

LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
PENYELESAIAN PERMAINAN KARTU 24 DENGAN
ALGORITMA BRUTE FORCE



Austin Gabriel Pardosi
13521084
Teknik Informatika 2021

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

BAB I

CHECK LIST STATUS PROGRAM

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan		
2. Program berhasil running		
3. Program dapat membaca input / generate sendiri dan memberikan luaran		
4. Solusi yang diberikan program memenuhi (berhasil mencapai 24)		
5. Program dapat menyimpan solusi dalam file teks	Hanya total solusi, selebihnya dikeluarkan di CLI	

BAB II

LANGKAH - LANGKAH ALGORITMA BRUTE FORCE

A. Algoritma Brute Force

- Program menampilkan interface GAME 24.
- Program memberi pilihan 4 angka inputan, baik itu menggunakan random generator ataupun input secara manual.
- Jika memilih random generator, maka program akan mengeluarkan 4 angka dengan range 1-13 yang merepresentasikan setiap kartu yang ada.
- Jika tidak maka program akan meminta tepat 4 buah angka untuk direpresentasikan sebagai sebuah kartu yang ada.
- Tiap kartu diubah tipe datanya agar dapat ditampilkan sesuai kriteria di interface.
- Tiap angka dimasukkan ke dalam array of integer untuk dilakukan Algoritma Brute Force.

B. Algoritma Brute Force

- Di set waktu awal untuk mengetahui total waktu program berjalan.
- Dilakukan permutasi dengan menggunakan iterasi bercabang untuk mem permutasi keempat angka yang dimiliki.
- Setiap operator dimisalkan dengan sebuah angka agar mudah dioperasikan, contoh : 0 adalah operasi penjumlahan, 1 adalah operasi pengurangan, 2 adalah operasi perkalian, 3 adalah operasi pembagian.
- Keempat angka yang dimiliki diproses dengan berbagai kemungkinan yang mungkin terjadi (menggunakan tipe data float untuk mengolah floating point).
- Beberapa kemungkinan yang mungkin terjadi :
 - (a ops b) ops (c ops d)
 - a ops ((b ops c) ops d)
 - a ops (b ops (c ops d))
 - ((a ops b) ops c) ops d
 - (a ops (b ops c)) ops d

- Waktu di set stop, dan dicari selisihnya dan dikeluarkan sebagai parameter waktu.

C. Algoritma Output Hasil

- Waktu ditampilkan di dalam interface.
- Program akan mengoutput ke terminal semua kemungkinan yang ada
- Program akan bertanya apakah ingin disimpan ke dalam file hasil yang sudah didapat.

