Tugas Kecil 1 IF2211 Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Disusun oleh:

13521132 – Dhanika Novlisariyanti

INSTITUT TEKNOLOGI BANDUNG

2023

DAFTAR ISI

BAB I
DESKRIPSI MASALAH

1.1 Algoritma Brute Force

Algoritma Brute Force adalah metode dalam menyelesaikan masalah secara langsung yang bergantung pada kekuatan *computing* dan mencoba segala kemungkinan dibandingkan teknik-teknik lain yang berfokus pada efisiensi. Biasanya algoritma brute force didasarkan pada pernyataan pada persoalan dan defisini atau konsep yang dilibatkan.

1.2 Permainan Kartu 24

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.

1.3 Pendekatan Algoritma Brute Force dalam menyelesaikan Permainan Kartu 24

1. Empat angka dari angka satu sampai lima belas dikumpulkan dari input pengguna atau dikumpulkan secara acak
2. Dalam proses perhitungannya, terdapat empat operator +, -, *, dan /. Untuk mendapatkan total 24, dapat menentukan kombinasi dari penggunaan kurung buka dan tutup. Terdapat lima kombinasi yang dapat digunakan

$$((a\ op\ b)\ op\ c)\ op\ d$$
$$(a\ op\ b)\ op\ (c\ op\ d)$$
$$(a\ op\ (b\ op\ c))\ op\ d$$
$$a\ op\ ((b\ op\ c)\ op\ d)$$
$$a\ op\ (b\ op\ (c\ op\ d))$$

   Selain menggunakan kombinasi operator, terdapat kombinasi dari keempat angka tersebut dan selalu unik sehingga terdapat 4x3x2x1 = 24 kombinasi.
3. Keempat angka dicari semua kemungkinan terlebih dahulu, lalu operator akan di telurusi satu-satu segala kemungkinan untuk mencari yang mencapai 24.
4. Keluarkan solusi ke terminal atau output file.

BAB 2
IMPLEMENTASI PROGRAM

Dalam pembuatan program ini, penulis menggunakan bahasa pemrograman Java. Struktur dari program ini terbagi menjadi Main. Java dan solution.java.

2.1. Main.java

File main digunakan sebagai driver utama untuk menjalankan program.

2.2 solution.java

a. public double fOp(char op, double left, double right)

   Atribut ini digunakan untuk mengembalikan hasil dari operator, angka pertama dan angka kedua.

b. public List<List<Integer>> permute(int[] array)

   Atribut ini digunakan untuk mengakomodasi permutasi angka dengan menerima input array integer dan mengembalikan dalam bentuk list of list dikarenakan untuk pemudahan pemasukan angka ke dalam list dibandingkan menggunakan array biasa.

c. public void permuteNumbers(int[] nums, List<Integer> insideAns,List<List<Integer>> ans,boolean freq[])

   Atribut ini digunakan untuk mengakomodasi permutasi angka dengan parameter array integer, List integer, List of List integer, dan Boolean.

d. public int getSolutionTotal(List<List<Integer>> num)

   Atribut ini digunakan untuk menghitung jumlah total solusi yang menghasilkan 24 dengan menerima input List of List integer.

e. public void printSolution(List<List<Integer>> num)

   Atribut ini digunakan untuk mengeluarkan hasil di terminal dengan parameter List of List integer.

f. public void writetoFile(List<List<Integer>> num, int[] array, String nameFile, double executionTime)

   Atribut ini digunakan untuk menulis hasil kombinasi yang menghasilkan 24, total solusi, dan waktu eksekusi kepada File.

BAB 3
SOURCE CODE

3.1 Repository Program

Link Repository: https://github.com/dhanikanovlisa/Tucil1_13521132.git

3.2 Source Code Program

```java
import java.util.*;
import java.util.List;
import java.util.InputMismatchException;

public class Main {

    Run | Debug
    public static void main(String[] args) {

        boolean start = true;
        Scanner sc = new Scanner(System.in);

        while (start) {

            // Menu
            System.out.println(x: "----------------------------------------------------------------");
            System.out.println(x: "Welcome");
            System.out.println(x: "Menu");
            System.out.println(x: "1. Input Number");
            System.out.println(x: "2. Generate number");
            System.out.println(x: "3. Exit");
            System.out.println(x: "----------------------------------------------------------------");

            try {
                // Input Menu
                System.out.print(s: "Your choice: ");
                int[] array = new int[4];
                int menuChoice = sc.nextInt();
                if (menuChoice == 1) {
```

Gambar 1 Main.java(1)

```java
                if (menuChoice == 1) {

                    boolean startInput = true;
                    boolean outToFile = true;
                    while (startInput) {
                        System.out.println(x: "Please input 4 number ranging from 0-15: ");

                        for (int i = 0; i < 4; i++) {
                            array[i] = sc.nextInt();
                        }

                        // Check
                        boolean found = true;
                        int j = 0;
                        while (j < 4 && found) {
                            if (array[j] <= 0 || array[j] > 15) {
                                found = false;
                            } else {
                                j++;
                            }

                        }
                        if (found) {
                            startInput = false;
                        } else {
                            startInput = true;
                        }
```

Gambar 2 Main.java(2)

```
            long startTime = System.nanoTime();
            solution permutation = new solution();
            List<List<Integer>> ans = permutation.permute(array);

            Set<Set<Integer>> ans2 = new HashSet(ans);
            List<List<Integer>> ans3 = new ArrayList(ans2);

            int total = permutation.getSolutionTotal(ans3);
            System.out.println();
            System.out.println("Total solusi: " + total);
            permutation.printSolution(ans3);

            long stopTime = System.nanoTime();
            double duration = (stopTime - startTime) / 100000;

            System.out.println("Execution Time: " + duration + "ms");
            System.out.println();

            while (outToFile) {
                System.out.println(x: "Do you want to keep it as a file?");
                System.out.println(x: "1. Yes");
                System.out.println(x: "2. No");
                System.out.print(s: "Insert number: ");
                int toFile = sc.nextInt();

                if (toFile == 1) {
                    System.out.print(s: "Write name file: ");
                    sc.nextLine();
                    String inputNameFile = sc.nextLine();
                    permutation.writetoFile(ans3, array, inputNameFile, duration);
                    System.out.println(x: "Succesfully make file");
```

Gambar 3 Main.java(3)

```
                if (toFile == 1) {
                    System.out.print(s: "Write name file: ");
                    sc.nextLine();
                    String inputNameFile = sc.nextLine();
                    permutation.writetoFile(ans3, array, inputNameFile, duration);
                    System.out.println(x: "Succesfully make file");
                    outToFile = false;
                } else if (toFile == 2) {
                    outToFile = false;
                } else {
                    System.out.println(x: "Please insert menu number between 1 - 2.\n");
                }

            }

        } else if (menuChoice == 2) {

            // Generate random number
            Random rand = new Random();
            boolean outtoFileRandom = true;
            for (int i = 0; i < 4; i++) {
                int randomNum = rand.nextInt(bound: 14) + 1;
                array[i] = randomNum;
            }

            System.out.println(x: "Here are your numbers: ");
            for (int i = 0; i < 4; i++) {
                System.out.println(array[i]);
            }
```

