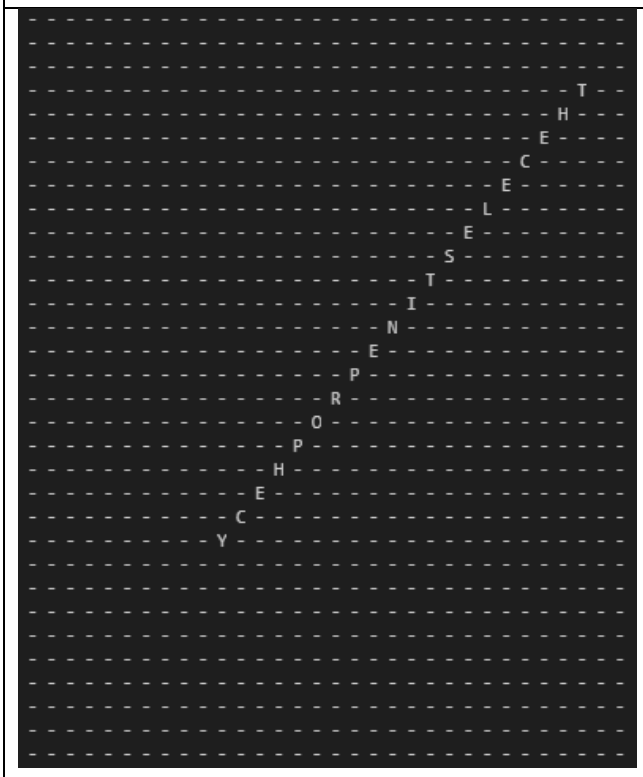

WHOMOVEDMYCHEESE

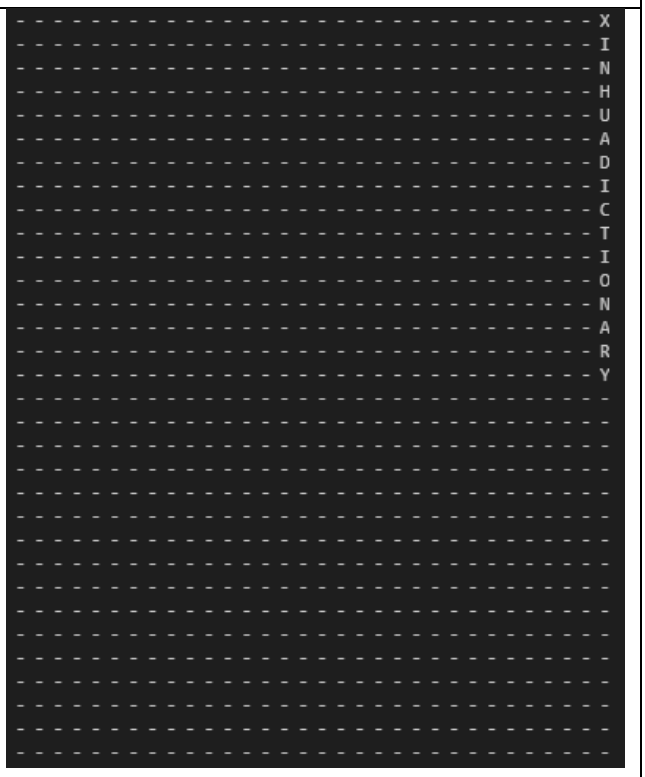

BEWSETTOLRAHC

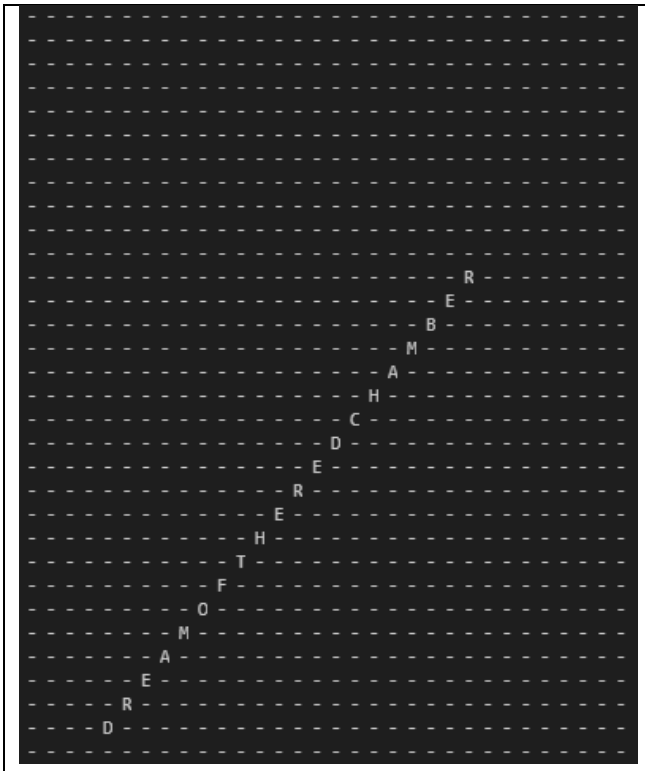

THECATCHERINTHERYE

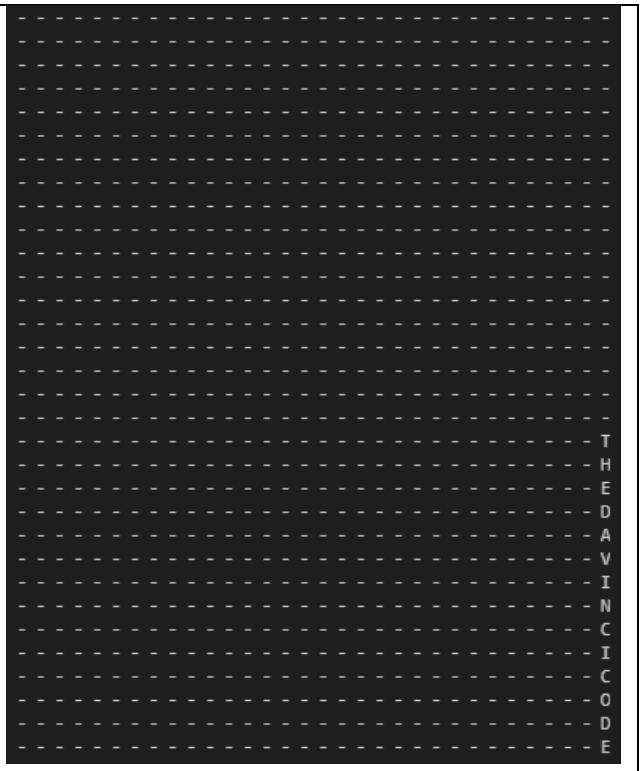

SWOLLIWEHTNIDNIW

