LAPORAN TUGAS KECIL IF2211 STRATEGI ALGORITMA

MUHAMMAD DHIWAUL AKBAR - 13521158

A. ALGORITMA BRUTE FORCE

dalam program mencari semua solusi yang mungkin untuk permainan kartu 24 ini , saya menggunakan tiga (for) loop yang mewakili 3 operator yang bisa digunakan. dengan setiap angka yang di loop mewakili satu operator yang akan coba digunakan (misal loop nya mulai dari nol , angka 0 untuk “+” , 1 untuk “-” , 2 untuk “*” ,dan 3 untuk “/”) , serta menggunakan dua pendekatan tanda kurung , dengan dua contoh sebagai berikut :

1. ((1+2) - 3) * 4
2. (1+2) - (3*4)

dan mencobakannya pada semua kemungkinan permutasi 4 bilangan tersebut

B. SOURCE CODE

```c
#include <stdio.h>
#include "tucil1.h"
#include <string.h>


void firstkartu24 (int a ,int b ,int c ,int d ) {
    char op[4] = {'+','-','*','/'} ;
    for (int i = 0 ; i < 4 ; i++) {
        int result = 0 ;
        char op1 = op[i];
        if (i==0) {
            result += (a+b);
        }
        else if (i==1) {
            result += (a-b);
        }
        else if (i==2) {
            result += (a*b);
        }
        else if (i==3) {
            result += (a/b);
        }

        for (int j = 0 ; j < 4 ;j++) {

            char op2 = op[j];
            if (j==0) {
                result += c ;
```

```
            }
            else if (j==1) {
                result -= c;
            }
            else if (j==2) {
                result *= c;
            }
            else if (j==3) {
                result /= c;
            }
            for (int k = 0 ; k < 4 ;k++) {

                char op3 = op[k];
                if (k== 0  ) {
                    result += d ;
                    if (result==24){
                        printf("((%d %c %d) %c %d) %c %d\n",
a,op1,b,op2,c,op3,d) ;
                    }
                }
                else if (k==1) {
                    result -= d;
                    if (result==24){
                        printf("((%d %c %d) %c %d) %c %d\n",
a,op1,b,op2,c,op3,d) ;
                    }
                }
                else if (k==2) {
                    result *= d ;
                    if (result==24){
                        printf("((%d %c %d) %c %d) %c %d\n",
a,op1,b,op2,c,op3,d) ;
                    }
                }
                else if (k==3) {
                    result /= d;
                    if (result==24){
                        printf("((%d %c %d) %c %d) %c %d\n",
a,op1,b,op2,c,op3,d) ;
                    }
                }

            }
```

```
        }
    }
}
void secondkartu24 (int a ,int b ,int c ,int d ) {
    char op[4] = {'+','-','*','/'} ;
    for (int i = 0 ; i < 4 ; i++) {
        int result = 0 ;
        int result1 = 0 ;
        char op1 = op[i];
        if (i==0) {
            result += (a+b);
        }
        else if (i==1) {
            result += (a-b);
        }
        else if (i==2) {
            result += (a*b);
        }
        else if (i==3) {
            result += (a/b);
        }

