

# Laporan Tugas Kecil IF-2211 2020/2021

## Cryptarithms

Tanur Rizaldi Rahardjo – 13519214 – K04

### 1. Algoritma

Untuk implementasi algoritma *brute force*, digunakan bahasa pemrograman C yang di-*compile* dan dijalankan dalam WSL ( *Windows subsystem for Linux* ). Sebagai pelengkap, digunakan *scripting* dalam *terminal bash* untuk mempermudah proses *recompile* dan uji kasus secara banyak.

*Library* yang digunakan dalam program adalah *stdlib.h*, *stdio.h*, *string.h*, dan *time.h*. *Library* *stdlib* dan *stdio* digunakan untuk keperluan *file processing* dan *I/O* program. *Library* *string* hanya digunakan untuk fungsi *strcpy* dan *library* *time* untuk kepentingan spesifikasi *runtime* program.

Program ini menerima argumen ketika dijalankan dan menggunakan argumen tersebut sebagai *file path* yang digunakan untuk uji kasus. Ketika *file* tersebut berhasil termuat, program akan memindahkan operan dan hasil penjumlahan dalam bentuk huruf ke *array of char* sebelum masuk algoritma *brute force*. Setelah validasi banyaknya operan, program akan membentuk *array of integer* bernama *alphaValue* dan *isFirstChar* berukuran 26 *integer* yang digunakan untuk menyimpan hubungan antara huruf dengan angka substitusi dan merupakan huruf pertama atau tidak.

Algoritma *brute force* program akan memulai pengecekan dengan substitusi 0 0 0 0 ... untuk semua huruf dan melakukan pengecekan apakah hasil operan dan hasil penjumlahan sama. Jika hasil sama, maka dilakukan pengecekan kevalidan angka unik pada jawaban. Pengecekan tersebut akan menghentikan program ketika solusi sudah didapatkan dan akan membuat solusi baru ketika pengecekan gagal.

Algoritma yang digunakan program ini merupakan algoritma yang sangat sederhana, sehingga sangat tidak teroptimisasi untuk kasus huruf yang banyak. Dapat dilihat secara sekilas bahwa program ini akan menguji seluruh kasus  $10^n$  dengan  $n$  adalah banyaknya huruf unik. Dibandingkan dengan program yang hanya menguji semua permutasi valid hanya  $n!$  kasus. Namun dikarenakan kasus yang hanya dibatasi dengan 10 karakter unik dan program ditulis dalam bahasa C, dipertimbangkan bahwa implementasi algoritma naif tersebut sangat mengurangi waktu dalam pembuatannya.

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat membaca file masukan dan menuliskan luaran.	✓	
4. Solusi <i>cryptarithmic</i> hanya benar untuk persoalan <i>cryptarithmic</i> dengan dua buah <i>operand</i> .	✓	
5. Solusi <i>cryptarithmic</i> benar untuk persoalan <i>cryptarithmic</i> untuk lebih dari dua buah <i>operand</i> .	✓	

## 2. Source Code

*Screenshot* dibawah merupakan *screenshot* pada algoritma *brute force* saja, tanpa *file handling* dan *argument checking*.

```
// Main Algorithm~
if (operandCount > 1) {~
    ....// Strip plus sign~
    ....operand[operandCount - 1][stringLength(operand[operandCount - 1]) - 1] = '\0';~

    ....// Array Initialize~
    ....int alphaValue[26], tempOperandSum, sumValue, isFirstChar[26];~
    ....for (int i = 0; i < 26; i++) {~
        ....alphaValue[i] = -1;~
        ....isFirstChar[i] = 0;~
    ....}~

    ....// Finding used alphabet~
    ....for (int i = 0; i < operandCount; i++) {~
        ....int j = 0;~
        ....while (operand[i][j] != '\0' && operand[i][j] != '\r') {~
            ....int tempAlphaIndex = ((int) operand[i][j]) - 65;~
            ....alphaValue[tempAlphaIndex] = 0;~
            ....if (j == 0)~
            ....alphaValue[tempAlphaIndex] = isFirstChar[tempAlphaIndex] = 1;~
            ....j++;~
        ....}~
    ....}~

    int tempIdx = 0;~
    while (sum[tempIdx] != '\0' && sum[tempIdx] != '\r') {~
        ....int tempAlphaIndex = ((int) sum[tempIdx]) - 65;~
        ....alphaValue[tempAlphaIndex] = 0;~
        ....if (tempIdx == 0)~
        ....alphaValue[tempAlphaIndex] = isFirstChar[tempAlphaIndex] = 1;~
        ....tempIdx++;~
    }~

    // Brute forcing~
    int testCount = 0, isValidCombination = 0;~
    puts("Searching ...");~
    clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &start);~
```

```

// int operandVal[16]; // DEBUG
do {
    sumValue = tempOperandSum = 0;

    // Temporary operand sum
    for (int i = 0; i < operandCount; i++) {
        int j = 0, operandValue = 0;
        while (operand[i][j] != '\0' && operand[i][j] != '\r') {
            operandValue = 10*operandValue + alphaValue[((int) operand[i][j]) - 65];
            j++;
        }
        // operandVal[i] = operandValue; // DEBUG
        tempOperandSum += operandValue;
    }

    // Temporary "Result" sum
    int a = 0;
    while (sum[a] != '\0' && sum[a] != '\r') {
        sumValue = 10*sumValue + alphaValue[((int) sum[a]) - 65];
        a++;
    }
    // printf("%d\n%d\n", sumValue, tempOperandSum); // DEBUG
}

