Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2022/2023

Disusun Oleh:

Nigel Sahl (13521043)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022/2023

A. Deskripsi Persoalan

Permainan kartu 24 adalah permainan kartu aritmatika dengan tujuan mencari cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24. Permainan ini menarik cukup banyak peminat dikarenakan dapat meningkatkan kemampuan berhitung serta mengasah otak agar dapat berpikir dengan cepat dan akurat. Permainan Kartu 24 biasa dimainkan dengan menggunakan kartu remi. Kartu remi terdiri dari 52 kartu yang terbagi menjadi empat suit (sekop, hati, keriting, dan wajik) yang masing-masing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Yang perlu diperhatikan hanyalah nilai kartu yang didapat (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). As bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri. Pada awal permainan moderator atau salah satu pemain mengambil 4 kartu dari dek yang sudah dikocok secara random. Permainan berakhir ketika pemain berhasil menemukan solusi untuk membuat kumpulan nilainya menjadi 24. Pengubahan nilai tersebut dapat dilakukan menggunakan operasi dasar matematika penjumlahan (+), pengurangan (-), perkalian (×), divisi (/) dan tanda kurung (()). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas.

B. Algoritma Brute Force

Pada persoalan ini, saya menyelesaikannya dengan algoritma Brute Force dengan kombinasi dan permutasi.

1. Pertama program mencari kemungkinan susunan dari empat buah angka dengan cara menaruh keempat angka ke dalam array misal A dan mengiterasi i, j, k, l sebanyak 4 kali sesuai panjang array A dan di akhir nilai array baru sebagai array hasil disesuaikan nilainya mulai dari indeks 0 sampai 3 dengan i, j, k, dan l. Lalu di iterasi j terdapat constrain tidak boleh sama dengan i, di iterasi k tidak boleh sama dengan i dan j, dan terakhir di iterasi l tidak boleh sama dengan i, j, dan k. Setelah itu di dalam loop l terdapat batasan untuk ke iterasi berikutnya yaitu i1, i2, dan i3, yaitu harus memenuhi kondisi dari fungsi boolean IsIn bernilai false.

2. Fungsi IsIn adalah fungsi yang mengembalikan true jika terdapat array hasil iterasi i, j, k, l pada data array permutasi. Iterasi-iterasi dan *constrains* di atas akan menghasilkan nilai-nilai array hasil sebagai permutasi dari ke empat angka yang ada.

3. Iterasi berikutnya adalah i1, i2, i3, sebagai iterasi untuk kombinasi susunan operasi yang ada dalam array operasi misal OP. Ketiga iterasi tersebut dimulai dari 0 sampai 3, sehingga terdapat 4x4x4 iterasi. Hasil tersebut disimpan ke dalam array hasil dengan indeks 0 sampai 3 yang di assign nilai i1, i2, dan i3. Hal di atas akan menghasilkan kombinasi dari keempat operasi dasar yaitu (+, -, x, /).

4. Setelah itu program menghitung banyaknya jumlah solusi dan menampilkan setiap solusi yang benar ke layar. Program akan memanggil fungsi yang diberi nama *TwentyFourG* atau *TwentyFourGSave* (jika ingin disimpan) untuk mengidentifikasi apakah rangkaian kombinasi dan permutasi di atas sesuai jika ditambahkan 5 kemungkinan susunan dua buah tanda kurung. Untuk mempermudah penulisan kita asumsikan ketiga OP adalah tanda tambah.
   a. ((n0 + n1) + n2) + n3
   b. (n0 + (n1 + n2)) + n3
   c. n0 + ((n1 + n2) + n3)
   d. n0 + (n1 + (n2 + n3))
   e. (n0 + n1) + (n2 + n3)

   Masing-masing kemungkinan tersebut diperiksa satu per satu dengan fungsi *calculate* yaitu fungsi untuk menghasilkan hasil operasi. Setalah diketahui hasilnya kemudain program membandingkan apakah sama dengan 24.0 atau tidak sama. Jika sama maka program akan meng-*increament* jumlah solusi dan menampilkan hasil ekspresi ke layar.

