LAPORAN TUGAS KECIL 1

IF2211 STRATEGI ALGORITMA

**Penyelesaian *Cryptarithmetic* dengan Algoritma *Brute Force***



Disusun oleh:

AKEYLA PRADIA NAUFAL

13519178

Algoritma yang Digunakan

Algoritma yang digunakan untuk menyelesaikan sebuah *cryptarithmetic* ini adalah:

1. Mula-mula, semua input yang berupa kata dimasukkan dan disimpan
2. Banyak huruf berbeda dihitung.
   1. Jika banyak huruf berbeda di semua kata ini melebihi 10, maka disimpulkan bahwa puzzle ini tidak memiliki solusi
   2. Jika banyak huruf berbeda di semua kata ini di antara 7 dan 10 (inklusif), maka:
      1. Misal
      2. Tentukan indeks terbesar di sehingga
      3. Cari indeks terbesar di sehingga
      4. Tukar posisi dan
      5. Balik semua elemen yang ada di posisi sampai
      6. Samakan digit di dengan huruf yang ada di puzzle sesuai dengan urutan inputnya. Cek terlebih dahulu apakah ada kata yang diawali digit .
      7. Cek apakah memenuhi. Jika ya, cek apakah solusi ini pernah ditemukan. Jika belum ditemukan, tuliskan solusi ini di layar dan catat
      8. Ulangi hingga
   3. Jika banyak huruf berbeda di semua kata ini dengan , maka:
      1. Ambil
      2. Jika memiliki dua digit yang sama, ganti nilai menjadi . Jika tidak, samakan digit di dengan huruf yang ada di puzzle sesuai dengan urutan inputnya. Cek terlebih dahulu apakah ada kata yang diawali digit .
      3. Cek apakah kumpulan digit dan huruf ini memenuhi puzzle. Jika ya, tuliskan solusi tersebut di layar.
      4. Naikkan terus nilai sampai

Source Program

import time

def solveManual(copyWords, letterToNum, trial, solusi):

    if(all([int(letterToNum[z[0]]) != 0 for z in copyWords])):

        for i in range(0,len(copyWords)):

            word = copyWords[i]

            for letter, num in letterToNum.items():

                word = word.replace(letter, str(num))

            copyWords[i] = word

        if(sum([int(z) for z in copyWords[:-1]]) == int(copyWords[-1]) and letterToNum not in solusi):

            for word in copyWords[:-1]:

                print(word)

            print('----+')

            print(copyWords[-1])

            print('!!!!!!')

            print("Jawaban ini ditemukan setelah " + str(trial) + " tes")

            print("\n")

            solusi.append(letterToNum)

def solveExternal(copyWords, letterToNum, trial, solusi, outFile):

    if(all([int(letterToNum[z[0]]) != 0 for z in copyWords])):

        for i in range(0,len(copyWords)):

            word = copyWords[i]

            for letter, num in letterToNum.items():

                word = word.replace(letter, str(num))

            copyWords[i] = word

        if(sum([int(z) for z in copyWords[:-1]]) == int(copyWords[-1]) and letterToNum not in solusi):

            for word in copyWords[:-1]:

                print(word, file = outFile)

            print('----+', file = outFile)

            print(copyWords[-1], file = outFile)

            print('!!!!!!', file = outFile)

            print("Jawaban ini ditemukan setelah " + str(trial) + " tes", file = outFile)

            print("\n")

            solusi.append(letterToNum)

def solver(words, letterList, solusi, tipe, outFile = None):

        start = time.time()

        if(len(letterList) < 7):

            for z in range(10 \*\* (len(letterList) - 1), 10 \*\* len(letterList)):

                if(len(set(y for y in str(z))) == len(letterList)):

                    letterToNum = dict(zip(letterList, [y for y in str(z)]))

                    copyWords = words.copy()

                    if tipe == 1:

                        solveManual(copyWords, letterToNum, z, solusi)

                    else:

                        solveExternal(copyWords, letterToNum, z, solusi, outFile)

        elif(len(letterList) <= 10):

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

            for z in range(1, 3265920):

                max = 0

                for i in range(0, 9):

                    if(numList[i] < numList[i+1]):

                        max = i

                maxG = max+1

                for j in range(max, 10):

                    if(numList[j] > numList[max]):

                        maxG = j

                temp = numList[max]

                numList[max] = numList[maxG]

                numList[maxG] = temp

                left = max+1

                right = 9

                while left < right:

                    temp = numList[left]

                    numList[left] = numList[right]

                    numList[right] = temp

                    left += 1

                    right -= 1

                letterToNum = dict(zip(letterList, numList[:len(letterList)]))

                copyWords = words.copy()

                if tipe == 1:

                    solveManual(copyWords, letterToNum, z, solusi)

                else:

                    solveExternal(copyWords, letterToNum, z, solusi, outFile)

        else:

            print("Huruf yang diberikan terlalu banyak")

        end = time.time()

        print("Waktu yang diperlukan adalah " + str(end - start) + " detik")

while(True):

    letterList = []

    words = []

    solusi = []

    print("--------------")

    print("--------------")

    print("Selamat datang di cryptarithm solver")

    print("Silakan memilih cara memasukkan input:")

    print("1. Manual")

    print("2. Melalui file eksternal")

    print("Masukkan pilihan Anda (1/2): ")

    masukan = int(input())

    while(masukan not in [1, 2]):

        print("Masukkan pilihan Anda (1/2): ")

        masukan = int(input())

    tipe = masukan

    if(masukan == 1):

        print("Silakan memasukkan cryptarithm yang ingin diselesaikan")

        firstWord = input()

        for x in firstWord:

            if x not in letterList:

                letterList.append(x)

        words.append(firstWord)

        while(True):

            secondWord = input()

            if secondWord[-1] == '+':

                for x in secondWord[:-1]:

                    if x not in letterList:

                        letterList.append(x)

                words.append(secondWord[:-1])

                break

            else:

                for x in secondWord:

                    if x not in letterList:

                        letterList.append(x)

                words.append(secondWord)

        bar = input()

        result = input()

        for x in result:

            if x not in letterList:

                letterList.append(x)

        words.append(result)

        print("Solusi:")

        print("------------")

        solver(words, letterList, solusi, tipe)

    elif(masukan == 2):

        print("Masukkan nama file yang ingin dibuka")

        inFile = input()

        inF = open(inFile, 'r')

        print("Masukkan nama file yang ingin ditulis")

        outFile = input()

        outF = open(outFile, 'w')

        firstWord = inF.readline()[:-1]

        for x in firstWord:

            if x not in letterList:

                letterList.append(x)

        words.append(firstWord)

        while(True):

            secondWord = inF.readline()[:-1]

            if secondWord[-1] == '+':

                for x in secondWord[:-1]:

                    if x not in letterList:

                        letterList.append(x)

                words.append(secondWord[:-1])

                break

            else:

                for x in secondWord:

                    if x not in letterList:

                        letterList.append(x)

                words.append(secondWord)

        bar = inF.readline()

        result = inF.readline()[:-1]

        for x in result:

            if x not in letterList:

                letterList.append(x)

        words.append(result)

        print("Solusi:", file  = outF)

        print("------------", file = outF)

        solver(words, letterList, solusi, tipe, outF)

        outF.close()