```

```

// Uniqueness test
int isUniqueCombination = 1;
for (int i = 0; i < 26 && isUniqueCombination; i++) {
    for (int j = i + 1; j < 26; j++) {
        if ((alphaValue[i] >= 0) && (alphaValue[i] == alphaValue[j])) {
            isUniqueCombination = 0;
            break;
        }
    }
}

// Solution check and solution set generator
if (isUniqueCombination && (tempOperandSum == sumValue))
    isValidCombination = 1;
else {
    for (int i = 0; i < 26; i++) {
        if (9 > alphaValue[i] && alphaValue[i] >= 0) {
            alphaValue[i]++;
            break;
        }
        else if (alphaValue[i] == 9) {
            if (isFirstChar[i])
                alphaValue[i] = 1;
            else
                alphaValue[i] = 0;
        }
    }
    testCount++;
} while (!isValidCombination);

```

### 3. Uji Kasus

```
brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanun Rizal/Ubuntu
SEND
MORE+
-----
MONEY

Searching ...
Time elapsed : 12.450609
Total test count : 24125458

Result
9567
1085+
-----
10652

D = 7
E = 5
M = 1
N = 6
O = 0
R = 8
S = 9
Y = 2
```

```
brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanun Rizal/Ubuntu
NUMBER
NUMBER+
-----
PUZZLE

Searching ...
Time elapsed : 42.255771
Total test count : 250571787

Result
201689
201689+
-----
403378

B = 6
E = 8
L = 7
M = 1
N = 2
P = 4
R = 9
U = 0
Z = 3
```

```
brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanun Rizal/Ubuntu
NO
GUN
NO+
-----
HUNT

Searching ...
Time elapsed : 0.009498
Total test count : 20340

Result
87
908
87+
-----
1082

G = 9
H = 1
N = 8
O = 7
T = 2
U = 0
```

```
^Cbrsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanun Rizal/Ubuntu/tcl
FORTY
TEN
TEN+
-----
SIXTY

Searching ...
Time elapsed : 730.198161
Total test count : 437943810

Result
29786
850
850+
-----
31486

E = 5
F = 2
I = 1
N = 0
O = 9
R = 7
S = 3
T = 8
X = 4
Y = 6
```

```

brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanur_Rizal/
MEMO
FROM+
-----
HOMER

Searching ...
Time elapsed : 0.071715
Total test count : 261695

Result
8485
7358+
-----
15843

E = 4
F = 7
H = 1
M = 8
O = 5
R = 3

brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanur_Rizal/Ubuntu/
HERE
SHE+
-----
COMES

Searching ...
Time elapsed : 1.318736
Total test count : 6888277

Result
9454
894+
-----
10348

C = 1
E = 4
H = 9
M = 3
O = 0
R = 5
S = 8

brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanur_Rizal/
COCA
COLA+
-----
OASIS

Searching ...
Time elapsed : 0.035378
Total test count : 162887

Result
8186
8106+
-----
16292

A = 6
C = 8
I = 9
L = 0
O = 1
S = 2

brsh@DESKTOP-LSL5JSB:/mnt/c/Users/Tanur_Rizal/Ubuntu/
JUNE
JULY+
-----
APRIL

Searching ...
Time elapsed : 191.759897
Total test count : 2022066046

Result
8435
8472+
-----
16907

A = 1
E = 5
I = 0
J = 8
L = 7
N = 3
P = 6
R = 9
U = 4
Y = 2

```

#### 4. Link Source Code

[Github Repository](#)