C. Source Program dalam Bahasa C

Dapat dilihat lengkapnya pada link repository:
https://github.com/NerbFox/Tucil1_13521043

1. Library yang disertakan dan defines

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "ADT_MesinKata/boolean.h"
#include "ADT_MesinKata/mesinkarakter.h"
#include "ADT_MesinKata/mesinkata.h"
#include "ADT_MesinKata/string.h"

#define maxHasil 700
#define maxStr 15
#define MILYAR 1000000000.0
```

2. Fungsi menu

```c
void menu()
{
    printf("\n\n -------------------- Make It  24 -------------------- \n");
    printf("\nMain Menu\n");
    printf("1. Input 4 Cards\n");
    printf("2. Generate 4 Random Numbers\n");
    printf("3. Exit\n");
    printf("Menu:  ");
    STARTCOMMAND();
    STARTWORD();
}
```

3. Fungsi IsIn

```
boolean IsIn(int arr[4], int (*arrPermut)[maxHasil][4], int np)
{
    boolean found = false;
    int i = 0;
    while (!found && i < np + 1)
    {
        if ((*arrPermut)[i][0] == arr[0] && (*arrPermut)[i][1] == arr[1] && (*arrPermut)[i][2] ==
        arr[2] && (*arrPermut)[i][3] == arr[3])
        {
            found = true;
        }
        i++;
    }
    return found;
}
```

4. Fungsi IsIn3

```
boolean IsIn3(float arr[3], float (hasil3)[maxHasil][3], int np)
{
    boolean found = false;
    int i = 0;
    while (!found && i < np + 1)
    {
        if ((hasil3)[i][0] == arr[0] && (hasil3)[i][1] == arr[1])
        {
            found = true;
        }
        i++;
    }
    return found;
}
```

Fungsi di atas memiliki tugas yaitu untuk memeriksa array hasil3, yakni setiap perhitungan di fungsi pengecekan apakah ekspresi bernilai 24, fungsi ini mengecek apakah indeks ke-0 dan ke-1 sudah ada dalam array hasil3 atau belum, kalau belum maka dimasukkan dan dinilai valid atau benar. Indeks ke-0 yaitu eksekusi perhitungan (calculate) pertama dari ekspresi, lalu indeks ke-2 berarti eksekusi perhitungan kedua dari ekspresi, dan indeks ketiga akan bernilai 24.0. Fungsi ini bertujuan mengeliminasi ekspresi yang bersifat **komutatif**.

5. Fungsi save

```
void save()
{
    printf("\nApakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)\n");
    printf("Input: ");
    STARTCOMMAND();
    STARTWORD();
}
```

6. Fungsi filename

```c
void fileName()
{
    printf("Nama File: ");
    STARTCOMMAND();
    STARTWORD();
    printf("\n");
}
```

7. Fungsi calculate

```c
float calculate(int a, char op, int b)
{
    float af = (float)a;
    if (op == '+')
    {
        return af + b;
    }
    else if (op == '-')
    {
        return af - b;
    }
    else if (op == '*')
    {
        return af * b;
    }
    else
    { // if (op=='/'){
        return af / b;
    }
}
```

8. Fungsi TwentyFourGSave

```
void TwentyFourGSave(int number[4], char op[3], char (*hasil)[maxHasil][maxStr], float (*hasil3)[maxHasil][3], int *neff, FILE *FileCr)
{

    float a, b, c, d, e;
    char temp;
    float hasil3n[3];
    // ((n0+n1)+n2)+n3  (*hasil3)[*neff][0]
    (hasil3n)[0] = calculate(number[0], op[0], number[1]);
    (hasil3n)[1] = calculate((hasil3n)[0], op[1], number[2]);
    (hasil3n)[2] = calculate((hasil3n)[1], op[2], number[3]);
    a = (hasil3n)[2];
    if (a == 24.00)···

    // n0+((n1+n2)+n3)
    (hasil3n)[0] = calculate(number[1], op[1], number[2]);
    (hasil3n)[1] = calculate((hasil3n)[0], op[2], number[3]);
    (hasil3n)[2] = calculate(number[0], op[0], (hasil3n)[1]);
    b = (hasil3n)[2];
    if (b == 24.0)···

