Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Disusun oleh:
Kenny Benaya Nathan
13521023

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

DAFTAR ISI

BAB I
ALGORITMA BRUTE FORCE

1.1 Algoritma Brute Force

Algoritma *Brute Force* adalah algoritma dengan pendekatan berprinsip *straightforward* untuk memecahkan suatu masalah. Sesuai dengan prinsipnya, algoritma ini bekerja dengan mencoba seluruh kemungkinan yang ada sehingga bisa menemukan solusi sesuai dengan definisi dari permasalahan yang ada. Karena cara kerjanya, algoritma ini bisa dipakai untuk menyelesaikan hampir semua permasalahan yang ada, namun akan merugikan kompleksitas dari programnya karena memakan ruang dan waktu yang sangat banyak sehingga bisa menjadi tidak efisien untuk kasus-kasus yang besar.

1.2 Pendekatan Algoritma *Brute Force* Pada Permainan Kartu 24

Permainan teka-teki 24 adalah permainan yang mengharuskan seseorang untuk mencari operasi-operasi yang bisa dilakukan pada 4 angka yang diberikan sehingga akan menghasilkan angka 24. Biasanya, permainan ini dilakukan dengan menggunakan kartu untuk memberikan 4 angka tersebut. Algoritma *brute force* bisa diaplikasikan untuk mencari seluruh solusi yang ada jika diberikan 4 kartu.

Untuk mencari seluruh solusi yang ada pada permainan 24 dengan pendekatan *brute force,* algoritma yang digunakan adalah sebagai berikut:

1. Setelah menerima 4 kartu yang akan dipakai, program akan melakukan permutasi untuk mencari seluruh kemungkinan cara penyusunan 4 kartu tersebut. Total paling banyak ada 24 kemungkinan cara untuk menyusun 4 kartu. Jika ada kemungkinan yang sama (karena ada kartu yang sama), maka program akan membiarkan kemungkinan tersebut dan tidak disimpan.
2. Pada setiap kemungkinan penyusunan kartu tersebut, program akan melakukan seluruh operasi yang bisa dilakukan. Pada kasus ini, operasi-operasi yang bisa dilakukan adalah operasi tambah (+), kurang (-), kali (*), dan bagi (/), beserta dengan tanda kurung (()) untuk menandakan alur perhitungannya. Dengan w, x, y, z adalah elemen kartu dan OP adalah operator, kemungkinan alur perhitungan yang bisa dilakukan adalah:
    a. ((w OP x) OP y) OP z
    b. (w OP (x OP y)) OP z
    c. (w OP x) OP (y OP z)
    d. w OP (x OP (y OP z))
    e. w OP ((x OP y) OP z)

Sebenarnya, ada kemungkinan alur perhitungan lain seperti w OP x OP y OP z yang tidak menggunakan tanda kurung sama sekali, namun penulis memutuskan untuk tidak memasukkannya ke dalam algoritma. Misalkan w + x + y + z akan memiliki alur yang berbeda

dengan w + x + y * z. Namun, sesuai dengan prinsip asosiatif (hasil perhitungan tiga angka atau lebih tidak bergantung pada pengelompokan dari angka yang dioperasikan), maka w + x + y + z sama saja dengan ((w + x) + y) + z). Sama halnya dengan w + x + y * z yang sama saja dengan (w + x) + (y * z). Maka dari itulah, penulis langsung melakukan operasi sesuai dengan alur pada lima pengelompokan di atas.

3. Jika perhitungan tersebut menghasilkan angka 24, maka program akan menyimpan perhitungannya.

BAB II
SOURCE PROGRAM

Pada penyelesaian masalah ini, penulis membuat program dalam bahasa pemrograman C++ versi C++14 yang sudah dibagi ke dalam satu file program utama dengan extensi .cpp dan enam file *header* program pembantu dengan ekstensi .hpp. Berikut adalah *source code* dari program *24 Puzzle Solver*.

