

Laporan Tugas
Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

Jusuf Junior Athala – 13519174

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

A. Langkah-langkah Menyelesaikan Cryptarithmic dengan Algoritma Brute Force

1. Pisahkan kata-kata operand dengan hasil penjumlahan dari kata-kata operand.
2. Buat list huruf-huruf yang muncul dan huruf-huruf yang merupakan huruf awal dari kata.
3. Jika huruf yang muncul kurang dari atau sama dengan 10 maka akan dimulai permutasi dari angka 0-9 dan menghubungkan angka-angka tersebut sebagai nilai dari setiap huruf-huruf yang muncul.
4. Jika terdapat huruf awal dari kata mendapatkan nilai 0 maka permutasi tersebut akan diabaikan.
5. Setiap kata akan mendapatkan nilai masing-masing berdasarkan nilai dari list huruf yang muncul, huruf akhir dari sebuah kata akan dikalikan dengan 10^0 , huruf ke-2 dari akhir akan dikalikan dengan 10^1 , dan seterusnya hingga huruf awal dari kata.
6. Jika terdapat jumlah dari operand sama dengan hasil penjumlahan dari operand, maka telah didapatkan solusi nya.

B. Source Code Program

```
# Nama      : Jusuf Junior Athala
# NIM       : 13519174
# Mata kuliah : IF2211 - Strategi Algoritma
# Kelas     : K-04
# Tugas     : Penyelesaian Cryptarithmic dengan Algoritma Brute Force
# Nama file  : 13519174_Tucil01.py

import itertools
import time

#Untuk mendapatkan value dari kata
def get_value(word, solution_mapped):
    value = 0
    factor = 1
    for letter in reversed(word):
        value += factor * solution_mapped[letter]
        factor *= 10
    return value

#Untuk menentukan huruf awal yang terdapat pada mapped solution
def lead_is_zero(solution_mapped, leads):
    for lead in leads:
        if solution_mapped.get(lead)==0 :
            return True
```

```

def crypta_permutation():
# Membuka dan membaca file soal.txt
    print("Program membaca file soal.txt")
    f=open("soal.txt", "r")
    contents =f.read()

# Memisahkan operand dan hasil
    operand, result = contents.lower().replace(' ', '').replace('-',
''.split('+\\n')
    result = result.replace ('\\n','')
# Memisahkan semua operand
    operand = operand.split('\\n')

# Membuat list untuk huruf-huruf yang muncul dan list untuk huruf yang
merupakan huruf awal dari kata
    letters = set(result)
    leads = set(result[0])
    for word in operand:
        leads.add(word[0])
        for letter in word:
            letters.add(letter)
    letters = list(letters)
    leads = list(leads)

    if len(leads)<11 :
        count = 0
        count_result = 0
        timer_start = time.perf_counter()
        print ("Program dimulai")
        print ()
        print (contents)
# Memulai permutasi dan melakukan mapping terhadap list huruf yang
muncul
        for perm in itertools.permutations(range(10), len(letters)):
            solution_mapped = dict(zip(letters, perm))
            if lead_is_zero(solution_mapped,leads) :          #
Melakukan continue dari iterasi jika terdapat huruf awal dengan value 0
                continue
            count = count+1
            if sum(get_value(word, solution_mapped) for word in
operand) == get_value(result, solution_mapped):
                timer_stop = time.perf_counter()
                count_result = count_result +1
                print()
                print('\\n'.join(str(get_value(word,
solution_mapped)).rjust(len(result)) for word in operand) + '\\n'+'-
'*len(result)+'\\n' + str(get_value(result, solution_mapped)))#" = {} (mapping:
{})).format(get_value(result, solution_mapped), solution_mapped))
                print("Hasil ditemukan, waktu yang diperlukan
untuk menemukan solusi ke-",count_result,"adalah : ", "%.4f" %(timer_stop-
timer_start)," detik")
                print("Jumlah tes substitusi yang dilakukan
untuk menemukan solusi ke-",count_result,"adalah : ", count," kali")
                print()
                print ("Program selesai, total waktu yang diperlukan:", "%.4f"
%(time.perf_counter()-timer_start)," detik")

