LAPORAN TUGAS KECIL 1 IF2211

Raden Haryosatyo Wisjnunandono

13520070

BAB I

ALGORITMA BRUTE FORCE

```
// Algoritma brute force utama ada di dalam fungsi matcher, fungsi ini menerima 2 parameter yakni vector (array dinamis) of char dari pattern
// dan teks dan mereturn index dari karakter pertama dari pattern yang sesuai dengan teks
int matcher(vector<char> P, vector<char> T){

// Mencari dan menyimpan informasi dari panjang pattern dan teks
    int m = P.size();
    int n = T.size();

//Inisialisasi iterator
    int i = 0;
    int j = 0;
    bool found = false;

//Mencari pattern dalam teks dengan iterasi satu-satu dari karakter teks yang indexnya 0 sampai ke
//indeks ke x dimana x adalah panjang teks - panjang pattern
    while ((i<=n-m) && not(found)){

//j adalah iterator yang mengiterasi tiap elemen dalam pattern
        j = 0;
        while ((j < m) && (P[j] == T[i+j])){
            j++;
        }

//jika j sudah sama dengan panjang pattern maka iterator sudah sampai ujung dan menandakan bahwa semua karakter dalam pattern ditemukan dalam
//teks secara sekuensial
        if (j == m){
            found = true;
        }
        else{
            i++;
        }
    }
//jika sudah ditemukan maka fungsi akan mereturn indeks karakter pertama ditemmukannya pattern dalam teks
    if (found){
        return i;
    }
    else{
        return -1;
    }
}
```

BAB II

SOURCE CODE

Disclaimer:

Untuk menghemat tempat, yang akan dilampirkan hanya file implementasinya saja (.cpp) untuk file header (.hpp) dapat dilihat pada link drive yang berisi source code yang lengkap

solver.cpp (algoritma brute force utama)

```cpp
#include <iostream>
#include <fstream>
#include <cstring>
#include <vector>
#include "solver.hpp"

using namespace std;

int matcher(vector<char> P, vector<char> T){
    int m = P.size();
    int n = T.size();
    int i = 0;
    int j = 0;
    bool found = false;

    while ((i<=n-m) && not(found)){
        j = 0;
        while ((j < m) && (P[j] == T[i+j])){
            j++;
        }
        if (j == m){
            found = true;
        }
        else{
            i++;
        }
    }

    if (found){
        return i;
    }
    else{
        return -1;
    }
}
```

fileinput.cpp (untuk membaca file input dalam folder ../test)

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <typeinfo>
#include "fileinput.hpp"
using namespace std;


void printmatrix(vector<vector<char>> mat)
{
  int row = mat.size();
  int col = mat[0].size();
  for (int i = 0; i < row ; i++)
  {
      if (mat[i].size() != 0){
          for (int j = 0 ; j < col ; j++)
       {
         cout<<mat[i][j];
         if (j != col-1)
         {
           cout<<" ";
         }
       }
       cout<<endl;
      }
   }
}
```

```cpp
void inputFile(string fileName) {

  string line;
  bool flag = false;
  fstream myfile ("../test/"+ fileName);
  while (!myfile.eof())
  {
    getline(myfile, line);
    vector<char> arr;
    if ((int)line[1] == 32)
    {
      for (int i = 0 ; i<line.size(); i=i+2)
      {
        arr.push_back(line[i]);
      }
      mat.push_back(arr);

    }

    else if ((int)line[1] >= 65 && (int)line[1] <=122 ){
      vector<char> patt;
      for (int i = 0 ; i<line.size(); i++)
      {
        if ((int)line[i] >= 65 && (int)line[i] <=122){
          patt.push_back(line[i]);
        }

      }
      listPatt.push_back(patt);

    }

  } mat.pop_back();

}
```

main.cpp (Program utama)

```cpp
#include <iostream>
#include <fstream>
#include "solver.hpp"
#include "fileinput.hpp"
#include <typeinfo>
#include <chrono>
#include <ctime>

using namespace std;

void printascii(vector<vector<char>> ex){
    for(auto x: ex){
        for(auto y: x){
            cout << (int)y << ' ';
        }
        cout << '\n';
    }
}

void printdefault(vector<char> ex){
    for(auto x: ex){
        cout << x;
    }
    cout << '\n';
}

vector<vector<char>> mat;
vector<vector<char>> listPatt;
int main(){
    string fileName;
    cout << "Masukkan nama file: ";
    cin >> fileName;
    cout << '\n';
    inputFile(fileName);

    vector<vector<char>> rlistPatt;
    for(int x = 0; x<listPatt.size(); x++){
        vector<char> rpatt;
        for (auto ir = listPatt[x].rbegin(); ir != listPatt[x].rend(); ++ir){
            rpatt.push_back(*ir);
        }
        rlistPatt.push_back(rpatt);

    }



    int counter = 0;
```

```
auto start = chrono::system_clock::now();
for (int j = 0; j < listPatt.size(); j++){
    // inisialisasi display matriks
    vector<vector<char>> disp;
    int row = mat.size();
    int col = mat[0].size();

    for (int i = 0; i < row ; i++)
    {
        vector<char> arrnew;
        for (int j = 0 ; j < col ; j++)
        {
            arrnew.push_back('-');
        }
        disp.push_back(arrnew);

    }
    int idx;

    //HORIZONTAL
    for (int i=0; i< mat.size(); i++){
        //KIRI KE KANAN
        // auto start = chrono::system_clock::now();
        idx = matcher(listPatt[j], mat[i]);
        // auto end = chrono::system_clock::now();
        // chrono::duration<double> elapsed_seconds = end-start;
        // chrono::duration<double> total_time = total_time + elapsed_seconds;
        counter++;

        if (idx != -1){
            for(int k = 0; k < listPatt[j].size(); k++){
                disp[i][idx] = listPatt[j][k];
                idx++;
            }
            printmatrix(disp);
            cout << '\n';
        }
        else{
            //kanan ke kiri
        //    auto start2 = chrono::system_clock::now();
           idx = matcher(rlistPatt[j], mat[i]);
            // auto efend = chrono::system_clock::now();
            // chrono::duration<double> elapsed_seconds = efend-start2;
            // chrono::duration<double> total_time = total_time + elapsed_seconds;
            counter++;

           if (idx != -1){
            for(int k = 0; k < rlistPatt[j].size(); k++){
                disp[i][idx] = rlistPatt[j][k];
                idx++;
            }
```

```
                printmatrix(disp);
                cout << '\n';
            }
            }
        }

        if (idx == -1){
        //VERTIKAL

        //inisialisasi vector of char veretical
            for (int i=0; i< mat[0].size(); i++){
                vector<char> vertical;
                for(int l=0; l<mat.size(); l++){
                    vertical.push_back(mat[l][i]);
                }
                // atas ke bawah
                // auto start = chrono::system_clock::now();
                idx = matcher(listPatt[j], vertical);
            // auto end = chrono::system_clock::now();
            // chrono::duration<double> elapsed_seconds = end-start;
            // chrono::duration<double> total_time = total_time + elapsed_seconds;
            counter++;
                if (idx != -1){
                    for(int k = 0; k < listPatt[j].size(); k++){
                        disp[idx][i] = listPatt[j][k];
                        idx++;
                    }
                    printmatrix(disp);
                    cout << '\n';
                }
                else {
                    // auto start2 = chrono::system_clock::now();
                    idx = matcher(rlistPatt[j], vertical);
                    // auto efend = chrono::system_clock::now();
                    // chrono::duration<double> elapsed_seconds = efend-start2;
                    // chrono::duration<double> total_time = total_time + elapsed_seconds;
                    counter++;
                    if (idx != -1){
                        for(int k = 0; k < rlistPatt[j].size(); k++){
                            disp[idx][i] = rlistPatt[j][k];
                            idx++;
                        }
                        printmatrix(disp);
                        cout << '\n';
                    }

                }
            }
        }
```

```
    if (idx == -1){
    //DIAGONAL

        //Diagonal kiri atas ke kanan bawah segitiga atas
        for(int ltc = 0; ltc < col-1; ltc++){
            vector<char> diagonalLR;
            int li = 0;
            int lj = ltc;
            while ((li < row) && (lj < col)){
                diagonalLR.push_back(mat[li][lj]);
                li++;
                lj++;
            }
            // auto start = chrono::system_clock::now();
            idx = matcher(rlistPatt[j], diagonalLR);
            // auto end = chrono::system_clock::now();
            // chrono::duration<double> elapsed_seconds = end-start;
            // chrono::duration<double> total_time = total_time + elapsed_seconds;
            counter++;
            if (idx != -1){
                for(int k = 0; k < rlistPatt[j].size(); k++){
                    disp[0+idx][ltc+idx] = rlistPatt[j][k];
                    idx++;
                }
                printmatrix(disp);
                cout << '\n';
            }
            else{
                // auto start2 = chrono::system_clock::now();
                idx = matcher(listPatt[j], diagonalLR);
                // auto efend = chrono::system_clock::now();
                // chrono::duration<double> elapsed_seconds = efend-start2;
                // chrono::duration<double> total_time = total_time + elapsed_seconds;
                counter++;
                if (idx != -1){
                    for(int k = 0; k < listPatt[j].size(); k++){
                        disp[0+idx][ltc+idx] = listPatt[j][k];
                        idx++;
                    }
                    printmatrix(disp);
                    cout << '\n';
                }

            }
        }
    }
```

```
if (idx == -1){
//Diagonal kiri atas ke kanan bawah segitiga bawah
for(int ltc = 0; ltc < row-1; ltc++){
    vector<char> diagonalLR;
    int li = ltc;
    int lj = 0;
    while ((li < row) && (lj < col)){
        diagonalLR.push_back(mat[li][lj]);
        li++;
        lj++;
    }
    // auto start = chrono::system_clock::now();
    idx = matcher(rlistPatt[j], diagonalLR);
    // auto end = chrono::system_clock::now();
    // chrono::duration<double> elapsed_seconds = end-start;
    // chrono::duration<double> total_time = total_time + elapsed_seconds;
    counter++;
    if (idx != -1){
        for(int k = 0; k < rlistPatt[j].size(); k++){
            disp[ltc+idx][0+idx] = rlistPatt[j][k];
            idx++;
        }
        printmatrix(disp);
        cout << '\n';
    }
    else{
        // auto start2 = chrono::system_clock::now();
        idx = matcher(listPatt[j], diagonalLR);
        // auto efend = chrono::system_clock::now();
        // chrono::duration<double> elapsed_seconds = efend-start2;
        // chrono::duration<double> total_time = total_time + elapsed_seconds;
        counter++;
        if (idx != -1){
            for(int k = 0; k < listPatt[j].size(); k++){
                disp[ltc+idx][0+idx] = listPatt[j][k];
                idx++;
            }
            printmatrix(disp);
            cout << '\n';
        }

    }
}
}
```

```
    if (idx == -1){
    //Diagonal kanan atas ke kiri bawah segitiga atas
    for(int rtc = 1; rtc < col; rtc++){
        vector<char> diagonalLR;
        int li = 0;
        int lj = rtc;
        while ((li < row) && (lj >= 0)){
            diagonalLR.push_back(mat[li][lj]);
            li++;
            lj--;
        }
        // auto start = chrono::system_clock::now();
        idx = matcher(rlistPatt[j], diagonalLR);
        // auto end = chrono::system_clock::now();
        // chrono::duration<double> elapsed_seconds = end-start;
        // chrono::duration<double> total_time = total_time + elapsed_seconds;
        counter++;
        if (idx != -1){
            for(int k = 0; k < rlistPatt[j].size(); k++){
                disp[0+idx][rtc-idx] = rlistPatt[j][k];
                idx++;
            }
            printmatrix(disp);
            cout << '\n';
        }
        else{
            // auto start2 = chrono::system_clock::now();
            idx = matcher(listPatt[j], diagonalLR);
            // auto efend = chrono::system_clock::now();
            // chrono::duration<double> elapsed_seconds = efend-start2;
            // chrono::duration<double> total_time = total_time + elapsed_seconds;
            counter++;
            if (idx != -1){
                for(int k = 0; k < listPatt[j].size(); k++){
                    disp[0+idx][rtc-idx] = listPatt[j][k];
                    idx++;
                }
                printmatrix(disp);
                cout << '\n';
            }

        }
    }
    }
```

```
    if (idx == -1){
    //Diagonal kanan atas ke kiri bawah segitiga bawah
    for(int ltc = 0; ltc < row-1; ltc++){
        vector<char> diagonalLR;
        int li = ltc;
        int lj = col-1;
        while ((li < row) && (lj >= 0)){
            diagonalLR.push_back(mat[li][lj]);
            li++;
            lj--;
        }
        // auto start = chrono::system_clock::now();
        idx = matcher(rlistPatt[j], diagonalLR);
        // auto end = chrono::system_clock::now();
        // chrono::duration<double> elapsed_seconds = end-start;
        // chrono::duration<double> total_time = total_time + elapsed_seconds;
        counter++;
        if (idx != -1){
            for(int k = 0; k < rlistPatt[j].size(); k++){
                disp[ltc+idx][col-1-idx] = rlistPatt[j][k];
                idx++;
            }
            printmatrix(disp);
            cout << '\n';
        }
        else{
            // auto start2 = chrono::system_clock::now();
            idx = matcher(listPatt[j], diagonalLR);
            // auto efend = chrono::system_clock::now();
            // chrono::duration<double> elapsed_seconds = efend-start2;
            // chrono::duration<double> total_time = total_time + elapsed_seconds;
            counter++;
            if (idx != -1){
                for(int k = 0; k < listPatt[j].size(); k++){
                    disp[ltc+idx][col-1-idx] = listPatt[j][k];
                    idx++;
                }
                printmatrix(disp);
                cout << '\n';
            }

        }
    }
    }
```

```
    auto end = chrono::system_clock::now();
    chrono::duration<double> elapsed_seconds = end-start;
    cout << "total perbadingan: " << counter << endl;
    cout << "elapsed time: " << elapsed_seconds.count() << "s\n";
    return 0;
}
```

BAB III

INPUT DAN OUTPUT

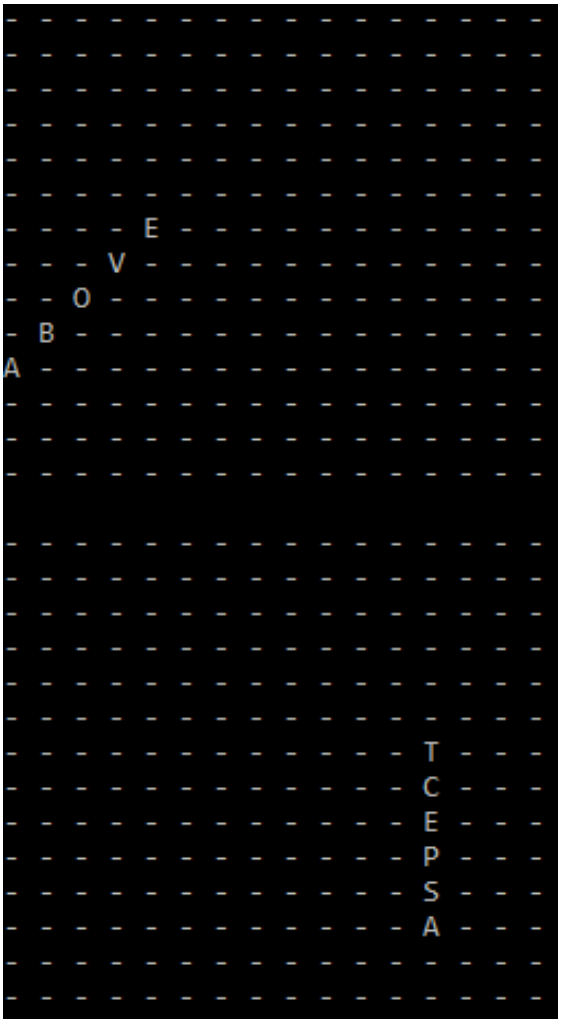
1. small1.txt

Input:

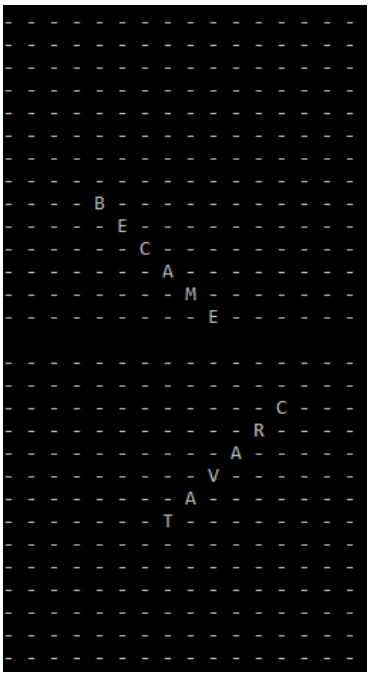
```
P L U T O F Q X P G N O S X N D
M L X Y E G F S Y Q B Q A O I O
L B Z D P B E R I I Q K C V L F
R E M N U L A I J D V R I T X Q
Q N T T Y M J C X V A N L C W T
B K Y E L T T I L V E B E F D P
S H I N E D Q P A R T X T R H V
J Z U V F E K T E N P F C Y S E
J H O D B F B U T W U A E L H L
U B L M T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

