Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma BruteForce

Oleh Kelvin Rayhan Alkarim

K03 – 13521005

1. Algoritma BruteForce

Algoritma bruteforce adalah metode untuk menemukan solusi dengan mencoba semua kemungkinan secara berurutan. Ini sering digunakan dalam kriptografi untuk mencoba semua kunci yang mungkin untuk mendekripsi pesan yang dienkripsi. Algoritma ini sangat sederhana, tetapi juga sangat lambat jika jumlah kemungkinan yang harus dicoba cukup besar.

1.1 **Algoritma Operasi**

```
char convertop(int op){
    switch (op){
        case 0:
            return '+';
            break;
        case 1:
            return '-';
            break;
        case 2:
            return '*';
            break;
        case 3:
            return '/';
            break;
    }

}

void operatorr(vector <char> &result){
    int i, j, k;
    for (i = 0; i < 4; i++){
        for (j = 0; j < 4; j++){
            for (k = 0; k < 4; k++){
                result.push_back(convertop(i));
                result.push_back(convertop(j));
                result.push_back(convertop(k));

            }
        }
    }
}

float operation (float x, float y, char z){
    switch (z){
        case '+':
            return x + y;
            break;
        case '-':
            return x - y;
            break;
        case '*':
            return x * y;
            break;
        case '/':
            return x / y;
            break;
    }
}
```

Pada fungsi ini saya menggunakan penerapan algoritma bruteforce. Dapat dilihat pada fungsi operator, terdapat 3 loop (I,j,k) karena dalam setiap perhitungan solusi pasti hanya terdapat 3 operasi lalu menggunakan looping untuk menemukan semua kombinasi dari 4 operator. Di fungsi ini, saya menyimbolkan + - * / dengan angka yaitu 0 1 2 3 yang nanti akan dikonversi menjadi char menggunakan fungsi convertop. Karena kombinasi dari operasi masih dalam bentuk char agar bisa menjalankan perhitungan sesuai dengan operasinya, saya membuat fungsi operation yang fungsi ini bertujuan untuk mengoperasikan 2 angka.

1.2 Algoritma Kurung dan Pencarian Solusi

```
void solution(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(operation(a,b,result[j]),c,result[j+1]),d,result[j+2]);
//((a+b)*c)/d
    if (sum == 24){
        // cout <<"((" << a << result[j] << b << ")" << result[j+1] << c << ")"
<< result[j+2] << d << endl;
        file[*count] = "((" + inttostring(a) + result[j] + inttostring(b) + ")" +
result[j+1] + inttostring(c) + ")" + result[j+2] + inttostring(d);
        *x += 1;
        *count += 1;
    }
    }

}

void solution2(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(a,operation(b,c,result[j+1]),result[j]),d,result[j+2]);
//(a*(b+c))/d
    if (sum == 24){
        // cout << "(" << a << result[j] << "(" << b << result[j+1] << c << "))"
<< result[j+2] << d << endl;
        file[*count] = "(" + inttostring(a) + result[j] + "(" + inttostring(b) +
result[j+1] + inttostring(c) + "))" + result[j+2] + inttostring(d);
        *x += 1;
        *count += 1;
    }
    }


}

void solution3(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(a,operation(operation(b,c,result[j+1]),d,result[j+2]),result[j]); //
a/((b+c)*d)
    if (sum == 24){
        // cout << a << result[j] << "((" << b << result[j+1] << c << ")" <<
result[j+2] << d << ")" << endl;
        file[*count] = inttostring(a) + result[j] + "((" + inttostring(b) +
result[j+1] + inttostring(c) + ")" + result[j+2] + inttostring(d) + ")";
        *x += 1;
        *count += 1;
    }
    }
}

void solution4(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(a,operation(b,operation(c,d,result[j+2]),result[j+1]),result[j]); //
a/(b*(c/d))
    if (sum == 24){
        // cout << a << result[j] << "(" << b << result[j+1] << "(" << c <<
result[j+2] << d << "))" << endl;
        file[*count] = inttostring(a) + result[j] + "(" + inttostring(b) +
result[j+1] + "(" + inttostring(c) + result[j+2] + inttostring(d) + "))";
        *x += 1;
        *count += 1;
    }
    }
}

void solution5(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(a,b,result[j]),operation(c,d,result[j+2]),result[j+1]);
//(a+b)/(c+d)
    if (sum == 24){
        // cout << "(" << a << result[j] << b << ")" << result[j+1] << "(" << c
<< result[j+2] << d << ")" << endl;
        file[*count] = "(" + inttostring(a) + result[j] + inttostring(b) + ")" +
result[j+1] + "(" + inttostring(c) + result[j+2] + inttostring(d) + ")";
        *x += 1;
        *count += 1;
    }
    }
}
```

Fungsi ini saya buat untuk mencari hasil dari perhitungan dengan menggabungkan mutasi angka dan kombinasi serta kemungkinan letak kurung yang dapat dioperasikan. Pada fungsi solution, a b c d merupakan angka yang akan di operasikan lalu saya memasukan satu persatu ke dalam fungsi operation untuk menghitung angka dengan kombinasi operasi yang telah dimasukan kedalam vector menggunakan loop.

```
int j = 0;
    for (int i = 0; i < numb1.size(); i += 4){



            solution(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
```

```
            solution2(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution3(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution4(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution5(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            j += 3;
        }
```