2.1 main.cpp

```
// MAIN PROGRAM

#include <iostream>
#include <vector>
#include <sstream>
#include <ctime>
#include <chrono>
#include <fstream>
#include "file.hpp"
#include "input.hpp"
#include "operator.hpp"
#include "permute.hpp"
#include "menu.hpp"

using namespace std;

void startMenu(vector <double> &cards, vector<vector<double>> &permutations,
vector<string> &results, int &total)
{
    using Clock = std::chrono::high_resolution_clock;

    // initiate main menu
    mainMenu(cards);

    // start clock
    auto start = Clock::now();

    // permutate user's input card
    permutasi(cards, 0, 4, permutations);

    // calculate all possible results
    for (int i = 0; i < permutations.size(); i++)
    {
        count1(permutations[i], total, results);
        count2(permutations[i], total, results);
        count3(permutations[i], total, results);
        count4(permutations[i], total, results);
        count5(permutations[i], total, results);
    }

    // print results
    cout << results.size() << " solution(s) found." << endl;
```

```
    for (int i = 0; i < results.size(); i++)
    {
        cout << results[i] << endl;
    }
    auto end = Clock::now();
    auto duration = std::chrono::duration_cast<std::chrono::milliseconds>(end
- start);
    cout << endl << "Time taken: " << duration.count() << "ms" << endl;

    saveResults(cards, results, total);
}

int main()
{
    // variable declaration
    int total = 0;
    vector<string> results;
    vector<double> cards;
    vector<vector<double>> permutations;

    // main program
    startMenu(cards, permutations, results, total);

    // back to main menu
    int prompt = backToMenu();
    if (prompt == 0)
    {
        cout << endl;
        cout << "Thank you for using 24 PUZZLE SOLVER." << endl;
        cout << "We'll still be here if you need us (again)." << endl;
        exit(0);
    }
    else
    {
        main();
    }

    return 0;
}
```

2.2 clearScreen.hpp

```
#ifdef _WIN32

// Windows
#define CLEAR "cls"
#else
// Other OS
#define CLEAR "clear"

#endif
```

2.3 file.hpp

```
// SAVING RESULTS

#ifndef FILE_HPP
#define FILE_HPP

using namespace std;

void saveResults(vector<double> cards, vector<string> results, int &total)
{
    while (true)
    {
        int choice;
        cout << "Do you want to save your results? (1/0): ";
        cin >> choice;
        if (choice == 1)
        {
            string fileName;
            cout << endl << "Insert the name of your file: ";
            cin >> fileName;
            while (fileName.length() == 0)
            {
                cout << "Please insert the name for your file: ";
                cin >> fileName;
            }

            ofstream userFile;
            userFile.open("test/" + fileName + ".txt");
            userFile << "24 PUZZLE SOLVER" << endl << endl << "Your Cards: ";
            for (int i = 0; i < cards.size(); i++)
            {
                if (cards[i] == 1)
                {
                    userFile << "A ";
                }
                else if (cards[i] == 11)
                {
                    userFile << "J ";
                }
                else if (cards[i] == 12)
                {
                    userFile << "Q ";
                }
                else if (cards[i] == 13)
                {
                    userFile << "K ";
                }
                else
                {
                    userFile << cards[i] << " ";
                }
            }

            // write solutions
            userFile << endl << results.size() << " solutions found." << endl
```

```
<< endl;
            for (int j = 0; j < results.size(); j++)
            {
                userFile << results[j] << endl;
            }
            userFile.close();
            break;
        }
        else if (choice == 0)
        {
            break;
        }
        else
        {
            cout << "Invalid input. Please try again." << endl;
        }
    }
}

#endif
```