```

```

        print ("Jumlah total tes yang dilakukan : ", count," kali")
        print ("Jumlah solusi yang ditemukan : ", count_result,"
solusi")

        f.close()

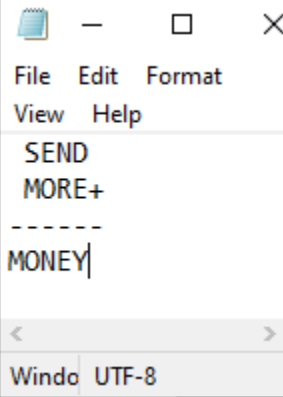
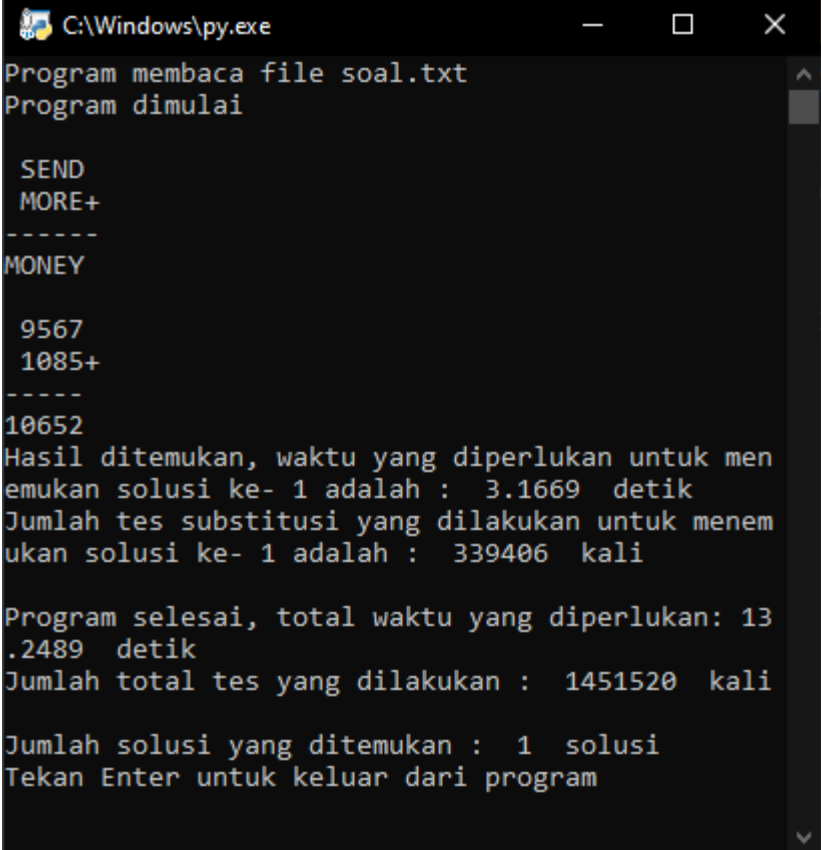
    else :

        "Huruf yang muncul lebih dari 10"

if __name__ == '__main__':
    crypta_permutation()
    selesai = input('Tekan Enter untuk keluar dari program')

```

C. Contoh Eksekusi Program (Input dan Output)

No	Input	Output
1		

2

soal.txt - Notepad

File Edit Format Vi

```
CLOCK
TICK
TOCK+
-----
PLANET
```

C:\Windows\py.exe

```
Program membaca file soal.txt
Program dimulai
```

```
CLOCK
TICK
TOCK+
-----
PLANET
```

```
90892
6592
6892+
```

```
-----
104376
```