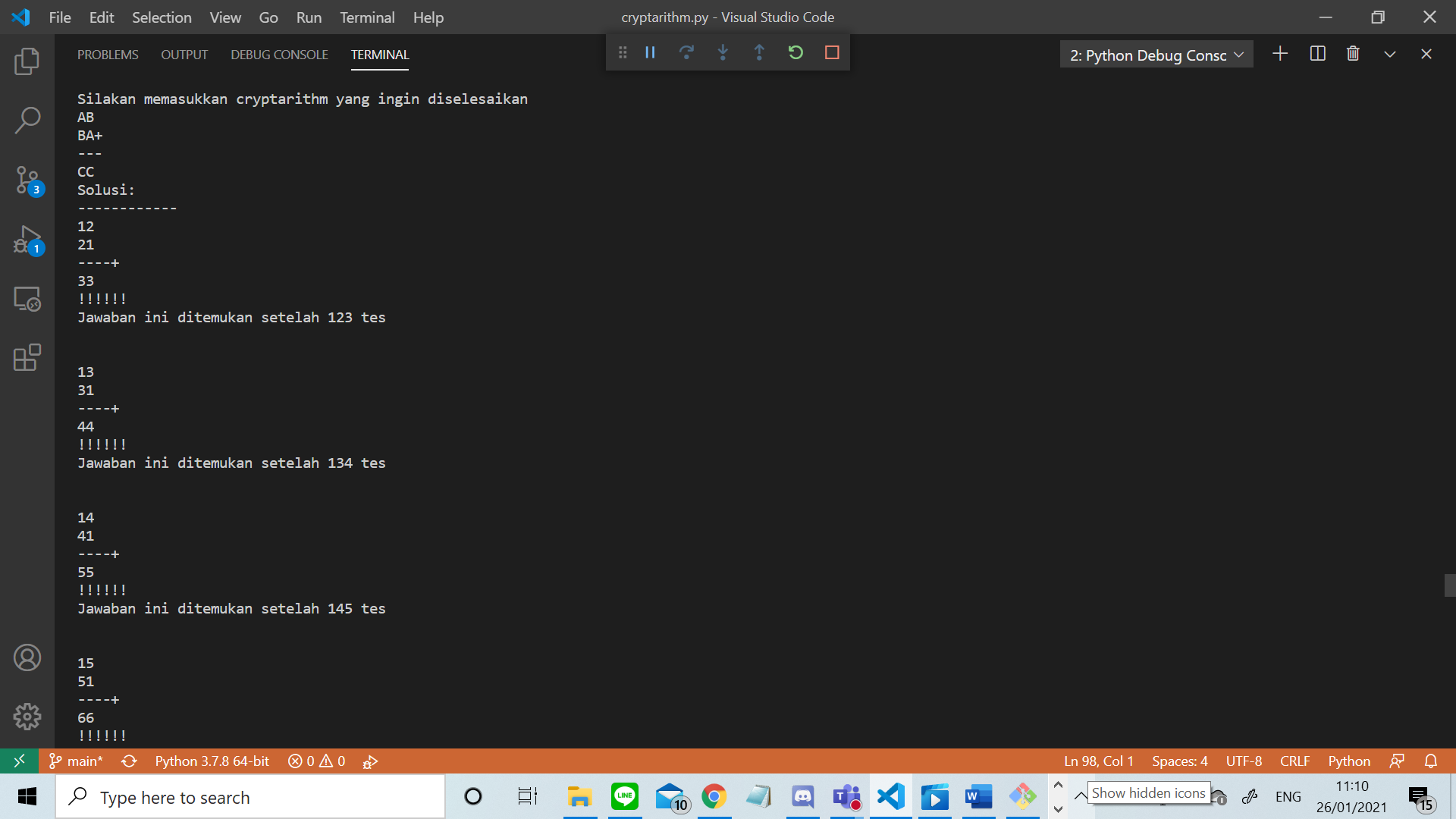
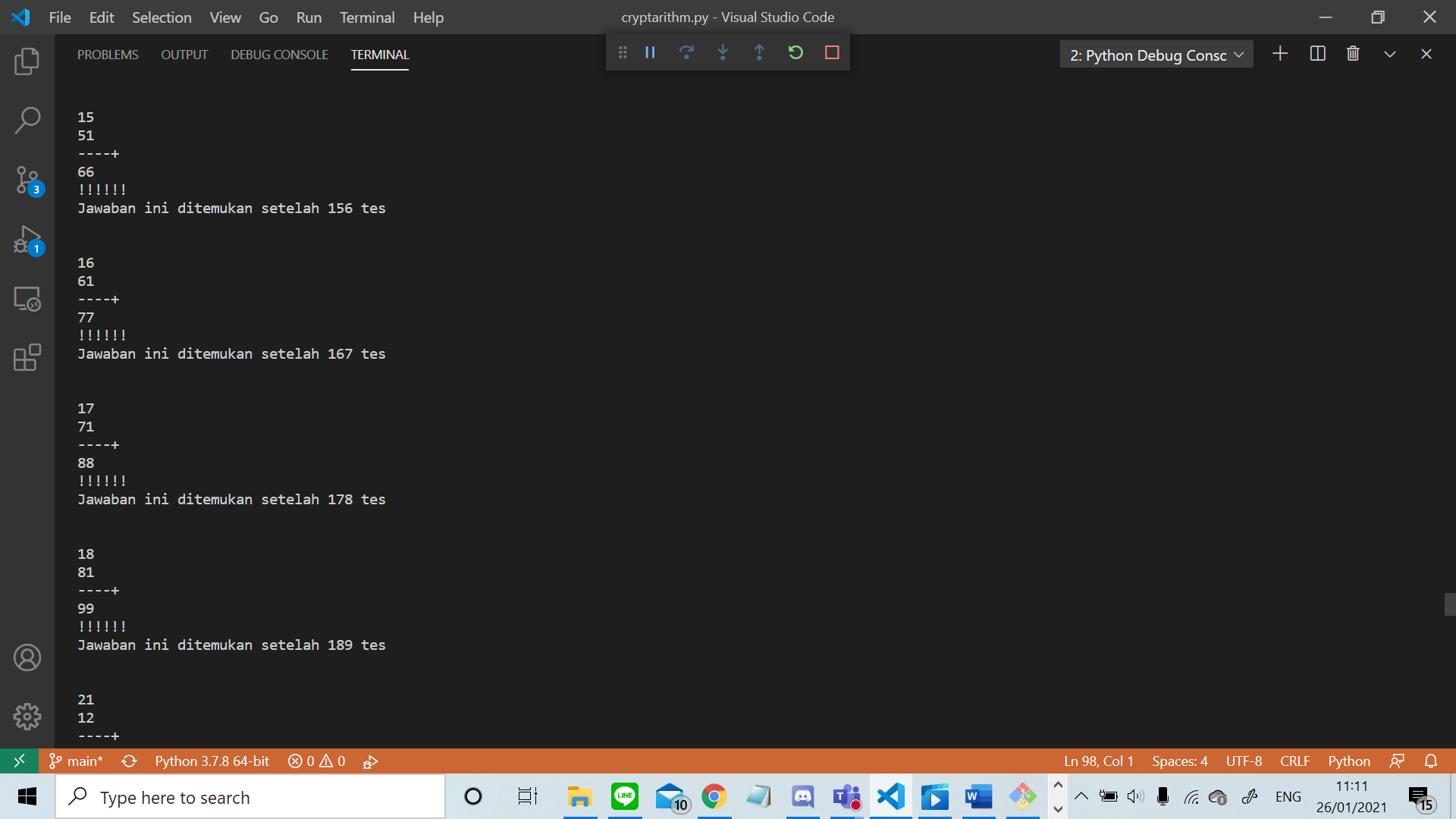
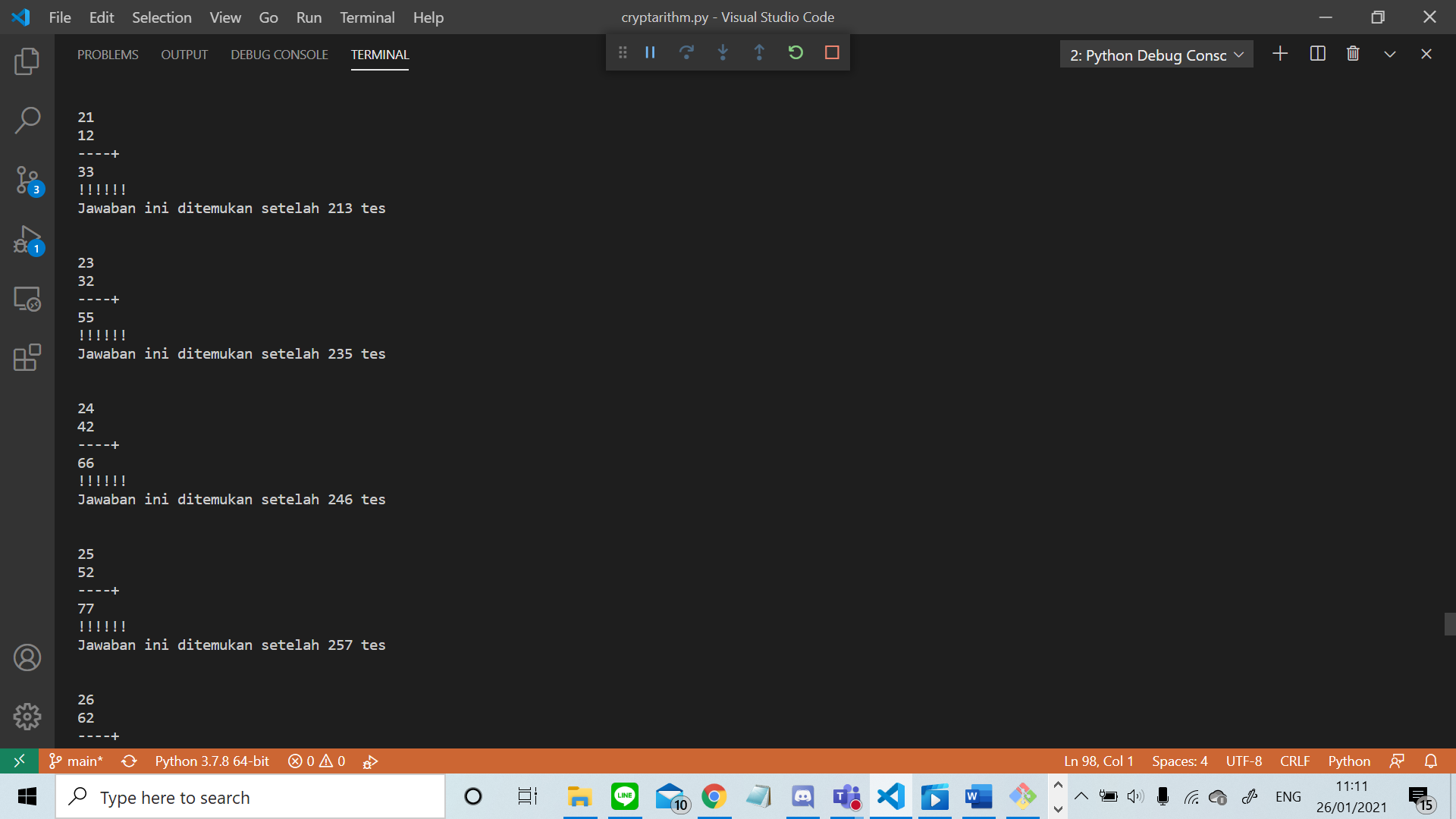
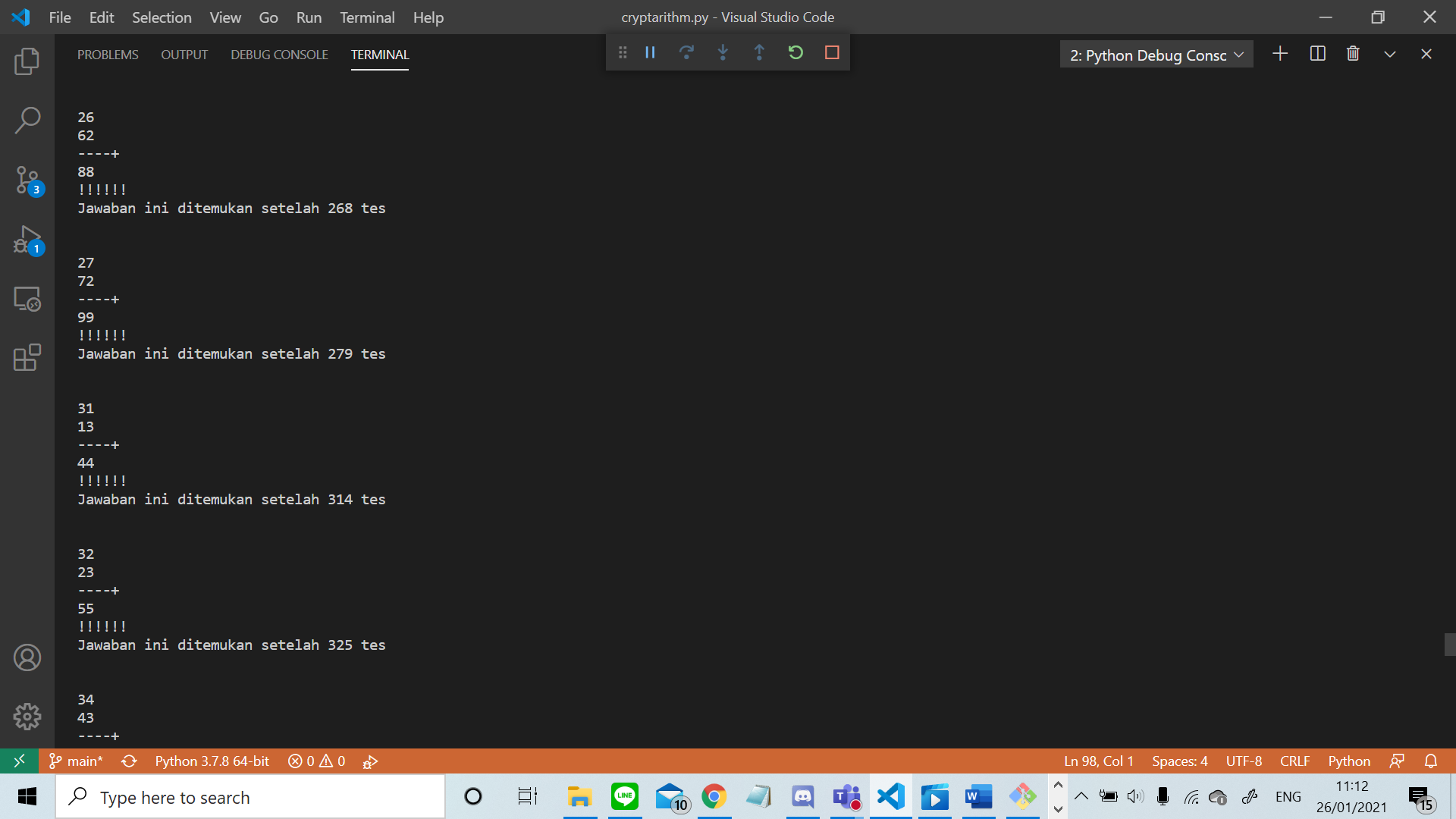
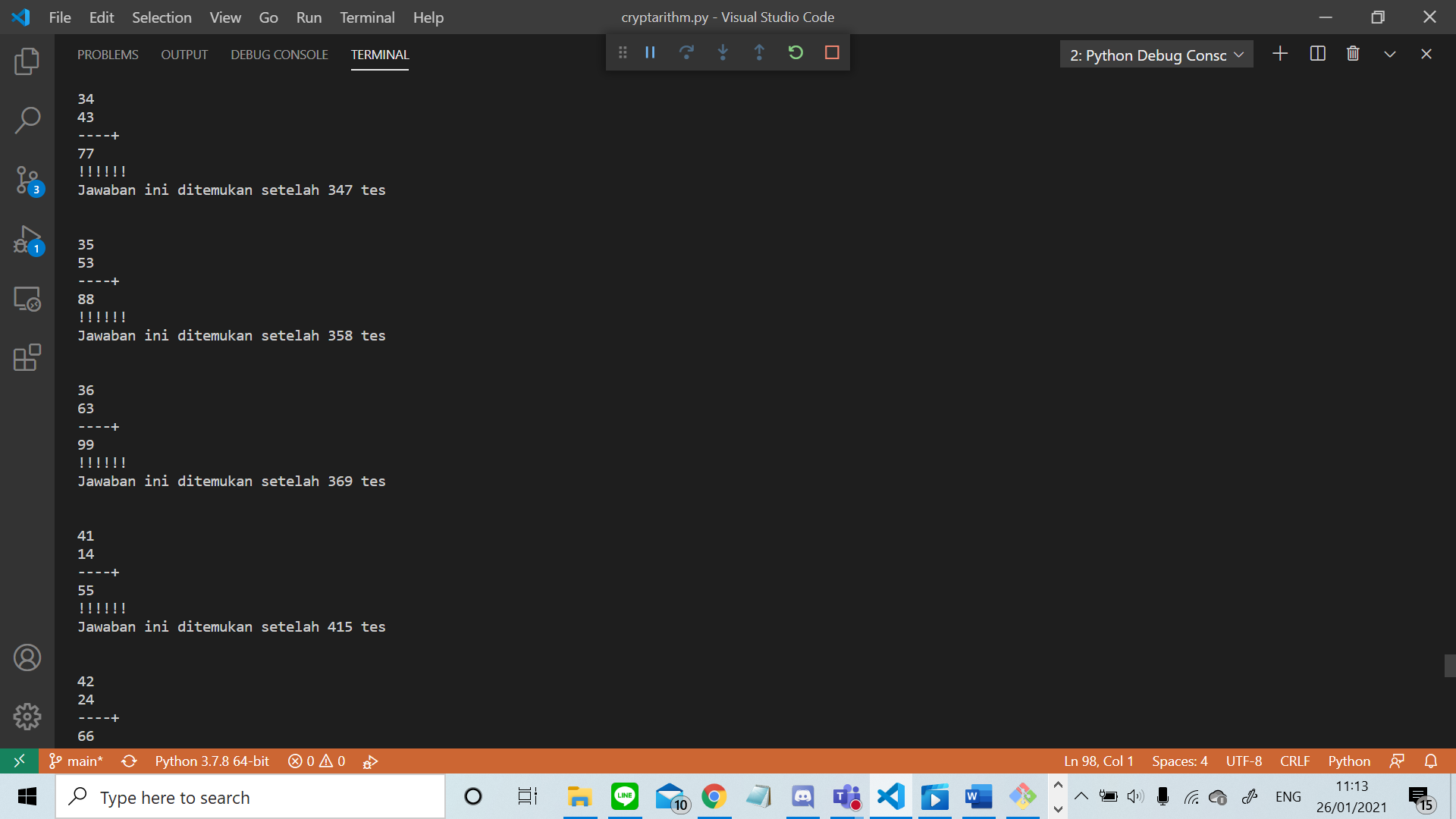
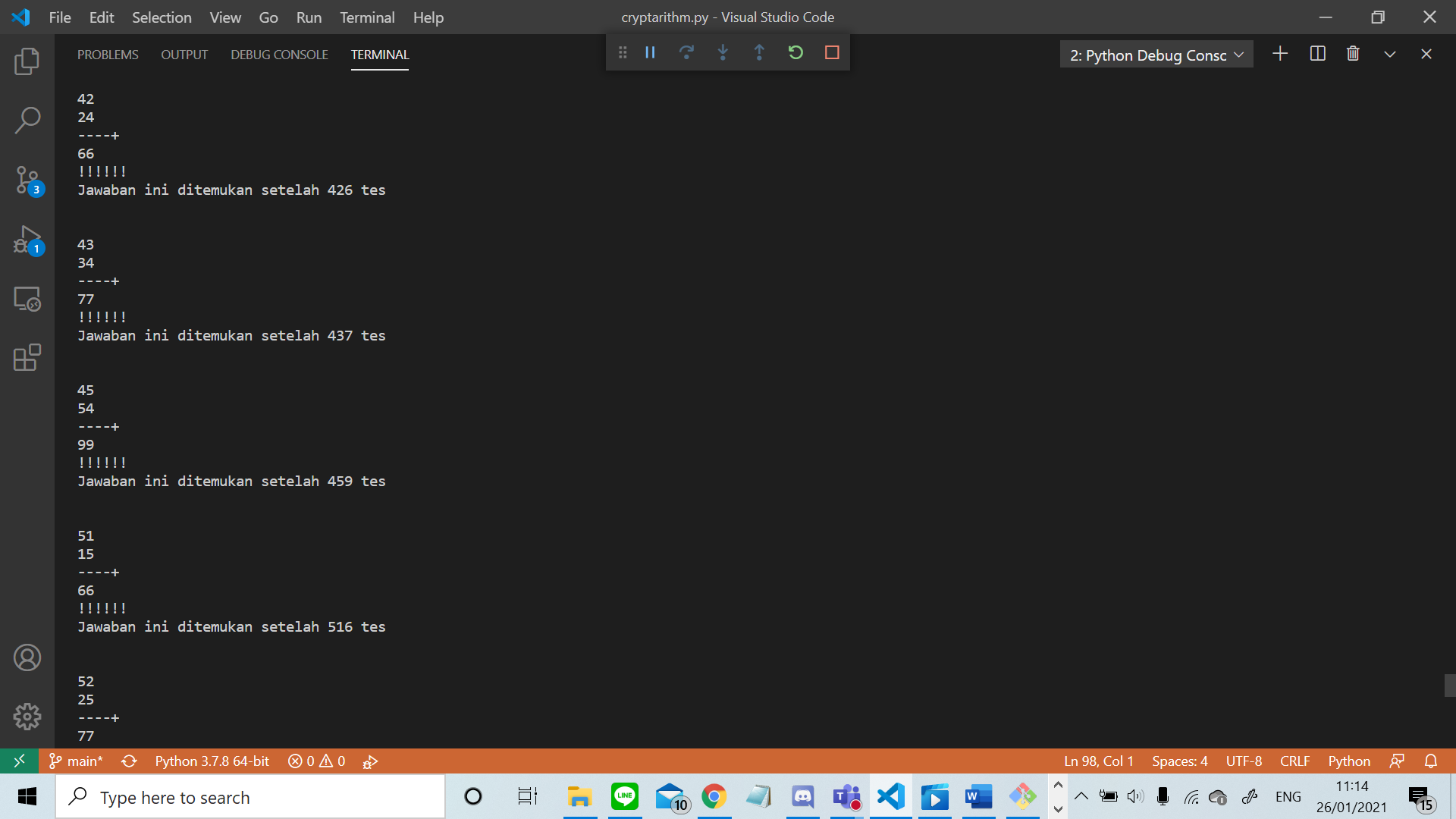
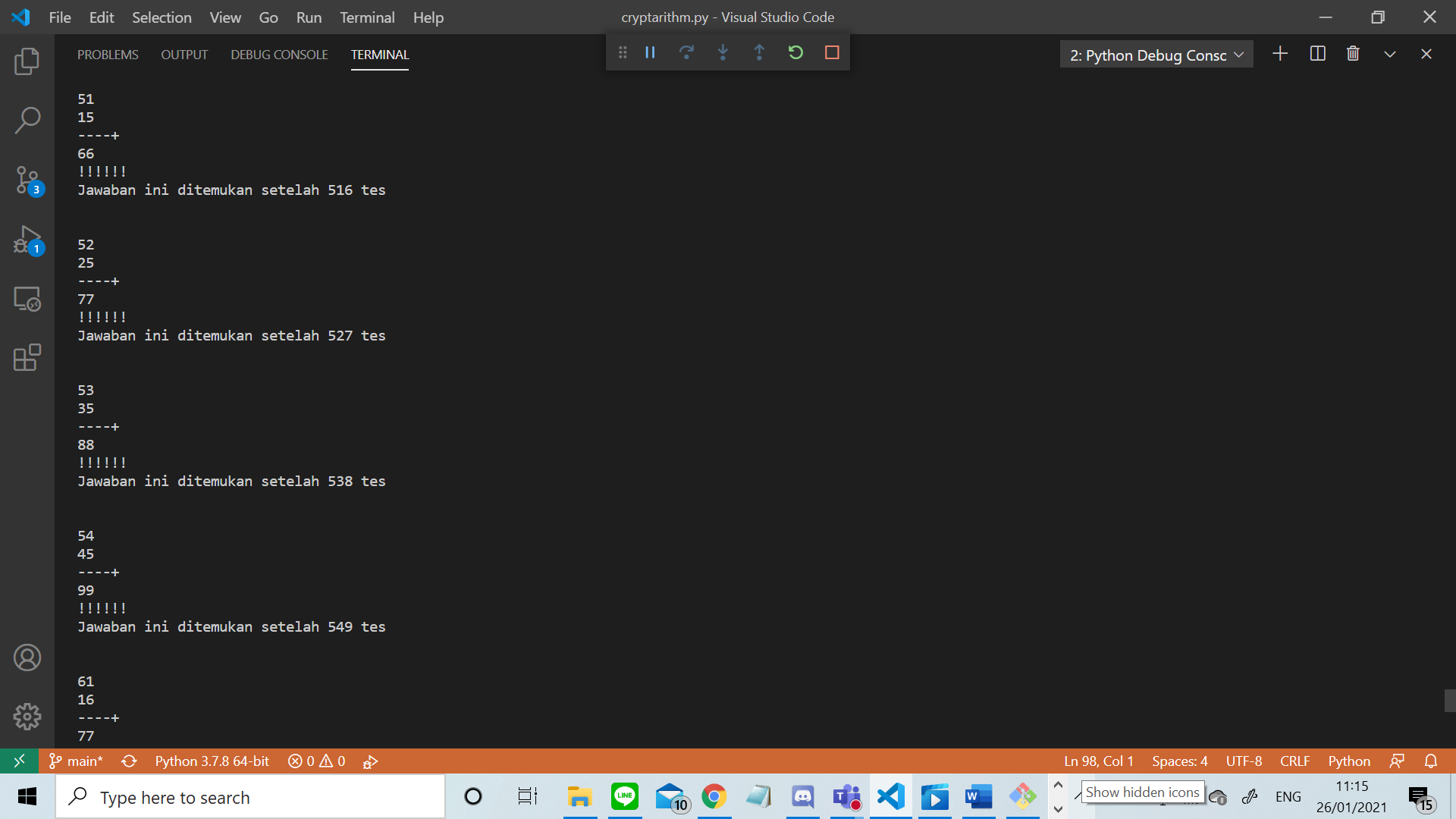
        inF.close()

