LAPORAN TUGAS KECIL
IF2211/Strategi Algoritma

Word Search Puzzle Solver

Dipersiapkan oleh:

Vionie Novencia Thanggestyo

13520006

K3

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

1. Algoritma Brute Force

Mula-mula program akan meminta masukan nama file puzzle, kemudian puzzle dan kata yang ingin di cari pada file tersebut akan di transformasikan ke dalam bentuk matrix. Program kemudian akan mencari kata-kata tersebut satu per satu dengan algoritma pencarian.

Pada program *Word Search Puzzle Solver*, terdapat 8 jenis pencarian. Saya menamainya sesuai dengan arah mata angin, yaitu

- SearchWtoE,pencarian dari kiri ke kanan
- SearchEtoW,pencarian dari kanan ke kiri
- SearchNtoS, pencarian dari atas ke bawah
- SearchStoN, pencarian dari bawah ke atas
- SearchNEtoSW, pencarian dari kanan atas ke kiri bawah secara diagonal
- SearchSWtoNE, pencarian dari kiri bawah ke kanan atas secara diagonal
- SearchSEtoNW, pencarian dari kanan bawah ke kiri atas secara diagonal
- SearchNWtoSE, pencarian dari kiri atas ke kanan bawah secara diagonal

Pertama-tama program akan melakukan pencarian dari kiri ke kanan. Program akan membandingkan huruf pertama pada kata yang ingin dicari dengan huruf paling kiri atas pada matrix. Apabila huruf pertama pada kata dengan huruf pada matrix puzzle sama, maka akan dilanjutkan perbandingan huruf kedua pada kata dan huruf pada kanan matrix puzzle, dan seterusnya. Apabila tidak sama, puzzle akan mengulang membandingkan dari huruf pertama kata yang ingin dicari. Apabila program telah menemukan kata yang ingin dicari, program akan berhenti membandingkan dan mencetak solusi untuk kata pertama. Jika tidak ditemukan kata yang ingin dicari pada teknik pencarian pertama, program akan melanjutkan pencarian kata ke teknik pencarian dari kanan ke kiri, dan begitu seterusnya.

2. *Source program*

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define boolean unsigned char
#define true 1
#define false 0
#define Maxrow 100
#define Maxcol 100
typedef char ElType;
typedef struct
{
   ElType contents[Maxrow][Maxcol];
   int rowEff;
   int colEff;
} Matrix;
#define ELMT(M, i, j)
(M).contents[(i)][(j)]

//Mentransformasikan puzzle pada file
ke bentuk matrix
void loadPuzzle(int *rows, int *cols,
int *nword, Matrix *puzzle, Matrix
*words){
    FILE *fptr;
    char now,prev;
    int row = *rows, col = *cols,
count= *nword;
    char* file;
    char path[] = "../test/";
    char fname[50]="";
    printf("Please enter puzzle file
name: ");
    scanf("%s",&fname);
    strcat(path,fname);

    fptr = fopen(path,"r");
    now = fgetc(fptr);

    while (now!= EOF){
        if(prev != now){
            if (now == '\n'){
                row++;
                col = 0;
                *rows = row;
            }
            else{
```

```
                if (now != ' ' ){
                    ELMT(*puzzle,row,co
l) = now;
                    col++;
                    if (*cols< col)
                        *cols = col;
                }
            }
            prev = now;
            now = fgetc(fptr);
        }
        else{
            int row = 0, col=0;
            now = fgetc(fptr);
            while(now!= EOF){
                if(now != '\n'){
                   ELMT(*words,row,col)
= now;
                    col++;
                    now = fgetc(fptr);
                }
                else{
                    ELMT(*words,row,col
) = '.';
                    row++;
                    col = 0;
                    *nword = row;
                    now = fgetc(fptr);
                }
            }
            ELMT(*words,row,col) = '.';
        }
    }
}

//Menghasilkan matrix berukuran Row  x
Col dengan isi '_'
void createSolvePuzzle(int rows, int
cols, Matrix *puzsol){
    for (int i=0; i<rows; i++){
        for(int j=0; j<cols*2; j+=2){
            ELMT(*puzsol,i,j) = '-';
            ELMT(*puzsol,i,j+1) = ' ';
        }
    }
}

//Pencarian dari kiri ke kanan
boolean searchWtoE(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q=0,passp=0, passw=0,
hitung=0;
    while (p < rows && !found &&
ELMT(words,i,passw)!='.'){
        while(q< cols && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,p,passp) ==
ELMT(words,i,passw)){
                passp++;
                passw++;
            }
            else{
                passw = 0;
                q++;
                passp = q;
            }
            now = ELMT(words,i,passw);
        }
        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p++;
            q=0;
            passp=0;
        }
    }
    *count = hitung;
    return found;
}
```

```
//pencarian dari kanan ke kiri
boolean searchEtoW(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q= cols-1,passp=cols-1,
passw=0, hitung=0;
    while (p < rows && !found &&
ELMT(words,i,passw)!='.'){
        while(q>=0 && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,p,passp) ==
ELMT(words,i,passw)){
                passp--;
                passw++;
            }
            else{
                passw = 0;
                q--;
                passp = q;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p++;
            q=cols-1;
            passp=q;
        }
    }
    *count = hitung;
    return found;
}

//pencarian dari atas ke bawah
```

```
boolean searchNtoS(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q=0,passp=0,
passw=0,hitung=0;
    while (p < cols && !found &&
ELMT(words,i,passw)!='.'){
        while(q< rows && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passp,p) ==
ELMT(words,i,passw)){
                passp++;
                passw++;
            }
            else{
                passw = 0;
                q++;
                passp = q;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = q;
            *col = p;
        }
        else{
            passw = 0;
            p++;
            q=0;
            passp=0;
        }
    }
    *count=hitung;
    return found;
}

//pencarian dari bawah ke atas
```

```
boolean searchStoN(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q=rows-1,passp=rows-1,
passw=0, hitung=0;
    while (p < cols && !found &&
ELMT(words,i,passw)!='.'){
        while(q>=0 && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passp,p) ==
ELMT(words,i,passw)){
                passp--;
                passw++;
            }
            else{
                passw = 0;
                q--;
                passp = q;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = q;
            *col = p;
        }
        else{
            passw = 0;
            p++;
            q=rows-1;
            passp=q;
        }
    }
    *count=hitung;
    return found;
}

//pencarian dari kanan bawah ke kiri
atas secara diagonal
```

```
boolean searchSEtoNW(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = rows-1, q=cols-
1,passh=q,passv=p,passw=0,hitung=0;
    while (p > 0 && !found &&
ELMT(words,i,passw)!='.'){
        while(q >=0 && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passv,passh)
== ELMT(words,i,passw)){
                passh--;
                passv--;
                passw++;
            }
            else{
                passw = 0;
                q--;
                passh = q;
                passv = p;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p--;
            q=cols-1;
            passv=p;
            passh=q;
        }
    }
    *count=hitung;
    return found;
}
```

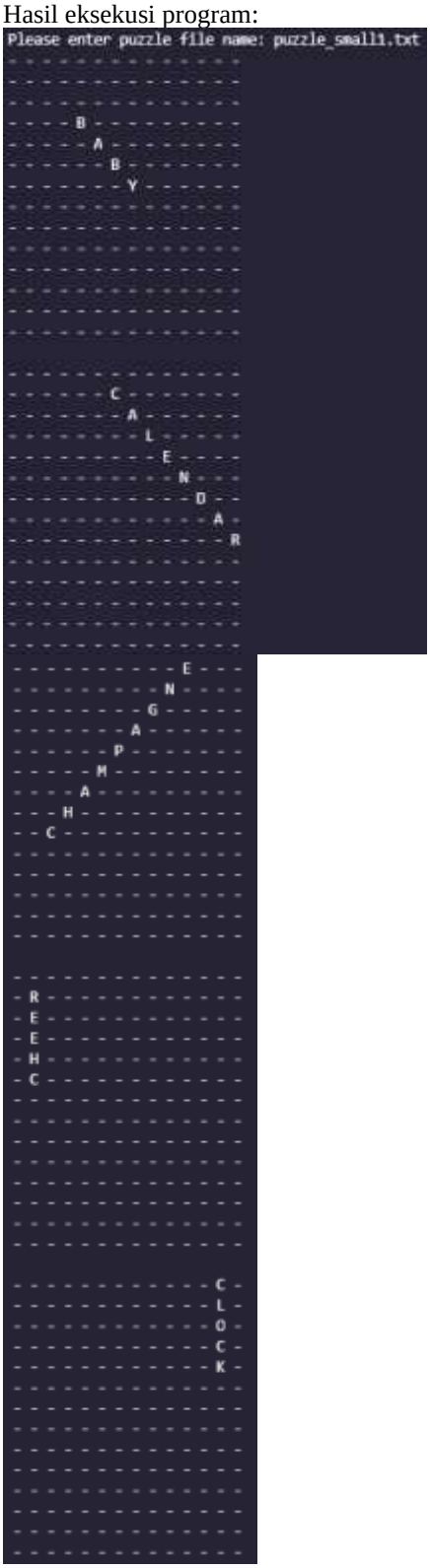
```
//pencarian dari kiri atas ke kanan bawah secara diagonal
boolean searchNWtoSE(Matrix puzzle, Matrix words, int i, int *row, int *col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q=0,passh=0,passv=0,passw=0,hitung=0;
    while (p < rows && !found && ELMT(words,i,passw)!='.'){
        while(q< cols && !found && ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passv,passh) == ELMT(words,i,passw)){
                passh++;
                passv++;
                passw++;
            }
            else{
                passw = 0;
                q++;
                passh = q;
                passv = p;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p++;
            q=0;
            passv=p;
            passh=q;
        }
    }
    *count = hitung;
    return found;
}

//pencarian dari kiri bawah ke kanan atas secara diagonal
boolean searchSWtoNE(Matrix puzzle, Matrix words, int i, int *row, int *col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = rows-1, q=0,passh=0,passv=p,passw=0,hitung=0;
    while (p >=0 && !found && ELMT(words,i,passw)!='.'){
        while(q< cols && !found && ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passv,passh) == ELMT(words,i,passw)){
                passh++;
                passv--;
                passw++;
            }
            else{
                passw = 0;
                q++;
                passh = q;
                passv = p;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p--;
            q=0;
            passv=p;
```

```
            passh=q;
        }
    }
    *count=hitung;
    return found;
}

//pencarian dari kiri atas ke kanan
bawah secara diagonal
boolean searchNEtoSW(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int *count){
    boolean found = false;
    char now;
    int rows = *row, cols = *col;
    int p = 0, q=cols-
1,passh=q,passv=p,passw=0,hitung=0;
    while (p < rows && !found &&
ELMT(words,i,passw)!='.'){
        while(q>=0 && !found &&
ELMT(words,i,passw)!='.'){
            hitung++;
            if(ELMT(puzzle,passv,passh)
== ELMT(words,i,passw)){
                passh--;
                passv++;
                passw++;
            }
            else{
                passw = 0;
                q--;
                passh = q;
                passv = p;
            }
            now = ELMT(words,i,passw);
        }

        if (now == '.'){
            *len = passw;
            found = true;
            *row = p;
            *col = q;
        }
        else{
            passw = 0;
            p++;
            q=cols-1;
            passv=p;
            passh=q;
        }
    }
    *count=hitung;
    return found;
}

//mengisi matrix solusi
void result(Matrix *puzsol, Matrix
puzzle, int rows, int cols,int len, int
option){
    if(option>=1 && option <=4){
        for (int i=0; i<len; i++){
            ELMT(*puzsol,rows,cols*2) =
ELMT(puzzle,rows,cols);
            if (option == 1){
                cols++;
            }else if(option == 2){
                cols--;
            }else if(option == 3){
                rows++;
            }else{
                rows--;
            }
        }
    }
    else{
        for (int i=0; i<len; i++){
            ELMT(*puzsol,rows,cols*2) =
ELMT(puzzle,rows,cols);
            if (option == 6){
                cols++;
                rows++;
            }else if(option == 5){
                cols--;
                rows--;
            }else if(option == 7){
                cols++;
                rows--;
            }else{
                cols--;
                rows++;
```

```
            }
        }
    }
}

//mencetak solusi
void displayResult(Matrix puzsol,
Matrix puzzle,int ori_row,int ori_col){
    for (int i=0; i<ori_row; i++){
        for(int j=0; j<ori_col*2;j++){
            printf("%c",ELMT(puzsol,i,j
));
        }
        printf("\n");
    }
}

boolean searchWord(Matrix puzzle,
Matrix words, int i, int *row, int
*col, int *len, int search_option, int
*count){
    boolean found = false;
    if (search_option == 1){found =
searchWtoE(puzzle,words,i,row,col,len,
count);}
    else if (search_option == 2){found
=
searchEtoW(puzzle,words,i,row,col,len,
count);}
    else if (search_option == 3){found
=
searchNtoS(puzzle,words,i,row,col,len,c
ount);}
    else if (search_option == 4){found
=
searchStoN(puzzle,words,i,row,col,len,c
ount);}
    else if (search_option == 5){found
=
searchSEtoNW(puzzle,words,i,row,col,len
,count);}
    else if (search_option == 6){found
=
searchNWtoSE(puzzle,words,i,row,col,len
,count);}
    else if (search_option == 7){found
=
searchSWtoNE(puzzle,words,i,row,col,len
,count);}
    else if (search_option == 8){found
=
searchNEtoSW(puzzle,words,i,row,col,len
,count);}
    return found;
}

void execute(Matrix puzzle, Matrix
puzsol, Matrix words, int Row, int Col,
int nword, int len, int *count){
    boolean found;
    int rows, cols, sum=0,counts;
    for (int i=0; i<=nword;i++){
        rows = Row; cols = Col;
        createSolvePuzzle(Row, Col,
&puzsol);
        int j=1;
        found = false;
        while(j<=9 && !found){
            found =
searchWord(puzzle,words,i,&rows,&cols,&
len,j,count);
            sum = sum + *count;
            if (found) {
                result(&puzsol,puzzle,r
ows,cols,len,j);
                displayResult(puzsol,pu
zzle,Row,Col);
                printf("\n");
            }
            j++;
        }
    }
    *count = sum;
}

int main(){
    Matrix puzzle, puzsol, words;
    int rows=0, cols=0, nword=0, len,
Row, Col,count;
```

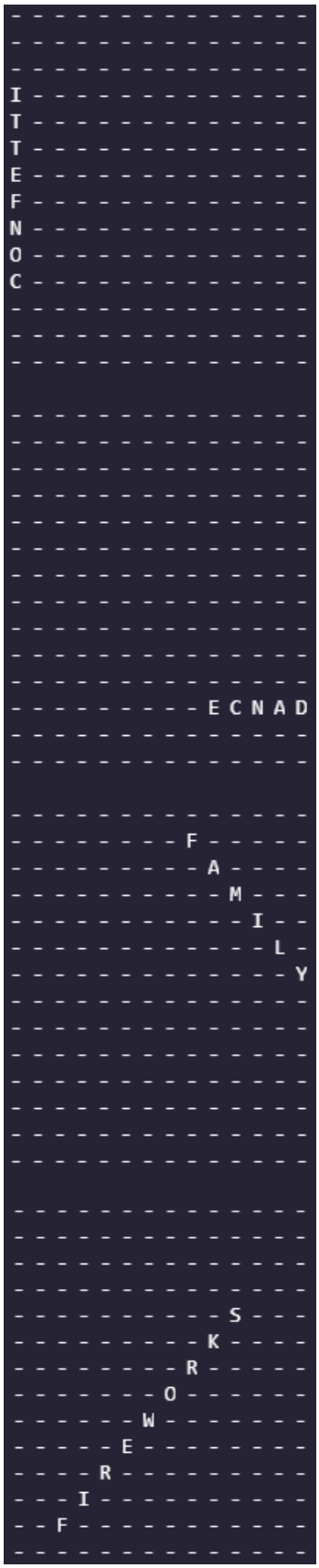
```
    loadPuzzle(&rows, &cols, &nword,
&puzzle, &words);
    Row = rows;
    Col = cols;

    clock_t t;
    t = clock();
    execute(puzzle,
puzsol,words,Row,Col,nword,len,&count);
    t = clock() - t;
    double time_taken =
((double)t)/CLOCKS_PER_SEC;
    printf("%d times comparison \n",
count);
    printf("Execution time: %f
seconds", time_taken);
    return 0;
}
```

Hasil eksekusi program:

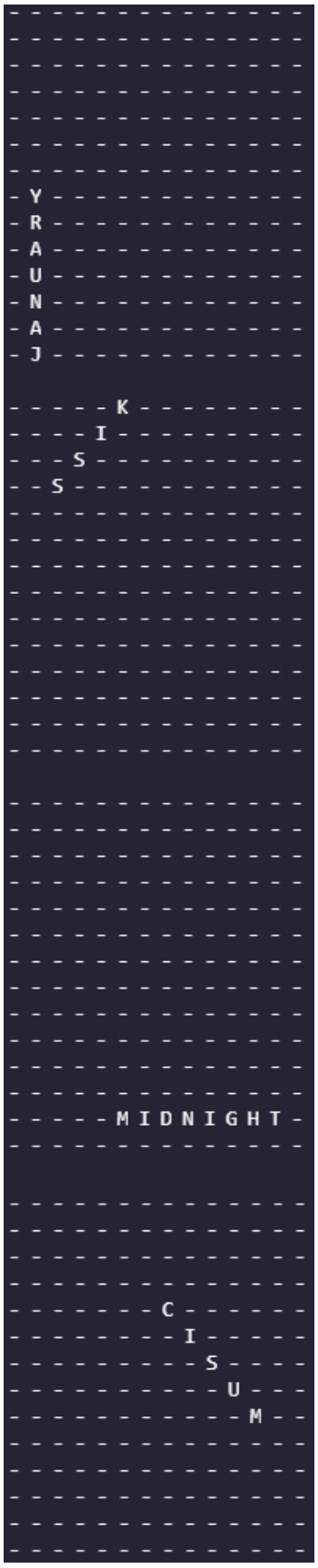


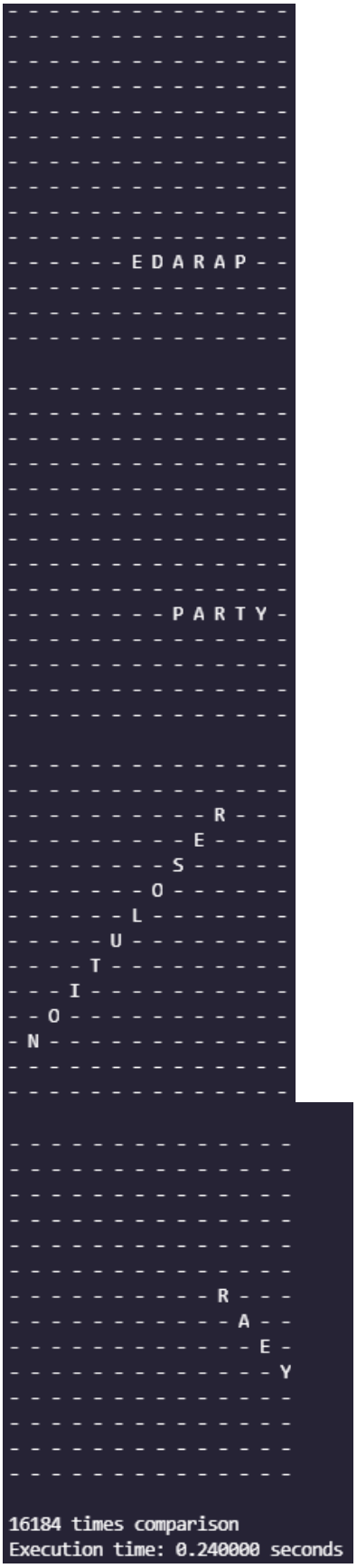
3. Screenshoot hasil eksekusi program
   a. Puzzle small 1

Word Search Puzzle:


ITTEFNOC
ECNAD
FAMILY
SKROWERIF


FIRST
LLABTOOF
YPPAH
HAT


YRAUNAJ
K
I
S
S
MIDNIGHT
C
I
S
U
M


EDARAP
PARTY
R
E
S
O
L
U
T
I
O
N
R
A
E
Y
16184 times comparison
Execution time: 0.240000 seconds

b. Puzzle small 2

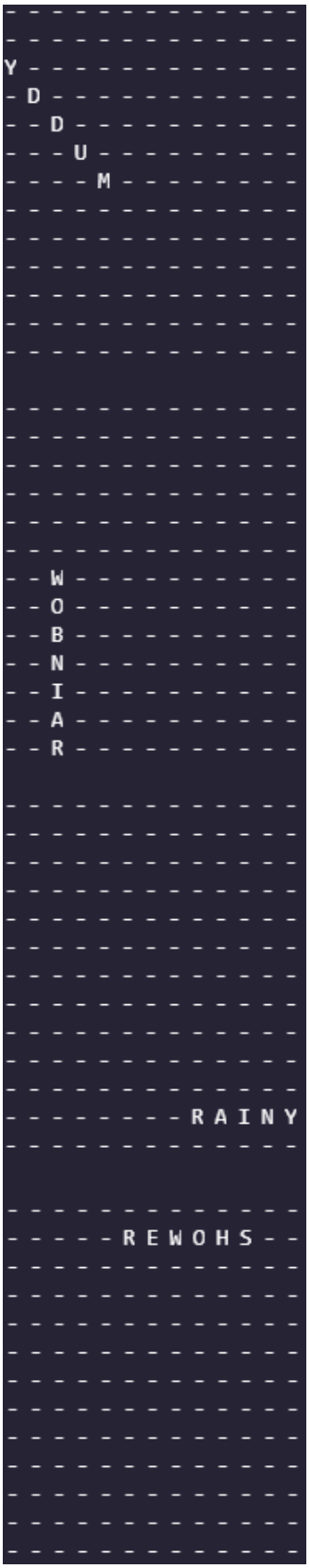
Word Search Puzzle:

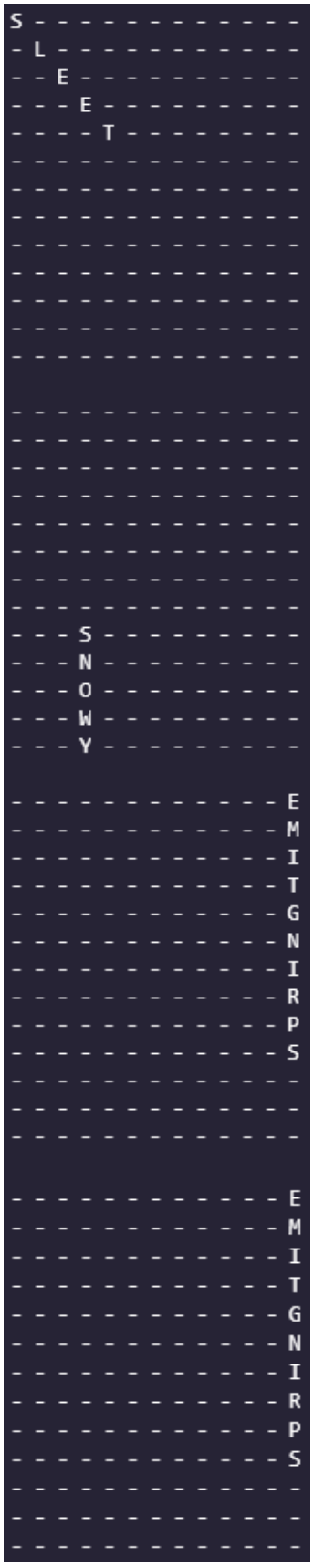
```
S H C R A M W I N D Y M E
A L R C H R E W O H S M M
Y I E S S Y A F I D T R I
O D C E L Z G T O O M A T
E I D L T E N N E G L W G
I K I U E E I A L W G G N
I H W O M R W N A G N Y I
C N O D G B A O U I S R R
O U B S F T H S T U E L P
I K N N R E T L N T A L S
A M I O E Y E N A B X M W
D M A W S M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

