

**Laporan Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II Tahun 2022/2023**

Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force*



Disusun oleh:

Azmi Hasna Zahrani

13521006

K03

**TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2023**

DAFTAR ISI

DAFTAR ISI	1
BAB I	2
ALGORITMA BRUTE FORCE	2
1.1 Algoritma Brute Force	2
1.2 Penggunaan Algoritma Brute Force dalam Menyelesaikan Permainan Kartu 24	2
BAB II	4
SOURCE PROGRAM DALAM BAHASA C++	4
2.1 File Main.cpp	4
2.2 File Function.h	6
BAB III	13
HASIL PENGUJIAN	13
3.1 Hasil Pengujian 1	13
3.2 Hasil Pengujian 2	15
3.3 Hasil Pengujian 3	18
3.4 Hasil Pengujian 4	19
3.5 Hasil Pengujian 5	20
3.6 Hasil Pengujian 6	20
3.7 Hasil Pengujian 7	21
BAB IV	25
LAMPIRAN	25
4.1 Tautan Repository	25

BAB I

ALGORITMA *BRUTE FORCE*

1.1 Algoritma *Brute Force*

Algoritma *Brute Force* merupakan salah satu algoritma yang kerap digunakan untuk memecahkan sebuah masalah. Algoritma ini bekerja dengan pendekatan yang *straightforward* sehingga dapat memecahkan masalah secara langsung dan sederhana. Karena pendekatannya yang *straightforward*, algoritma ini dinilai kurang efisien untuk menyelesaikan permasalahan yang memiliki ukuran masukan yang besar. Meskipun begitu, *Brute Force* acap kali digunakan sebagai pembanding untuk algoritma lain yang lebih efisien.

Algoritma *brute force* memiliki beberapa kelebihan, antara lain dapat digunakan untuk menyelesaikan sebagian besar masalah, sederhana, mudah dimengerti, dapat menghasilkan algoritma untuk beberapa penyelesaian masalah penting, serta dapat menghasilkan algoritma baku. Meskipun algoritma ini memiliki kekurangan seperti tidak efisien dan lambat untuk masukan yang besar, terdapat beberapa masalah yang hanya dapat diselesaikan menggunakan algoritma ini. Bahkan, algoritma ini dapat menjadi alternatif ketika ragu dengan algoritma lain.

1.2 Penggunaan Algoritma *Brute Force* dalam Menyelesaikan Permainan Kartu 24

Permainan kartu 24 mengharuskan seseorang untuk membuat operasi matematika dengan empat angka yang muncul pada kartu sehingga menghasilkan nilai 24. Untuk menghasilkan nilai 24, terdapat berbagai macam kemungkinan operasi matematika yang ada. Kemungkinan-kemungkinan operasi tersebut dapat dimodelkan dengan algoritma *brute force*. Dalam hal ini, penulis membuat program dalam bahasa pemrograman C++ dengan beberapa proses yang ada di dalamnya sehingga didapat semua kemungkinan operasi. Adapun langkah-langkah yang ada dalam program yang dibuat adalah sebagai berikut.

Mulanya, program membaca file input yang berupa angka pada kartu remi, yaitu dapat berupa A (*Ace*), 2, 3, 4, 5, 6, 7, 8, 9, 10, J (*Jack*), Q (*Queen*), dan K (*King*), yang masing-masing memiliki nilai berurut dari 1 sampai 13. Karena terdapat beberapa input yang berupa huruf sehingga tidak bisa langsung dioperasikan, program akan menerjemahkan huruf ke dalam angka sehingga dapat dilakukan operasi matematika. Kemudian, keempat angka tersebut akan melalui operasi permutasi untuk menentukan posisi angka dan simbol yang ada. Simbol-simbol matematika yang digunakan adalah +, -, *, /, dan (). Untuk posisi simbol kurung, penulis

