LAPORAN TUGAS KECIL I

IF2211 STRATEGI ALGORITMA

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah

IF2211 Strategi Algoritma

Disusun oleh:

Muhammad Helmi Hibatullah

13520014

K-02

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER 2 TAHUN 2021/2022

DAFTAR ISI

Algoritma *Brute Force*

Algoritma *Brute force* adalah pendekatan yang lempang untuk memecahkan suatu persoalan. Pada tugas kecil pertama ini, persoalan yang akan diselesaikan adalah permainan *Word Search Puzzle*. Permainan ini adalah permainan kata dimana permain harus menemukan beberapa kata tersembunyi dalam kumpulan huruf acak.

Pencarian pola kata pada papan permainan dilakukan dengan mengiterasi setiap karakter papan kemudian mencocokkan setiap karakter selanjutnya dengan pola yang dicari pada delapan arah. Berikut adalah langkah-langkahnya.

1. Iterasi setiap karakter papan. Mulai dari pojok kiri atas sampai ke pojok kanan bawah. Bandingkan apakah karakter papan tersebut sama dengan karakter pertama pada pola. Jika berbeda, maka pencarian dihentikan dan maju ke iterasi selanjutnya (pencarian sekuensial).
2. Jika karakter pertama pada pola sama dengan karakter papan, maka lakukan pengulangan sebanyak delapan kali untuk melakukan pencarian pada paling banyak delapan arah. Delapan arah tersebut adalah horizontal ke kiri, horizontal ke kanan, vertikal ke bawah, vertikal ke atas, diagonal ke kanan bawah, diagonal ke kiri atas, diagonal ke kiri bawah, dan diagonal ke kanan atas.
3. Simpan nilai indeks karakter papan selanjutnya (sesuai dengan arah yang tentukan pada langkah nomor 3) dan indeks karakter pola selanjutnya. Jika indeks karakter papan selanjutnya melebihi batas papan, maka pencarian dihentikan dan maju ke pengulangan selanjutnya untuk arah yang baru.
4. Jika indeks karakter papan selanjutnya masih pada batas papan, bandingkan apakah karakter papan selanjutnya sama dengan karakter pola. Jika berbeda, pencarian dihentikan dan maju ke pengulangan selanjutnya untuk arah yang baru (kembali ke nomor 4 dengan arah yang berbeda).
5. Ulangi langkah pada nomor 4 dan 5 sampai indeks karakter pola sama dengan panjang pola.
6. Jika karakter pola tidak pernah sampai ke panjang pola, maka pola tersebut tidak ditemukan pada papan. Jika karakter pola dapat sampai ke panjang pola, maka lakukan pewarnaan pada papan untuk menandai jawaban. Pada langkah ini pencarian pola sudah dihentikan.
7. Indeks pencarian karakter pertama pada papan pada langkah nomor 1 digunakan untuk menandai indeks dimana pewarnaan akan dimulai. Kemudian dilakukan pewarnaan mengikuti arah pencarian pada langkah nomor 3 sampai dengan panjang pola.
8. Ulangi langkah pada nomor 1-7 untuk pola selanjutnya.

Kode Program

Kode program ditulis dalam bahasa C. Kode dibuat dengan menggunakan *Abstract Data Type* (ADT) sehingga terdapat kode yang hanya berisi header dari ADT tersebut dan tersimpan pada folder bernama header.