        for (int j = 0 ; j < 4 ;j++) {

            char op2 = op[j];
            if (j==0) {
                result1 += (c+d) ;
            }
            else if (j==1) {
                result1 += (c-d);
            }
            else if (j==2) {
                result1 += (c*d);
            }
            else if (j==3) {
                result1 += (c/d);
            }
            for (int k = 0 ; k < 4 ;k++) {

                char op3 = op[k];
                if (k== 0  ) {
                    result += result1 ;
                    if (result==24){
```

```
                        printf("(%d %c %d) %c (%d %c %d)\n",
a,op1,b,op3,c,op2,d) ;
                    }
                }
                else if (k==1) {
                    result -= result1;
                    if (result==24){
                        printf("(%d %c %d) %c (%d %c %d)\n",
a,op1,b,op3,c,op2,d) ;
                    }
                }
                else if (k==2) {
                    result *= result1 ;
                    if (result==24){
                        printf("(%d %c %d) %c (%d %c %d)\n",
a,op1,b,op3,c,op2,d) ;
                    }
                }
                else if (k==3) {
                    result /= result1;
                    if (result==24){
                        printf("(%d %c %d) %c (%d %c %d)\n",
a,op1,b,op3,c,op2,d) ;
                    }
                }

            }
        }
    }
}
int sumsolusi (int a, int b, int c, int d) {
    int solusi = 0 ;
    for (int i = 0 ; i < 4 ; i++) {
        int result = 0 ;

        if (i==0) {
            result += a+b;
        }
        else if (i==1) {
            result += a-b;
        }
        else if (i==2) {
            result += a*b;
```

```
        }
        else if (i==3) {
            result += a/b;
        }

        for (int j = 0 ; j < 4 ;j++) {


            if (j==0) {
                result += c ;
            }
            else if (j==1) {
                result -= c;
            }
            else if (j==2) {
                result *= c;
            }
            else if (j==3) {
                result /= c;
            }
            for (int k = 0 ; k < 4 ;k++) {


                if (k== 0  ) {
                    result += d ;
                    if (result==24){
                        solusi ++ ;
                    }
                }
                else if (k==1) {
                    result -= d;
                    if (result==24){
                        solusi ++ ;
                    }
                }
                else if (k==2) {
                    result *= d ;
                    if (result==24){
                        solusi ++ ;
                    }
                }
                else if (k==3) {
                    result /= d;
```

```
                    if (result==24){
                        solusi ++ ;
                    }
                }

            }
        }
    }

    return solusi ;
}
int sumsolusi1 (int a, int b, int c, int d) {
