**LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA**

1. Algoritma Brute Force

Program akan memberikan beberapa output untuk panduan memulai kalkulasi, setelah input nama file timer akan dimulai. Program akan membaca file lalu input akan dibuat menjadi objek list. Untuk tiap-tiap baris akan disimpan dalam list of lists dengan ilustrasi [[‘s’,’e’], [‘a’,’b’]], kemudian akan dibentuk satu list baru yang berisikan seluruh huruf dengan tiap elemennya unik.

Berikutnya akan dibentuk suatu list value yang berisikan 10 angka yang mungkin dipakai. List tersebut kemudian akan dihasilkan salah satu permutasinya dari total permutasi 10! Permutasi yang dihasilkan akan dicocokkan dengan list seluruh huruf dengan batasan huruf yang mewakili angka pertama pada persoalan tidak boleh 0. Jika setelah dicocokkan ada huruf yang mewakili angka pertama dan bernilai 0 (merepresentasikan angka 0) maka program akan langsung mengulang ke pembuatan permutasi. Jika seluruh huruf valid, akan dibentuk suatu list of lists yang berisikan angka tiap huruf dengan ilustrasi [[‘s’,’e’], [‘a’,’b’], [‘c’,’d’]] -> [[1,2], [3,4], [5,6]]. Isi dari list of lists ini kemudian akan dibentuk menjadi satu list yang berisi angka per operan, menurut contoh sebelumnya menjadi seperti ini [12,34,56]. Menurut persoalan cryptarithm, seharusnya index pertama dari list terbaru dijumlahkan dengan index kedua akan menghasilkan index ketiga (akan berlaku index lebih banyak jika operan lebih dari 2). Jika setelah dibandingkan, penjumlahan 2 (atau 3,dst) index awal dengan index terakhir tidak sama, maka program akan mengulang ke pembuatan permutasi baru. Jika hasil perbadningan sama, program akan langsung keluar perulangan dan list terbaru akan disalin ke template list jawaban yang berada di luar perulangan.

Persoalan yang masuk akan ditampilkan ke layar. Kemudain, jika template list jawaban terisi dengan jawaban benar, akan ditampilkan solusi dari persoalan sedangkan jika memang tidak ada solusi maka akan ditampilkan “no solution(s)”. Terakhir ada 2 keluaran yaitu jumlah perulangan yang dilakukan dan waktu program berjalan dari setelah membaca file hingga keluaran terakhir (jumlah perulangan).

1. Source Code Program

import time

import copy

def swap(a,i,j):

#prosedur swap sederhana

a[i],a[j] = a[j],a[i]

#open file

print("Asumsikan seluruh input valid, dengan batasan valid jumlah huruf unik<=10")

name = input("Masukkan nama file dengan format: 1.txt;2.txt;...;10.txt\n")

file = open("../test/"+name,"r")

print("Sabar yaa, lagi diitung")

start = time.time()

listall = []

lt = []

answer =[]

isi = file.readlines()

for i in isi:

temp = []

for j in i:

if(j=='-' or j=='+' or j=="\n"):

break

else:

temp.append(j)

listall.append(temp)

listall.remove([])

#membentuk list berisi huruf unik dari list of lists yang berisi soal

for i in listall:

for j in i:

if j not in lt:

lt.append(j)

#inisiasi permutasi

value = [0,1,2,3,4,5,6,7,8,9]

limit = len(value)-1

tag = 0

test\_made = 0

while tag!=limit:

#generate 1 of 10! permutations

if (tag==0):

swap(value,tag,tag+1)

tag += 1

while (tag<len(value)-1 and value[tag]>value[tag+1]):

tag += 1

else:

if (value[tag+1]>value[0]):

swap(value,tag+1,0)

else:

x = 0

y = tag

mid = (x+y) // 2

val = value[tag+1]

while(not (value[mid]<val and value[mid-1]>val)):

if (value[mid]<val):

y = mid - 1

else:

x = mid + 1

mid = (x+y) // 2

swap(value,tag+1,mid)

for i in range(tag//2 + 1):

swap(value,i,tag-i)

tag = 0

#menentukan apakah suatu huruf dapat merepresentasikan angka 0

#dengan mengecek list angka hasil permutasi dengan list huruf

leading\_error = False

for i in range(len(lt)):

for j in listall:

if ((lt[i]==j[0]) and (value[i]==0)):

leading\_error = True

break

if not leading\_error:

#menghubungkan angka dengan huruf yang merepresentasikan

listallnum = copy.deepcopy(listall)

for i in listallnum:

for j in range(len(i)):

for k in range(len(lt)):

if (i[j]==lt[k]):

i[j] = value[k]

#mengubah list angka yang masih berformat sama dengan list huruf

#menjadi list dengan isi integer

vallist = []

for i in listallnum:

currval = 0

for j in range(len(i)):

currval += i[j]\*(10\*\*(len(i)-j-1))

vallist.append(currval)

#menghitung total dari operan yang dimiliki

total = 0

operan = len(vallist)

for i in range(operan-1):

total += vallist[i]

test\_made += 1

if(total==vallist[operan-1]):

#total dari operan yang ada sama dengan index list angka terakhir

#yang merepresentasikan jawaban yang benar

answer = vallist.copy()

break #tag = limit -> tag==limit

else:

del listallnum

del vallist

#output soal

for i in range(len(listall)):

for j in listall[i]:

print(j,end="")

if i == len(listall)-2:

print("+")

print("-------")

else:

print()

#output solusi

if answer!=[]:

print()

print("Solusinya ini yaaa, thank you telah sabar menunggu")

for i in range(len(answer)):

if i == len(answer) - 2:

print(answer[i],end="")

print("+")

print("-------")

else:

print(answer[i])

print()

else:

print("no solution(s)")

print("program looped for", test\_made, "iterations")

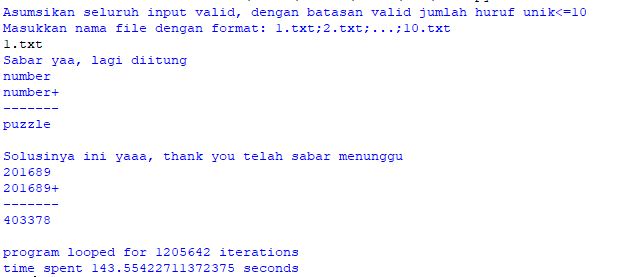
end = time.time()

print("time spent", end-start, "seconds")