board.h dan board.c

board.h adalah header dari ADT Board dan board.c merupakan body-nya. ADT Board ini digunakan untuk menampung data-data tentang papan permainan, seperti ukuran baris kolom, karakter-karakternya, dan warna-warna yang harus ditampilkannya.

```c
/**
 * File: board.h
 *
 * ADT Board
 * Menampung data papan permainan
 */

#ifndef BOARD_H
#define BOARD_H

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "boolean.h"

#define DR  "\x1b[31m"  // DARK RED
#define DG  "\x1b[32m"  // DARK GREEN
#define DY  "\x1b[33m"  // DARK YELLOW
#define DB  "\x1b[34m"  // DARK BLUE
#define DM  "\x1b[35m"  // DARK MAGENTA
#define DC  "\x1b[36m"  // DARK CYAN
#define BR  "\x1b[91m"  // BRIGHT RED
#define BG  "\x1b[92m"  // BRIGHT GREEN
#define BY  "\x1b[93m"  // BRIGHT YELLOW
#define BB  "\x1b[94m"  // BRIGHT BLUE
#define BM  "\x1b[95m"  // BRIGHT MAGENTA
#define BC  "\x1b[96m"  // BRIGHT CYAN
```

```
#define RE  "\x1b[0m"   // RESET

extern char *colors[];

typedef struct
{
    char symbol;
    char color[3];  // WH = WHITE
} coloredChar;

typedef struct
{
    coloredChar **buffer;
    int row;
    int col;
} board;

void createBoard(board *brd);
void readBoardDimensionFromFile(FILE *fp, board *brd, boolean *isSuccess);
void readBoardBufferFromFile(FILE *fp, board *brd);
void printBoard(board brd);

#endif
```

```
/**
 * File: board.c
 *
 * ADT Board
 * Menampung data papan permainan
 */

#include "../header/board.h"

/* array of colors */
char *colors[] = { "DR\0", "DG\0", "DY\0", "DB\0", "DM\0", "DC\0", "BR\0",
"BG\0", "BY\0", "BB\0", "BM\0", "BC\0" };

void createBoard(board *brd) {
    (*brd).col = 0;
    (*brd).row = 1;
```

```
}

void readBoardDimensionFromFile(FILE *fp, board *brd, boolean *isSuccess)
{
    char tempchar;

    /* read columns */
    while (tempchar != '\n') {
        fscanf(fp, "%c", &tempchar);
        if (tempchar != ' ' && tempchar != '\n') {
            (*brd).col++;
        }
    }
    /* read rows */
    char str[500];
    while (*fgets(str, sizeof(str), fp) != '\n') {
        (*brd).row++;
    }
    /* allocate buffer */
    (*brd).buffer = (coloredChar **) malloc ((*brd).row *
sizeof(coloredChar *));
    if ((*brd).buffer != NULL) {
        for (int i = 0; i < (*brd).row; i++) {
            (*brd).buffer[i] = (coloredChar *) malloc ((*brd).col *
sizeof(coloredChar));
            if ((*brd).buffer[i] == NULL) {
                printf("Board allocation failed.\n");
                *isSuccess = false;
                break;
            }
        }
        /* assign all symbols to white */
        for (int i = 0; i < (*brd).row; i++) {
            for (int j = 0; j < (*brd).col; j++) {
                strcpy((*brd).buffer[i][j].color, "WH\0");
            }
        }
        *isSuccess = true;
    } else {
        printf("Board allocation failed.\n");
```

```
        *isSuccess = false;
    }
}

void readBoardBufferFromFile(FILE *fp, board *brd) {
    char tempchar;

    for (int i = 0; i < (*brd).row; i++) {
        for (int j = 0; j < (*brd).col; j++) {
            fscanf(fp, " %c", &brd->buffer[i][j]);
        }
    }
    /* to skip the extra 2 newlines */
    fscanf(fp, "%c", &tempchar);
    fscanf(fp, "%c", &tempchar);

    printf("Board %d x %d loaded successfully.\n", (*brd).row,
(*brd).col);
}

void printBoard(board brd) {
    printf("\n");
    for (int i = 0; i < brd.col*2 + 3; i++) {
        printf("=");
    }
    printf("\n");
    for (int i = 0; i < brd.row; i++) {
        for (int j = 0; j < brd.col; j++) {
            if (j == 0) {
                printf("| ");
            }
            if (strcmp(brd.buffer[i][j].color, "DR") == 0) {
                printf(DR "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "DG") == 0) {
                printf(DG "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "DY") == 0) {
                printf(DY "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "DB") == 0) {
                printf(DB "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "DM") == 0) {
```

```
                printf(DM "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "DC") == 0) {
                printf(DC "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BR") == 0) {
                printf(BR "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BG") == 0) {
                printf(BG "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BY") == 0) {
                printf(BY "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BB") == 0) {
                printf(BB "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BM") == 0) {
                printf(BM "%c " RE, brd.buffer[i][j].symbol);
            } else if (strcmp(brd.buffer[i][j].color, "BC") == 0) {
                printf(BC "%c " RE, brd.buffer[i][j].symbol);
            } else {
                printf("%c ", brd.buffer[i][j].symbol);
            }
        }
        printf("|\n");
    }
    for (int i = 0; i < brd.col*2 + 3; i++) {
        printf("=");
    }
    printf("\n\n");
}
```