    int solusi2 = 0 ;
    for (int i = 0 ; i < 4 ; i++) {
        int result = 0 ;
        int result1 = 0 ;

        if (i==0) {
            result += (a+b);
        }
        else if (i==1) {
            result += (a-b);
        }
        else if (i==2) {
            result += (a*b);
        }
        else if (i==3) {
            result += (a/b);
        }

        for (int j = 0 ; j < 4 ;j++) {


            if (j==0) {
                result1 += (c+d) ;
            }
            else if (j==1) {
                result1 += (c-d);
            }
            else if (j==2) {
                result1 += (c*d);
            }
            else if (j==3) {
```

```
                result1 += (c/d);
            }
            for (int k = 0 ; k < 4 ;k++) {


                if (k== 0  ) {
                    result += result1 ;
                    if (result==24){
                        solusi2 ++ ;
                    }
                }
                else if (k==1) {
                    result -= result1;
                    if (result==24){
                        solusi2 ++ ;
                    }
                }
                else if (k==2) {
                    result *= result1 ;
                    if (result==24){
                        solusi2 ++ ;
                    }
                }
                else if (k==3) {
                    result /= result1;
                    if (result==24){
                        solusi2 ++ ;
                    }
                }

            }
        }
    }
    return solusi2 ;
}
int inputan(){
    char str;
    int n = 0 ;
    scanf("%c" , str) ;
    if (str=='A' || str=='a') {
        n += 1 ;
    }
    else if (str=='2') {
```

```
        n += 2 ;
    }
    else if (str=='3') {
        n += 3 ;
    }
    else if (str=='4') {
        n += 4 ;
    }
    else if (str=='5') {
        n += 5 ;
    }
    else if (str=='6') {
        n += 6 ;
    }
    else if (str=='7') {
        n += 7 ;
    }
    else if (str=='8') {
        n += 8 ;
    }
    else if (str=='9') {
        n += 9 ;
    }
    else if (str=='J' || str=='j') {
        n += 11 ;
    }
    else if (str=='Q' || str=='q') {
        n += 12 ;
    }
    else if (str=='K' || str=='k') {
        n += 13 ;
    }
    else {
        n += 10 ;
    }
    return n;
}

int main() {


    int d[4] = {0,0,0,0};
    for (int r=0 ; r<4 ; r++ ) {
```

```
        d[r] += inputan() ;
    }
    int s = sumsolusi(d[0],d[1],d[2],d[3]) +
sumsolusi(d[0],d[1],d[3],d[2]) +sumsolusi(d[0],d[2],d[1],d[3])
+sumsolusi(d[0],d[2],d[3],d[1]) +sumsolusi(d[0],d[3],d[1],d[2])
+sumsolusi(d[0],d[3],d[2],d[1]) +sumsolusi(d[1],d[0],d[2],d[3])
+sumsolusi(d[1],d[0],d[3],d[2]) +sumsolusi(d[2],d[0],d[1],d[3])
+sumsolusi(d[2],d[0],d[3],d[1]) +sumsolusi(d[3],d[0],d[1],d[2])
+sumsolusi(d[3],d[0],d[2],d[1]) +sumsolusi(d[1],d[2],d[0],d[3]) +
sumsolusi(d[1],d[2],d[3],d[0]) + sumsolusi(d[1],d[3],d[0],d[2]) +
sumsolusi(d[1],d[3],3,d[0]) +  sumsolusi(d[2],d[1],d[0],d[3]) +