2.4 input.hpp

```
// INPUT CARDS

#ifndef INPUT_HPP
#define INPUT_HPP

using namespace std;

// Converts string to double
double stodouble(string kar)
{
    double n;
    if (kar == "A")
    {
        n = 1;
    }
    else if (kar == "J")
    {
        n = 11;
    }
    else if (kar == "Q")
    {
        n = 12;
    }
    else if (kar == "K")
    {
        n = 13;
    }
    else
    {
        n = stoi(kar);
    }
```

```
    return n;
}

// Checks validity of user's card
bool checkCard(string kartu)
{

    if (isdigit(kartu[0]) || kartu == "10")
    {
        if (kartu == "0" || kartu == "1" || kartu.length() > 2)
        {
            return false;
        }
        else
        {
            return true;
        }
    }
    else
    {
        if (kartu == "A" || kartu == "J" || kartu == "Q" || kartu == "K")
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

// User inputs manually
void manualInput(vector<double> &cards)
{
    // user input
    string inputCard[4];
    bool inputValid = false;
    // Check if the cards are valid
    while (!inputValid)
    {
        cout << "Enter 4 cards (ex: A 9 4 K): ";
        cin >> inputCard[0] >> inputCard[1] >> inputCard[2] >> inputCard[3];
        inputValid = checkCard(inputCard[0]) && checkCard(inputCard[1]) &&
checkCard(inputCard[2]) && checkCard(inputCard[3]);
        if (!inputValid)
        {
            cout << inputCard[0] << endl;
            cout << "Invalid input. Please try again." << endl;
        }
    }
    cout << "Input Accepted." << endl << endl;

    // Convert the cards to int
    for (int i = 0; i < 4; i++)
    {
        cards.push_back(stodouble(inputCard[i]));
```

```
    }
}

// Generate random cards
void randomInput(vector<double> &cards)
{
    srand(time(NULL));
    cout << "Generating random cards..." << endl << endl;
    nanosleep((const struct timespec[]){{0, 500000000L}}, NULL);
    cout << "Cards generated." << endl;
    for (int i = 0; i < 4; i++)
    {
        cards.push_back((rand() % 13) + 1);
        if (cards[i] == 1)
        {
            cout << "A ";
        }
        else if (cards[i] == 11)
        {
            cout << "J ";
        }
        else if (cards[i] == 12)
        {
            cout << "Q ";
        }
        else if (cards[i] == 13)
        {
            cout << "K ";
        }
        else
        {
            cout << cards[i] << " ";
        }
    }
    cout << endl;
}
#endif
```

2.5 menu.hpp

```
// MAIN MENU

#include "clearScreen.hpp"
#ifndef MENU_HPP
#define MEMU_HPP

using namespace std;

// Banner
void banner()
{
    system(CLEAR);
    cout <<
"================================================================================
```

```
================================================" << endl << endl;
    cout << ".------..------.
.------..------..------..------..------..------.
.------..------..------..------..------..------." << endl;
    cout << "|2.--. ||4.--. | .-.   |P.--. ||U.--. ||Z.--. ||Z.--. ||L.--.
||E.--. | .-.   |S.--. ||O.--. ||L.--. ||V.--. ||E.--. ||R.--. |" << endl;
    cout << "| (\\/) || :/\\: |((5))  | :/\\: || (\\/) || :(): || :(): ||
:/\\: || (\\/) |((5))  | :/\\: || :/\\: || :/\\: || :(): || (\\/) || :(): |"
<< endl;
    cout << "| :\\/: || :\\/: | '-.-. | (__) || :\\/: || ()() || ()() || (__)
|| :\\/: | '-.-. | :\\/: || :\\/: || (__) || ()() || :\\/: || ()() |" << endl;
    cout << "| '--'2|| '--'4|  ((1))| '--'P|| '--'U|| '--'Z|| '--'Z|| '--'L||
'--'E|  ((1))| '--'S|| '--'O|| '--'L|| '--'V|| '--'E|| '--'R|" << endl;
    cout << "`------'`------'   '-'
`------'`------'`------'`------'`------'`------'   '-'
`------'`------'`------'`------'`------'`------'" << endl << endl;
    cout <<
"================================================================================
================================================" << endl << endl;
    cout << "WELCOMT TO 24 PUZZLE SOLVER" << endl;
    cout << "Your lovely saviour when you play 24 puzzle card game with your
friends and you know you can't count <3" << endl << endl;
}

// Menu for user's input method
int inputMenu()
{
    cout << "How would you like to input your cards?" << endl;
    cout << "1. Input it Yourself" << endl;
    cout << "2. Generate Random Cards" << endl;
    cout << "3. Exit" << endl << endl;
    while (true)
    {
        int choice;
        cout << "Your choice (ex: 1): ";
        cin >> choice;
        if (choice == 1 || choice == 2 || choice == 3)
        {
            return choice;
        }
        else
        {
            cout << "Invalid input, please try again." << endl;
        }
    }
}

// Back to starting menu
int backToMenu()
{
    int input;
    while (true)
    {
        cout << "Do you want to go back to main menu? (1/0): ";
        cin >> input;
        if (input == 1 || input == 0)
```

```
        {
            return input;
        }
        else
        {
            cout << "Invalid input" << endl;
        }
    }
}

// Main menu
void mainMenu(vector<double> &cards)
{
    banner();

    switch(inputMenu())
    {
        case 1:
            manualInput(cards);
            break;
        case 2:
            randomInput(cards);
            break;
        case 3:
            exit(0);
    }
}

#endif
```