patternList.h dan patternList.c

patternList.h adalah header dari ADT Pattern List dan patternList.c merupakan body-nya. ADT Pattern List ini digunakan untuk menampung daftar pola kata yang akan dicari, seperti jumlah pola, jumlah karakter terbanyak dalam suatu pola, dan pola itu sendiri.

```
/**
 * File: patternList.h
 *
 * ADT Pattern List
 * Menampung data pola
 */
```

```c
#ifndef PATTERNLIST_H
#define PATTERNLIST_H

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

typedef struct
{
    char *buffer;
    int length;
} pattern;

typedef struct
{
    int count;
    int maxLength;
    pattern *list;
} patternList;

void createPatternList(patternList *ptl);
void readPatternListDimensionFromFile(FILE *fp, patternList *ptl, boolean
*isSuccess);
void readPatternListBufferFromFile(FILE *fp, patternList *ptl);
void printPatternList(patternList ptl);
void printPattern(pattern ptn);

#endif
```

```c
/**
 * File: patternList.c
 *
 * ADT Pattern List
 * Menampung data pola
 */

#include "../header/patternList.h"

void createPatternList(patternList *ptl) {
```

```
    (*ptl).maxLength = 0;
    (*ptl).count = 1;
}

void readPatternListDimensionFromFile(FILE *fp, patternList *ptl, boolean
*isSuccess) {
    char tempchar;
    int charCount;

    /* count the number of pattern and the max length */
    charCount = 0;
    while ((tempchar = fgetc(fp)) != EOF) {
        if (tempchar == '\n' && charCount > (*ptl).maxLength) {
            (*ptl).maxLength = charCount;
        }
        if (tempchar == '\n') {
            charCount = 1;
            (*ptl).count++;
        } else {
            charCount++;
        }
    }
    /* allocate the buffer */
    (*ptl).list = (pattern *) malloc ((*ptl).count * sizeof(pattern));
    if ((*ptl).list != NULL) {
        for (int i = 0; i < (*ptl).count; i++) {
            (*ptl).list[i].buffer = (char *) malloc ((*ptl).maxLength *
sizeof(char));
            if ((*ptl).list[i].buffer == NULL) {
                printf("List of pattern allocation failed.\n");
                *isSuccess = false;
                break;
            }
        }
        *isSuccess = true;
    } else {
        printf("List of pattern allocation failed.\n");
        *isSuccess = false;
    }
}
```

```
void readPatternListBufferFromFile(FILE *fp, patternList *ptl) {
    char tempchar;
    int i = 0;
    int j = 0;
    while ((tempchar = fgetc(fp)) != EOF) {
        (*ptl).list[i].buffer[j] = tempchar;
        if (tempchar == '\n') {
            (*ptl).list[i].buffer[j] = '\0';
            (*ptl).list[i].length = j;
            i++;
            j = 0;
        } else {
            j++;
        }
    }
    (*ptl).list[i].buffer[j] = '\0';
    (*ptl).list[i].length = j;
    printf("%d Pattern(s) loaded successfully.\n", (*ptl).count);
}

void printPatternList(patternList ptl) {
    for (int i = 0; i < ptl.count; i++) {
        printf("%s\n", ptl.list[i].buffer);
    }
}

void printPattern(pattern ptn) {
    for (int i = 0; i < ptn.length; i++) {
        printf("%c", ptn.buffer[i]);
    }
}
```

boolean.h

File ini hanya berisi kode untuk mendefinisikan nilai true adalah satu dan false adalah nol.

```
/**
 * File: boolean.h
```

```
 *
 * Mendefinisikan true dan false
*/

#ifndef _BOOLEAN_h
#define _BOOLEAN_h

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

#endif
```

fileloader.c

File ini berisi prosedur untuk membaca file masukkan kemudian menampung isinya ke dalam ADT yang sudah didefinisikan sebelumnya.

```
/**
 * File: fileloader.c
 *
 * Berisi prosedur untuk membaca file
 * kemudian menampung isinya ke dalam
 * ADT yang sudah dibuat
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../header/boolean.h"
#include "../header/board.h"
#include "../header/patternList.h"

int loadFile(char *filename, board *brd, patternList *ptl) {
    char path[500] = "../test/";
    strcat(path, filename);
    FILE *fp = fopen(path, "r");

    if (fp == NULL) {
```

```
        printf("Error. File Not Found.\n");
        printf("==================================================\n");
        return 0;
    } else {
        boolean boardSuccess, patternSuccess;

        printf("%s opened.\n", filename);
        createBoard(brd);
        readBoardDimensionFromFile(fp, brd, &boardSuccess);
        if (boardSuccess) {
            createPatternList(ptl);
            readPatternListDimensionFromFile(fp, ptl, &patternSuccess);
            if (patternSuccess) {
                rewind(fp);

                readBoardBufferFromFile(fp, brd);
                readPatternListBufferFromFile(fp, ptl);
            }
        }
        fclose(fp);
    }
    printf("==================================================\n");
}
```