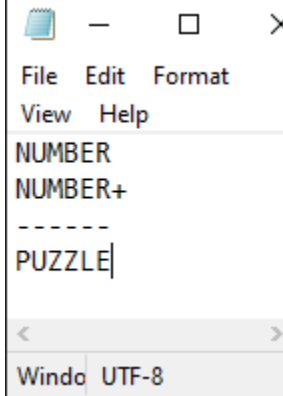
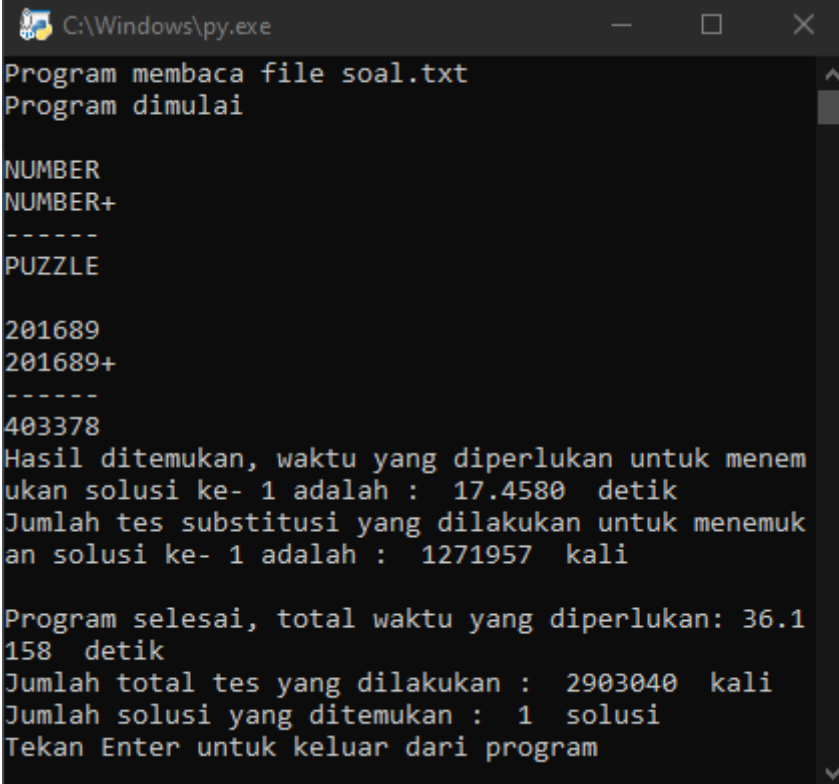
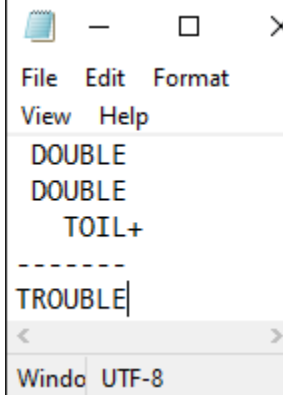
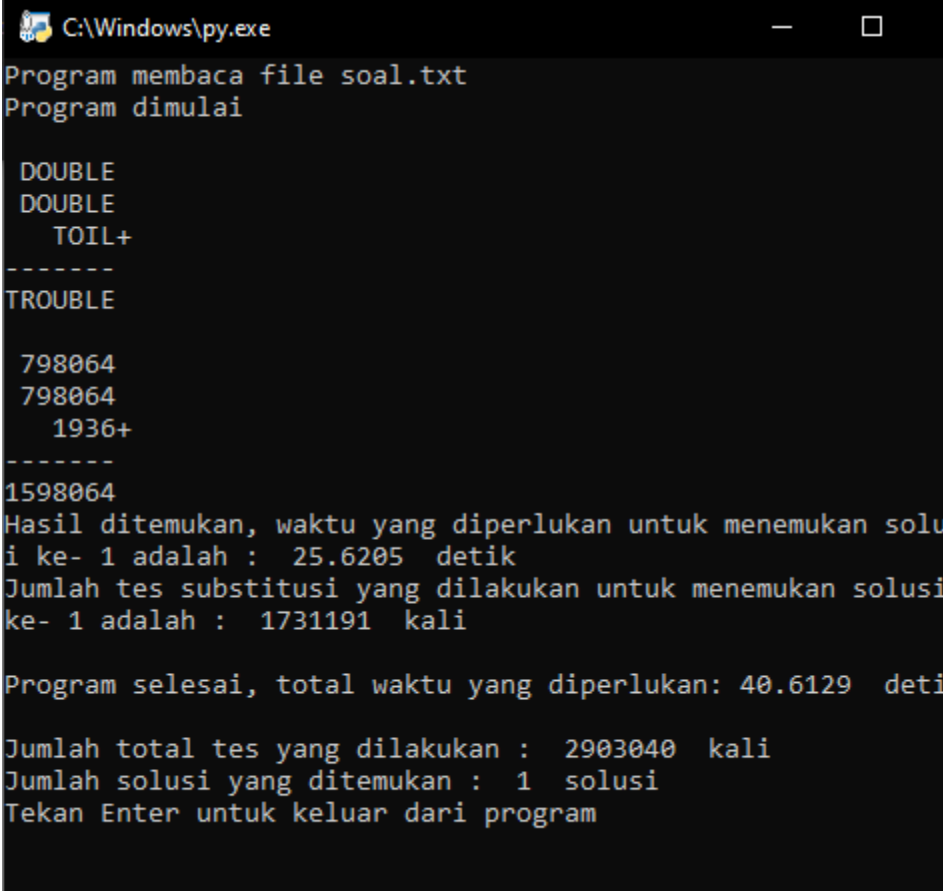
```
Hasil ditemukan, waktu yang diperlukan untuk
menemukan solusi ke- 1 adalah : 3.6124 de
tik
```

