

**LAPORAN TUGAS KECIL 1**  
**IF2211 STRATEGI ALGORITMA**  
**SEMESTER II TAHUN 2020/2021**

**PENYELESAIAN CRYPTARITHMETIC DENGAN**  
**ALGORITMA BRUTE FORCE**



Disusun oleh:

Melita

13519063

K-02

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2021**

## BAGIAN I: ALROTIMA BRUTE FORCE

Langkah-langkah pada algoritma yang digunakan adalah sebagai berikut.

1. Baca *input* nama *file* soal *cryptarithmic*.
2. Buka *file* tersebut, lalu simpan isinya ke dalam sebuah *array*.
3. Mulai hitung waktu eksekusi program.
4. Cetak soal *cryptarithmic* hasil masukan tadi.
5. Pisahkan semua *operand* dengan hasil penjumlahan.
6. Cek jumlah huruf unik pada soal, jika lebih dari 10, soal tidak bisa dihitung.
7. Jika huruf unik kurang dari sama dengan 10, maka mulai perhitungan sebagai berikut.
  - a. Inisialisasi nilai kombinasi pertama. Karena daftar nilai huruf dimulai dari hasil, nilai huruf pertama tidak boleh nol. Jadi, nilai diinisialisasi sebagai 10234... sesuai jumlah huruf unik pada soal.
  - b. Lakukan pengecekan pada daftar nilai huruf yang dibuat. Jika ada huruf awal kata yang bernilai nol, maka lanjut ke kombinasi percobaan berikutnya.
  - c. Jika tidak ada huruf awal kata yang bernilai nol, hitung hasil penjumlahan semua *operand* dengan nilai percobaan. Tahap ini dihitung sebagai satu kali hasil tes. Bandingkan hasilnya dengan nilai hasil pada soal. Jika sama, maka jawaban ditemukan. Jika tidak, coba kombinasi berikutnya\*, lalu ulangi langkah b dan c.
8. Jika jawaban ditemukan, cetak hasil jawaban dalam bentuk penjumlahan seperti soal beserta waktu eksekusi program dan total tes yang dilakukan. Jika jawaban tidak ditemukan, cetak tulisan "Tidak ada solusi yang ditemukan."

\*Kombinasi berikutnya dicari dengan menambahkan 1 pada kombinasi sebelumnya, lalu mengecek apakah semua angkanya unik atau tidak. Jika tidak, tambahkan 1 terus sampai ditemukan kombinasi angka yang unik. Contoh: Kata "MONEY", jika kombinasi nilai sebelumnya 10234 (M = 1, O = 0, N = 2, E = 3, Y = 4), nilai setelahnya yang dicoba adalah  $10234 + 1 = 10235$ .