solver.c

File ini berisi tiga prosedur yaitu addColor, search, dan solve yang tujuan umumnya adalah untuk menyelesaikan persoalan pada tugas kecil ini dengan menggunakan algoritma *brute force*. Fungsi addColor berguna untuk menambahkan warna pada papan jika pola kata yang dicari sudah ditemukan, kemudian prosedur search digunakan untuk mencari pola kata pada baris dan kolom yang sudah ditentukan. Terakhir prosedur solve digunakan untuk mengiterasi setiap karakter papan dari pojok kiri atas hingga pojok kanan bawah.

```
/**
 * File: solver.c
 *
 * Berisi prosedur untuk menyelesaikan
 * persoalan dengan algoritma brute force
 */
```

```c
#include <stdio.h>
#include <stdlib.h>
#include "../header/boolean.h"
#include "../header/board.h"
#include "../header/patternList.h"

int horizontal[] = { 1, -1, 0,  0, 1, -1, -1,  1 };
int vertical[]   = { 0,  0, 1, -1, 1, -1,  1, -1 };
/**
 * Jika i adalah indeks pada array horizontal dan vertikal, maka
 * i = 0 -> horizontal ke kiri
 * i = 1 -> horizontal ke kanan
 * i = 2 -> vertikal ke bawah
 * i = 3 -> vertikal ke atas
 * i = 4 -> diagonal ke kanan bawah
 * i = 5 -> diagonal ke kiri atas
 * i = 6 -> diagonal ke kiri bawah
 * i = 7 -> diagonal ke kanan atas
 */

void addColor(board *brd, int patternLength, int row, int col, int color,
int direction) {
    int i, rowDirection, colDirection;

    rowDirection = row;
    colDirection = col;
    for (i = 0; i < patternLength; i++) {
        strcpy((*brd).buffer[rowDirection][colDirection].color,
colors[color]);

        rowDirection += vertical[direction];
        colDirection += horizontal[direction];
    }
}

void search(board brd, pattern ptn, int *count, int row, int col, boolean
*found, int color) {
    int i, j, rowDirection, colDirection;
```

```
    *found = false;
    if (brd.buffer[row][col].symbol == ptn.buffer[0]) {
        *count += 1;
        i = 0;
        while (i < 8 && !*found) {
            rowDirection = row + vertical[i];
            colDirection = col + horizontal[i];
            j = 1;
            while (j < ptn.length && 0 <= rowDirection && rowDirection <
brd.row && 0 <= colDirection && colDirection < brd.col) {
                if (ptn.buffer[j] ==
brd.buffer[rowDirection][colDirection].symbol) {
                    *count += 1;
                    rowDirection += vertical[i];
                    colDirection += horizontal[i];
                    j++;
                } else {
                    *count += 1;
                    break;
                }
            }

            if (j == ptn.length) {
                *found = true;
                addColor(&brd, ptn.length, row, col, color, i);
            } else {
                i++;
            }
        }
    } else {
        *count += 1;
    }
}

void solve(board brd, pattern ptn, int *totalCount, int color, int
*foundCount, int maxLength) {
    boolean found;
    int i, j, count, length;

    found = false;
```

```
    count = 0;
    i = 0;
    while (i < brd.row && !found) {
        j = 0;
        while (j < brd.col && !found) {
            search(brd, ptn, &count, i, j, &found, color);
            j++;
        }
        i++;
    }

    char *status[] = { "not found", "found" };
    *totalCount += count;
    if (maxLength > 13) {
        printf("%-*s %-9s", maxLength, ptn.buffer, status[found]);
    } else {
        printf("%-13s %-9s", ptn.buffer, status[found]);
    }
    *foundCount += 1;
    printf(" %13d\n", count);
}
```

main.c

File ini berisi program utama yang akan dijalankan oleh pengguna.

```
/**
 * File: main.c
 */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "board.c"
#include "patternList.c"
#include "fileloader.c"
#include "solver.c"

#define BILLION  1000000000.0
```

```
int main() {
    int tempchar, count, foundCount;
    char filename[1001];
    struct timespec start, end;
    board brd;
    patternList ptl;

    printf("==================================================\n");
    printf("Input configuration file: ");
    scanf("%s", filename);
    printf("--------------------------------------------------\n");
    if (loadFile(filename, &brd, &ptl) != 0) {
        char *header[] = { "No.", "Word Searched", "Status",
"Comparison(s)"};
        printf("%s %-*s %-9s %s\n", header[0], ptl.maxLength, header[1],
header[2], header[3]);

        count = 0;
        foundCount = 0;
        clock_gettime(CLOCK_REALTIME, &start);
        for (int i = 0; i < ptl.count; i++) {
            printf("%2d. ", i+1);
            solve(brd, ptl.list[i], &count, i%12, &foundCount,
ptl.maxLength);
        }
        clock_gettime(CLOCK_REALTIME, &end);
        double totalTime = (end.tv_nsec - start.tv_nsec)/BILLION;

        printBoard(brd);
        printf("Pattern found (%d/%d).\n", foundCount, ptl.count);
        printf("Total time needed: %lfs.\n", totalTime);
        printf("Total comparison:  %d.\n\n", count);

        free(brd.buffer);
        free(ptl.list->buffer);
        free(ptl.list);
    }
    printf("Input any number to exit. ");
    scanf("%d", &tempchar);
}
```