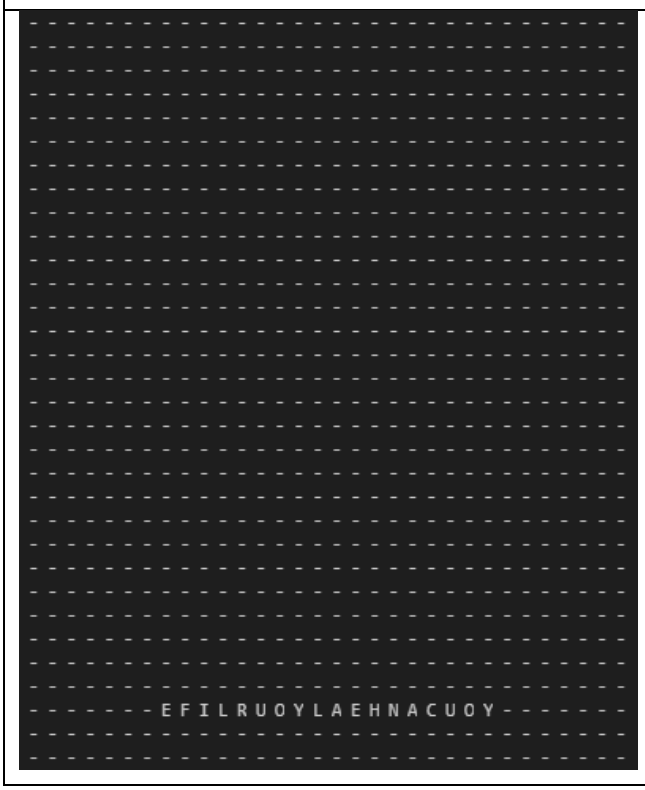

DIARYOFANNEFRANK


T
H
E
C
E
L
E
S
T
I
N
E
P
R
O
P
H
E
C
Y


X
I
N
H
U
A
D
I
C
T
I
O
N
A
R
Y



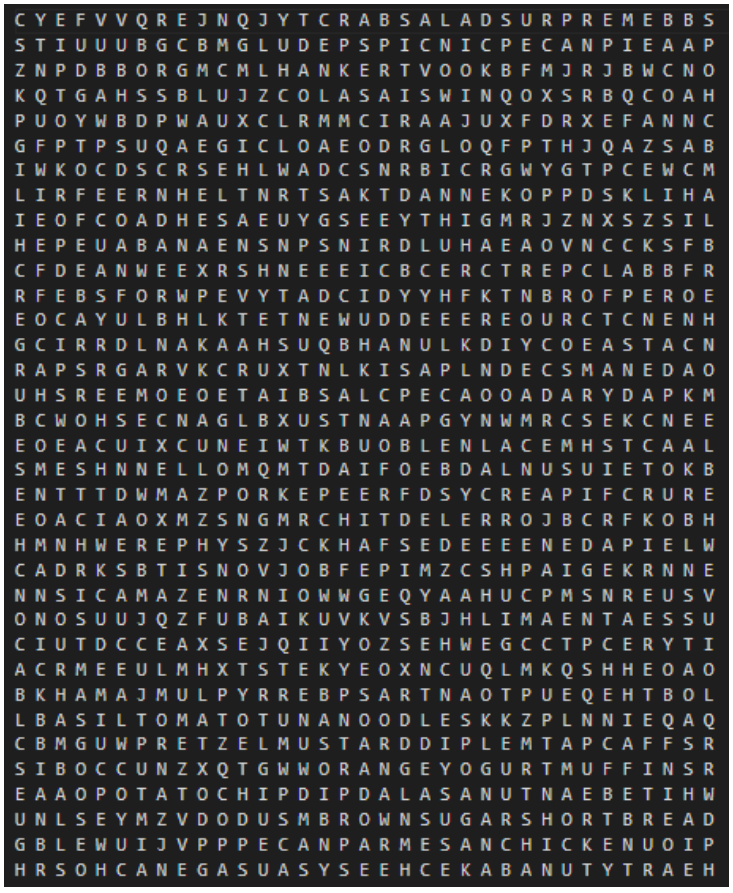





```
Waktu eksekusi program: 1.2968750000 seconds.
Total perbandingan huruf: 33348.
```