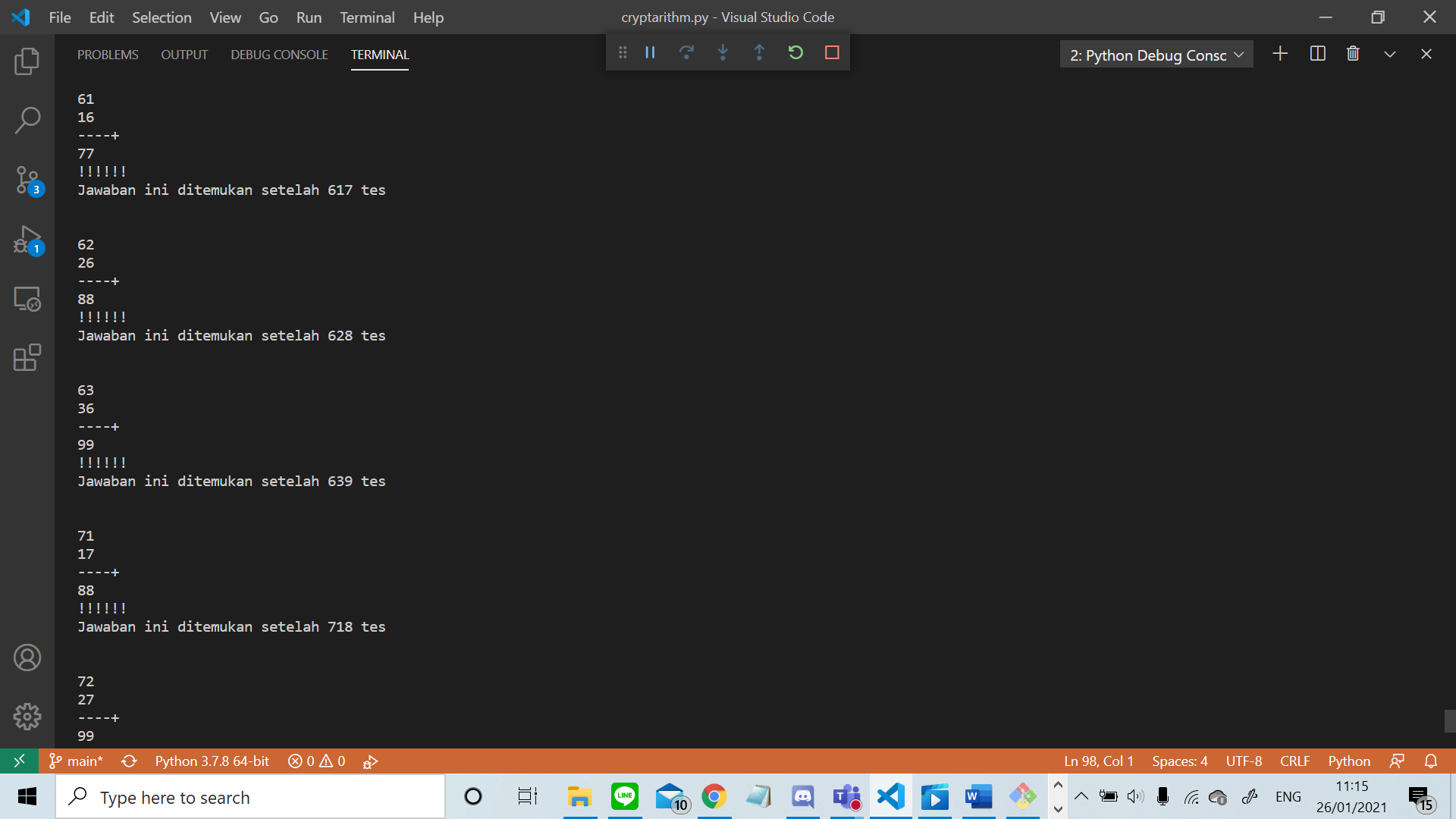
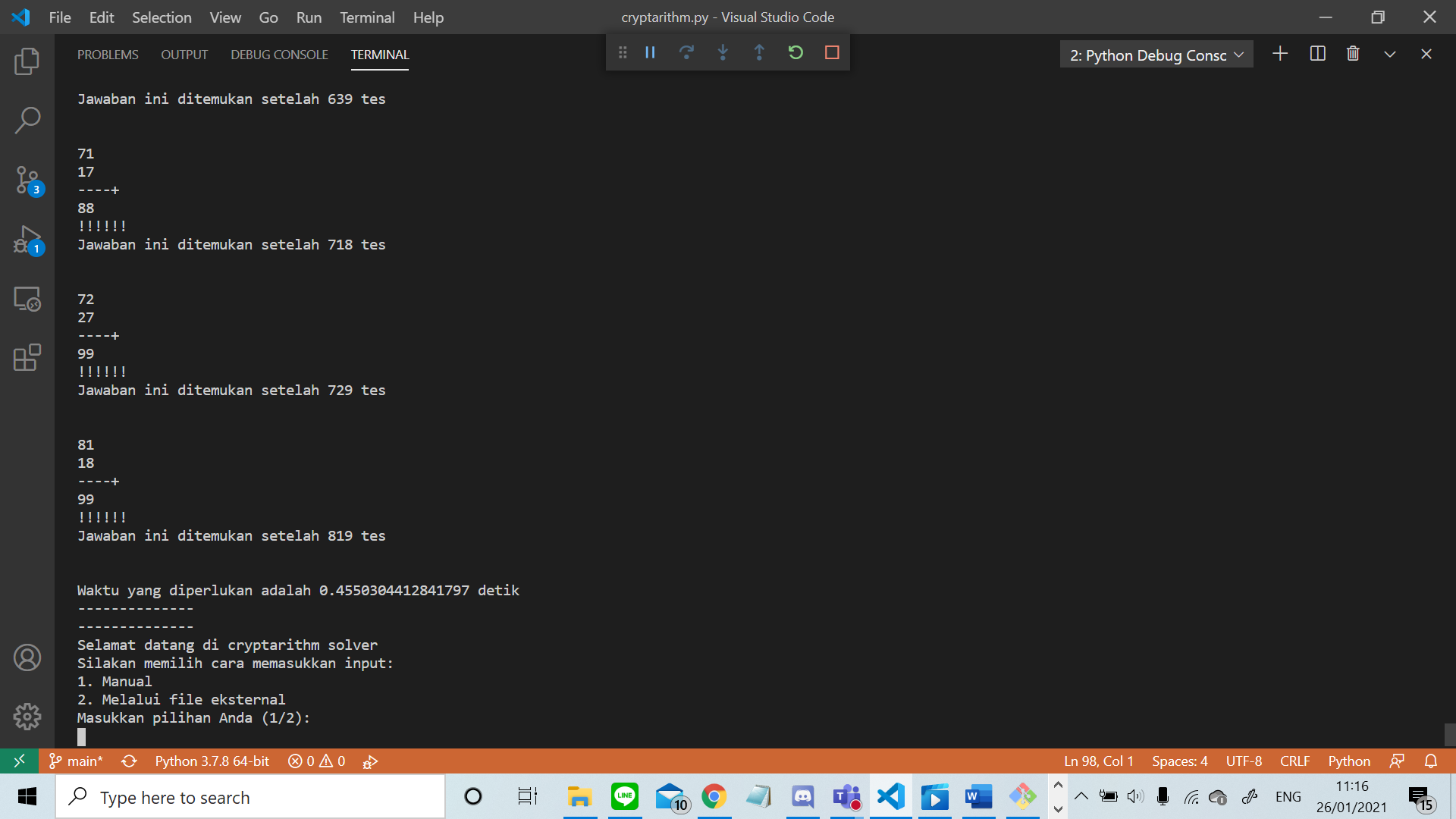
    letterList.clear()