    // (n0+(n1+n2))+n3
    (hasil3n)[0] = calculate(number[1], op[1], number[2]);
    (hasil3n)[1] = calculate(number[0], op[0], (hasil3n)[0]);
    (hasil3n)[2] = calculate((hasil3n)[1], op[2], number[3]);
    c = (hasil3n)[2];
    if (c == 24.0)···

    // n0+(n1+(n2+n3))
    (hasil3n)[0] = calculate(number[2], op[2], number[3]);
    (hasil3n)[1] = calculate(number[1], op[1], (hasil3n)[0]);
    (hasil3n)[2] = calculate(number[0], op[0], (hasil3n)[1]);
    d = (hasil3n)[2];
    if (d == 24.00)···

    // (n0+n1)+(n2+n3)
    (hasil3n)[0] = calculate(number[2], op[2], number[3]);
    (hasil3n)[1] = calculate(number[0], op[0], number[1]);
    (hasil3n)[2] = calculate((hasil3n)[1], op[1], (hasil3n)[0]);
    e = (hasil3n)[2];
    if (e == 24.0)···
}
```

Untuk setiap kondisi (if statement) seperti ini:

```
if (a == 24.00)
{
    fprintf(FileCr, "((%d%c%d)%c%d)%c%d\n", number[0], op[0], number[1], op[1], number[2], op[2], number[3])
    (*neff)++;
    (*hasil3)[*neff][0] = hasil3n[0];
    (*hasil3)[*neff][1] = hasil3n[1];
    (*hasil3)[*neff][2] = a;
}
```

9. Fungsi TwentyFourG

Sama seperti yang di atas namun hanya *print* solusi saja

```
if (a == 24.00)
{
    printf("((%d%c%d)%c%d)%c%d\n", number[0], op[0], number[1], op[1], number[2], op[2], number[3]);
    (*neff)++;
    (*hasil3)[*neff][0] = hasil3n[0];
    (*hasil3)[*neff][1] = hasil3n[1];
    (*hasil3)[*neff][2] = a;
}
```

10. Fungsi TwentyFourSolutions
    Sama halnya seperti fungsi sebelumnya hanya berbeda pada:

```
if (a == 24.00)
{
    (*neff)++;
}
```

11. Main function

```
int main()
{
    // KAMUS
    int arr[5], n = 0, i, j, k, l, ni, z, i1, i2, i3, np;
    arr[4] = 0;
    int atas = 13, bawah = 1;
    float TwentyFour = 24.0;
    char Hasil[maxHasil][maxStr];
    int neff = -1;
    char Op[4] = {'+', '-', '*', '/'};
    char OpHasil[3];
    int arrHasil[4];
    int arrPermut[maxHasil][4];
    float hasil3[maxHasil][3];
    char endOfStr = '\0';
    boolean cek = false;
    boolean simpan = false;
    struct timespec awal, akhir;
    double waktu;

    // char FiName[100];
    char ResultName[100] = "../doc/Hasil/";
    char typeName[] = ".txt";
    char *FileName;
    FILE *FileCr;

    // Menu
    menu();

    while (!(isWordSame(currentWord, "3")))···

    printf("\n ------------------- terima kasih ------------------- \n\n");
    return 0;
}
```

Program utama di atas terdiri dari kamus, inisiasi awal, dan pengulangan utama. Pada pengulangan utama, program akan terus berjalan dan mengulang kembali dari awal selama pengguna tidak memilih ‘3’ dalam Main Menu. Program akan berjalan pada menu 1 atau menu 2 (tergantung pilihan pengguna) dan di akhir akan dikembalikan lagi ke Main Menu jika pengguna memilih ‘0’ atau exit jika pengguna memilih ‘3’.

12. Dalam instruksi pengulangan *while*
    Program di bawah adalah inisiasi awal kembali agar setiap pengulangan inisiasi dapat dikondisikan ke keadaaan awal.

```
n = 0;
arr[4] = 0;
TwentyFour = 24.0;
neff = -1;
np = -1;
endOfStr = '\0';
cek = false;
simpan = false;

while (!(isWordSame(currentWord, "1") || isWordSame(currentWord, "2") || isWordSame(currentWord, "3")))
{
    printf("\n---Masukan tidak sesuai---\n");
    menu();
}
```

    Program di bawah adalah menu 1, menu 2, dan *exception handler* jika pengguna salah memasukkan angka, karakter kartu, kekurangan masukan, atau kelebihan masukan.