2.6 operator.hpp

```
// FUNCTIONS FOR CALCULATING 24 POINTS

#ifndef OPERATOR_HPP
#define OPERATOR_HPP

using namespace std;

char opes[4] = {'+', '-', '*', '/'};

// Format double to string
string format(double n)
{
    stringstream ss;
    ss.precision(2);
    ss << n;

    string s = ss.str();
    return s;
}

// Returns binary operator
double op(double a, int ope, double b)
```

```
{
    switch (ope)
    {
    case 0:
        return a + b;
    case 1:
        return a - b;
    case 2:
        return a * b;
    case 3:
        return a / b;
    default:
        break;
    }
}

// ((a OP b) OP c) OP d
void count1(vector<double> arr, int total, vector<string> &result)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                double temp = op(op(op(arr[0], i, arr[1]), j, arr[2]), k,
arr[3]);
                if (temp == 24)
                {
                    result.push_back("((" + format(arr[0]) + " " + opes[i] + "
" + format(arr[1]) + ")" + " " + opes[j] + " " + format(arr[2]) + ")" + " " +
opes[k] + " " + format(arr[3]));
                    total++;
                }
            }
        }
    }
}

// (a OP (b OP c)) OP d
void count2(vector<double> arr, int total, vector<string> &result)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                double temp = op(op(arr[0], i, op(arr[1], j, arr[2])), k,
arr[3]);
                if (temp == 24)
                {
                    result.push_back("(" + format(arr[0]) + " " + opes[i] + "
" + "(" + format(arr[1]) + " " + opes[j] + " " + format(arr[2]) + "))" + " " +
opes[k] + " " + format(arr[3]));
                    total++;
```

```
                }
            }
        }
    }
}

// (a OP b) OP (c OP d)
void count3(vector<double> arr, int total, vector<string> &result)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                double temp = op(op(arr[0], i, arr[1]), j, op(arr[2], k,
arr[3]));
                if (temp == 24)
                {
                    result.push_back("(" + format(arr[0]) + " " + opes[i] + "
" + format(arr[1]) + ")" + " " + opes[j] + " " + "(" + format(arr[2]) + " " +
opes[k] + " " + format(arr[3]) + ")");
                    total++;
                }
            }
        }
    }
}

// a OP ((b OP c) OP d)
void count4(vector<double> arr, int total, vector<string> &result)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                double temp = op(arr[0], i, op(op(arr[1], j, arr[2]), k,
arr[3]));
                if (temp == 24)
                {
                    result.push_back(format(arr[0]) + " " + opes[i] + " " +
"((" + format(arr[1]) + " " + opes[j] + " " + format(arr[2]) + ")" + " " +
opes[k] + " " + format(arr[3]) + ")");
                    total++;
                }
            }
        }
    }
}

// a OP (b OP (c OP d))
void count5(vector<double> arr, int total, vector<string> &result)
{
    for (int i = 0; i < 4; i++)
```

```
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                double temp = op(arr[0], i, op(arr[1], j, op(arr[2], k,
arr[3])));
                if (temp == 24)
                {
                    result.push_back(format(arr[0]) + " " + opes[i] + " " +
"(" + format(arr[1]) + " " + opes[j] + " " + "(" + format(arr[2]) + " " +
opes[k] + " " + format(arr[3]) + "))");
                    total++;
                }
            }
        }
    }
}
#endif
```

2.7 permute.hpp

```
// PERMUTATION ALGORITHM

#ifndef PERMUTE_HPP
#define PERMUTE_HPP
using namespace std;

// Check if the permutation is not yet inserted
bool validPermute(vector<double> arr, int start, int end,
vector<vector<double>> result)
{
    for (int i = 0; i < result.size(); i++)
    {
        // cout << result[i][0];
        if (arr == result[i])
        {
            return false;
        }
    }
    return true;
}

void permutasi(vector<double> arr, int start, int end, vector<vector<double>>
&result)
{
    if (start == end)
    {
        if (validPermute(arr, start, end, result))
        {
            result.push_back(arr);
        }
    } else
    {
```

```
        for (int i = start; i < end; i++)
        {
            swap(arr[start], arr[i]);
            permutasi(arr, start + 1, end, result);
            swap(arr[start], arr[i]);
        }
    }
}

#endif
```