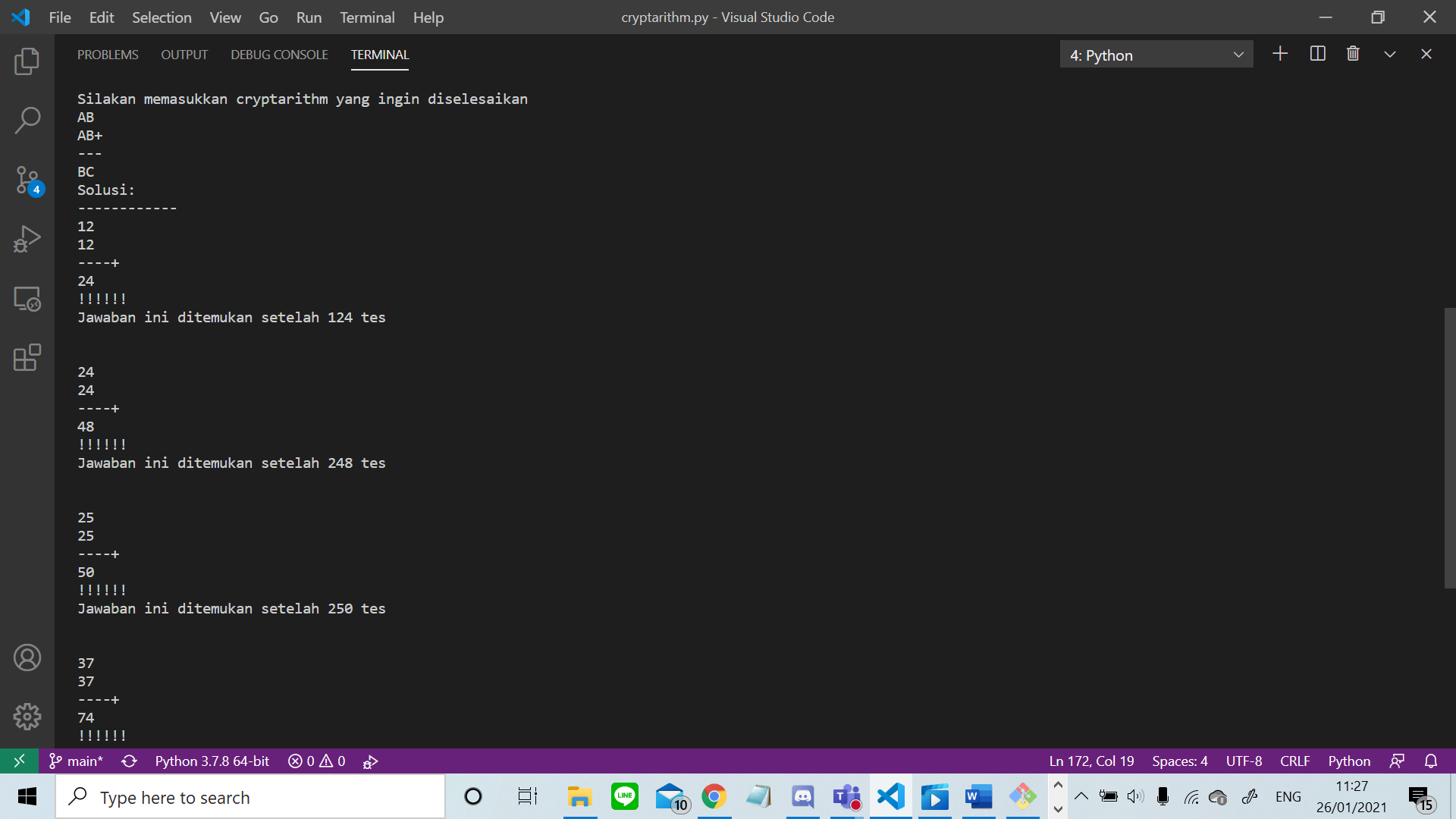
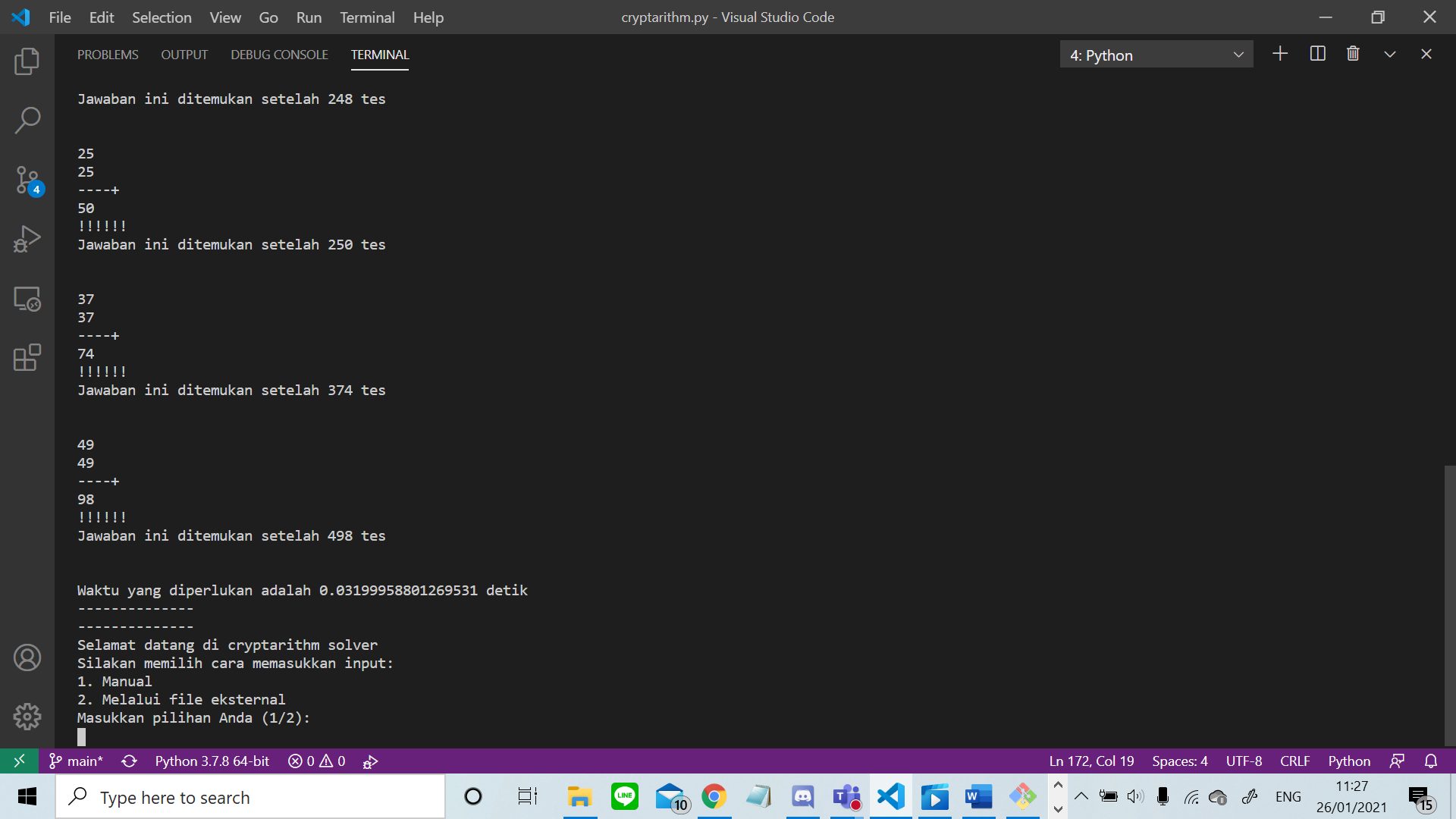
    solusi.clear()