## BAGIAN II: SOURCE CODE

Bahasa yang dipilih adalah Python 3. *Source code* program adalah sebagai berikut.

```
# PROGRAM CRYPTARITHMETIC
import time

# INISIALISASI VARIABEL
tabelHuruf = []
nilaiHuruf = []

# DEFINISI FUNGSI
def CharToInt(char):
    if char == '1':
        return 1
    elif char == '2':
        return 2
    elif char == '3':
        return 3
    elif char == '4':
        return 4
    elif char == '5':
        return 5
    elif char == '6':
        return 6
    elif char == '7':
        return 7
    elif char == '8':
        return 8
    elif char == '9':
        return 9
    else:
        return 0

def IntToChar(n):
    if n == 0:
        return '0'
    elif n == 1:
        return '1'
    elif n == 2:
        return '2'
    elif n == 3:
        return '3'
    elif n == 4:
        return '4'
    elif n == 5:
        return '5'
    elif n == 6:
        return '6'
    elif n == 7:
        return '7'
    elif n == 8:
        return '8'
    else:
        return '9'

def StringToInt(string):
    hasil = 0
    for a in string:
        hasil = hasil*10 + CharToInt(a)

    return hasil
```

```

def LetterToNumber(string):
    hasil = ""
    for a in string:
        i = 0
        while a != tabelHuruf[i]:
            i = i + 1
        if nilaiHuruf[i] != 'x':
            hasil += nilaiHuruf[i]
    return StringToInt(hasil)

def InitTabelHuruf():
    global hurufOp
    global hurufHasil
    global tabelHuruf
    for char in hurufHasil:
        if char not in tabelHuruf:
            tabelHuruf.append(char)
    for kata in hurufOp:
        for char in kata:
            if char not in tabelHuruf:
                tabelHuruf.append(char)

def InitNilaiHuruf():
    global tabelHuruf
    global nilaiHuruf
    for i in range(0, len(tabelHuruf)):
        if i == 0:
            nilaiHuruf.append('1')
        elif i == 1:
            nilaiHuruf.append('0')
        else:
            nilaiHuruf.append(IntToChar(i))

def IsHurufAwalNoI():
    hurufawal = False
    global hurufOp
    global hurufHasil
    global tabelHuruf
    global nilaiHuruf

    i = 0
    while (not hurufawal) and (i < len(tabelHuruf)):
        if nilaiHuruf[i] == '0':
            #Jika ada yang nilainya nol
            for kata in hurufOp:
                if kata[0] == tabelHuruf[i]:
                    hurufawal = True
                if hurufHasil[0] == tabelHuruf[i]:
                    hurufawal = True
            i = i + 1
    return hurufawal

def ListToInt(li):
    hasil = 0
    for a in li:
        hasil = hasil * 10 + CharToInt(a)
    return hasil

```

```

def IntToList(angka, length):
    hasil = []
    while angka > 0:
        x = angka % 10
        angka = angka // 10
        hasil.append(IntToChar(x))
    hasil2 = []
    for i in range(0, length-len(hasil)):
        hasil2.append('0')
    for j in hasil[::-1]:
        hasil2.append(j)
    return hasil2

def IsNotUnique():
    temp = []
    notunik = False
    global nilaiHuruf
    for char in nilaiHuruf:
        if char not in temp:
            temp.append(char)
        else:
            notunik = True
    return notunik

def NextTry():
    global nilaiHuruf
    while True:
        curr = ListToInt(nilaiHuruf)
        curr = curr + 1
        newlist = IntToList(curr, len(nilaiHuruf))
        for i in range(0, len(nilaiHuruf)):
            nilaiHuruf[i] = newlist[i]
        if not IsNotUnique():
            break

def IsMaxList():
    global nilaiHuruf
    i = 0
    isMax = True
    while (isMax) and (i < len(nilaiHuruf)):
        if nilaiHuruf[i] != len(nilaiHuruf) - 1:
            isMax = False
        else:
            i = i + 1
    if isMax:
        print(nilaiHuruf)
    return isMax

# PROGRAM UTAMA
# Setting: Berhenti setelah satu jawaban jika True
satujawab = True

# Menerima nama file input dan baca file
namaFile = input("Masukkan path file input: ")
file = open(namaFile, "r")
isiteks = file.readlines()
file.close()

# Proses input ke variabel dan print soal
# Mulai hitung waktu di sini
starttime = time.time()

```

```

hurufOp = []
hurufHasil = ""
bacahasil = False
print()
print("Hasil input:")
for baris in isiteks:
    print(baris.rstrip())
    a = baris.strip()
    if a[-1:] == '+':
        hurufOp.append(a[:-1])
    elif a[0] == '-':
        bacahasil = True
    elif bacahasil:
        hurufHasil += a
    else:
        hurufOp.append(a)

# Mulai percobaan
count = 0
jawab = False
selesai = False
print()
print("Jawaban:")
InitTabelHuruf()

if len(tabelHuruf) > 10:
    print("Tidak bisa dihitung karena huruf unik lebih dari 10.")
else:
    InitNilaiHuruf()

    while not selesai:
        if count > 0:
            NextTry()

        if not IsHurufAwalNo1():
            if not satujawab:
                if IsMaxList():
                    selesai = True

        totalOperand = 0
        count = count + 1
        for kata in hurufOp:
            # Cari hasil penjumlahan semua operand
            totalOperand += LetterToNumber(kata)
        totalHasil = LetterToNumber(hurufHasil)
        if totalHasil == totalOperand:
            # Jika benar, cetak jawaban
            jawab = True
            if satujawab:
                selesai = True