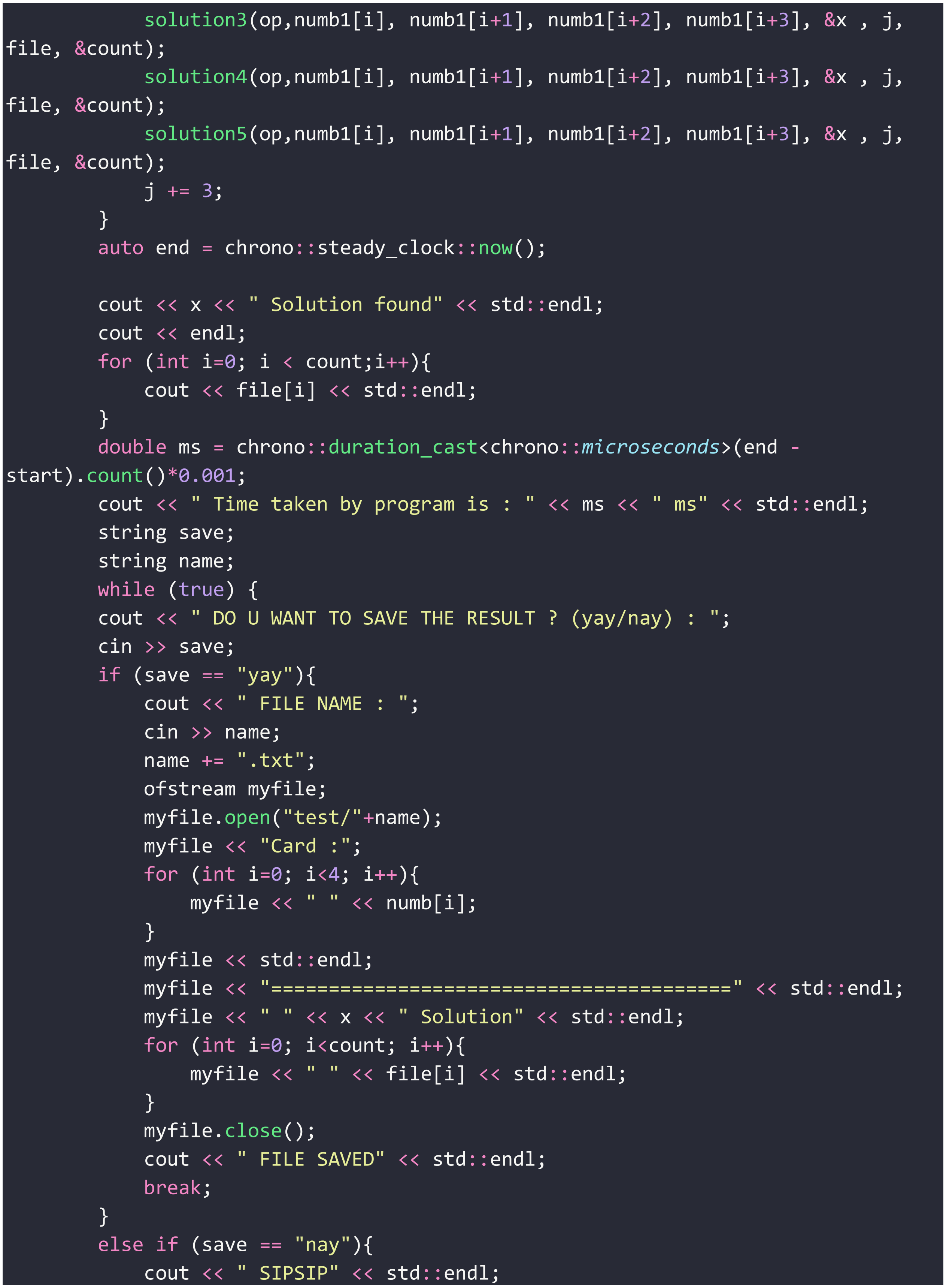
Lalu saya menggunakan algoritma bruteforce pada angka di bagian diatas pada main program. Jika digabungkan, program akan menjalankan semua kemungkinan dari urutan angka, urutan operasi dan kurung untuk mendapatkan hasil 24 lalu memasukkannya kedalam array of string.

2. Source Code Program

2.1 24CardGame.cpp

```
#include <iostream>
#include <cstdlib>
#include <chrono>
#include <fstream>
#include <bits/stdc++.h>
#include "function.cpp"
using namespace std;

int main()
{
    using Clock = chrono::high_resolution_clock;
    string file[1000];
    cout <<
"=================================================================================
==========" << endl;
    cout <<".------..------.     .------..------..------..------.     .------..--
----..------..------." << endl ;
    cout <<"|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--.
||A.--. ||M.--. ||E.--. |" << endl ;
    cout <<"| (\\/) || :/\\: ((5)) | :/\\: || (\\/) || :(): || :/\\: ((5)) |
:/\\: || (\\/) || (\\/) || (\\/) |" << endl ;
    cout <<"| :\\/: || :\\/: |'-.-.| :\\/: || :\\/: || ()() || (__) |'-.-.| :\\/:
|| :\\/: || :\\/: || :\\/: |" << endl ;
    cout <<"| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G||
'--'A|| '--'M|| '--'E|" << endl ;
    cout <<"`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`--
----'`------'`------'" << endl ;
```