Tangkapan Layar Uji Coba Program

Ukuran Kecil

File input: small-animals.txt

Input:

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

SNAIL
DOG
KOALA
SQUIRREL
ELEPHANT
FROG
TURTLE
CHICKEN
DOLPHIN
HORSE
UNICORN
PIG
```

Output:

```
================================================
Input configuration file: animal-small.txt
------------------------------------------------
animal-small.txt opened.
Board 15 x 15 loaded successfully.
12 Pattern(s) loaded successfully.
================================================
No. Word Searched Status    Comparison(s)
 1. SNAIL         found                30
 2. DOG           found                91
 3. KOALA         found                87
 4. SQUIRREL      found               177
 5. ELEPHANT      found               153
 6. FROG          found                30
 7. TURTLE        found               132
 8. CHICKEN       found               170
 9. DOLPHIN       found               140
10. HORSE         found               121
11. UNICORN       found                18
12. PIG           found               216

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

Pattern found (12/12).
Total time needed: 0.033151s.
Total comparison:  1365.

Input any number to exit. 0
```

File input: small-flowers.txt

Input:

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

ANEMONE
CARNATION
CHRYSANTHEMUM
DAFFODIL
FREESIA
GLADIOLUS
HYACINTH
LILY
ORCHID
ROSE
SUNFLOWER
TULIP
```

Output:

```
================================================
Input configuration file: flowers-small.txt
------------------------------------------------
flowers-small.txt opened.
Board 15 x 15 loaded successfully.
12 Pattern(s) loaded successfully.
================================================
No. Word Searched  Status    Comparison(s)
 1. ANEMONE        found                288
 2. CARNATION      found                 21
 3. CHRYSANTHEMUM  found                246
 4. DAFFODIL       found                 14
 5. FREESIA        found                226
 6. GLADIOLUS      found                161
 7. HYACINTH       found                166
 8. LILY           found                 77
 9. ORCHID         found                188
10. ROSE           found                281
11. SUNFLOWER      found                101
12. TULIP          found                 23

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

Pattern found (12/12).
Total time needed: 0.034623s.
Total comparison:  1792.

Input any number to exit. 0
```

File input: small-foods.txt

Input:

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

BATAGOR
CILOK
CIRENG
COLENAK
COMBRO
GEHU
GEPUK
KAREDOK
LOTEK
MISRO
OPAK
SEBLAK
SERABI
SOTO
```

Output:

```
================================================
Input configuration file: foods-small.txt
------------------------------------------------
foods-small.txt opened.
Board 15 x 15 loaded successfully.
14 Pattern(s) loaded successfully.
================================================
No. Word Searched Status    Comparison(s)
 1. BATAGOR       found               259
 2. CILOK         found               205
 3. CIRENG        found                94
 4. COLENAK       found               131
 5. COMBRO        found               297
 6. GEHU          found                 9
 7. GEPUK         found                22
 8. KAREDOK       found               178
 9. LOTEK         found               229
10. MISRO         found                80
11. OPAK          found                24
12. SEBLAK        found               105
13. SERABI        found               130
14. SOTO          found               197

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

Pattern found (14/14).
Total time needed: 0.039915s.
Total comparison:  1960.

Input any number to exit. 0
```

Ukuran Sedang

File input: medium-dinosaurs.txt

Input:

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

ALLOSAURUS
ANKYLOSAURUS
BRONTOSAURUS
COMPSOGNATHUS
CORYTHOSAURUS
DILOPHOSAURUS
HADROSAURUS
MEGALOSAURUS
PTERODACTYL
SPINOSAURUS
STEGOSAURUS
TREX
TRICERATOPS
UTAHRAPTOR
VELOCIRAPTOR
```

Output:

```
================================================
Input configuration file: dinosaurs-medium.txt
------------------------------------------------
dinosaurs-medium.txt opened.
Board 20 x 20 loaded successfully.
15 Pattern(s) loaded successfully.
================================================
No. Word Searched  Status    Comparison(s)
 1. ALLOSAURUS     found              293
 2. ANKYLOSAURUS   found              382
 3. BRONTOSAURUS   found              380
 4. COMPSOGNATHUS  found               21
 5. CORYTHOSAURUS  found               37
 6. DILOPHOSAURUS  found               54
 7. HADROSAURUS    found              287
 8. MEGALOSAURUS   found              389
 9. PTERODACTYL    found              282
10. SPINOSAURUS    found              489
11. STEGOSAURUS    found              472
12. TREX           found              215
13. TRICERATOPS    found               41
14. UTAHRAPTOR     found              605
15. VELOCIRAPTOR   found              458

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

Pattern found (15/15).
Total time needed: 0.045489s.
Total comparison:  4405.

Input any number to exit. 1
```