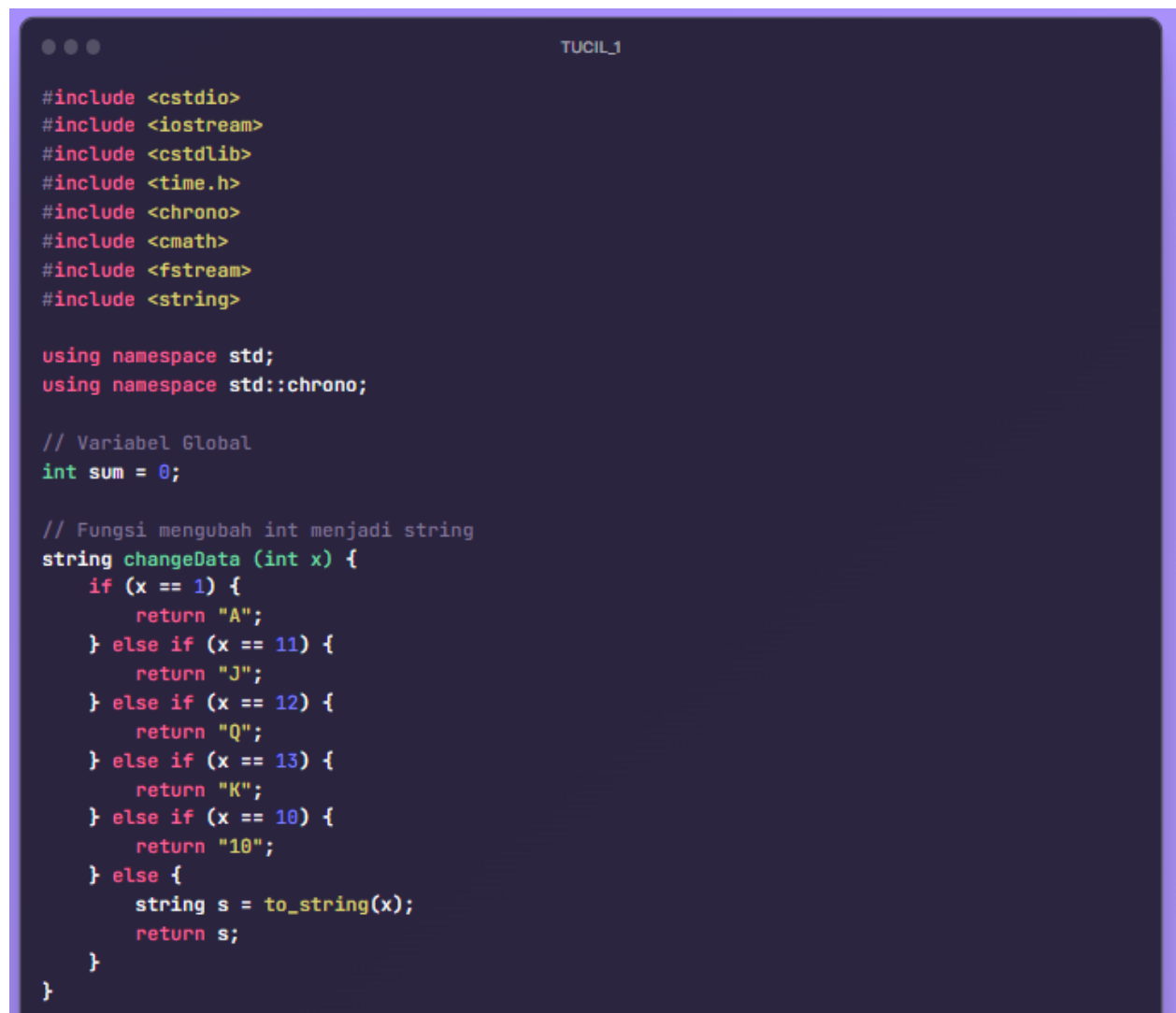
BAB III

SOURCE PROGRAM

Program ini dibagi menjadi 1 bagian, yaitu main program. Pada program ini, diterapkan konsep pemrograman berbasis prosedural.

A. Main Program

Program ini terdiri dari beberapa menu yang dapat melakukan berbagai instruksi, diantaranya yaitu:



```
TUCIL_1

#include <cstdio>
#include <iostream>
#include <cstdlib>
#include <time.h>
#include <chrono>
#include <cmath>
#include <fstream>
#include <string>

using namespace std;
using namespace std::chrono;

// Variabel Global
int sum = 0;

// Fungsi mengubah int menjadi string
string changeData (int x) {
    if (x == 1) {
        return "A";
    } else if (x == 11) {
        return "J";
    } else if (x == 12) {
        return "Q";
    } else if (x == 13) {
        return "K";
    } else if (x == 10) {
        return "10";
    } else {
        string s = to_string(x);
        return s;
    }
}
```

```

// Fungsi mengubah string menjadi int
int changeType (string x) {
    char c[x.length()];
    int i=0;

    // ALGORITMA
    // Instansiasi for loop
    while (i < x.length()) {
        c[i] = x[i];
        i++;
    }

    if (c[0] == 'A') {
        return 1;
    } else if (c[0] == 'K') {
        return 11;
    } else if (c[0] == 'Q') {
        return 12;
    } else if (c[0] == 'J') {
        return 13;
    } else if (c[0] == '1' && c[1] == '0') {
        return 10;
    } else {
        return (int)c[0]-48 ;
    }
}

// Operator:
// 0 -> +
// 1 -> -
// 2 -> *
// 3 -> /

float operat0r (int opr, float x, float y) {
    if (opr == 0) {
        return x+y;
    } else if (opr == 1) {
        return x-y;
    } else if (opr == 2) {
        return x*y;
    } else if (opr == 3) {
        return (float)x/(float)y;
    }
}