Gambar 4 Main.java(4)

```
        } else if (menuChoice == 2) {

            // Generate random number
            Random rand = new Random();
            boolean outtoFileRandom = true;
            for (int i = 0; i < 4; i++) {
                int randomNum = rand.nextInt(bound: 14) + 1;
                array[i] = randomNum;
            }

            System.out.println(x: "Here are your numbers: ");
            for (int i = 0; i < 4; i++) {
                System.out.println(array[i]);
            }

            long startTimeRandom = System.nanoTime();
            solution permutation = new solution();
            List<List<Integer>> ansRandom = permutation.permute(array);
            int totalRandom = permutation.getSolutionTotal(ansRandom);
            Set<Set<Integer>> ansRandom2 = new HashSet(ansRandom);
            List<List<Integer>> ansRandom3 = new ArrayList(ansRandom2);
            System.out.println("Total solusi: " + totalRandom);
            permutation.printSolution(ansRandom3);

            long stopTimeRandom = System.nanoTime();
            double durationRandom = (stopTimeRandom - startTimeRandom) / 100000;

            System.out.println("Execution Time: " + durationRandom + "ms");
            System.out.println();
```

Gambar 5 Main.java(5)

```
            while (outtoFileRandom) {
                System.out.println(x: "Do you want to keep it as a file?");
                System.out.println(x: "1. Yes");
                System.out.println(x: "2. No");
                System.out.print(s: "Insert number: ");
                int toFile = sc.nextInt();
                if (toFile == 1) {
                    System.out.print(s: "Write name file: ");
                    sc.nextLine();
                    String inputNameFileRandom = sc.nextLine();
                    permutation.writetoFile(ansRandom3, array, inputNameFileRandom, durationRandom);
                    System.out.println(x: "Succesfully make file");
                    outtoFileRandom = false;
                } else if (toFile == 2) {
                    outtoFileRandom = false;
                } else {
                    System.out.println(x: "Please insert menu number between 1 - 2.\n");
                }
            }
        }

        else if (menuChoice == 3) {
            start = false;
        } else {
            System.out.println(x: "Please insert menu number between 1 - 3.\n");
        }

    } catch (InputMismatchException e) {
        sc.nextLine();
        System.out.println(x: "Integers only, please.");
    }
```

Gambar 6 Main.java(6)

```java
import java.util.ArrayList;
import java.util.List;
import java.io.FileWriter;
import java.io.IOException;

public class solution {

    char[] op = { '+', '-', '*', '/' };
    double result = 24;

    public double fOp(char op, double left, double right) {
        double result = 0;
        if (op == '+') {
            result += left + right;
        } else if (op == '-') {
            result += left - right;
        } else if (op == '*') {
            result += left * right;
        } else if (op == '/') {
            result += left / right;
        }
        return result;
    }

    public List<List<Integer>> permute(int[] array) {
        List<List<Integer>> ans = new ArrayList<>();
        List<Integer> insideAns = new ArrayList<>();
        boolean freq[] = new boolean[array.length];
        permuteNumbers(array, insideAns, ans, freq);
        return ans;
    }
```

Gambar 7 solution.java(1)

```java
public void permuteNumbers(int[] nums, List<Integer> insideAns, List<List<Integer>> ans, boolean freq[]){
    if (insideAns.size() == nums.length) {
        ans.add(new ArrayList<>(insideAns));
        return;
    }
    for (int i = 0; i < nums.length; i++) {
        if (!freq[i]) {
            freq[i] = true;
            insideAns.add(nums[i]);
            permuteNumbers(nums, insideAns, ans, freq);
            insideAns.remove(insideAns.size() - 1);
            freq[i] = false;

        }
    }
}
```

Gambar 8 solution.java(2)

```java
public int getSolutionTotal(List<List<Integer>> num) {

    int count = 0;
    for (int i = 0; i < num.size(); i++) {

        for (int a = 0; a < 4; a++) {
            for (int b = 0; b < 4; b++) {
                for (int c = 0; c < 4; c++) {

                    if (fOp(op[c], fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)), num.get(i).get(index: 2)),
                            num.get(i).get(index: 3)) == result) {
                        count++;
                        // (a op (b op c)) op d
                    }
                    if (fOp(op[c], fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2))),
                            num.get(i).get(index: 3)) == result) {
                        count++;
                    }
                    if (fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)),
                            fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3))) == result) {
                        count++;
                    }
                    if (fOp(op[a], num.get(i).get(index: 0), fOp(op[c], fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2)),
                            num.get(i).get(index: 3))) == result) {
                        count++;
                    }
                    if (fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1),
                            fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3)))) == result) {
                        count++;
                    }
                }
            }
        }
    }

    return count;
```

Gambar 9 solution.java(3)

```java
public void printSolution(List<List<Integer>> num) {
    System.out.println(x: "-----------------------------------------------");
    for (int i = 0; i < num.size(); i++) {
        for (int a = 0; a < 4; a++) {
            for (int b = 0; b < 4; b++) {
                for (int c = 0; c < 4; c++) {

                    if (fOp(op[c], fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)), num.get(i).get(index: 2)),
                            num.get(i).get(index: 3)) == result) {
                        System.out.println(
                                "( ( " + num.get(i).get(index: 0) + " " + op[a] + " " + num.get(i).get(index: 1) + " ) " + op[b]
                                        + " " + num.get(i).get(index: 2) + " ) " + op[c] + " " + num.get(i).get(index: 3));
                        // (a op (b op c)) op d
                    }
                    if (fOp(op[c], fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2))),
                            num.get(i).get(index: 3)) == result) {
                        System.out.println("( " + num.get(i).get(index: 0) + " " + op[a] + " ( " + num.get(i).get(index: 1) + " "
                                + op[b] + " " + num.get(i).get(index: 2) + " ) ) " + op[c] + " " + num.get(i).get(index: 3));
                    }
                    if (fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)),
                            fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3))) == result) {
                        System.out.println("( " + num.get(i).get(index: 0) + " " + op[a] + " " + num.get(i).get(index: 1) + " ) "
                                + op[b] + " ( " + num.get(i).get(index: 2) + " " + op[c] + " " + num.get(i).get(index: 3) + " )");
                    }
                    if (fOp(op[a], num.get(i).get(index: 0), fOp(op[c], fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2)),
                            num.get(i).get(index: 3))) == result) {
                        System.out.println(num.get(i).get(index: 0) + " " + op[a] + " ( ( " + num.get(i).get(index: 1) + " "
                                + op[b] + " " + num.get(i).get(index: 2) + " ) " + op[c] + " " + num.get(i).get(index: 3) + " )");

                    }
                    if (fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1),
                            fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3)))) == result) {
                        System.out.println(num.get(i).get(index: 0) + " " + op[a] + " ( " + num.get(i).get(index: 1) + " " + op[b]
                                + " ( " + num.get(i).get(index: 2) + " " + op[c] + " " + num.get(i).get(index: 3) + " ) )");
                    }
                }
            }
        }
    }
```