ABOVE
ASPECT
BECAME
CRAVAT
DIVINE
LITTLE
LOST
MOLE
PLUTO
SHADY
SHINE
SONG
STUN
UPSHOT
```

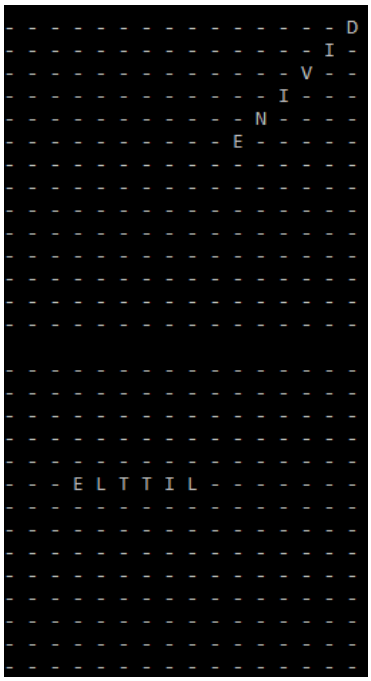
Output:

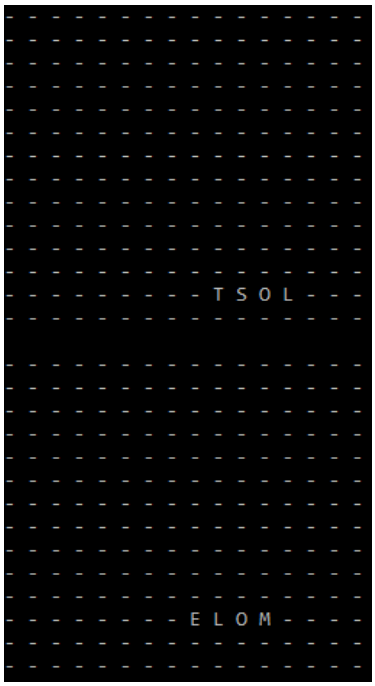
```
total perbadingan: 2232
elapsed time: 5.38798s
```

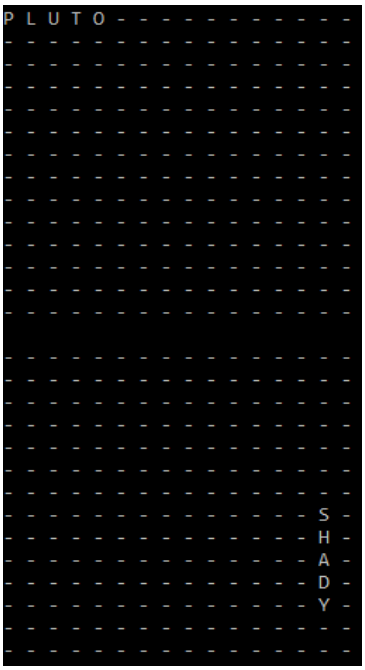


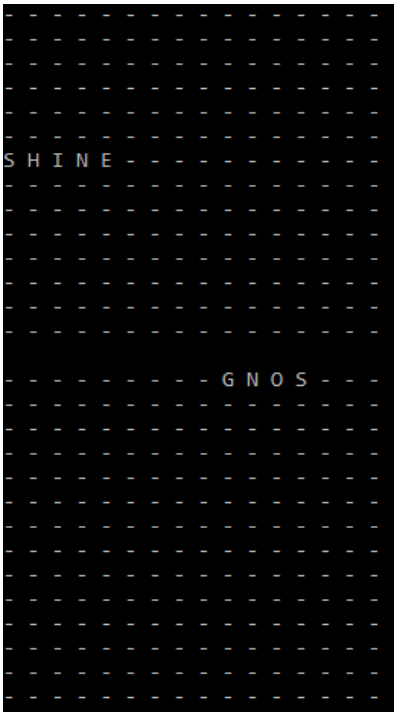
Output:

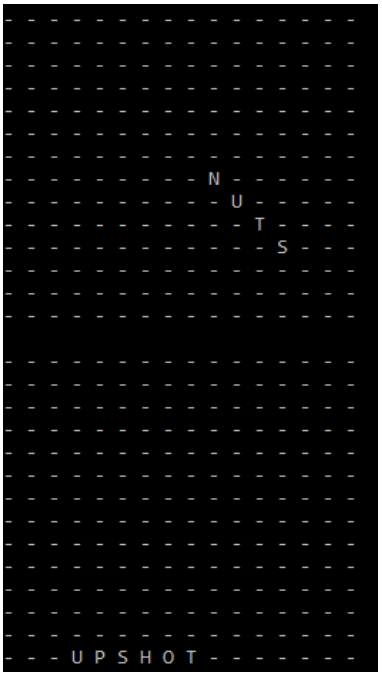







PLUTO
SHADY


SHINE
GNOS


NUTS
UPSHOT

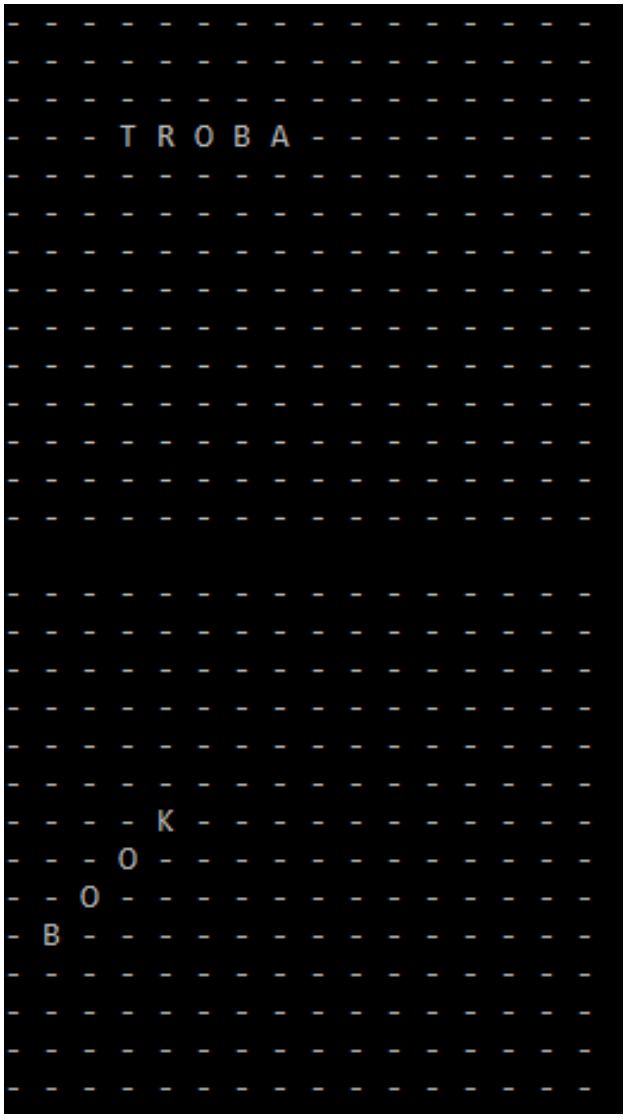
2. small_2.txt

Input:

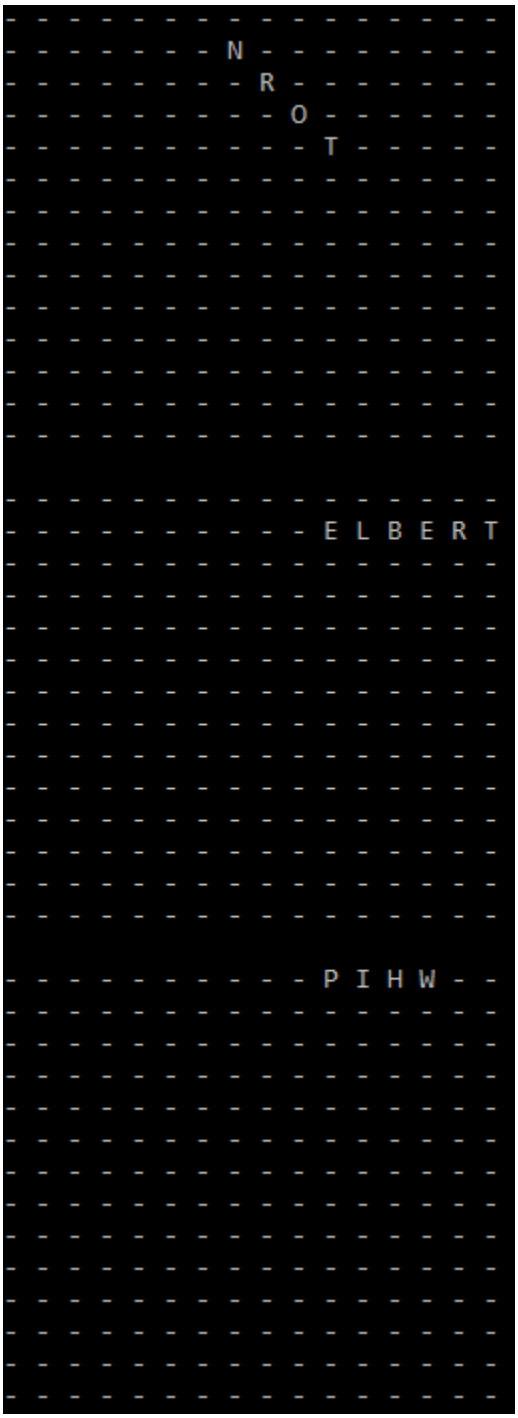
```
X D M M P V P M M M P I H W Q I
J B U D A I H N C J E L B E R T
U S T C V J J J R X X L T B F O
X A C T R O B A D O K A B J J S
H U I H V U B X E L T T J X V Q
I C D T E E R U P E R E Z D E K
I Y B Q K B N N K S I M H M N O
Q A X O Z O T T Y X N V A Y A Z
M G O J T O P W V M G N W T H G
F B G R C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

ABORT
BOOK
BOOST
DICTUM
DIRTY
METAL
OATS
PUREE
RINGED
SAUCY
SEXISM
TORN
TREBLE
WHIP
```

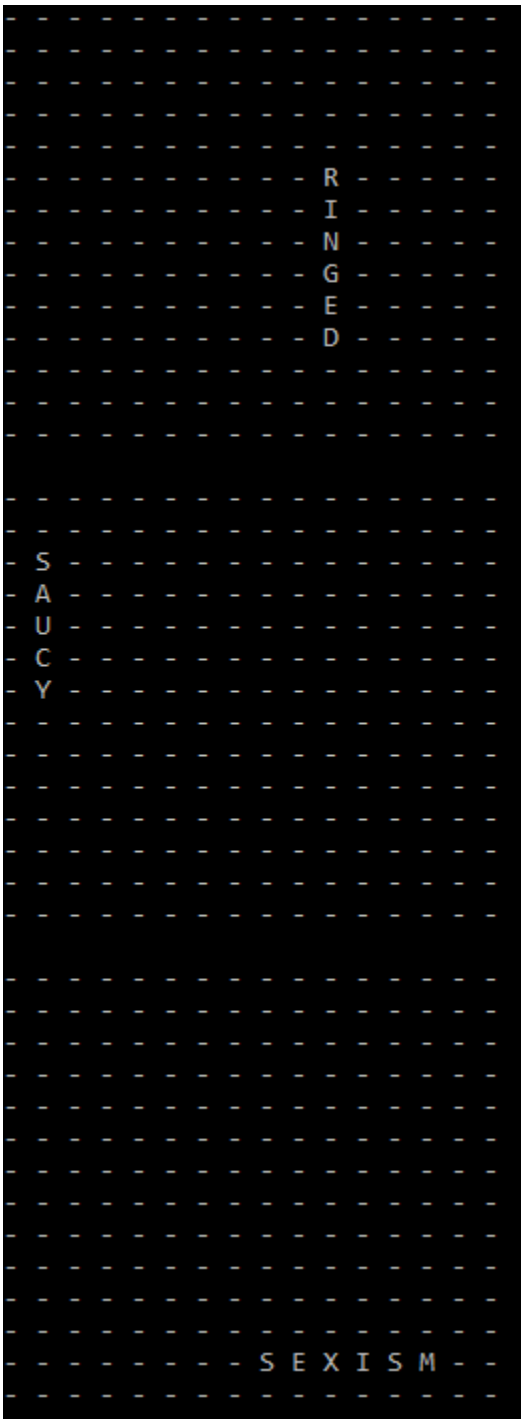
Output:

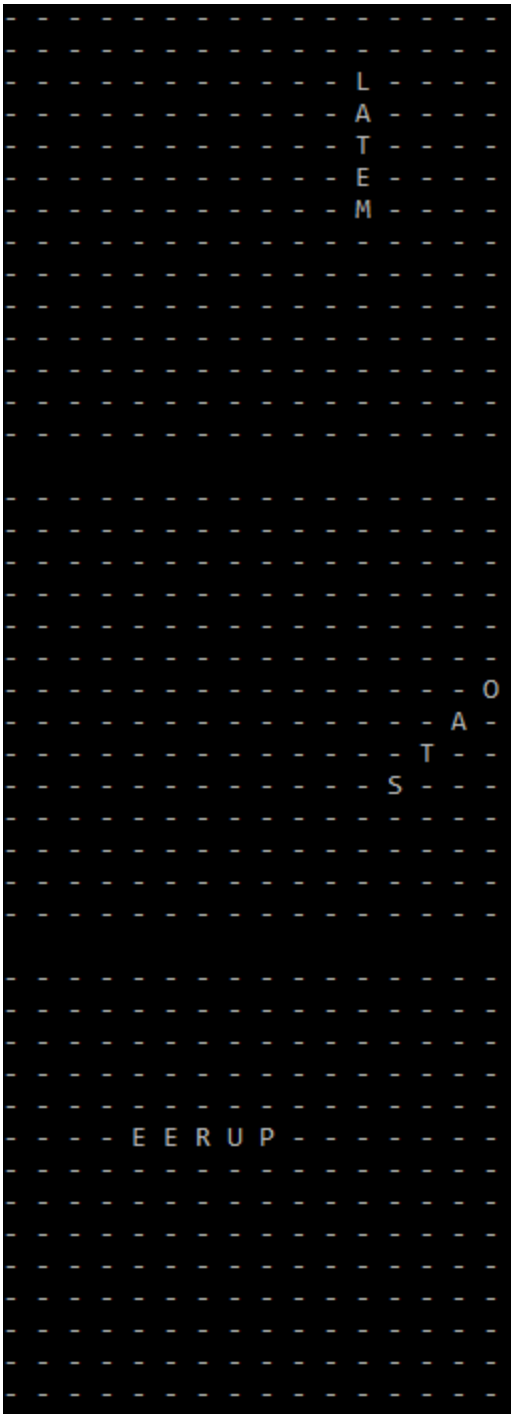
```
total perbadingan: 2401
elapsed time: 4.17139s
```

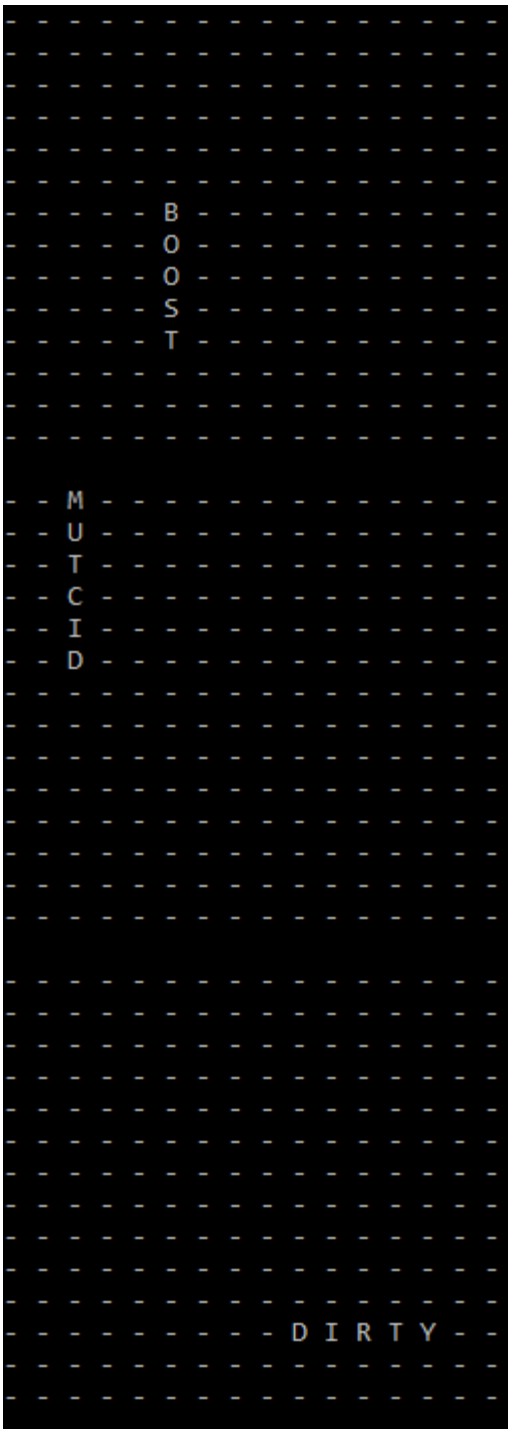


Output:






LATEM
OAT
S
EERUP


BOOST
MUTCID
DIRTY

3. small_3.txt

Input

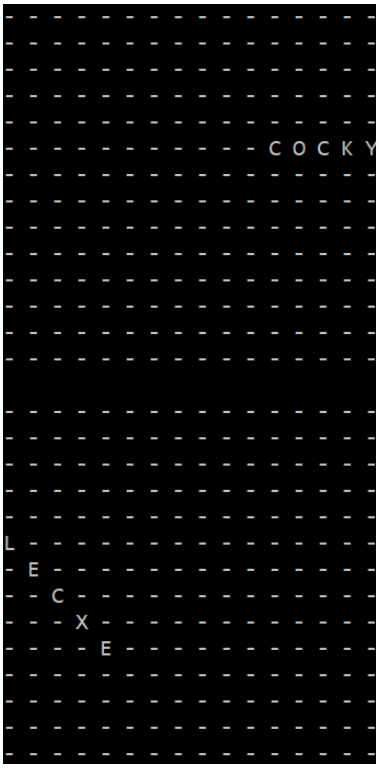
```
P F Y W W K M Z E X O L Q S B H
G K E X S S X A K K S X I K O J
T N T T T D F H I U R X J N C X
A F N R L R C C I N B A J H S D
K B R S F S Q T R O E C O B Z M
L L O A M Z E E B U T C O C K Y
S E N R S J G K B V L E B I U F
I M C A L N N P T L K J V Y Q Z
B O W X I K H H I A P L A C O F
K A A L E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