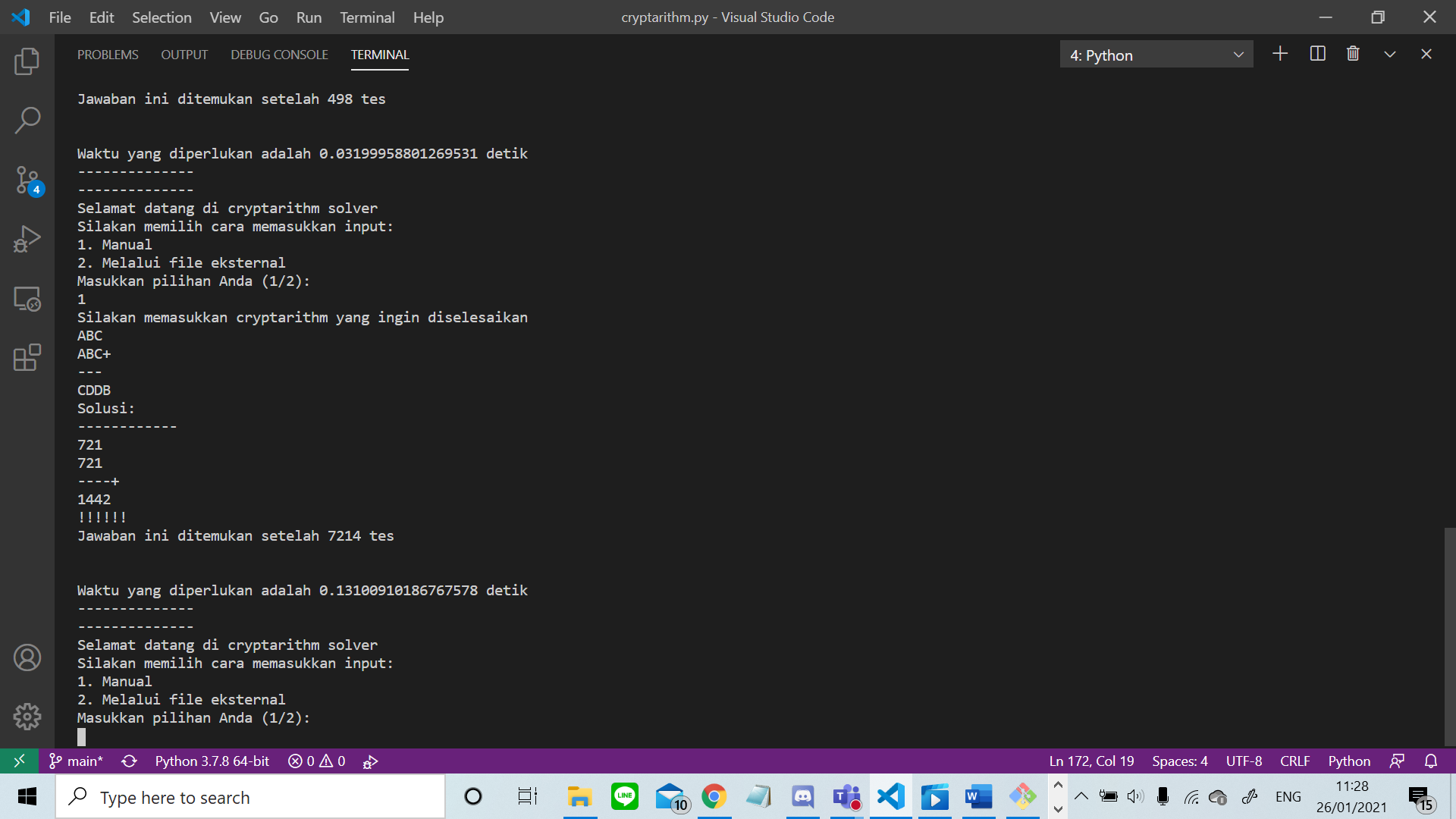
    solusi.clear()