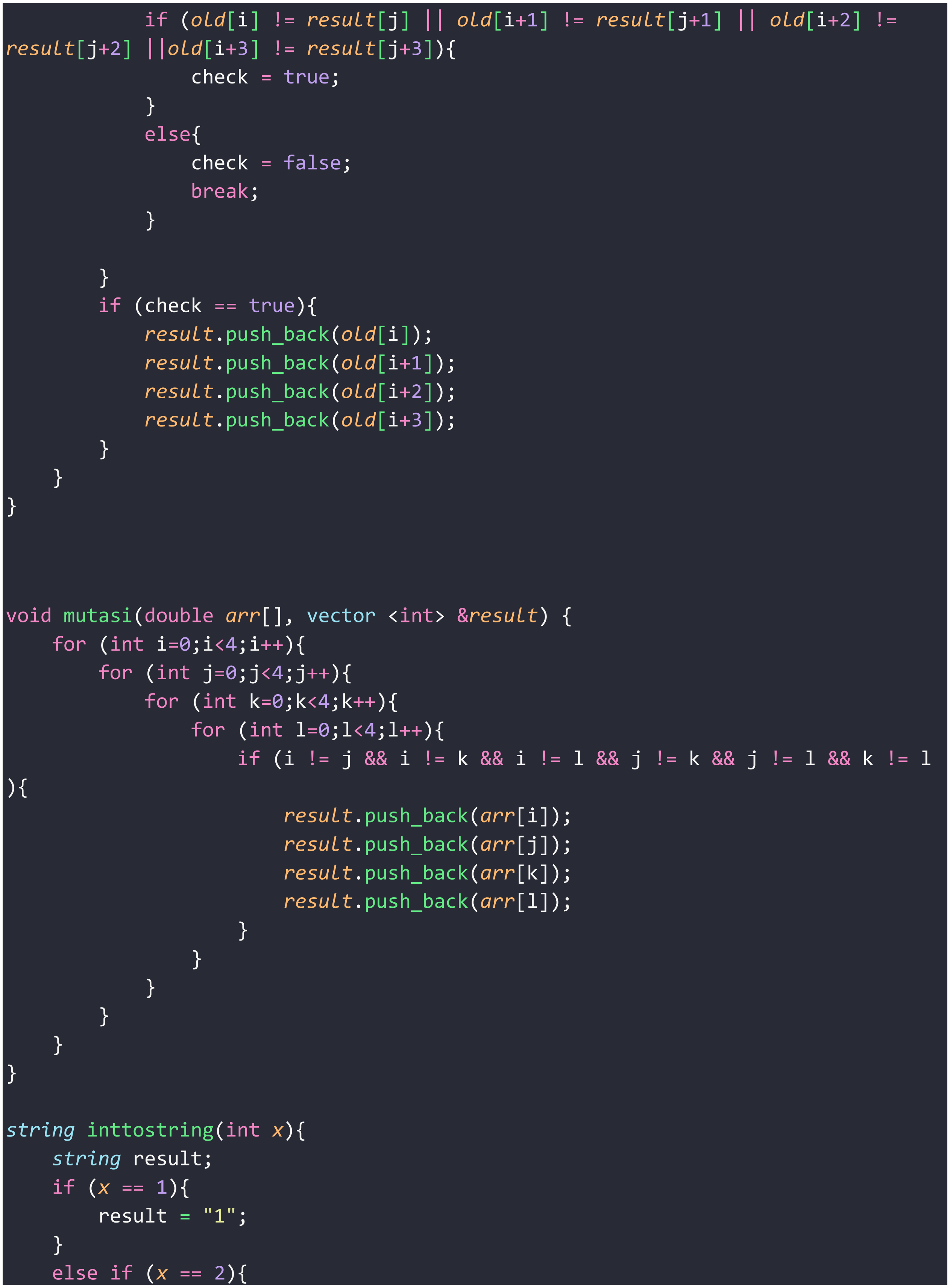
```
    cout << "==========================================================================================" << endl;
    cout << "                                  WELCOME";
    cout << " TO";
    cout << " 24 CARD GAME" << std::endl;
    cout << "==========================================================================================" << endl;

    cout << " PLEASE CHOOSE TYPE OF INPUT" << std::endl;
    cout << " 1. MANUAL" << std::endl;
    cout << " 2. RANDOM" << std::endl;
    cout << " 3. EXIT" << std::endl;
    vector <char> op;
    vector <int> numb1;
    vector <int> temp;
    int count;
    int choice;
    cin >> choice;
    double numb[4];
    switch (choice) {
        case 1:{
            cout << " MANUAL" << std::endl;
            string card[4];
            bool check = false;
            while(check == false){
                cout << " ENTER 4 NUMBERS" << std::endl;
                cin >> card[0] >> card[1] >> card[2] >> card[3];
            if(
            card[0] == "A" || card[1] == "A" || card[2] == "A" || card[3] == "A" ||
            card[0] == "2" || card[1] == "2" || card[2] == "2" || card[3] == "2" ||
            card[0] == "3" || card[1] == "3" || card[2] == "3" || card[3] == "3" ||
            card[0] == "4" || card[1] == "4" || card[2] == "4" || card[3] == "4" ||
            card[0] == "5" || card[1] == "5" || card[2] == "5" || card[3] == "5" ||
            card[0] == "6" || card[1] == "6" || card[2] == "6" || card[3] == "6" ||
            card[0] == "7" || card[1] == "7" || card[2] == "7" || card[3] == "7" ||
            card[0] == "8" || card[1] == "8" || card[2] == "8" || card[3] == "8"
||
            card[0] == "9" || card[1] == "9" || card[2] == "9" || card[3] == "9"
||
            card[0] == "10" || card[1] == "10" || card[2] == "10" || card[3] ==
"10" ||
            card[0] == "J" || card[1] == "J" || card[2] == "J" || card[3] == "J"
||
            card[0] == "Q" || card[1] == "Q" || card[2] == "Q" || card[3] == "Q"
||
            card[0] == "K" || card[1] == "K" || card[2] == "K" || card[3] ==
"K"||
            card[0] != card[1] || card[0] != card[2] || card[0] != card[3] ||
card[1] != card[2] || card[1] != card[3] || card[2] != card[3] )
            {
                check = true;

            }
            else
            {
                cout << " INVALID INPUT" << std::endl;
            }
            }

            for (int i = 0; i < 4 ; i++){
                if (card[i] == "A"){
                    numb[i] = 1;
                }
                else if (card[i] == "J"){
                    numb[i] = 11;
                }
                else if (card[i] == "Q"){
                    numb[i] = 12;
                }
                else if (card[i] == "K"){
                    numb[i] = 13;
                }
                else {
                    numb[i] = stoi(card[i]);
                }
            }
            break;
        }
        case 2:
            cout << " RANDOM" << std::endl;
            srand(time(0));
            for (int i = 0; i < 4; i++)
            {
                numb[i] = rand() % 13 + 1;
            }
            for (int i = 0; i < 4; i++)
            {
                if (numb[i] == 1){
                    cout << "A ";
                }
                else if (numb[i] == 11){
                    cout << "J ";
                }
                else if (numb[i] == 12){
                    cout << "Q ";
                }
                else if (numb[i] == 13){
                    cout << "K ";
                }
                \
                else {
                    cout << numb[i] << " ";
                }
            }
            cout << std::endl;
            break;
        case 3:
            cout << " LETS PLAY AGAIN SOMETIME... :(" << std::endl;
            return 0;
            break;
}
    auto start = chrono::steady_clock::now();
    mutasi(numb,temp);
    validmutasi(numb1, temp);
    operatorr(op);
    int x = 0;
    int j = 0;
    for (int i = 0; i < numb1.size(); i += 4){


            solution(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution2(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
```



```
            solution3(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution4(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            solution5(op,numb1[i], numb1[i+1], numb1[i+2], numb1[i+3], &x , j,
file, &count);
            j += 3;
        }
        auto end = chrono::steady_clock::now();

        cout << x << " Solution found" << std::endl;
        cout << endl;
        for (int i=0; i < count;i++){
            cout << file[i] << std::endl;
        }
        double ms = chrono::duration_cast<chrono::microseconds>(end -
start).count()*0.001;
        cout << " Time taken by program is : " << ms << " ms" << std::endl;
        string save;
        string name;
        while (true) {
        cout << " DO U WANT TO SAVE THE RESULT ? (yay/nay) : ";
        cin >> save;
        if (save == "yay"){
            cout << " FILE NAME : ";
            cin >> name;
            name += ".txt";
            ofstream myfile;
            myfile.open("test/"+name);
            myfile << "Card :";
            for (int i=0; i<4; i++){
                myfile << " " << numb[i];
            }
            myfile << std::endl;
            myfile << "========================================" << std::endl;
            myfile << " " << x << " Solution" << std::endl;
            for (int i=0; i<count; i++){
                myfile << " " << file[i] << std::endl;
            }
            myfile.close();
            cout << " FILE SAVED" << std::endl;
            break;
        }
        else if (save == "nay"){
            cout << " SIPSIP" << std::endl;
            break;
        }
        else {
            cout << " INVALID INPUT" << std::endl;
        }
        }
        string again;
        while (true){
        cout << " DO U WANT TO PLAY AGAIN ? (yay/nay) : ";
        cin >> again;
        if (again == "yay"){
            cout << " LETS PLAY AGAIN" << std::endl;
            main();
            break;
        }
        else if (again == "nay"){
            cout << " BYE BYE" << std::endl;
            break;
        }
        else {
            cout << " INVALID INPUT" << std::endl;
        }
        }


}
```