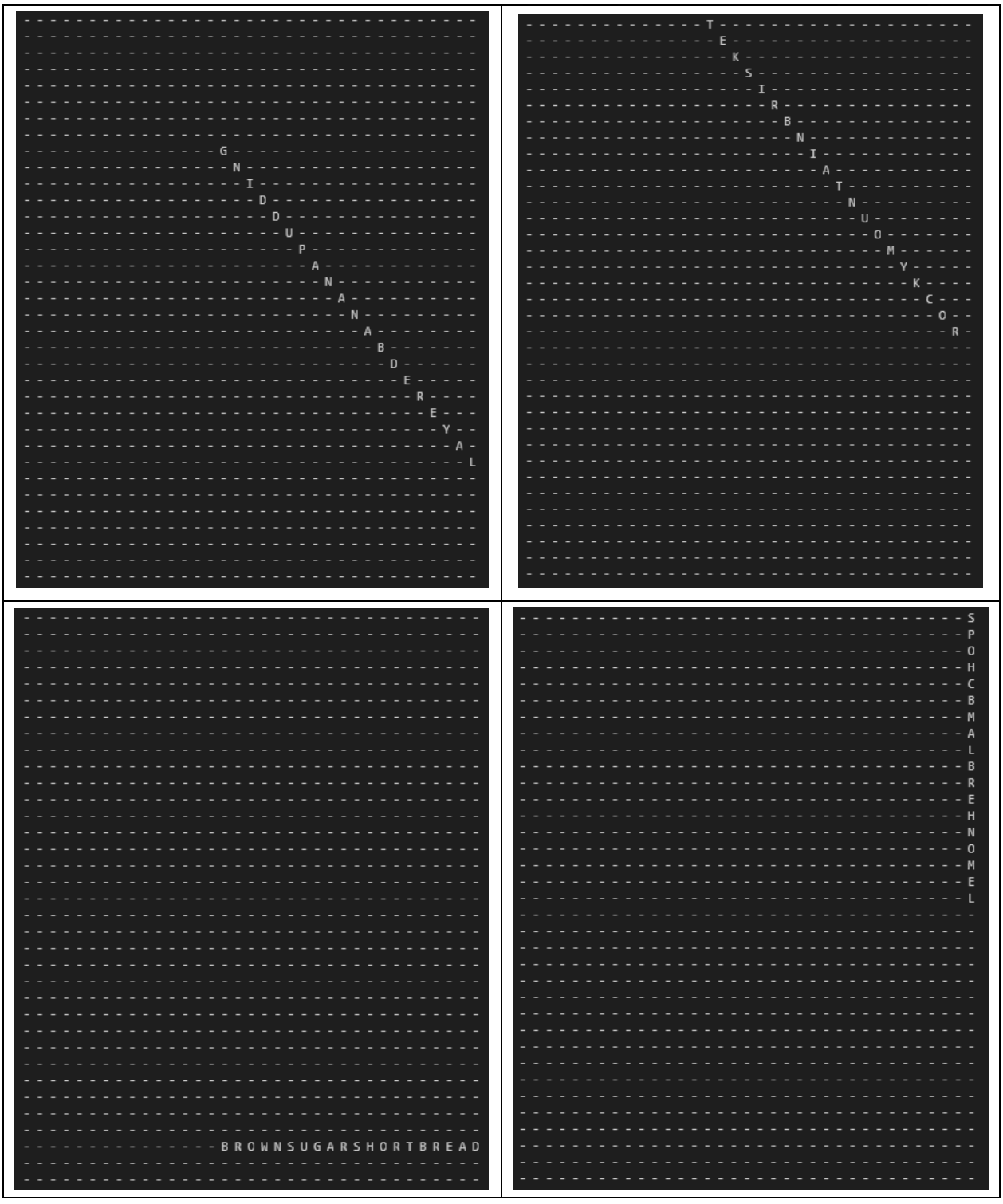
h. Puzzle berukuran large 2

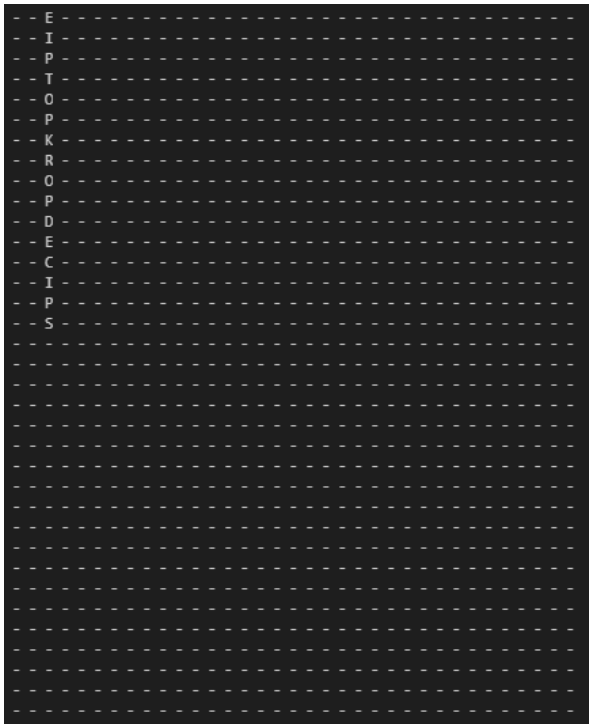
1. Input Program

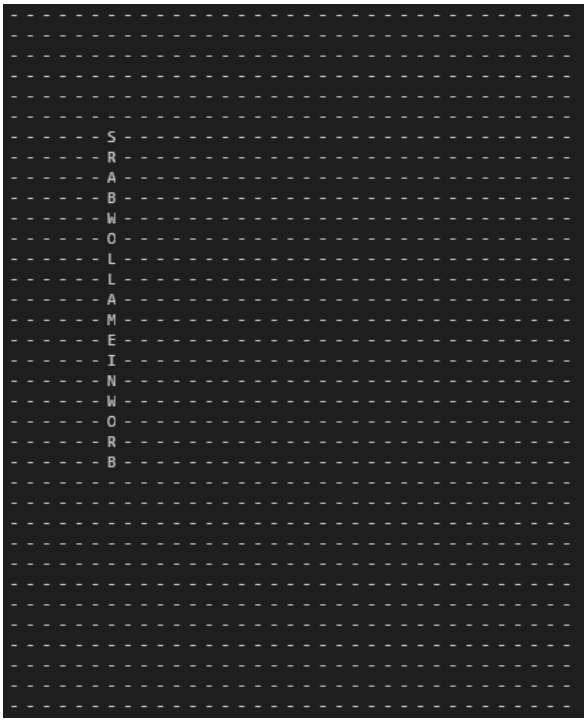


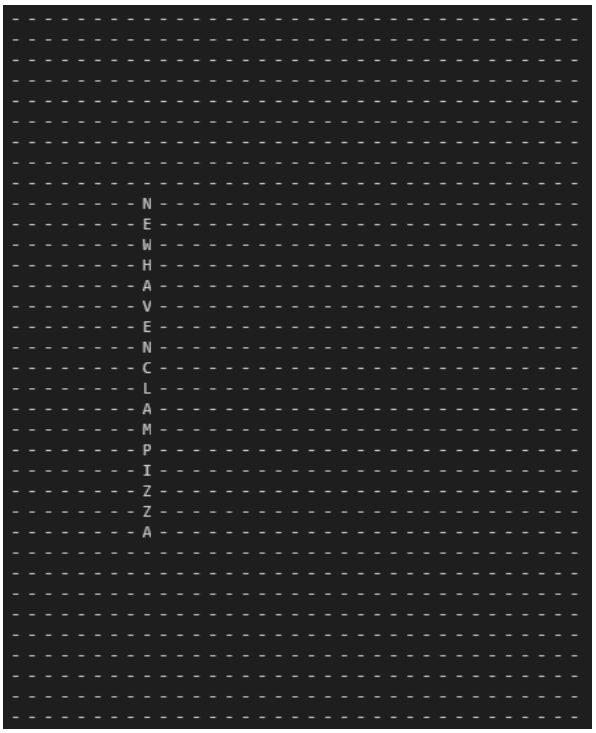
ALMOND TURKEY CASSEROLE
DUCK WITH CHERRY SAUCE
PESTO PORK ROAST
APPLE HAM STEAK
FIESTA CHEESE BALL
PICNIC PECAN PIE
BACON CHEESEBURGER CHILI
FUDGE SUNDAE SAUCE
PLUM GLAZED RIBS
BACON SWISS BREAD
FUDGY TOFFEE BARS
POTATO CHIP DIP
BANANA CHIFFON CAKE
HARVEST STUFFED CHICKEN
PRETZEL MUSTARD DIP
BANANA CREAM ECLAIRS
HEARTY TUNA BAKE
RASPBERRY PLUM JAM
BASIL TOMATO TUNA NOODLES
HOMESTYLE FRIED CHICKEN
ROAST CHRISTMAS GOOSE
BEETS IN ORANGE SAUCE
LAYERED BANANA PUDDING
ROCKY MOUNTAIN BRISKET
BROWN SUGAR SHORTBREAD