File input: medium-fishes.txt

Input:

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

AMBERJACK
ANCHOVY
BASS
CARP
DORY
COD
GOBY
DAMSELFISH
HAKE
LUMPFISH
MACKEREL
REDISH
SALMON
TROUT
WHITEFISH
TETRA
PIRANHA
HALIBUT
BREAM
```

Output:

```
================================================
Input configuration file: fishes-medium.txt
------------------------------------------------
fishes-medium.txt opened.
Board 22 x 22 loaded successfully.
19 Pattern(s) loaded successfully.
================================================
No. Word Searched Status    Comparison(s)
 1. AMBERJACK     found               61
 2. ANCHOVY       found              309
 3. BASS          found              487
 4. CARP          found              544
 5. DORY          found              557
 6. COD           found              384
 7. GOBY          found              214
 8. DAMSELFISH    found              123
 9. HAKE          found              315
10. LUMPFISH      found              202
11. MACKEREL      found              290
12. REDISH        found              202
13. SALMON        found              584
14. TROUT         found              149
15. WHITEFISH     found              302
16. TETRA         found              361
17. PIRANHA       found              350
18. HALIBUT       found              353
19. BREAM         found              149

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

Pattern found (19/19).
Total time needed: 0.055343s.
Total comparison:  5936.

Input any number to exit. 1
```

File input: medium-trains.txt

Input:

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

MALABAR
RANGGAJATI
GUMARANG
CIREMAI
WIJAYAKUSUMA
HARINA
MATARAM
LODAYA
KERTANEGARA
PANGANDARAN
SANCAKA
SRIWIJAYA
SINGASARI
JAYABAYA
BANGUNKARTA
BOGOWONTO
KAMANDAKA
KALIGUNG
BRANTAS
PANGRANGO
```

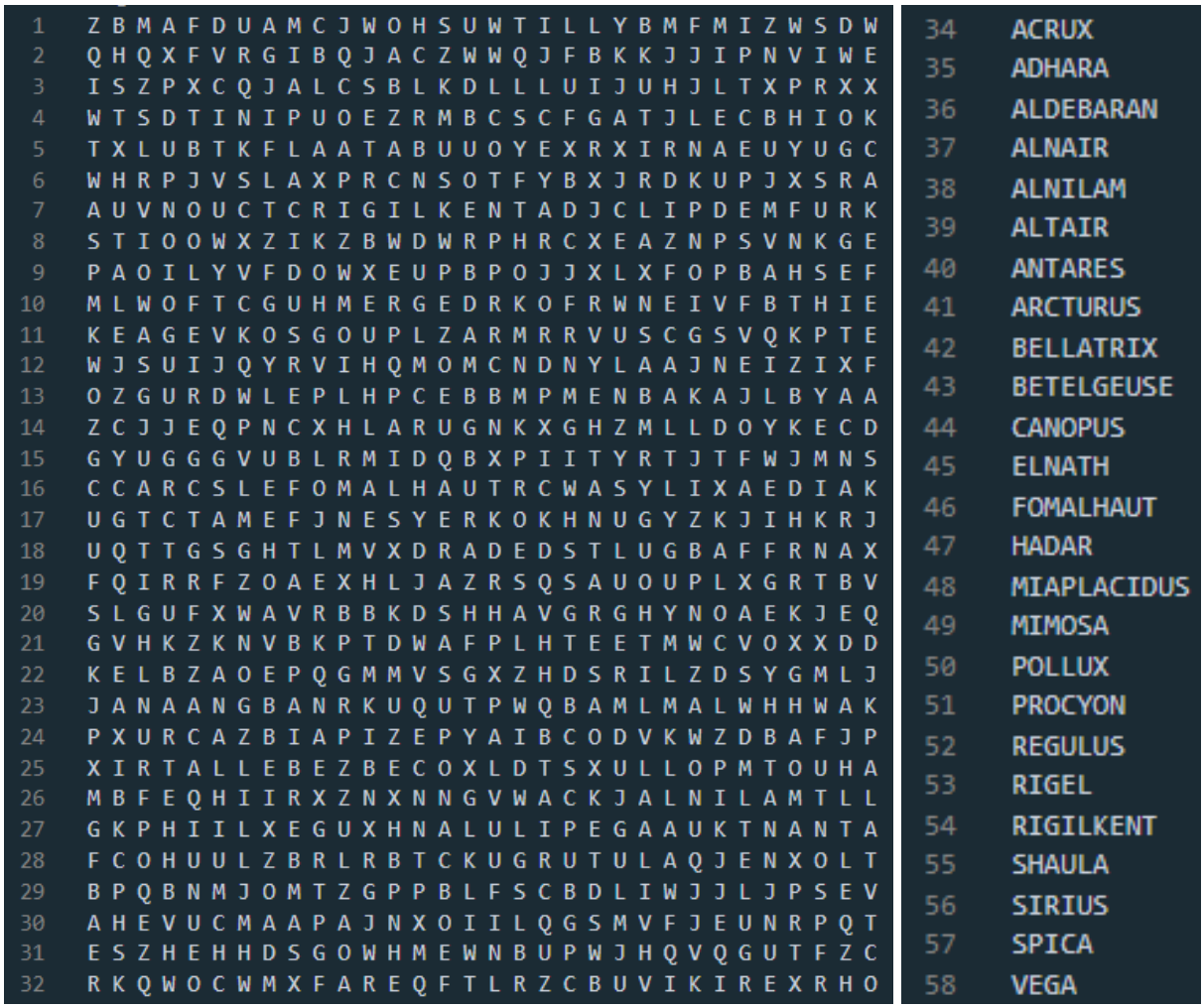
Output:

```
================================================
Input configuration file: trains-medium.txt
------------------------------------------------
trains-medium.txt opened.
Board 24 x 24 loaded successfully.
20 Pattern(s) loaded successfully.
================================================
No. Word Searched Status    Comparison(s)
 1. MALABAR       found               15
 2. RANGGAJATI    found               26
 3. GUMARANG      found              751
 4. CIREMAI       found              338
 5. WIJAYAKUSUMA  found              331
 6. HARINA        found              278
 7. MATARAM       found              118
 8. LODAYA        found               85
 9. KERTANEGARA   found              197
10. PANGANDARAN   found               28
11. SANCAKA       found               62
12. SRIWIJAYA     found              274
13. SINGASARI     found              751
14. JAYABAYA      found              256
15. BANGUNKARTA   found              676
16. BOGOWONTO     found              291
17. KAMANDAKA     found               13
18. KALIGUNG      found              380
19. BRANTAS       found               88
20. PANGRANGO     found              290

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

Pattern found (20/20).
Total time needed: 0.057923s.
Total comparison:  5248.
```