Gambar 10 solution.java(4)

```java
public void writetoFile(List<List<Integer>> num, int[] array, String nameFile, double executionTime) {

    int total = getSolutionTotal(num);

    try {

        FileWriter writeFile = new FileWriter(nameFile + ".txt");
        for (int idx = 0; idx < 4; idx++) {
            writeFile.write(array[idx] + " ");
        }
        writeFile.write(str: "\n");
        writeFile.write("Total Solution: " + total + "\n");
        writeFile.write("Execution Time: " + executionTime + "ms \n");

        for (int i = 0; i < num.size(); i++) {
            for (int a = 0; a < 4; a++) {
                for (int b = 0; b < 4; b++) {
                    for (int c = 0; c < 4; c++) {

                        if (fOp(op[c],
                                fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)), num.get(i).get(index: 2)),
                                num.get(i).get(index: 3)) == result) {
                            writeFile.write("( ( " + num.get(i).get(index: 0) + " " + op[a] + " " + num.get(i).get(index: 1)
                                    + " ) " + op[b] + " " + num.get(i).get(index: 2) + " ) " + op[c] + " "
                                    + num.get(i).get(index: 3) + "\n");
                            // (a op (b op c)) op d
                        }
                        if (fOp(op[c],
                                fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2))),
                                num.get(i).get(index: 3)) == result) {
                            writeFile.write("( " + num.get(i).get(index: 0) + " " + op[a] + " ( " + num.get(i).get(index: 1) + " "
                                    + op[b] + " " + num.get(i).get(index: 2) + " ) ) " + op[c] + " " + num.get(i).get(index: 3)
                                    + "\n");
                        }
                        if (fOp(op[b], fOp(op[a], num.get(i).get(index: 0), num.get(i).get(index: 1)),
                                fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3))) == result) {
                            writeFile.write("( " + num.get(i).get(index: 0) + " " + op[a] + " " + num.get(i).get(index: 1) + " ) "
                                    + op[b] + " ( " + num.get(i).get(index: 2) + " " + op[c] + " " + num.get(i).get(index: 3)
                                    + " )" + "\n");
```

Gambar 11 solution.java(5)

```java
                        }
                        if (fOp(op[a], num.get(i).get(index: 0), fOp(op[c],
                                fOp(op[b], num.get(i).get(index: 1), num.get(i).get(index: 2)), num.get(i).get(index: 3))) == result) {
                            writeFile.write(num.get(i).get(index: 0) + " " + op[a] + " ( ( " + num.get(i).get(index: 1) + " "
                                    + op[b] + " " + num.get(i).get(index: 2) + " ) " + op[c] + " " + num.get(i).get(index: 3)
                                    + " )" + "\n");

                        }
                        if (fOp(op[a], num.get(i).get(index: 0), fOp(op[b], num.get(i).get(index: 1),
                                fOp(op[c], num.get(i).get(index: 2), num.get(i).get(index: 3)))) == result) {
                            writeFile.write(num.get(i).get(index: 0) + " " + op[a] + " ( " + num.get(i).get(index: 1) + " "
                                    + op[b] + " ( " + num.get(i).get(index: 2) + " " + op[c] + " " + num.get(i).get(index: 3)
                                    + " ) )" + "\n");
                        }
                    }
                }
            }
        }
        writeFile.close();

    } catch (IOException e) {
        System.out.println(x: "An error occurred.");
        e.printStackTrace();
    }
}
```

Gambar 12 solution.java(6)

BAB 4
HASIL EKSEKUSI

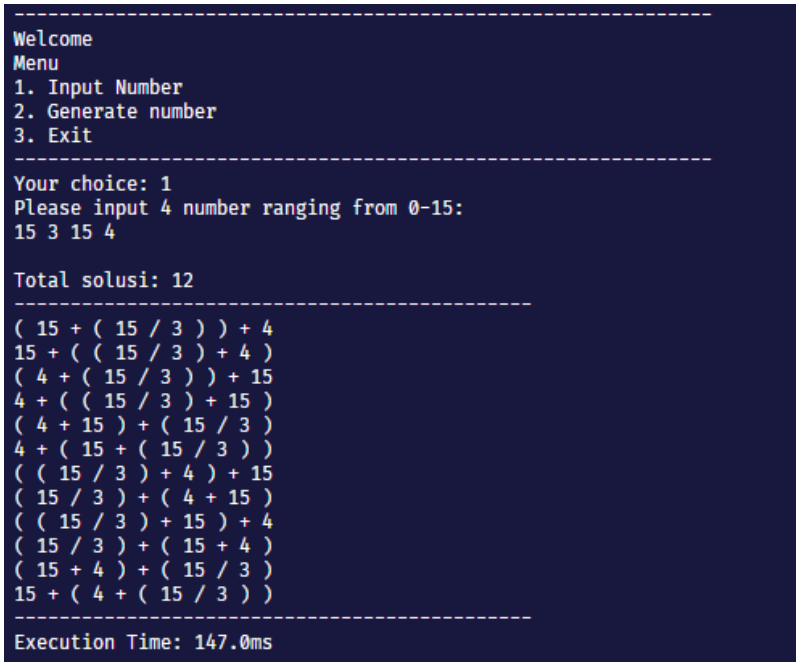
```
------------------------------------------------------------
Welcome
Menu
1. Input Number
2. Generate number
3. Exit
------------------------------------------------------------
Your choice: 1
Please input 4 number ranging from 0-15:
3 7 8 9

Total solusi: 16
------------------------------------------------
3 * ( ( 9 + 7 ) - 8 )
3 * ( 9 + ( 7 - 8 ) )
( ( 9 + 7 ) - 8 ) * 3
( 9 + ( 7 - 8 ) ) * 3
3 * ( ( 7 + 9 ) - 8 )
3 * ( 7 + ( 9 - 8 ) )
( ( 7 + 9 ) - 8 ) * 3
( 7 + ( 9 - 8 ) ) * 3
3 * ( ( 7 - 8 ) + 9 )
3 * ( 7 - ( 8 - 9 ) )
3 * ( ( 9 - 8 ) + 7 )
3 * ( 9 - ( 8 - 7 ) )
( ( 7 - 8 ) + 9 ) * 3
( 7 - ( 8 - 9 ) ) * 3
( ( 9 - 8 ) + 7 ) * 3
( 9 - ( 8 - 7 ) ) * 3
------------------------------------------------
Execution Time: 1474.0ms
```

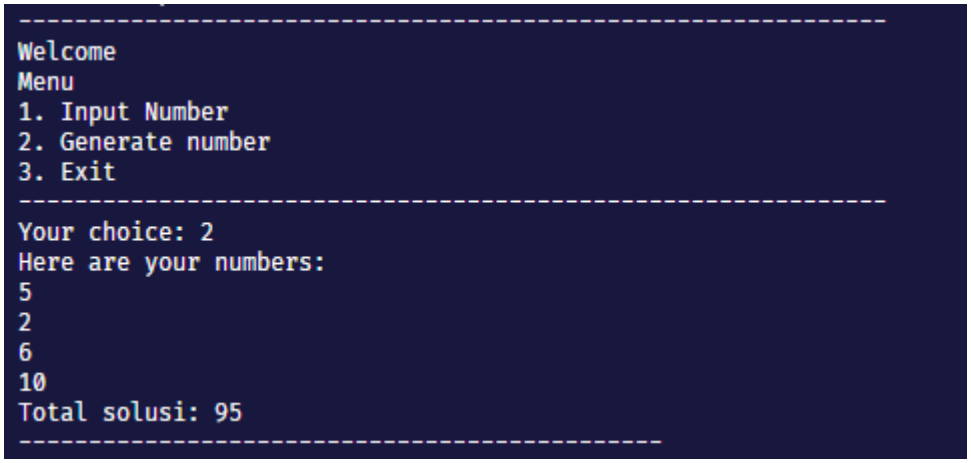
Gambar 13 tc1

```
------------------------------------------------------------
Welcome
Menu
1. Input Number
2. Generate number
3. Exit
------------------------------------------------------------
Your choice: 2
Here are your numbers:
14
1
10
13
Total solusi: 0
------------------------------------------------
------------------------------------------------
Execution Time: 182.0ms
```