menuliskannya langsung pada program. Namun, untuk simbol lain digunakan perulangan untuk menempatkannya di setiap kemungkinan posisi, begitu pula untuk posisi keempat angka. Sembari memposisikan angka dan simbol, pada program terdapat proses perhitungan dengan syarat kondisi. Apabila hasil yang didapat dari perhitungan tersebut sama dengan 24, maka posisi angka dan simbol yang dimaksud akan disimpan, dihitung, dan ditampilkan ketika permutasi selesai. Pada akhir program, akan ditampilkan jumlah operasi dari angka inputan yang dapat menghasilkan nilai 24, operasi itu sendiri, serta opsi untuk menyimpan jawaban. Apabila dipilih opsi untuk menyimpan jawaban, angka masukan, jumlah operasi, dan operasi yang menghasilkan nilai 24 akan disimpan dalam format file txt.

BAB II

SOURCE PROGRAM DALAM BAHASA C++

Pada program yang penulis buat, terdapat dua buah file program yaitu Main.cpp dan Function.h.

2.1 File Main.cpp

[illegible]

```

    cout << "2. No\n";
    cout << "Input: ";
    cin >> input2;
    if (input2 == 1){
        main();
    } else if (input2 == 2){
        cout << "Thank you for using this program\n";
    } else {
        cout << "Invalid input\n";
        cout << "Thank you for using this program\n";
    }
} else if (input == 2){
    random_input();
    cout << "Do you want to play again?\n";
    cout << "1. Yes\n";
    cout << "2. No\n";
    cout << "Input: ";
    cin >> input2;
    if (input2 == 1){
        main();
    } else if (input2 == 2){
        cout << "Thank you for using this program\n";
    } else {
        cout << "Invalid input\n";
        cout << "Thank you for using this program\n";
    }
} else {
    cout << "Invalid input\n";
    cout << "Do you want to try again?\n";
    cout << "1. Yes\n";
    cout << "2. No\n";
    cout << "Input: ";
    cin >> input3;
    if (input3 == 1){
        main();
    } else if (input3 == 2){
        cout << "Thank you for using this program\n";
    } else {
        cout << "Invalid input\n";
        cout << "Thank you for using this program\n";
    }
}

```

```

    }

}

return 0;
}

```

2.2 File Function.h

```

#include <iostream>
#include <vector>
#include <string>
#include <cmath>
#include <string>
#include <algorithm>
#include <ctime>
#include <fstream>
using namespace std;

string c1, c2, c3, c4;
char op[4]= {'+', '-', '*', '/'};
vector <int> val;
vector <int> arr;
vector <string> s;
string hsl;

int convert_cards(string c){
    if (c == "A"){
        return 1;
    } else if (c == "J"){
        return 11;
    } else if (c == "Q"){
        return 12;
    } else if (c == "K"){
        return 13;
    } else if (c == "2" || c == "3" || c == "4" || c == "5" || c == "6" ||
c == " 7" || c == "8" || c == "9" || c == "10"){
        return stoi (c);
    }
    else {
        cout << "Invalid input\n";
    }
}

```

```

        return 0;
    }
}

float convert_op(char op[], float left, float right){
    float ans;
    for (int i = 0; i < 4; i++){
        if (op[i] == '+'){
            ans = left + right;
        } else if (op[i] == '-'){
            ans = left - right;
        } else if (op[i] == '*'){
            ans = left * right;
        } else if (op[i] == '/'){
            ans = left / right;
        } return ans;
    }
}

void bracket(vector <int> arr, char op[]){
    time_t start, end;
    start = clock();
    float ans1, ans2, ans3, ans4, ans5, ans6, ans7;
    int count = 0;
    vector <float> ans;
    for (int i = 0; i < 4; i++){
        for (int j = 0; j < 4; j++){
            for (int k = 0; k < 4; k++){
                for (int l = 0; l < 4; l++){
                    for (int m = 0; m < 4; m++){
                        for (int n = 0; n < 4; n++){
                            for (int o = 0; o < 4; o++){
                                if (i != j && i != k && i != l && j != k
&& j != l && k != l){

                                    // ((i op j) op k) op l
                                    ans1 = convert_op(&op[o],
convert_op(&op[n], convert_op(&op[m], (float)arr[i], (float)arr[j]),
(float)arr[k]), (float)arr[l]);

                                    if(abs(ans1-24) < 0.0001){