BREEZY
CHILLY
CLOUDY
FOGGY
FRESH
GUSTY
MARCH
MELTING
MUDDY
RAINBOW
RAINY
SHOWER
SLEET
SNOWY
SPRINGTIME
SUNNY
THAWING
WARM
WATER
WET
WINDY
```

```
Please enter puzzle file name: puzzle_small2.txt
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - Y - - - - - - -
- - - - - Z - - - - - - -
- - - - - E - - - - - - -
- - - - - E - - - - - - -
- - - - - R - - - - - - -
- - - - - B - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -

- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - Y - - - - - - -
- - - - L - - - - - - - -
- - - L - - - - - - - - -
- - I - - - - - - - - - -
- H - - - - - - - - - - -
C - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -

- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - C L O U D Y - -
```

```
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - F - - - - -
- - - - - - - - O - - - -
- - - - - - - - - G - - -
- - - - - - - - - - G - -
- - - - - - - - - - - Y -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
```


FRESH
GUSTY
HCRAM
GNITLEM


YDDUM
WOBNIAR
RAINY
REWOHS


SLEEET
SNOWY
EMITTING
SNIRPS
EMITTING
SNIRPS


SUNNY
GNIWATHT
MRAW
RETAW

```
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - T - - - - - - - - -
- - - - - E - - - - - - - -
- - - - - - W - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -

- - - - - - W I N D Y - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -

14071 times comparison
Execution time: 0.209000 seconds
```

```
AIRPORT
AUTO
CAMERA
CRUISE
DIRECTIONS
FERRY
GUIDE
HIKE
HOTEL
MAP
MOTEL
MOTORCYCLE
PASSENGER
PASSPORT
ROUTE
SHIP
TICKET
TOUR
TRAIN
VACATION
WORLD
```

c. Small Puzzle 3

Word Search Puzzle:

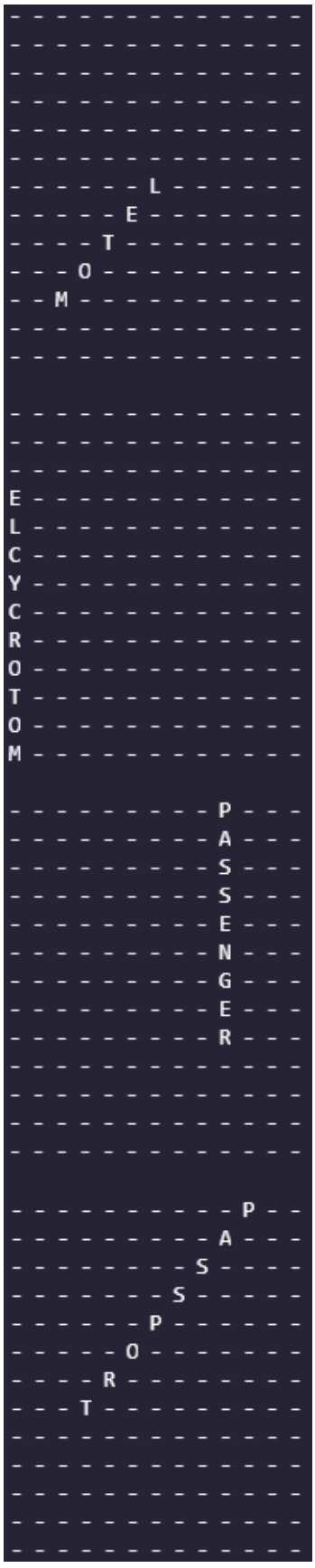
```
E N O I T A C A V P P T X
P L O R H O T E L A A I T
I T O U R O N I S S S C D
E R E E A L L S E S Y K I
L T K H E E P T S E S E R
C E I N C O U E O N F T E
Y A H T R O L D G G F T C
C P M T R E C U L E H P T
R A E E T O I R R R I H I
O M U O R D P R U H O M O
T A M N E A Y R S I S W N
O A U T O P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

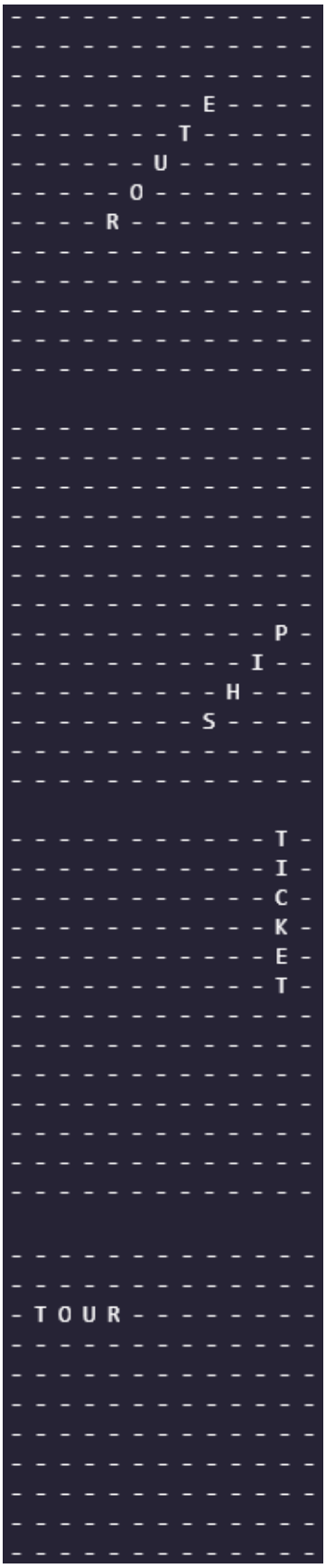
Hasil eksekusi:

```
Please enter puzzle file name: puzzle_small3.txt
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - T - - - - - - - - - -
- - - - R - - - - - - - - -
- - - - - O - - - - - - - -
- - - - - - P - - - - - - -
- - - - - - - R - - - - - -
- - - - - - - - I - - - - -
- - - - - - - - - A - - - -

- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- - - - - - - - - - - - - -
- A U T O - - - - - - - - -
- - - - - - - - - - - - - -
```


CAMERA
CRUISE
DIRECTIONS
FERRY


GUIDE
EKIH
HOTEL
PAM


MOTEL
CYCLOTOM
PASSENGER
PASSPORT


ROUTE
PHISH
TICKET
TOUR

```
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - - - - - - -
- - - - - - - D - - - - -
- - - - - - - - L - - - -
- - - - - - - - - R - - -
- - - - - - - - - - O - -
- - - - - - - - - - - W -
- - - - - - - - - - - - -
- - - - - - - - - - - - -

14481 times comparison
Execution time: 0.250000 seconds
```

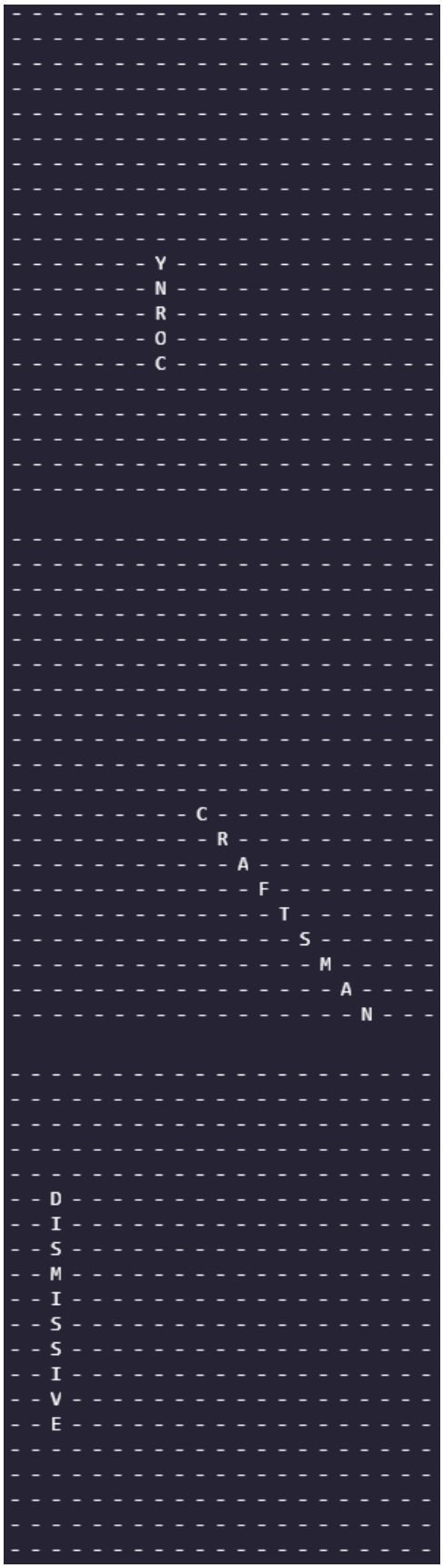
d. Puzzle medium 1

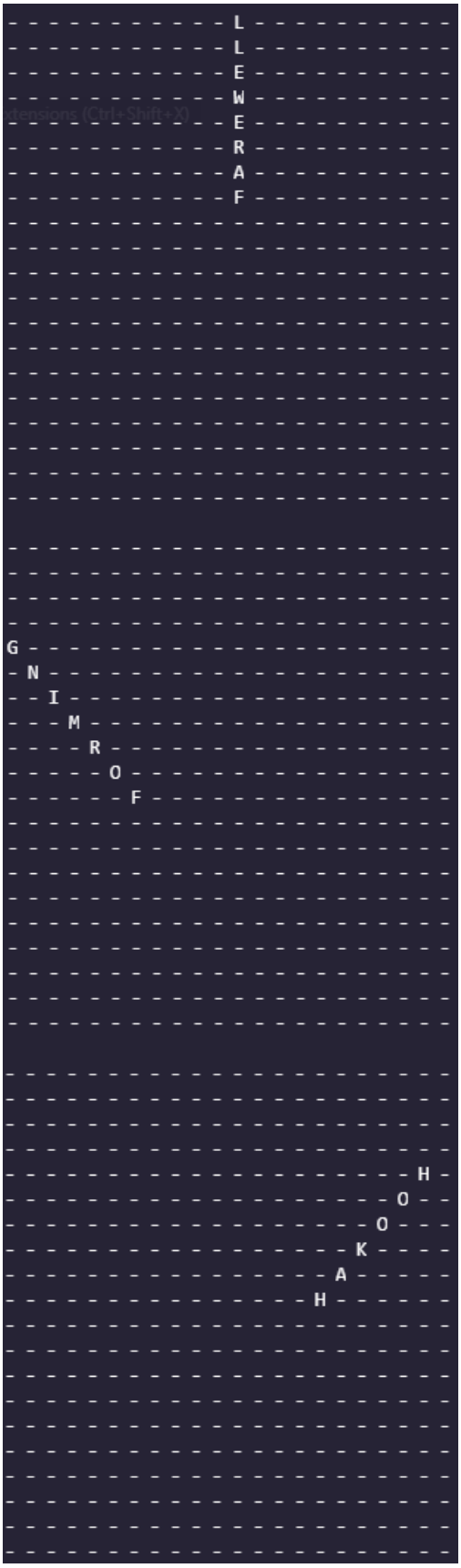
Word Search Puzzle:

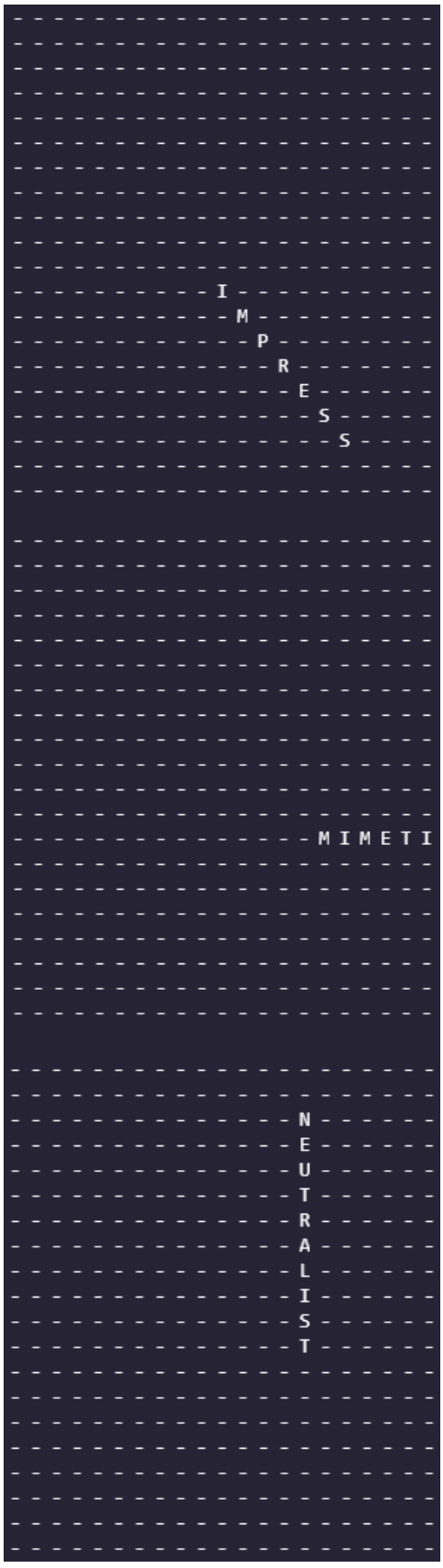
```
W R F I T K E I T O E N C Y
P R I U I R C H F N E A L R
T E R S T H A A G A R T O E
I E S V B E R A L E M Y C D
T H T A Y A P C S E S I K I
T C S T H M B O I K N E L B
E E S T A D L Y R S R D A Y
F Y Y H I U N O Y T U A A H
N R C E T Y W P E A R M E R
O A Q I N E P N P A R T Y Y
C U O H R A E D A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

BABY
CALENDAR
CHAMPAGNE
CHEER
CLOCK
CONFETTI
DANCE
FAMILY
FIREWORKS
FIRST
FOOTBALL
HAPPY
HAT
JANUARY
KISS
MIDNIGHT
MUSIC
PARADE
PARTY
RESOLUTION
YEAR
```

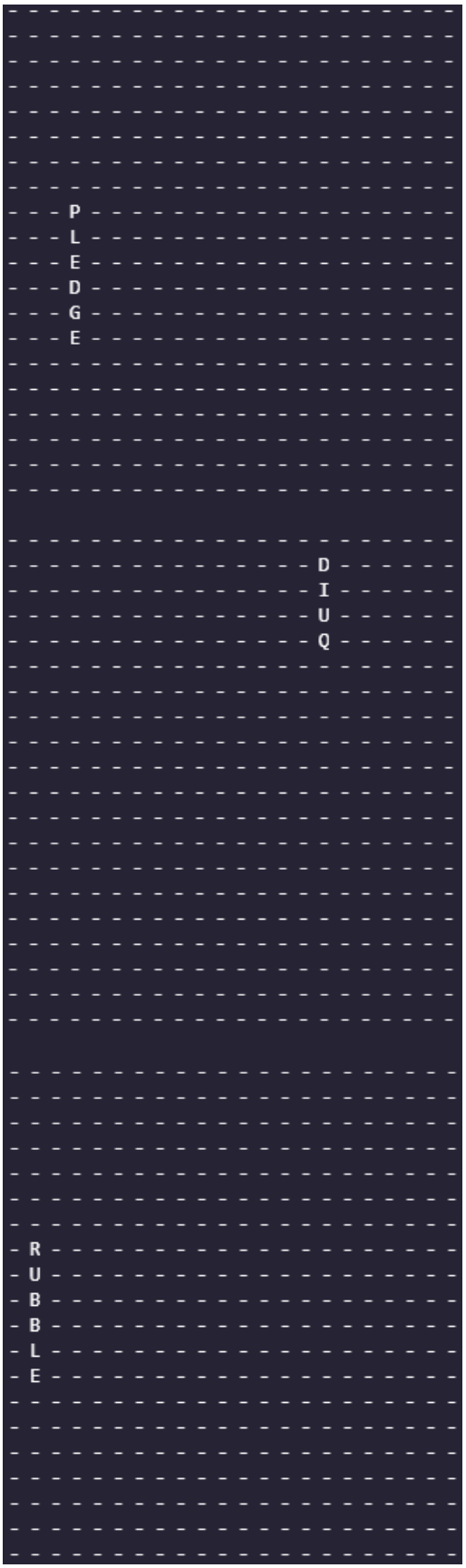
Hasil eksekusi:

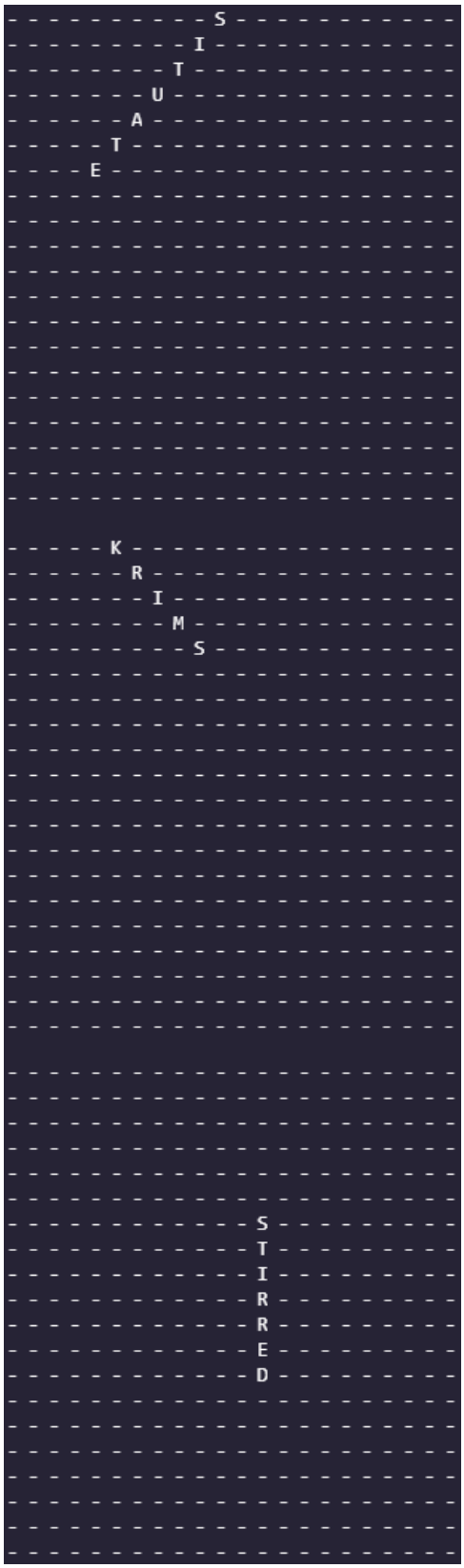
```
Please enter puzzle file name: puzzle_medium1.txt
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - G - - - - - - - - - - - - -
- - - - - - - - N - - - - - - - - - - - - -
- - - - - - - - I - - - - - - - - - - - - -
- - - - - - - - R - - - - - - - - - - - - -
- - - - - - - - I - - - - - - - - - - - - -
- - - - - - - - M - - - - - - - - - - - - -
- - - - - - - - D - - - - - - - - - - - - -
- - - - - - - - A - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - B L E S S E D - - - - -

- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - I - - - - - - - - - - - - - - - - - - -
- - L - - - - - - - - - - - - - - - - - - -
- - I - - - - - - - - - - - - - - - - - - -
- - H - - - - - - - - - - - - - - - - - - -
- - C - - - - - - - - - - - - - - - - - - -
```