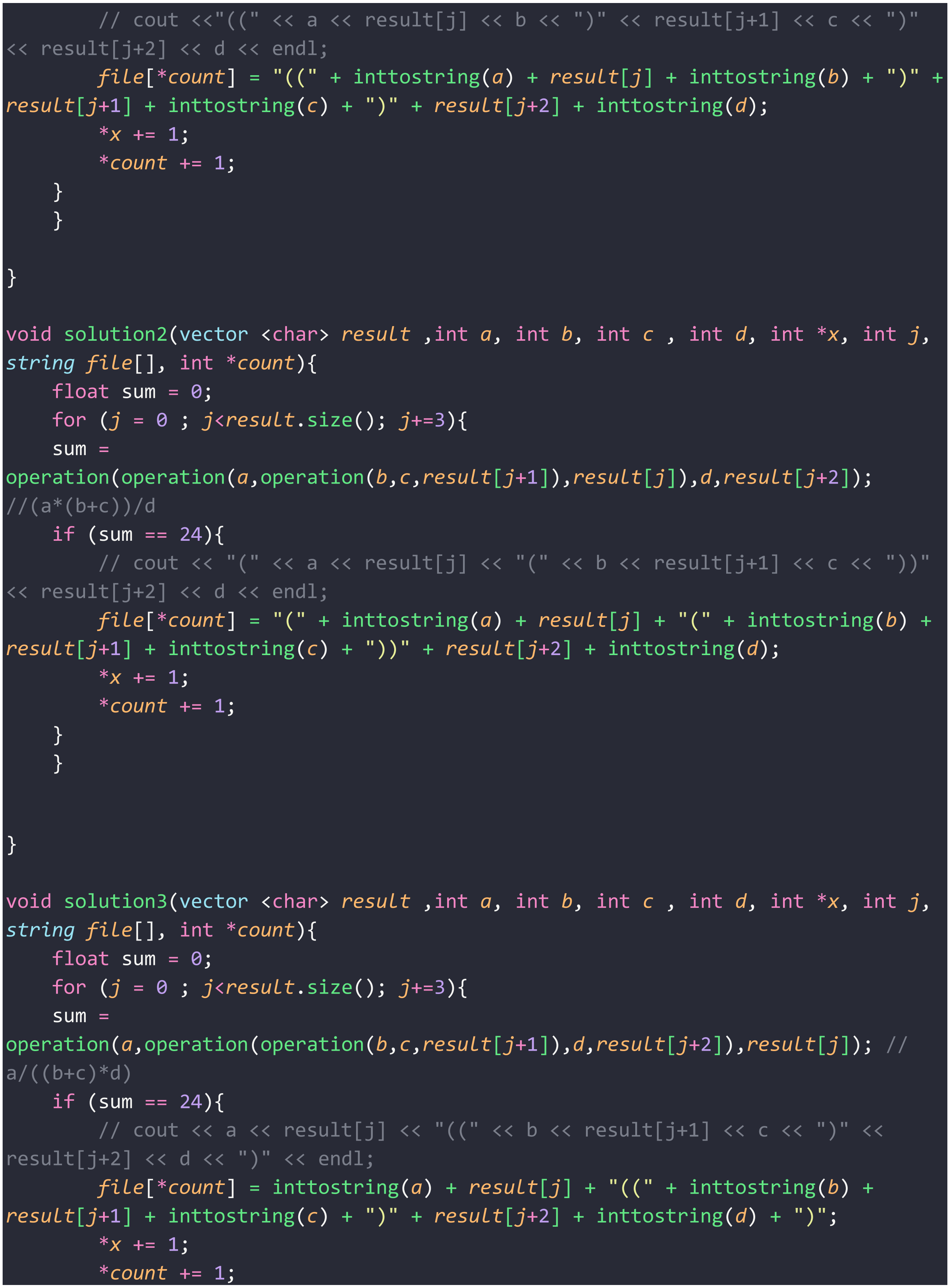
2.2 **fungsion.cpp**

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <vector>
using namespace std;

void validmutasi(vector <int> &result, vector <int> old){
    bool check = false;
    result.push_back(old[0]);
    result.push_back(old[1]);
    result.push_back(old[2]);
    result.push_back(old[3]);
    // cout << result[0] << result[1] << result[2] << result[3] << endl;
    // cout << old[0] << old[1] << old[2] << old[3] << endl;
    for (int i = 4; i<old.size();i+=4){
        for (int j = 0; j<result.size();j+=4){
```



```
            if (old[i] != result[j] || old[i+1] != result[j+1] || old[i+2] !=
result[j+2] ||old[i+3] != result[j+3]){
                check = true;
            }
            else{
                check = false;
                break;
            }

        }
        if (check == true){
            result.push_back(old[i]);
            result.push_back(old[i+1]);
            result.push_back(old[i+2]);
            result.push_back(old[i+3]);
        }
    }
}



void mutasi(double arr[], vector <int> &result) {
    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++){
                    if (i != j && i != k && i != l && j != k && j != l && k != l
){
                        result.push_back(arr[i]);
                        result.push_back(arr[j]);
                        result.push_back(arr[k]);
                        result.push_back(arr[l]);
                    }
                }
            }
        }
    }
}

string inttostring(int x){
    string result;
    if (x == 1){
        result = "1";
    }
    else if (x == 2){
        result = "2";
    }
    else if (x == 3){
        result = "3";
    }
    else if (x == 4){
        result = "4";
    }
    else if (x == 5){
        result = "5";
    }
    else if (x == 6){
        result = "6";
    }
    else if (x == 7){
        result = "7";
    }
    else if (x == 8){
        result = "8";
    }
    else if (x == 9){
        result = "9";
    }
    else if (x == 10){
        result = "10";
    }
    else if (x == 11){
        result = "11";
    }
    else if (x == 12){
        result = "12";
    }
    else if (x == 13){
        result = "13";
    }
    else if (x == 14){
        result = "14";
    }
    else if (x == 15){
        result = "15";
    }
    else if (x == 16){
        result = "16";
    }
    else if (x == 17){
        result = "17";
    }
    else if (x == 18){
        result = "18";
    }
    else if (x == 19){
        result = "19";
    }
    else if (x == 20){
        result = "20";
    }
    else if (x == 21){
        result = "21";
    }
    else if (x == 22){
        result = "22";
    }
    else if (x == 23){
        result = "23";
    }
    else if (x == 24){
        result = "24";
    }
    return result;
}

char convertop(int op){
    switch (op){
        case 0:
            return '+';
            break;
        case 1:
            return '-';
            break;
        case 2:
            return '*';
            break;
        case 3:
            return '/';
            break;
        default:
            return ' ';
            break;
    }
```

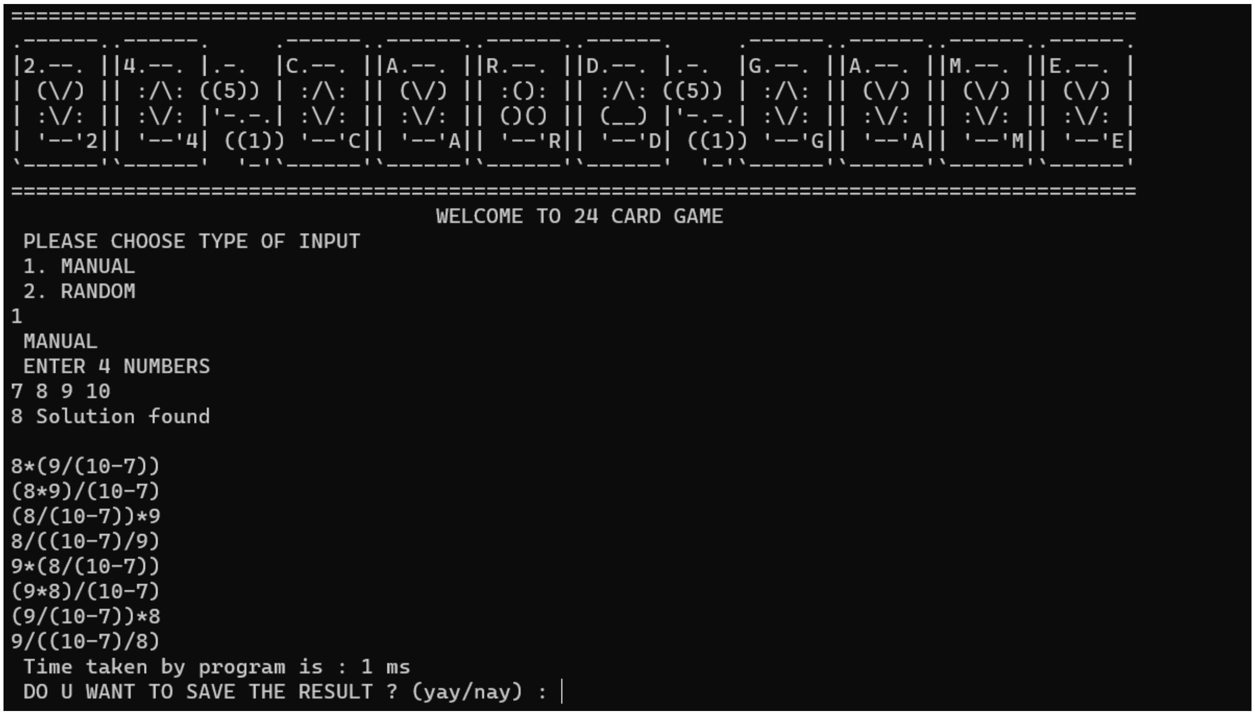
```
}