```

```

j = 0
for kata in hurufOp:
    j = j + 1
    for i in range(0, len(hurufHasil) - len(kata)):
        print(" ", end="")
        print(LetterToNumber(kata), end="")
    if j == len(hurufOp):
        print("+")
    else:
        print()
for char in hurufHasil:
    print("-", end="")
print()
print(LetterToNumber(hurufHasil))
print()

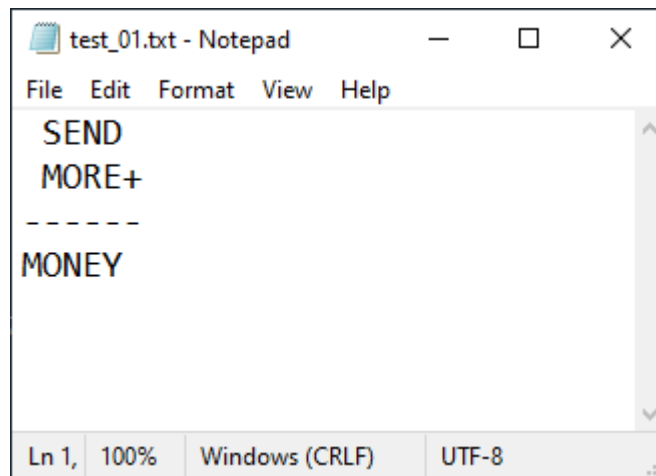
if not jawab:
    print("Tidak ada solusi yang ditemukan.")

print()
print("Waktu eksekusi program:", time.time() - starttime, "detik.")
print("Total tes yang dilakukan:", count, "kali.")

```

### BAGIAN III: SCREENSHOTS

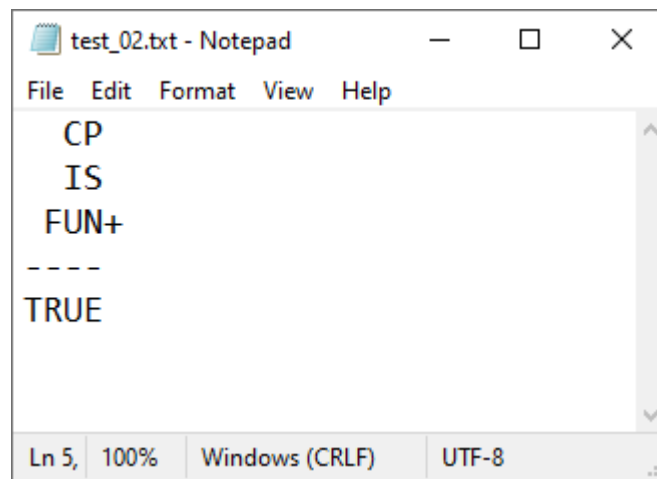
Contoh 1



```
Masukkan path file input: ../test/test_01.txt
Hasil input:
SEND
MORE+
-----
MONEY
Jawaban:
9567
1085+
-----
10652
waktu eksekusi program: 4.705206394195557 detik.
Total tes yang dilakukan: 11217 kali.
>>>
```