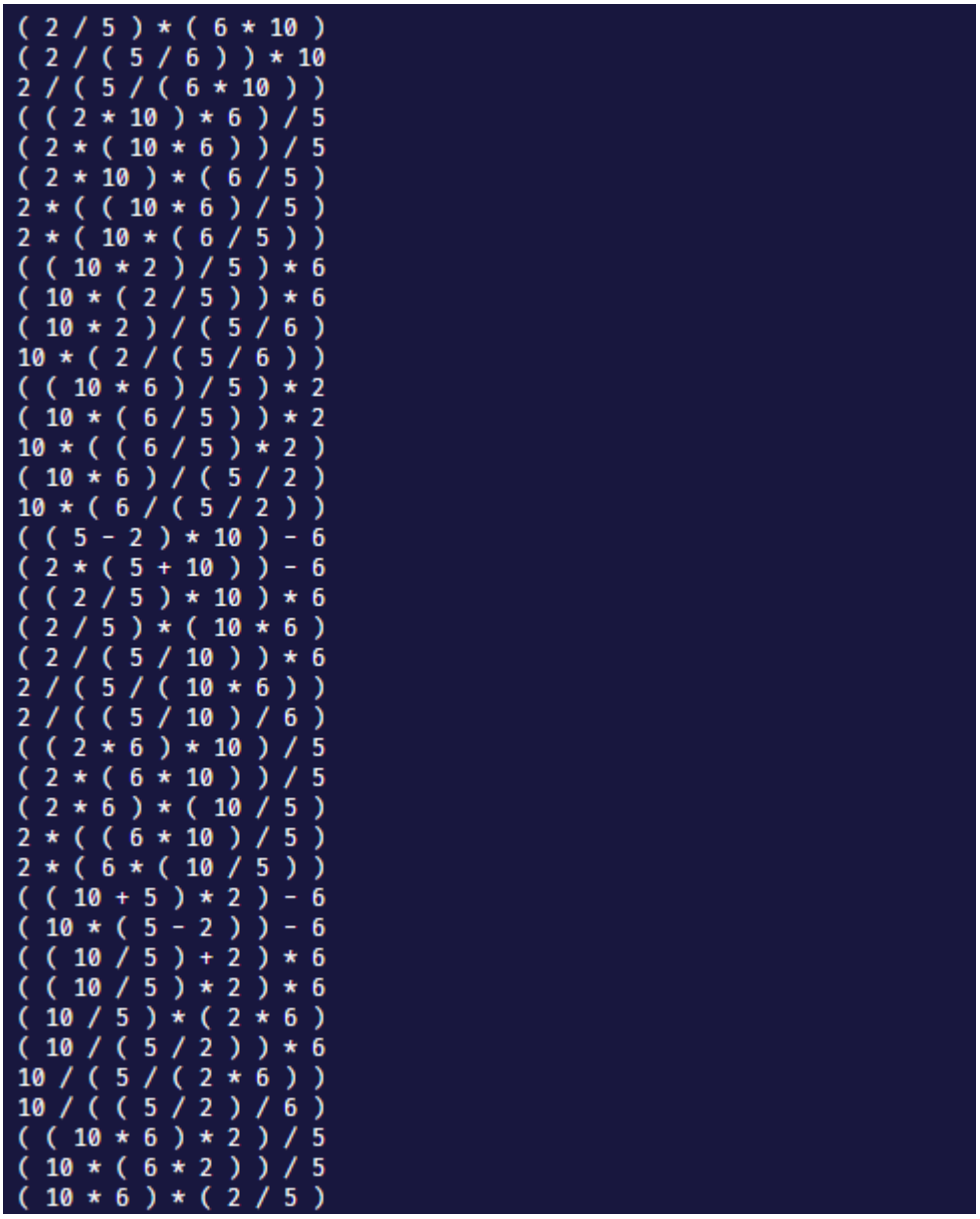
Gambar 14 tc2



Gambar 15 tc3



Gambar 16 tc4(1)



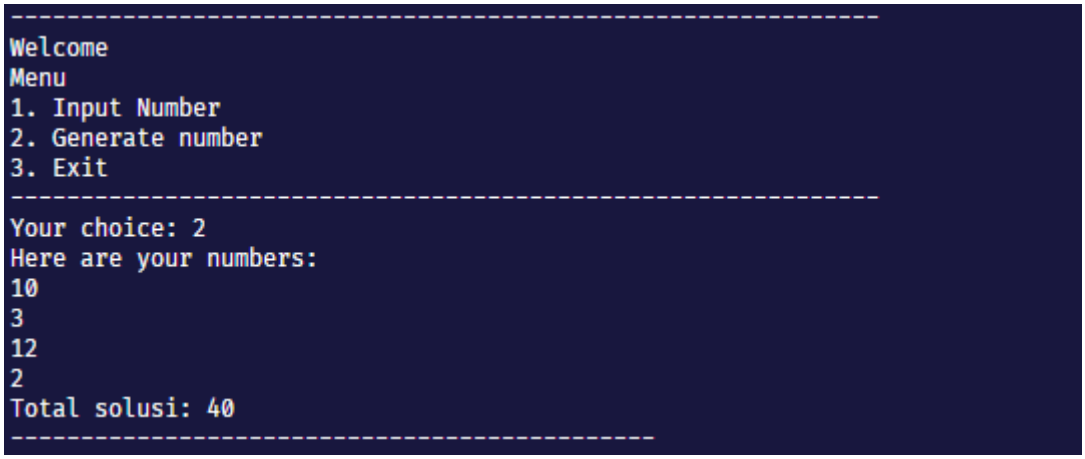
Gambar 17 tc4(2)

```
10 * ( ( 6 * 2 ) / 5 )
( ( 6 / 5 ) * 2 ) * 10
( 6 / 5 ) * ( 2 * 10 )
( 6 / ( 5 / 2 ) ) * 10
6 / ( 5 / ( 2 * 10 ) )
6 / ( ( 5 / 2 ) / 10 )
( ( 6 * 10 ) * 2 ) / 5
( 6 * ( 10 * 2 ) ) / 5
( 6 * 10 ) * ( 2 / 5 )
6 * ( ( 10 * 2 ) / 5 )
6 * ( 10 * ( 2 / 5 ) )
( ( 2 * 6 ) / 5 ) * 10
( 2 * ( 6 / 5 ) ) * 10
2 * ( ( 6 / 5 ) * 10 )
( 2 * 6 ) / ( 5 / 10 )
2 * ( 6 / ( 5 / 10 ) )
( 2 + ( 10 / 5 ) ) * 6
( 2 * ( 10 + 5 ) ) - 6
( ( 2 * 10 ) / 5 ) * 6
( 2 * ( 10 / 5 ) ) * 6
2 * ( ( 10 / 5 ) * 6 )
( 2 * 10 ) / ( 5 / 6 )
2 * ( 10 / ( 5 / 6 ) )
( ( 6 / 5 ) * 10 ) * 2
( 6 / 5 ) * ( 10 * 2 )
( 6 / ( 5 / 10 ) ) * 2
6 / ( 5 / ( 10 * 2 ) )
6 / ( ( 5 / 10 ) / 2 )
6 * ( 2 + ( 10 / 5 ) )
( ( 6 * 2 ) * 10 ) / 5
( 6 * ( 2 * 10 ) ) / 5
( 6 * 2 ) * ( 10 / 5 )
6 * ( ( 2 * 10 ) / 5 )
6 * ( 2 * ( 10 / 5 ) )
( ( 10 / 5 ) * 6 ) * 2
( 10 / 5 ) * ( 6 * 2 )
( 10 / ( 5 / 6 ) ) * 2
10 / ( 5 / ( 6 * 2 ) )
10 / ( ( 5 / 6 ) / 2 )
```