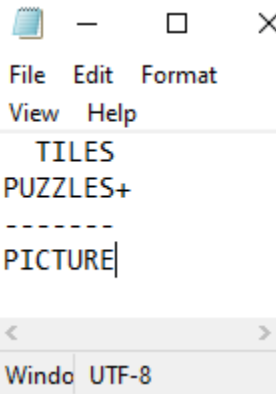
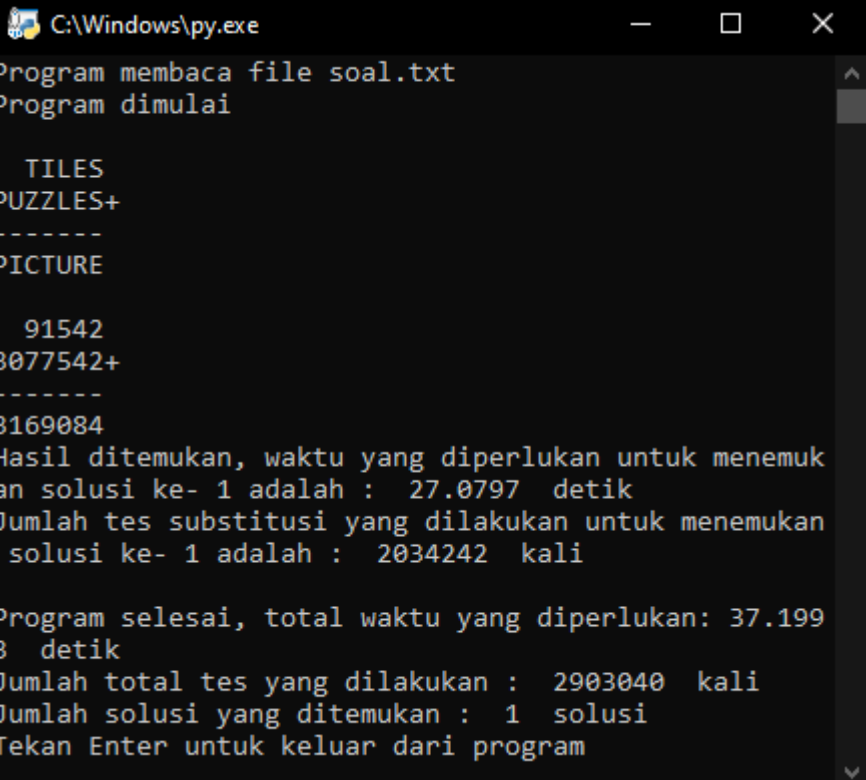
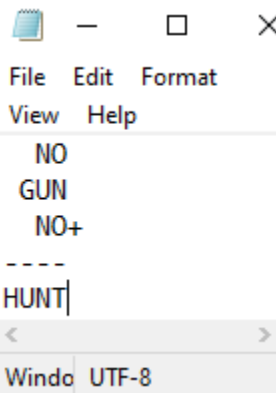
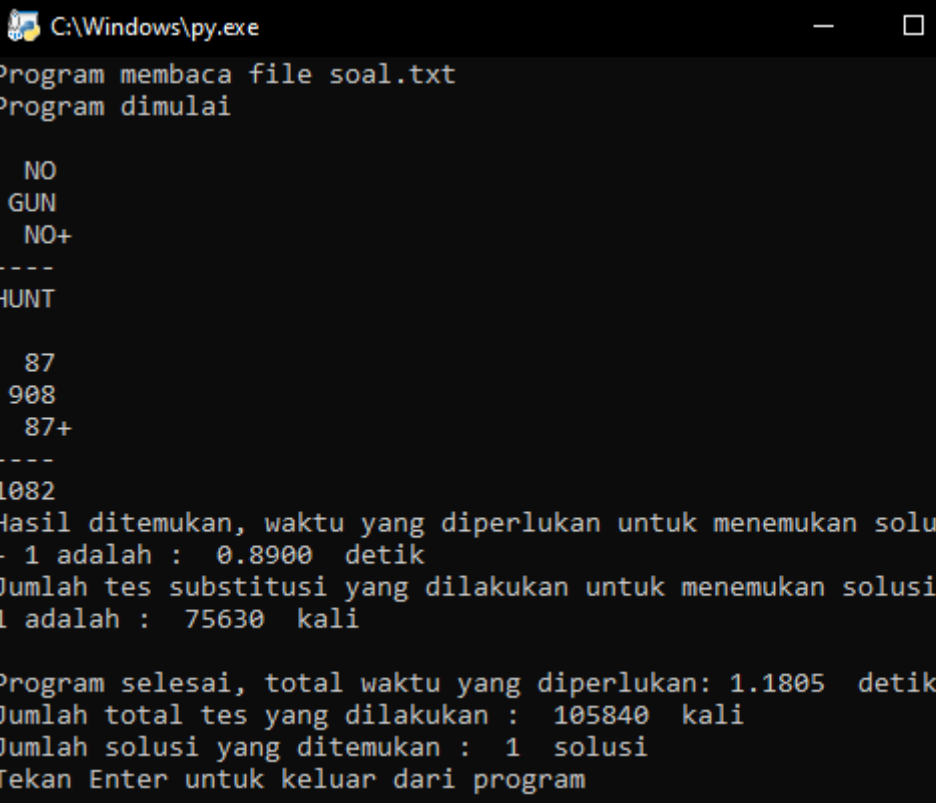
```
Jumlah tes substitusi yang dilakukan untuk m
enemukan solusi ke- 1 adalah : 304226 kali
```

```
Program selesai, total waktu yang diperlukan
: 34.2646 detik
```