ANEW
BRINE
COCKY
EXCEL
FOCAL
IDLING
JOWL
LOAM
MAINE
ROME
SUITE
THRILL
TOXIC
TUBE
```

Output:

```
total perbadingan: 2401
elapsed time: 5.93424s
```

```
- - - W - - - - - - - - - - - - -
- - E - - - - - - - - - - - - - -
- N - - - - - - - - - - - - - - -
A - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - -
- - - - - - - - E - - - - - - - -
- - - - - - - - N - - - - - - - -
- - - - - - - - I - - - - - - - -
- - - - - - - - R - - - - - - - -
- - - - - - - - B - - - - - - - -
```

Output:



4. medium_1.txt

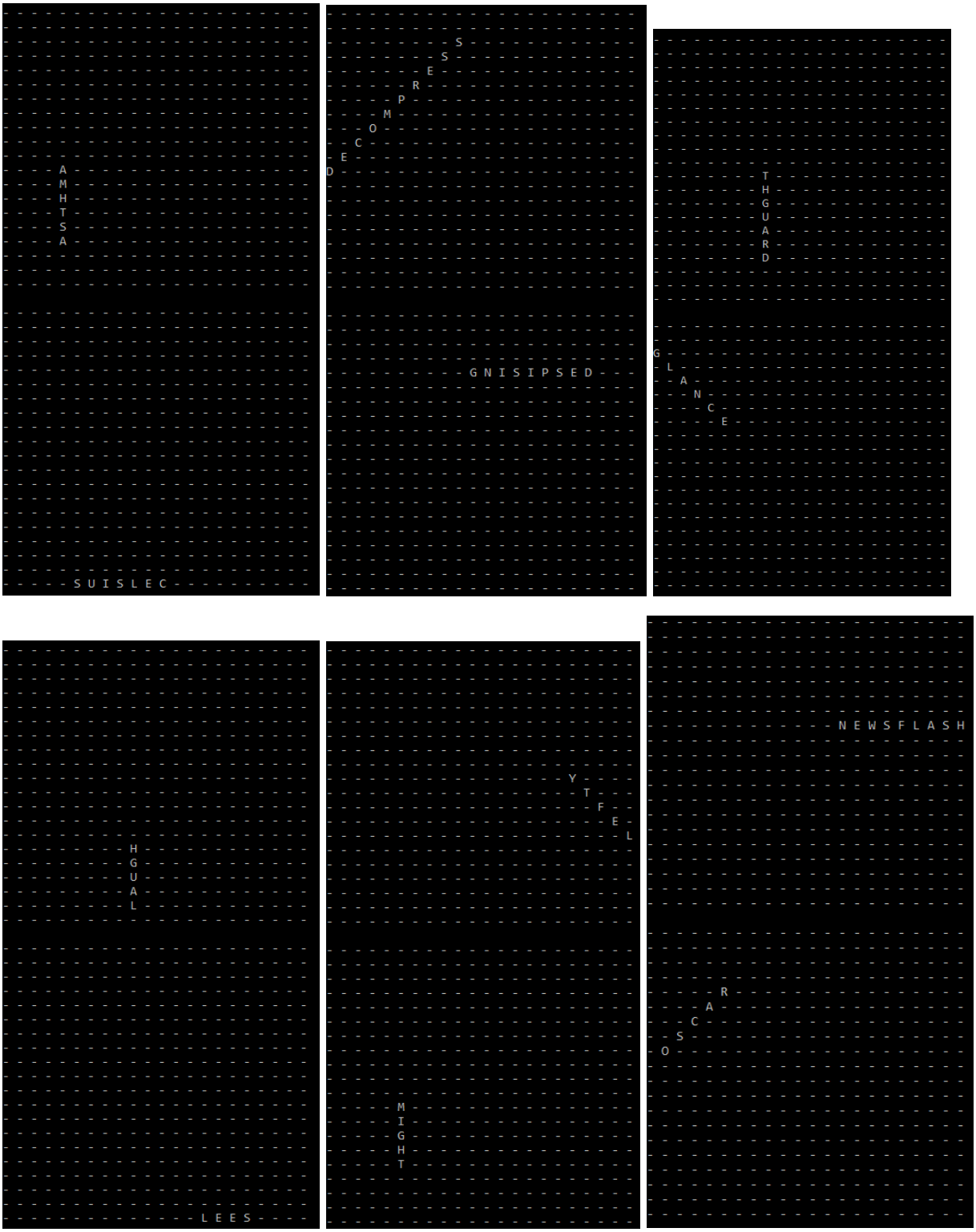
Input:

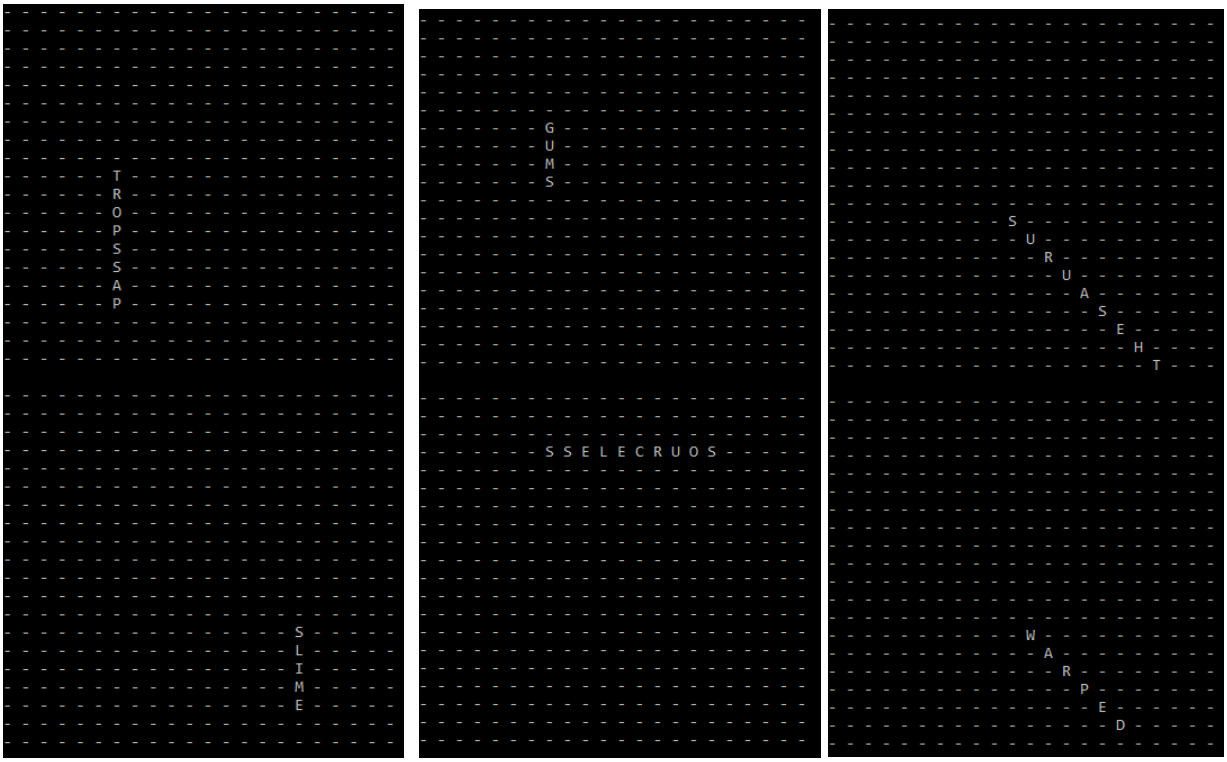
```
O H E C W W T E I C B Q V O L K H M N D I T
A S S E R T I N G I V Y W S H Y D B U N P Y
G B W R I C N F H S X O R C M J M U N J R N
P L K A S C Q S S E L E C R U O S W M T M J
U O A J Y R Q E E J G N I S I P S E D D T Y
P A I N A B R Q D W W P S W A D S Q F B E Q
I I J C C P R G U U N P U X D Q K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

Output:

```
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - A - - - - -
- - - - - - - - - - - - - - - - - - B - - - - -
- - - - - - - - - - - - - - - - - - - A - - - -
- - - - - - - - - - - - - - - - - - - - S - - -
- - - - - - - - - - - - - - - - - - - - - H - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - -
A S S E R T I N G - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - -
```

Output:


AMHTSA
SUISLEC
SSERPMOCED
GNISIPSED
THGUARD
GLANCE
HGUAL
LEES
YTFEL
MIGHT
NEWSFLASH
RACSO



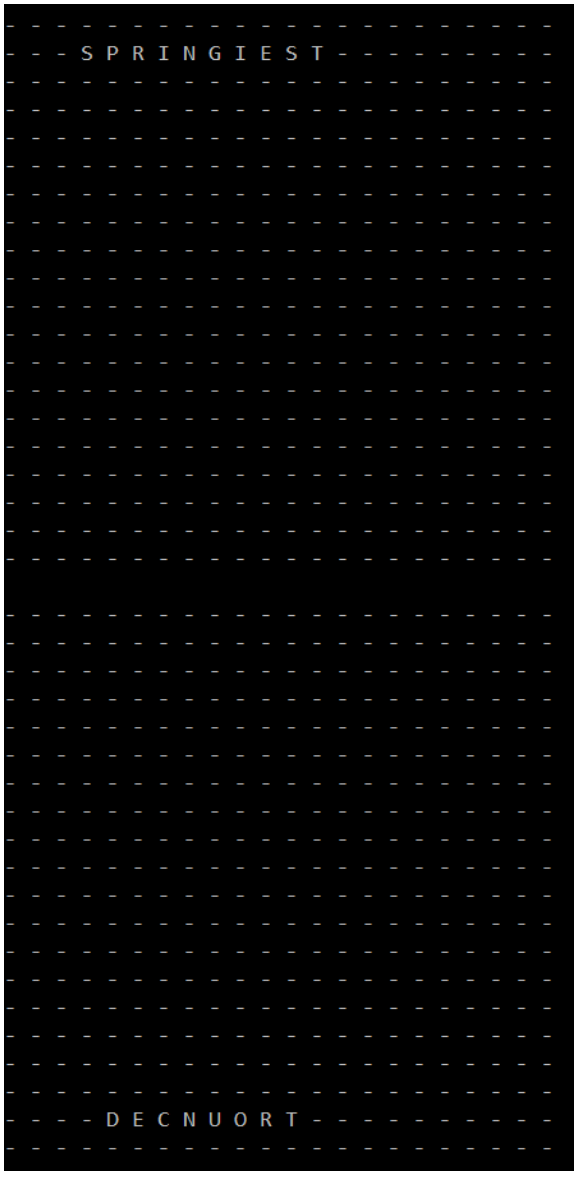

```
total perbadingan: 4463
elapsed time: 10.9923s
```

5. medium_2.txt

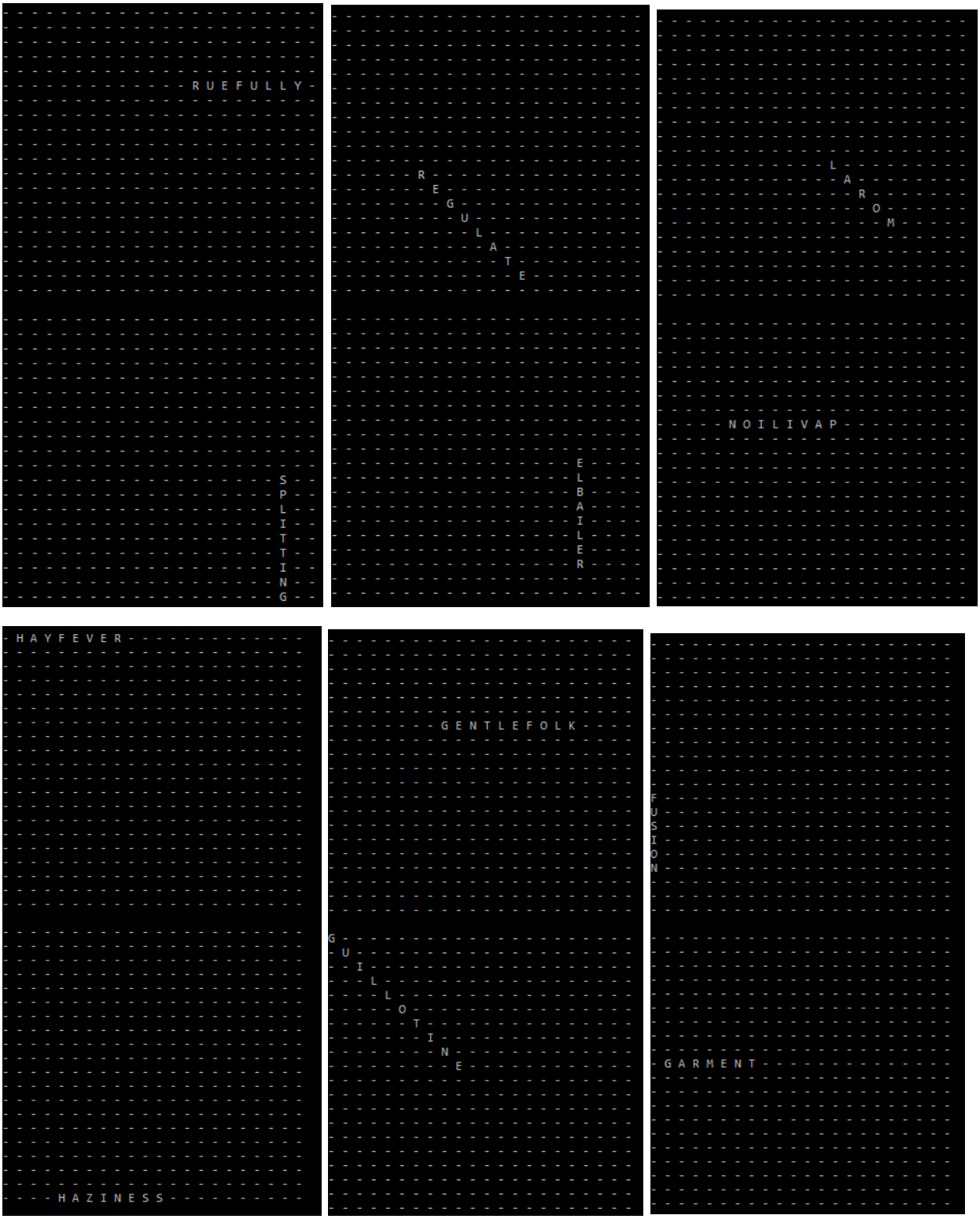
Input:

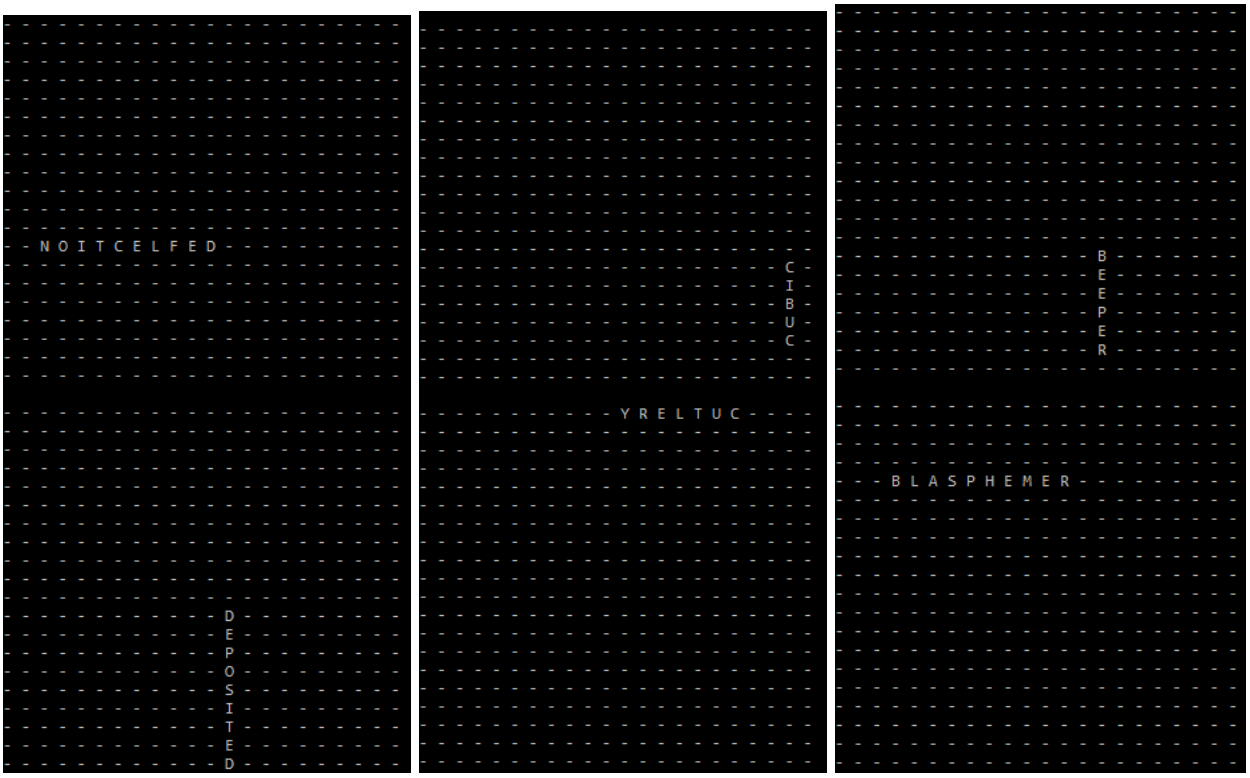
```
G H A Y F E V E R U L Y R E L T U C A F K E
X U N S P R I N G I E S T G T X F A U F O U
W I I O Z W Q F V J J A O K D I X F Y D W X
L F Q L O Q L A W D G H R O Y P U K V T D N
A A T B L A S P H E M E R I X M T B Q A R Z
N S G V Q O P G V L T J A R U E F U L L Y M
Q K J A H P T W G E N T L E F O L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

BEEPER
BLASPHEMER
CUBIC
CUTLERY
DEFLECTION
DEPOSITED
FUSION
GARMENT
GENTLEFOLK
GUILLOTINE
HAYFEVER
HAZINESS
MORAL
PAVILION
REGULATE
RELIABLE
RUEFULLY
SPLITTING
SPRINGIEST
TROUNCED
```