```

```

char changeOperator (int opr) {
    if (opr == 0) {
        return '+';
    } else if (opr == 1) {
        return '-';
    } else if (opr == 2) {
        return '*';
    } else if (opr == 3) {
        return '/';
    }
}

float operate1 (int a, int b, int c, int d, int op1, int op2, int op3) {
    // (a ops b) ops (c ops d)
    float res = operat0r(op2, operat0r(op1, a, b), operat0r(op3, c, d));
    return res;
}

float operate2 (int a, int b, int c, int d, int op1, int op2, int op3) {
    // a ops ((b ops c) ops d)
    float res = operat0r(op1, a, operat0r(op3, operat0r(op2, b, c), d));
    return res;
}

float operate3 (int a, int b, int c, int d, int op1, int op2, int op3) {
    // a ops (b ops (c ops d))
    float res = operat0r(op1, a, operat0r(op2, b, operat0r(op3, c, d)));
    return res;
}

float operate4 (int a, int b, int c, int d, int op1, int op2, int op3) {
    // ((a ops b) ops c) ops d
    float res = operat0r(op3, operat0r(op2, operat0r(op1, a, b), c), d);
    return res;
}

float operate5 (int a, int b, int c, int d, int op1, int op2, int op3) {
    // (a ops (b ops c)) ops d
    float res = operat0r(op3, operat0r(op1, a, operat0r(op2, b, c)), d);
    return res;
}

```

```

// ALGORITMA UTAMA
// Untuk menghitung total solusi yang ada
int totalSolusi (int w, int x, int y, int z) {
    for(int m=0; m<4; m++) {
        for(int n=0; n<4; n++) {
            for(int o=0; o<4; o++) {
                float nilai1 = operate1(w, x, y, z, m, n, o);
                float nilai2 = operate2(w, x, y, z, m, n, o);
                float nilai3 = operate3(w, x, y, z, m, n, o);
                float nilai4 = operate4(w, x, y, z, m, n, o);
                float nilai5 = operate5(w, x, y, z, m, n, o);
                if(nilai1 > 23.99999 && nilai1 < 24.00001) { // Floating point handler
                    sum++;
                }
                if(nilai2 > 23.99999 && nilai2 < 24.00001) {
                    sum++;
                }
                if(nilai3 > 23.99999 && nilai3 < 24.00001) {
                    sum++;
                }
                if(nilai4 > 23.99999 && nilai4 < 24.00001) {
                    sum++;
                }
                if(nilai5 > 23.99999 && nilai5 < 24.00001) {
                    sum++;
                }
            }
        }
    }
    return sum;
}

```

```

// Untuk memprint seluruh solusi yang ada
void solver (int w, int x, int y, int z) {
    for(int m=0; m<4; m++) {
        for(int n=0; n<4; n++) {
            for(int o=0; o<4; o++) {
                // printf("%d %d %d %d %d %d %d\n", value[i], value[j], value[k], value[l], m,
                n, o);

                float nilai1 = operate1(w, x, y, z, m, n, o);
                float nilai2 = operate2(w, x, y, z, m, n, o);
                float nilai3 = operate3(w, x, y, z, m, n, o);
                float nilai4 = operate4(w, x, y, z, m, n, o);
                float nilai5 = operate5(w, x, y, z, m, n, o);
                if(nilai1 > 23.99999 && nilai1 < 24.00001) { // Floating point handler
                    printf("(%d %c %d) %c (%d %c %d)\n", w, changeOperator(m), x,
                    changeOperator(n), y, changeOperator(o), z);
                }
                if(nilai2 > 23.99999 && nilai2 < 24.00001) {
                    printf("(%d %c ((%d %c %d) %c %d)\n", w, changeOperator(m), x,
                    changeOperator(n), y, changeOperator(o), z);
                }
                if(nilai3 > 23.99999 && nilai3 < 24.00001) {
                    printf("(%d %c (%d %c (%d %c %d))\n", w, changeOperator(m), x,
                    changeOperator(n), y, changeOperator(o), z);
                }
                if(nilai4 > 23.99999 && nilai4 < 24.00001) {
                    printf("(%d %c %d) %c %d) %c %d\n", w, changeOperator(m), x,
                    changeOperator(n), y, changeOperator(o), z);
                }
                if(nilai5 > 23.99999 && nilai5 < 24.00001) {
                    printf("(%d %c (%d %c %d)) %c %d\n", w, changeOperator(m), x,
                    changeOperator(n), y, changeOperator(o), z);
                }
            }
        }
    }
}