```
if (isWordSame(currentWord, "1"))
{
    printf("\nMenu 1\n");
    printf("Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q\n");
    printf("Input 4 Cards = ");
    STARTCOMMAND();
    STARTWORD();
    isLineCorrect(&arr);

    while (!(arr[4]))
    {
        printf("\n---Masukan tidak sesuai---\n");
        printf("Input = ");
        STARTCOMMAND();
        STARTWORD();
        isLineCorrect(&arr);
        // printf("%d\n",arr[0]);···
    }
}
else
{
    printf("\nMenu 2\n");
    printf("Generate 4 numbers\n");
    for (i = 0; i < 4; i++)
    {
        arr[i] = (rand() % (atas - bawah + 1)) + bawah;
    }
    printf("4 random numbers: %d %d %d %d\n", arr[0], arr[1], arr[2], arr[3]);
}
```

Di bawah ini merupakan gambaran sepintas dari fungsi isLineCorrect yaitu mengecek apakah pengguna memasukkan input sesuai dengan ketentuan.

```
void isLineCorrect(int (*arr)[])
{
    /* Mengembalikan true jika kedua kalimat sesuai dengan ketentuan permainan yakni terdiri dari
    A, J, Q, K, 1-10, dan false jika tidak (kalimat di sini berarti satu baris dengan spasi) */

    /* KAMUS */
    int i = 0, n = 0;
    // boolean sesuai = (l.Length != 0);
    (*arr)[4] = 1;

    /* ALGORITMA */
    while (n < 4)
    {
        // if (! (isWordSame(currentWord, 'A') || isWordSame(currentWord, 'K') || isWordSame(curren
        if (isWordSame(currentWord, "A"))
        {
            (*arr)[n] = 1;
            i++;
        }
        else if (isWordSame(currentWord, "J"))
        {
            (*arr)[n] = 11;
            i++;
        }
        else if (isWordSame(currentWord, "Q"))
        {
            (*arr)[n] = 12;
            i++;
        }
        else if (isWordSame(currentWord, "K"))
        {
```

Program Utama Brute Force

```
// Program Utama Brute Force
// waktu mulai
clock_gettime(CLOCK_REALTIME, &awal);
// Permurtasi 4 angka dengan i,j,k,l dan beberapa kondisi
ni = 1;
for (i = 0; i < 4; i++)···
// waktu selesai
clock_gettime(CLOCK_REALTIME, &akhir);
if (neff != -1)···
else···
```

Iterasi di dalam algoritma Brute Force pada program. Penjelasan iterasi ini terdapat pada penjelasan algoritma Brute Force.

```
for (i = 0; i < 4; i++)
{
    for (j = 0; j < 4; j++)
    {
        if (j != i)
        {
            for (k = 0; k < 4; k++)
            {
                if (k != i && k != j)
                {
                    for (l = 0; l < 4; l++)
                    {
                        if (l != i && l != j && l != k)
                        {
                            arrHasil[0] = arr[i];
                            arrHasil[1] = arr[j];
                            arrHasil[2] = arr[k];
                            arrHasil[3] = arr[l];

                            if (!IsIn(arrHasil, &arrPermut, np))
                            {
                                np++;
                                arrPermut[np][0] = arrHasil[i];
                                arrPermut[np][1] = arrHasil[j];
                                arrPermut[np][2] = arrHasil[k];
                                arrPermut[np][3] = arrHasil[l];

                                // Kombinasi, 3 dari 4 buah operasi
                                for (i1 = 0; i1 < 4; i1++)
                                {
                                    for (i2 = 0; i2 < 4; i2++)
                                    {
                                        for (i3 = 0; i3 < 4; i3++)
                                        {
                                            OpHasil[0] = Op[i1];
                                            OpHasil[1] = Op[i2];
                                            OpHasil[2] = Op[i3];
                                            TwentyFourSolutions(arrHasil, OpHasil, &neff);
                                            ni++;
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