void operatorr(vector <char> &result){
    int i, j, k;
    for (i = 0; i < 4; i++){
        for (j = 0; j < 4; j++){
            for (k = 0; k < 4; k++){
                    result.push_back(convertop(i));
                    result.push_back(convertop(j));
                    result.push_back(convertop(k));

            }
        }
    }
}

float operation (float x, float y, char z){
    switch (z){
        case '+':
            return x + y;
            break;
        case '-':
            return x - y;
            break;
        case '*':
            return x * y;
            break;
        case '/':
            return x / y;
            break;
        default:
            return 0;
            break;
    }
}

void solution(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(operation(a,b,result[j]),c,result[j+1]),d,result[j+2]);
//((a+b)*c)/d
    if (sum == 24){
```



```
        // cout <<"((" << a << result[j] << b << ")" << result[j+1] << c << ")"
<< result[j+2] << d << endl;
        file[*count] = "((" + inttostring(a) + result[j] + inttostring(b) + ")" +
result[j+1] + inttostring(c) + ")" + result[j+2] + inttostring(d);
        *x += 1;
        *count += 1;
    }
    }

}

void solution2(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(a,operation(b,c,result[j+1]),result[j]),d,result[j+2]);
//(a*(b+c))/d
    if (sum == 24){
        // cout << "(" << a << result[j] << "(" << b << result[j+1] << c << "))"
<< result[j+2] << d << endl;
        file[*count] = "(" + inttostring(a) + result[j] + "(" + inttostring(b) +
result[j+1] + inttostring(c) + "))" + result[j+2] + inttostring(d);
        *x += 1;
        *count += 1;
    }
    }


}

void solution3(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(a,operation(operation(b,c,result[j+1]),d,result[j+2]),result[j]); //
a/((b+c)*d)
    if (sum == 24){
        // cout << a << result[j] << "((" << b << result[j+1] << c << ")" <<
result[j+2] << d << ")" << endl;
        file[*count] = inttostring(a) + result[j] + "((" + inttostring(b) +
result[j+1] + inttostring(c) + ")" + result[j+2] + inttostring(d) + ")";
        *x += 1;
        *count += 1;
    }
    }
}

void solution4(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(a,operation(b,operation(c,d,result[j+2]),result[j+1]),result[j]); //
a/(b*(c/d))
    if (sum == 24){
        // cout << a << result[j] << "(" << b << result[j+1] << "(" << c <<
result[j+2] << d << "))" << endl;
        file[*count] = inttostring(a) + result[j] + "(" + inttostring(b) +
result[j+1] + "(" + inttostring(c) + result[j+2] + inttostring(d) + "))";
        *x += 1;
        *count += 1;
    }
    }
}

void solution5(vector <char> result ,int a, int b, int c , int d, int *x, int j,
string file[], int *count){
    float sum = 0;
    for (j = 0 ; j<result.size(); j+=3){
    sum =
operation(operation(a,b,result[j]),operation(c,d,result[j+2]),result[j+1]);
//(a+b)/(c+d)
    if (sum == 24){
        // cout << "(" << a << result[j] << b << ")" << result[j+1] << "(" << c
<< result[j+2] << d << ")" << endl;
        file[*count] = "(" + inttostring(a) + result[j] + inttostring(b) + ")" +
result[j+1] + "(" + inttostring(c) + result[j+2] + inttostring(d) + ")";
        *x += 1;
        *count += 1;
    }
    }
}
```

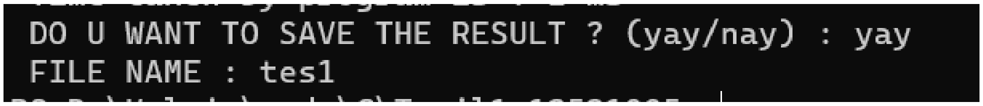
3. Input dan Output

3.1 **Contoh input manual dengan solusi**

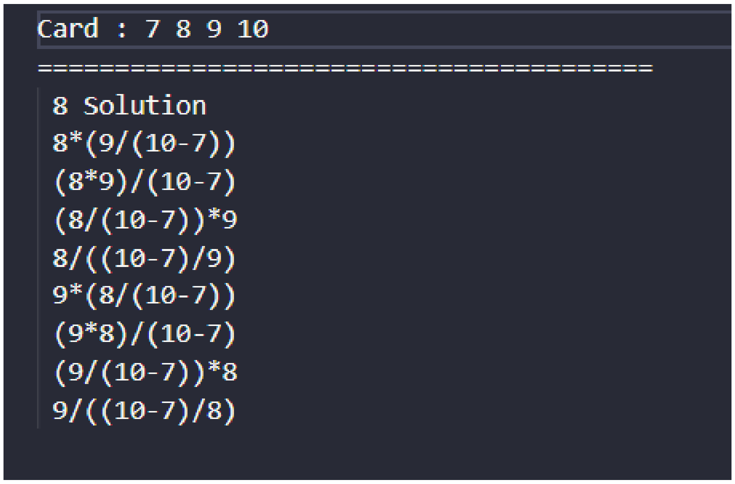


```
==========================================================================================
.------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                  WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
1
 MANUAL
 ENTER 4 NUMBERS
7 8 9 10
8 Solution found