```



```

hs1 = "(" + to_string(arr[i]) +
op[m] + to_string(arr[j]) + ")" + op[n] + to_string(arr[k]) + ")" + op[o]
+ to_string(arr[l]);

hs1.erase(remove(hs1.begin(),
hs1.end(), ' '), hs1.end());

s.push_back(hs1);
count += 1;
}
// (i op (j op k)) op l
ans2 = convert_op(&op[o],
convert_op(&op[m], (float)arr[i], convert_op(&op[n], (float)arr[j],
(float)arr[k])), (float)arr[l]);
if(abs(ans2-24) < 0.0001){
hs1 = "(" + to_string(arr[i]) +
op[m] + "(" + to_string(arr[j]) + op[n] + to_string(arr[k]) + ")" + op[o]
+ to_string(arr[l]);

hs1.erase(remove(hs1.begin(),
hs1.end(), ' '), hs1.end());

s.push_back(hs1);
count += 1;
}
// i op ((j op k) op l)
ans3 = convert_op(&op[m],
(float)arr[i], convert_op(&op[o], convert_op(&op[n], (float)arr[j],
(float)arr[k]), (float)arr[l]));
if(abs(ans3-24) < 0.0001){
hs1 = to_string(arr[i]) + op[m] +
"(" + to_string(arr[j]) + op[n] + to_string(arr[k]) + ")" + op[o] +
to_string(arr[l]) + ")";

hs1.erase(remove(hs1.begin(),
hs1.end(), ' '), hs1.end());

s.push_back(hs1);
count += 1;
}
// i op (j op (k op l))
ans4 = convert_op(&op[m],
(float)arr[i], convert_op(&op[n], (float)arr[j], convert_op(&op[o],
(float)arr[k], (float)arr[l])));
if(abs(ans4-24) < 0.0001){

```



```

        cout << "-----\n" << "Time
taken: ";
        cout << (end - start) / double(CLOCKS_PER_SEC) << " seconds\n";
    }

void save_file(vector<string> s, string file_name){
    ofstream file("./test/" + file_name + ".txt");
    file << "Cards: " << val[0] << " " << val[1] << " " << val[2] << " "
<< val[3] << endl;
    file << "There are " << s.size() << " solutions\n";
    file << "-----\n";
    for (int i = 0; i < s.size(); i++){
        file << s[i] << endl;
    }
    file.close();
}

void keyboard_input(){
    string save;
    string file_name;
    cout << "Cards must be 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, or A\n";
    cout << "Enter 4 cards: ";
    cin >> c1 >> c2 >> c3 >> c4 ;
    vector<string> cards = {c1, c2, c3, c4};
    bool found = false;
    for (int i = 0; i < 4; i++){
        if (cards[i] == "2" || cards[i] == "3" || cards[i] == "4" ||
cards[i] == "5" || cards[i] == "6" || cards[i] == "7" || cards[i] == "8"
|| cards[i] == "9" || cards[i] == "10" || cards[i] == "J" || cards[i] ==
"Q" || cards[i] == "K" || cards[i] == "A"){
            }
        else {
            cout << "Invalid input\n";
            cout << "Cards must be 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, or
A\n";

            found = true;
        }
    }
    if (found == false){
        val.push_back(convert_cards(c1));
    }
}