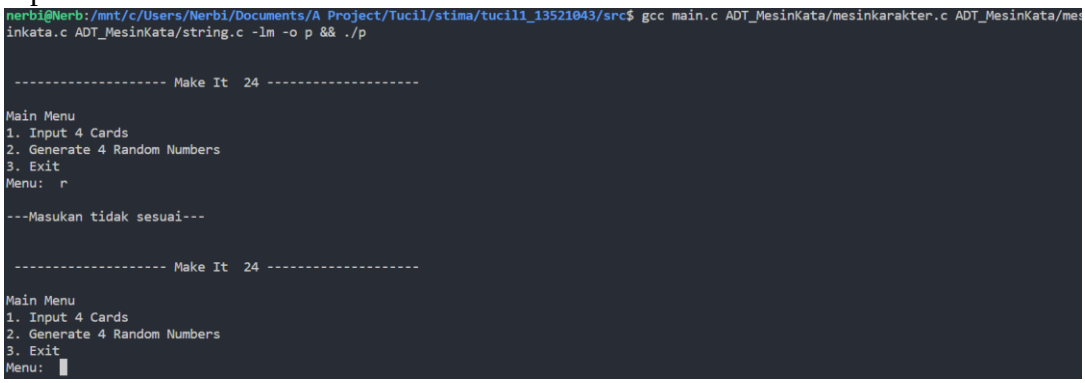
Di bawah ini merupakan program untuk menampilkan solusi dan menyimpan solusi jika simpan bernilai *true*. Perbedaan dengan iterasi di atas adalah pada bagian dalam iterasi program menampilkan solusi dengan fungsi twentyFourG, untuk menyimpannya dengan twentyFourGSave.

```
// Program untuk Menampilkan solusi ke layar
n = 0;
neff = -1;
np = -1;
ni = 1;
for (i = 0; i < 4; i++) ···

// Program untuk menyimpan solusi jika diinginkan
save();
if (isWordSame(currentWord, "y")) ···
if (simpan) ···

waktu = (akhir.tv_sec - awal.tv_sec) + (akhir.tv_nsec - awal.tv_nsec) / MILYAR;
printf("Waktu Eksekusi: %f seconds\n", (waktu));
```

```
// Menu
// menu();
printf("type '0' (Main Menu) and '3' (exit) : ");
STARTCOMMAND();
STARTWORD();
if (isWordSame(currentWord, "0"))
{
    menu();
}
```

D. Screenshot

1. Input Salah



```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 4 3

---Masukan tidak sesuai---
Input = 90 2 3 A

---Masukan tidak sesuai---
Input = 4    5   6  JK

---Masukan tidak sesuai---
Input = 2 3 4 5 6

---Masukan tidak sesuai---
Input = A J K 12

---Masukan tidak sesuai---
Input = fdfadsf adsfadsf

---Masukan tidak sesuai---
Input = A J K Q

37 solutions found
```

2. Input: 6 6 6 6
   Di bawah ini gambar sebelum

```
nerbi@Nerb:/mnt/c/Users/Nerbi/Documents/A Project/Tucil/stima/tucil1_13521043/src$ gcc mai
inkata.c ADT_MesinKata/string.c -lm -o p && ./p


 -------------------- Make It  24 --------------------

Main Menu
1. Input 4 Cards
2. Generate 4 Random Numbers
3. Exit
Menu:  1

Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 6 6 6 6

7 solutions found
((6+6)+6)+6
6+((6+6)+6)
(6+(6+6))+6
6+(6+(6+6))
(6+6)+(6+6)
(6*6)-(6+6)
((6*6)-6)-6

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input: y
Nama File: input1

File tersimpan
Waktu Eksekusi: 0.000022 seconds
type '0' (Main Menu) and '3' (exit) :
```

README.md | main.c M | input1.txt U | Tucil1-Stima-2023.pdf

Stima > Tucil1_13521043 > doc > Hasil > input1.txt

```
Hasil Semua Solusi
Input numbers: 6 6 6 6
((6+6)+6)+6
6+((6+6)+6)
(6+(6+6))+6
6+(6+(6+6))
(6+6)+(6+6)
(6*6)-(6+6)
((6*6)-6)-6
```

File akan tersimpan pada directory doc/Hasil/nama_file.txt dan di atas terletak pada doc/Hasil/input1.txt