```



```

int main() {
    string hasil[300];
    string unik[300];
    printf("\nSelamat Datang di Permainan Kartu 24\n");
    printf("\n");
    printf("  _____ \n");
    printf("|       |       |       |       | \n");
    printf("|  G   |  A   |  M   |  E   |  24 | \n");
    printf("|_____|_____|_____|_____| \n");

    int inp;
    while (true) {
        printf("\n");
        printf("Pilihan input :\n");
        printf("1. Random Generator\n");
        printf("2. Input Manual\n");
        printf("Silahkan memilih jenis inputan : ");
        scanf("%d", &inp);
        if (inp > 0 && inp < 3) {
            break;
        } else {
            printf("\nMasukkan tidak sesuai\n");
            printf("Silahkan memilih kembali jenis inputan!");
        }
    }

    if(inp == 1) {
        int N = 12;
        srand(time(0));
        int a1 = (rand() % N) + 1;
        int a2 = (rand() % N) + 1;
        int a3 = (rand() % N) + 1;
        int a4 = (rand() % N) + 1;
        string x1, x2, x3, x4;
        x1 = changeData(a1);
        x2 = changeData(a2);
        x3 = changeData(a3);
        x4 = changeData(a4);
        printf("%s %s %s %s\n", x1.c_str(), x2.c_str(), x3.c_str(), x4.c_str());
        int value[4] = {a1, a2, a3, a4};
        // Algoritma Utama Dipanggil
        printf("\n");
        auto start = std::chrono::high_resolution_clock::now();
    }
}

```

```

for (int i = 0; i <= 3; i++) {
    for (int j = 0; j <= 3; j++) {
        for (int k = 0; k <= 3; k++) {
            for (int l = 0; l <= 3; l++) {
                if (i != j && i != k && i != l && j != k && j != l && k != l) {
                    totalSolusi(value[i], value[j], value[k], value[l]);
                }
            }
        }
    }
}

printf("-----");
if (sum == 0) {
    printf("\nTidak Ada Solusi\n");
} else {
    printf("\n%d solutions\n", sum);
}

for (int i = 0; i <= 3; i++) {
    for (int j = 0; j <= 3; j++) {
        for (int k = 0; k <= 3; k++) {
            for (int l = 0; l <= 3; l++) {
                if (i != j && i != k && i != l && j != k && j != l && k != l) {
                    solver(value[i], value[j], value[k], value[l]);
                }
            }
        }
    }
}

auto stop = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> durations = stop - start;
printf("Waktu yang dibutuhkan : ");
printf("%lf", durations.count());
printf(" seconds\n");

char opt;
while (true) {
    printf("\n");
    printf("Apakah ingin menyimpan solusi? (Y/N) : ");
    scanf(" %c", &opt);
    if (opt == 'N' || opt == 'n') {
        printf("\nTerima Kasih Sudah Bermain\n");
        break;
    }
}

```

```

        break;
    } else if (opt == 'Y' || opt == 'y') {
        std::ofstream outfile;
        outfile.open("output.txt");

        outfile << "Hasil Game 24 : " << endl;
        for (int i = 0; i <= 3; i++) {
            for (int j = 0; j <= 3; j++) {
                for (int k = 0; k <= 3; k++) {
                    for (int l = 0; l <= 3; l++) {
                        if (i != j && i != k && i != l && j != k && j != l && k != l)
                        {
                            totalSolusi(value[i], value[j], value[k], value[l]);
                        }
                    }
                }
            }
        }

        outfile << "-----" << endl;
        if (sum == 0) {
            outfile << "Tidak Ada Solusi" << endl;
        } else {
            outfile << sum/2 << " solutions" << endl;
        }

        for (int i = 0; i <= 3; i++) {
            for (int j = 0; j <= 3; j++) {
                for (int k = 0; k <= 3; k++) {
                    for (int l = 0; l <= 3; l++) {
                        if (i != j && i != k && i != l && j != k && j != l && k != l)
                        {
                            solver(value[i], value[j], value[k], value[l]);
                        }
                    }
                }
            }
        }

        outfile.close();
        break;
    } else {
        printf("\nMasukkan tidak sesuai\n");
        printf("Silahkan memilih kembali jenis inputan!");
    }
}

```

```

} else {
    printf("Ketik 4 buah angka/huruf pada kartu yang diinginkan : ");
    char a[3], b[3], c[3], d[3];
    scanf("%s %s %s %s", a, b, c, d);
    string x1, x2, x3, x4;
    x1 = a;
    x2 = b;
    x3 = c;
    x4 = d;
    int a1, a2, a3, a4;
    a1 = changeType(x1);
    a2 = changeType(x2);
    a3 = changeType(x3);
    a4 = changeType(x4);
    int value[4] = {a1, a2, a3, a4};
    // Validasi Inputan belum selesai
    auto start = std::chrono::high_resolution_clock::now();

    for (int i = 0; i <= 3; i++) {
        for (int j = 0; j <= 3; j++) {
            for (int k = 0; k <= 3; k++) {
                for (int l = 0; l <= 3; l++) {
                    if (i != j && i != k && i != l && j != k && j != l && k != l) {
                        totalSolusi(value[i], value[j], value[k], value[l]);
                    }
                }
            }
        }
    }

    printf("-----");
    if (sum == 0) {
        printf("\nTidak Ada Solusi\n");
    } else {
        printf("\n%d solutions\n", sum);
    }

    for (int i = 0; i <= 3; i++) {
        for (int j = 0; j <= 3; j++) {
            for (int k = 0; k <= 3; k++) {
                for (int l = 0; l <= 3; l++) {
                    if (i != j && i != k && i != l && j != k && j != l && k != l) {
                        solver(value[i], value[j], value[k], value[l]);
                    }
                }
            }
        }
    }
}