```

```

        val.push_back(convert_cards(c2));
        val.push_back(convert_cards(c3));
        val.push_back(convert_cards(c4));
    }
    cout << "Cards: " << val[0] << " " << val[1] << " " << val[2] << " "
<< val[3] << endl;
    bracket(val, op);
    cout << "-----\n";
    cout << "Do you want to save the result to a file? (y/n): ";
    cin >> save;
    if (save == "y" || save == "Y"){
        cout << "Enter file name: ";
        cin >> file_name;
        save_file(s, file_name);
    }
    else if( save == "n" || save == "N"){
    }
    else {
        cout << "Invalid input\n";
    }
}

void random_input(){
    string save;
    string file_name;
    srand(time(0));
    for (int i = 0; i < 4; i++){
        val.push_back(rand() % 13 + 1);
    }
    cout << "Cards: " << val[0] << " " << val[1] << " " << val[2] << " "
<< val[3] << endl;
    bracket(val, op);
    cout << "-----\n";
    cout << "Do you want to save the result to a file? (y/n): ";
    cin >> save;
    if (save == "y" || save == "Y"){
        cout << "Enter file name: ";
        cin >> file_name;
        save_file(s, file_name);
    }
}

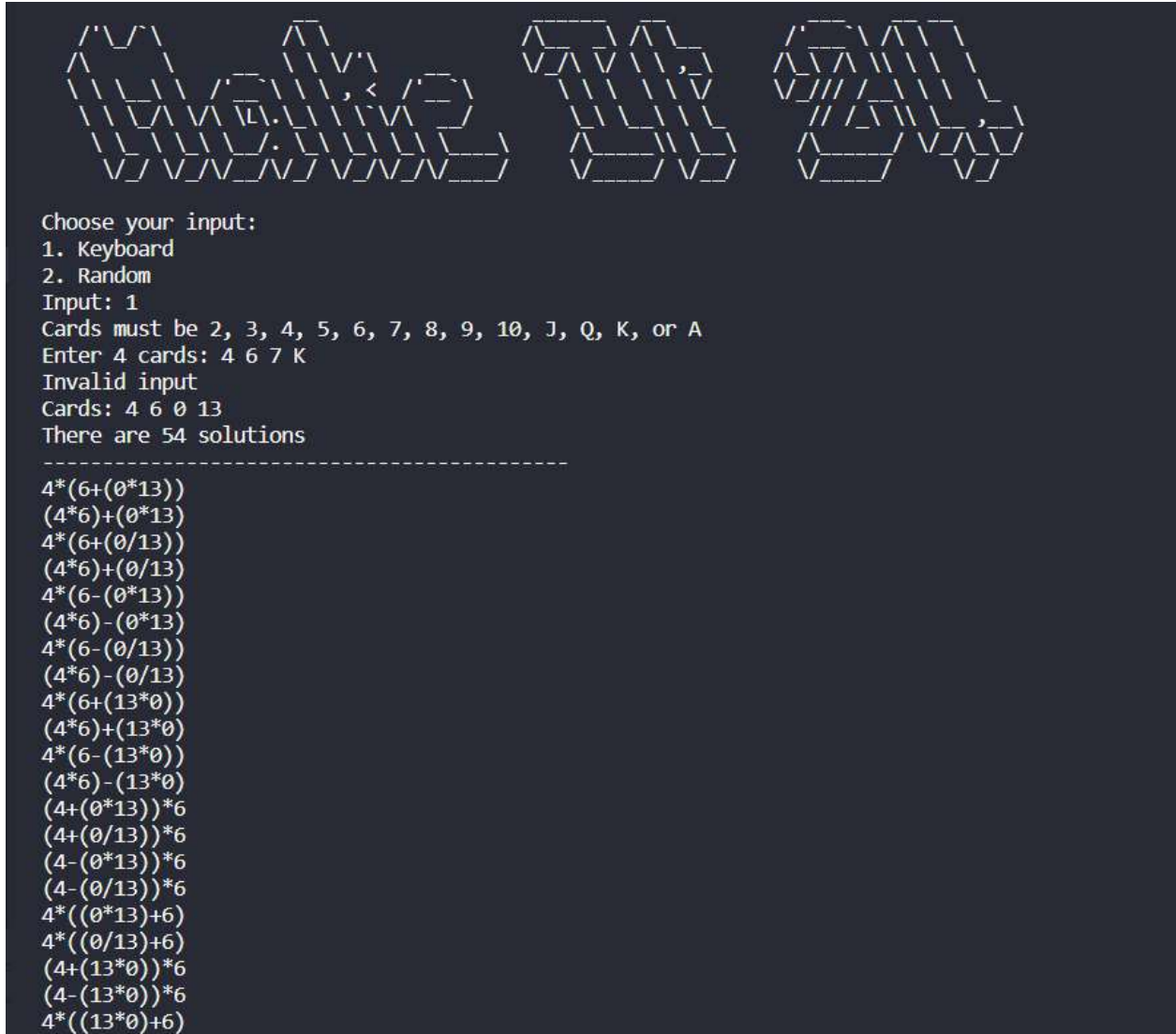
```

```
else if( save == "n" || save == "N"){  
    }  
else {  
    cout << "Invalid input\n";  
}  
}
```