Di bawah ini merupakan, solusi-solusi dari fungsi IsIn3 jika dipakai dan beberapa tambahan. Dapat terlihat bahwa solusi-solusi di atas terdapat ekspresi-ekspresi yang bersifat komutatif dari yang lainnya. Sehingga jika dieliminasi yang bersifat komutatif dan hanya diambil satu maka hasilnya seperti di bawah.

```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 6 6 6 6

4 solutions found
((6+6)+6)+6
(6+6)+(6+6)
(6*6)-(6+6)
((6*6)-6)-6

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input:
```

Namun, untuk selanjutnya digunakan algoritma dengan asumsi unik yaitu tidakmasalah jika ada yang komutatif misal (a+b) beda dengan (b+a) karena unik.

3. Input: 2 3 4 5

```
Input 4 Cards = 2 3 4 5

43 solutions found
2*((3+4)+5)
2*(3+(4+5))
2*((3+5)+4)
2*(3+(5+4))
2*((4+3)+5)
2*(4+(3+5))
2*((4+5)+3)
2*(4+(5+3))
2*((5+4)+3)
2*(5+(4+3))
((3-2)+5)*4
(3-(2-5))*4
((3/2)+5)*4
((3+4)+5)*2
3*(4*(5/2))
(3*4)*(5/2)
((3+5)-2)*4
(3+(5-2))*4
3*((5/2)*4)
(3*(5/2))*4
((3+5)+4)*2
(3+(5+4))*2
4*((3-2)+5)
4*(3-(2-5))
4*((3/2)+5)
((4+3)+5)*2
(4+(3+5))*2
4*((3+5)-2)
4*(3+(5-2))
4*(3*(5/2))
(4*3)*(5/2)
((4+5)+3)*2
(4+(5+3))*2
4*((5+3)-2)
4*(5+(3-2))
4*(5+(3/2))
((5-2)+3)*4
(5-(2-3))*4
((5/2)*3)*4
(5/2)*(3*4)
((5+3)+4)*2
(5+(3+4))*2
```

```
Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input: n
Waktu Eksekusi: 0.000104 seconds
type '0' (Main Menu) and '3' (exit) :
```

4. Input: 2 4 4 K

```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 2 4 4 k

---Masukan tidak sesuai---
Input = 2 4 4 K

20 solutions found
2*(4*(13/4))
(2*4)*(13/4)
2*(4*(13/4))
(2*4)*(13/4)
4*((2*13)/4)
4*(2*(13/4))
(4*2)*(13/4)
((4*13)-4)/2
4*(13/(4-2))
4*((13/4)*2)
(4*(13/4))*2
4*(13/(4/2))
4*((2*13)/4)
4*(2*(13/4))
(4*2)*(13/4)
((4*13)-4)/2
4*(13/(4-2))
4*((13/4)*2)
(4*(13/4))*2
4*(13/(4/2))

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input: y
Nama File: in3

File tersimpan
Waktu Eksekusi: 0.000220 seconds
type '0' (Main Menu) and '3' (exit) :
```

Stima > Tucil1_13521043 > doc > Hasil > in3.txt

```
1   Hasil Semua Solusi
2   Input numbers: 2 4 4 13
3   2*(4*(13/4))
4   (2*4)*(13/4)
5   2*(4*(13/4))
6   (2*4)*(13/4)
7   4*((2*13)/4)
8   4*(2*(13/4))
9   (4*2)*(13/4)
10  ((4*13)-4)/2
11  4*(13/(4-2))
12  4*((13/4)*2)
13  (4*(13/4))*2
14  4*(13/(4/2))
15  4*((2*13)/4)
16  4*(2*(13/4))
17  (4*2)*(13/4)
18  ((4*13)-4)/2
19  4*(13/(4-2))
20  4*((13/4)*2)
21  (4*(13/4))*2
22  4*(13/(4/2))
23
```

5. Input: A Q J K

```
39 solutions found
((1/12)+11)+13
(1/12)+(11+13)
1*(12*(13-11))
(1*12)*(13-11)
((1/12)+13)+11
(1/12)+(13+11)
(1+(13/11))*12
((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))
12*((13/11)+1)
13+((1/12)+11)
(13+(1/12))+11
(13-(1/12))+11
13-((1/12)-11)
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)*(1*12)
((13-11)/1)*12
(13-(11/1))*12
((13/11)+1)*12

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input: n
Waktu Eksekusi: 0.000138 seconds
type '0' (Main Menu) and '3' (exit) :
```