```

```

    }
    }
}

}

auto stop = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> durations = stop - start;
printf("Waktu yang dibutuhkan : ");
printf("%lf", durations.count());
printf(" seconds\n");

char opt;
while (true) {
    printf("\n");
    printf("Apakah ingin menyimpan solusi? (Y/N) : ");
    scanf(" %c", &opt);
    if (opt == 'N' || opt == 'n') {
        printf("\nTerima Kasih Sudah Bermain\n");
        break;
    } else if (opt == 'Y' || opt == 'y') {
        std::ofstream outfile;
        outfile.open("output.txt");

        outfile << "Hasil Game 24 : " << endl;
        for (int i = 0; i <= 3; i++) {
            for (int j = 0; j <= 3; j++) {
                for (int k = 0; k <= 3; k++) {
                    for (int l = 0; l <= 3; l++) {
                        if (i != j && i != k && i != l && j != k && j != l && k != l)
                    {
                        totalSolusi(value[i], value[j], value[k], value[l]);
                    }
                }
            }
        }
        outfile << "-----" << endl;
        if (sum == 0) {
            outfile << "Tidak Ada Solusi" << endl;
        } else {
            outfile << sum/2 << " solutions" << endl;
        }
    }
}

```

```

        for (int i = 0; i <= 3; i++) {
            for (int j = 0; j <= 3; j++) {
                for (int k = 0; k <= 3; k++) {
                    for (int l = 0; l <= 3; l++) {
                        if (i != j && i != k && i != l && j != k && j != l && k != l)
                    {
                        solver(value[i], value[j], value[k], value[l]);
                    }
                }
            }
        }
        outfile.close();
        break;
    } else {
        printf("\nMasukkan tidak sesuai\n");
        printf("Silahkan memilih kembali jenis inputan!");
    }
}
}
}

```


BAB IV

SCREENSHOT INPUT DAN OUTPUT

A. Test Case 1

Selamat Datang di Permainan Kartu 24

G	A	M	E	24
---	---	---	---	----

Pilihan input :

1. Random Generator

2. Input Manual

Silahkan memilih jenis inputan : 2

Ketik 4 buah angka/huruf pada kartu yang diinginkan : 2 A 3 4

242 solutions

```
((2 + 1) + 3) * 4
(2 + (1 + 3)) * 4
(2 * 1) * (3 * 4)
2 * ((1 * 3) * 4)
2 * (1 * (3 * 4))
((2 * 1) * 3) * 4
(2 * (1 * 3)) * 4
(2 / 1) * (3 * 4)
((2 / 1) * 3) * 4
2 / (1 / (3 * 4))
(2 / (1 / 3)) * 4
2 / ((1 / 3) / 4)
(2 * 1) * (4 * 3)
2 * ((1 * 4) * 3)
2 * (1 * (4 * 3))
((2 * 1) * 4) * 3
(2 * (1 * 4)) * 3
(2 / 1) * (4 * 3)
((2 / 1) * 4) * 3
2 / (1 / (4 * 3))
(2 / (1 / 4)) * 3
2 / ((1 / 4) / 3)
((2 + 3) + 1) * 4
(2 + (3 + 1)) * 4
(2 * 3) * (1 * 4)
2 * ((3 * 1) * 4)
2 * (3 * (1 * 4))
((2 * 3) * 1) * 4
(2 * (3 * 1)) * 4
2 * ((3 / 1) * 4)
((2 * 3) / 1) * 4
(2 * (3 / 1)) * 4
(2 * 3) / (1 / 4)
```

```
(4 * 1) * (2 * 3)
4 * ((1 * 2) * 3)
4 * (1 * (2 * 3))
((4 * 1) * 2) * 3
(4 * (1 * 2)) * 3
(4 / 1) * (2 * 3)
((4 / 1) * 2) * 3
4 / (1 / (2 * 3))
(4 / (1 / 2)) * 3
4 / ((1 / 2) / 3)
4 * ((1 + 3) + 2)
4 * (1 + (3 + 2))
(4 * 1) * (3 * 2)
4 * ((1 * 3) * 2)
4 * (1 * (3 * 2))
((4 * 1) * 3) * 2
(4 * (1 * 3)) * 2
(4 / 1) * (3 * 2)
((4 / 1) * 3) * 2
4 / (1 / (3 * 2))
(4 / (1 / 3)) * 2
4 / ((1 / 3) / 2)
4 * ((3 + 2) + 1)
4 * (3 + (2 + 1))
(4 * 3) * (2 * 1)
4 * ((3 * 2) * 1)
4 * (3 * (2 * 1))
((4 * 3) * 2) * 1
(4 * (3 * 2)) * 1
(4 * 3) * (2 / 1)
4 * ((3 * 2) / 1)
4 * (3 * (2 / 1))
((4 * 3) * 2) / 1
(4 * (3 * 2)) / 1
4 * ((3 + 1) + 2)
4 * (3 + (1 + 2))
(4 * 3) * (1 * 2)
4 * ((3 * 1) * 2)
4 * (3 * (1 * 2))
((4 * 3) * 1) * 2
(4 * (3 * 1)) * 2
4 * ((3 / 1) * 2)
((4 * 3) / 1) * 2
(4 * (3 / 1)) * 2
(4 * 3) / (1 / 2)
4 * (3 / (1 / 2))
```