BAB III

HASIL PENGUJIAN

3.1 Hasil Pengujian 1




```

6*(4+(0*13))
(6*4)+(0*13)
6*(4+(0/13))
(6*4)+(0/13)
6*(4-(0*13))
(6*4)-(0*13)
6*(4-(0/13))
(6*4)-(0/13)
6*(4+(13*0))
(6*4)+(13*0)
6*(4-(13*0))
(6*4)-(13*0)
(6+(0*13))*4
(6+(0/13))*4
(6-(0*13))*4
(6-(0/13))*4
6*((0*13)+4)
6*((0/13)+4)
(6+(13*0))*4
(6-(13*0))*4
6*((13*0)+4)
((0*13)+4)*6
(0*13)+(4*6)
((0/13)+4)*6
(0/13)+(4*6)
((0*13)+6)*4
(0*13)+(6*4)
((0/13)+6)*4
(0/13)+(6*4)
((13*0)+4)*6
(13*0)+(4*6)
((13*0)+6)*4
(13*0)+(6*4)
-----
Time taken: 0.012 seconds
-----
Do you want to save the result to a file? (y/n): y
Enter file name: testcase1
Do you want to play again?
1. Yes
2. No

```

3.2 Hasil Pengujian 2



```
Choose your input:
1. Keyboard
2. Random
Input: 2
Cards: 6 3 2 9
There are 80 solutions
-----
((6-3)+9)*2
(6-(3-9))*2
((6+2)/3)*9
(6+2)/(3/9)
((6*2)+3)+9
(6*2)+(3+9)
((6+2)*9)/3
(6+2)*(9/3)
(6-2)*(9-3)
((6*2)+9)+3
(6*2)+(9+3)
((6/2)*9)-3
(6/(2/9))-3
((6+9)-3)*2
(6+(9-3))*2
6*(9-(3+2))
6*((9-3)-2)
6*(9-(2+3))
6*((9-2)-3)
((6*9)/2)-3
(6*(9/2))-3
(3+(6*2))+9
3+((6*2)+9)
(3+(2*6))+9
3+((2*6)+9)
(3-(2/6))*9
```