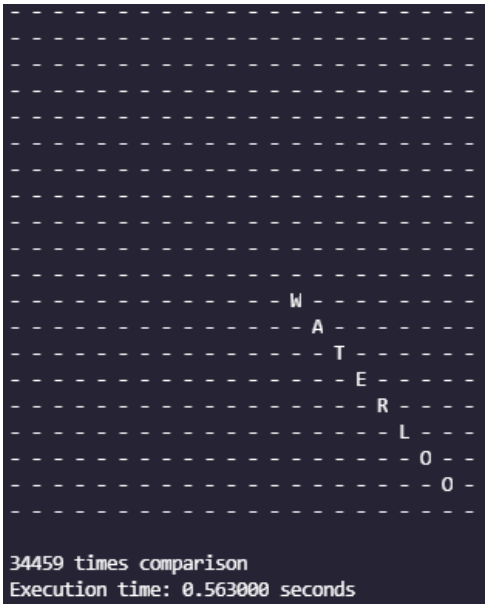

YNROC
CRAFTSMAN
DISMISSIVE


LLEWERAF
GNIMROF
HOOKAH


IMPRESS
MIMETIC
NEUTRALIST

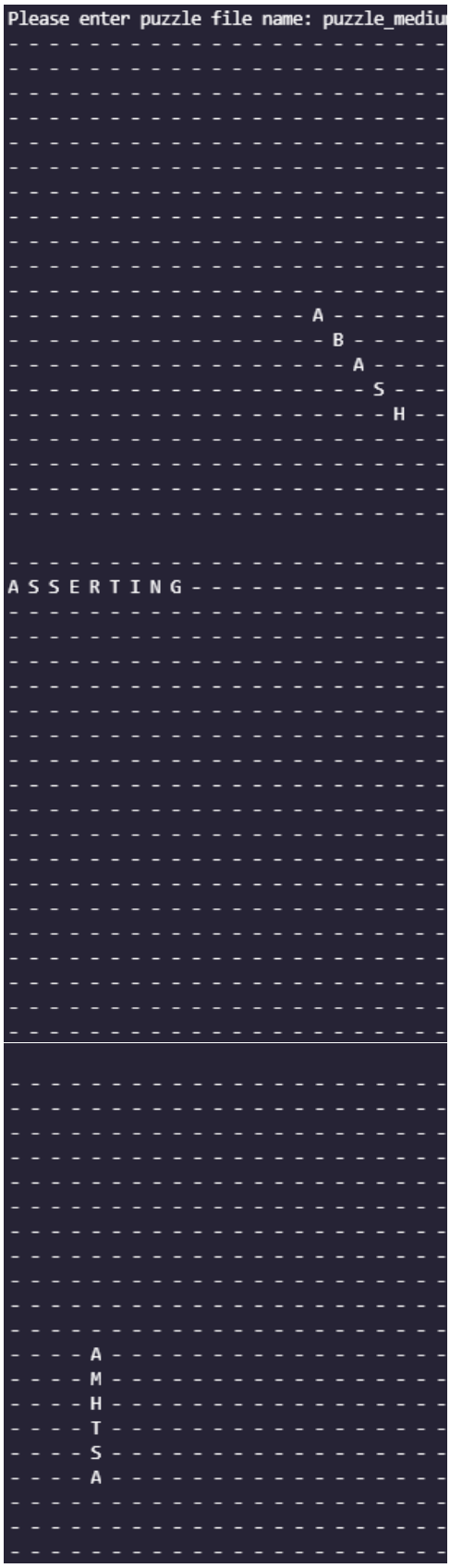

PLEDGE
DIUQ
RUBBLE

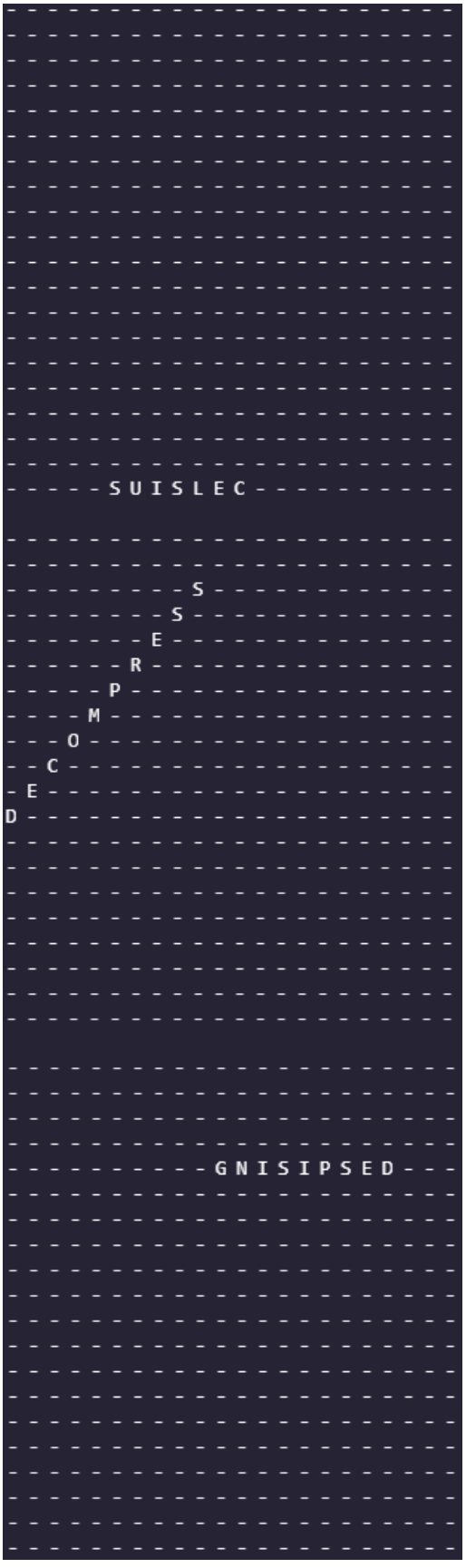


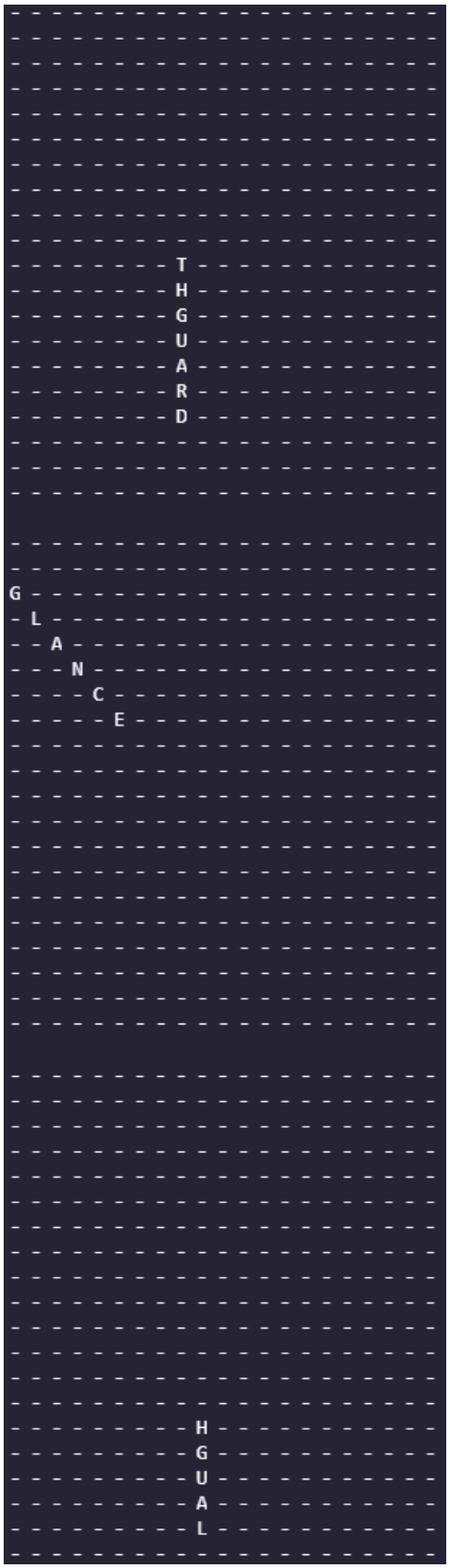


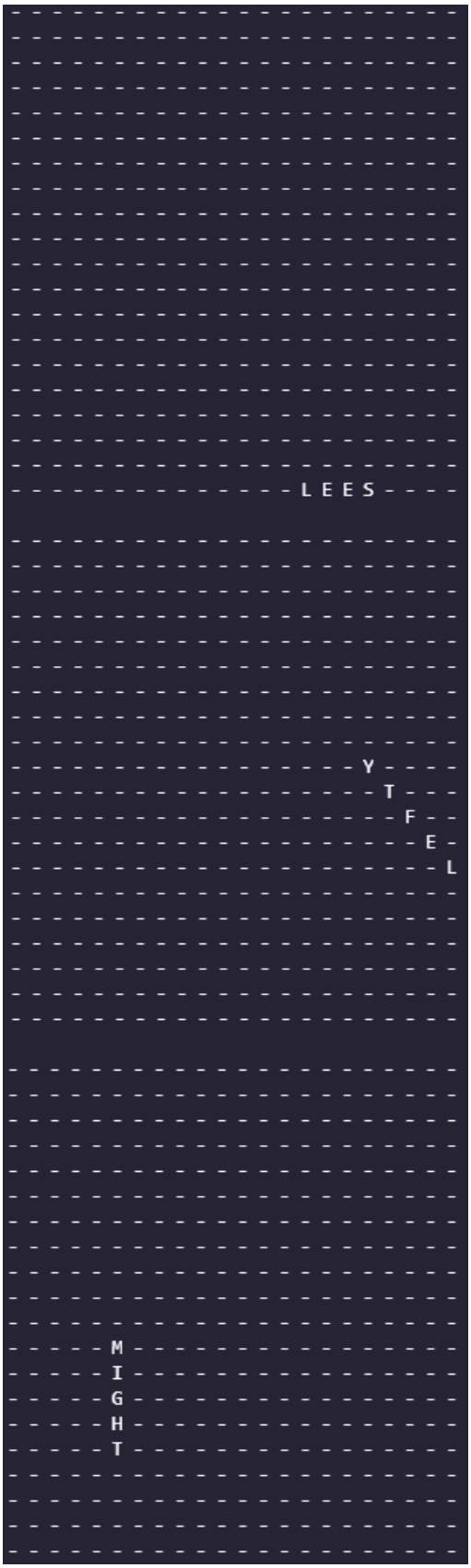
e. Puzzle Medium 2

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

ABASH
ASSERTING
ASTHMA
CELSIUS
DECOMPRESS
DESPISING
DRAUGHT
GLANCE
LAUGH
LEES
LEFTY
MIGHT
NEWSFLASH
OSCAR
PASSPORT
SLIME
SMUG
SOURCELESS
THESAURUS
WARPED
```


Please enter puzzle file name: puzzle_mediu
A
B
A
S
H
ASSERTING
A
M
H
T
S
A


SUISLEC
S
S
E
R
P
M
O
C
E
D
GNISIPSED


THGUARD
GLANCE
HGUAL


LEES
YTFEL
MIGHT

NEWSFLASH

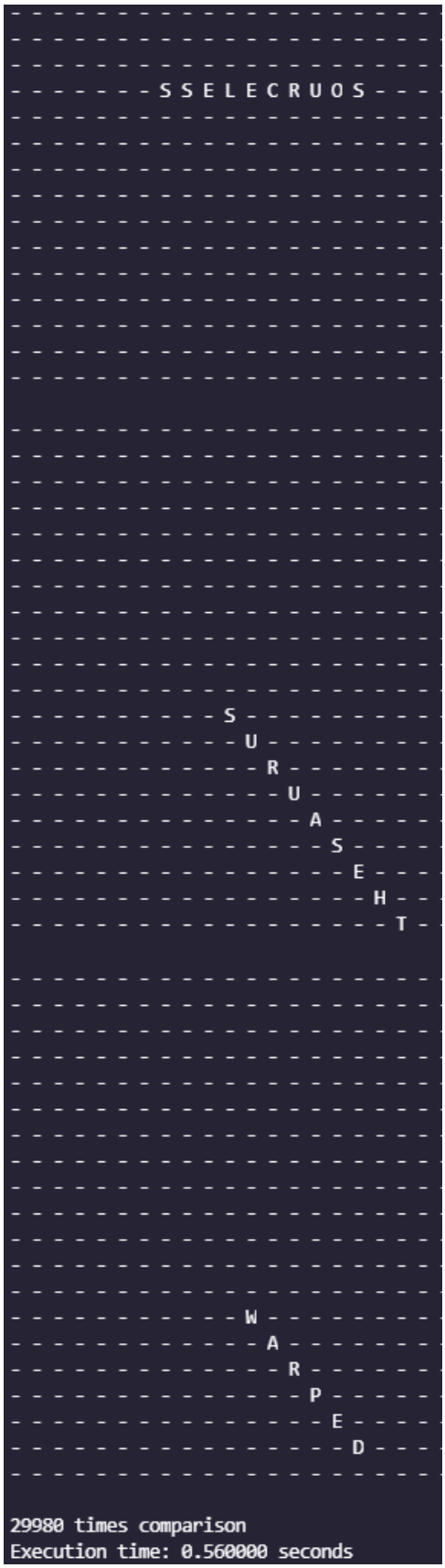
OSCAR

OSCAR

TROPSSAP

SLIME

GUMS

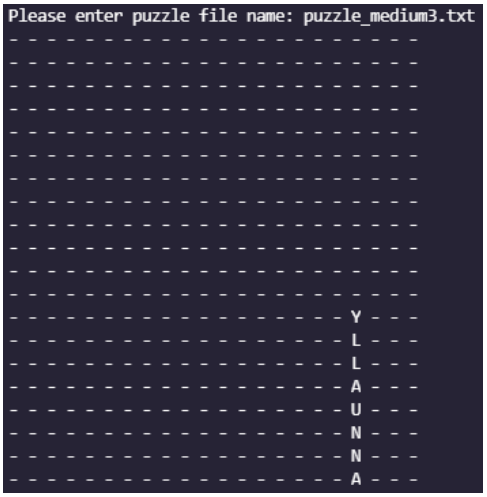


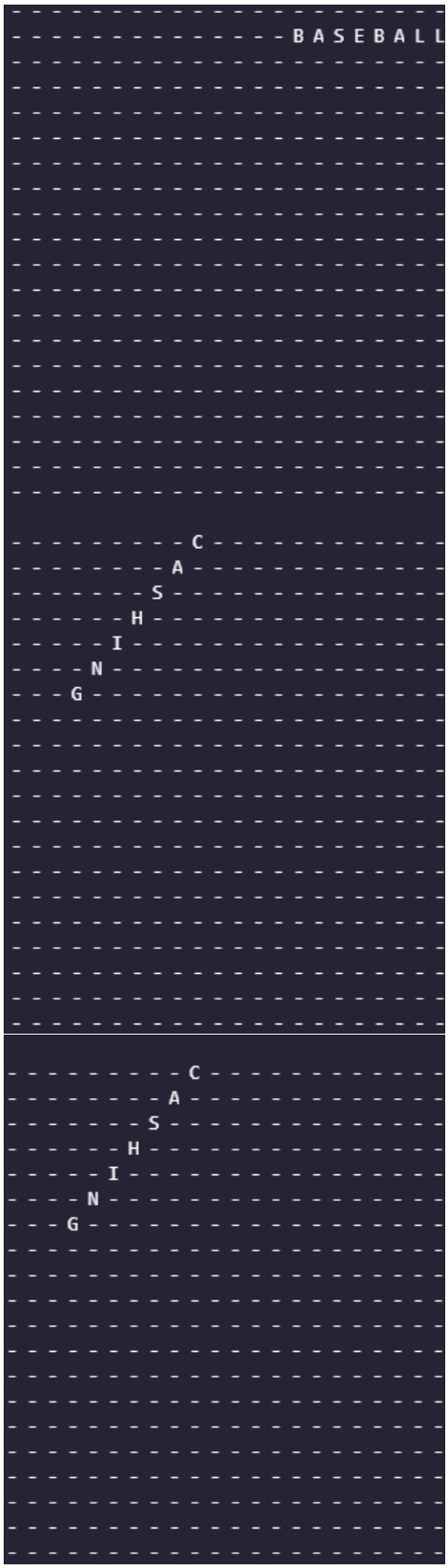
f. Puzzle medium 3

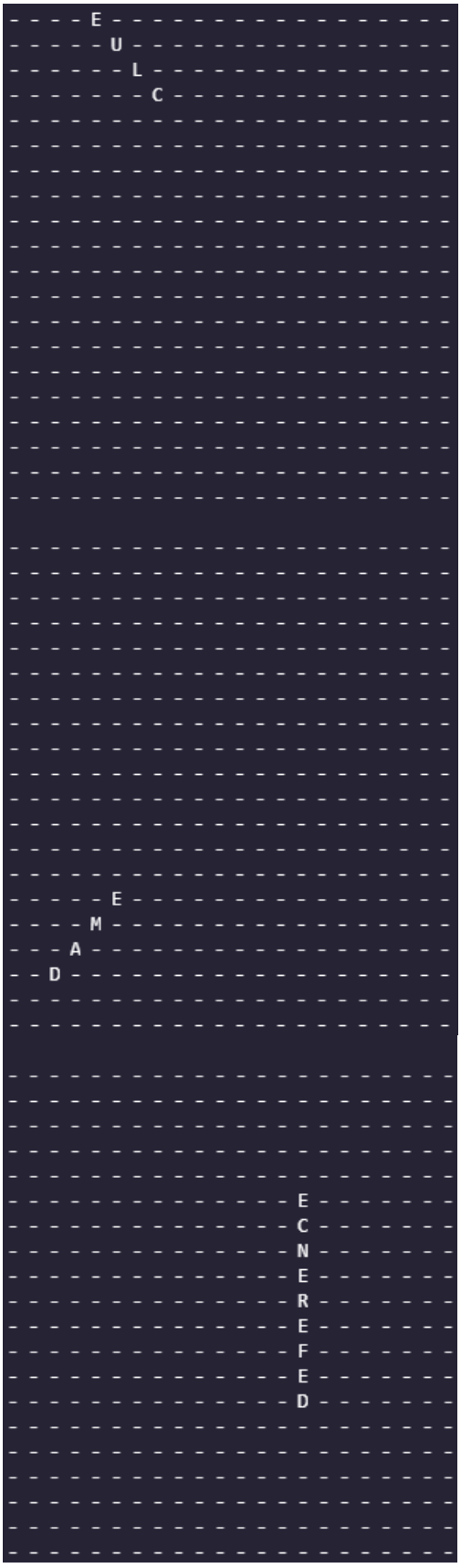
Word Search Puzzle:

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

ANNUALLY
BASEBALL
CASHING
CLUE
DAME
DEFERENCE
EDIBLE
GUILTLESS
HAILING
HORSEMAN
MODALITY
OVULAR
PERSONAL
REINVENT
SCARCITIES
SCENARIO
SLEAZY
SUNNY
TEXTBOOKS
UPWARDLY
```

Hasil Eksekusi:




BASEBALL
CASHING
CASHING


EULER
EMAD
ECNERFED

```
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - E - - - - - - - - - -
- - - - - - - - - D - - - - - - - - - - -
- - - - - - - - I - - - - - - - - - - - -
- - - - - - - B - - - - - - - - - - - - -
- - - - - - L - - - - - - - - - - - - - -
- - - - - E - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
```

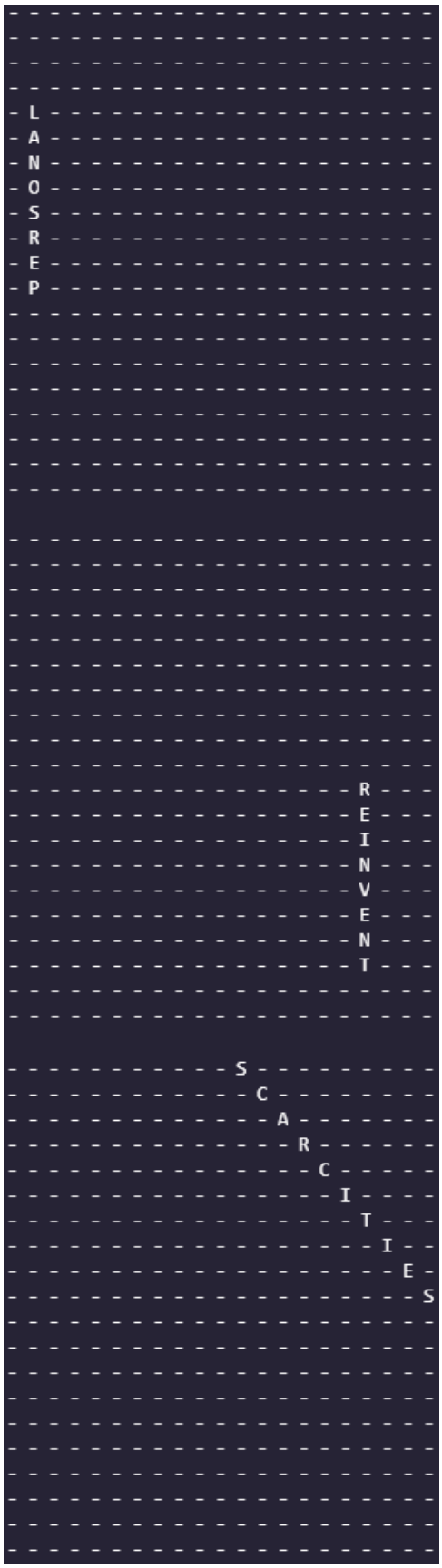
```
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - G - - - - - - - -
- - - - - - - - - - - - U - - - - - - - -
- - - - - - - - - - - - I - - - - - - - -
- - - - - - - - - - - - L - - - - - - - -
- - - - - - - - - - - - T - - - - - - - -
- - - - - - - - - - - - L - - - - - - - -
- - - - - - - - - - - - E - - - - - - - -
- - - - - - - - - - - - S - - - - - - - -
- - - - - - - - - - - - S - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
```

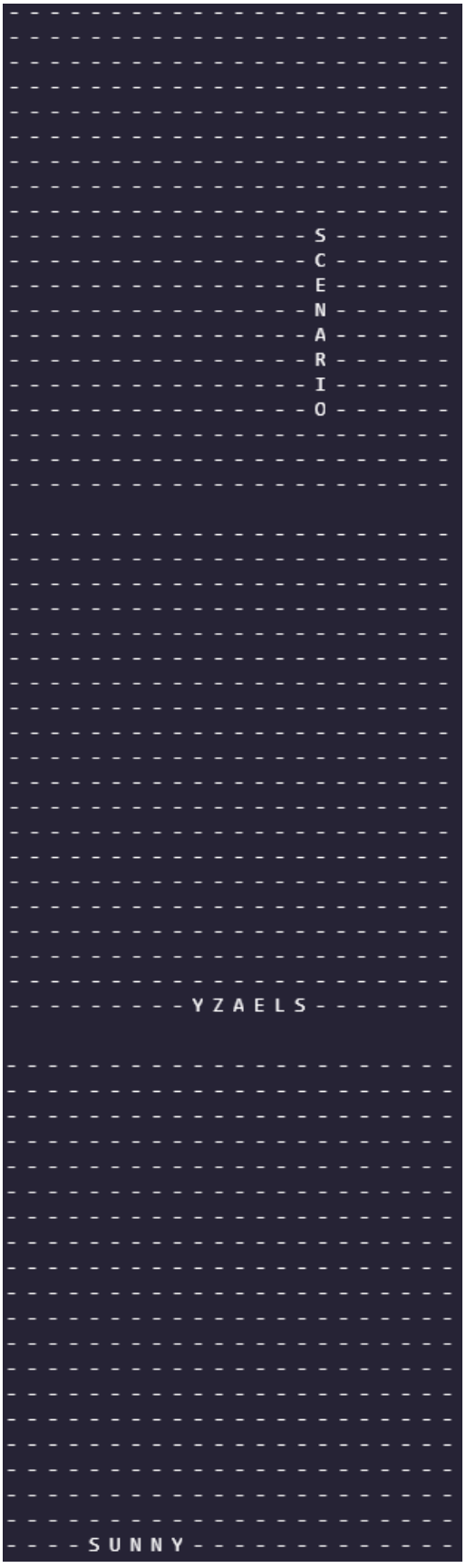
```
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - H - - - - - - - - - - - - - - - - - -
- - A - - - - - - - - - - - - - - - - - -
- - I - - - - - - - - - - - - - - - - - -
- - L - - - - - - - - - - - - - - - - - -
- - I - - - - - - - - - - - - - - - - - -
- - N - - - - - - - - - - - - - - - - - -
- - G - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - -
```