8*(9/(10-7))
(8*9)/(10-7)
(8/(10-7))*9
8/((10-7)/9)
9*(8/(10-7))
(9*8)/(10-7)
(9/(10-7))*8
9/((10-7)/8)
 Time taken by program is : 1 ms
 DO U WANT TO SAVE THE RESULT ? (yay/nay) :
```

Save file.txt

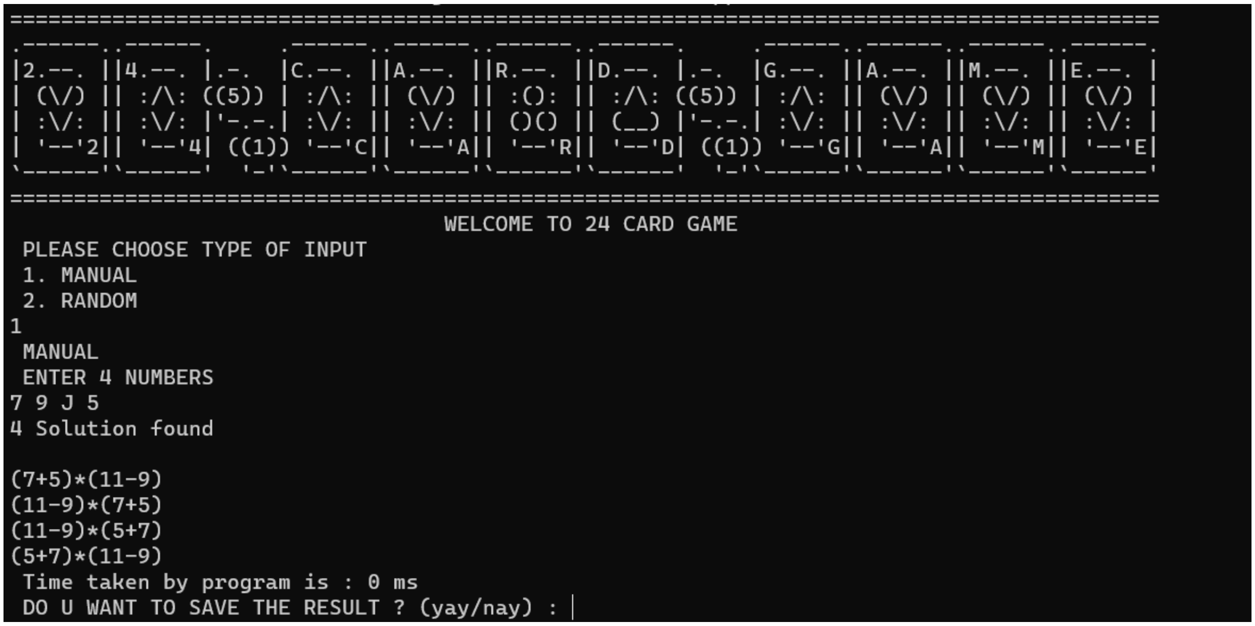


```
 DO U WANT TO SAVE THE RESULT ? (yay/nay) : yay
 FILE NAME : tes1
```



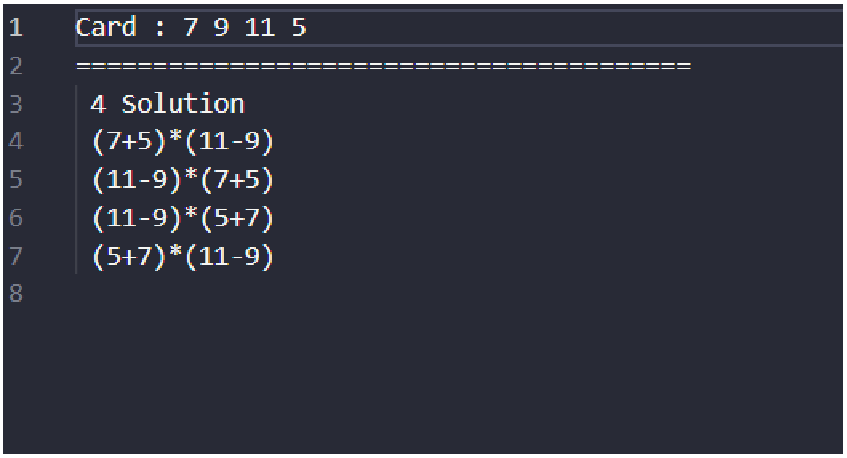
```
Card : 7 8 9 10
========================================
 8 Solution
 8*(9/(10-7))
 (8*9)/(10-7)
 (8/(10-7))*9
 8/((10-7)/9)
 9*(8/(10-7))
 (9*8)/(10-7)
 (9/(10-7))*8
 9/((10-7)/8)
```

3.2 **Contoh input manual dengan solusi (2)**



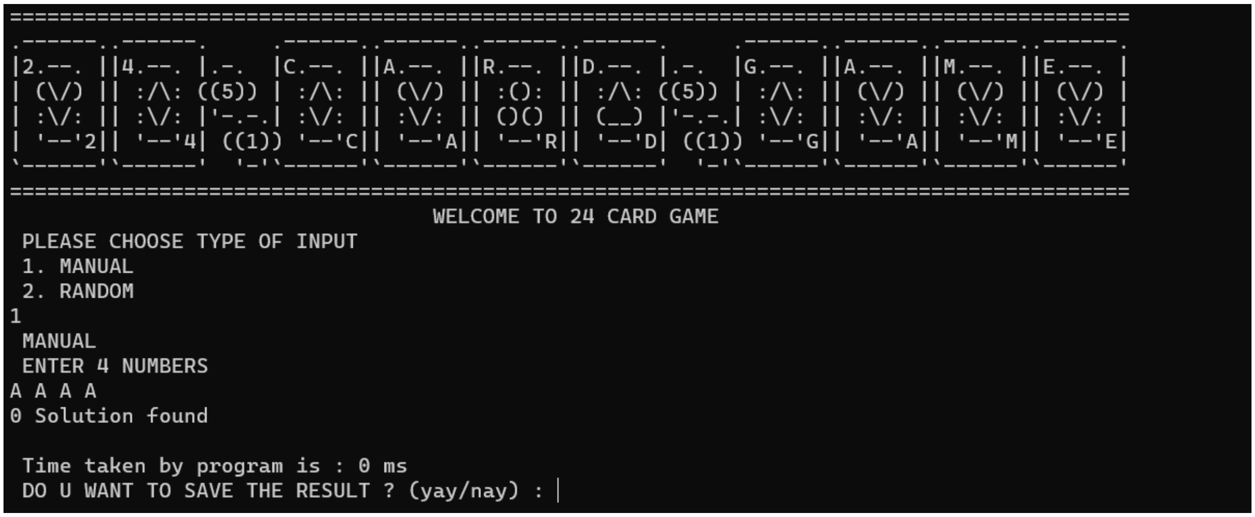
```
==========================================================================================
.------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                  WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
1
 MANUAL
 ENTER 4 NUMBERS
7 9 J 5
4 Solution found