## Contoh 2

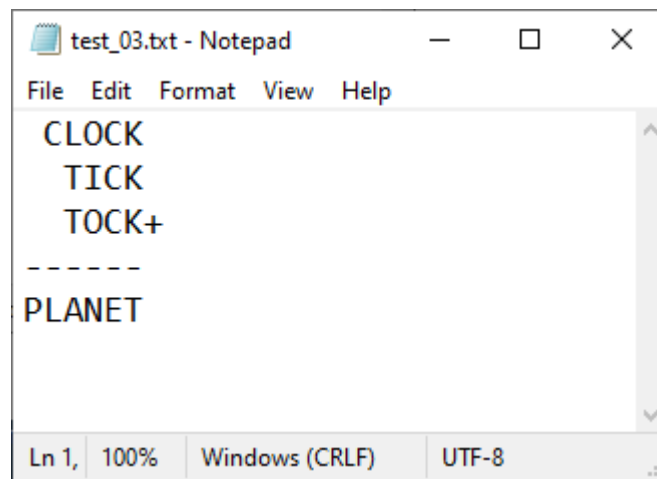


```
Masukkan path file input: ../test/test_02.txt
Hasil input:
  CP
  IS
  FUN+
  ----
  TRUE

Jawaban:
  34
  65
  928+
  ----
  1027

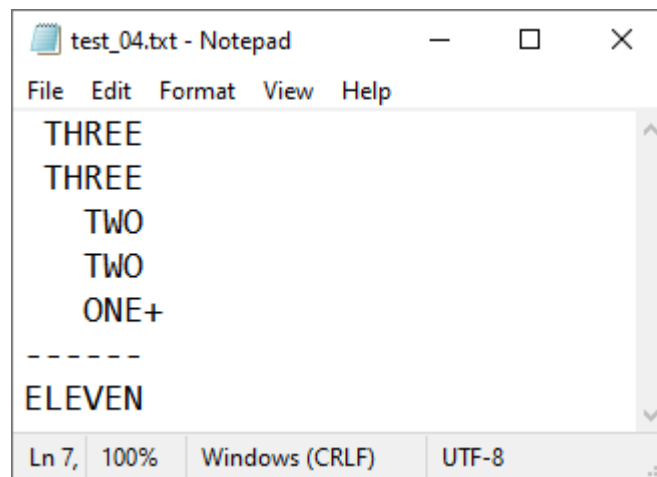
waktu eksekusi program: 52.130096197128296 detik.
Total tes yang dilakukan: 2888 kali.
>>>
```

### Contoh 3



```
Masukkan path file input: ../test/test_03.txt
Hasil input:
CLOCK
TICK
TOCK+
-----
PLANET
Jawaban:
90892
6592
6892+
-----
104376
waktu eksekusi program: 267.7563097476959 detik.
Total tes yang dilakukan: 11231 kali.
>>>
```

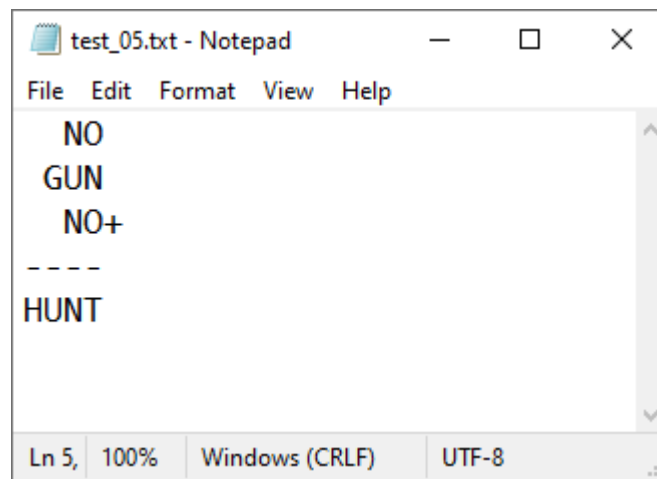
#### Contoh 4



```
Masukkan path file input: ../test/test_04.txt
Hasil input:
THREE
THREE
TWO
TWO
ONE+
-----
ELEVEN
Jawaban:
84611
84611
803
803
391+
-----
171219

waktu eksekusi program: 842.7525861263275 detik.
Total tes yang dilakukan: 200121 kali.
>>>
```