Gambar 18 tc4(3)

```
( ( 10 * 2 ) * 6 ) / 5
( 10 * ( 2 * 6 ) ) / 5
( 10 * 2 ) * ( 6 / 5 )
10 * ( ( 2 * 6 ) / 5 )
10 * ( 2 * ( 6 / 5 ) )
( ( 6 * 2 ) / 5 ) * 10
6 * ( ( 2 / 5 ) * 10 )
( 6 * 2 ) / ( 5 / 10 )
6 * ( 2 / ( 5 / 10 ) )
6 * ( ( 10 / 5 ) + 2 )
( ( 6 * 10 ) / 5 ) * 2
( 6 * ( 10 / 5 ) ) * 2
6 * ( ( 10 / 5 ) * 2 )
( 6 * 10 ) / ( 5 / 2 )
6 * ( 10 / ( 5 / 2 ) )
( ( 5 + 10 ) * 2 ) - 6
-----------------------------------------------
Execution Time: 1017.0ms
```

Gambar 19 tc4(4)



Gambar 20 tc5(1)

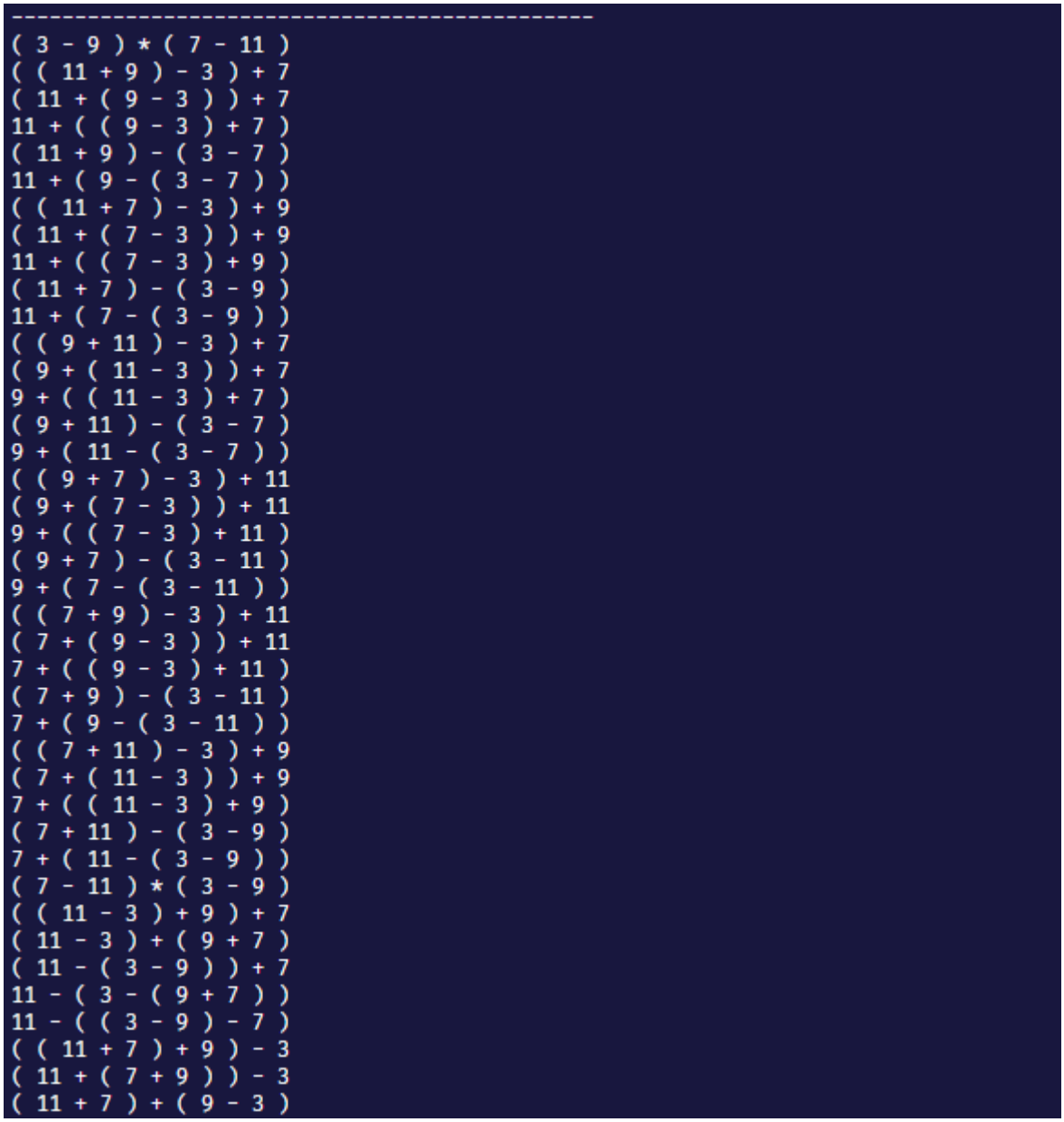
```
-------------------------------------------
( 12 * 10 ) / ( 3 + 2 )
12 * ( 10 / ( 3 + 2 ) )
( 12 / ( 2 + 3 ) ) * 10
12 / ( ( 2 + 3 ) / 10 )
( 2 * 10 ) + ( 12 / 3 )
( ( 2 * 10 ) - 12 ) * 3
( 3 * 10 ) - ( 12 / 2 )
( 12 * 10 ) / ( 2 + 3 )
12 * ( 10 / ( 2 + 3 ) )
12 * ( ( 10 / 2 ) - 3 )
( 12 * 3 ) - ( 2 + 10 )
( ( 12 * 3 ) - 2 ) - 10
( 12 / ( 3 + 2 ) ) * 10
( 12 / 3 ) + ( 2 * 10 )
12 / ( ( 3 + 2 ) / 10 )
( 10 / ( 3 + 2 ) ) * 12
10 / ( ( 3 + 2 ) / 12 )
( 10 * 12 ) / ( 2 + 3 )
10 * ( 12 / ( 2 + 3 ) )
( 10 * 3 ) - ( 12 / 2 )
( 10 * 2 ) + ( 12 / 3 )
( ( 10 * 2 ) - 12 ) * 3
( 12 * 3 ) - ( 10 + 2 )
( ( 12 * 3 ) - 10 ) - 2
( 12 / 3 ) + ( 10 * 2 )
( 3 * 12 ) - ( 10 + 2 )
( ( 3 * 12 ) - 10 ) - 2
( 3 * ( 2 + 10 ) ) - 12
3 * ( ( 2 * 10 ) - 12 )
( 10 * 12 ) / ( 3 + 2 )
10 * ( 12 / ( 3 + 2 ) )
( ( 10 + 2 ) * 3 ) - 12
( 10 / ( 2 + 3 ) ) * 12
10 / ( ( 2 + 3 ) / 12 )
( ( 10 / 2 ) - 3 ) * 12
( 3 * ( 10 + 2 ) ) - 12
3 * ( ( 10 * 2 ) - 12 )
( 3 * 12 ) - ( 2 + 10 )
( ( 3 * 12 ) - 2 ) - 10
( ( 2 + 10 ) * 3 ) - 12
-------------------------------------------
Execution Time: 652.0ms
```