LEMON HERB LAMB CHOPS
SPICED PORK POTPIE
BROWNIE MALLOW BARS
NEW HAVEN CLAM PIZZA
SUNNY SPONGE CAKE
BUTTERSCOTCH HARD CANDY
NO BAKE LIME CHEESECAKE
SURPRISE MONKEY BREAD
CHEESY SAUSAGE NACHOS
OATY PEANUT BUTTER COOKIE
SWEET AND SOUR HAM BALLS
CHEESY TUNA NOODLES
ORANGE JELLY CANDIES
SWEETENED WHIPPED CREAM
CINNAMON MOCHA COFFEE
ORANGE YOGURT MUFFINS
TART LEMON SORBET
CLASSIC YELLOW CUPCAKES
PEAR GINGERBREAD CAKE
TEXMEX CORNBREAD
CRAB SALAD SURPREME
PECAN PARMESAN CHICKEN
WHITE BEAN TUNA SALAD
CRACKED CHEDDAR MUFFINS
PEPPERY BLACK BEAN SALSA

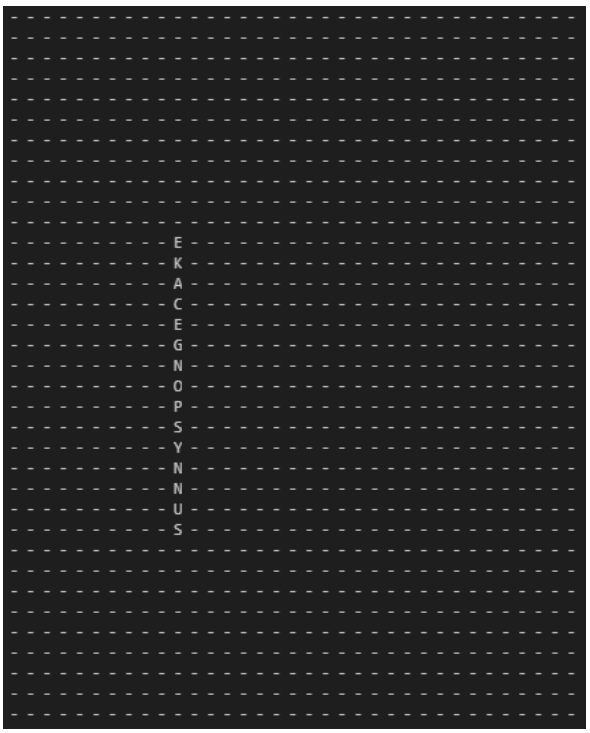
2. Output program
   Karena keterbatasan ruang pada terminal, hanya beberapa kata saja yang dapat dilihat hasil outputnya.