Waktu yang dibutuhkan : 0.177096 seconds

Apakah ingin menyimpan solusi? (Y/N) : N

Terima Kasih Sudah Bermain

B. Test Case 2

Selamat Datang di Permainan Kartu 24

G	A	M	E	24
---	---	---	---	----

Pilihan input :

1. Random Generator

2. Input Manual

Silahkan memilih jenis inputan : 1

10 3 2 Q

40 solutions

$(10 / (3 + 2)) * 12$

$10 / ((3 + 2) / 12)$

$(10 * 3) - (12 / 2)$

$((10 + 2) * 3) - 12$

$(10 / (2 + 3)) * 12$

$10 / ((2 + 3) / 12)$

$((10 / 2) - 3) * 12$

$(10 * 2) + (12 / 3)$

$((10 * 2) - 12) * 3$

$(10 * 12) / (3 + 2)$

$10 * (12 / (3 + 2))$

$(10 * 12) / (2 + 3)$

$10 * (12 / (2 + 3))$

$(3 * (10 + 2)) - 12$

$3 * ((10 * 2) - 12)$

$(3 * 10) - (12 / 2)$

$(3 * (2 + 10)) - 12$

$3 * ((2 * 10) - 12)$

$(3 * 12) - (10 + 2)$

$((3 * 12) - 10) - 2$

$(3 * 12) - (2 + 10)$

$((3 * 12) - 2) - 10$

$((2 + 10) * 3) - 12$

$(2 * 10) + (12 / 3)$

$((2 * 10) - 12) * 3$

$(12 * 10) / (3 + 2)$

$12 * (10 / (3 + 2))$

$(12 * 10) / (2 + 3)$

$12 * (10 / (2 + 3))$

$12 * ((10 / 2) - 3)$

$(12 * 3) - (10 + 2)$

$((12 * 3) - 10) - 2$

$(12 / 3) + (10 * 2)$

$(12 * 3) - (2 + 10)$

$((12 * 3) - 2) - 10$

$(12 / 3) + (2 * 10)$

$(12 / (3 + 2)) * 10$

Waktu yang dibutuhkan : 0.033002 seconds

Apakah ingin menyimpan solusi? (Y/N) : Y

C. Test Case 3

```
Selamat Datang di Permainan Kartu 24



|   |   |   |   |    |
|---|---|---|---|----|
| G | A | M | E | 24 |
|---|---|---|---|----|



Pilihan input :
1. Random Generator
2. Input Manual
Silahkan memilih jenis inputan : 1
9 A 3 6

-----
24 solutions
((9 + 1) * 3) - 6
9 - ((1 - 6) * 3)
(9 - 1) * (6 - 3)
9 - (3 * (1 - 6))
(9 * (3 - 1)) + 6
((9 / 3) + 1) * 6
9 + (3 * (6 - 1))
9 + ((6 - 1) * 3)
((1 + 9) * 3) - 6
(1 + (9 / 3)) * 6
(1 - 9) * (3 - 6)
(3 * (9 + 1)) - 6
((3 - 1) * 9) + 6
(3 * (1 + 9)) - 6
(3 - 6) * (1 - 9)
(3 * (6 - 1)) + 9
6 - (9 * (1 - 3))
6 + (9 * (3 - 1))
6 * ((9 / 3) + 1)
6 * (1 + (9 / 3))
6 - ((1 - 3) * 9)
((6 - 1) * 3) + 9
(6 - 3) * (9 - 1)
6 + ((3 - 1) * 9)
Waktu yang dibutuhkan : 0.018003 seconds

Apakah ingin menyimpan solusi? (Y/N) : Y
```

```
Tucil1_13521084 > test > output.txt
1 Hasil Game 24 :
2 -----
3 24 solutions
4
```