Gambar 21 tc5(2)

```
-------------------------------------------------------------
Welcome
Menu
1. Input Number
2. Generate number
3. Exit
-------------------------------------------------------------
Your choice: 1
Please input 4 number ranging from 0-15:
9 3 11 7

Total solusi: 94
```

Gambar 22 tc6(1)



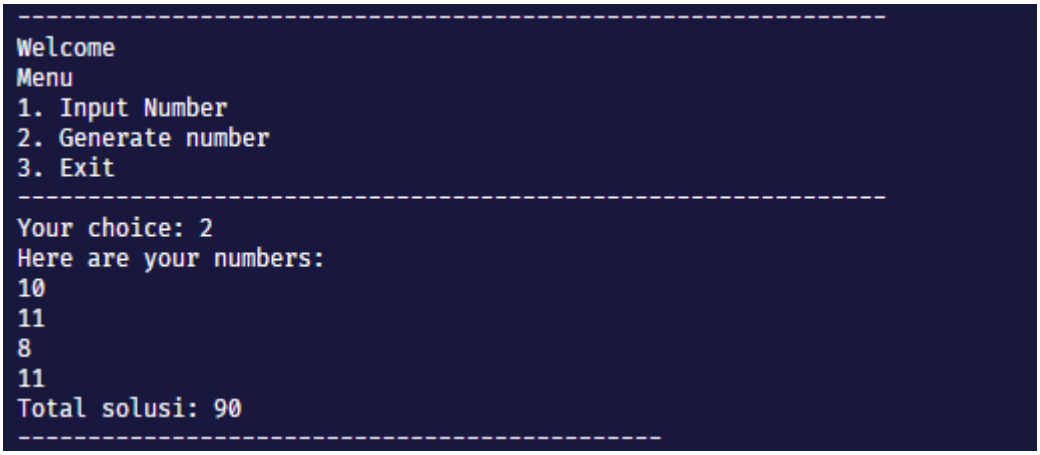
Gambar 23 tc6(2)

```
11 + ( 7 + ( 9 - 3 ) )
( 11 - 7 ) * ( 9 - 3 )
( ( 9 - 3 ) + 11 ) + 7
( 9 - 3 ) + ( 11 + 7 )
( 9 - ( 3 - 11 ) ) + 7
9 - ( 3 - ( 11 + 7 ) )
9 - ( ( 3 - 11 ) - 7 )
( 9 - 3 ) * ( 11 - 7 )
( ( 9 + 7 ) + 11 ) - 3
( 9 + ( 7 + 11 ) ) - 3
( 9 + 7 ) + ( 11 - 3 )
9 + ( ( 7 + 11 ) - 3 )
9 + ( 7 + ( 11 - 3 ) )
( ( 11 + 9 ) + 7 ) - 3
( 11 + ( 9 + 7 ) ) - 3
( 11 + 9 ) + ( 7 - 3 )
11 + ( ( 9 + 7 ) - 3 )
11 + ( 9 + ( 7 - 3 ) )
( ( 11 - 3 ) + 7 ) + 9
( 11 - 3 ) + ( 7 + 9 )
( 11 - ( 3 - 7 ) ) + 9
11 - ( 3 - ( 7 + 9 ) )
11 - ( ( 3 - 7 ) - 9 )
( ( 7 + 9 ) + 11 ) - 3
( 7 + ( 9 + 11 ) ) - 3
( 7 + 9 ) + ( 11 - 3 )
7 + ( ( 9 + 11 ) - 3 )
7 + ( 9 + ( 11 - 3 ) )
( ( 7 - 3 ) + 11 ) + 9
( 7 - 3 ) + ( 11 + 9 )
( 7 - ( 3 - 11 ) ) + 9
7 - ( 3 - ( 11 + 9 ) )
7 - ( ( 3 - 11 ) - 9 )
( ( 9 - 3 ) + 7 ) + 11
( 9 - 3 ) + ( 7 + 11 )
( 9 - ( 3 - 7 ) ) + 11
9 - ( 3 - ( 7 + 11 ) )
9 - ( ( 3 - 7 ) - 11 )
( ( 9 + 11 ) + 7 ) - 3
( 9 + ( 11 + 7 ) ) - 3
( 9 + 11 ) + ( 7 - 3 )
9 + ( ( 11 + 7 ) - 3 )
9 + ( 11 + ( 7 - 3 ) )
```