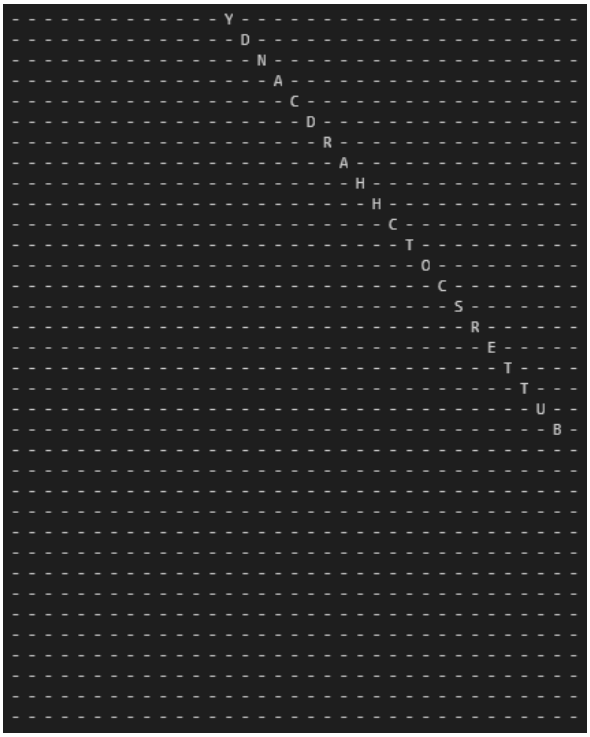


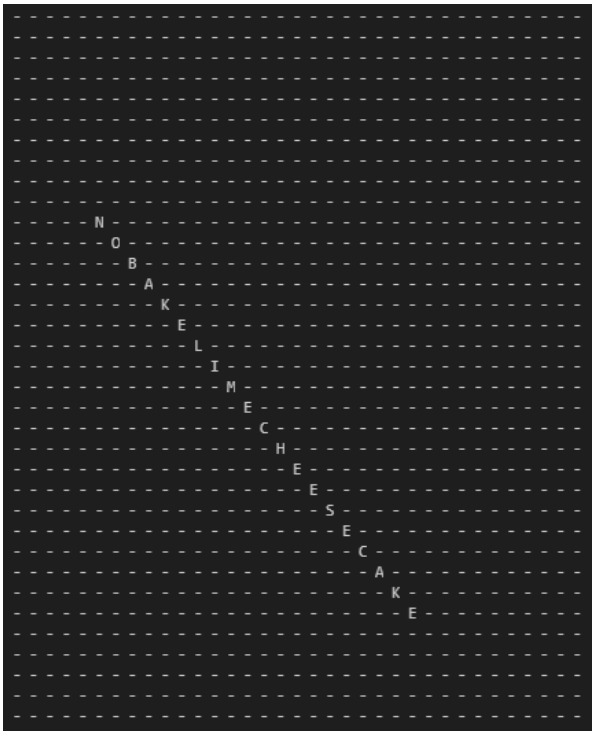

EIPTOPKROPDECIPS

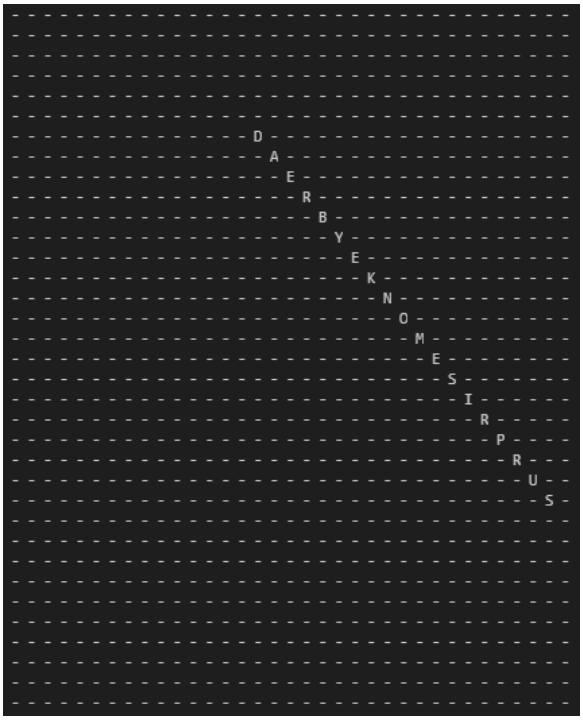

SRABWOLLAMEINWORB

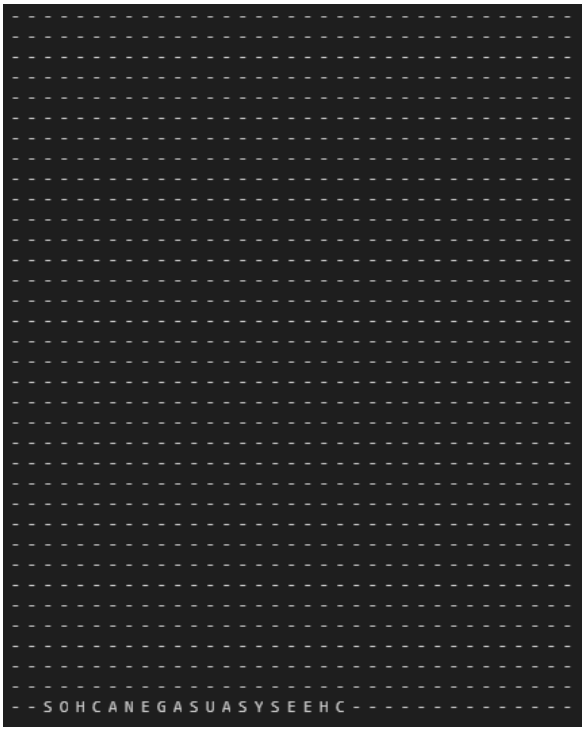

NEWHAVENCLAMPIZZA

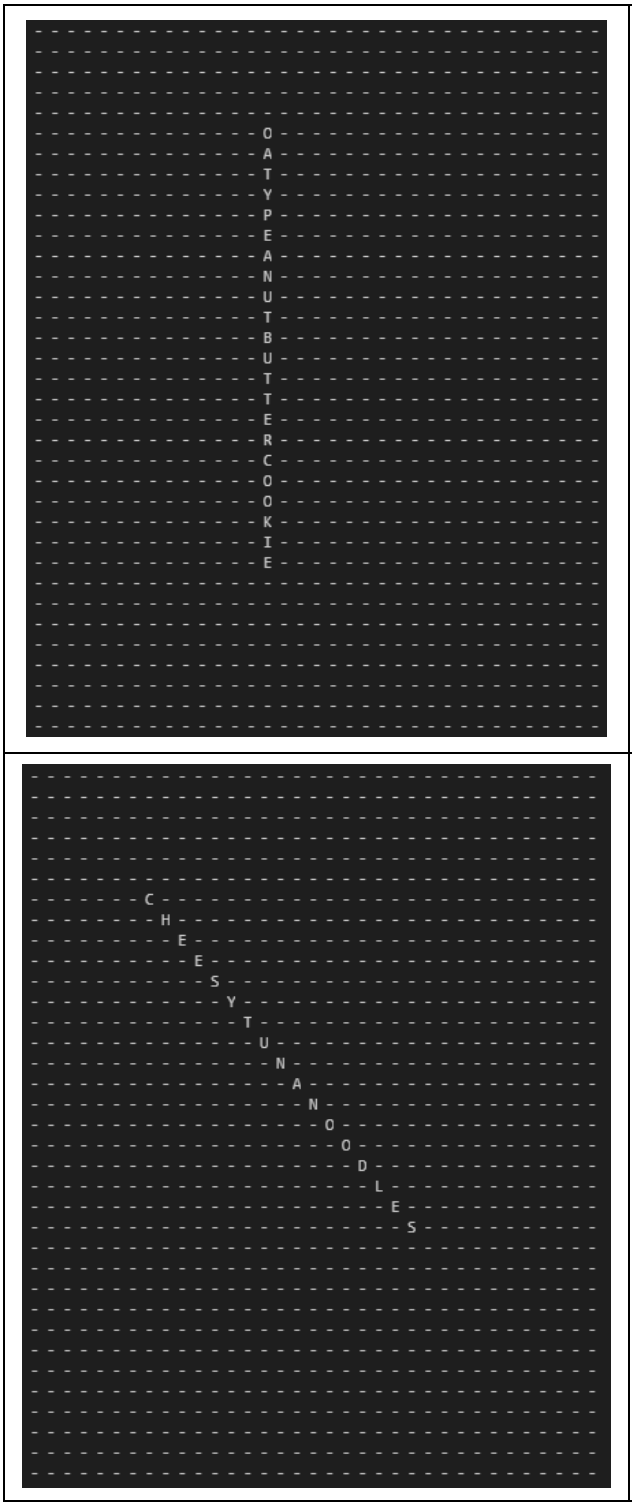

EKACEGNOPSYNNUS