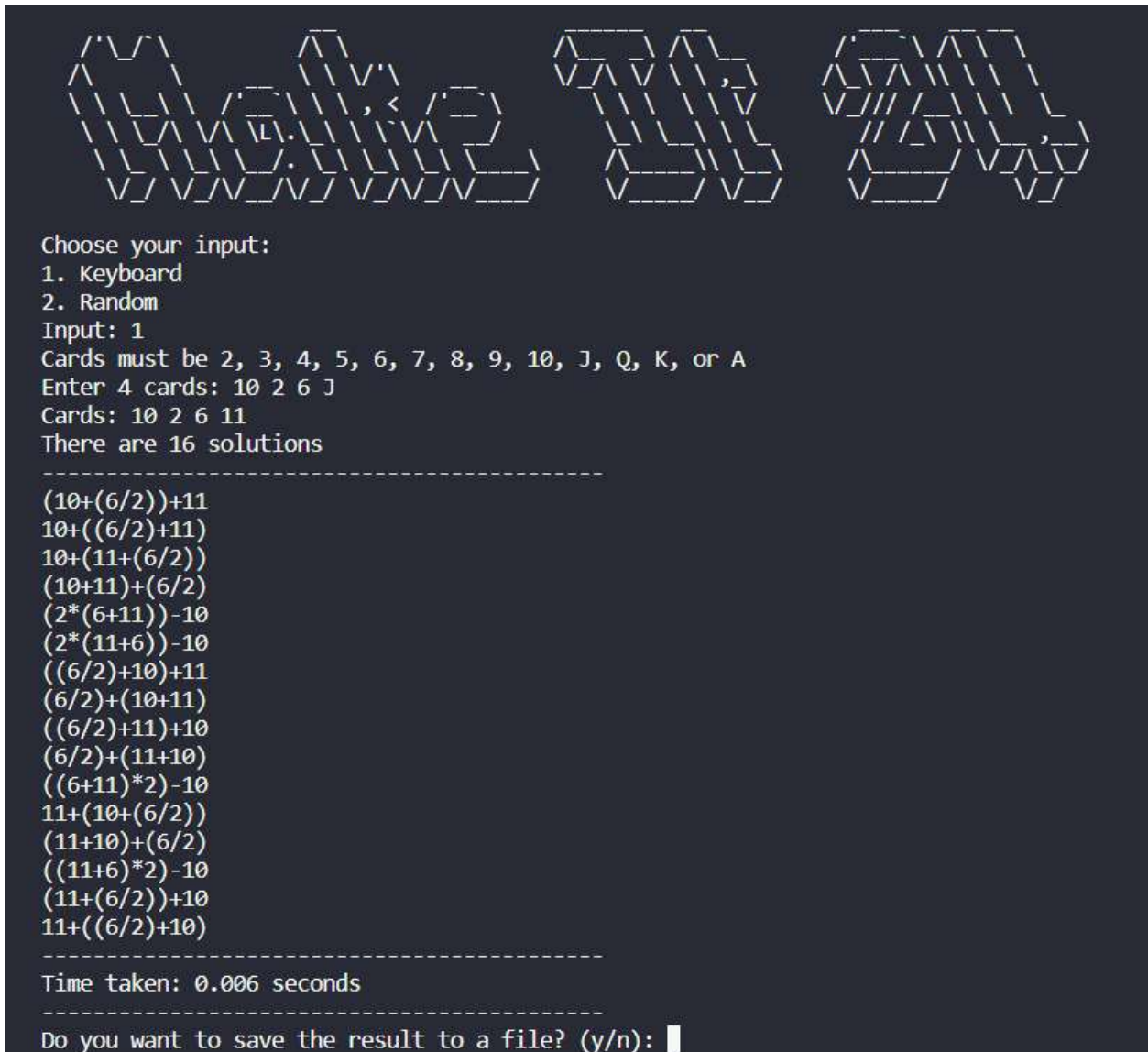

$3+(9+(6*2))$
 $(3+9)+(6*2)$
 $(3*9)-(6/2)$
 $3+(9+(2*6))$
 $(3+9)+(2*6)$
 $(3-9)*(2-6)$
 $((2+6)/3)*9$
 $(2+6)/(3/9)$
 $(2-6)*(3-9)$
 $((2*6)+3)+9$
 $(2*6)+(3+9)$
 $2*((6-3)+9)$
 $2*(6-(3-9))$
 $((2+6)*9)/3$
 $(2+6)*(9/3)$
 $((2*6)+9)+3$
 $(2*6)+(9+3)$
 $2*((6+9)-3)$
 $2*(6+(9-3))$
 $2*((9+6)-3)$
 $2*(9+(6-3))$
 $2*((9-3)+6)$
 $2*(9-(3-6))$
 $((9+6)-3)*2$
 $(9+(6-3))*2$
 $(9+(6*2))+3$
 $9+((6*2)+3)$
 $(9*(6+2))/3$
 $9*((6+2)/3)$
 $((9*6)/2)-3$
 $(9*(6/2))-3$
 $9+(3+(6*2))$
 $(9+3)+(6*2)$
 $((9-3)+6)*2$
 $(9-(3-6))*2$
 $(9-3)*(6-2)$
 $(9*3)-(6/2)$
 $(9/3)*(6+2)$
 $9/(3/(6+2))$
 $9+(3+(2*6))$
 $(9+3)+(2*6)$

```
(9-(3+2))*6  
((9-3)-2)*6  
9*(3-(2/6))  
(9/3)*(2+6)  
9/(3/(2+6))  
(9+(2*6))+3  
9+((2*6)+3)  
(9*(2+6))/3  
9*((2+6)/3)  
((9/2)*6)-3  
(9/(2/6))-3  
(9-(2+3))*6  
((9-2)-3)*6
```

```
-----  
Time taken: 0.024 seconds
```

```
-----  
Do you want to save the result to a file? (y/n): 
```