sumsolusi(d[2],d[1],d[3],d[0]) + sumsolusi(d[3],d[1],d[0],d[2]) +
sumsolusi(d[3],d[1],d[2],d[0]) +   sumsolusi(d[2],d[3],d[0],d[1])
+sumsolusi(d[2],d[3],d[1],d[0]) + sumsolusi(d[3],d[2],d[0],d[1])
+sumsolusi(d[3],d[2],d[1],d[0]) + sumsolusi1(d[0],d[1],d[2],d[3])
+ sumsolusi1(d[0],d[1],d[3],d[2])
+sumsolusi1(d[0],d[2],d[1],d[3]) +sumsolusi1(d[0],d[2],d[3],d[1])
+sumsolusi1(d[0],d[3],d[1],d[2]) +sumsolusi1(d[0],d[3],d[2],d[1])
+sumsolusi1(d[1],d[0],d[2],d[3]) +sumsolusi1(d[1],d[0],d[3],d[2])
+sumsolusi1(d[2],d[0],d[1],d[3]) +sumsolusi1(d[2],d[0],d[3],d[1])
+sumsolusi1(d[3],d[0],d[1],d[2]) +sumsolusi1(d[3],d[0],d[2],d[1])
+sumsolusi1(d[1],d[2],d[0],d[3]) +
sumsolusi1(d[1],d[2],d[3],d[0]) + sumsolusi1(d[1],d[3],d[0],d[2])
+ sumsolusi1(d[1],d[3],d[2],d[0]) +
sumsolusi1(d[2],d[1],d[0],d[3]) + sumsolusi1(d[2],d[1],d[3],d[0])
+ sumsolusi1(d[3],d[1],d[0],d[2]) +
sumsolusi1(d[3],d[1],d[2],d[0]) +
sumsolusi1(d[2],d[3],d[0],d[1]) +sumsolusi1(d[2],d[3],d[1],d[0])
+ sumsolusi1(d[3],d[2],d[0],d[1])
+sumsolusi1(d[3],d[2],d[1],d[0]) ;
    printf ("ada %d solusi yang memungkinkan\n", s);

    firstkartu24(d[0],d[1],d[2],d[3]) ;
    firstkartu24(d[0],d[1],d[3],d[2]) ;
    firstkartu24(d[0],d[2],d[1],d[3]) ;
    firstkartu24(d[0],d[2],d[3],d[1]) ;
    firstkartu24(d[0],d[3],d[1],d[2]) ;
    firstkartu24(d[0],d[3],d[2],d[1]) ;
    firstkartu24(d[1],d[0],d[2],d[3]) ;
    firstkartu24(d[1],d[0],d[3],d[2]) ;
    firstkartu24(d[2],d[0],d[1],d[3]) ;
    firstkartu24(d[2],d[0],d[3],d[1]) ;
    firstkartu24(d[3],d[0],d[1],d[2]) ;
```

```
    firstkartu24(d[3],d[0],d[2],d[1]) ;
    firstkartu24(d[1],d[2],d[0],d[3]) ;
    firstkartu24(d[1],d[2],d[3],d[0]) ;
    firstkartu24(d[1],d[3],d[0],d[2]) ;
    firstkartu24(d[1],d[3],d[2],d[0]) ;
    firstkartu24(d[2],d[1],d[0],d[3]) ;
    firstkartu24(d[2],d[1],d[3],d[0]) ;
    firstkartu24(d[3],d[1],d[0],d[2]) ;
    firstkartu24(d[3],d[1],d[2],d[0]) ;
    firstkartu24(d[2],d[3],d[0],d[1]) ;
    firstkartu24(d[2],d[3],d[1],d[0]) ;
    firstkartu24(d[3],d[2],d[0],d[1]) ;
    firstkartu24(d[3],d[2],d[1],d[0]) ;
    secondkartu24(d[0],d[1],d[2],d[3]) ;
    secondkartu24(d[0],d[1],d[3],d[2]) ;
    secondkartu24(d[0],d[2],d[1],d[3]) ;
    secondkartu24(d[0],d[2],d[3],d[1]) ;
    secondkartu24(d[0],d[3],d[1],d[2]) ;
    secondkartu24(d[0],d[3],d[2],d[1]) ;
    secondkartu24(d[1],d[0],d[2],d[3]) ;
    secondkartu24(d[1],d[0],d[3],d[2]) ;
    secondkartu24(d[2],d[0],d[1],d[3]) ;
    secondkartu24(d[2],d[0],d[3],d[1]) ;
    secondkartu24(d[3],d[0],d[1],d[2]) ;
    secondkartu24(d[3],d[0],d[2],d[1]) ;
    secondkartu24(d[1],d[2],d[0],d[3]) ;
    secondkartu24(d[1],d[2],d[3],d[0]) ;
    secondkartu24(d[1],d[3],d[0],d[2]) ;
    secondkartu24(d[1],d[3],d[2],d[0]) ;
    secondkartu24(d[2],d[1],d[0],d[3]) ;
    secondkartu24(d[2],d[1],d[3],d[0]) ;
    secondkartu24(d[3],d[1],d[0],d[2]) ;
    secondkartu24(d[3],d[1],d[2],d[0]) ;
    secondkartu24(d[2],d[3],d[0],d[1]) ;
    secondkartu24(d[2],d[3],d[1],d[0]) ;
    secondkartu24(d[3],d[2],d[0],d[1]) ;