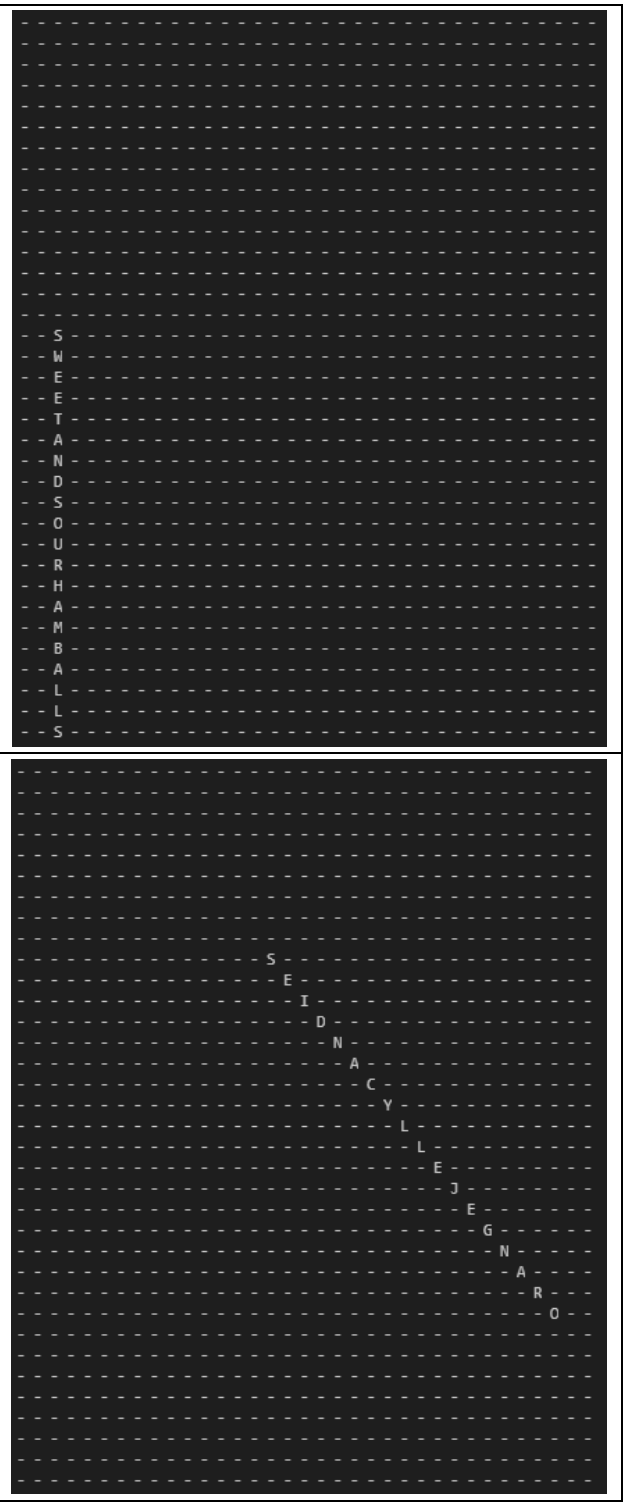






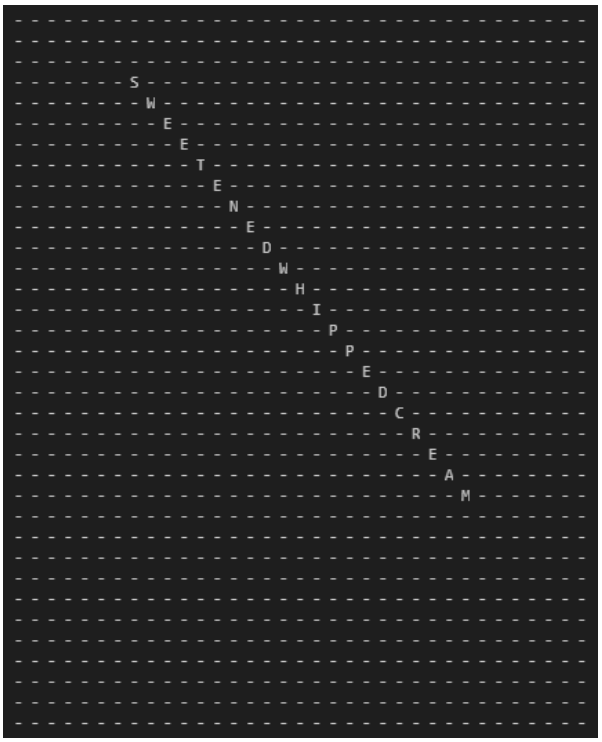

SOHCANEGASUASYSEEHC

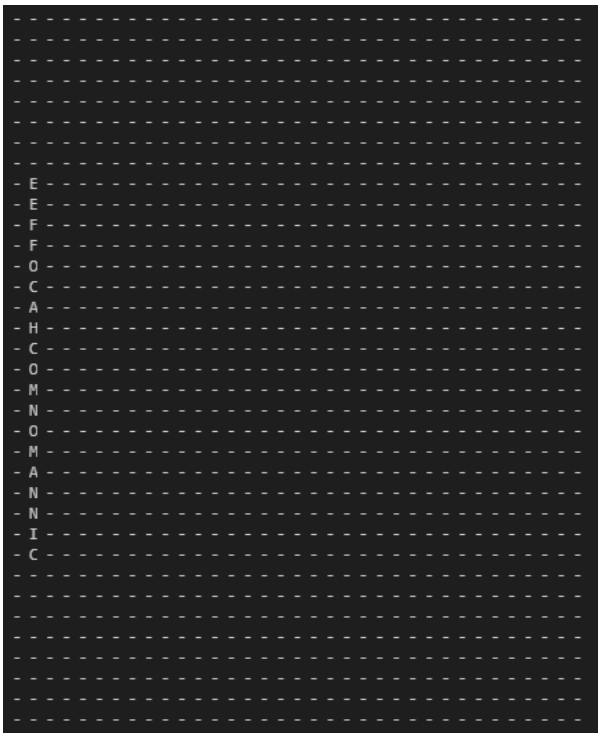

OATYPEANUTBUTTERCOOKIE

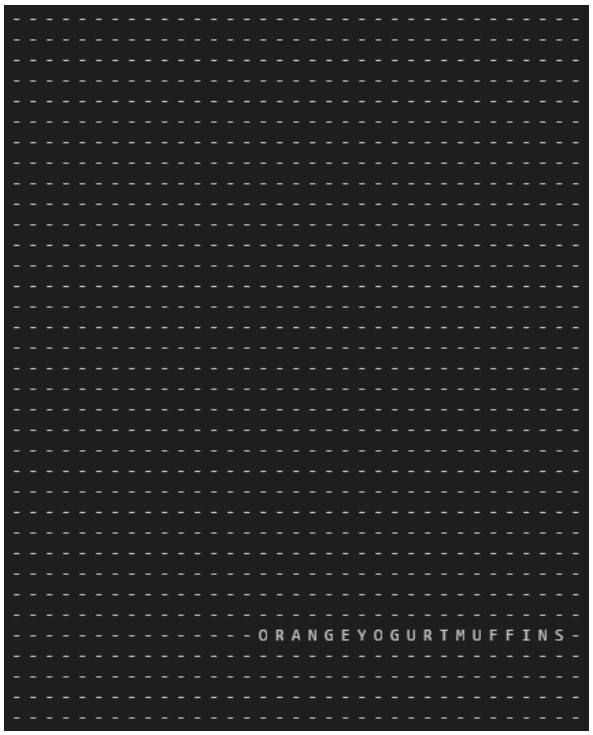

SWEETANDSOURHAMBALLS

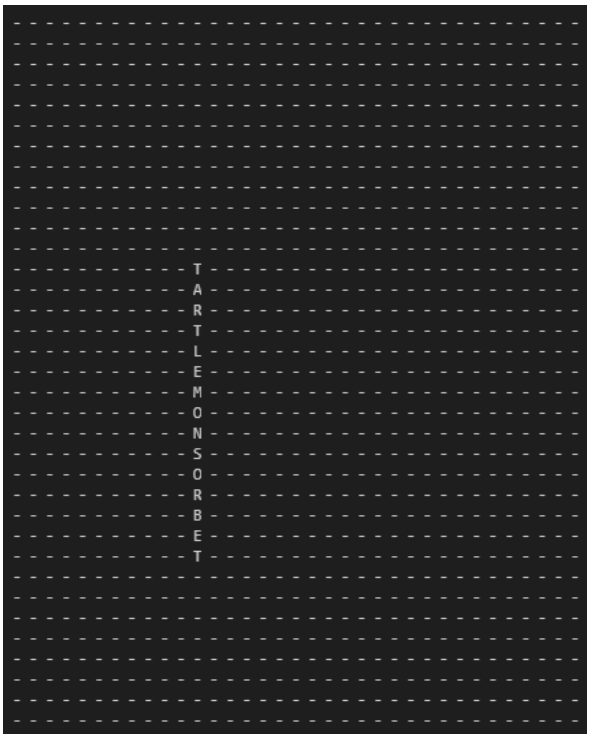
CHEESY
TUNA
NOODLES