D. Test Case 4

Selamat Datang di Permainan Kartu 24

G	A	M	E	24
---	---	---	---	----

Pilihan input :

1. Random Generator
2. Input Manual

Silahkan memilih jenis inputan : 1

Q 4 5 8

48 solutions

12 + ((4 * 5) - 8)
(12 + (4 * 5)) - 8
(12 - (4 + 5)) * 8
((12 - 4) - 5) * 8
12 - (4 * (5 - 8))
12 + (4 * (8 - 5))
(12 - 4) * (8 - 5)
12 + ((5 * 4) - 8)
(12 + (5 * 4)) - 8
(12 - (5 + 4)) * 8
((12 - 5) - 4) * 8
12 - ((5 - 8) * 4)
(12 - 8) + (4 * 5)
12 - (8 - (4 * 5))
12 + ((8 - 5) * 4)
(12 - 8) + (5 * 4)
12 - (8 - (5 * 4))
((12 - 8) * 5) + 4
(4 - 12) * (5 - 8)
4 + ((12 - 8) * 5)
4 + (5 * (12 - 8))
(4 * 5) + (12 - 8)
((4 * 5) + 12) - 8
4 - (5 * (8 - 12))
((4 * 5) - 8) + 12
(4 * 5) - (8 - 12)
4 - ((8 - 12) * 5)
(4 * (8 - 5)) + 12
(5 * (12 - 8)) + 4
(5 * 4) + (12 - 8)
((5 * 4) + 12) - 8
((5 * 4) - 8) + 12
(5 * 4) - (8 - 12)

4 + (5 * (12 - 8))
(4 * 5) + (12 - 8)
((4 * 5) + 12) - 8
4 - (5 * (8 - 12))
((4 * 5) - 8) + 12
(4 * 5) - (8 - 12)
4 - ((8 - 12) * 5)
(4 * (8 - 5)) + 12
(5 * (12 - 8)) + 4
(5 * 4) + (12 - 8)
((5 * 4) + 12) - 8
((5 * 4) - 8) + 12
(5 * 4) - (8 - 12)
(5 * 8) - (12 + 4)
((5 * 8) - 12) - 4
(5 - 8) * (4 - 12)
(5 * 8) - (4 + 12)
((5 * 8) - 4) - 12
8 * (12 - (4 + 5))
8 * ((12 - 4) - 5)
8 * (12 - (5 + 4))
8 * ((12 - 5) - 4)
(8 - 5) * (12 - 4)
(8 * 5) - (12 + 4)
((8 * 5) - 12) - 4
((8 - 5) * 4) + 12
(8 * 5) - (4 + 12)
((8 * 5) - 4) - 12

Waktu yang dibutuhkan : 0.048008 seconds

Apakah ingin menyimpan solusi? (Y/N) : N

Terima Kasih Sudah Bermain

E. Test Case 5

```
Selamat Datang di Permainan Kartu 24

| G | A | M | E | 24 |
|---|---|---|---|---|

Pilihan input :
1. Random Generator
2. Input Manual
Silahkan memilih jenis inputan : 2
Ketik 4 buah angka/huruf pada kartu yang diinginkan : 7 8 7 7
-----
Tidak Ada Solusi
Waktu yang dibutuhkan : 0.004004 seconds
```