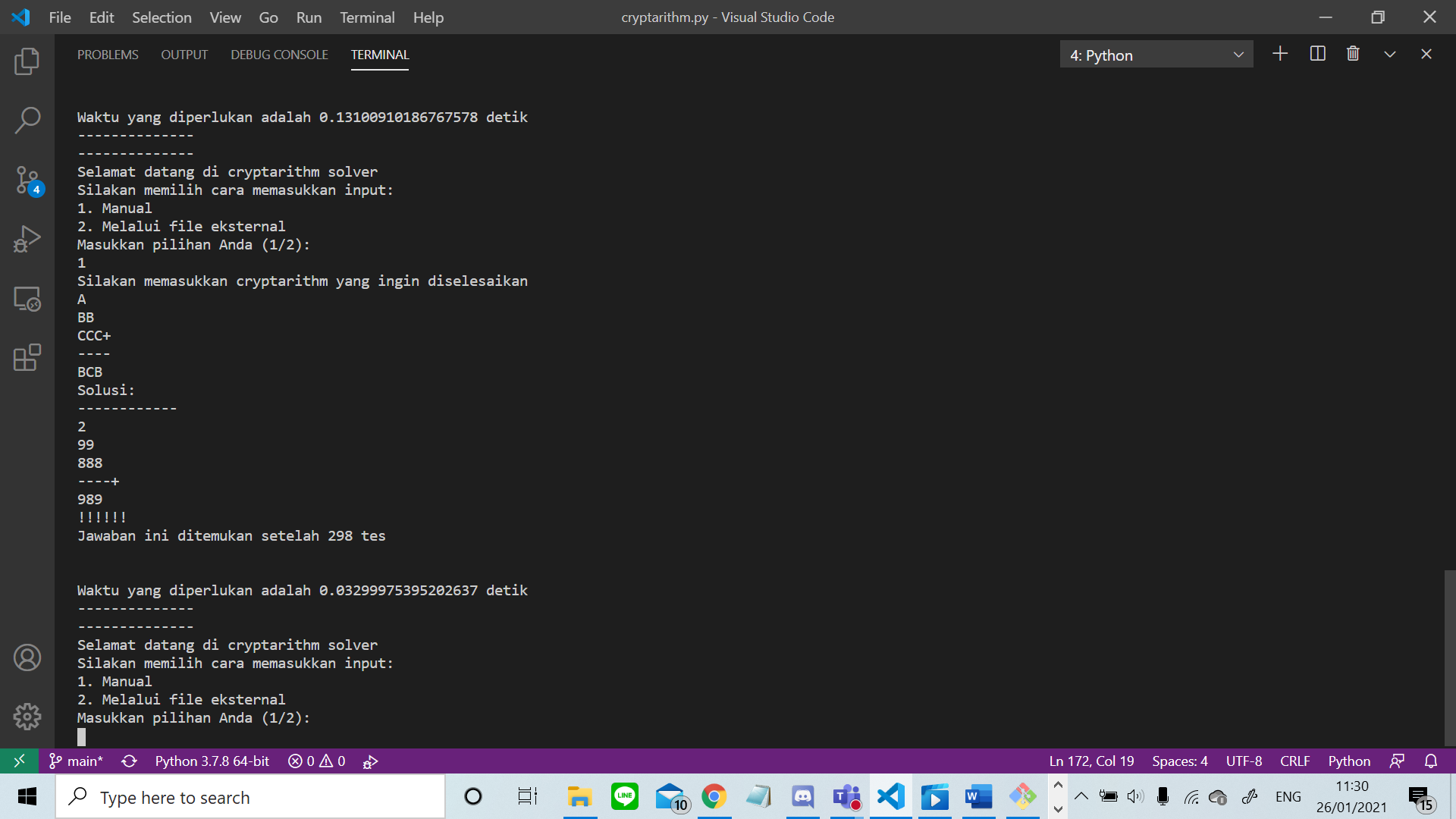
Screenshoot Hasil

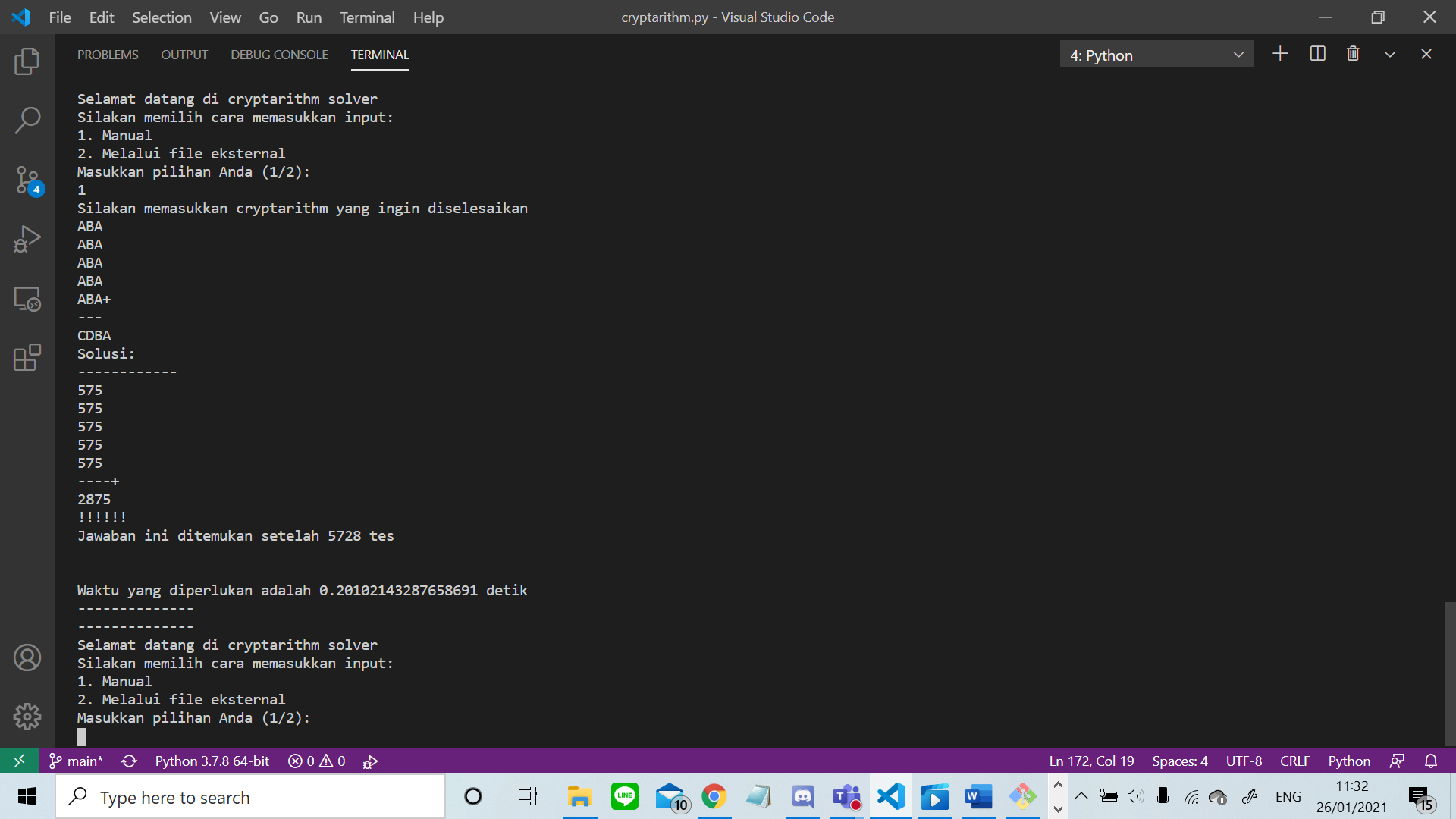
      

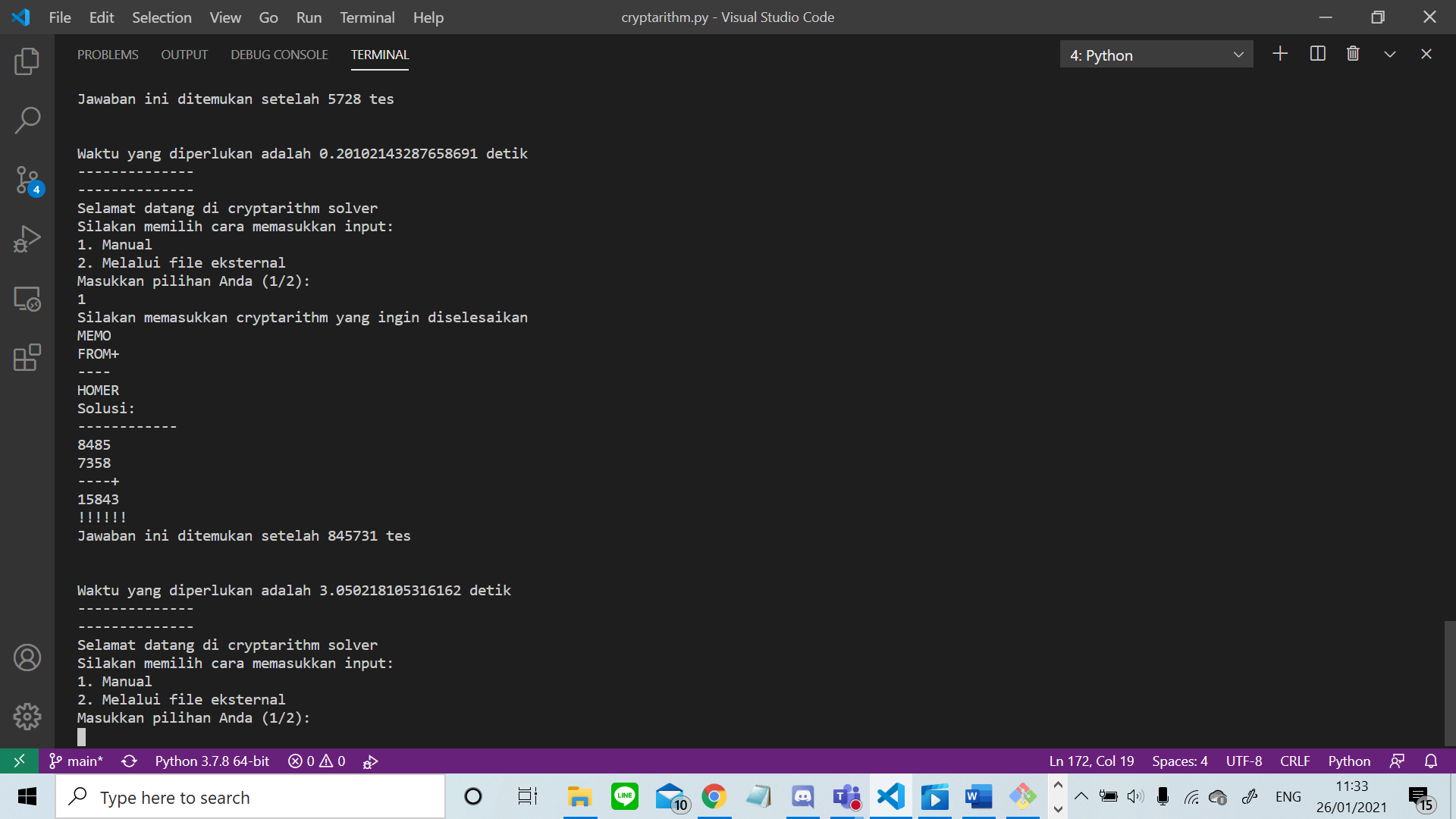
 

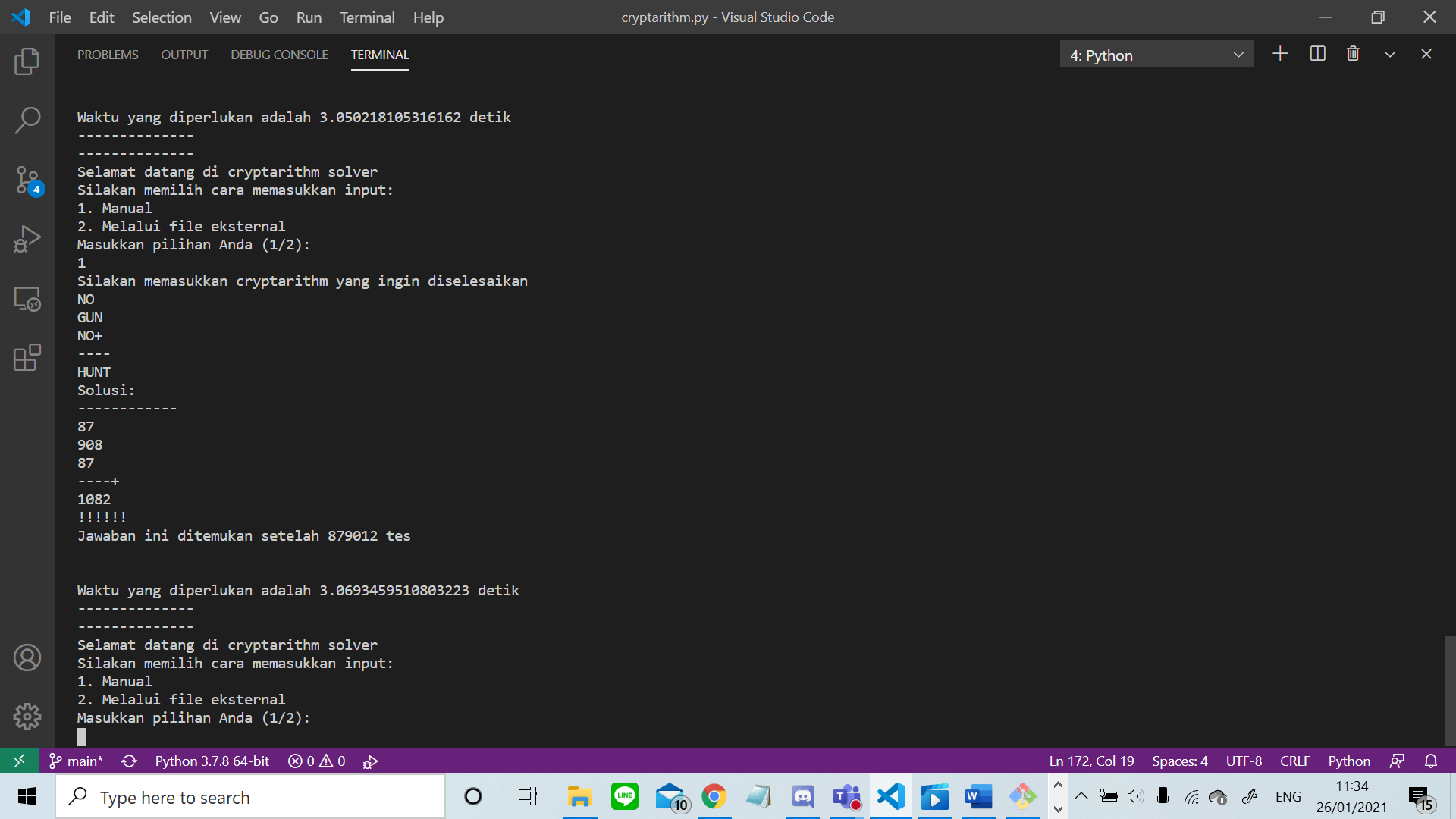
 