1. Source Code Program

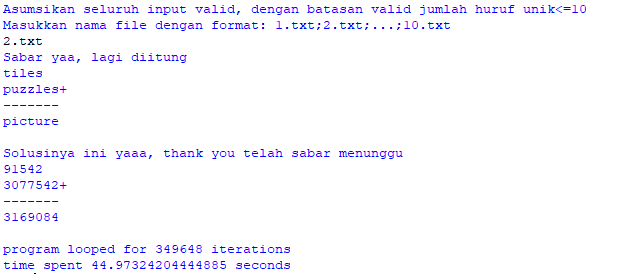
3.1 Test Case 1





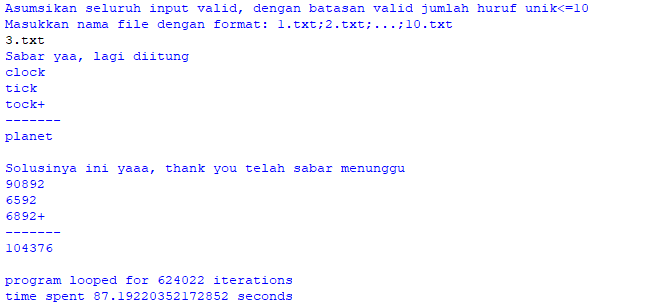
* 1. Test Case 2





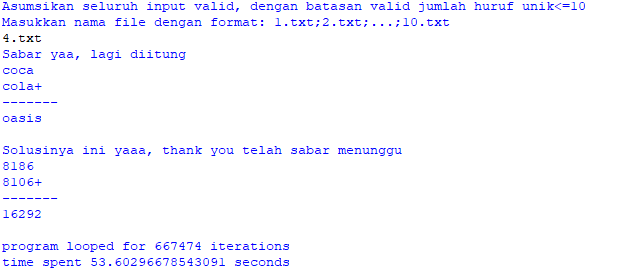
* 1. Test Case 3



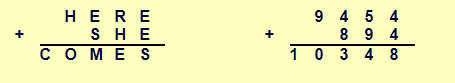


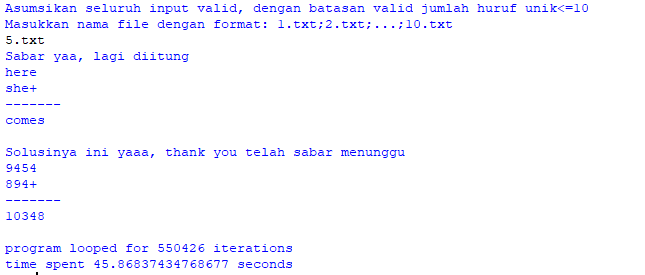
* 1. Test Case 4



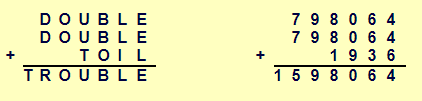


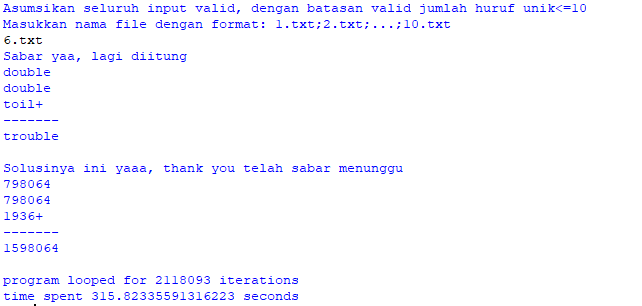
* 1. Test Case 5





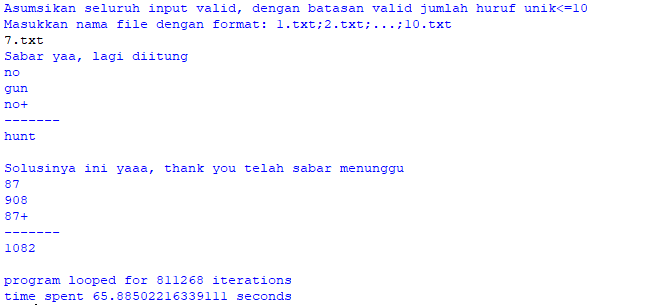
* 1. Test Case 6



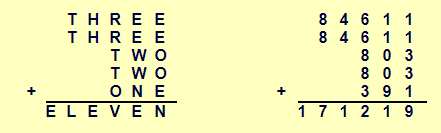


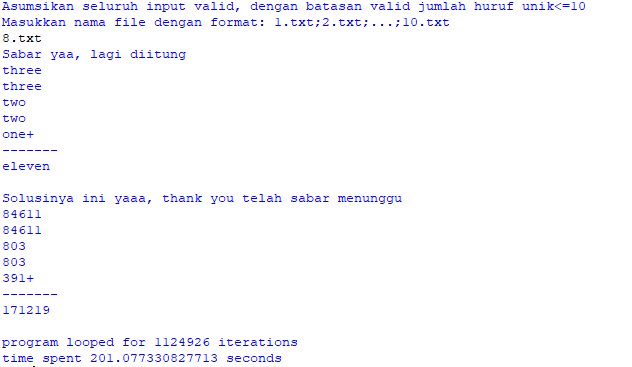
* 1. Test Case 7





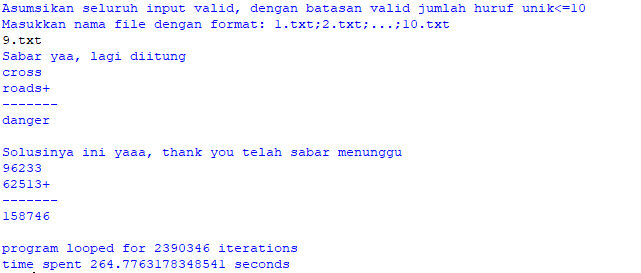
* 1. Test Case 8





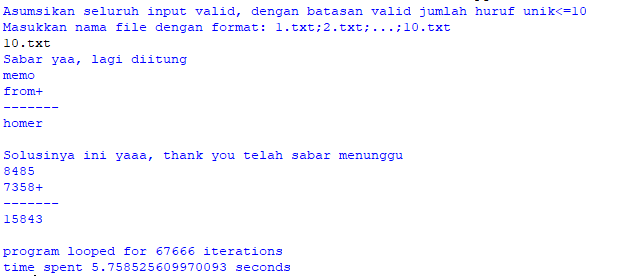
* 1. Test Case 9





3.10 Test Case 10





1. Alamat Drive Kode Program, Repository Github

<https://github.com/ayahyaaa/IF2211-Stima-Tucil1/blob/main/src/cryptarythm.py>