F. Test Case 6

```
Selamat Datang di Permainan Kartu 24

| G | A | M | E | 24 |
|---|---|---|---|---|

Pilihan input :
1. Random Generator
2. Input Manual
Silahkan memilih jenis inputan : 2
Ketik 4 buah angka/huruf pada kartu yang diinginkan : A K Q J
-----
32 solutions
(1 * 12) * (13 - 11)
1 * (12 * (13 - 11))
1 * ((13 - 11) * 12)
((1 * 13) - 11) * 12
(1 * (13 - 11)) * 12
(12 * 1) * (13 - 11)
12 * ((1 * 13) - 11)
12 * (1 * (13 - 11))
(12 / 1) * (13 - 11)
12 / (1 / (13 - 11))
12 * (13 - (1 * 11))
12 * ((13 * 1) - 11)
12 * ((13 / 1) - 11)
12 * ((13 - 11) * 1)
12 * (13 - (11 * 1))
(12 * (13 - 11)) * 1
12 * ((13 - 11) / 1)
12 * (13 - (11 / 1))
(12 * (13 - 11)) / 1
(13 - (1 * 11)) * 12
((13 * 1) - 11) * 12
((13 / 1) - 11) * 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 - 11) * (12 * 1)
((13 - 11) * 12) * 1
(13 - 11) * (12 / 1)
((13 - 11) * 12) / 1
Waktu yang dibutuhkan : 0.031522 seconds

Apakah ingin menyimpan solusi? (Y/N) : N

Terima Kasih Sudah Bermain
```

G. Test Case 7

```
Selamat Datang di Permainan Kartu 24

  G  A  M  E  24

Pilihan input :
1. Random Generator
2. Input Manual
Silahkan memilih jenis inputan : 1
6 10 10 J

-----
Tidak Ada Solusi
Waktu yang dibutuhkan : 0.003735 seconds
```

H. Test Case 8

```
Selamat Datang di Permainan Kartu 24

  G  A  M  E  24

Pilihan input :
1. Random Generator
2. Input Manual
Silahkan memilih jenis inputan : 1
8 2 J 7

-----
114 solutions
(8 - 2) + (11 + 7)
((8 - 2) + 11) + 7
8 - (2 - (11 + 7))
(8 - (2 - 11)) + 7
8 - ((2 - 11) - 7)
(8 - 2) * (11 - 7)
(8 - 2) + (7 + 11)
((8 - 2) + 7) + 11
8 - (2 - (7 + 11))
(8 - (2 - 7)) + 11
8 - ((2 - 7) - 11)
8 * ((2 * 7) - 11)
8 + ((11 - 2) + 7)
((8 + 11) - 2) + 7
(8 + (11 - 2)) + 7
(8 + 11) - (2 - 7)
8 + (11 - (2 - 7))
(8 + 11) + (7 - 2)
8 + ((11 + 7) - 2)
8 + (11 + (7 - 2))
```

```
11 - ((2 - 7) - 8)
(11 + 7) + (8 - 2)
11 + ((7 + 8) - 2)
11 + (7 + (8 - 2))
((11 + 7) + 8) - 2
(11 + (7 + 8)) - 2
((11 - 7) + 8) * 2
(11 - (7 - 8)) * 2
(11 - 7) * (8 - 2)
11 + ((7 - 2) + 8)
((11 + 7) - 2) + 8
(11 + (7 - 2)) + 8
(11 + 7) - (2 - 8)
11 + (7 - (2 - 8))
7 + ((8 - 2) + 11)
((7 + 8) - 2) + 11
(7 + (8 - 2)) + 11
(7 + 8) - (2 - 11)
7 + (8 - (2 - 11))
(7 + 8) + (11 - 2)
7 + ((8 + 11) - 2)
7 + (8 + (11 - 2))
((7 + 8) + 11) - 2
(7 + (8 + 11)) - 2
(7 - 2) + (8 + 11)
((7 - 2) + 8) + 11
7 - (2 - (8 + 11))
(7 - (2 - 8)) + 11
7 - ((2 - 8) - 11)
(7 - 2) + (11 + 8)
((7 - 2) + 11) + 8
7 - (2 - (11 + 8))
(7 - (2 - 11)) + 8
7 - ((2 - 11) - 8)
((7 * 2) - 11) * 8
(7 + 11) + (8 - 2)
7 + ((11 + 8) - 2)
7 + (11 + (8 - 2))
((7 + 11) + 8) - 2
(7 + (11 + 8)) - 2
7 + ((11 - 2) + 8)
((7 + 11) - 2) + 8
(7 + (11 - 2)) + 8
(7 + 11) - (2 - 8)
7 + (11 - (2 - 8))
(7 - 11) * (2 - 8)
Waktu yang dibutuhkan : 0.090261 seconds

Apakah ingin menyimpan solusi? (Y/N) : N

Terima Kasih Sudah Bermain
```


BAB V
LINK REPOSITORY

https://github.com/AustinPardosi/Tucil1_13521084.g