Output:



Output:


RUEFULLY
SPLITTING
REGULATE
ELBAILER
LARROM
NOILIVAP
HAYFEVER
HAZINESS
GENTLEFOLK
GUILLOTINE
FUSION
GARMENT




```
total perbadingan: 4664
elapsed time: 9.92685s
```

6. medium_3.txt

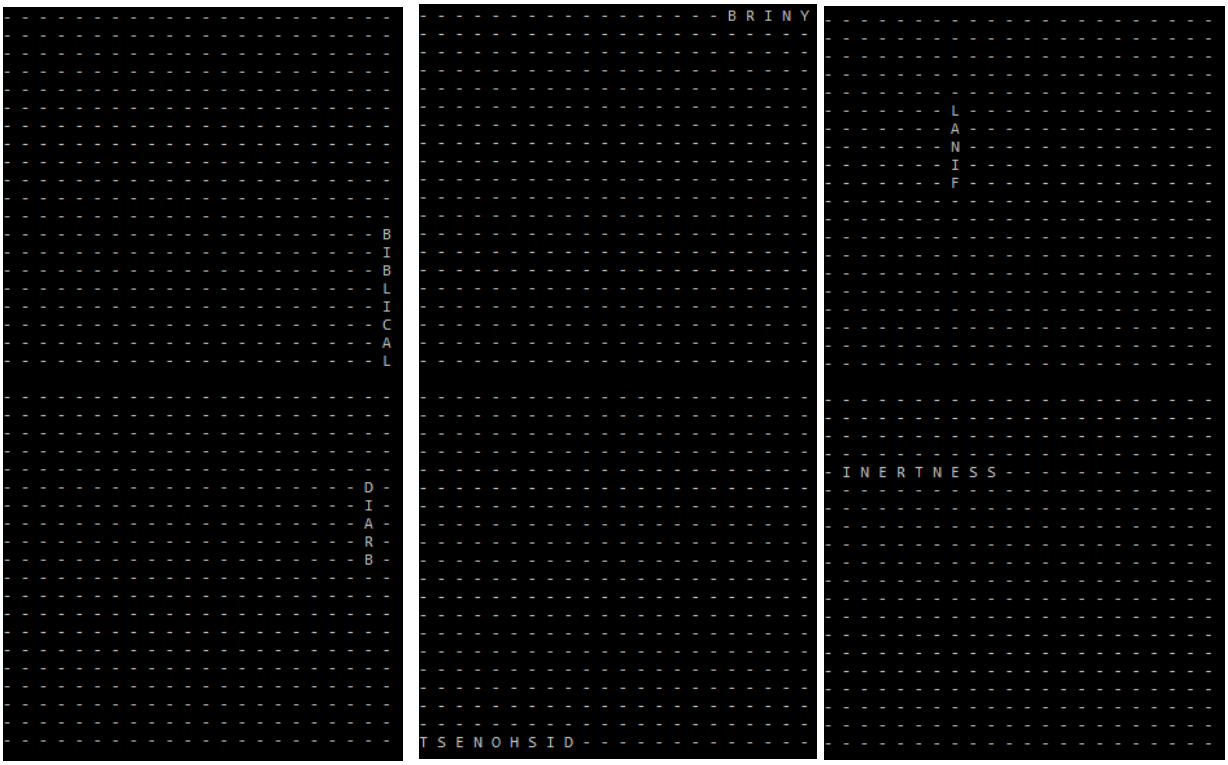
Input:

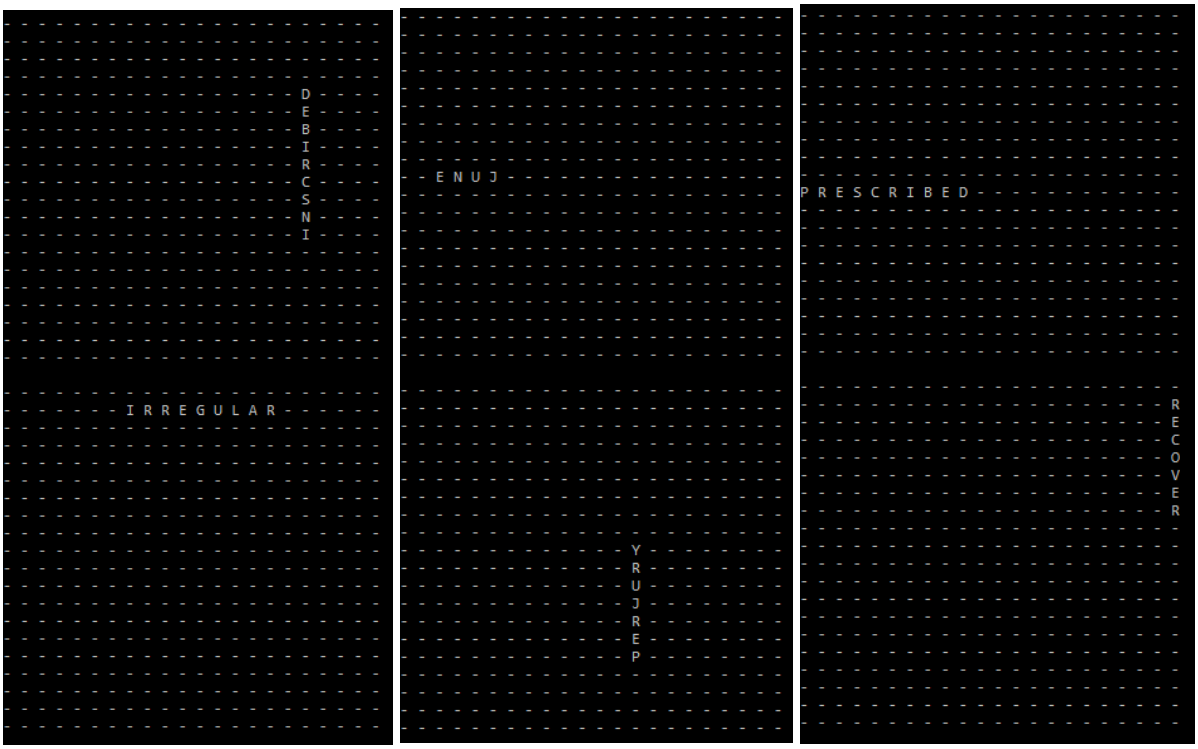
```
D Z N F B F F F F Y P T H R U X Y B R I N Y
V G C B X D X I R R E G U L A R J Y J I R R
X J L R G M N U U T M V O H J B X T J H F E
I B W G K D W P U S O L I D F L T L Q U E C
I I N E R T N E S S Z E S F W B T D A D C O
O O N F C H B L K D G C V N G Z X E D H D V
S M H I N V N A B V O N C N Y S M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

ADHERENCE
AMUSE
BIBLICAL
BRAID
BRINY
DISHONEST
FINAL
INERTNESS
INSCRIBED
IRREGULAR
JUNE
PERJURY
PRESCRIBED
RECOVER
SACKING
SHOWER
SOLID
SPATULA
TOTALLY
URUGUAY
```

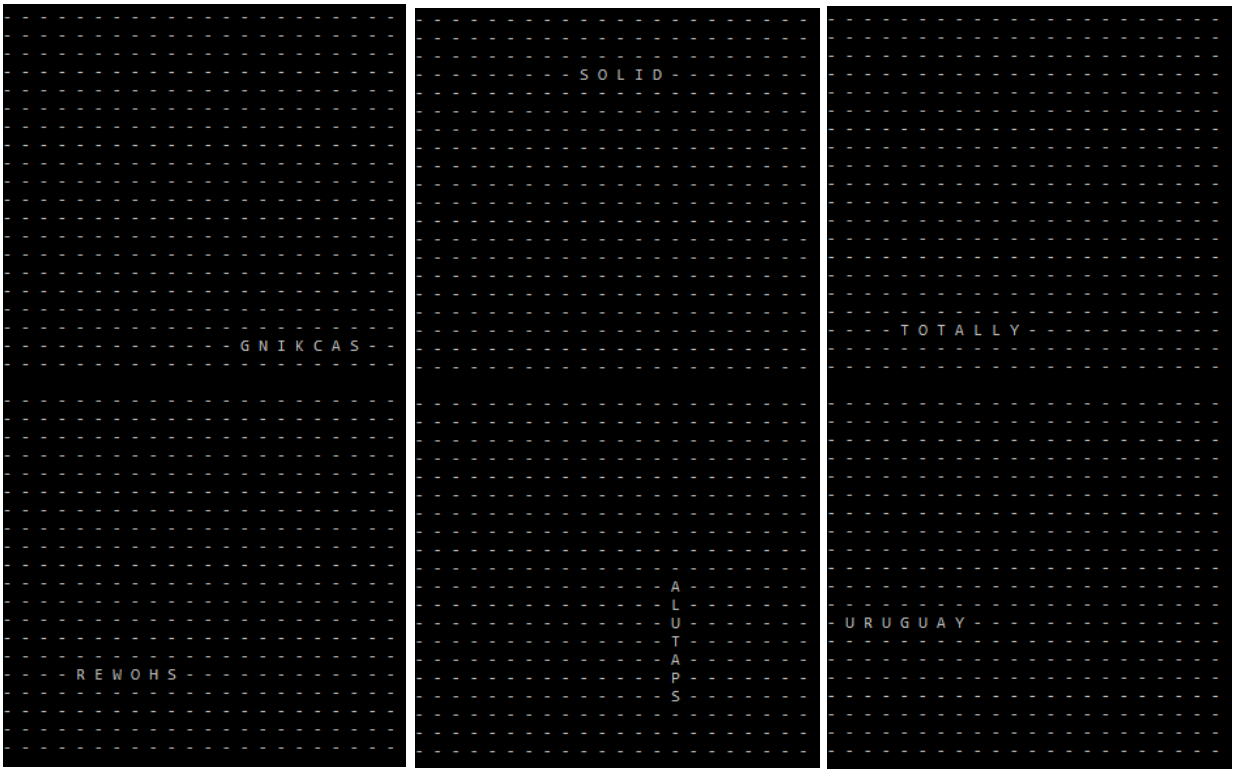
Output:

```
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - E - - - - - - - - - -
- - - - - - - - - - - C - - - - - - - - - -
- - - - - - - - - - - N - - - - - - - - - -
- - - - - - - - - - - E - - - - - - - - - -
- - - - - - - - - - - R - - - - - - - - - -
- - - - - - - - - - - E - - - - - - - - - -
- - - - - - - - - - - H - - - - - - - - - -
- - - - - - - - - - - D - - - - - - - - - -
- - - - - - - - - - - A - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - E S U M A - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - -
```

Output:


BILLICAL
DIARB
BRINY
TSENOHSID
LANIF
INERTNESS

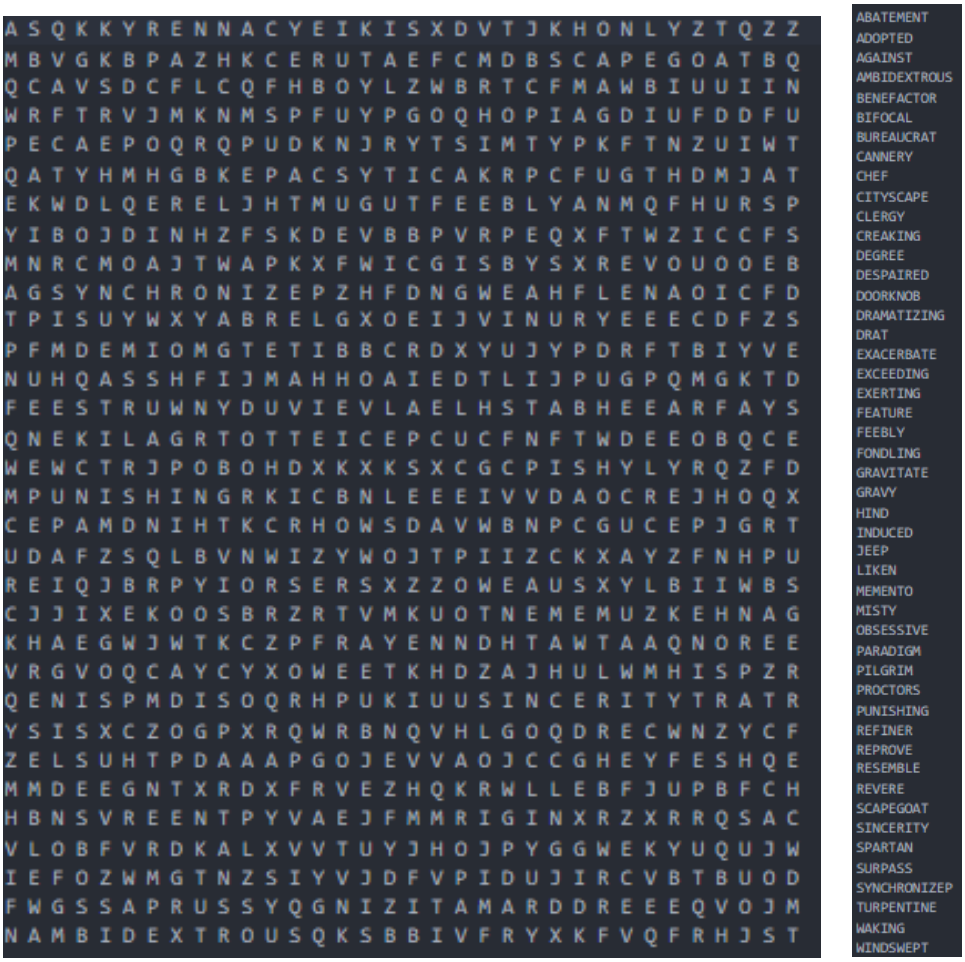

DEBIRCSNI
IRREGULAR
ENUJ
YRUJREP
PRESCRIBED
RECOVER




```
total perbadingan: 4347
elapsed time: 9.72706s
```

7. large_1.txt

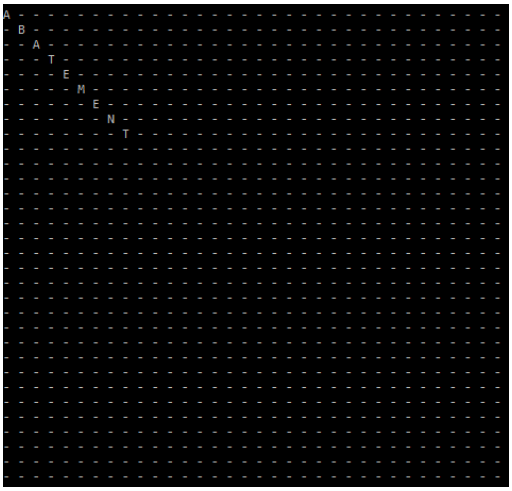
Input:



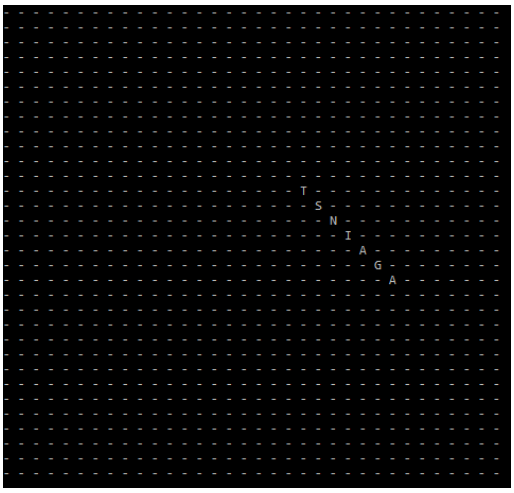
```
A S Q K K Y R E N N A C Y E I K I S X D V T J K H O N L Y Z T Q Z Z
M B V G K B P A Z H K C E R U T A E F C M D B S C A P E G O A T B Q
Q C A V S D C F L C Q F H B O Y L Z W B R T C F M A W B I U U I I N
W R F T R V J M K N M S P F U Y P G O Q H O P I A G D I U F D D F U
P E C A E P O Q R Q P U D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