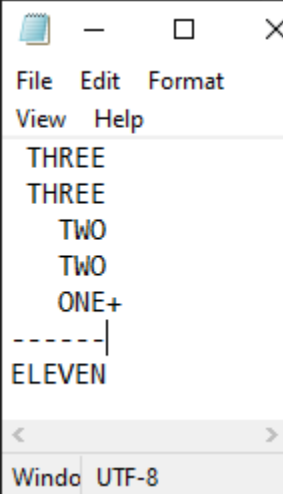
```
Jumlah total tes yang dilakukan : 2540160
kali
```

```
Jumlah solusi yang ditemukan : 1 solusi
Tekan Enter untuk keluar dari program
```

3	 <pre> File Edit Format View Help NUMBER NUMBER+ ----- PUZZLE < > Windo UTF-8 </pre>	 <pre> C:\Windows\py.exe Program membaca file soal.txt Program dimulai NUMBER NUMBER+ ----- PUZZLE 201689 201689+ ----- 403378 Hasil ditemukan, waktu yang diperlukan untuk menemukan solusi ke- 1 adalah : 17.4580 detik Jumlah tes substitusi yang dilakukan untuk menemukan solusi ke- 1 adalah : 1271957 kali Program selesai, total waktu yang diperlukan: 36.1158 detik Jumlah total tes yang dilakukan : 2903040 kali Jumlah solusi yang ditemukan : 1 solusi Tekan Enter untuk keluar dari program </pre>