### Contoh 5

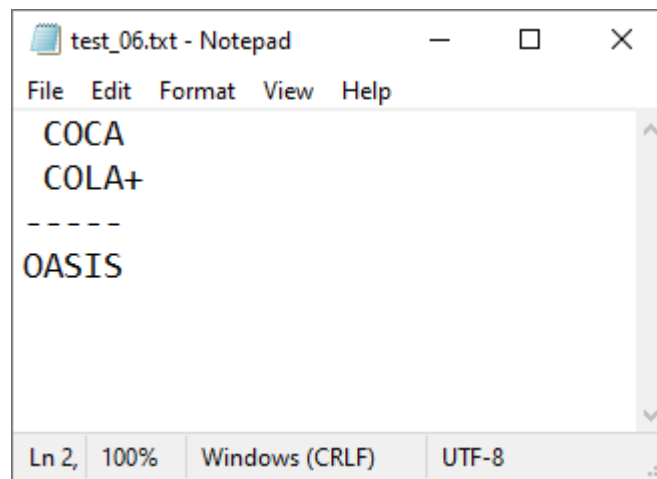


```
Masukkan path file input: ../test/test_05.txt
Hasil input:
  NO
  GUN
  NO+
  ----
HUNT

Jawaban:
  87
  908
  87+
  ----
1082

waktu eksekusi program: 0.1618645191192627 detik.
Total tes yang dilakukan: 1285 kali.
>>>
```

## Contoh 6

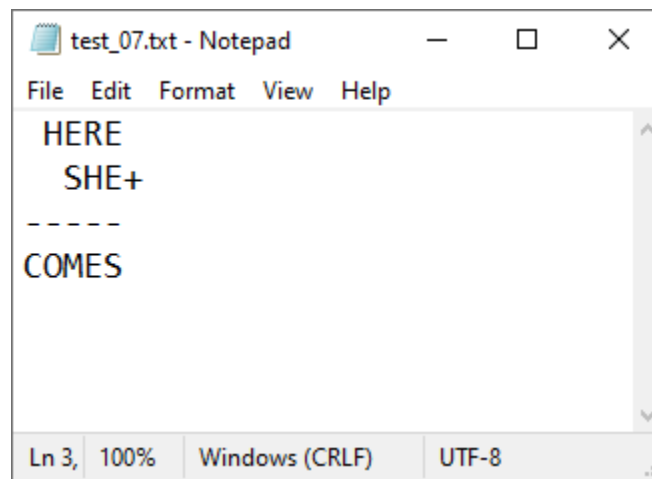


```
Masukkan path file input: ../test/test_06.txt
Hasil input:
COCA
COLA+
-----
OASIS

Jawaban:
8186
8106+
-----
16292

waktu eksekusi program: 0.651336669921875 detik.
Total tes yang dilakukan: 7946 kali.
>>>
```