Ukuran Besar

File input: big-stars.txt

Input:



Output:

```
==========================================
Input configuration file: stars-big.txt
------------------------------------------
stars-big.txt opened.
Board 32 x 32 loaded successfully.
25 Pattern(s) loaded successfully.
==========================================
No. Word Searched Status    Comparison(s)
 1. ACRUX         found              1152
 2. ADHARA        found              1107
 3. ALDEBARAN     found              1149
 4. ALNAIR        found               880
 5. ALNILAM       found              1303
 6. ALTAIR        found               578
 7. ANTARES       found               724
 8. ARCTURUS      found                24
 9. BELLATRIX     found              1043
10. BETELGEUSE    found               812
11. CANOPUS       found              1150
12. ELNATH        found              1347
13. FOMALHAUT     found               618
14. HADAR         found               484
15. MIAPLACIDUS   found                26
16. MIMOSA        found               817
17. POLLUX        found              1038
18. PROCYON       found               539
19. REGULUS       found              1005
20. RIGEL         found               721
21. RIGILKENT     found               289
22. SHAULA        found               927
23. SIRIUS        found                42
24. SPICA         found               335
25. VEGA          found               387
```

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

Pattern found (25/25).
Total time needed: 0.073500s.
Total comparison:  18497.
```

File input: big-horses.txt
Input:

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

```
AGNESDIGITAL
AIRGROOVE
BIWAHAYAHIDE
CURRENCHAN
DAIWASCARLET
EISHINFLASH
ELCONDORPASA
FINEMOTION
FUJIKISEKI
GOLDCITY
GRASSWONDER
HARUURARA
HISHIAKEBONO
INESFUJIN
KAWAKAMIPRINCESS
KINGHALO
MARUZENSKY
MATIKANETANNHAUSER
MAYANOTOPGUN
MEISHODOTO
MEJIROMCQUEEN
MIHONOBOURBON
NARITATAISHIN
NICENATURE
OGURICAP
RICESHOWER
SAKURABAKUSHINO
SAKURACHIYONOO
SEIUNSKY
SPECIALWEEK
SUPERCREEK
TMOPERAO
TAIKISHUTTLE
TAMAMOCROSS
TOKAITEIO
VODKA
WINNINGTICKET
YUKINOBIJIN
ZENNOROBROY
```

Output:

```
================================================
Input configuration file: horses-big.txt
------------------------------------------------
horses-big.txt opened.
Board 34 x 34 loaded successfully.
39 Pattern(s) loaded successfully.
================================================
No. Word Searched        Status     Comparison(s)
 1. AGNESDIGITAL         found                1266
 2. AIRGROOVE            found                 854
 3. BIWAHAYAHIDE         found                1411
 4. CURRENCHAN           found                 398
 5. DAIWASCARLET         found                 256
 6. EISHINFLASH          found                  32
 7. ELCONDORPASA         found                 143
 8. FINEMOTION           found                 110
 9. FUJIKISEKI           found                1084
10. GOLDCITY             found                1315
11. GRASSWONDER          found                1201
12. HARUURARA            found                 803
13. HISHIAKEBONO         found                1091
14. INESFUJIN            found                  31
15. KAWAKAMIPRINCESS     found                 774
16. KINGHALO             found                 162
17. MARUZENSKY           found                1348
18. MATIKANETANNHAUSER   found                 808
19. MAYANOTOPGUN         found                 508
20. MEISHODOTO           found                1087
21. MEJIROMCQUEEN        found                 829
22. MIHONOBOURBON        found                 242
23. NARITATAISHIN        found                  29
24. NICENATURE           found                  39
25. OGURICAP             found                1008
26. RICESHOWER           found                 627
27. SAKURABAKUSHINO      found                1267
28. SAKURACHIYONOO       found                 849
29. SEIUNSKY             found                1377
30. SPECIALWEEK          found                  23
31. SUPERCREEK           found                 304
32. TMOPERAO             found                1068
33. TAIKISHUTTLE         found                 779
34. TAMAMOCROSS          found                1252
35. TOKAITEIO            found                 944
36. VODKA                found                 394
37. WINNINGTICKET        found                1315
38. YUKINOBIJIN          found                 523
39. ZENNOROBROY          found                 514
```

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

Pattern found (39/39).
Total time needed: 0.132625s.
Total comparison:  28065.
```

File input: big-cars.txt
Input:

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

```
38 ABARTH
39 ACURA
40 ALFAROMEO
41 ASTONMARTIN
42 AUDI
43 BENTLEY
44 BMW
45 BUICK
46 CADILLAC
47 CHEVROLET
48 CHRYSLER
49 CITROEN
50 DACIA
51 DODGE
52 FERRARI
53 FIAT
54 FORD
55 GMC
56 HONDA
57 HUMMER
58 HYUNDAI
59 INFINITI
60 ISUZU
61 JAGUAR
62 JEEP
63 KIA
64 LAMBORGHINI
65 LANCIA
66 LANDROVER
67 LEXUS
68 LINCOLN
69 LOTUS
70 MASERATI
71 MAZDA
72 MERCEDESBENZ
73 MERCURY
74 MINI
75 MITSUBISHI
76 NISSAN
77 OPEL
78 PEUGEOT
79 PONTIAC
80 PORSCHE
```

Output:

```
========================================
Input configuration file: cars-big.txt
----------------------------------------
cars-big.txt opened.
Board 36 x 36 loaded successfully.
43 Pattern(s) loaded successfully.
========================================
No. Word Searched Status    Comparison(s)
 1. ABARTH        found                62
 2. ACURA         found                23
 3. ALFAROMEO     found              1158
 4. ASTONMARTIN   found              1681
 5. AUDI          found              1750
 6. BENTLEY       found               435
 7. BMW           found               143
 8. BUICK         found               567
 9. CADILLAC      found               626
10. CHEVROLET     found               275
11. CHRYSLER      found               344
12. CITROEN       found               767
13. DACIA         found               888
14. DODGE         found               496
15. FERRARI       found              1416
16. FIAT          found               979
17. FORD          found               815
18. GMC           found               246
19. HONDA         found              1009
20. HUMMER        found              1348
21. HYUNDAI       found               620
22. INFINITI      found              1605
23. ISUZU         found               225
24. JAGUAR        found               740
25. JEEP          found               596
26. KIA           found                41
27. LAMBORGHINI   found               430
28. LANCIA        found              1537
29. LANDROVER     found              1775
30. LEXUS         found              1127
31. LINCOLN       found              1774
32. LOTUS         found              1768
33. MASERATI      found              1256
34. MAZDA         found               639
35. MERCEDESBENZ  found               765
36. MERCURY       found               266
37. MINI          found              1587
38. MITSUBISHI    found              1003
39. NISSAN        found               244
40. OPEL          found               565
41. PEUGEOT       found              1231
42. PONTIAC       found               257
43. PORSCHE       found               684
```

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

Pattern found (43/43).
Total time needed: 0.128131s.
Total comparison:  35763.
```

Tautan Repository Github

Berikut tautan yang dapat diakses untuk menuju ke kode program.
https://github.com/mhelmih/Tucil1_13520014

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	✔	