4	 <pre> File Edit Format View Help DOUBLE DOUBLE TOIL+ ----- TROUBLE < > Windo UTF-8 </pre>	 <pre> C:\Windows\py.exe Program membaca file soal.txt Program dimulai DOUBLE DOUBLE TOIL+ ----- TROUBLE 798064 798064 1936+ ----- 1598064 Hasil ditemukan, waktu yang diperlukan untuk menemukan solusi ke- 1 adalah : 25.6205 detik Jumlah tes substitusi yang dilakukan untuk menemukan solusi ke- 1 adalah : 1731191 kali Program selesai, total waktu yang diperlukan: 40.6129 detik Jumlah total tes yang dilakukan : 2903040 kali Jumlah solusi yang ditemukan : 1 solusi Tekan Enter untuk keluar dari program </pre>

5		
6		

7

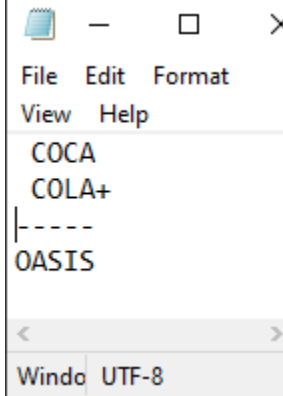
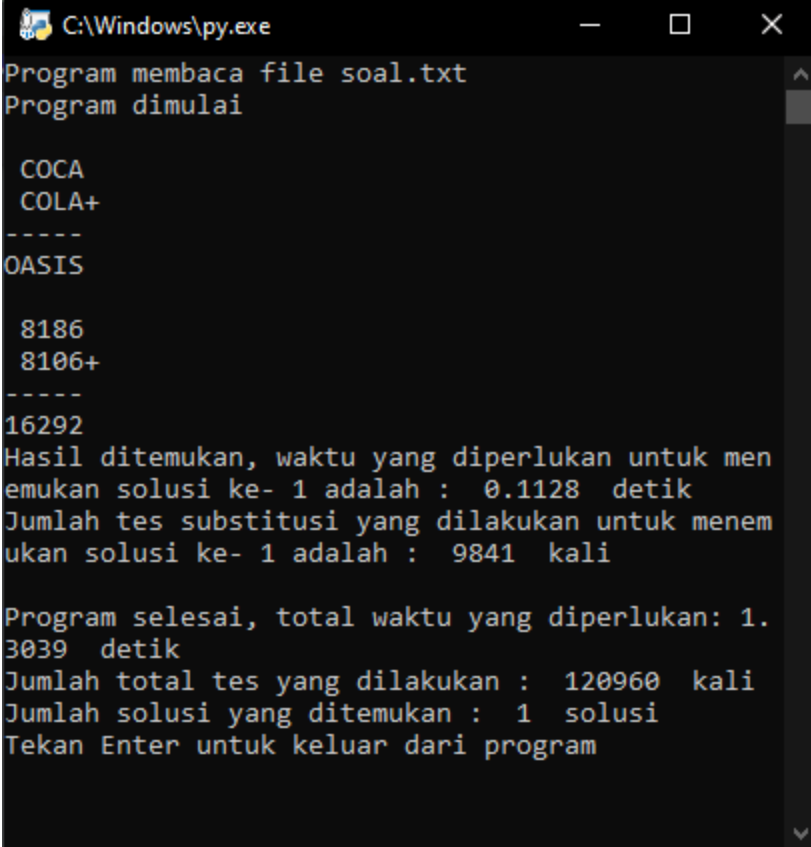
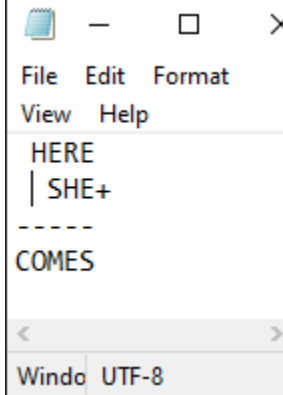
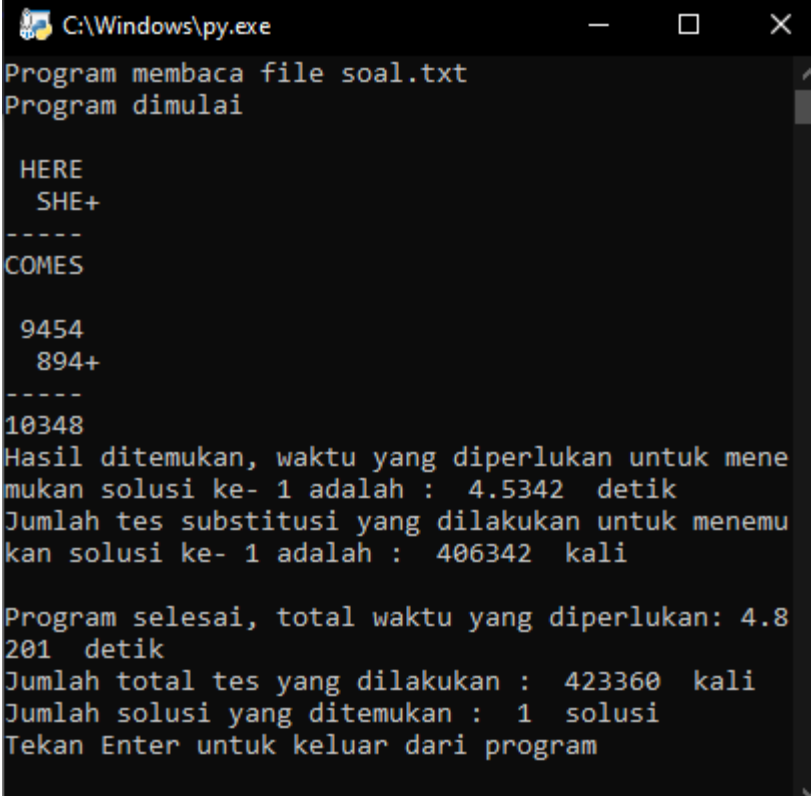