    secondkartu24(d[3],d[2],d[1],d[0]) ;

    return 0 ;
}
```

C. SCREENSHOT INPUT & OUTPUT

```
6 2 3 4
ada 119 solusi yang memungkinkan
((6 + 2) * 3) - 4
((6 + 2) * 3) / 4
((6 / 2) + 3) * 4
((6 + 2) - 4) * 3
((6 + 2) / 4) * 3
((6 - 2) + 4) * 3
((6 - 3) * 2) * 4
((6 * 3) + 2) + 4
((6 - 3) * 4) * 2
((6 * 3) + 4) + 2
((6 * 4) - 2) - 3
((6 * 4) - 2) / 3
((6 * 4) / 2) - 3
((6 * 4) / 2) / 3
((6 * 4) - 3) - 2
((6 * 4) - 3) / 2
((6 * 4) / 3) - 2
((6 * 4) / 3) / 2
((2 + 6) * 3) - 4
((2 + 6) * 3) / 4
((2 + 6) - 4) * 3
((2 + 6) / 4) * 3
((3 * 6) + 2) + 4
((3 * 6) + 4) + 2
((4 * 6) - 2) - 3
((4 * 6) - 2) / 3
((4 * 6) / 2) - 3
((4 * 6) / 2) / 3
((4 * 6) - 3) - 2
((4 * 6) - 3) / 2
((4 * 6) / 3) - 2
((4 * 6) / 3) / 2
((2 * 3) - 6) * 4
((2 * 3) / 6) * 4
((2 / 3) + 6) * 4
((2 * 3) * 4) - 6
((2 * 3) * 4) / 6
((2 / 3) + 4) * 6
((2 * 4) - 6) * 3
((2 * 4) / 6) * 3
((2 + 4) * 3) + 6
((2 * 4) * 3) - 6
((2 * 4) * 3) / 6
((3 - 2) * 6) * 4
((3 * 2) - 6) * 4
((3 * 2) / 6) * 4
((3 / 2) * 6) * 4
((3 - 2) * 4) * 6
```

```
((3 * 2) * 4) - 6
((3 * 2) * 4) / 6
((3 / 2) * 4) * 6
((4 - 2) + 6) * 3
((4 * 2) - 6) * 3
((4 * 2) / 6) * 3
((4 / 2) + 6) * 3
((4 + 2) * 3) + 6
((4 * 2) * 3) - 6
((4 * 2) * 3) / 6
((3 * 4) - 6) * 2
((3 * 4) / 6) * 2
((3 * 4) * 2) - 6
((3 * 4) * 2) / 6
((4 * 3) - 6) * 2
((4 * 3) / 6) * 2
((4 * 3) * 2) - 6
((4 * 3) * 2) / 6
(6 - 2) * (3 - 4)
(6 - 2) + (4 * 3)
(6 / 2) * (4 - 3)
(6 / 2) + (4 / 3)
(6 * 3) + (2 + 4)
(6 / 3) * (2 * 4)
(6 / 3) * (2 / 4)
(6 - 3) * (4 - 2)
(6 * 3) + (4 + 2)
(6 * 4) - (2 + 3)
(6 * 4) / (2 + 3)
(6 * 4) - (2 - 3)
(6 * 4) / (2 - 3)
(6 * 4) - (2 * 3)
(4 * 6) - (3 * 2)
(4 * 6) / (3 * 2)
(4 * 6) - (3 / 2)
(4 * 6) / (3 / 2)
(2 / 4) + (3 * 6)
(2 / 4) + (3 / 6)
(4 - 2) * (6 - 3)
(4 / 2) * (6 - 3)
(3 * 4) + (6 - 2)
(3 / 4) + (6 * 2)
(4 - 3) * (6 * 2)
(4 * 3) + (6 - 2)
(4 / 3) * (6 * 2)
```

```
8 2 3 4
ada 106 solusi yang memungkinkan
((8 - 2) - 3) * 4
((8 - 2) / 3) * 4
((8 - 2) * 4) - 3
((8 - 2) * 4) / 3
((8 / 2) + 4) * 3
((8 * 3) - 2) - 4
((8 * 3) - 2) / 4
((8 * 3) / 2) - 4
((8 * 3) / 2) / 4
((8 * 3) - 4) - 2
((8 * 3) - 4) / 2
((8 * 3) / 4) - 2
((8 * 3) / 4) / 2
((8 + 4) * 2) - 3
((8 + 4) * 2) / 3
((8 - 4) * 2) * 3
((8 + 4) - 3) * 2
((8 + 4) / 3) * 2
((8 - 4) * 3) * 2
((3 * 8) - 2) - 4
((3 * 8) - 2) / 4
((3 * 8) / 2) - 4
((3 * 8) / 2) / 4
((3 * 8) - 4) - 2
((3 * 8) - 4) / 2
((3 * 8) / 4) - 2
((3 * 8) / 4) / 2
((4 + 8) * 2) - 3
((4 + 8) * 2) / 3
((4 + 8) - 3) * 2
((4 + 8) / 3) * 2
((2 * 3) - 8) * 4
((2 * 3) / 8) * 4
((2 - 3) + 4) * 8
((2 * 3) * 4) - 8
((2 * 3) * 4) / 8
((2 * 4) - 8) * 3
((2 * 4) / 8) * 3
((2 / 4) + 8) * 3
((2 * 4) * 3) - 8
((2 * 4) * 3) / 8
((2 / 4) + 3) * 8
((3 * 2) - 8) * 4
((3 * 2) / 8) * 4
((3 * 2) * 4) - 8
((3 * 2) * 4) / 8
((4 * 2) - 8) * 3
((4 * 2) / 8) * 3
((4 * 2) * 3) - 8
```

```
((4 * 2) * 3) / 8
((3 * 4) - 8) * 2
((3 * 4) / 8) * 2
((3 * 4) * 2) - 8
((3 * 4) * 2) / 8
((4 * 3) - 8) * 2
((4 * 3) / 8) * 2
((4 - 3) + 2) * 8
((4 * 3) * 2) - 8
((4 * 3) * 2) / 8
((4 / 3) + 2) * 8
(8 - 2) + (3 * 4)