BAB III
HASIL PENGUJIAN PROGRAM

3.1 Uji 1 (6 6 6 6)

3.2 Uji 2 (7 8 9 10)

3.3 Uji 3 (7 8 9 10)

```
How would you like to input your cards?
1. Input it Yourself
2. Generate Random Cards
3. Exit

Your choice (ex: 1): 1
Enter 4 cards (ex: A 9 4 K): Q J K A
Input Accepted.

32 solution(s) found.
(12 * (13 - 11)) * 1
(12 * (13 - 11)) / 1
12 * ((13 - 11) * 1)
12 * ((13 - 11) / 1)
12 * (13 - (11 * 1))
12 * (13 - (11 / 1))
12 * ((13 * 1) - 11)
12 * ((13 / 1) - 11)
12 * (13 - (1 * 11))
(12 * 1) * (13 - 11)
(12 / 1) * (13 - 11)
12 * ((1 * 13) - 11)
12 * (1 * (13 - 11))
12 / (1 / (13 - 11))
((13 - 11) * 12) * 1
((13 - 11) * 12) / 1
(13 - 11) * (12 * 1)
(13 - 11) * (12 / 1)
((13 - 11) * 1) * 12
((13 - 11) / 1) * 12
(13 - (11 * 1)) * 12
(13 - (11 / 1)) * 12
(13 - 11) * (1 * 12)
(13 - 11) / (1 / 12)
((13 * 1) - 11) * 12
((13 / 1) - 11) * 12
(13 - (1 * 11)) * 12
((1 * 13) - 11) * 12
(1 * (13 - 11)) * 12
1 * ((13 - 11) * 12)
(1 * 12) * (13 - 11)
1 * (12 * (13 - 11))

Time taken: 9ms
Do you want to save your results? (1/0): 1
```

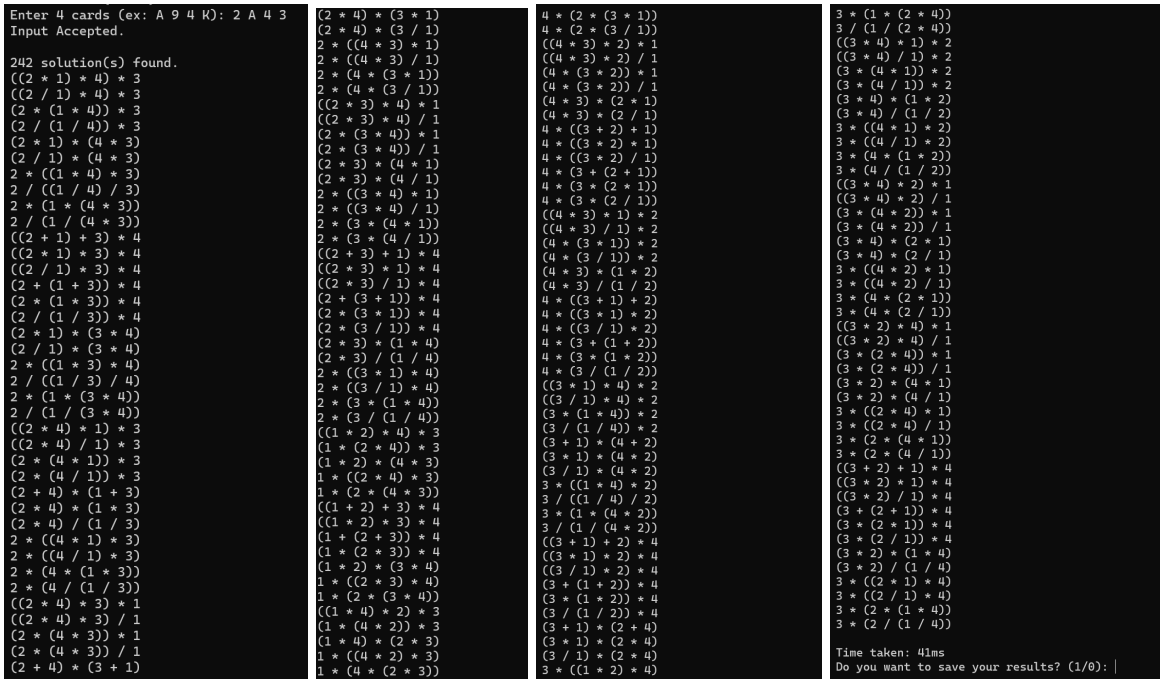
3.4 Uji 4 (10 3 9 6)

```
Your choice (ex: 1): 1
Enter 4 cards (ex: A 9 4 K): 10 3 9 6
Input Accepted.

34 solution(s) found.
((10 + 3) - 9) * 6
((10 / 3) * 9) - 6
(10 + (3 - 9)) * 6
(10 / (3 / 9)) - 6
((10 - 9) + 3) * 6
((10 * 9) / 3) - 6
(10 - (9 - 3)) * 6
(10 * (9 / 3)) - 6
(10 - 6) * (9 - 3)
((3 + 10) - 9) * 6
(3 + (10 - 9)) * 6
((3 - 9) + 10) * 6
(3 - (9 - 10)) * 6
(3 - 9) * (6 - 10)
((9 / 3) * 10) - 6
(9 / (3 / 10)) - 6
(9 - 3) * (10 - 6)
((9 * 10) / 3) - 6
(9 * (10 / 3)) - 6
(9 * 6) - (10 * 3)
9 * (6 - (10 / 3))
(9 * 6) - (3 * 10)
6 * ((3 - 9) + 10)
6 * (3 - (9 - 10))
6 * ((3 + 10) - 9)
6 * (3 + (10 - 9))
(6 * 9) - (3 * 10)
(6 * 9) - (10 * 3)
6 * ((10 - 9) + 3)
6 * (10 - (9 - 3))
(6 - (10 / 3)) * 9
(6 - 10) * (3 - 9)
6 * ((10 + 3) - 9)
6 * (10 + (3 - 9))

Time taken: 9ms
Do you want to save your results? (1/0):
```