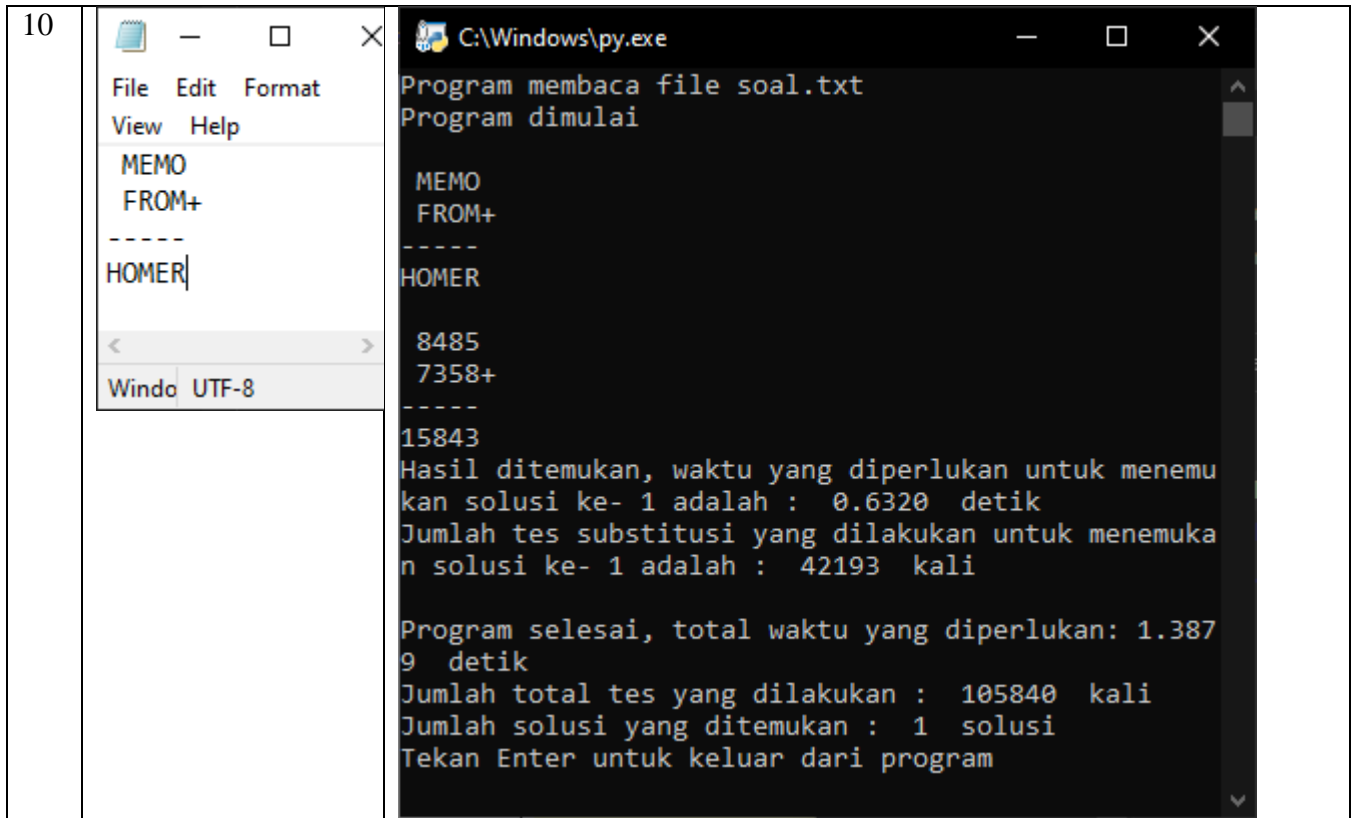
```
C:\Windows\py.exe
Program membaca file soal.txt
Program dimulai

THREE
THREE
TWO
TWO
ONE+
-----
ELEVEN

84611
84611
803
803
391+
-----
171219
Hasil ditemukan, waktu yang diperlukan untuk menemukan
solusi ke- 1 adalah : 35.3881 detik
Jumlah tes substitusi yang dilakukan untuk menemukan
solusi ke- 1 adalah : 2246360 kali

Program selesai, total waktu yang diperlukan: 40.215
2 detik
Jumlah total tes yang dilakukan : 2540160 kali
Jumlah solusi yang ditemukan : 1 solusi
Tekan Enter untuk keluar dari program
```