(7+5)*(11-9)
(11-9)*(7+5)
(11-9)*(5+7)
(5+7)*(11-9)
 Time taken by program is : 0 ms
 DO U WANT TO SAVE THE RESULT ? (yay/nay) :
```

Save file.txt



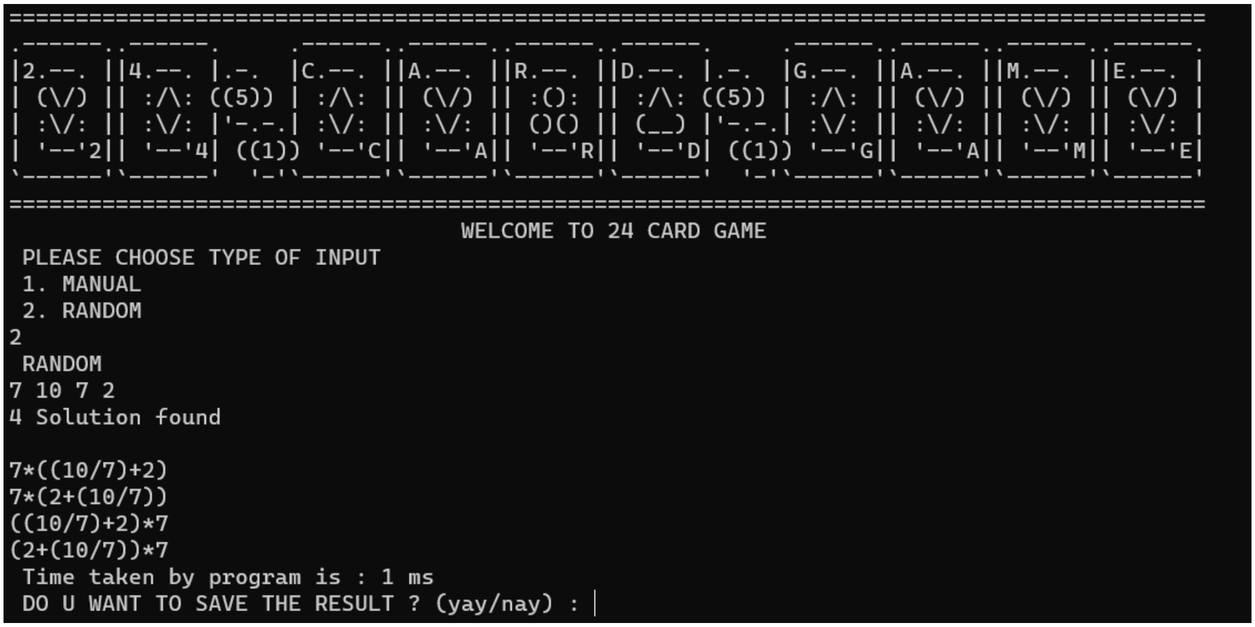
```
Card : 7 9 11 5
========================================
 4 Solution
 (7+5)*(11-9)
 (11-9)*(7+5)
 (11-9)*(5+7)
 (5+7)*(11-9)
```

3.3 **Contoh input manual tidak ada solusi**



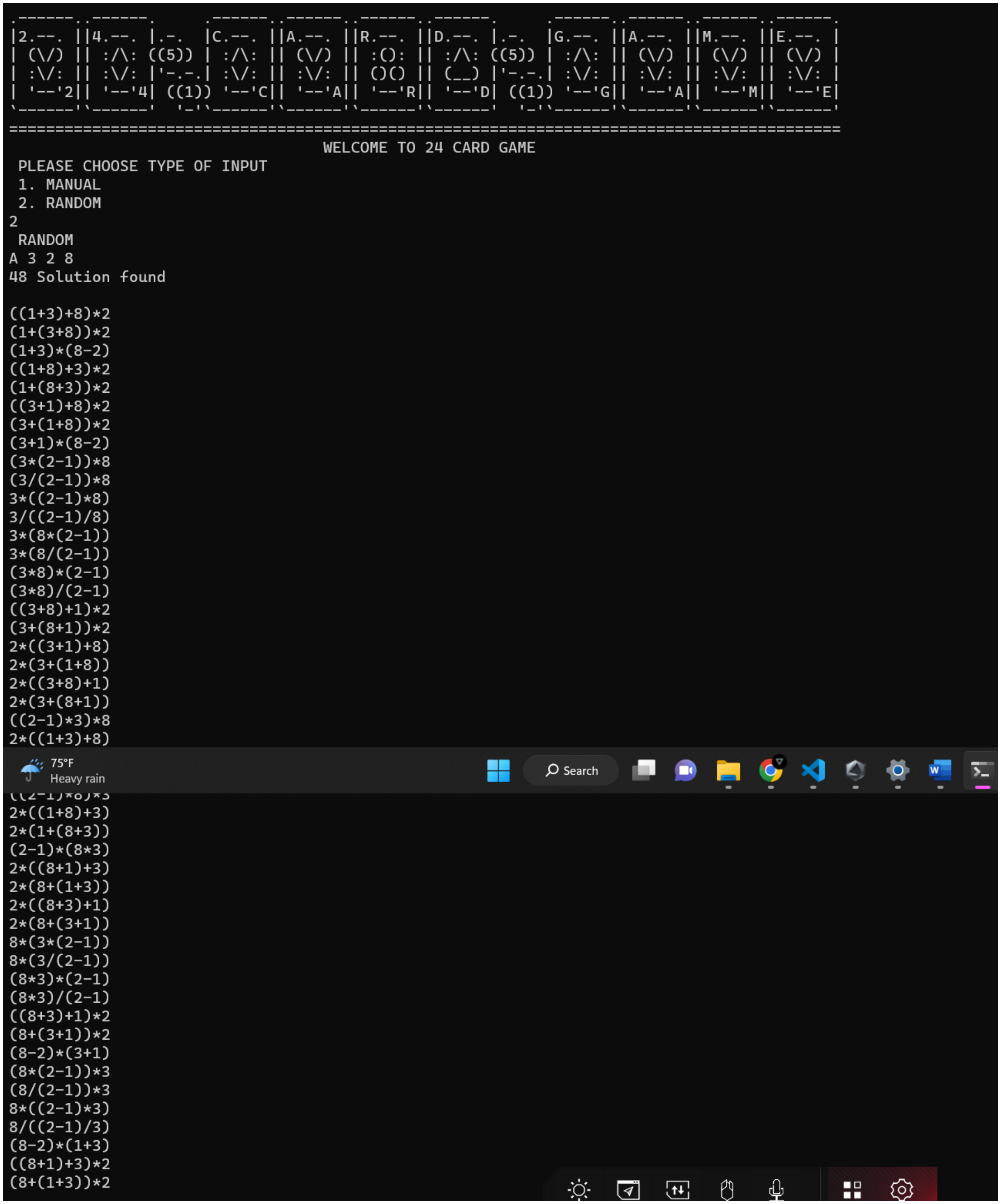
```
==========================================================================================
 .------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                  WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
1
 MANUAL
 ENTER 4 NUMBERS
A A A A
0 Solution found

 Time taken by program is : 0 ms
 DO U WANT TO SAVE THE RESULT ? (yay/nay) :
```

3.4 **Contoh input random**



```
==========================================================================================
 .------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                  WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
2
 RANDOM
7 10 7 2
4 Solution found