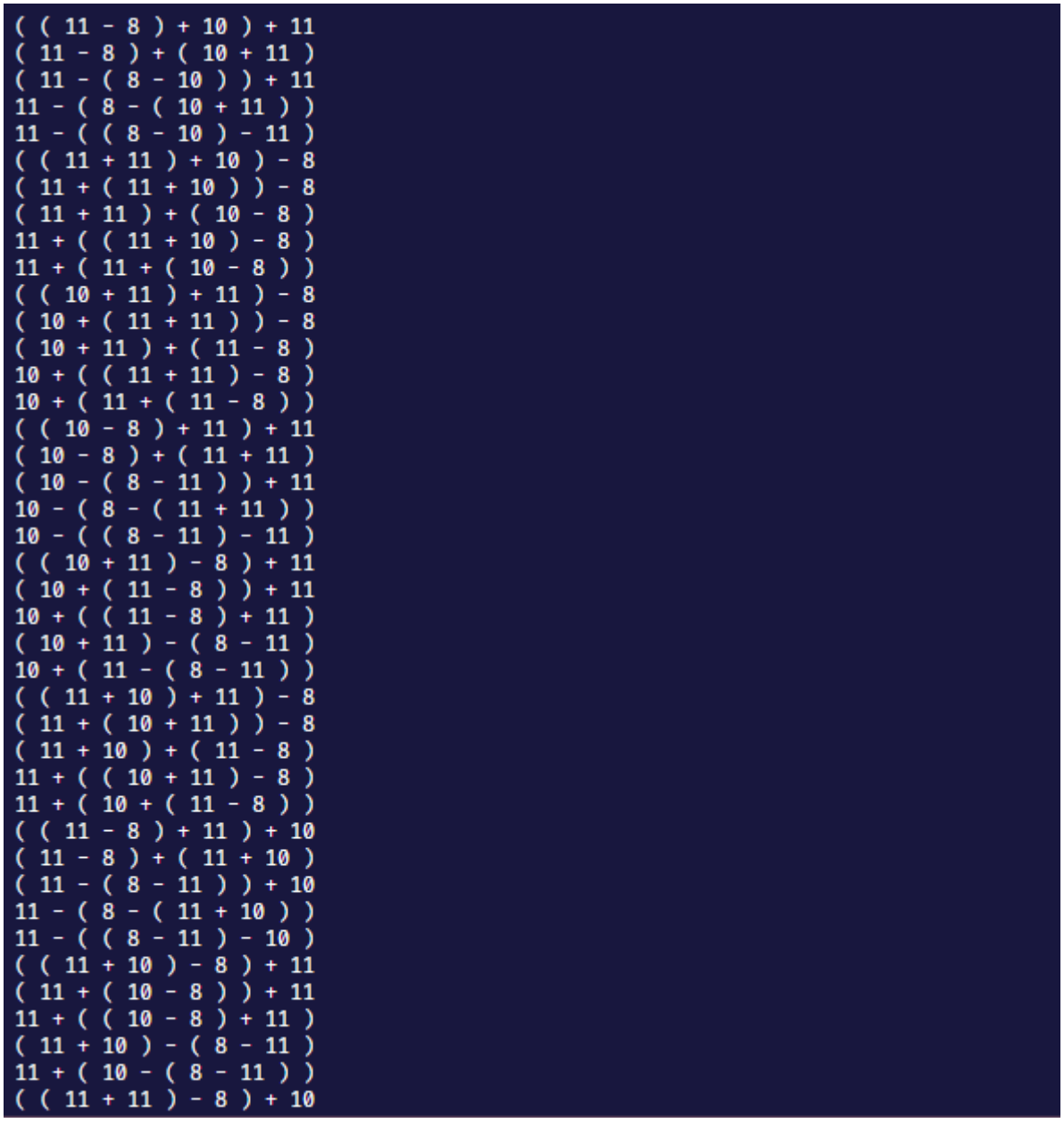
Gambar 24 tc6(3)

```
( ( 7 - 3 ) + 9 ) + 11
( 7 - 3 ) + ( 9 + 11 )
( 7 - ( 3 - 9 ) ) + 11
7 - ( 3 - ( 9 + 11 ) )
7 - ( ( 3 - 9 ) - 11 )
( ( 7 + 11 ) + 9 ) - 3
( 7 + ( 11 + 9 ) ) - 3
( 7 + 11 ) + ( 9 - 3 )
7 + ( ( 11 + 9 ) - 3 )
7 + ( 11 + ( 9 - 3 ) )
-----------------------------------------------
Execution Time: 739.0ms
```

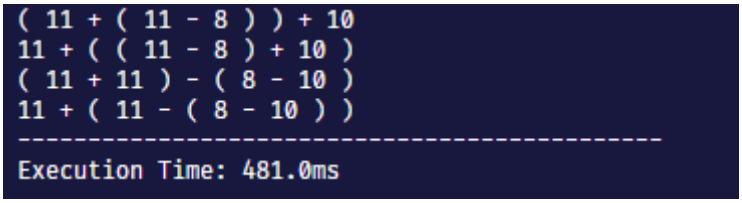
Gambar 25 tc6(4)



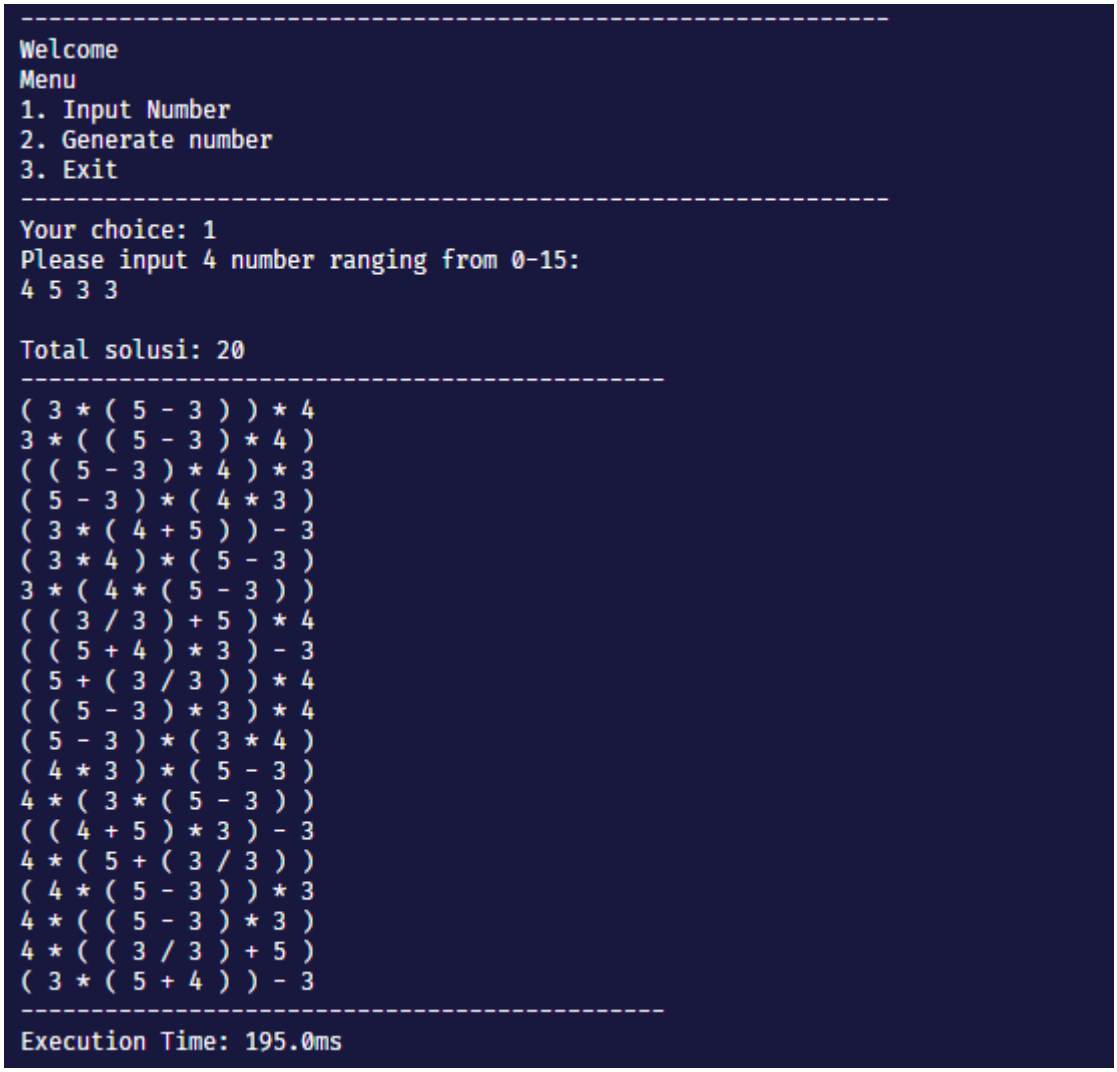
Gambar 26 tc7(1)