8		 <pre> C:\Windows\py.exe Program membaca file soal.txt Program dimulai COCA COLA+ ----- OASIS 8186 8106+ ----- 16292 Hasil ditemukan, waktu yang diperlukan untuk men emukan solusi ke- 1 adalah : 0.1128 detik Jumlah tes substitusi yang dilakukan untuk menemu ukan solusi ke- 1 adalah : 9841 kali Program selesai, total waktu yang diperlukan: 1. 3039 detik Jumlah total tes yang dilakukan : 120960 kali Jumlah solusi yang ditemukan : 1 solusi Tekan Enter untuk keluar dari program </pre>
9		 <pre> C:\Windows\py.exe Program membaca file soal.txt Program dimulai HERE SHE+ ----- COMES 9454 894+ ----- 10348 Hasil ditemukan, waktu yang diperlukan untuk mene mukan solusi ke- 1 adalah : 4.5342 detik Jumlah tes substitusi yang dilakukan untuk menemu kan solusi ke- 1 adalah : 406342 kali Program selesai, total waktu yang diperlukan: 4.8 201 detik Jumlah total tes yang dilakukan : 423360 kali Jumlah solusi yang ditemukan : 1 solusi Tekan Enter untuk keluar dari program </pre>



D. Checklist

Berikut adalah link untuk file program :

<https://github.com/jusufjathala/Cryptarithmetic>

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