## Contoh 7

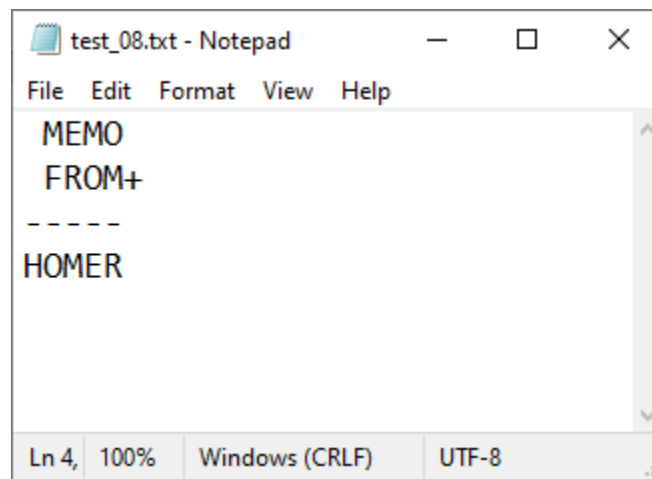


```
Masukkan path file input: ../test/test_07.txt
Hasil input:
  HERE
  SHE+
  -----
  COMES

Jawaban:
  9454
  894+
  -----
  10348

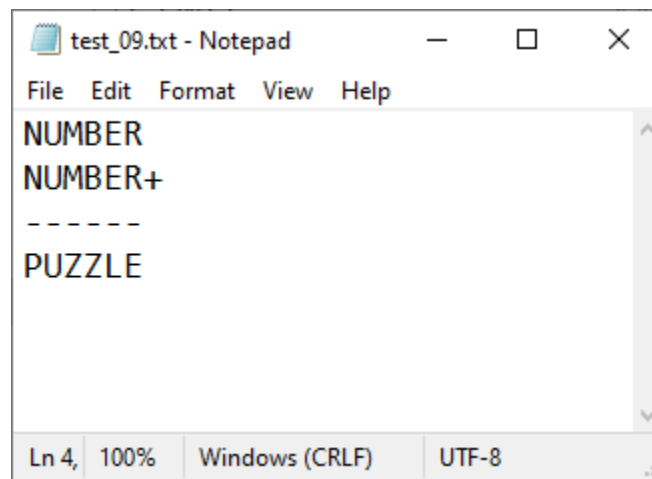
waktu eksekusi program: 0.17763090133666992 detik.
Total tes yang dilakukan: 1058 kali.
>>>
```

## Contoh 8



```
Masukkan path file input: ../test/test_08.txt
Hasil input:
MEMO
FROM+
-----
HOMER
Jawaban:
8485
7358+
-----
15843
waktu eksekusi program: 0.5887305736541748 detik.
Total tes yang dilakukan: 6452 kali.
>>>
```

### Contoh 9



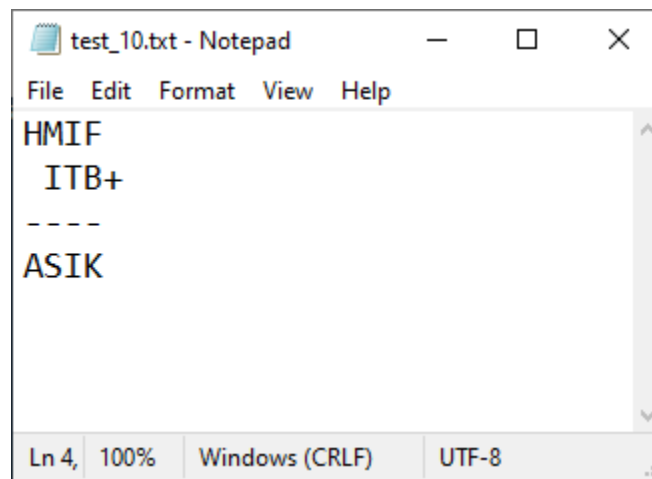
```
Masukkan path file input: ../test/test_09.txt
Hasil input:
NUMBER
NUMBER+
-----
PUZZLE

Jawaban:
201689
201689+
-----
403378

waktu eksekusi program: 3616.109523296356 detik.
Total tes yang dilakukan: 981148 kali.
>>>
```



### Contoh 10



```
Masukkan path file input: ../test/test_10.txt
Hasil input:
HMIF
ITB+
----
ASIK

Jawaban:
1637
398+
----
2035

waktu eksekusi program: 776.9210493564606 detik.
Total tes yang dilakukan: 288756 kali.
>>>>
```

#### BAGIAN IV: ALAMAT DAN CHECKLIST

Alamat kode program:

<https://github.com/riisuki/cryptarithmic>

Checklist:

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>	✓	