3.3 Hasil Pengujian 3



3.4 Hasil Pengujian 4

[illegible]

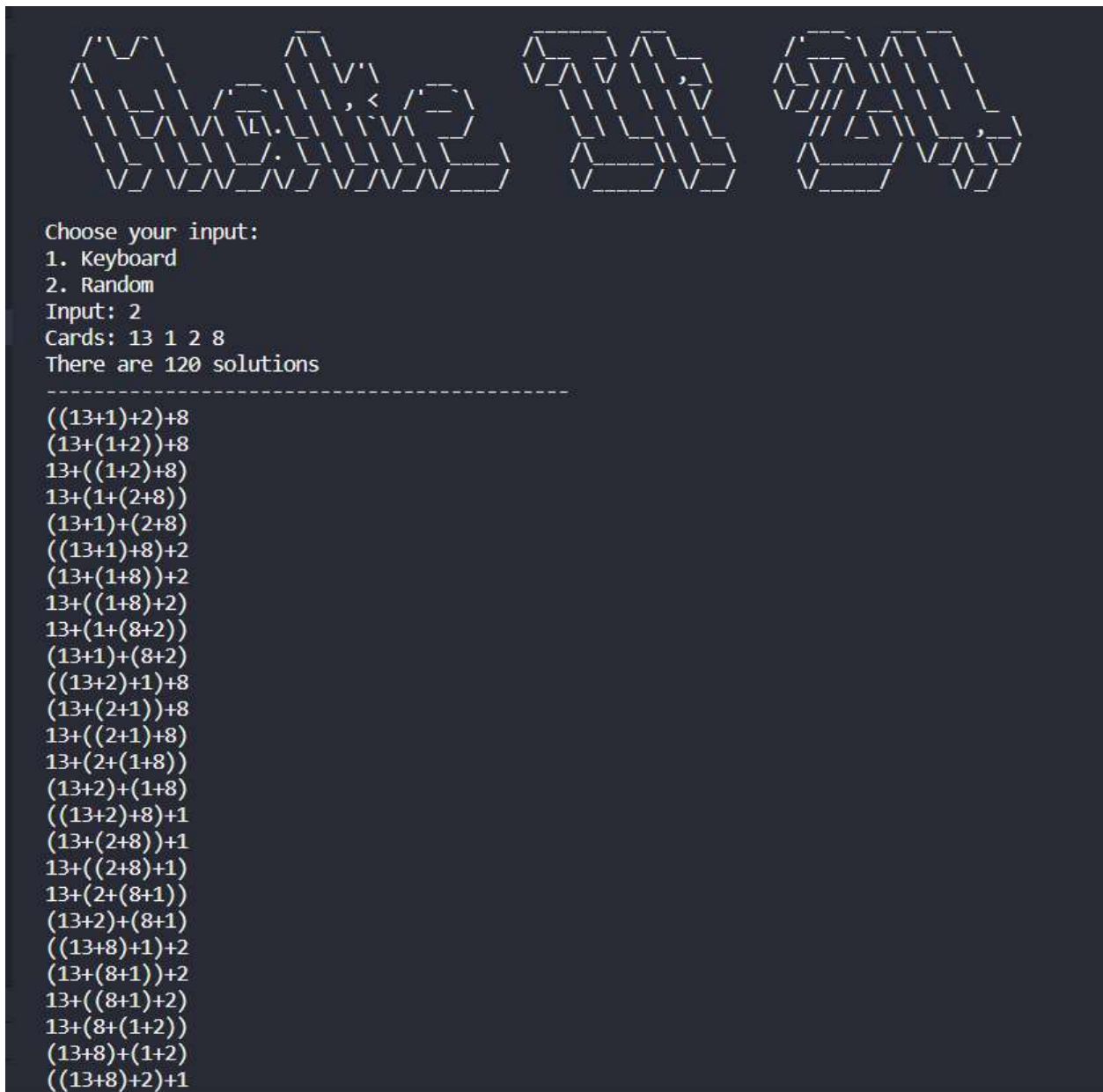
3.5 Hasil Pengujian 5



3.6 Hasil Pengujian 6



3.7 Hasil Pengujian 7




```

(13+(8+2))+1
13+((8+2)+1)
13+(8+(2+1))
(13+8)+(2+1)
((1+13)+2)+8
(1+(13+2))+8
1+((13+2)+8)
1+(13+(2+8))
(1+13)+(2+8)
((1+13)+8)+2
(1+(13+8))+2
1+((13+8)+2)
1+(13+(8+2))
(1+13)+(8+2)
((1+2)+13)+8
(1+(2+13))+8
1+((2+13)+8)
1+(2+(13+8))
(1+2)+(13+8)
((1+2)+8)+13
(1+(2+8))+13
1+((2+8)+13)
1+(2+(8+13))
(1+2)+(8+13)
((1+8)+13)+2
(1+(8+13))+2
1+((8+13)+2)
1+(8+(13+2))
(1+8)+(13+2)
((1+8)+2)+13
(1+(8+2))+13
1+((8+2)+13)
1+(8+(2+13))
(1+8)+(2+13)
((2+13)+1)+8
(2+(13+1))+8
2+((13+1)+8)
2+(13+(1+8))
(2+13)+(1+8)
((2+13)+8)+1
(2+(13+8))+1

```

$2+((13+8)+1)$
 $2+(13+(8+1))$
 $(2+13)+(8+1)$
 $((2+1)+13)+8$
 $(2+(1+13))+8$
 $2+((1+13)+8)$
 $2+(1+(13+8))$
 $(2+1)+(13+8)$
 $((2+1)+8)+13$
 $(2+(1+8))+13$