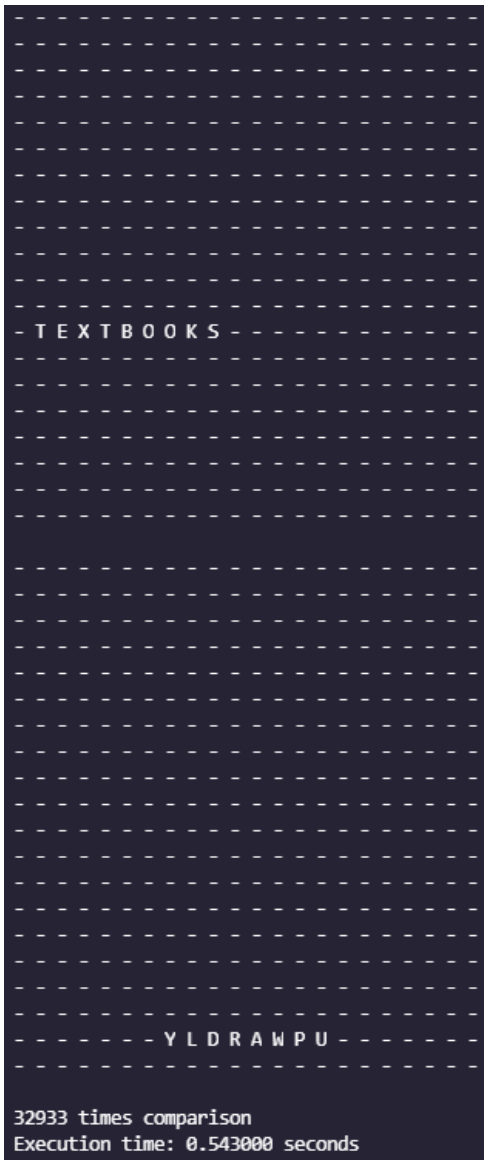
```
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - H - - - - - - - - - -
- - - - - - - - - - - O - - - - - - - - - -
- - - - - - - - - - - R - - - - - - - - - -
- - - - - - - - - - - S - - - - - - - - - -
- - - - - - - - - - - E - - - - - - - - - -
- - - - - - - - - - - M - - - - - - - - - -
- - - - - - - - - - - A - - - - - - - - - -
- - - - - - - - - - - N - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - Y - - - - - - - - - - -
- - - - - - - - - T - - - - - - - - - - - -
- - - - - - - - I - - - - - - - - - - - - -
- - - - - - - L - - - - - - - - - - - - - -
- - - - - - A - - - - - - - - - - - - - - -
- - - - - D - - - - - - - - - - - - - - - -
- - - - O - - - - - - - - - - - - - - - - -
- - - M - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - R - - - - - - - - - - - - -
- - - - - - - A - - - - - - - - - - - - - -
- - - - - - L - - - - - - - - - - - - - - -
- - - - - U - - - - - - - - - - - - - - - -
- - - - V - - - - - - - - - - - - - - - - -
- - - O - - - - - - - - - - - - - - - - - -
```


LANOSREP
REINVENT
SCARCITIES

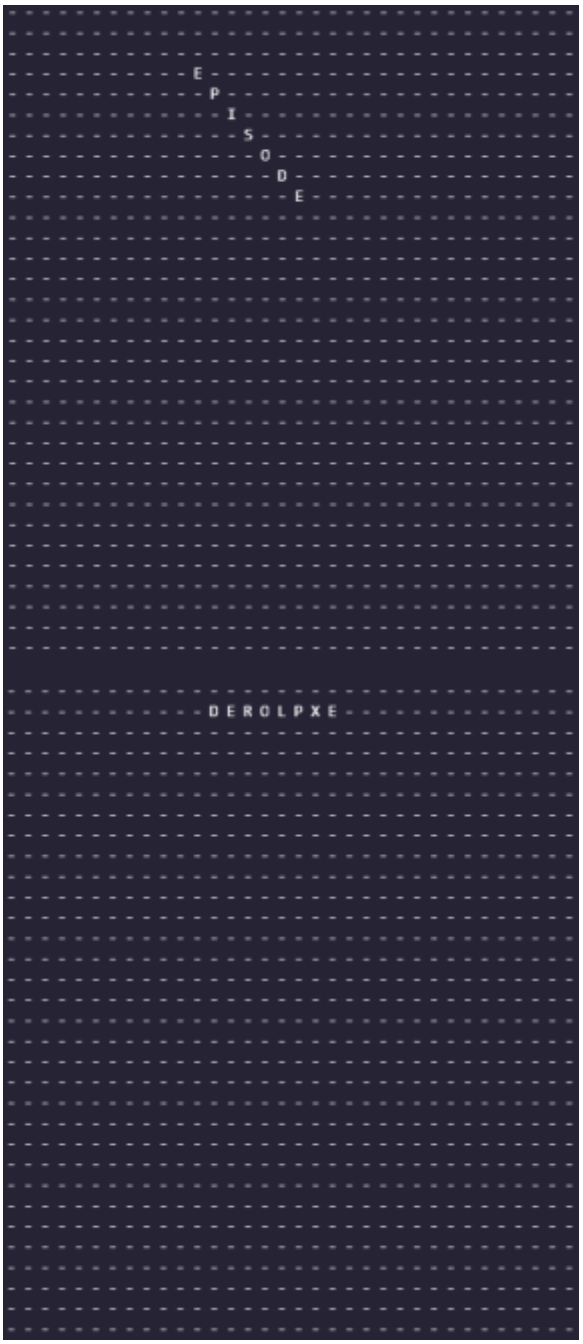

SCENARIO
YZAELS
SUNNY

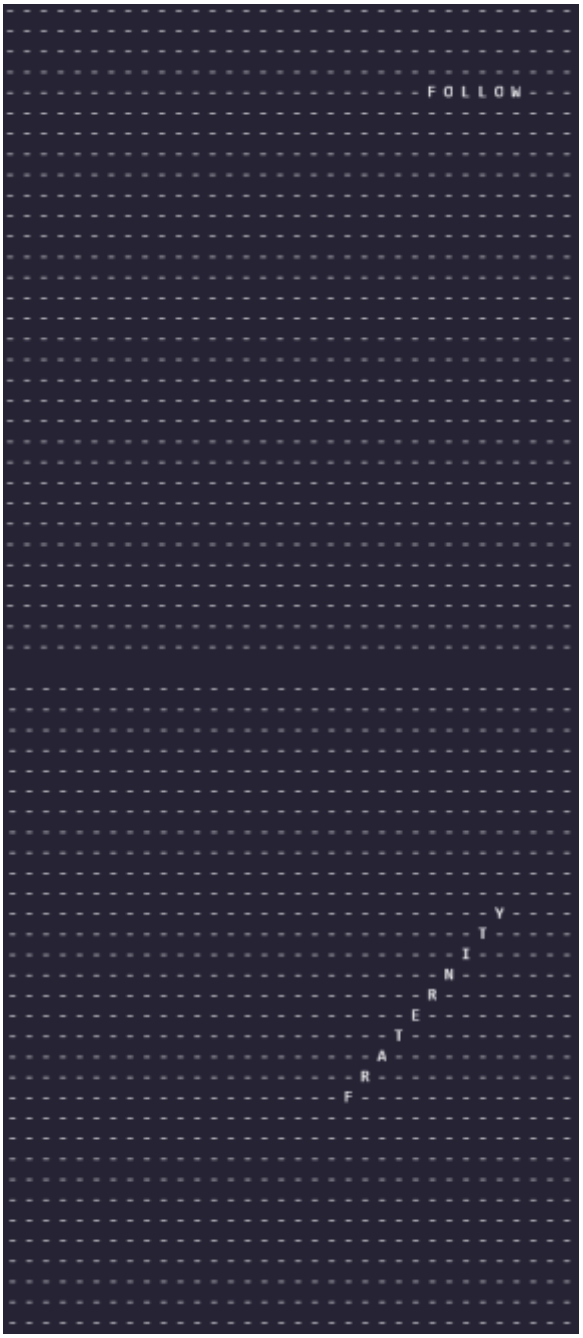


g. Puzzle large 1
Word Search Puzzle:

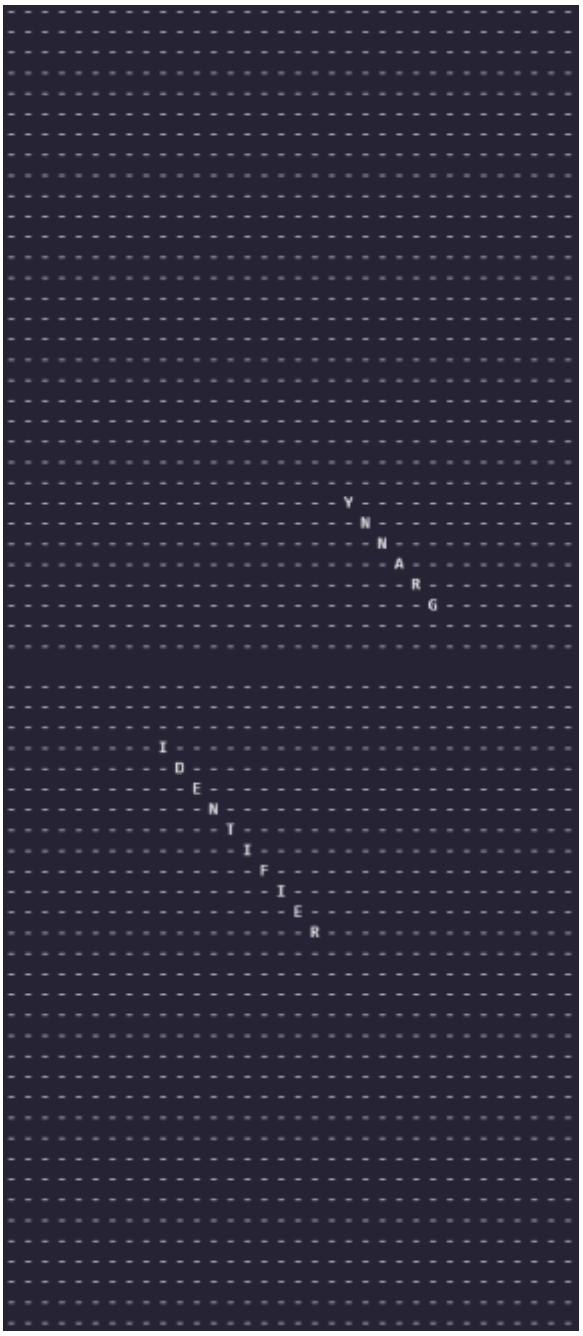
Hasil eksekusi:

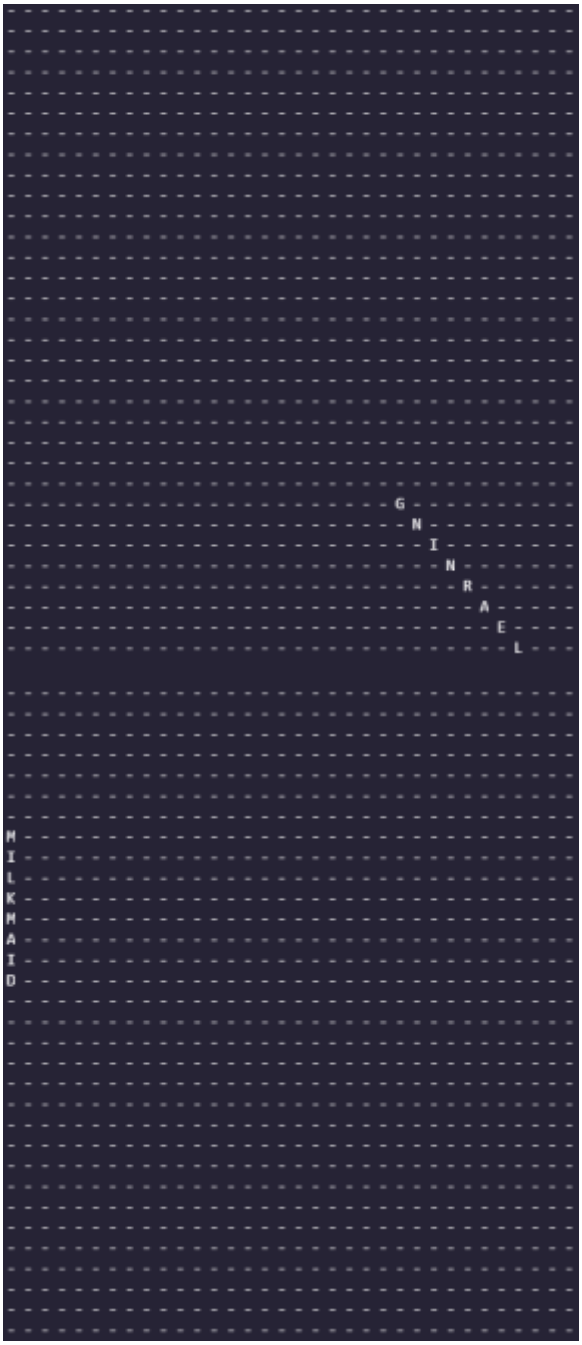

EMPRESSE
GNITALUME

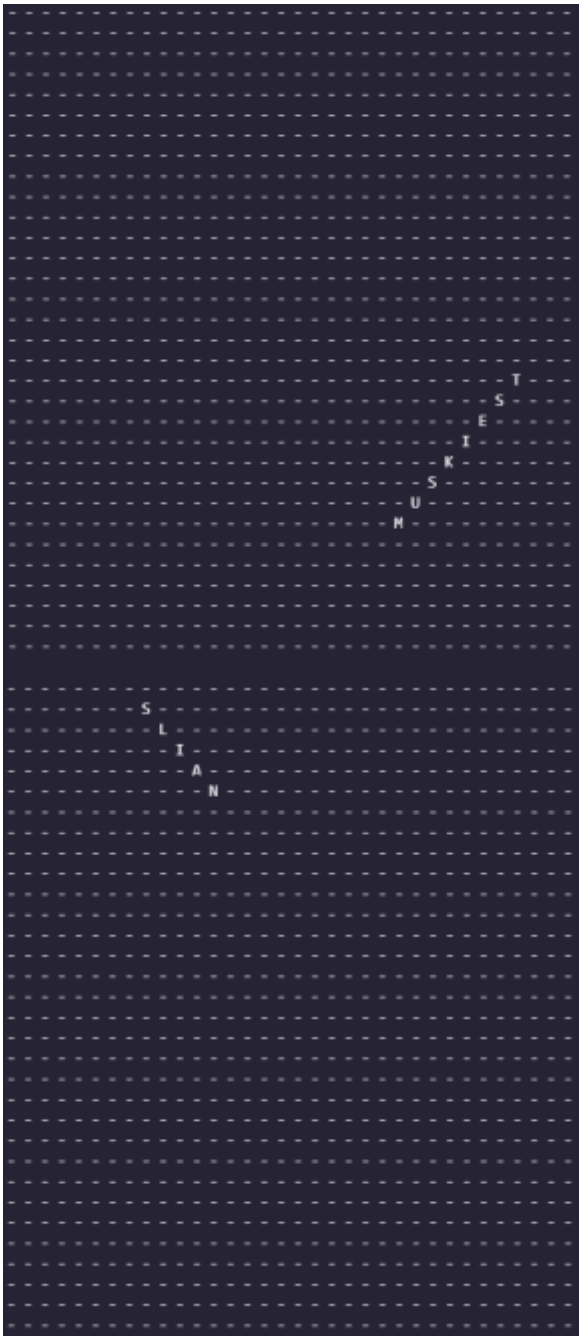

EPISODE
DEROLPXE

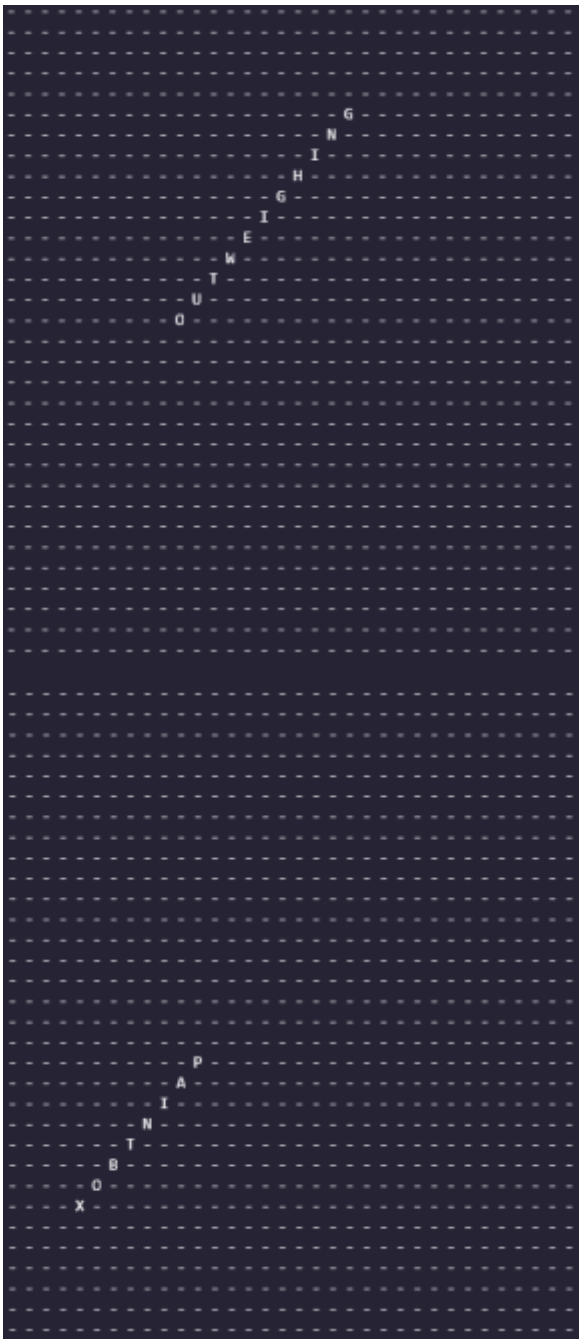

FOLLOW
Y
T
I
N
R
E
T
A
R
F

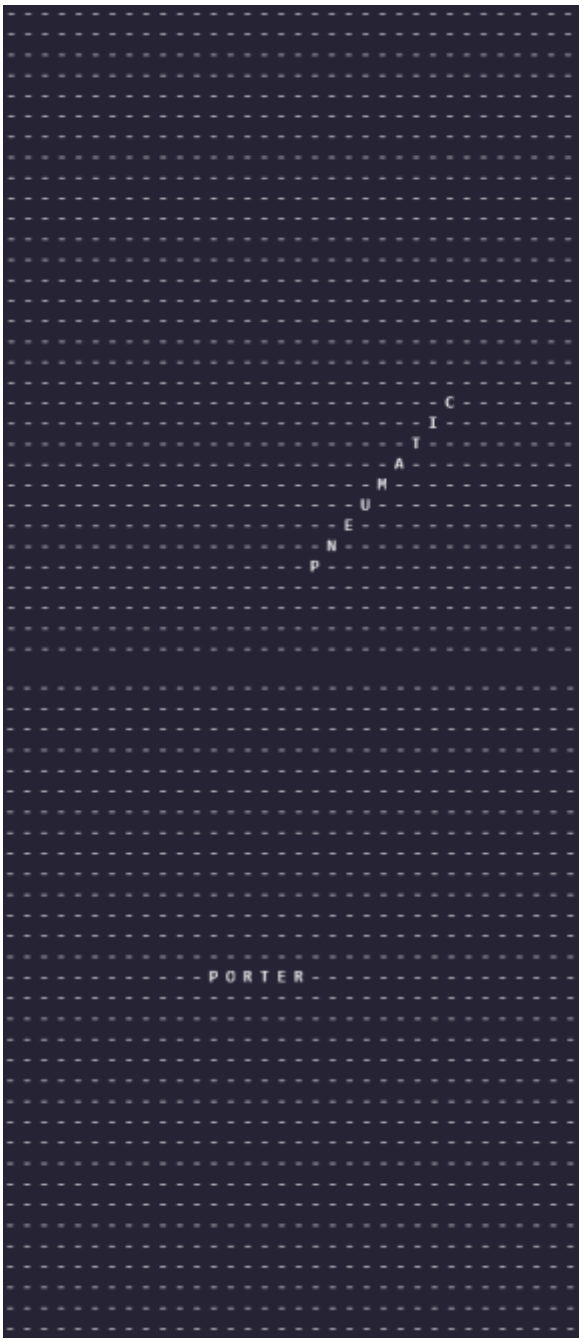

FUSION
W
O
G
S
A
L
G

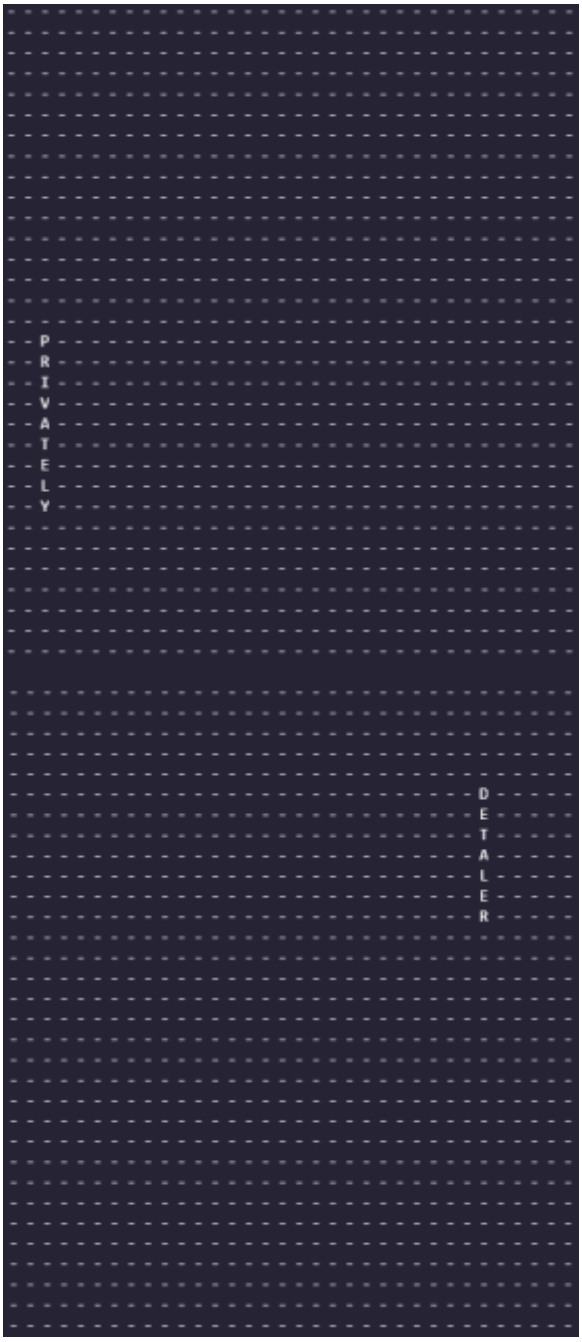

YNNARG
IDENTIFIER

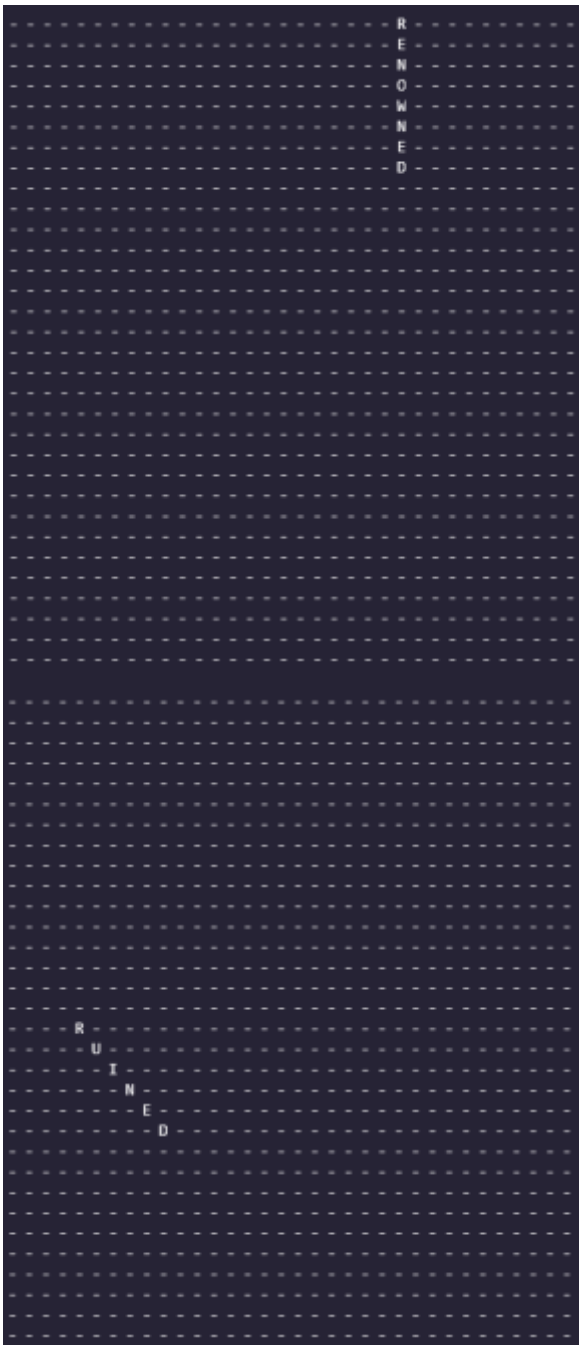

GNINRAEL
MILKMAID

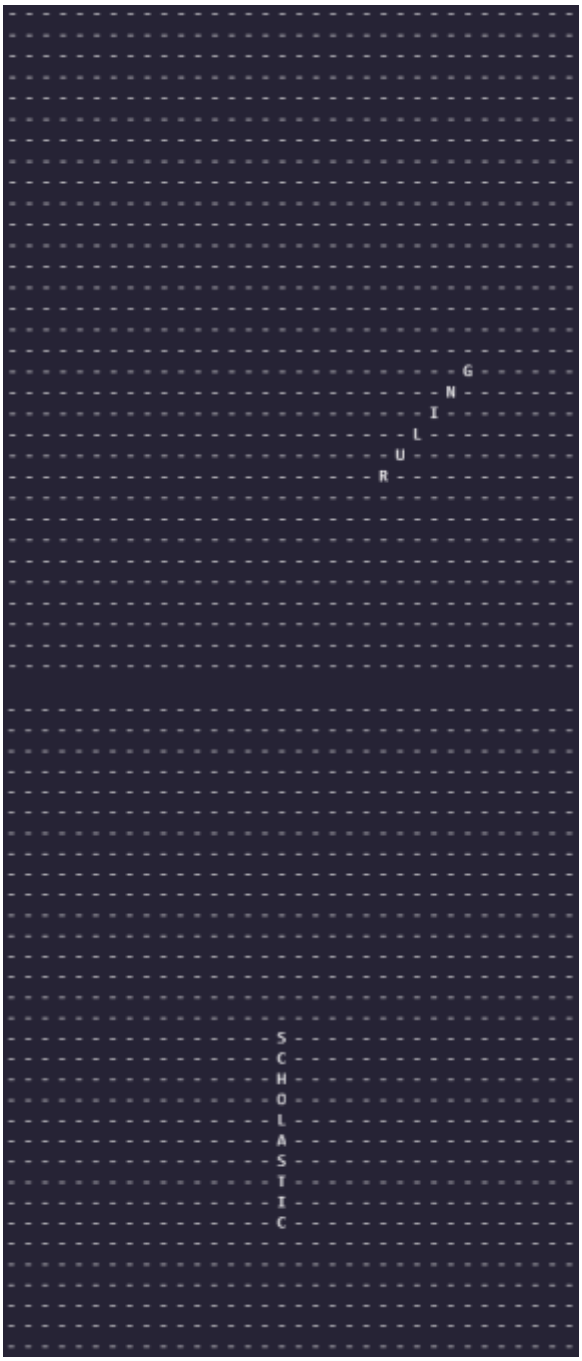

T
S
E
I
K
S
U
M
S
L
I
A
N

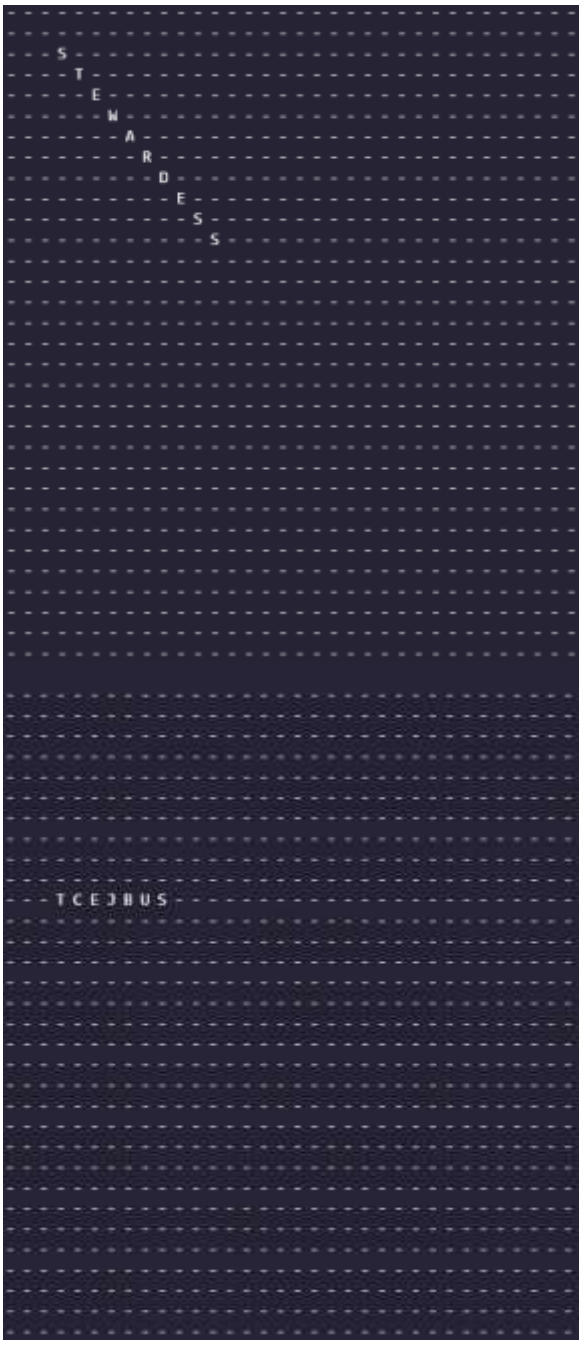

G
N
I
H
G
I
E
W
T
U
O
P
A
I
N
T
B
O
X

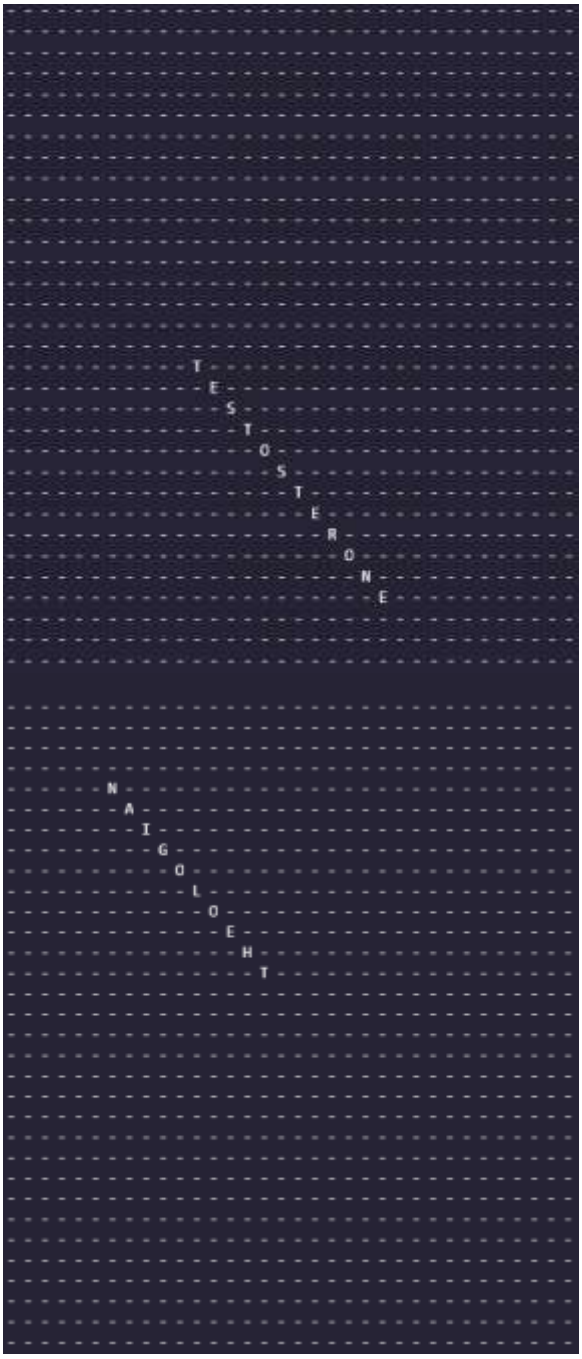

PNEUMATIC
PORTER

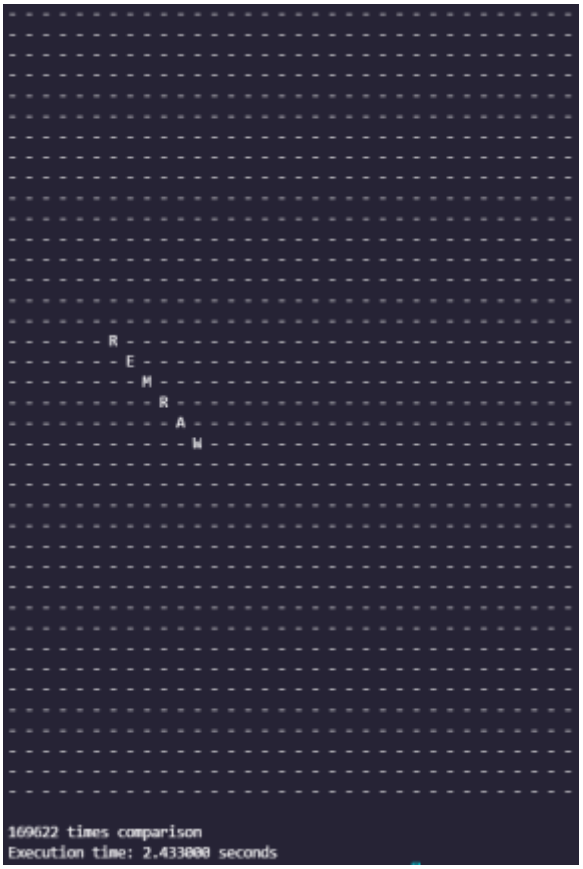

PRIVATELY
DETAILER

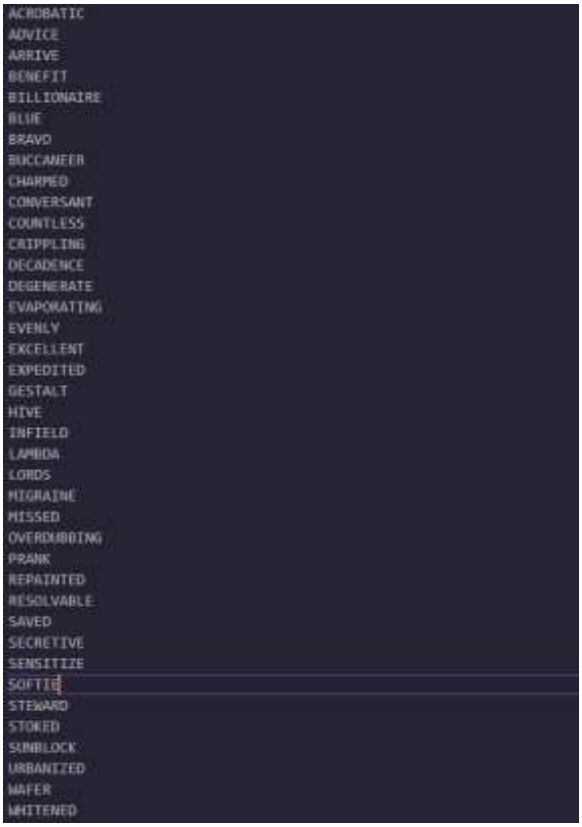

RENOWNED
RUINED


GNILURSCHOLASTIC

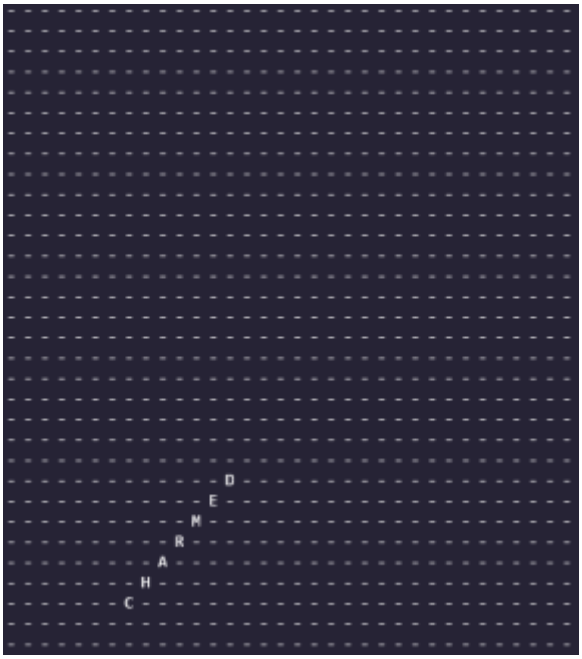

STEWARDESS
TCEJBUS


TESTOSTERONE
NAIGOLOETH



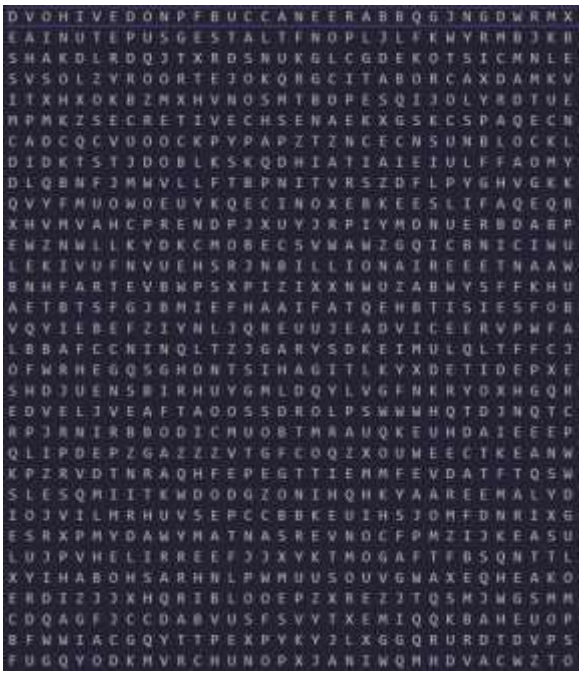


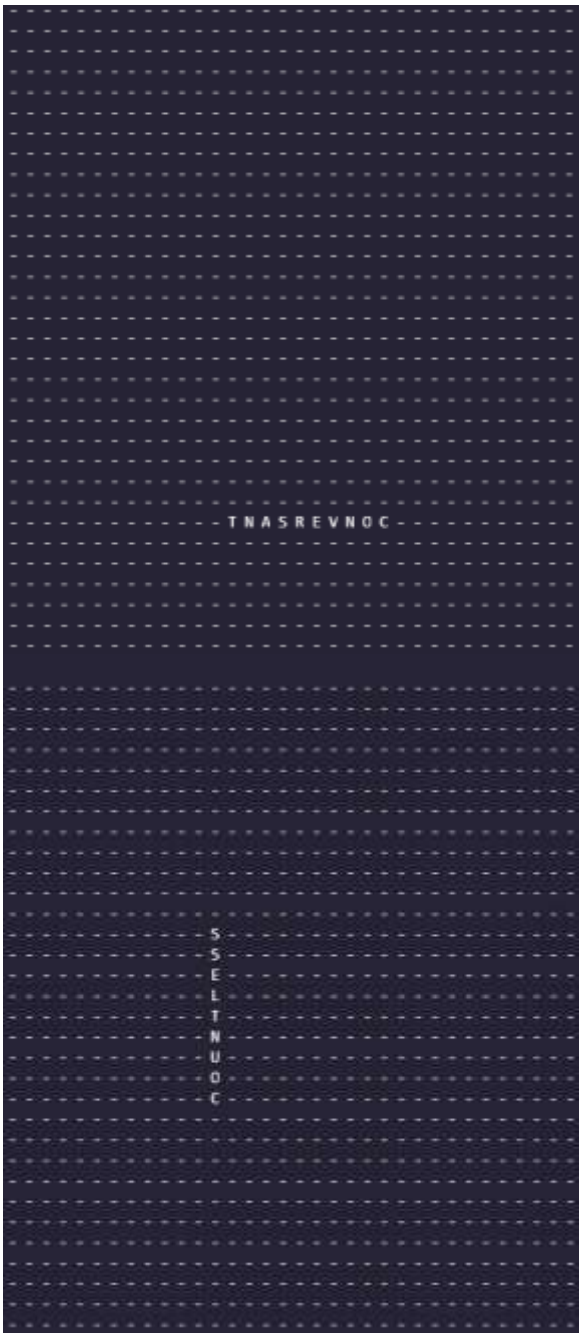
Hasil eksekusi:

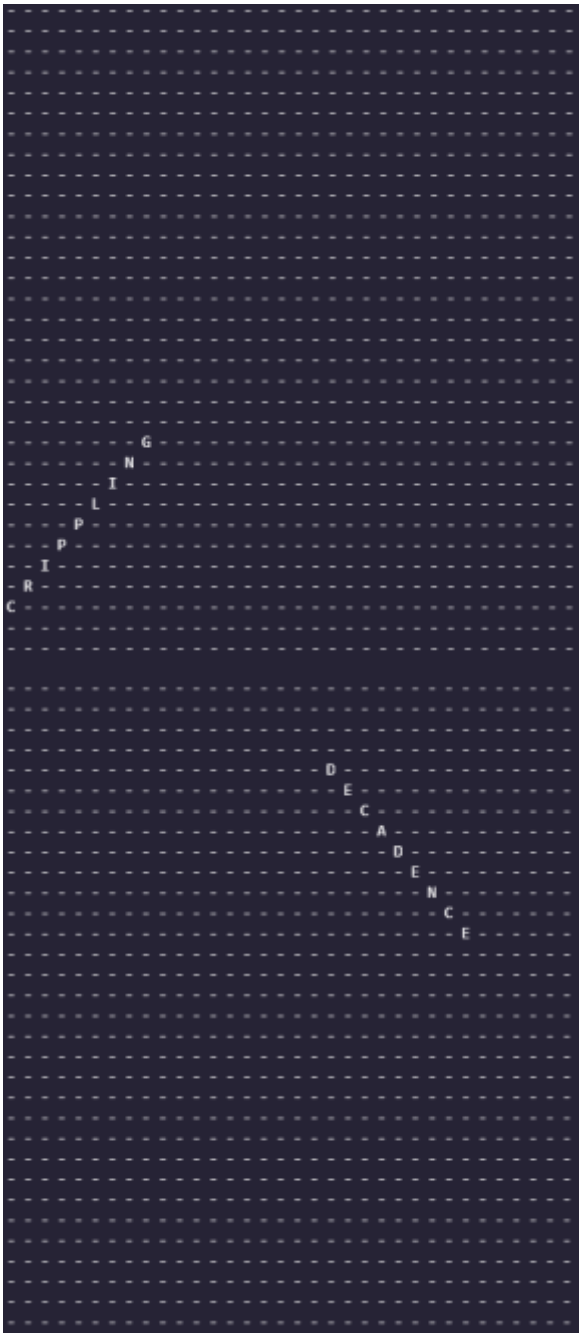


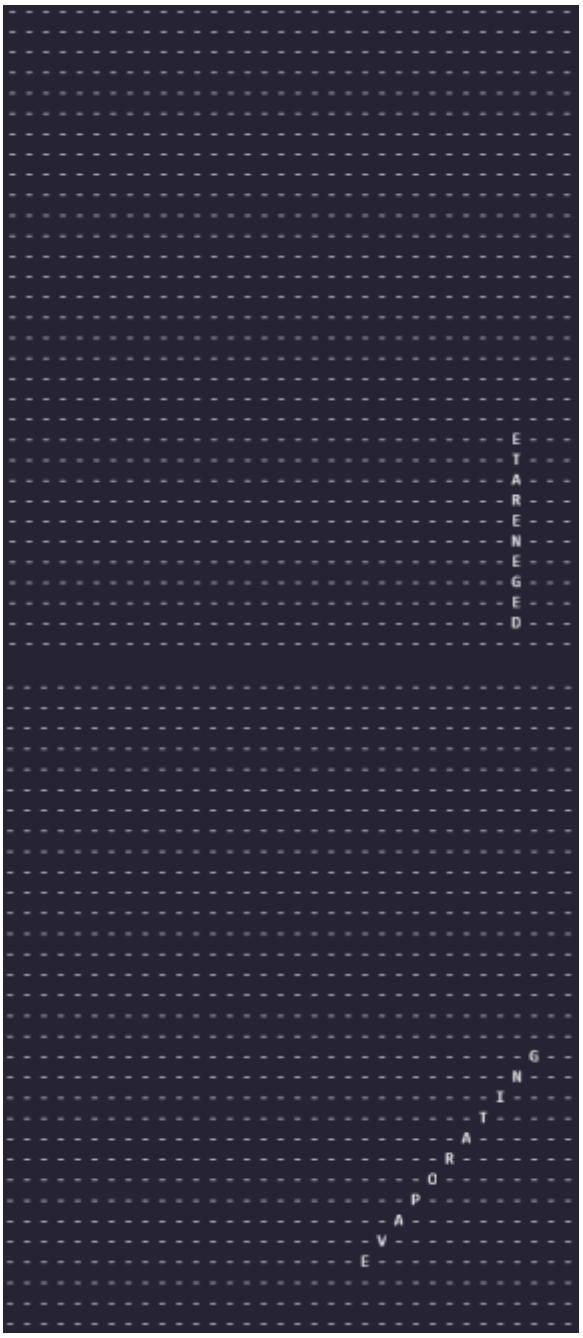
h. Puzzle large 2

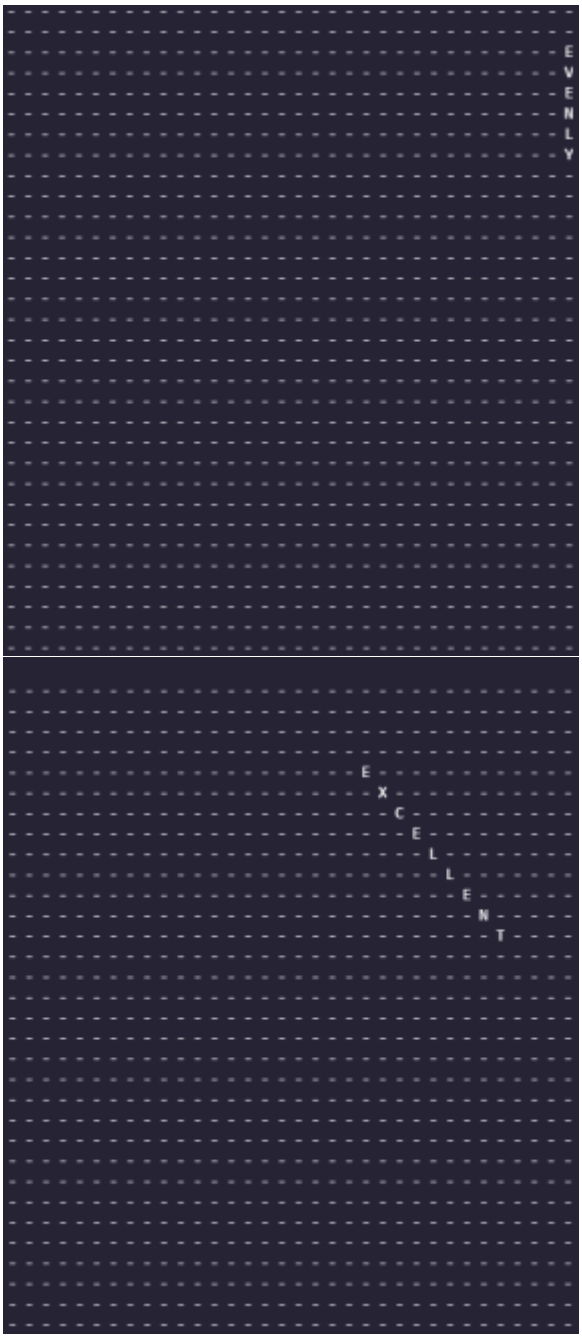
Word Search Puzzle:



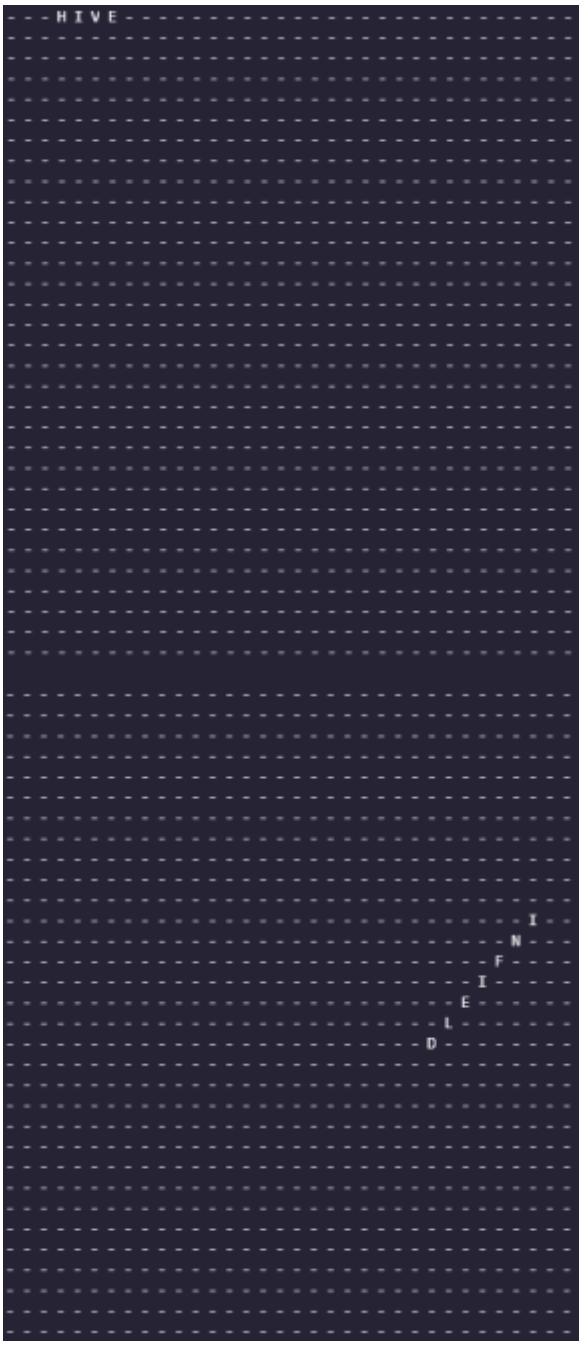

CONVERSATION
COUNSELLING

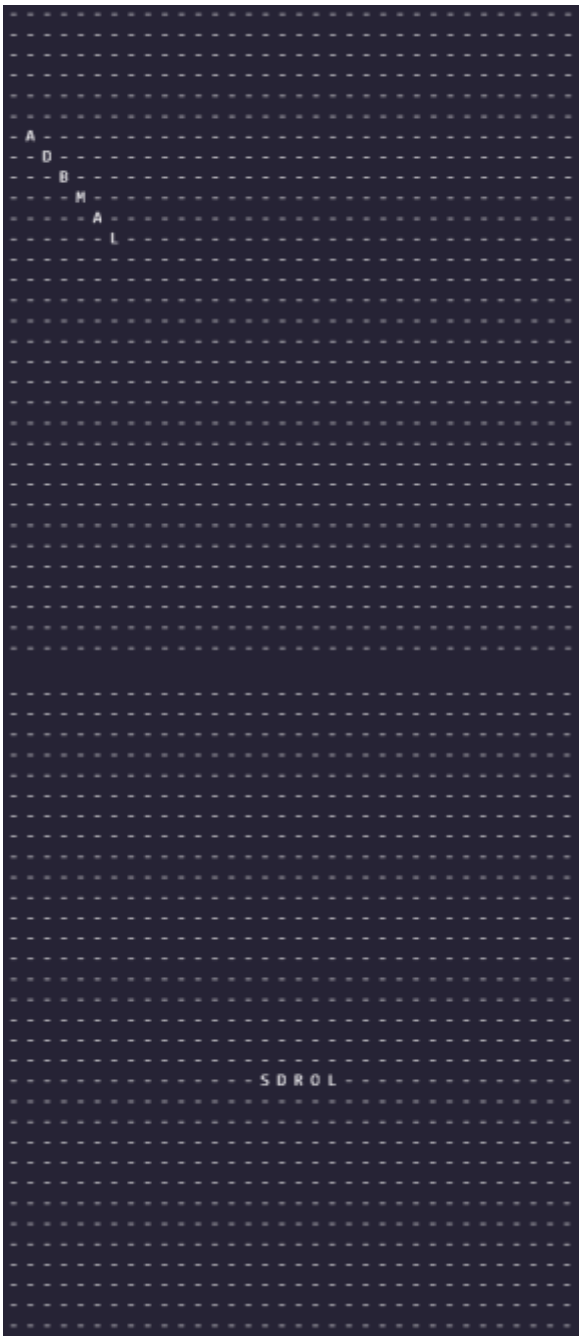

CRIPPLING
DECADENCE

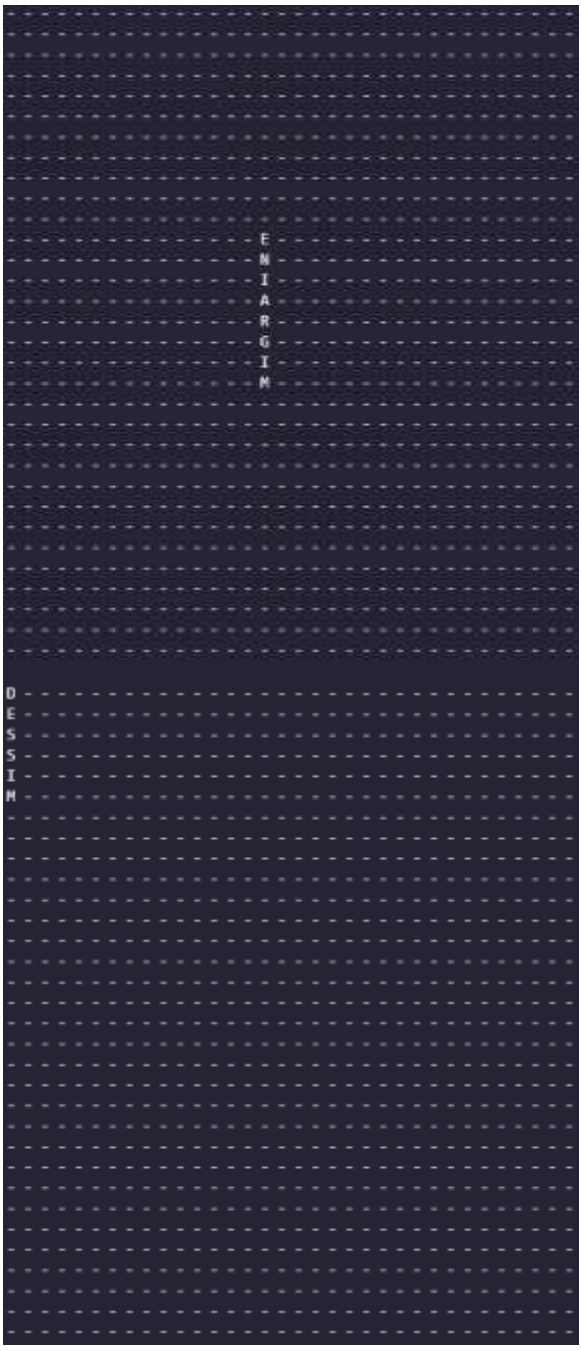

ENGAGED
TARGETING
EVAPORATE

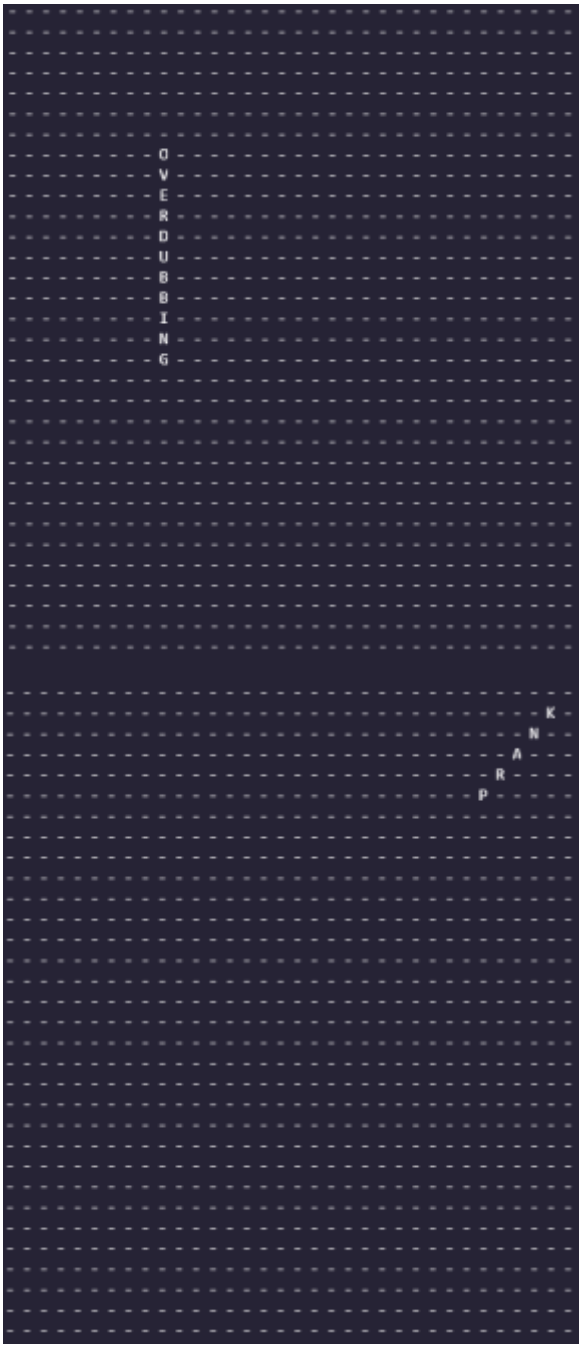

EVENLY
EXCELLENT


DETIDEPXE
GESTALT


HIVE
I
N
F
I
E
L
D

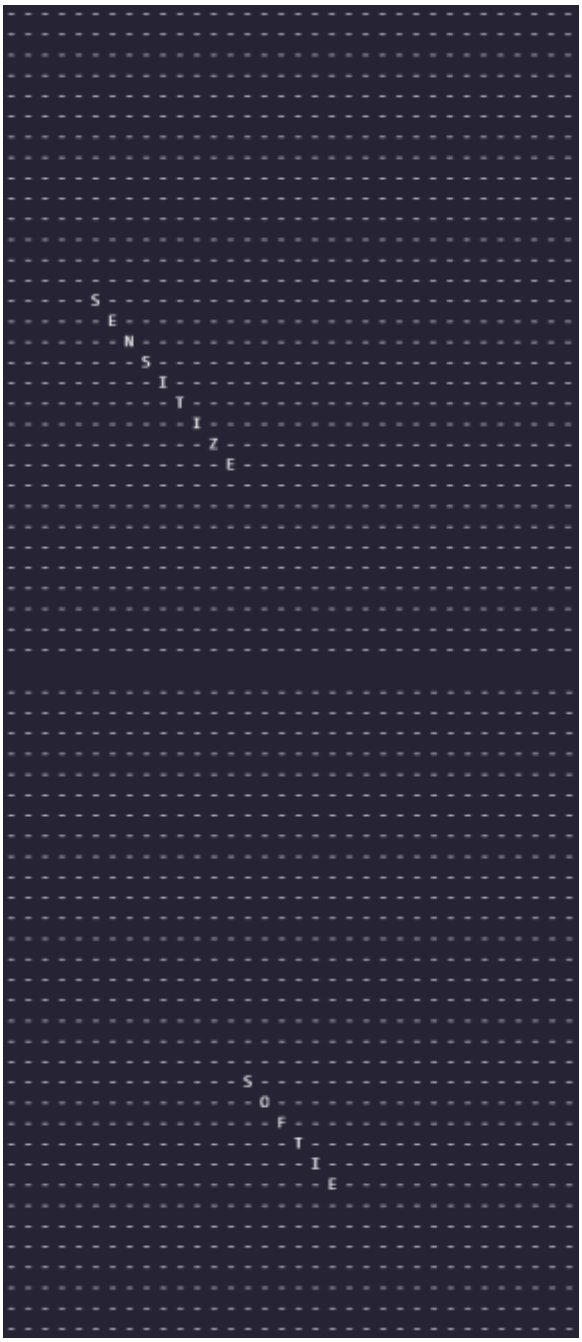

ADBMAL
SDROL

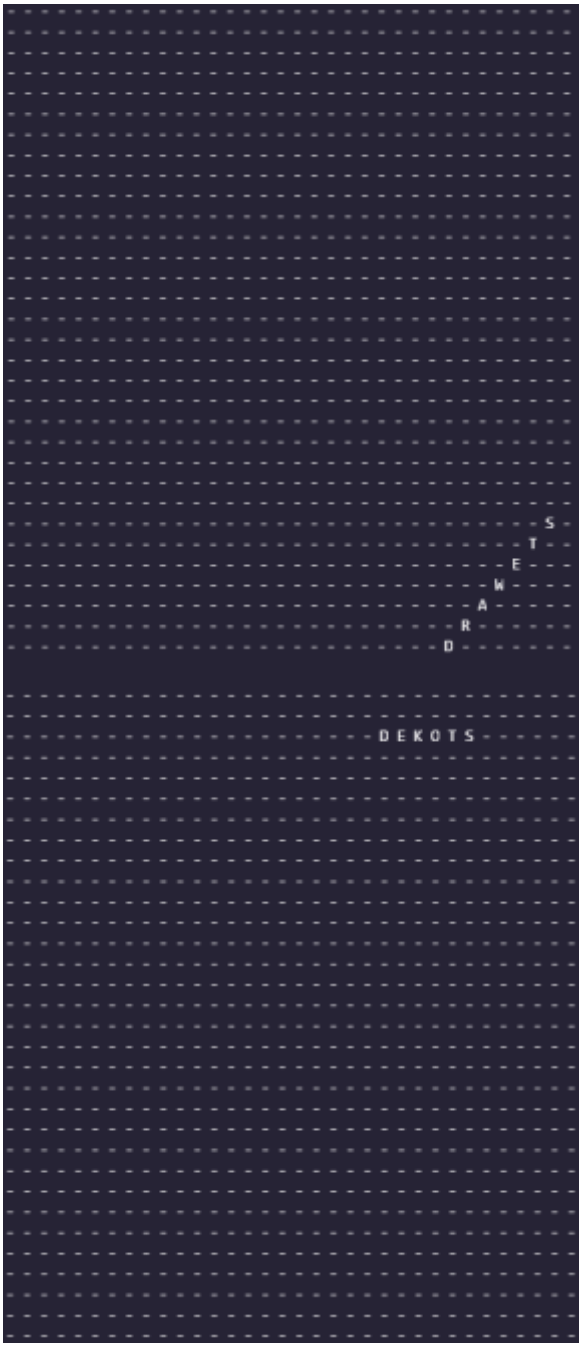

ENIARGIM
DESSIM

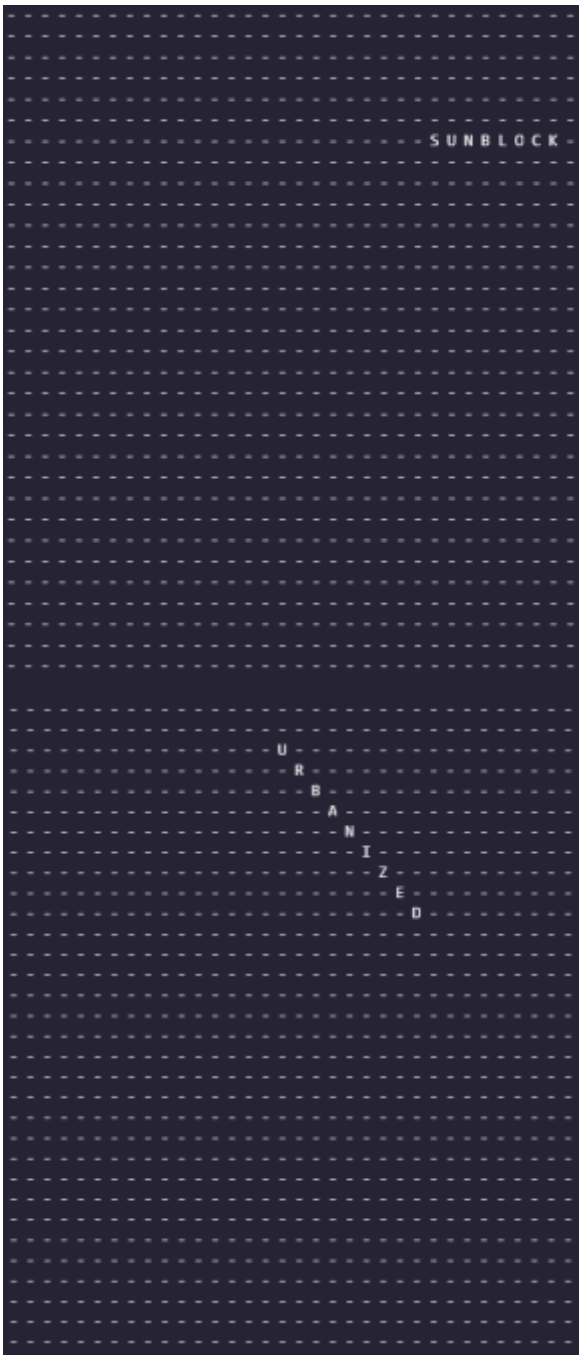

OVERDUBBING
KNARP

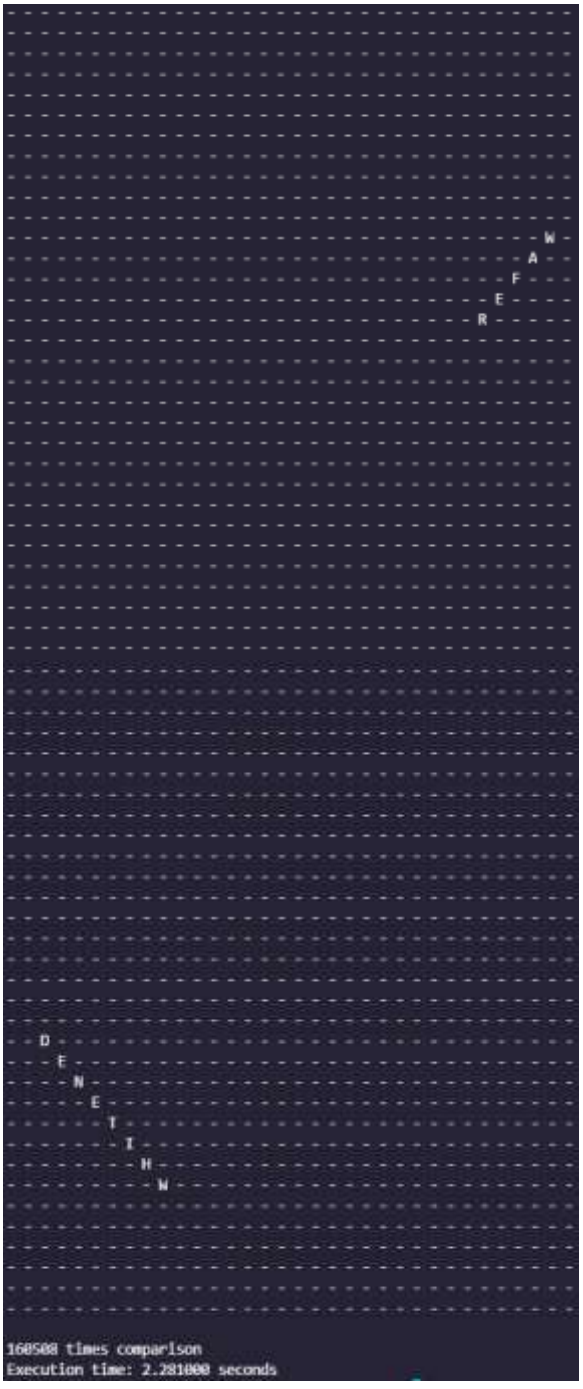

REPAINTED
ELBAVLOSER


DEVASE
SECRETIVE

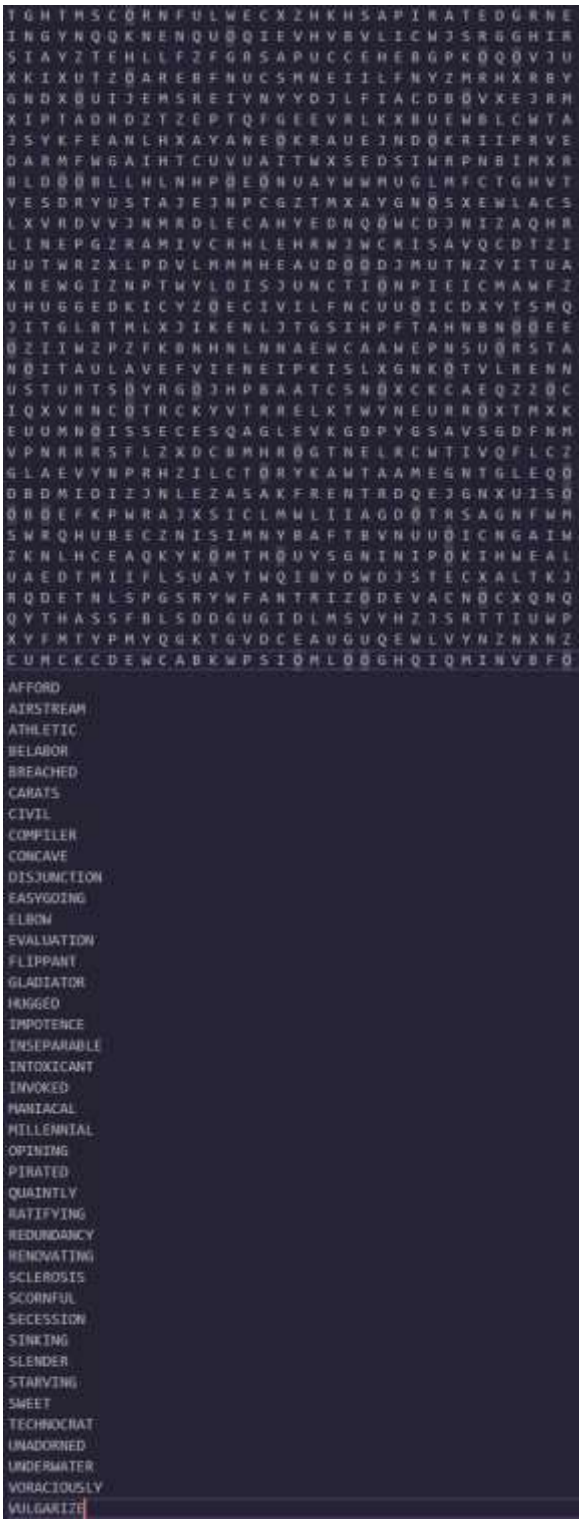

SENSITIZE
SOFTIE

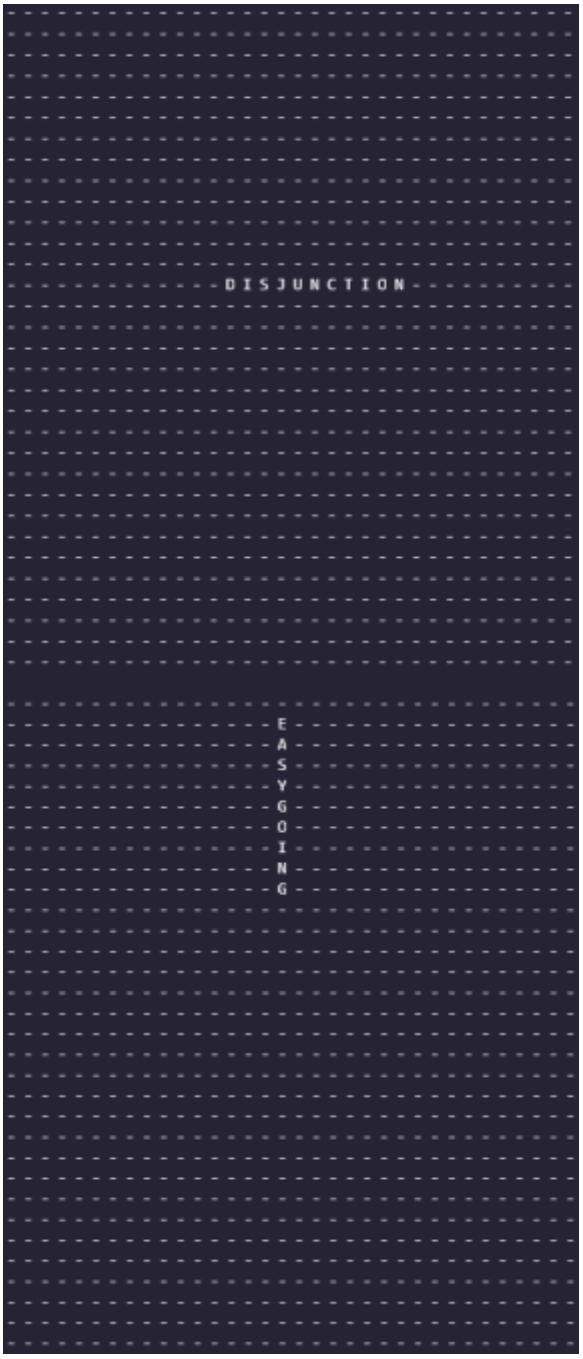

STEWARD
DEKOTS

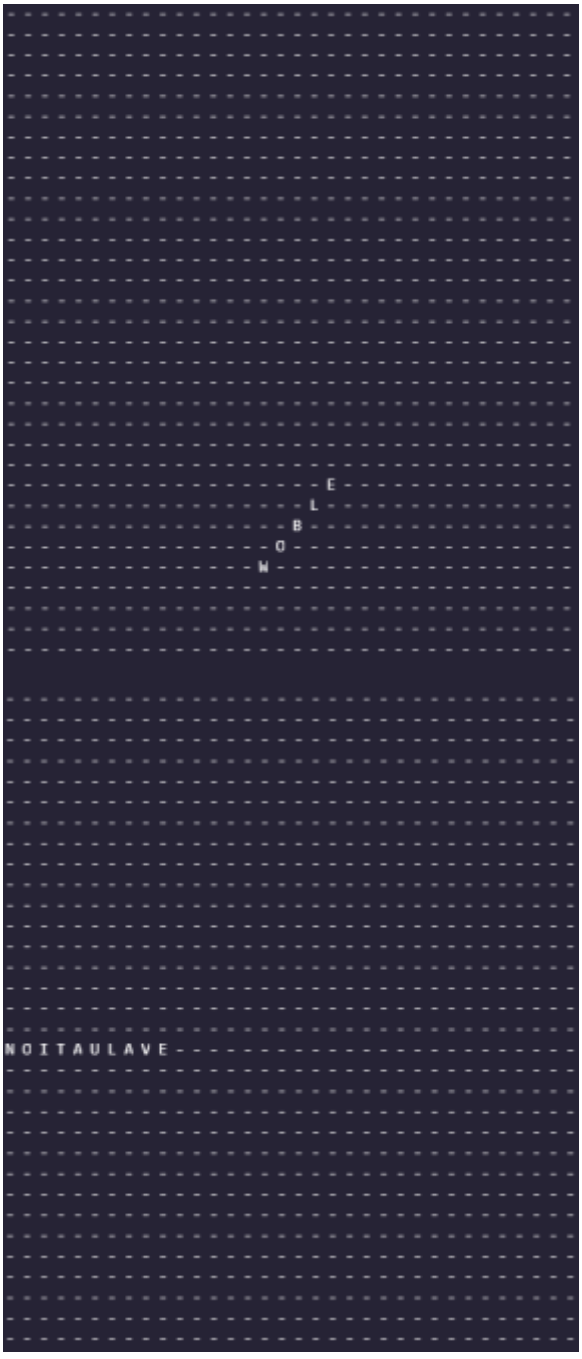

SUNBLOCK
URBANIZED

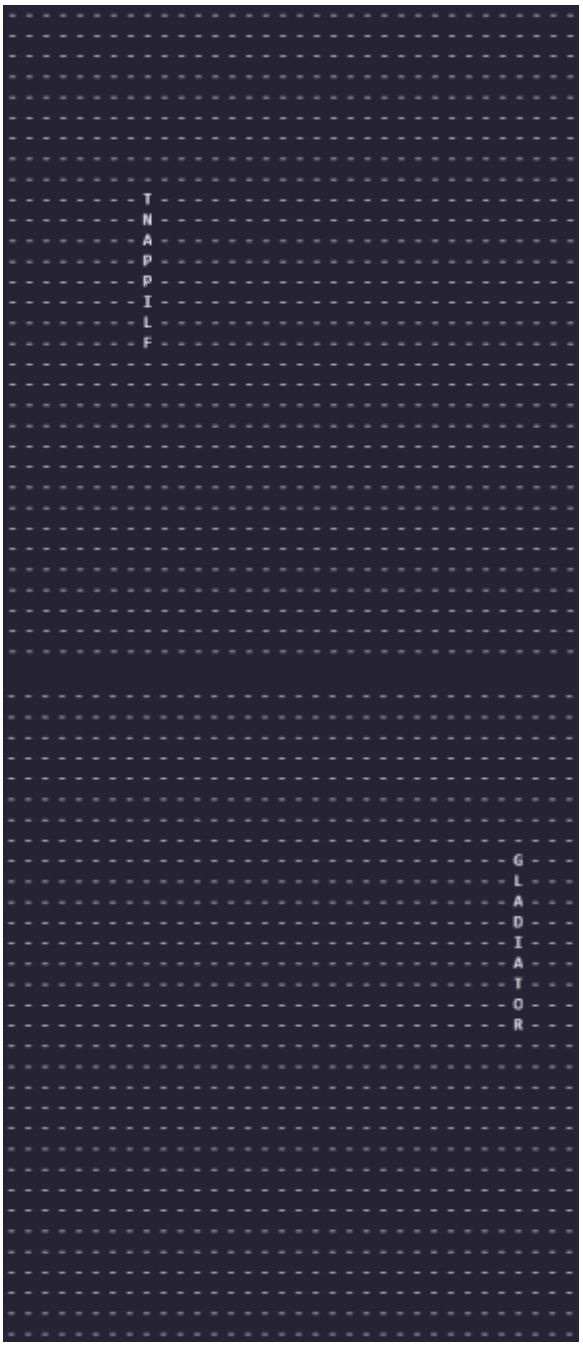


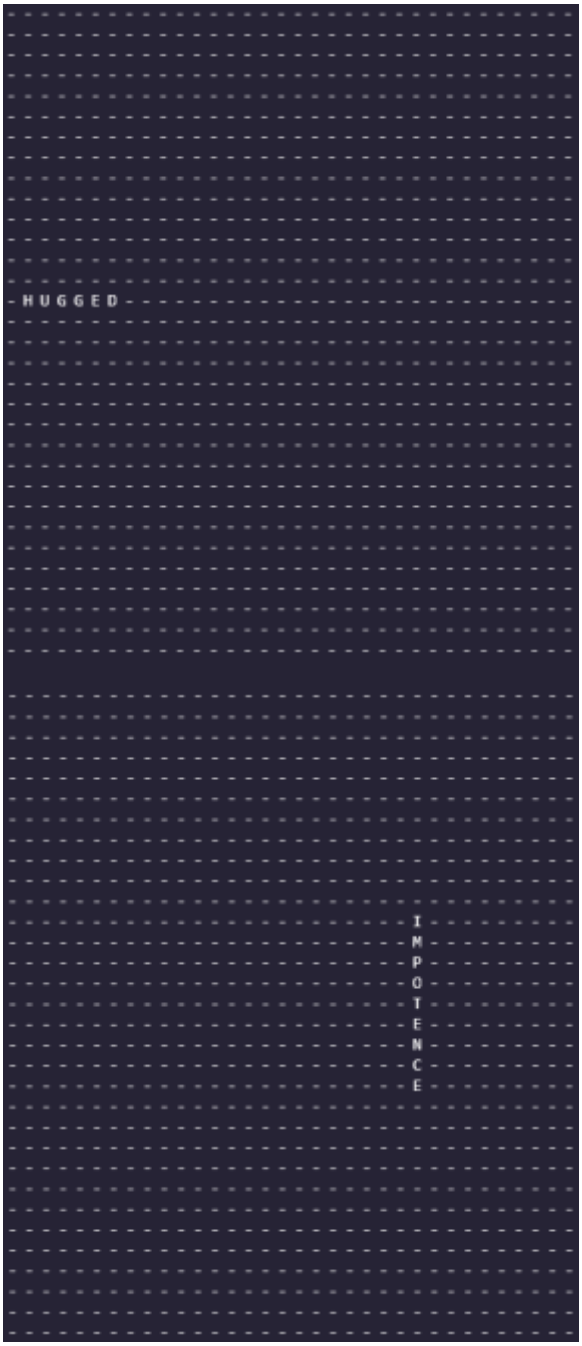
i. Puzzle large 3
Word search puzzle:

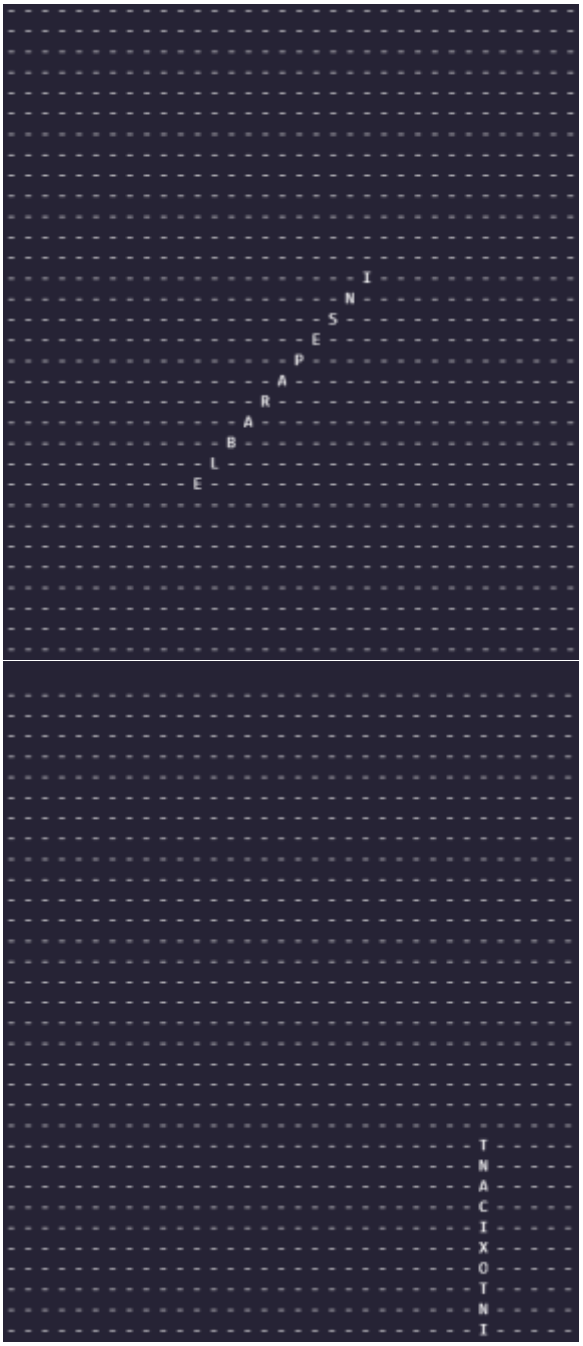


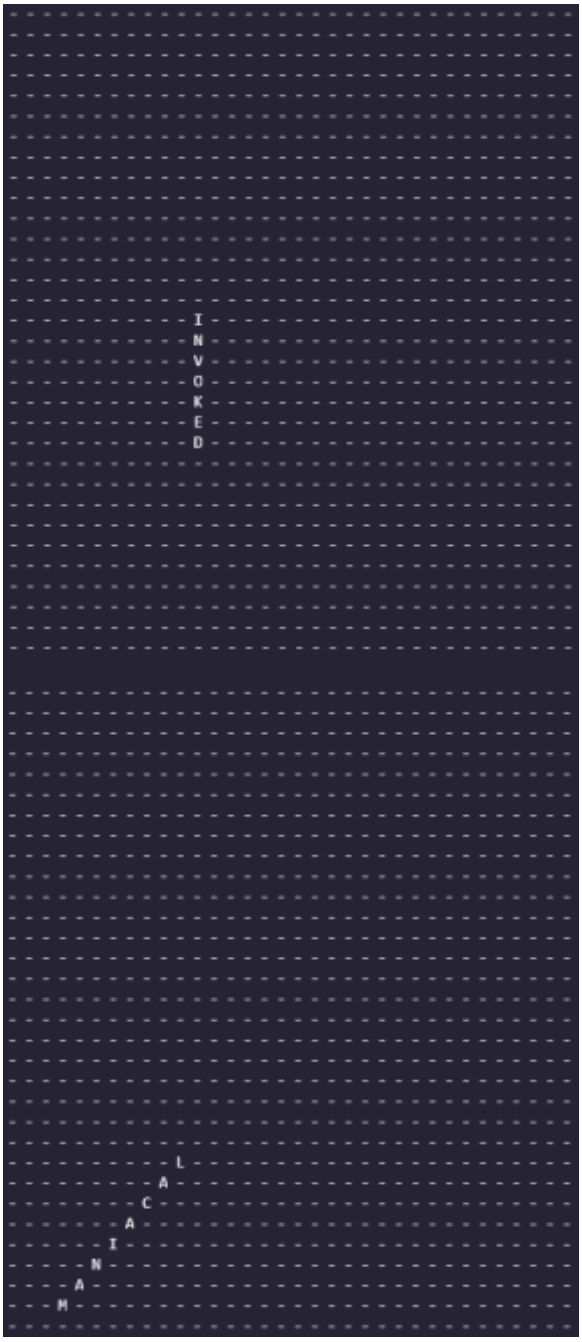

DISJUNCTION
EASYGOING

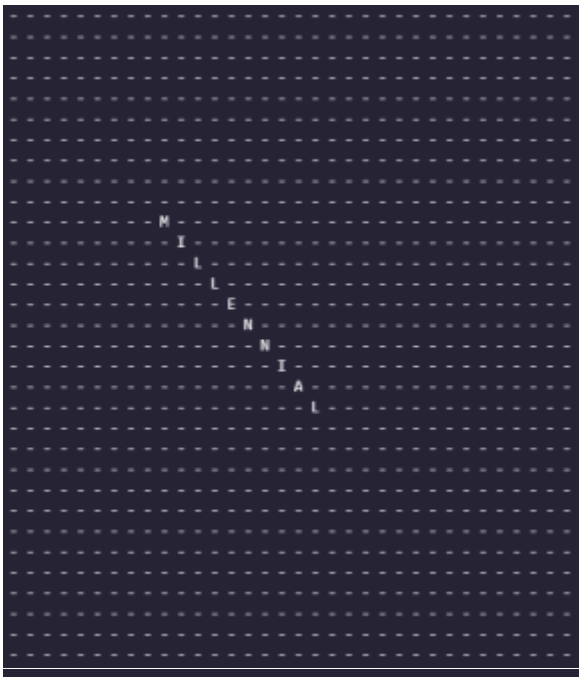

ELBOW
NOITAULAVE

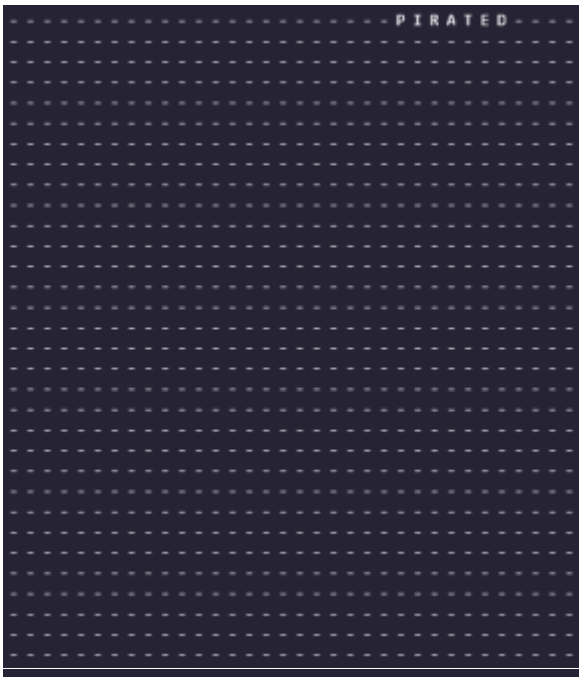

TNAPILF
GLADIATOR

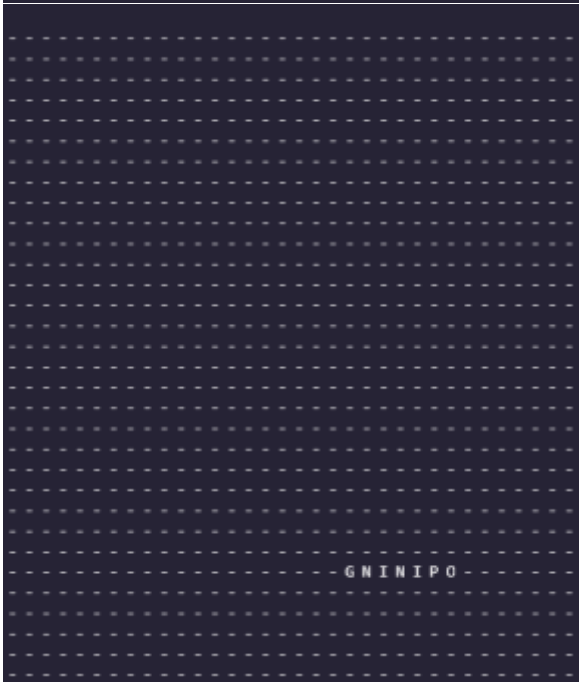

HUGGED
IMPOTENCE

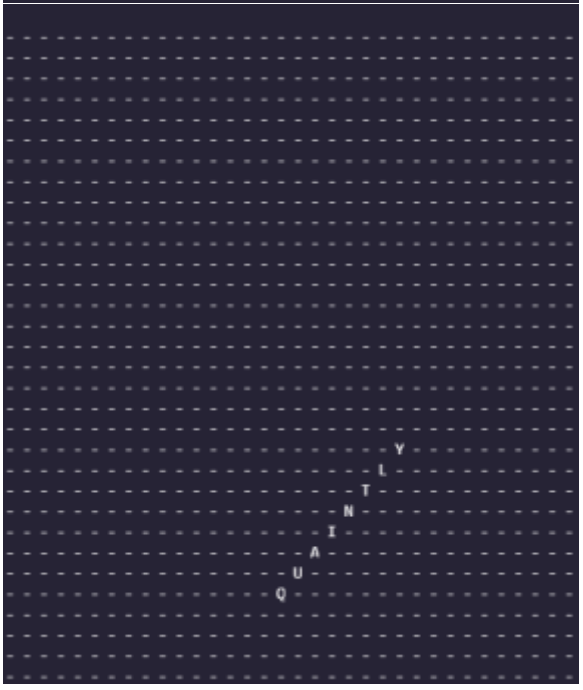

INSEPARABLE
TNACIXOTNI

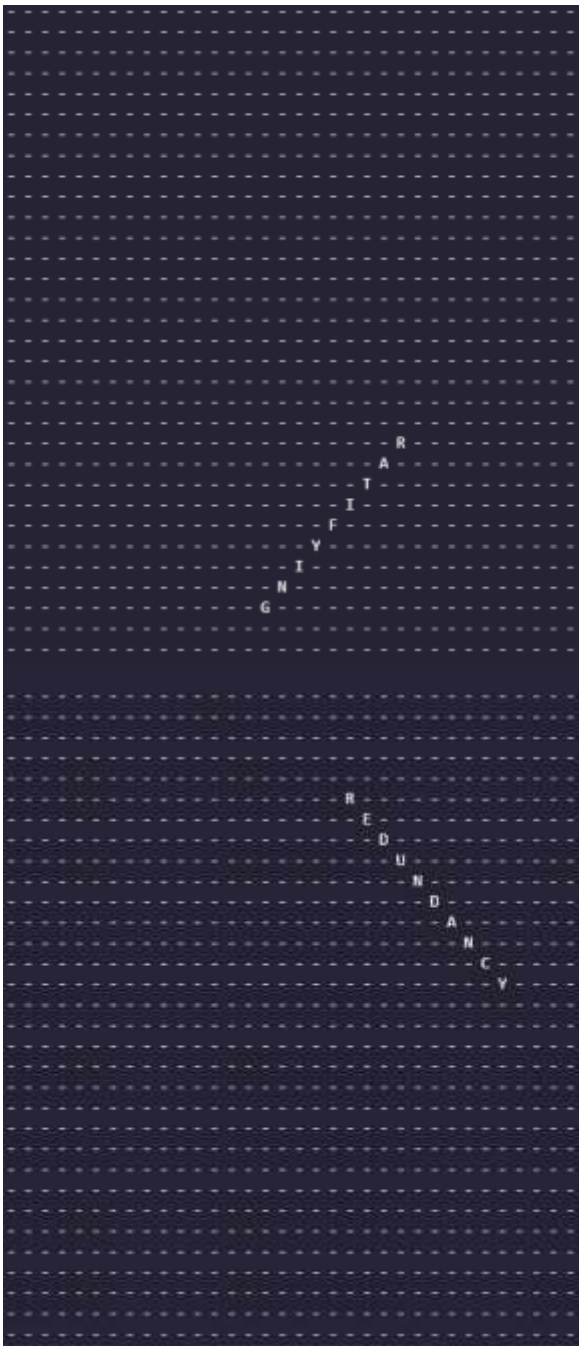

INVOKED
LAACAINAM

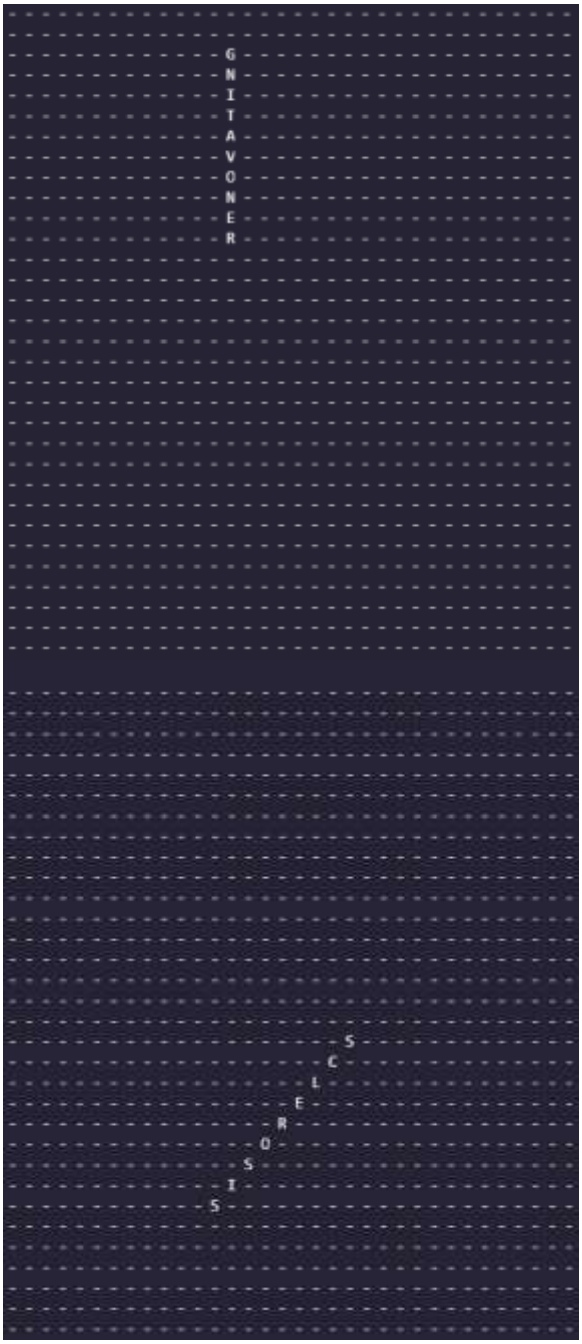

MILLENNIAL

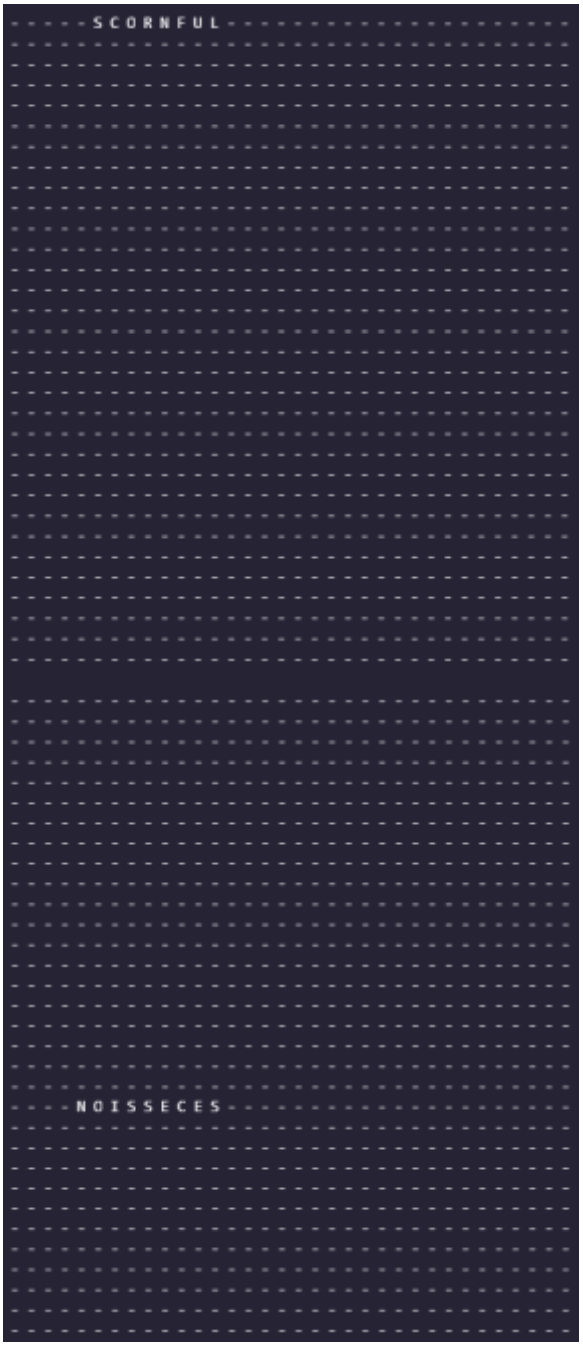

PIRATED

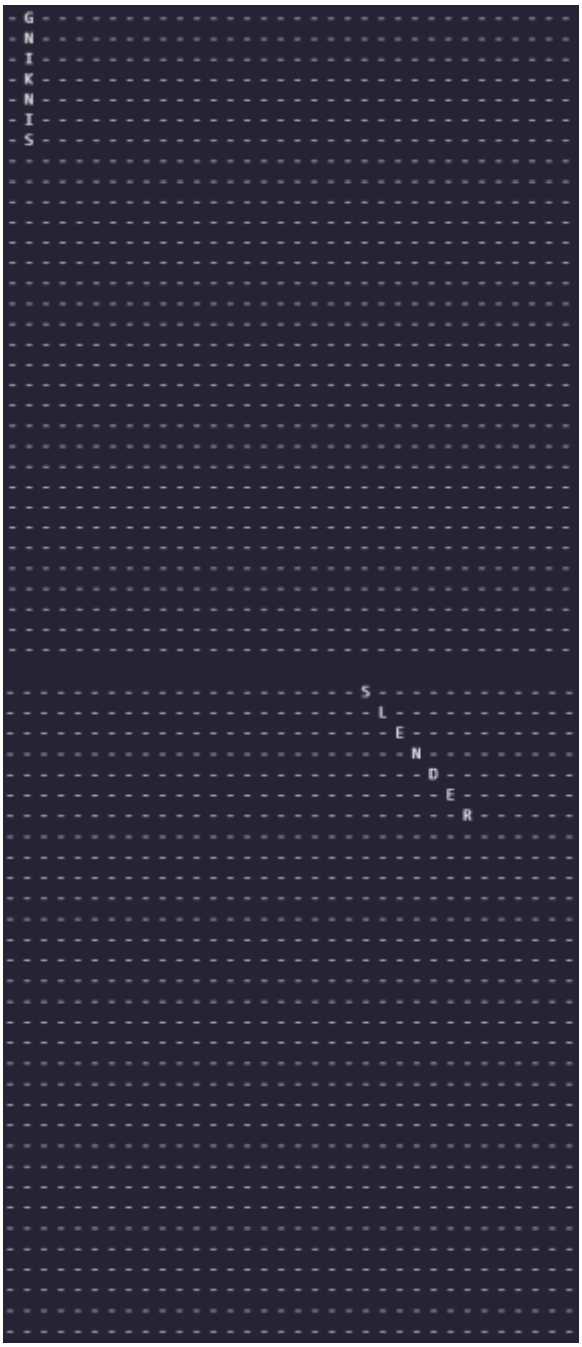

GNINIPO

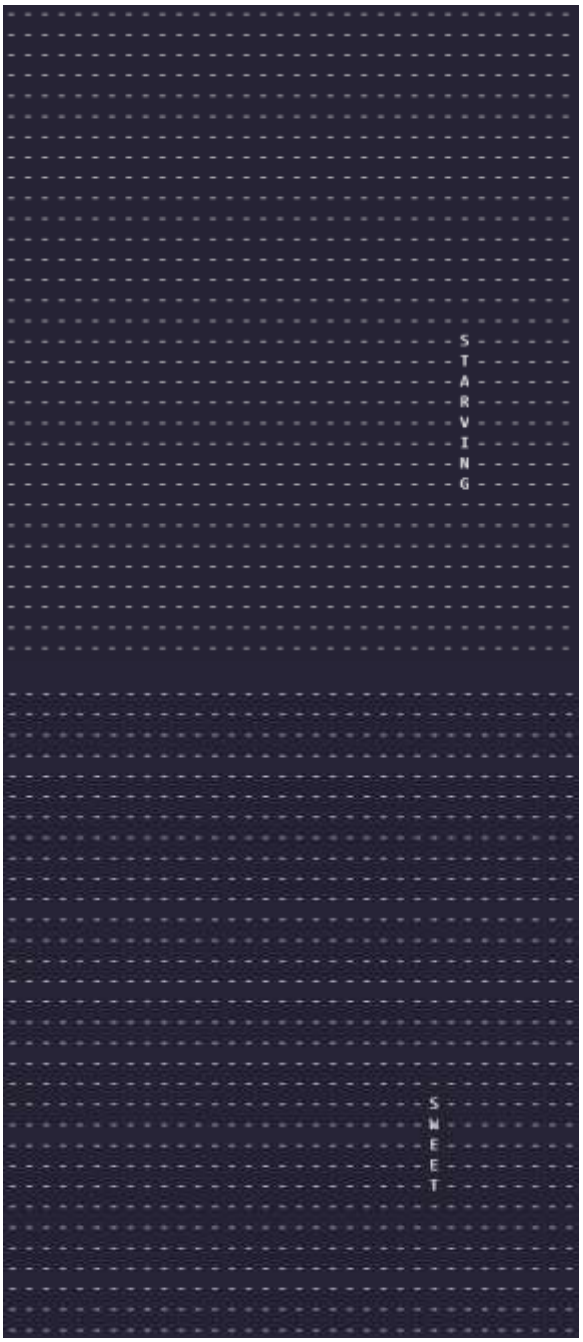

QUAINTLY

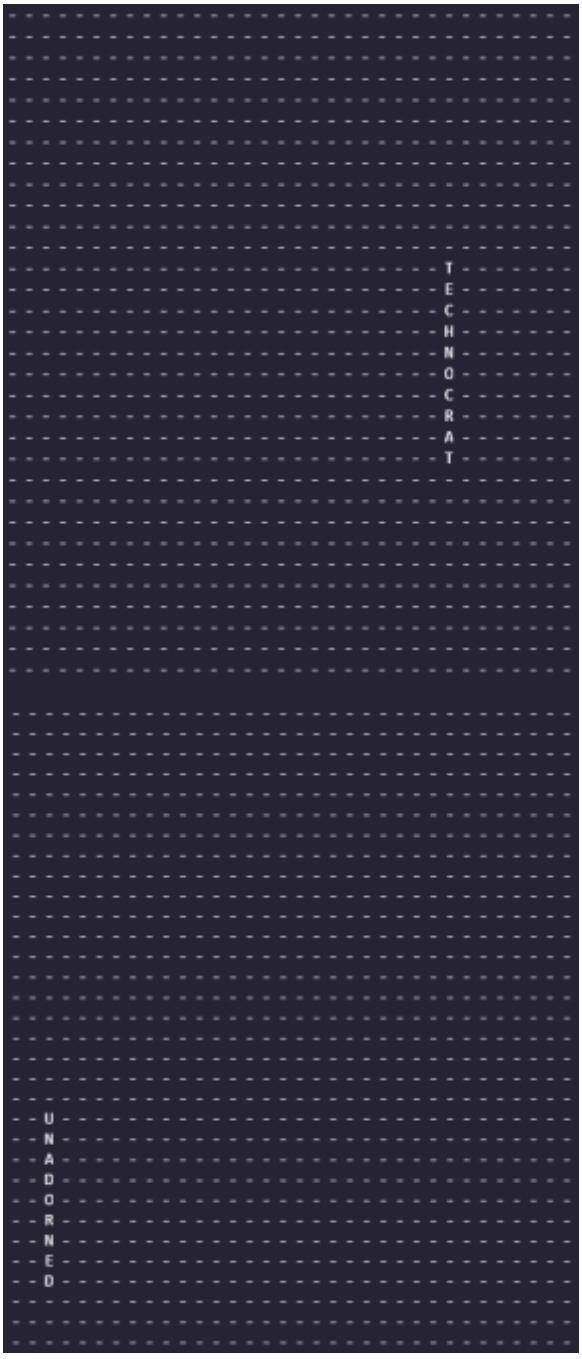

R
A
T
I
F
Y
I
N
G
R
E
D
U
N
D
A
N
C
Y

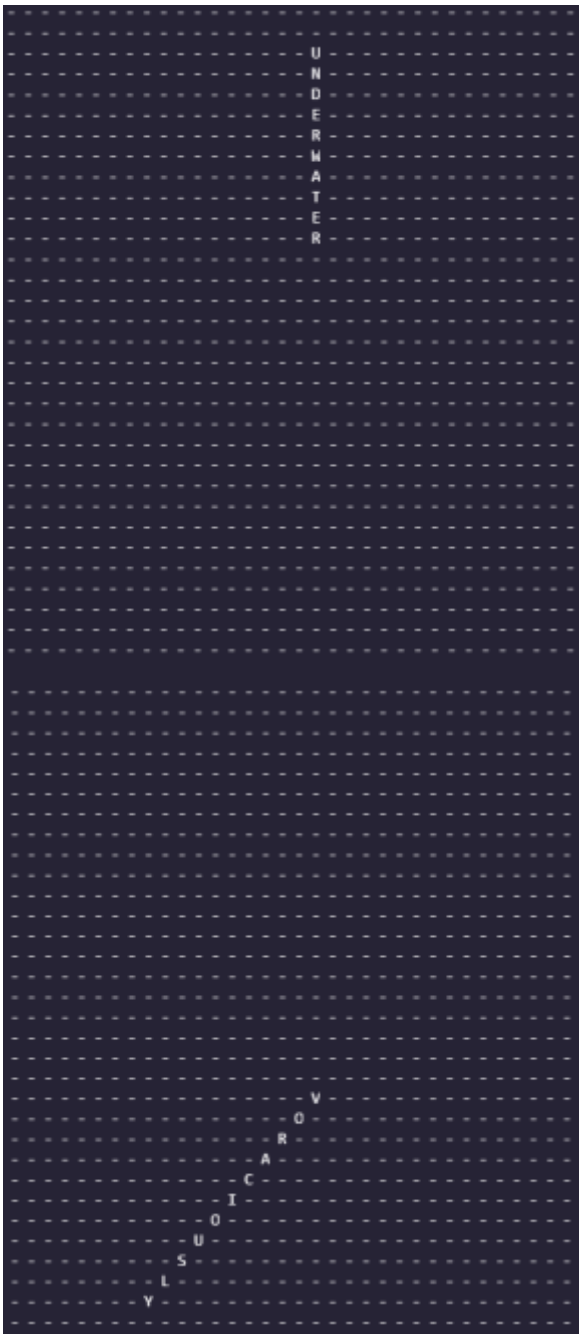

G
N
I
T
A
V
O
N
E
R
S
C
L
E
R
O
S
I
S

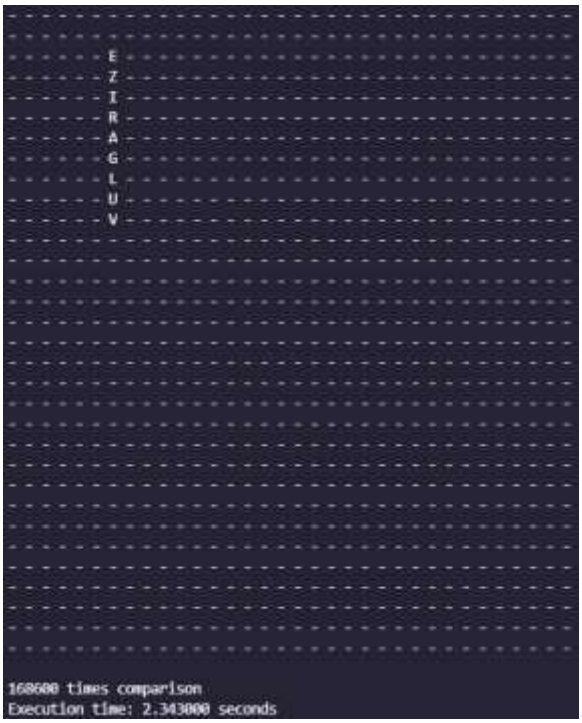

SCORNFUL
NOISSECES


GNIKNIS
SLENDER


STARVING
SWEET


TECHNOCRAT
UNADORNED





Pada screenshot hasil eksekusi puzzle large, tidak semua kata dapat ditampilkan karena keterbatasan terminal, sehingga pada hasil screenshot tidak terdapat kata-kata yang terletak di awal list. Saya terlambat membaca QnA, sehingga waktu eksekusi yang diperoleh adalah waktu pencarian dan pencetakan solusi (bukan waktu pencarian saja).

4. Kode program
   https://github.com/VionieNovencia/Word-Search-Puzzle-Solver

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