(8 - 2) + (3 / 4)
(8 / 2) * (3 - 4)
(8 * 2) + (4 - 3)
(8 / 2) + (4 * 3)
(8 * 3) - (2 + 4)
(8 * 3) / (2 + 4)
(2 * 4) + (8 - 3)
(4 * 2) + (8 - 3)
```

```
5 2 3 4
ada 44 solusi yang memungkinkan
((5 - 2) + 3) * 4
((5 / 2) * 3) * 4
((5 / 2) * 4) * 3
((5 + 3) + 4) * 2
((5 + 4) + 3) * 2
((3 + 5) + 4) * 2
((4 + 5) + 3) * 2
((2 * 3) - 5) * 4
((2 * 3) / 5) * 4
((2 * 3) * 4) - 5
((2 * 3) * 4) / 5
((2 * 4) - 5) * 3
((2 * 4) / 5) * 3
((2 * 4) * 3) - 5
((2 * 4) * 3) / 5
((3 - 2) + 5) * 4
((3 * 2) - 5) * 4
((3 * 2) / 5) * 4
((3 / 2) + 5) * 4
((3 * 2) * 4) - 5
((3 * 2) * 4) / 5
((4 * 2) - 5) * 3
((4 * 2) / 5) * 3
((4 * 2) * 3) - 5
((4 * 2) * 3) / 5
((3 + 4) + 5) * 2
((3 * 4) - 5) * 2
((3 * 4) / 5) * 2
((3 * 4) * 2) - 5
((3 * 4) * 2) / 5
((4 + 3) + 5) * 2
((4 * 3) - 5) * 2
((4 * 3) / 5) * 2
((4 * 3) * 2) - 5
((4 * 3) * 2) / 5
(5 - 2) * (4 - 3)
(5 - 2) + (4 / 3)
(5 - 3) * (2 * 4)
(5 - 3) * (2 / 4)
(5 + 3) + (4 * 2)
(5 * 4) + (2 - 3)
(3 + 5) + (4 * 2)
(4 * 5) + (2 - 3)
(2 - 4) + (5 / 3)
```

```
10 2 3 4
ada 63 solusi yang memungkinkan
((10 - 2) * 3) - 4
((10 - 2) * 3) / 4
((10 * 2) - 3) + 4
((10 * 2) / 3) + 4
((10 - 2) - 4) * 3
((10 - 2) / 4) * 3
((10 * 2) + 4) - 3
((10 * 2) + 4) / 3
((10 / 3) * 2) * 4
((10 / 3) * 4) * 2
((10 - 4) + 2) * 3
((2 * 10) - 3) + 4
((2 * 10) / 3) + 4
((2 * 10) + 4) - 3
((2 * 10) + 4) / 3
((2 * 3) - 10) * 4
((2 * 3) / 10) * 4
((2 * 3) * 4) - 10
((2 * 3) * 4) / 10
((2 - 4) + 10) * 3
((2 * 4) - 10) * 3
((2 * 4) / 10) * 3
((2 * 4) * 3) - 10
((2 * 4) * 3) / 10
((3 * 2) - 10) * 4
((3 * 2) / 10) * 4
((3 * 2) * 4) - 10
(3 / 4) + (2 / 10)
(4 * 3) + (10 + 2)
(4 - 3) * (2 * 10)
(4 - 3) * (2 / 10)
(4 * 3) + (2 + 10)
(4 / 3) * (2 * 10)
(4 / 3) * (2 / 10)
```

```
9 2 3 4
ada 59 solusi yang memungkinkan
((9 / 2) + 4) * 3
((9 + 3) * 2) - 4
((9 + 3) * 2) / 4
((9 - 3) - 2) * 4
((9 - 3) / 2) * 4
((9 / 3) * 2) * 4
((9 + 3) - 4) * 2
((9 + 3) / 4) * 2
((9 - 3) * 4) - 2
((9 - 3) * 4) / 2
((9 / 3) * 4) * 2
((3 + 9) * 2) - 4
((3 + 9) * 2) / 4
((3 + 9) - 4) * 2
((3 + 9) / 4) * 2
((2 * 3) - 9) * 4
((2 * 3) / 9) * 4
((2 * 3) * 4) - 9
((2 * 3) * 4) / 9
((2 * 4) - 9) * 3
((2 * 4) / 9) * 3
(3 * 2) + (9 - 4)
(4 + 2) + (9 - 3)
(4 - 2) * (9 + 3)
(4 / 2) * (9 + 3)
(4 - 2) * (3 + 9)
(4 / 2) * (3 + 9)
```

```
7 2 3 4
ada 37 solusi yang memungkinkan
((7 / 2) + 3) * 4
((7 + 3) * 2) + 4
((7 - 3) + 2) * 4
((3 + 7) * 2) + 4
((2 - 3) + 7) * 4
((2 * 3) - 7) * 4
((2 * 3) / 7) * 4
((2 * 3) * 4) - 7
((2 * 3) * 4) / 7
((2 * 4) - 7) * 3
((2 * 4) / 7) * 3
((2 * 4) * 3) - 7
((2 * 4) * 3) / 7
((3 * 2) - 7) * 4
((3 * 2) / 7) * 4
((3 * 2) * 4) - 7
((3 * 2) * 4) / 7
((4 * 2) - 7) * 3
((4 * 2) / 7) * 3
((4 * 2) * 3) - 7
((4 * 2) * 3) / 7
((3 * 4) - 7) * 2
((3 * 4) / 7) * 2
((3 * 4) * 2) - 7
((3 * 4) * 2) / 7
((4 * 3) - 7) * 2
((4 * 3) / 7) * 2
((4 * 3) * 2) - 7
((4 * 3) * 2) / 7
(7 / 2) * (4 - 3)
(7 / 2) + (4 / 3)
(7 - 3) * (2 + 4)
(7 / 3) * (2 * 4)
(7 / 3) * (2 / 4)
(7 - 3) * (4 + 2)
(7 + 4) + (3 / 2)
(4 + 7) + (3 / 2)
```

D. LINK REPOSITORY
https://github.com/Dhiwa27/Tucil1_13521158.git