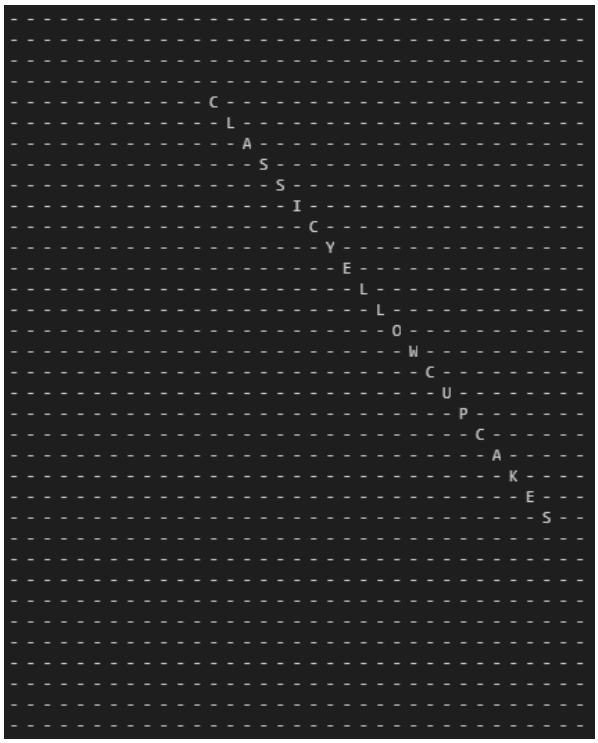
SEIDNACYLLEJEGNARO

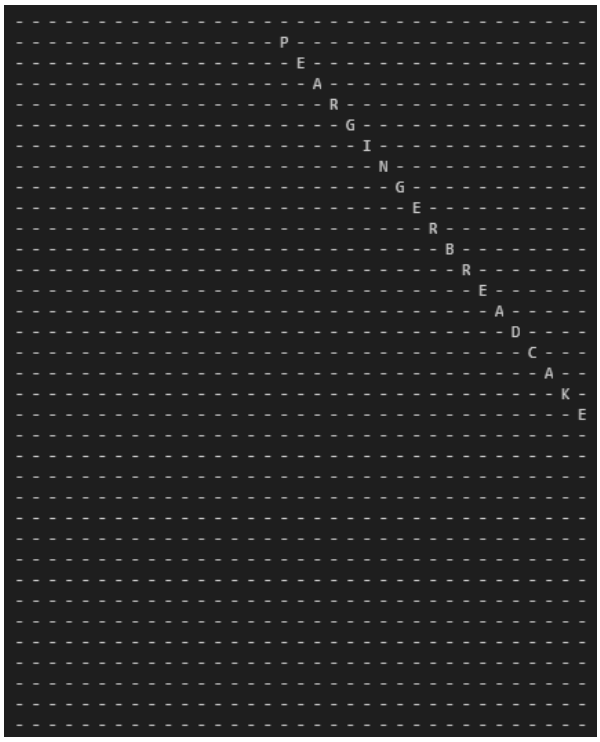

SWEETENDERPPEDCREAM

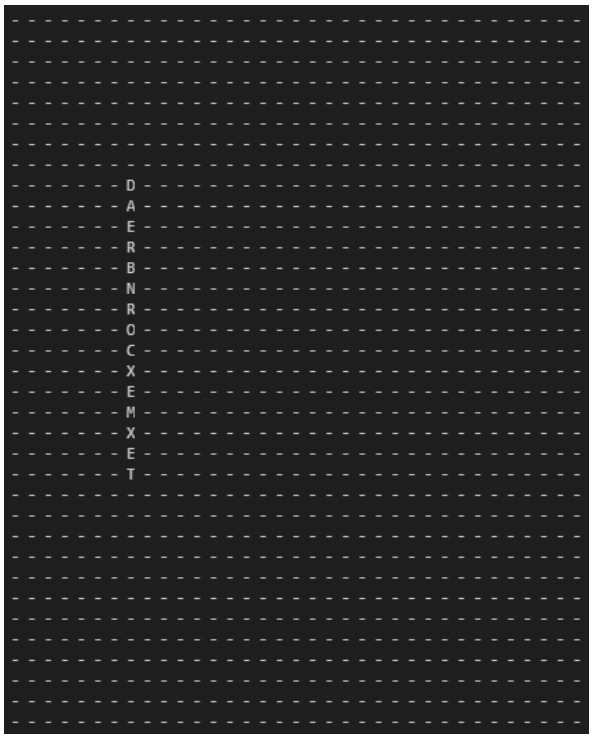

EEFFOCAHCOMNOMANNIC

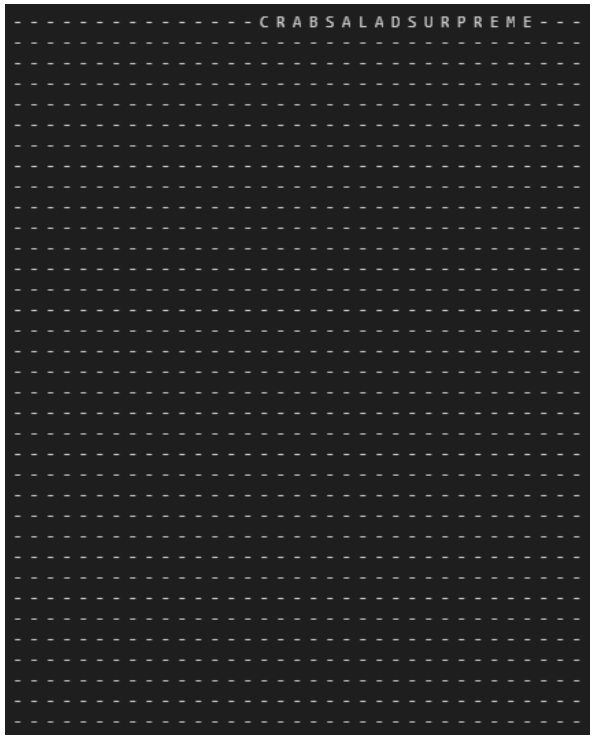

ORANGEYOGURTMUFFINS

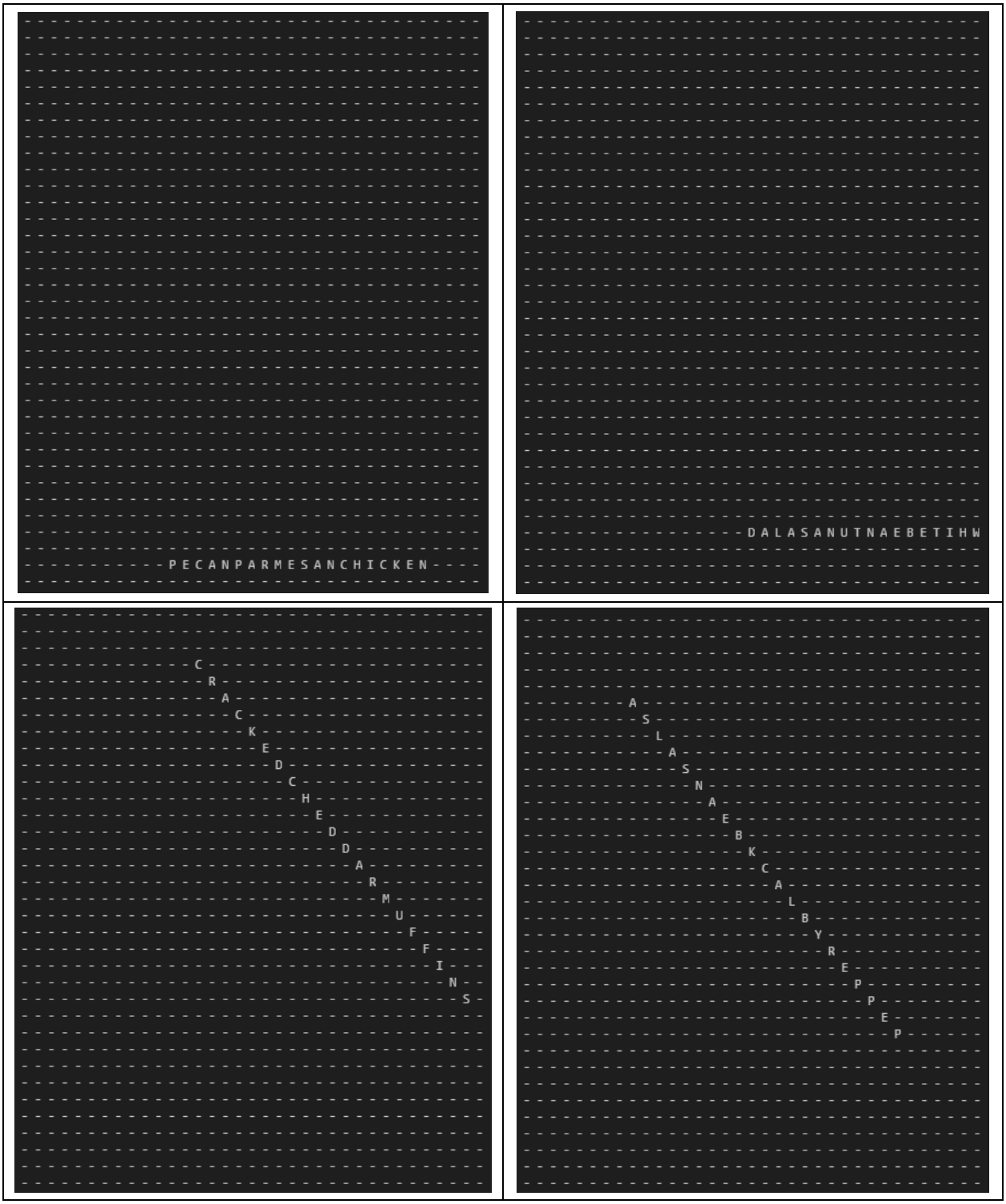

TARTLEMONSORBET


CLASSICYELLOWCUPCAKES


PEARGINGERBREADCAKE


DAERBNROCXEMXET


CRABSALADSURPREME




```
Waktu eksekusi program: 1.8437500000 seconds.
Total perbandingan huruf: 43436.
```

D. Alamat Drive Kode Program

IF2211-Strategi-Algoritma/Tucil1 at main · ikmalalfaozi/IF2211-Strategi-Algoritma (github.com)