```
ABATEMENT
ADOPTED
AGAINST
AMBIDEXTROUS
BENEFACTOR
BIFOCAL
BUREAUCRAT
CANNERY
CHEF
CITYSCAPE
CLERGY
CREAKING
DEGREE
DESPAIRED
DOORKNOB
DRAMATIZING
DRAT
EXACERBATE
EXCEEDING
EXERTING
FEATURE
FEEBLY
FONDLING
GRAVITATE
GRAVY
HIND
INDUCED
JEEP
LIKEN
MEMENTO
MISTY
OBSESSIVE
PARADIGM
PILGRIM
PROCTORS
PUNISHING
REFINER
REPROVE
RESEMBLE
REVERE
SCAPEGOAT
SINCERITY
SPARTAN
SURPASS
SYNCHRONIZEP
TURPENTINE
WAKING
WINDSWEPT
```

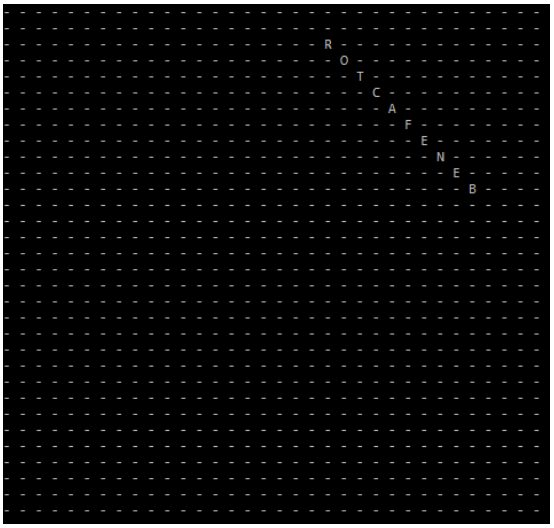
Output:

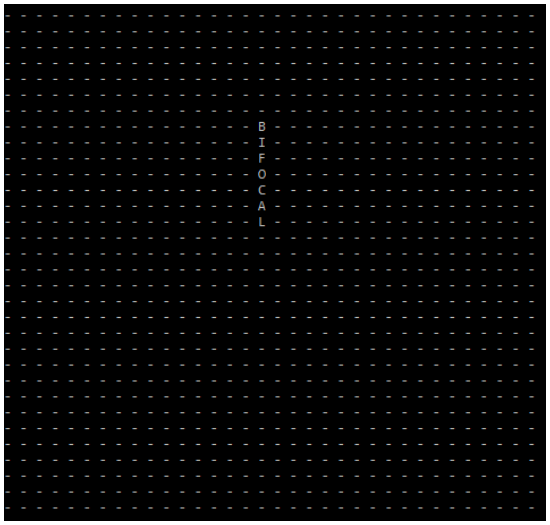

ABATEMENT

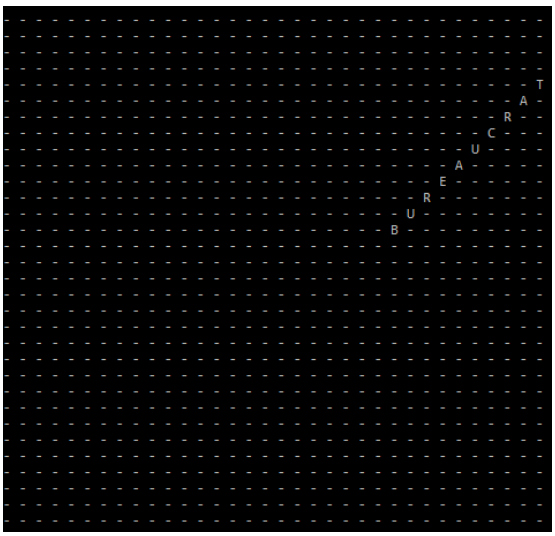

ADOPTED

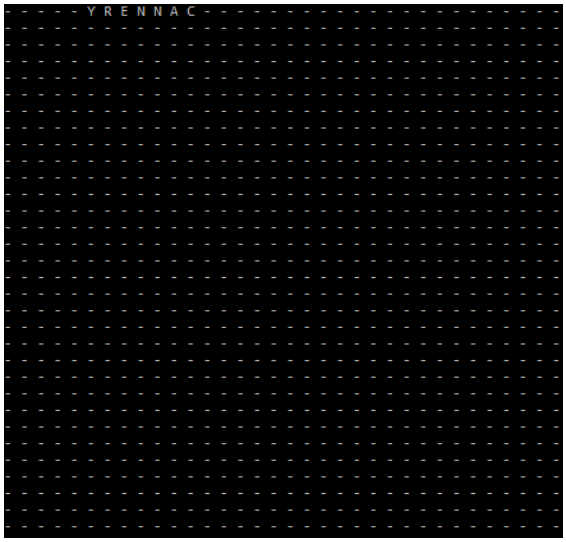

TSUNAMI

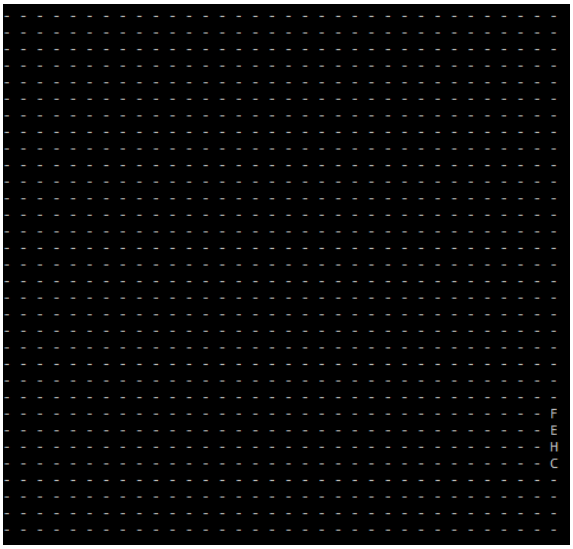

AMBIDEXTROUS


ROTCAFFEINE


BIFOCAL

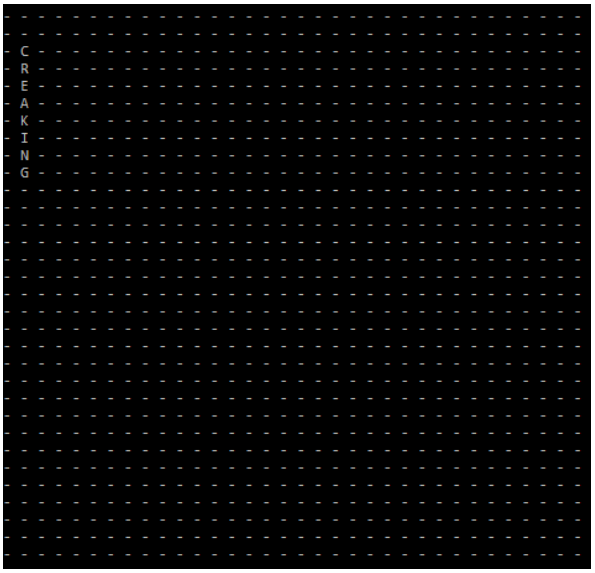

TAR
CUA
ERU
B

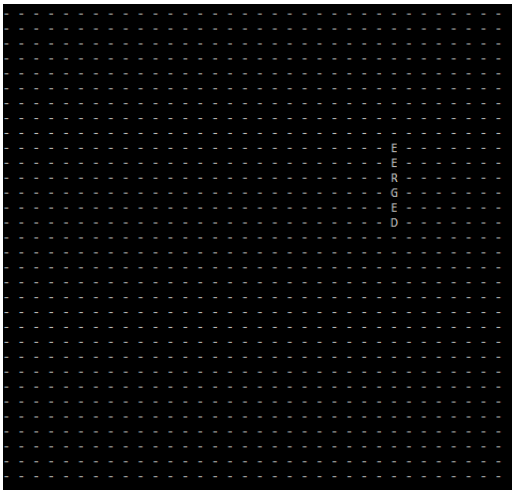

YRENNAC

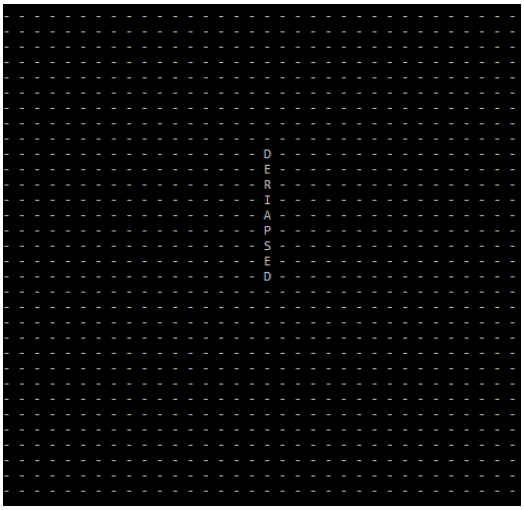

FEHC

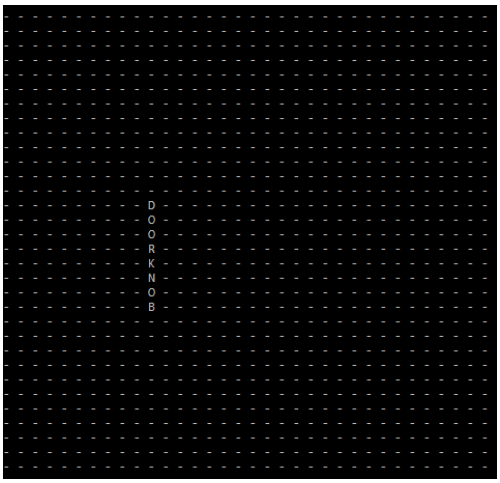

EPACSYTIC

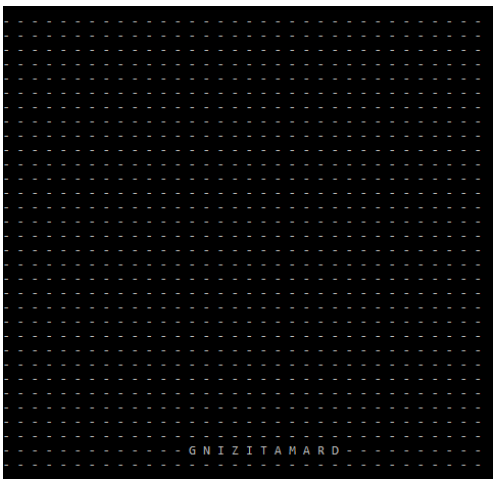

CLEARY

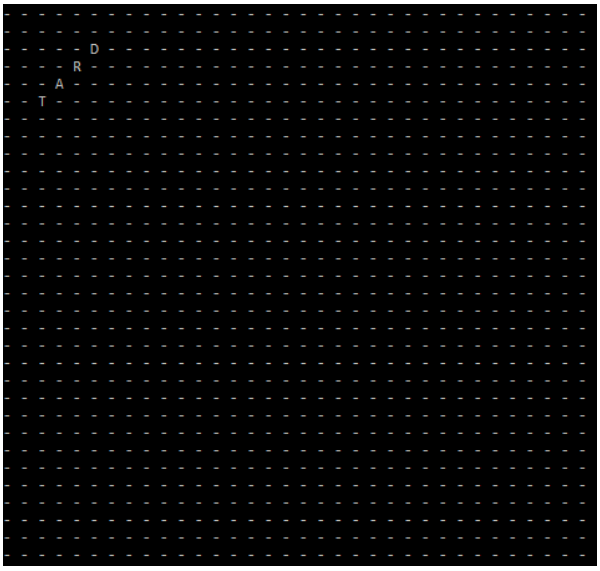

CREAKING

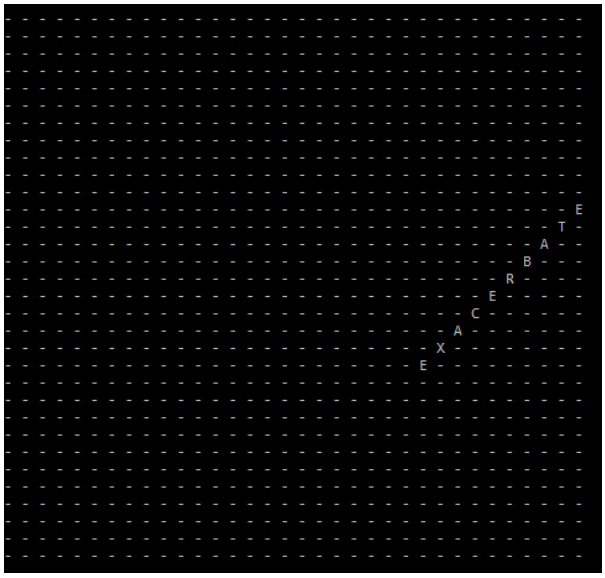

EERGED

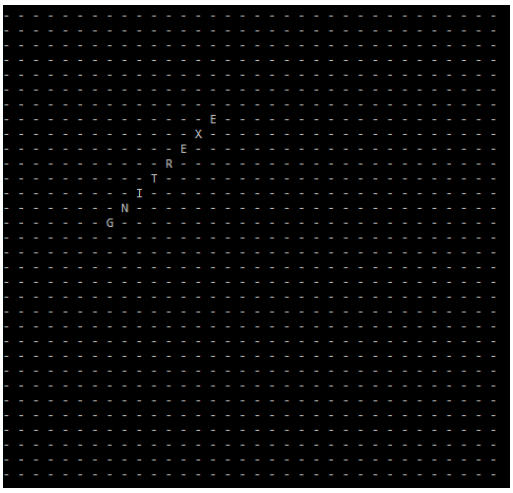

DERIAPSED

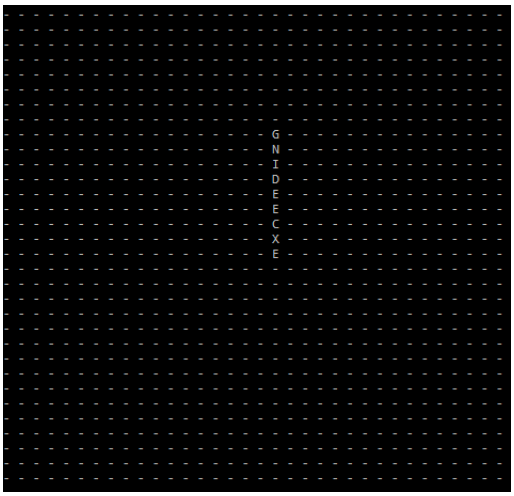

DOORKNOB

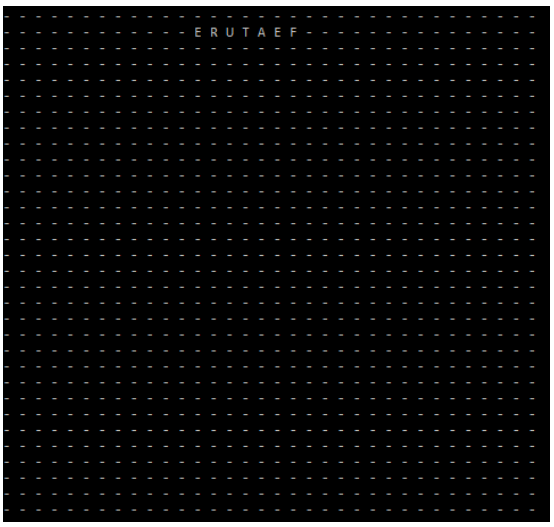

GNIZITAMARD


D
R
A
T


E
T
A
B
R
E
C
A
X
E

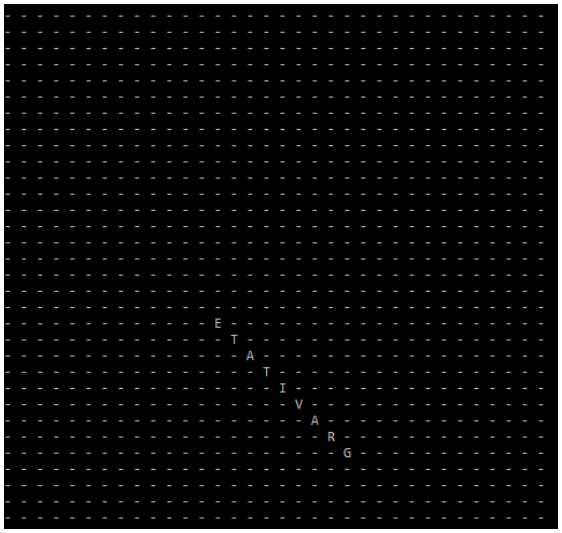

E
X
E
R
T
I
N
G


G
N
I
D
E
E
C
X
E


ERUTAEF

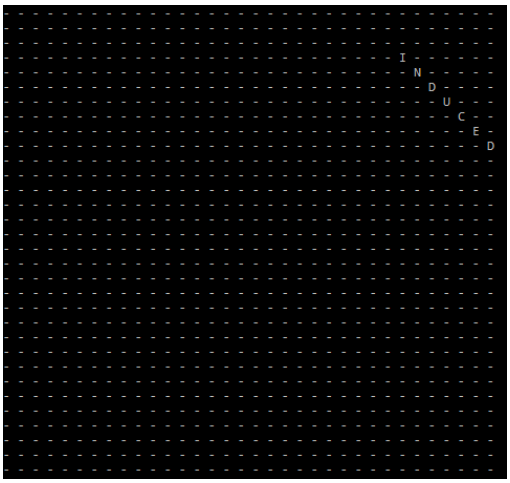

FEEBLY

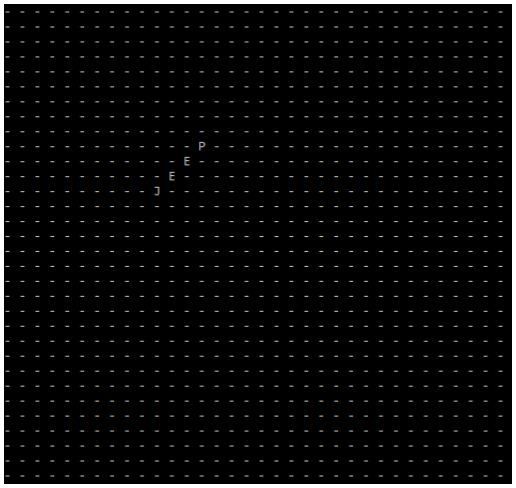

GNILDNOF

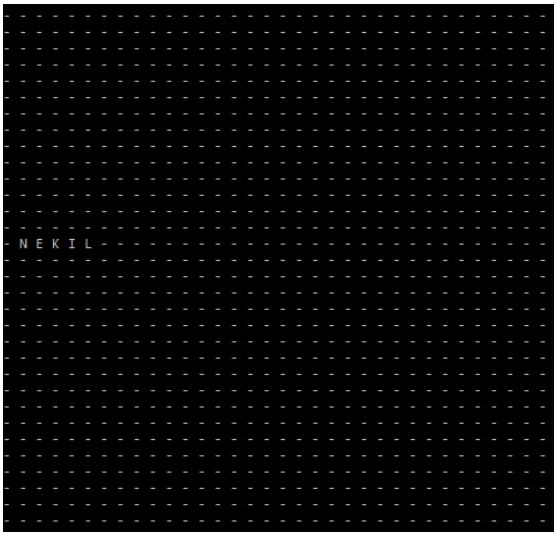

ETATIVARG


GRAVY

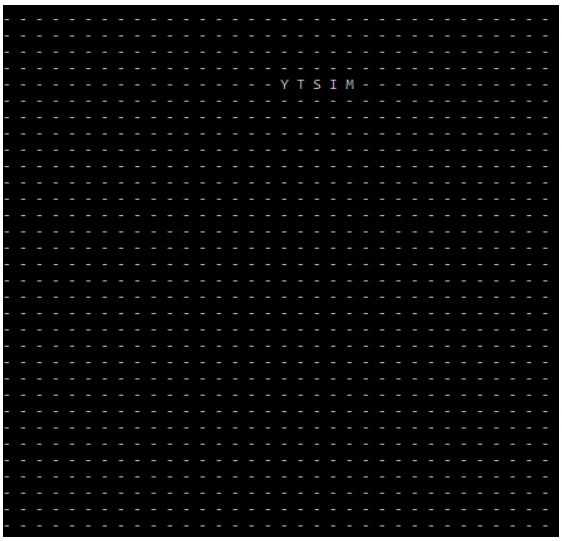

DNIH

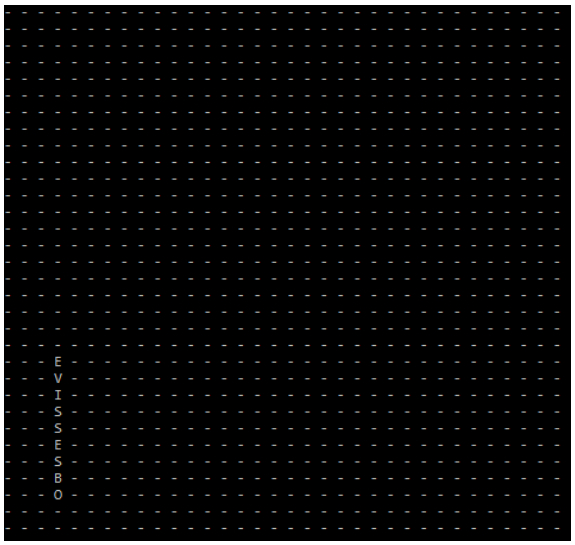

INDUCED

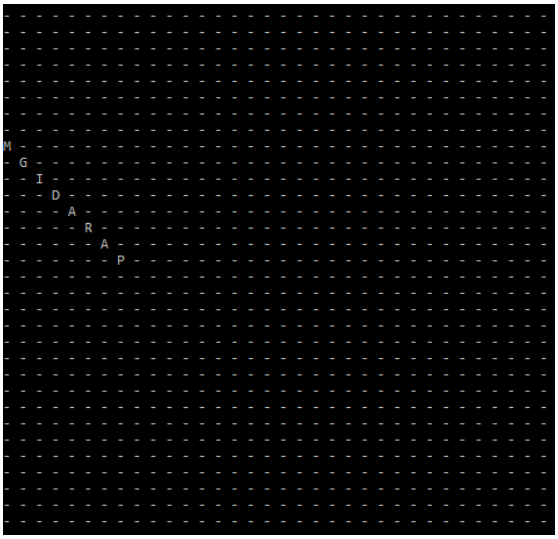

PEEJ


NEKIL

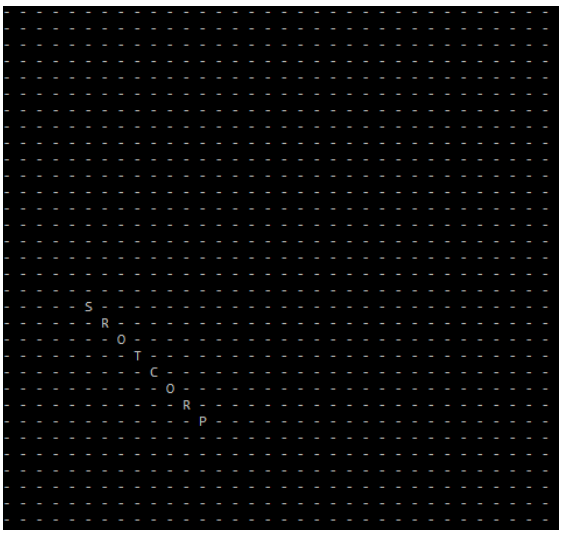

OTNEMEM


YTSIM