7*((10/7)+2)
7*(2+(10/7))
((10/7)+2)*7
(2+(10/7))*7
 Time taken by program is : 1 ms
 DO U WANT TO SAVE THE RESULT ? (yay/nay) :
```

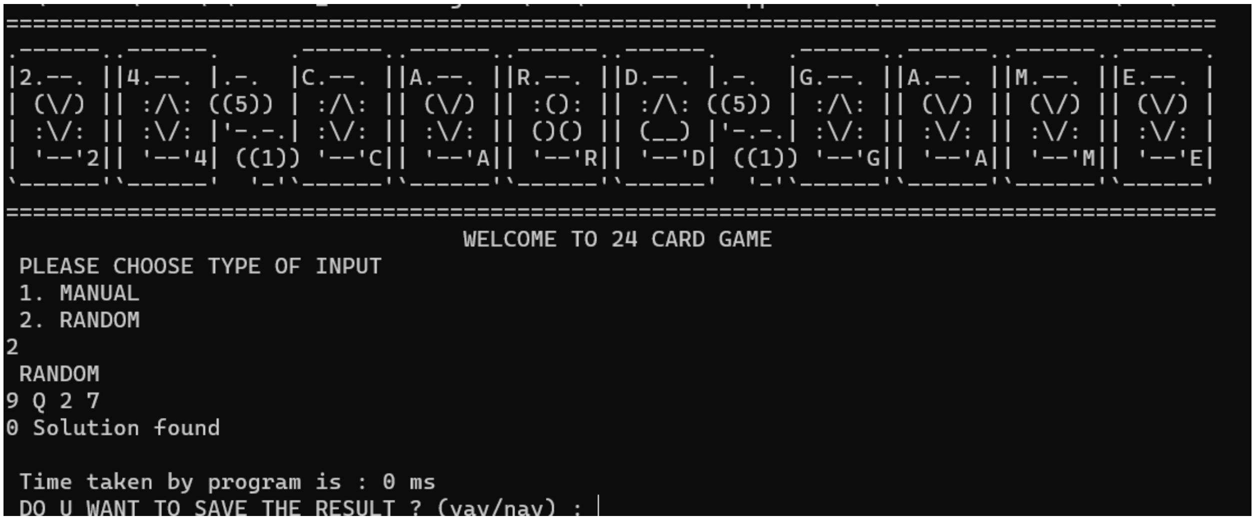
3.5 **Contoh input random (2)**



```
.------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                        WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
2
 RANDOM
A 3 2 8
48 Solution found

((1+3)+8)*2
(1+(3+8))*2
(1+3)*(8-2)
((1+8)+3)*2
(1+(8+3))*2
((3+1)+8)*2
(3+(1+8))*2
(3+1)*(8-2)
(3*(2-1))*8
(3/(2-1))*8
3*((2-1)*8)
3/((2-1)/8)
3*(8*(2-1))
3*(8/(2-1))
(3*8)*(2-1)
(3*8)/(2-1)
((3+8)+1)*2
(3+(8+1))*2
2*((3+1)+8)
2*(3+(1+8))
2*((3+8)+1)
2*(3+(8+1))
((2-1)*3)*8
2*((1+3)+8)
```

```
((2-1)*8)*3
2*((1+8)+3)
2*(1+(8+3))
(2-1)*(8*3)
2*((8+1)+3)
2*(8+(1+3))
2*((8+3)+1)
2*(8+(3+1))
8*(3*(2-1))
8*(3/(2-1))
(8*3)*(2-1)
(8*3)/(2-1)
((8+3)+1)*2
(8+(3+1))*2
(8-2)*(3+1)
(8*(2-1))*3
(8/(2-1))*3
8*((2-1)*3)
8/((2-1)/3)
(8-2)*(1+3)
((8+1)+3)*2
(8+(1+3))*2
```

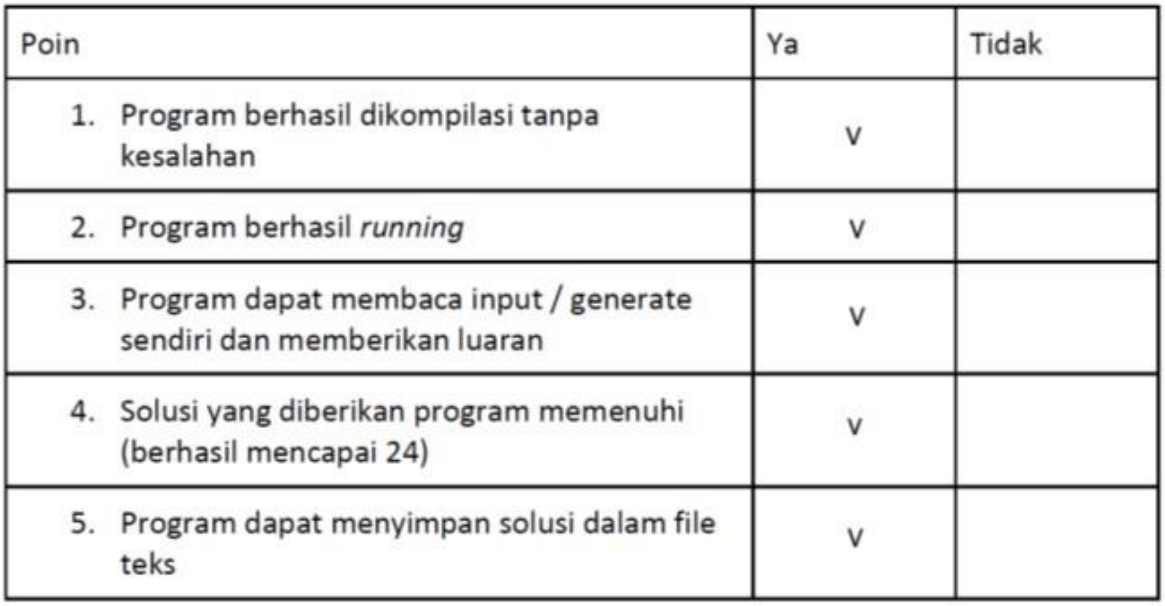
3.6 **Contoh input random (3)**



```
==========================================================================================
.------..------.     .------..------..------..------.     .------..------..------..------.
|2.--. ||4.--. |.-.  |C.--. ||A.--. ||R.--. ||D.--. |.-.  |G.--. ||A.--. ||M.--. ||E.--. |
| (\/) || :/\: ((5)) | :/\: || (\/) || :(): || :/\: ((5)) | :/\: || (\/) || (\/) || (\/) |
| :\/: || :\/: |'-.-.| :\/: || :\/: || ()() || (__) |'-.-.| :\/: || :\/: || :\/: || :\/: |
| '--'2|| '--'4| ((1)) '--'C|| '--'A|| '--'R|| '--'D| ((1)) '--'G|| '--'A|| '--'M|| '--'E|
`------'`------'  '-'`------'`------'`------'`------'  '-'`------'`------'`------'`------'
==========================================================================================
                                  WELCOME TO 24 CARD GAME
 PLEASE CHOOSE TYPE OF INPUT
 1. MANUAL
 2. RANDOM
2
 RANDOM
9 Q 2 7
0 Solution found

 Time taken by program is : 0 ms
 DO U WANT TO SAVE THE RESULT ? (yay/nay) :
```

4. Link Repository

https://github.com/kelvinra/Tucil1_13521005



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	