3.5 Uji 5 (2 A 4 3)



3.6 Uji Acak 1 (4 7 K K)

3.7 Uji Acak 2 (5 9 J Q)

3.8 Uji Acak 3 (Q 6 6 J)

```
================================================================================================================================================

 .------..------.       .------..------..------..------..------..------.       .------..------..------..------..------..------..------.
|2.--. ||4.--. | .-.   |P.--. ||U.--. ||Z.--. ||Z.--. ||L.--. ||E.--. | .-.   |S.--. ||O.--. ||L.--. ||V.--. ||E.--. ||R.--. |
| (\/) || :/\: |((5))  | :/\: || (\/) || :(): || :(): || :/\: || (\/) |((5))  | :/\: || :/\: || :/\: || :(): || (\/) || :(): |
| :\/: || :\/: | '-.-. | (__) || :\/: || ()() || ()() || (__) || :\/: | '-.-. | :\/: || :\/: || (__) || ()() || :\/: || ()() |
| '--'2|| '--'4|  ((1))| '--'P|| '--'U|| '--'Z|| '--'Z|| '--'L|| '--'E|  ((1))| '--'S|| '--'O|| '--'L|| '--'V|| '--'E|| '--'R|
 `------'`------'   '-' `------'`------'`------'`------'`------'`------'   '-' `------'`------'`------'`------'`------'`------'

================================================================================================================================================

WELCOMT TO 24 PUZZLE SOLVER
Your lovely saviour when you play 24 puzzle card game with your friends and you know you can't count <3

How would you like to input your cards?
1. Input it Yourself
2. Generate Random Cards
3. Exit

Your choice (ex: 1): 2
Generating random cards...

Cards generated.
Q 6 6 J
12 solution(s) found.
(12 + (6 / 6)) + 11
12 + ((6 / 6) + 11)
(12 + 11) + (6 / 6)
12 + (11 + (6 / 6))
((6 / 6) + 12) + 11
(6 / 6) + (12 + 11)
((6 / 6) + 11) + 12
(6 / 6) + (11 + 12)
(11 + (6 / 6)) + 12
11 + ((6 / 6) + 12)
(11 + 12) + (6 / 6)
11 + (12 + (6 / 6))

Time taken: 11ms
Do you want to save your results? (1/0):
```

LAMPIRAN

Tautan *Repository* : https://github.com/kennypanjaitan/Tucil1_13521023

Tabel *Checklist Goal*

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	V	
2	Program berhasil running	V	
3	Program dapat membaca input / generate sendiri dan memberikan luaran	V	
4	Solusi yang diberikan program memenuhi (berhasil mencapai 24)	V	
5	Program dapat menyimpan solusi dalam file teks	V	