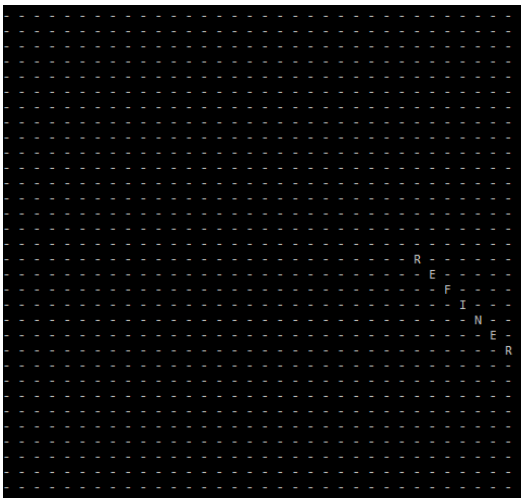

EVISESBO


MAGIDARAP

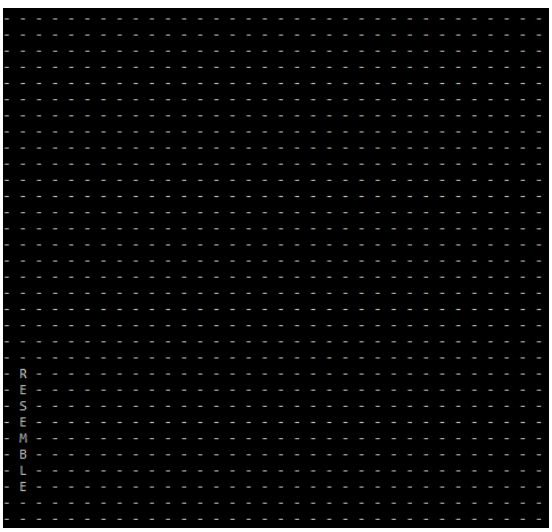

MIRGLIP

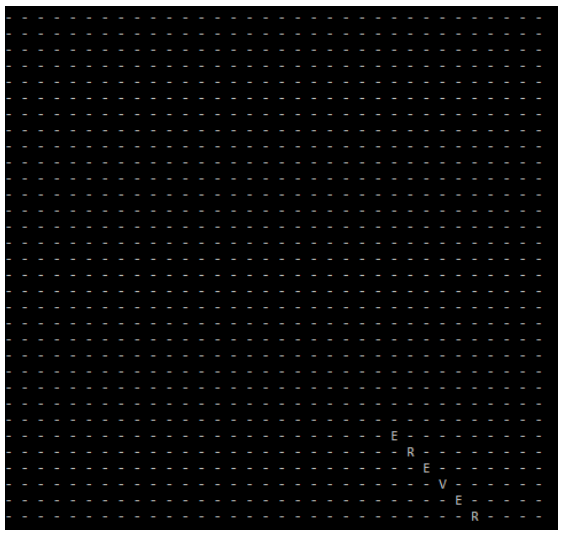

SCROTORP

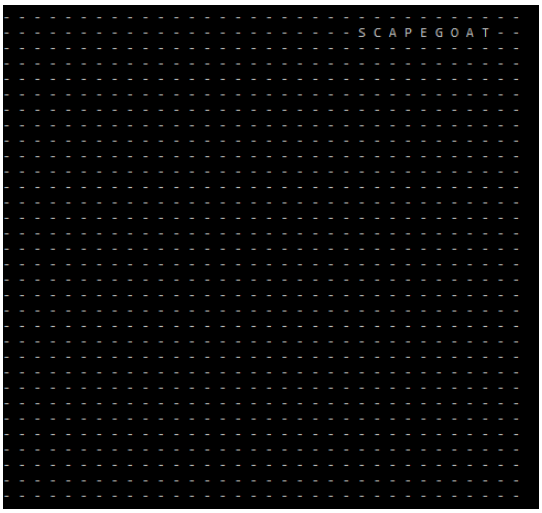

PUNISHING

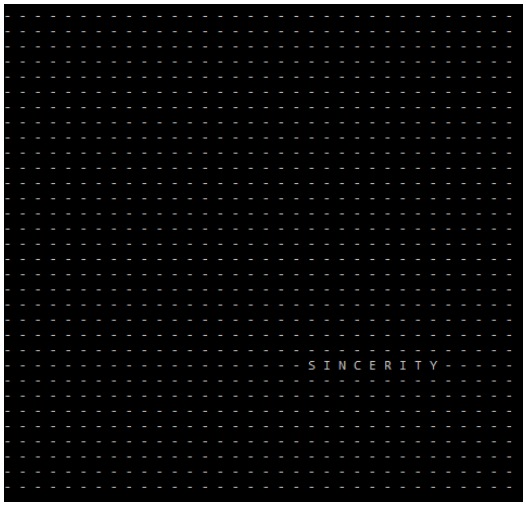

REFINER

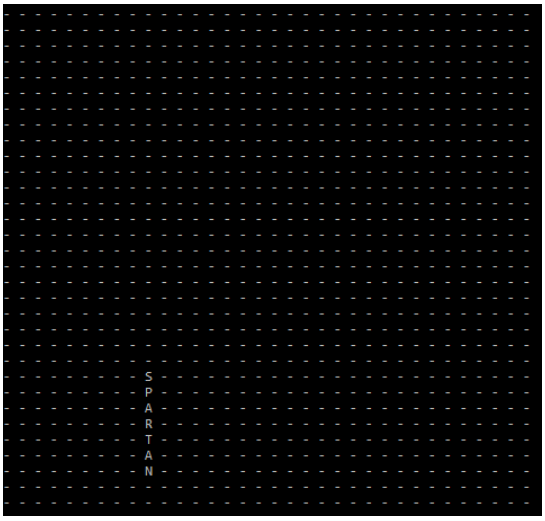

REPROVE

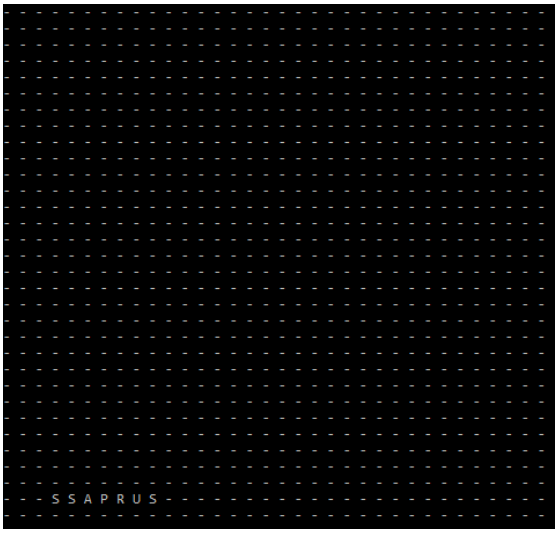

RESEMBLE


ERREVER


SCAPEGOAT


SINCERITY


SPARTAN


SSAPRUS

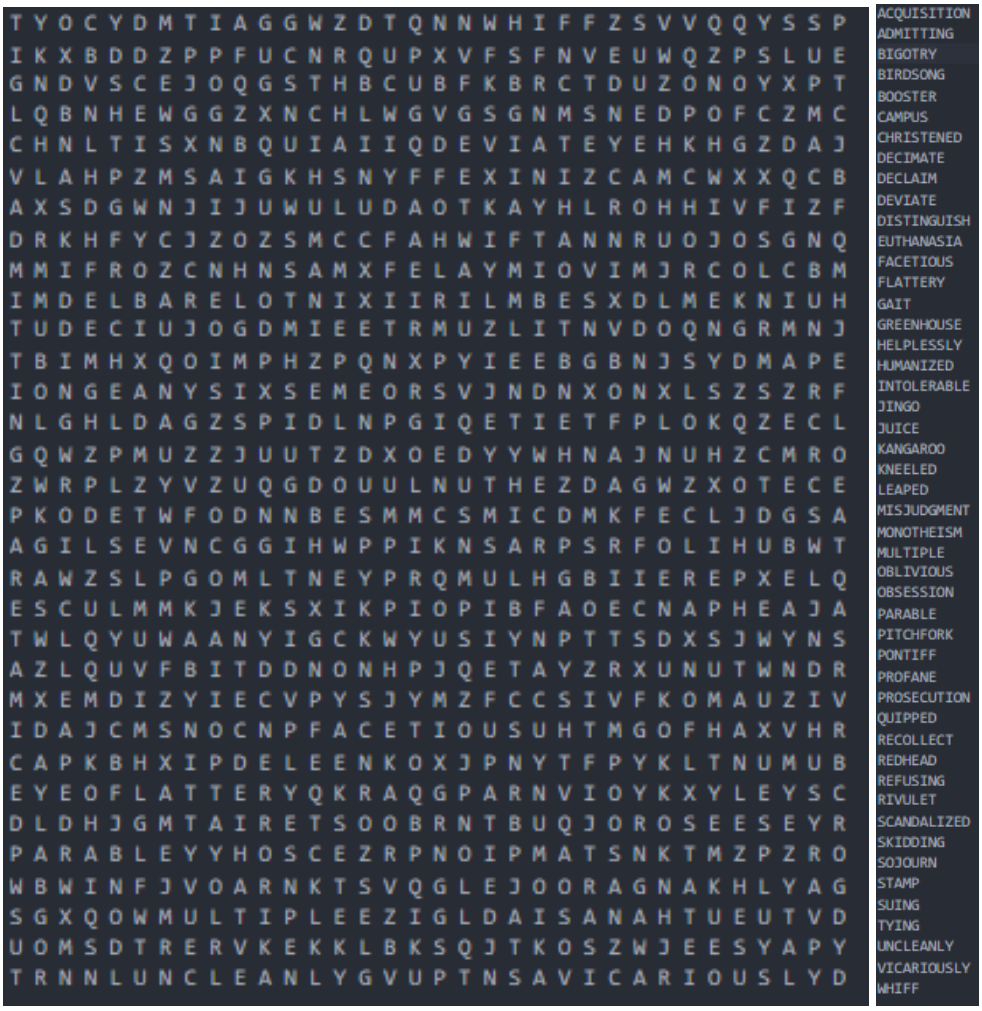

SYNCHRONIZEP


ENITNEPRUT

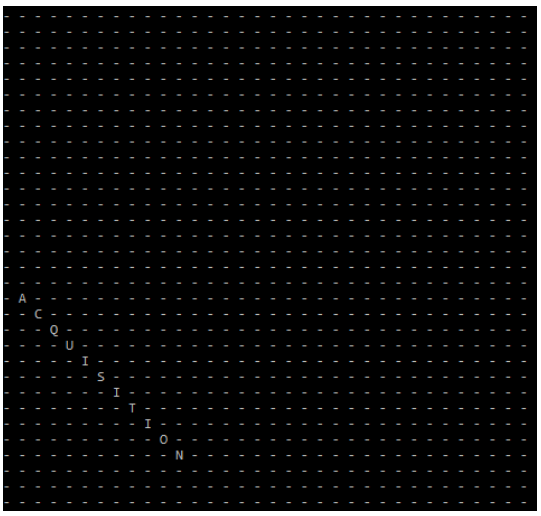
```
total perbadingan: 18017
elapsed time: 72.5469s
```

8. large_2.txt

Input:

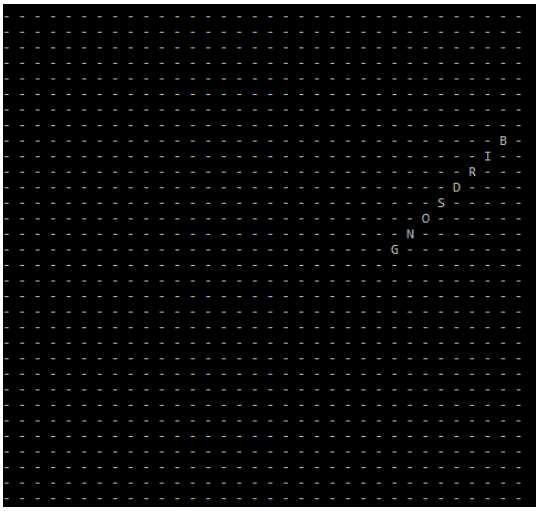

T Y O C Y D M T I A G G W Z D T Q N N W H I F F Z S V V Q Q Y S S P
I K X B D D Z P P F U C N R Q U P X V F S F N V E U W Q Z P S L U E
G N D V S C E J O Q G S T H B C U B F K B R C T D U Z O N O Y X P T
L Q B N H E W G G Z X N C H L W G V G S G N M S N E D P O F C Z M C
C H N L T I S X N B Q U I A I I Q D E V I A T E Y E H K H G Z D A J
V L A H P Z M S A I G K H S N Y F F E X I N I Z C A M C W X X Q C B
A X S D G W N J I J U W U L U D A O T K A Y H L R O H H I V F I Z F
D R K H F Y C J Z O Z S M C C F A H W I F T A N N R U O J O S G N Q
M M I F R O Z C N H N S A M X F E L A Y M I O V I M J R C O L C B M
I M D E L B A R E L O T N I X I I R I L M B E S X D L M E K N I U H
T U D E C I U J O G D M I E E T R M U Z L I T N V D O Q N G R M N J
T B I M H X Q O I M P H Z P Q N X P Y I E E B G B N J S Y D M A P E
I O N G E A N Y S I X S E M E O R S V J N D N X O N X L S Z S Z R F
N L G H L D A G Z S P I D L N P G I Q E T I E T F P L O K Q Z E C L
G Q W Z P M U Z Z J U U T Z D X O E D Y Y W H N A J N U H Z C M R O
Z W R P L Z Y V Z U Q G D O U U L N U T H E Z D A G W Z X O T E C E