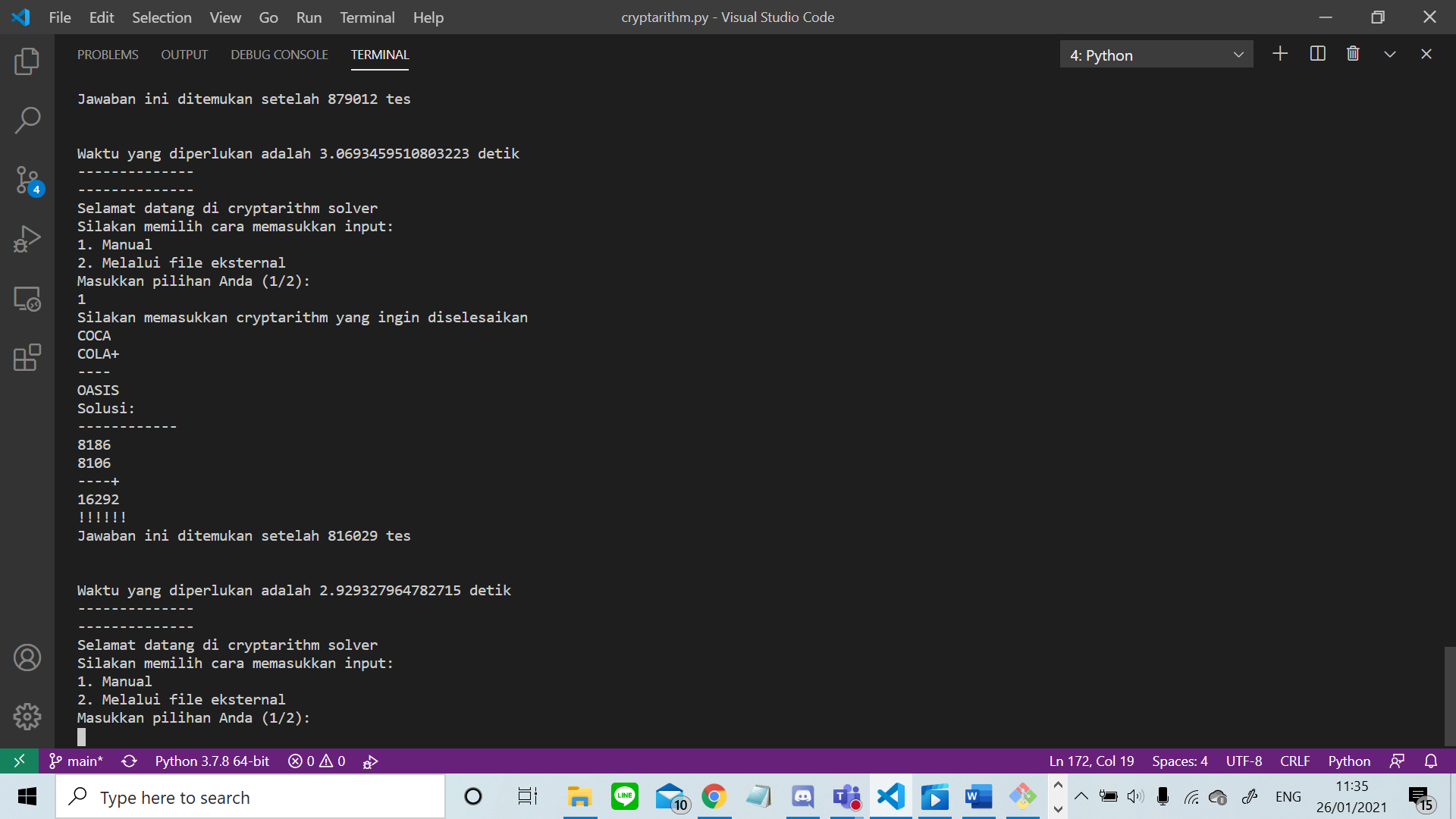


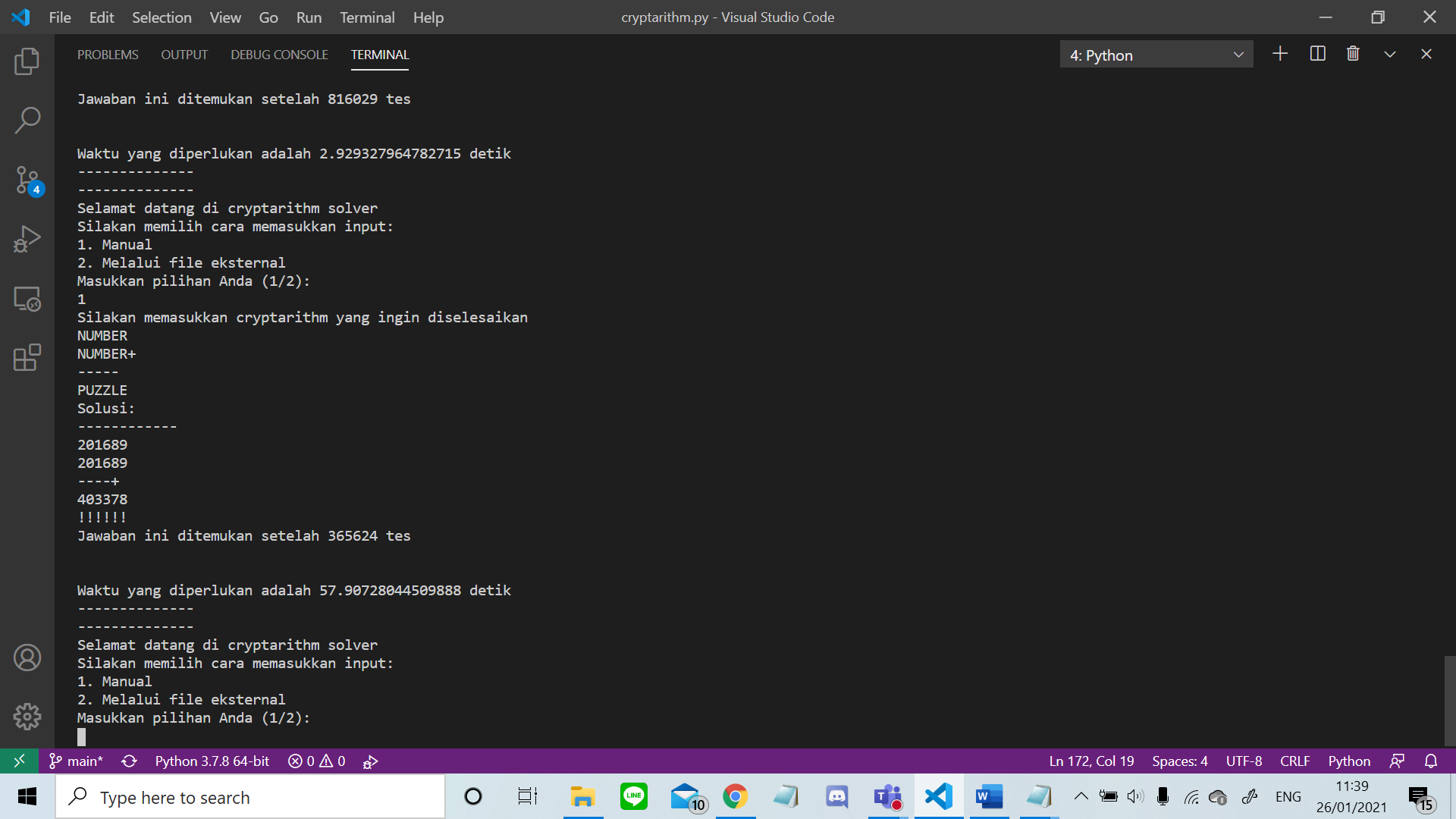




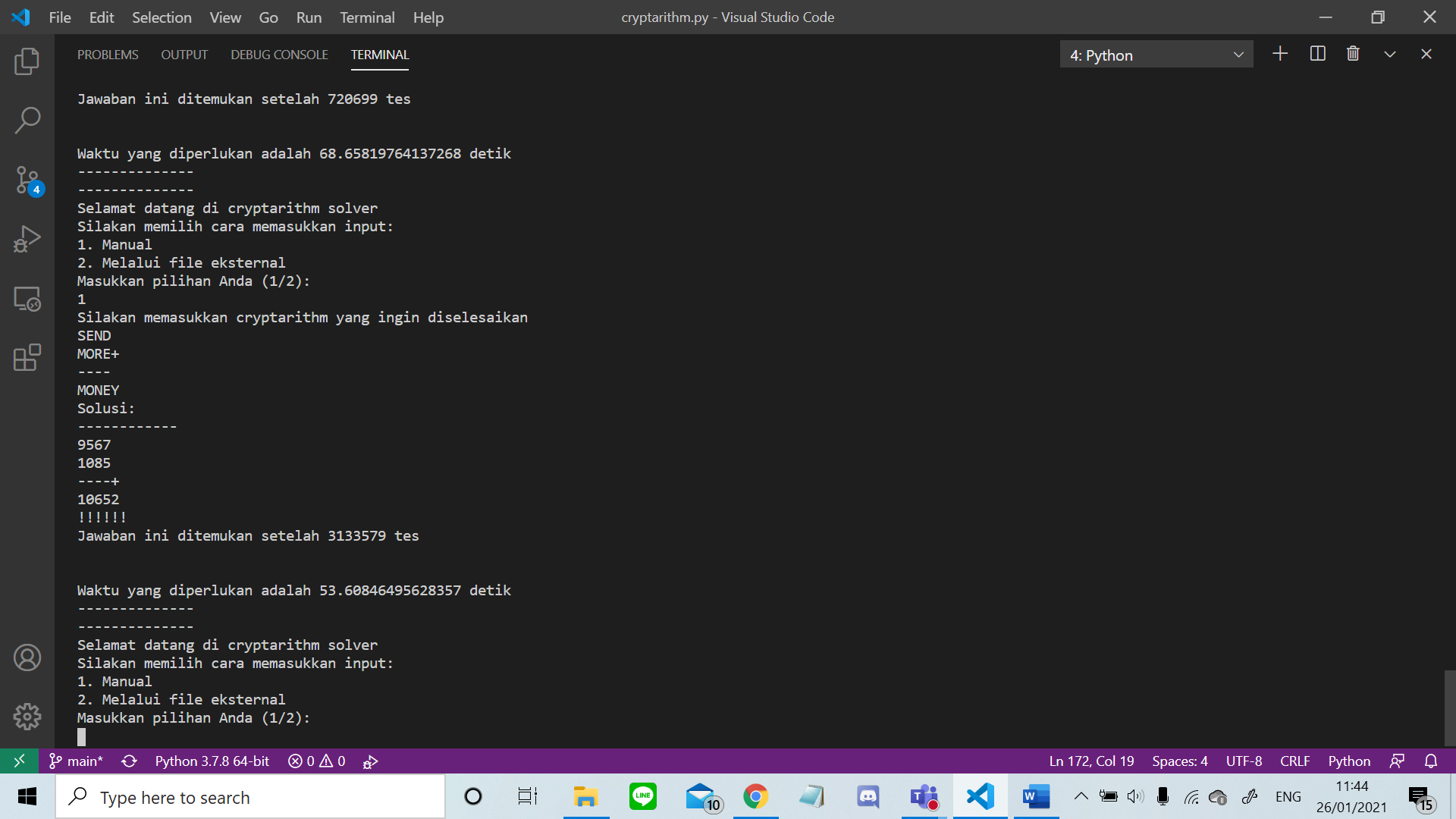


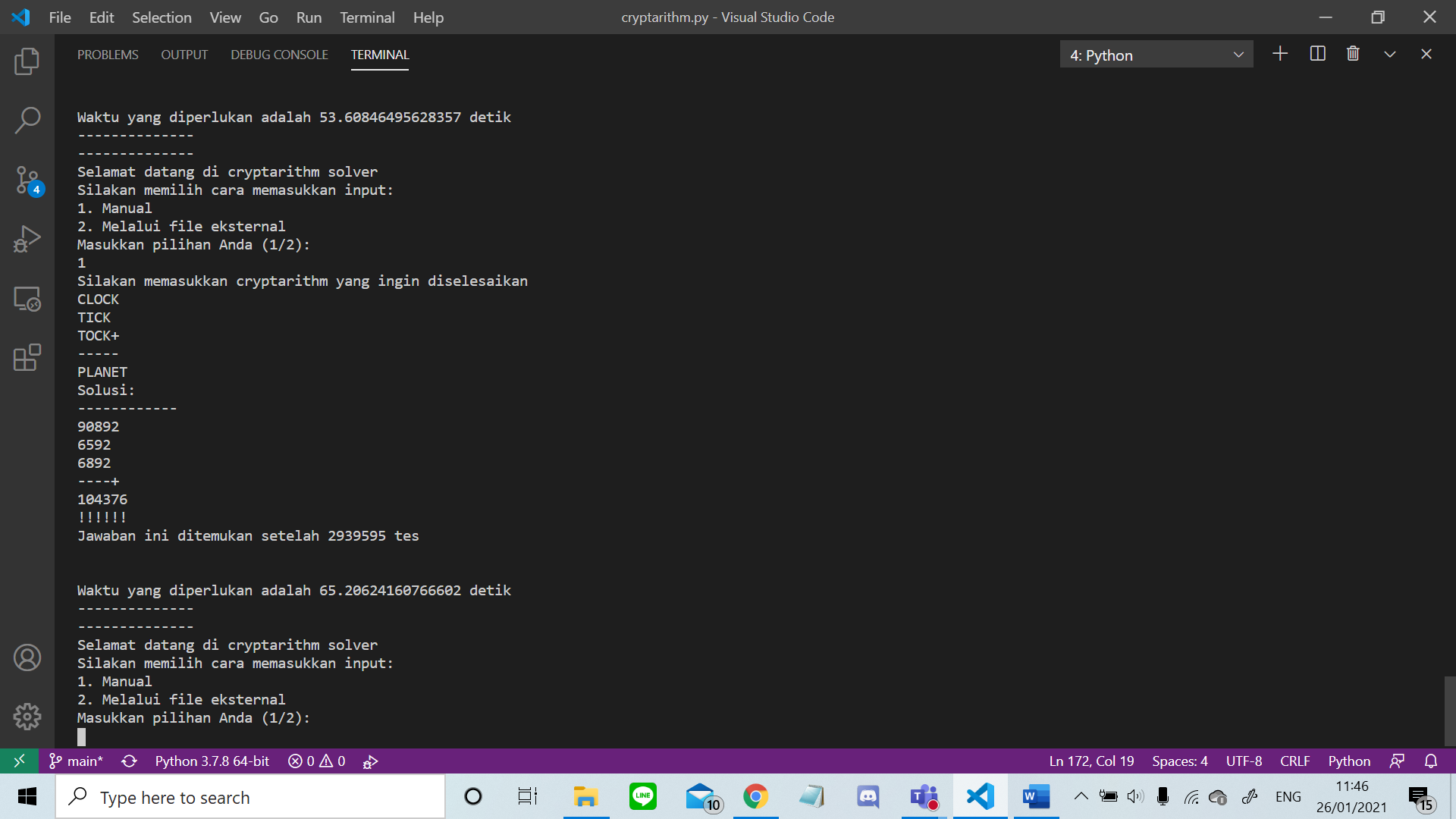


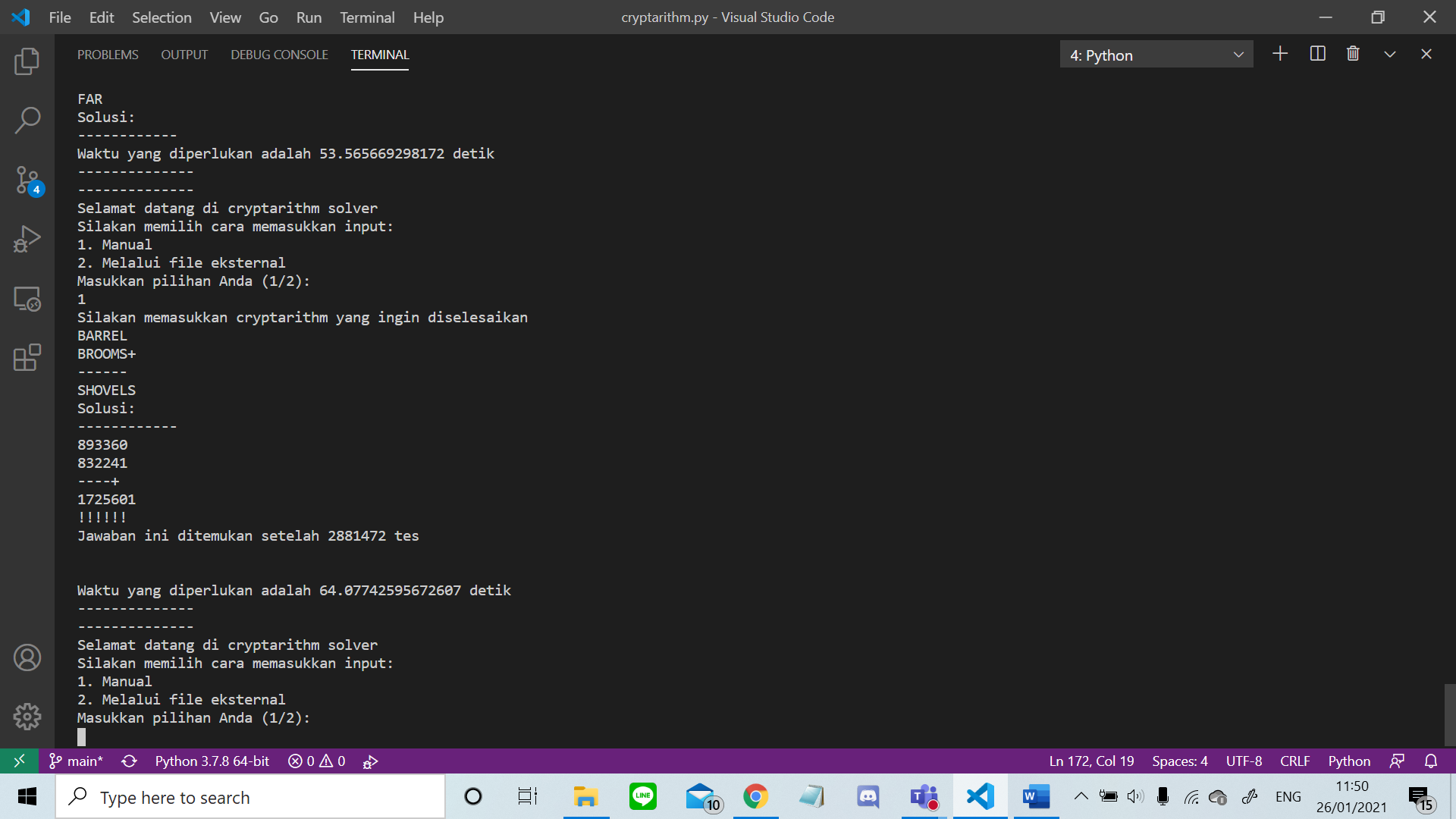