 $2+((1+8)+13)$
 $2+(1+(8+13))$
 $(2+1)+(8+13)$
 $((2+8)+13)+1$
 $(2+(8+13))+1$
 $2+((8+13)+1)$
 $2+(8+(13+1))$
 $(2+8)+(13+1)$
 $((2+8)+1)+13$
 $(2+(8+1))+13$
 $2+((8+1)+13)$
 $2+(8+(1+13))$
 $(2+8)+(1+13)$
 $((8+13)+1)+2$
 $(8+(13+1))+2$
 $8+((13+1)+2)$
 $8+(13+(1+2))$
 $(8+13)+(1+2)$
 $((8+13)+2)+1$
 $(8+(13+2))+1$
 $8+((13+2)+1)$
 $8+(13+(2+1))$
 $(8+13)+(2+1)$
 $((8+1)+13)+2$
 $(8+(1+13))+2$
 $8+((1+13)+2)$
 $8+(1+(13+2))$
 $(8+1)+(13+2)$
 $((8+1)+2)+13$
 $(8+(1+2))+13$
 $8+((1+2)+13)$


```
8+(1+(2+13))
(8+1)+(2+13)
((8+2)+13)+1
(8+(2+13))+1
8+((2+13)+1)
8+(2+(13+1))
(8+2)+(13+1)
((8+2)+1)+13
(8+(2+1))+13
8+((2+1)+13)
8+(2+(1+13))
(8+2)+(1+13)
```

```
-----
Time taken: 0.041 seconds
```

```
-----
Do you want to save the result to a file? (y/n): ☐
```

BAB IV

LAMPIRAN

4.1 Tautan *Repository*

https://github.com/goodgirlwannabe/Tucil1_13521006.git