P K O D E T W F O D N N B E S M M C S M I C D M K F E C L J D G S A
A G I L S E V N C G G I H W P P I K N S A R P S R F O L I H U B W T
R A W Z S L P G O M L T N E Y P R Q M U L H G B I I E R E P X E L Q
E S C U L M M K J E K S X I K P I O P I B F A O E C N A P H E A J A
T W L Q Y U W A A N Y I G C K W Y U S I Y N P T T S D X S J W Y N S
A Z L Q U V F B I T D D N O N H P J Q E T A Y Z R X U N U T W N D R
M X E M D I Z Y I E C V P Y S J Y M Z F C C S I V F K O M A U Z I V
I D A J C M S N O C N P F A C E T I O U S U H T M G O F H A X V H R
C A P K B H X I P D E L E E N K O X J P N Y T F P Y K L T N U M U B
E Y E O F L A T T E R Y Q K R A Q G P A R N V I O Y K X Y L E Y S C
D L D H J G M T A I R E T S O O B R N T B U Q J O R O S E E S E Y R
P A R A B L E Y Y H O S C E Z R P N O I P M A T S N K T M Z P Z R O
W B W I N F J V O A R N K T S V Q G L E J O O R A G N A K H L Y A G
S G X Q O W M U L T I P L E E Z I G L D A I S A N A H T U E U T V D
U O M S D T R E R V K E K K L B K S Q J T K O S Z W J E E S Y A P Y
T R N N L U N C L E A N L Y G V U P T N S A V I C A R I O U S L Y D
ACQUISITION
ADMITTING
BIGOTRY
BIRDSONG
BOOSTER
CAMPUS
CHRISTENED
DECIMATE
DECLAIM
DEVIATE
DISTINGUISH
EUTHANASIA
FACETIOUS
FLATTERY
GAIT
GREENHOUSE
HELPLESSLY
HUMANIZED
INTOLERABLE
JINGO
JUICE
KANGAROO
KNEELED
LEAPED
MISJUDGMENT
MONOTHEISM
MULTIPLE
OBLIVIOUS
OBSESSION
PARABLE
PITCHFORK
PONTIFF
PROFANE
PROSECUTION
QUIPPED
RECOLLECT
REDHEAD
REFUSING
RIVULET
SCANDALIZED
SKIDDING
SOJOURN
STAMP
SUING
TYING
UNCLEANLY
VICARIOUSLY
WHIFF

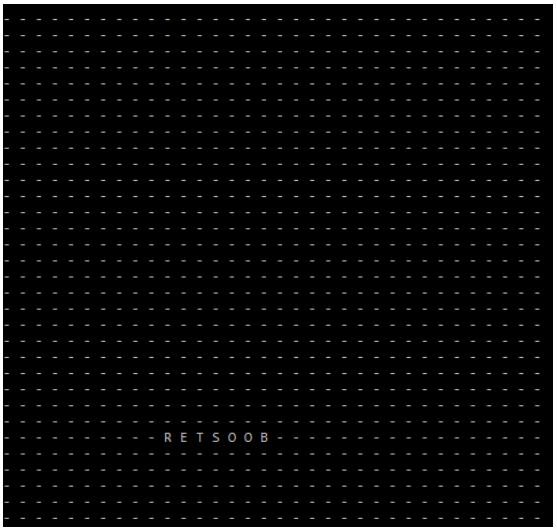
Output:

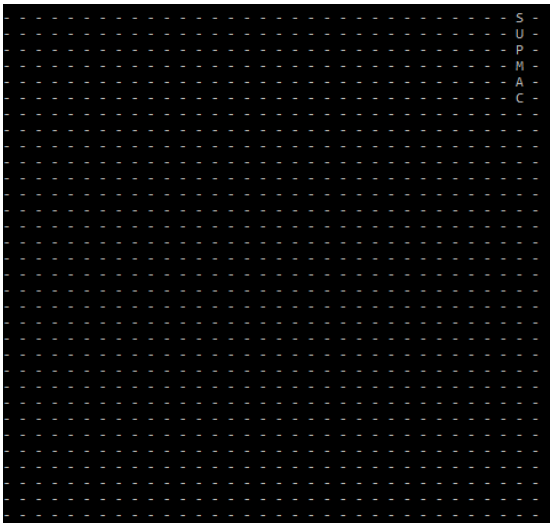

A
C
Q
U
I
S
I
T
I
O
N

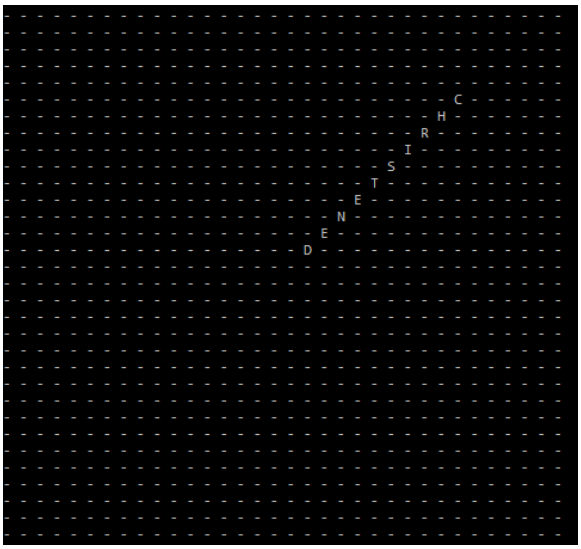

ADMITTING

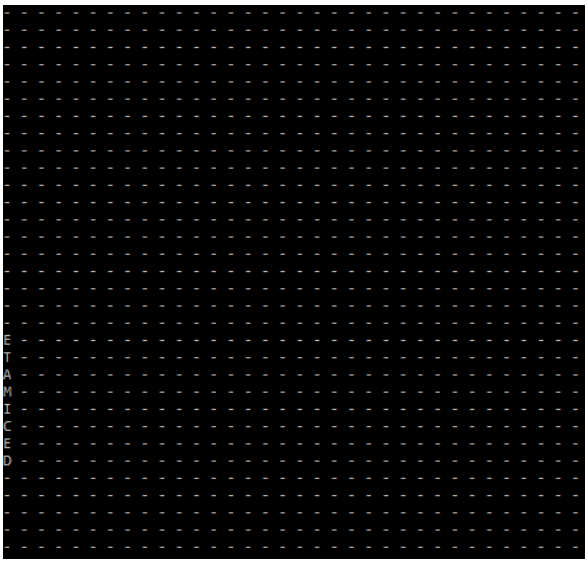

BIGOTRY