6. Input: 2 A 3 A

```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 2 1 2 3

---Masukan tidak sesuai---
Input = 2 A 3 A

No solutions found
```

```
Waktu Eksekusi: 0.000100 seconds
type '0' (Main Menu) and '3' (exit) :
```

7. Input: 7 6 7 2

```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 7 6 7 2

4 solutions found
(7-(7/2))*6
6*(7-(7/2))
6*(7-(7/2))
(7-(7/2))*6

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input: y
Nama File: in4

File tersimpan
Waktu Eksekusi: 0.000099 seconds
type '0' (Main Menu) and '3' (exit) :
```

Stima > Tucil1_13521043 > doc > Hasil > in4.txt

```
1   Hasil Semua Solusi
2   Input numbers: 7 6 7 2
3   (7-(7/2))*6
4   6*(7-(7/2))
5   6*(7-(7/2))
6   (7-(7/2))*6
7
```

8. Input: 10 8 2 4

```
Menu 1
Make it 24 with Cards : A,2,3,4,5,6,7,8,9,10,J,K,Q
Input 4 Cards = 10 8 2 4

144 solutions found
((10+8)+2)+4
10+((8+2)+4)
(10+(8+2))+4
```

```
((2+4)+8)+10
2+((4+8)+10)
(2+(4+8))+10
2+(4+(8+10))
(4+2)+(10+8)

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input:
Waktu Eksekusi: 0.000153 seconds
type '0' (Main Menu) and '3' (exit) :
```

Tidak semua ditampilkan karena terlalu banyak

9. Random 1

```
 -------------------- Make It  24 --------------------

Main Menu
1. Input 4 Cards
2. Generate 4 Random Numbers
3. Exit
Menu:  2

Menu 2
Generate 4 numbers
4 random numbers: 2 2 13 4

26 solutions found
2+((2*13)-4)
(2+(2*13))-4
2+((13*2)-4)
(2+(13*2))-4
((2*13)+2)-4
(2*13)+(2-4)
((2*13)-4)+2
(2*13)-(4-2)
(2*13)-(4/2)
(2-4)+(13*2)
2-(4-(13*2))
2+((13*2)-4)
(2+(13*2))-4
((2*13)+2)-4
(2*13)+(2-4)
((2*13)-4)+2
(2*13)-(4-2)
(2*13)-(4/2)
(2-4)+(13*2)
2-(4-(13*2))
((13*2)-4)+2
(13*2)-(4-2)
(13*2)-(4/2)
((13*2)-4)+2
(13*2)-(4-2)
(13*2)-(4/2)

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input:
```

10. Random 2

```
Menu 2
Generate 4 numbers
4 random numbers: 1 5 13 2

24 solutions found
1+((5*2)+13)
(1+(5*2))+13
1+(13+(5*2))
(1+13)+(5*2)
1+(13+(2*5))
(1+13)+(2*5)
(1+2)*(13-5)
(5-1)*(13/2)
((5*2)+1)+13
(5*2)+(1+13)
((5*2)+13)+1
(5*2)+(13+1)
(5/2)*(13-1)
13+(1+(2*5))
(13+1)+(2*5)
(13-5)*(1+2)
13+((5*2)+1)
(13+(5*2))+1
(13-5)*(2+1)
13+((2*5)+1)
(13+(2*5))+1
(13/2)*(5-1)
((2*5)+13)+1
(2*5)+(13+1)

Apakah Jawaban yang akan dihasilkan ingin disimpan dalam txt file? (y/n)
Input:
```

11. Exit

```
 -------------------- Make It  24 --------------------

Main Menu
1. Input 4 Cards
2. Generate 4 Random Numbers
3. Exit
Menu:  3

 ------------------- terima kasih -------------------

nerbi@Nerb:/mnt/c/Users/Nerbi/Documents/A Project/Tucil/stima/tucil1_13521043/src$
```

E. Link Repository

https://github.com/NerbFox/Tucil1_13521043