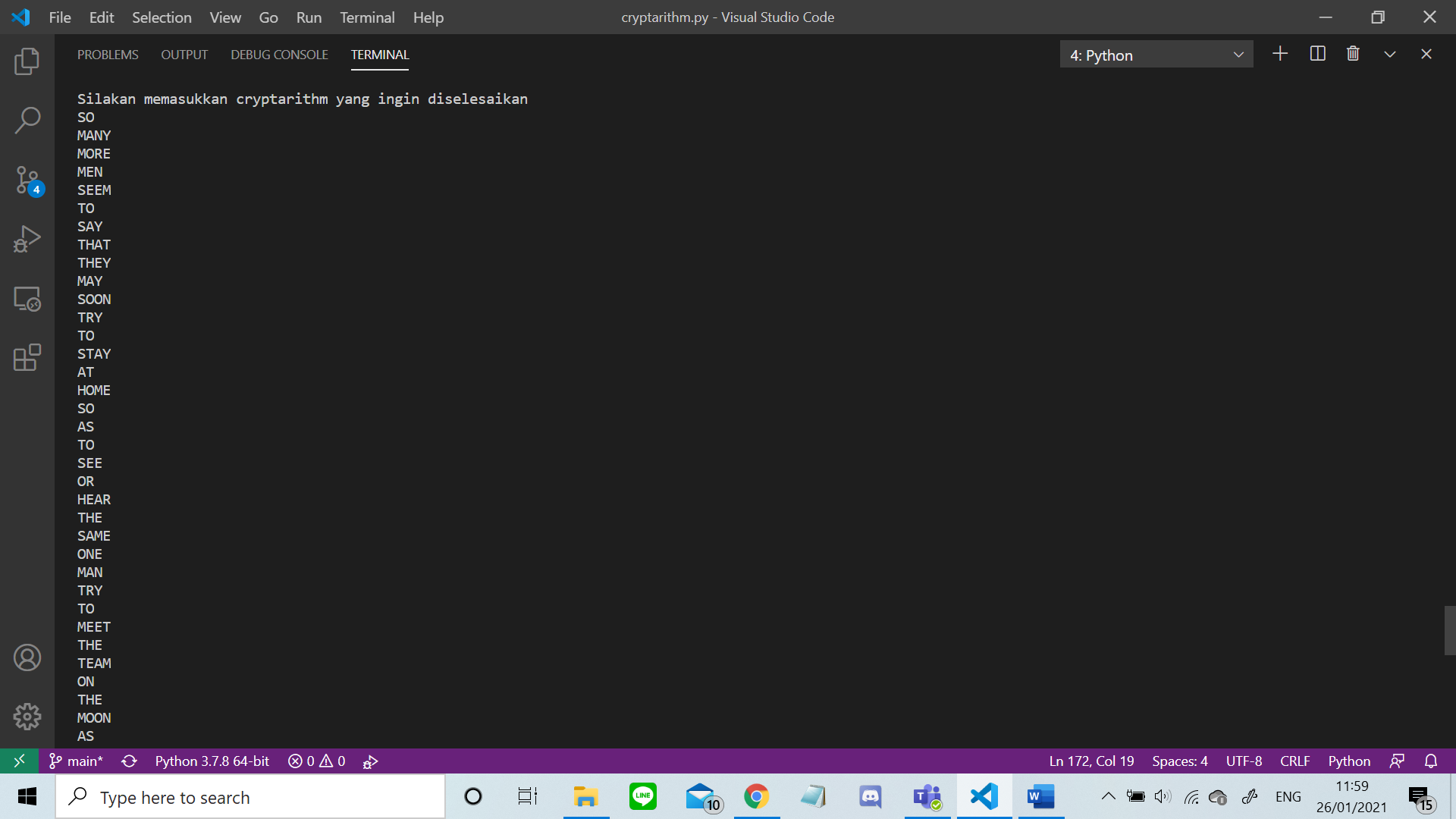
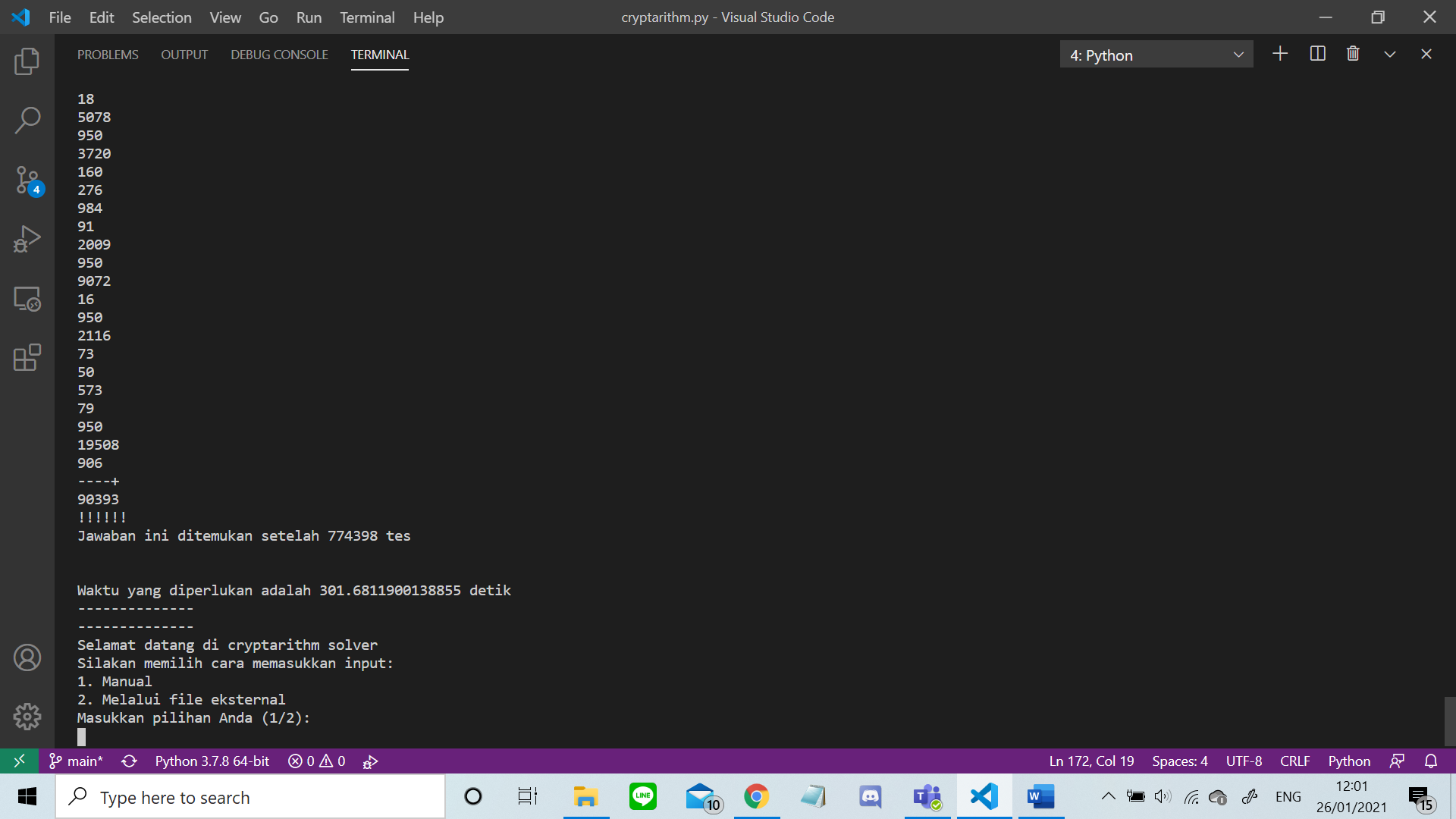










Alamat Kode Program dan Check List

<https://github.com/anakpindahan/Cryptarithm-solver>