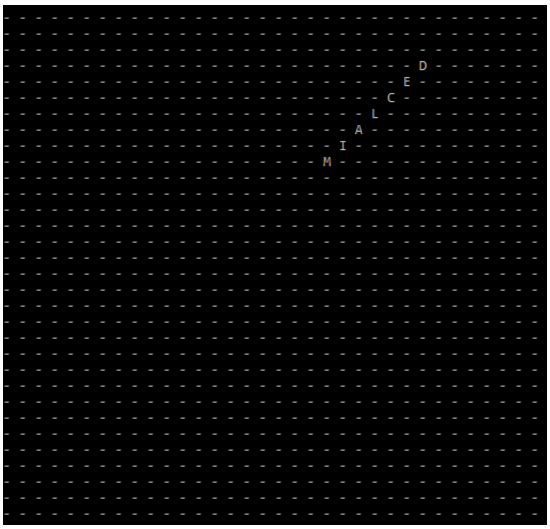

BIRDSONG


BOOTSER

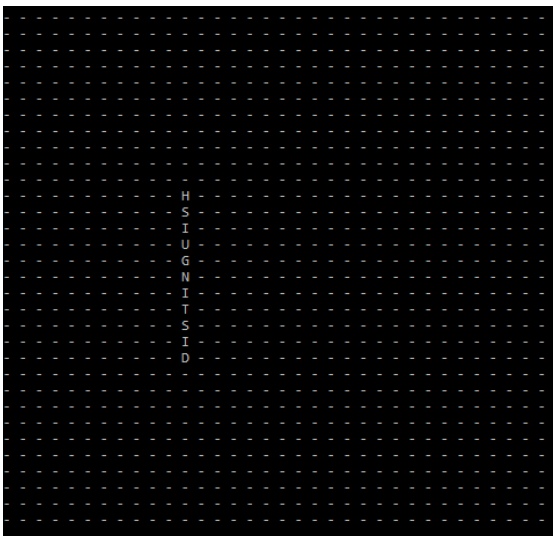

SUPMAC


DENTISTRY
CHEMICAL


DECIMATE


DECLAIM

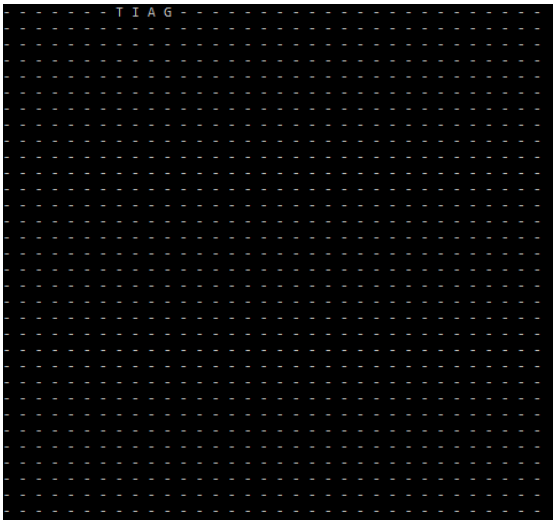

DEVIATE

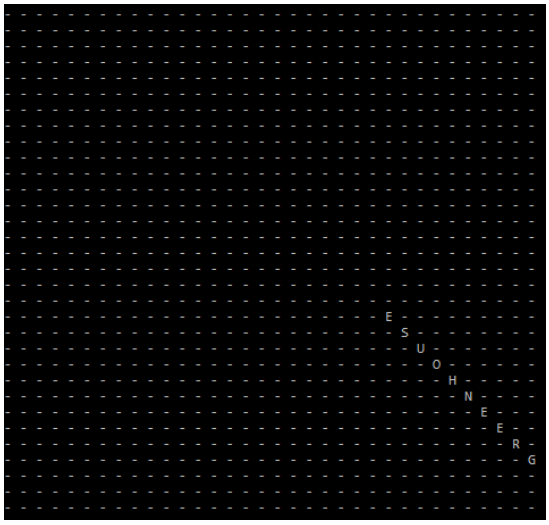

HSIUGNITSID

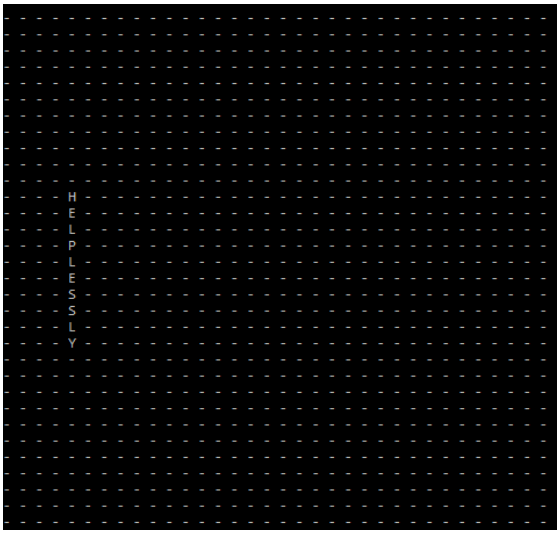

AISANAHTUE


FACETIOUS

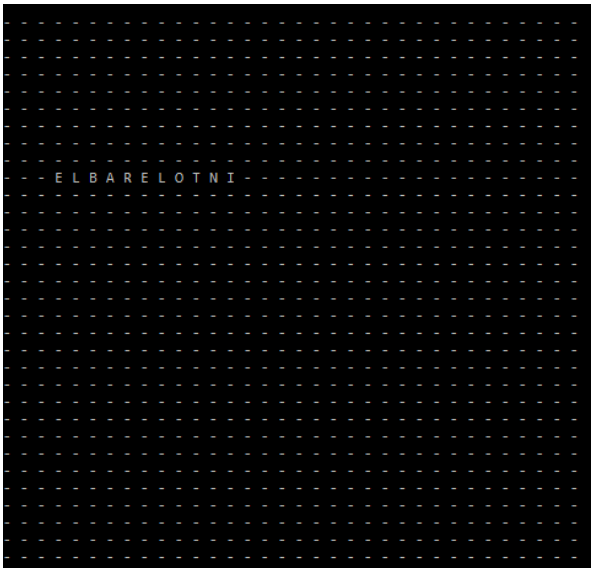

FLATTERY

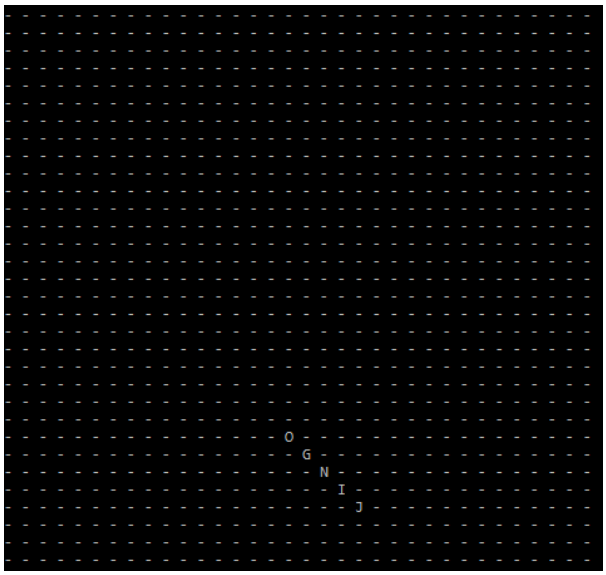

TIAG

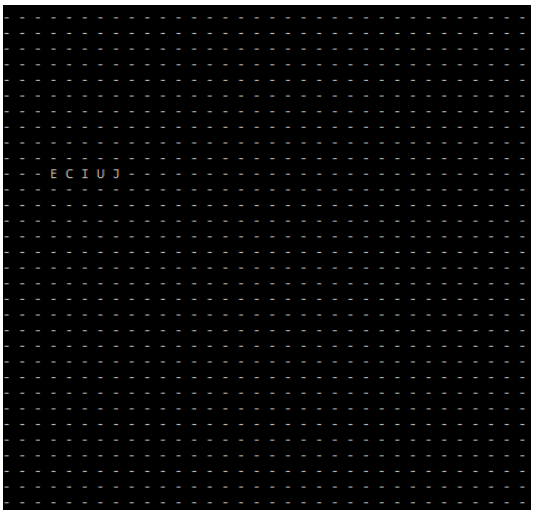

ESUOHNEERG

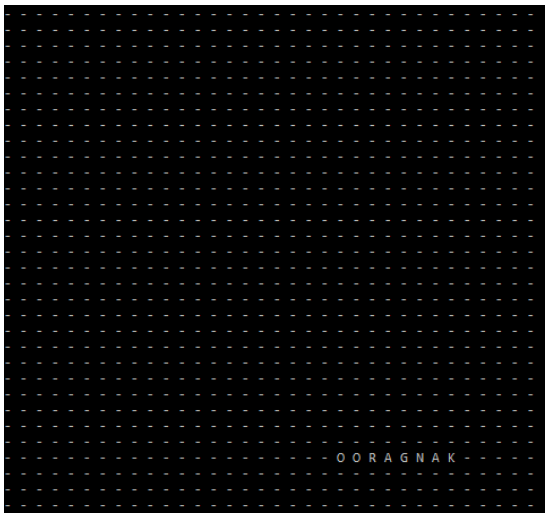

HELPLESSLY


HUMANIZED


ELBARELOTNI


OGNIJ


ECIUJ


OORAGNAK


DELEENK

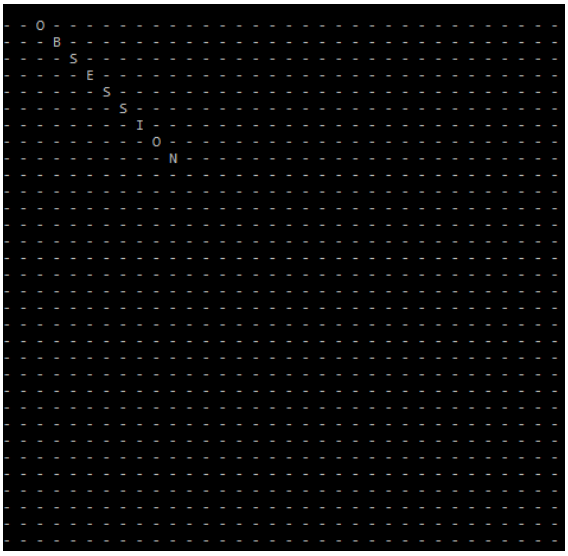

LEAPED

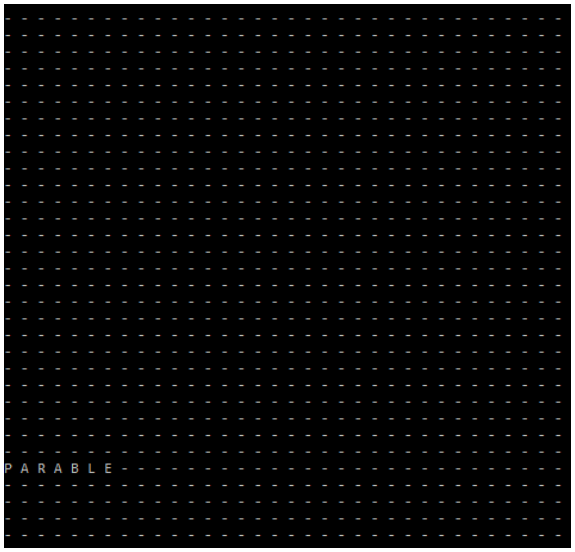

MISJUDGMENT


MISTHEONOM

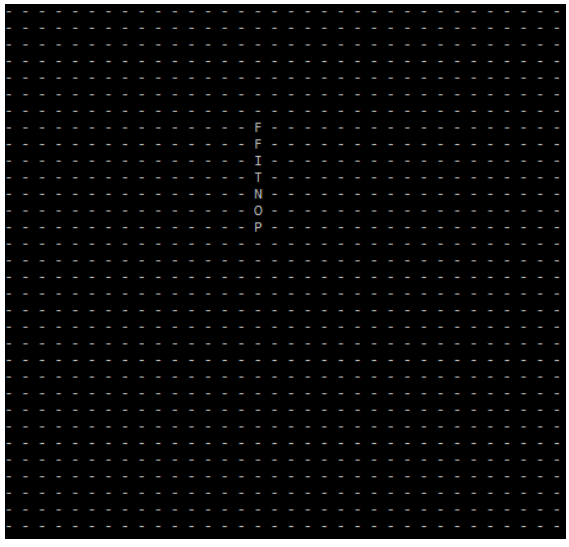

MULTIPLE


OBLIVIOUS

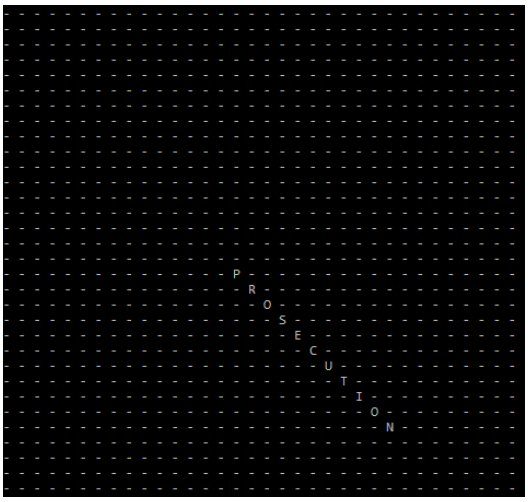

OBSESSION

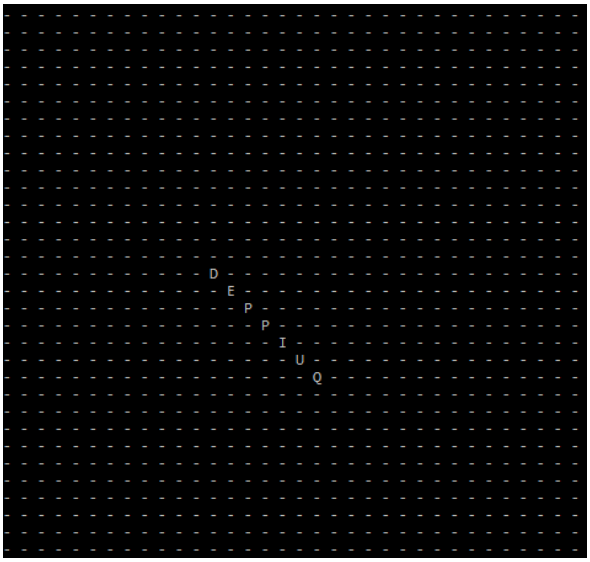

PARABLE


PITCHFORK


FFITNOP

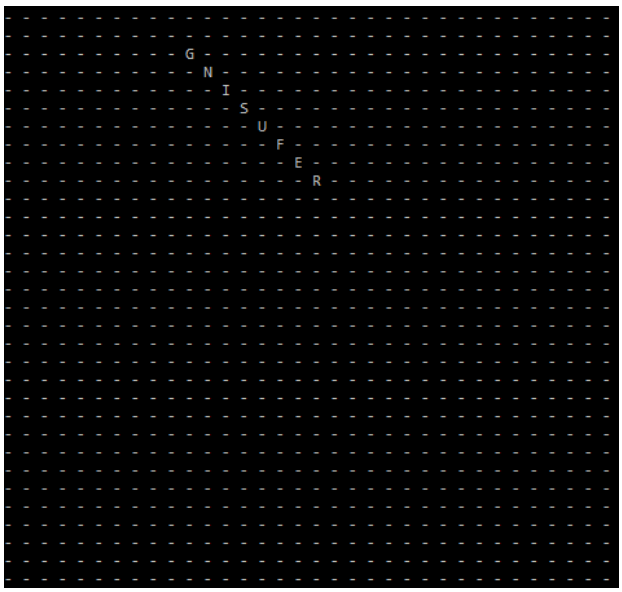

E
N
A
F
O
R
P

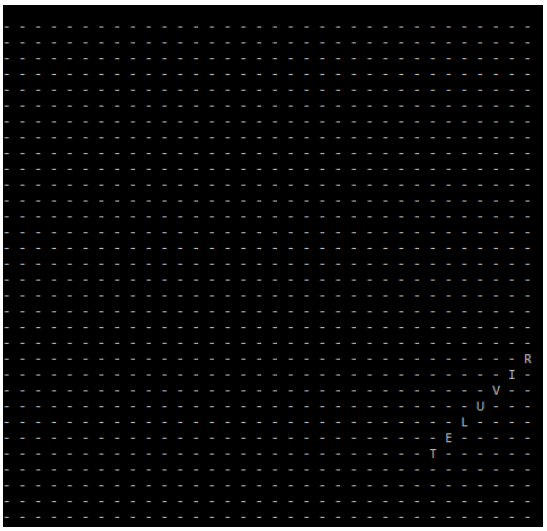

P
R
O
S
E
C
U
T
I
O
N

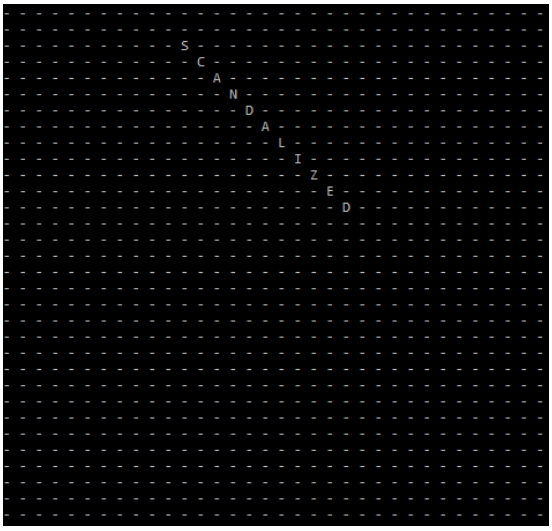

D
E
P
P
I
U
Q

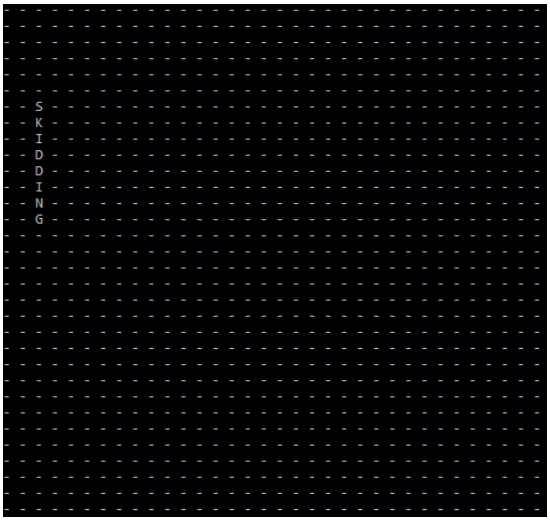

R
E
C
O
L
L
E
C
T

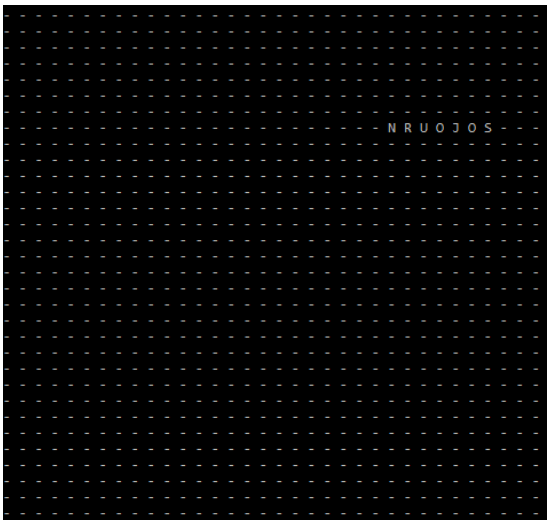

R
E
D
H
E
A
D

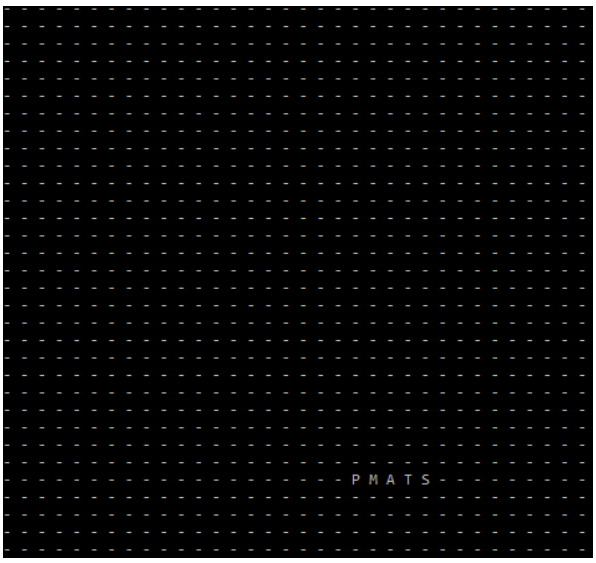

G
N
I
S
U
F
E
R

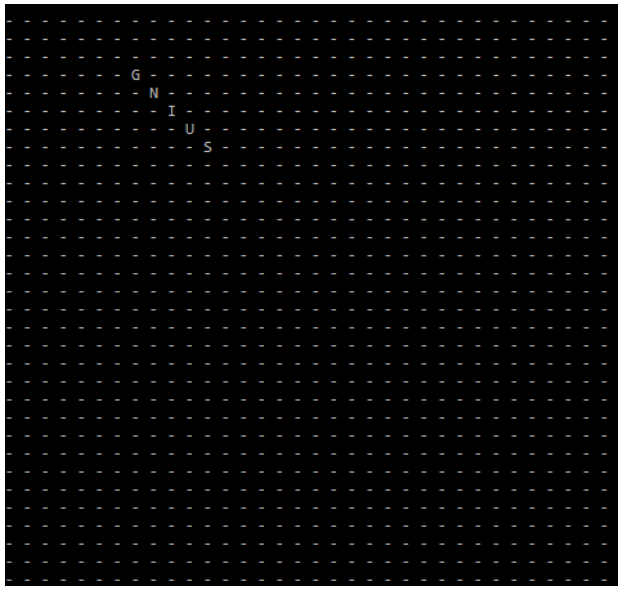

RIVULET

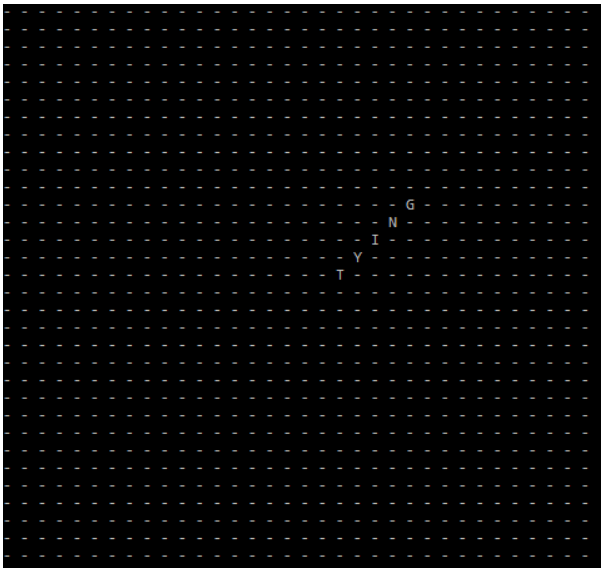

SCANDALIZED

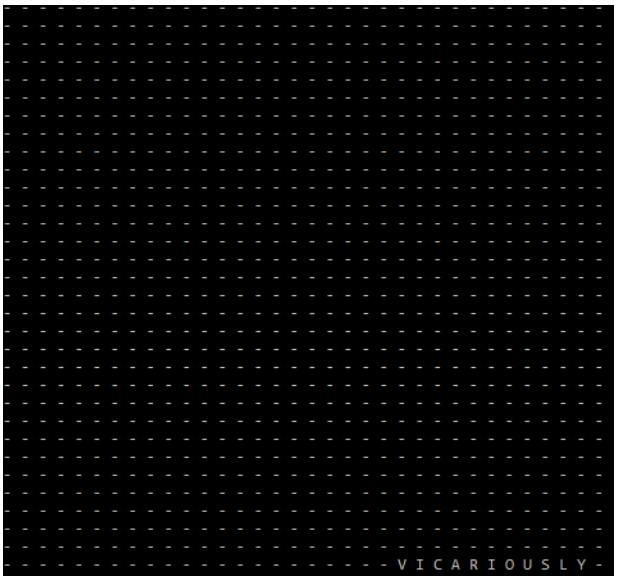

SKIDDING

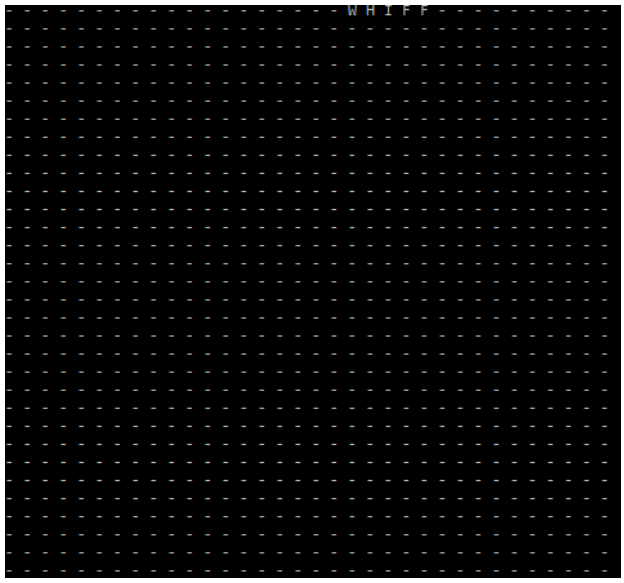

NRUOJOS


PMATS


GNIUS








```
total perbadingan: 17954
elapsed time: 55.5219s
```

BAB IV

KESIMPULAN

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

Link drive:

https://drive.google.com/drive/folders/1yZhQ7oUrIyqsDA30q60BwIdxex0HdvSF?usp=sharing

(buka dengan akun std)