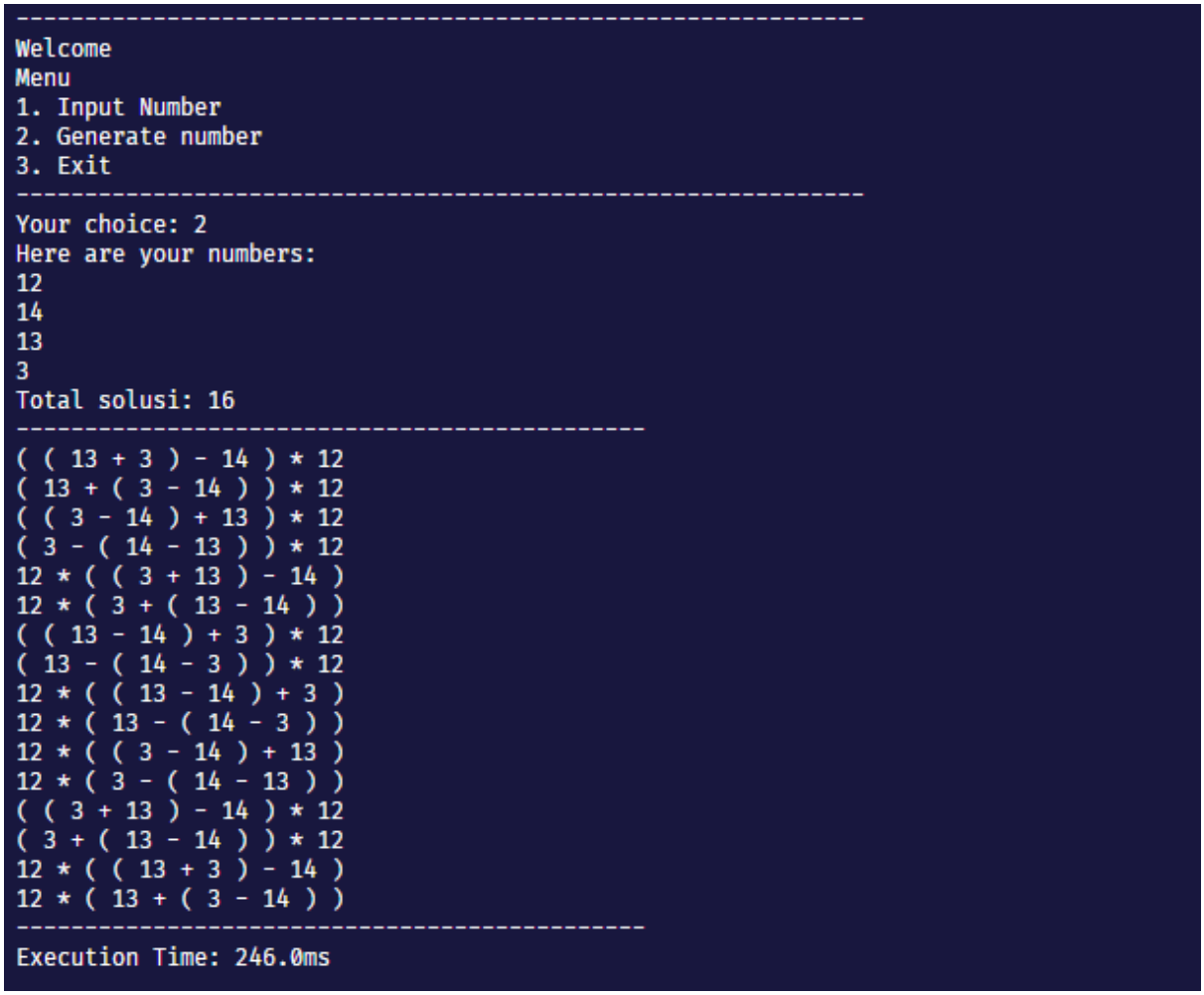
Gambar 27 tc7(2)



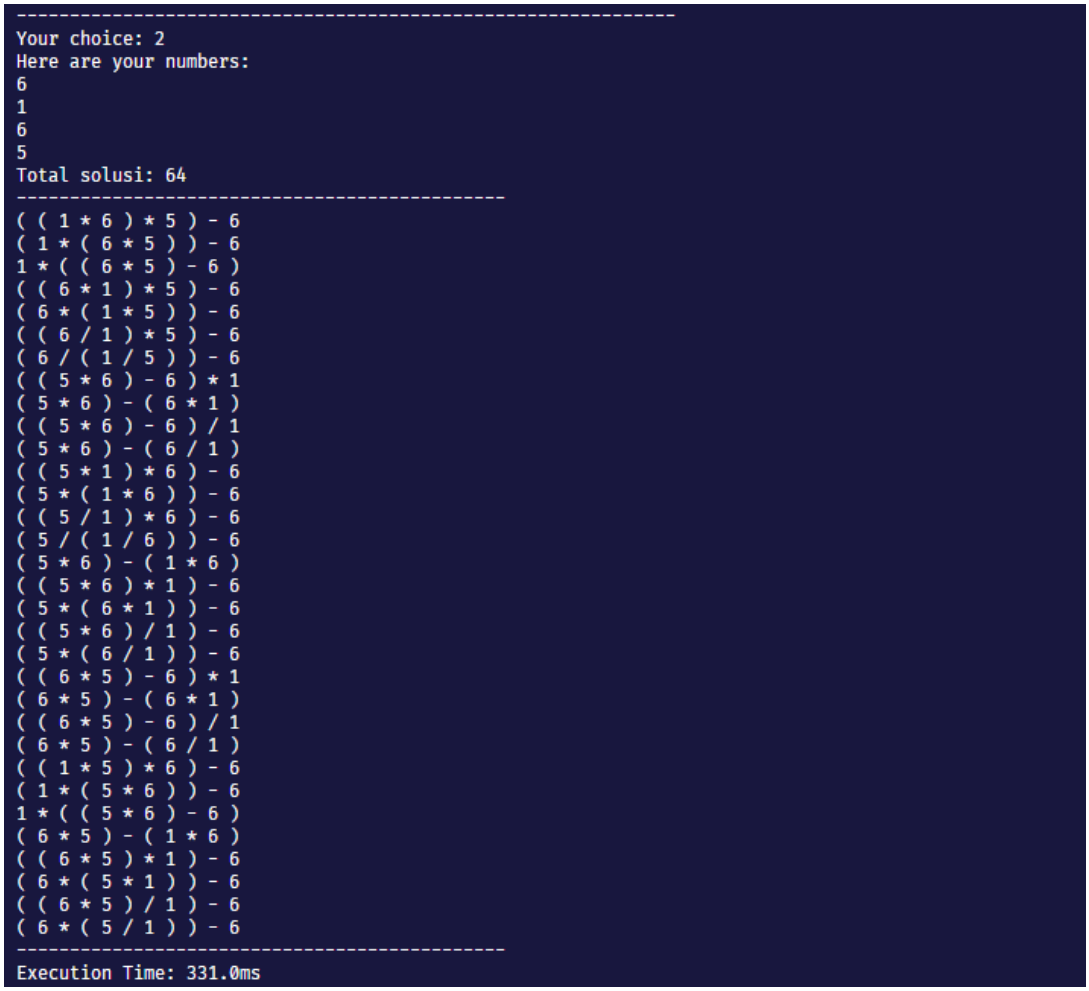
Gambar 28 tc7(3)



Gambar 29 tc8(1)



Gambar 30 tc9



Gambar 31 tc10

REFERENSI

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf

LAMPIRAN

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	✓	