**Tugas Kecil 2 IF2121 Strategi Algoritma**

**Penyelesaian Penyusunan Mata Kuliah dengan Topological Search (Penerapan Decrease and Conquer)**



Kevin Ryan

13519191

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2021

# Bab I

## Algoritma Topological Search dan Pendekatannya terhadap Decrease and Conquer

Bab ini berisi tentang bagaimana penulis mengimplementasikan topological search dalam kode program bahasa Python beserta langkah-langkah program.

1. Program akan dimulai dengan membaca file input terlebih dahulu. User dapat menginput nama file yang ingin diproses oleh algoritma. Jika user ingin menambahkan testcase, maka user dapat menambahkan file .txt di folder input.
2. Program akan memproses file txt tersebut dan menuliskannya di file txt eksternal yang bernama processed\_input.txt . Hal yang diproses pada tahap ini adalah mengganti string “.” dengan string kosong, atau dengan kata lain, menghapus karakter “.”. Setelah pemrosesan, program akan mensetup sebuah list bernama matkul\_arr dan mengisinya dengan list matkul yang dibaca dari file processed\_input.txt dan mengisinya di list yang bernama matkul\_arr .
3. Program akan menginisialisasi variabel not\_done yang berfungsi sebagai kondisi while loop untuk program berjalan, dan menginisialisasi variabel error yang akan digunakan kemudian oleh program sebagai pengecekan apakah input mempunyai solusi.
4. Dalam loop, program akan dimulai dengan menginisialisasi list bernama matkul\_sems\_list dan mengisinya dengan object matkul yang memiliki derajat = 0.
5. Jika matkul\_sems\_list kosong, maka program akan memberikan error message dan break.
6. Jika matkul\_sems\_list tidak kosong, matkul\_sems\_list akan di append ke dalam urutan\_matkul. Catatan : list urutan\_matkul berkemungkinan berisi list.
7. Program akan mengeloop list matkul, dan akan mengurangi derajat seluruh mata kuliah requisit.
8. Program akan menghapus matkul dengan derajat <=0 dari list matkul\_arr
9. Program akan melakukan looping hingga solusi tercapai.
10. Setelah keluar dari loop, program akan mencetak hasil program berdasarkan list urutan\_matkul.

Pendekatan topological search kepada decrease and conquer adalah topological search akan melakukan decrease setiap kali program akan menghapus matkul yang memiliki derajat <= 0 dari list dan akan menyelesaikan persoalan dengan cara mengurangi derajat dari seluruh matkul requisit.

# Bab II

## Source Code

Bagian ini berisi source code program (tanpa comment). Program dengan comment dapat dilihat pada link yang dapat diakses pada Bab IV

**input\_process.py**

def process\_input(file) :

directory = "../input/"+file

with open(directory, 'r') as infile, open("processed\_input.txt", 'w') as outfile :

tmp = infile.read().replace('.', '')

outfile.write(tmp)

**matkul.py**

class matkul:

def \_\_init\_\_(self, name, req):

self.req\_arr = []

self.name = name

self.derajat = len(req)

for char in req :

self.req\_arr.append(char)

**setup.py**

import matkul

def setup():

matkul\_arr = []

with open("processed\_input.txt", 'r') as file:

for line in file :

tmpline = line.strip()

tmp = tmpline.split(',')

for i in range(1, len(tmp)):

tmp[i] = tmp[i].lstrip()

req\_arr = []

if len(tmp)>1 :

for i in range(1, len(tmp)):

req\_arr.append(tmp[i])

tmp\_matkul = matkul.matkul(tmp[0],req\_arr) matkul\_arr.append(tmp\_matkul)

return matkul\_arr

**main.py**

import input\_process

import matkul

import setup

file = str(input("Masukkan nama file : "))

input\_process.process\_input(file)

matkul\_arr = setup.setup()

urutan\_matkul = []

#Initializing variables

not\_done = True

error = False

while not\_done :

ada\_nol = False #Boolean to check if there is one or multiple matkul has the degree of 0

matkul\_sems\_list = [] #List to list all matkul in the semester

array\_of\_derajat = [matkul\_arr[i].derajat for i in range(len(matkul\_arr))] #List of all matkul's derajat (used to prevent confusion of decrementing degrees)

matkul\_sems\_list = [matkul\_arr[i] for i in range(len(matkul\_arr)) if matkul\_arr[i].derajat == 0]

if matkul\_sems\_list == [] :

error = True

print("Tidak bisa disusun")

break

urutan\_matkul.append(matkul\_sems\_list)

for i in range(len(matkul\_arr)):

if (array\_of\_derajat[i] == 0) :

matkul\_arr[i].derajat -= 1

for j in range(len(matkul\_arr)):

if i != j and (matkul\_arr[i].name in matkul\_arr[j].req\_arr) :

matkul\_arr[j].derajat -= 1

matkul\_arr = [matkul\_arr[i] for i in range(len(matkul\_arr)) if matkul\_arr[i].derajat >= 0]

not\_done = False

for i in range(len(matkul\_arr)):

not\_done = not\_done or matkul\_arr[i].derajat >= 0

if not(error) :

index = 1

for listmatkul in urutan\_matkul :

print("Semester %d : " % index)

for i in range(len(listmatkul)-1):

print(listmatkul[i].name, end=", ")

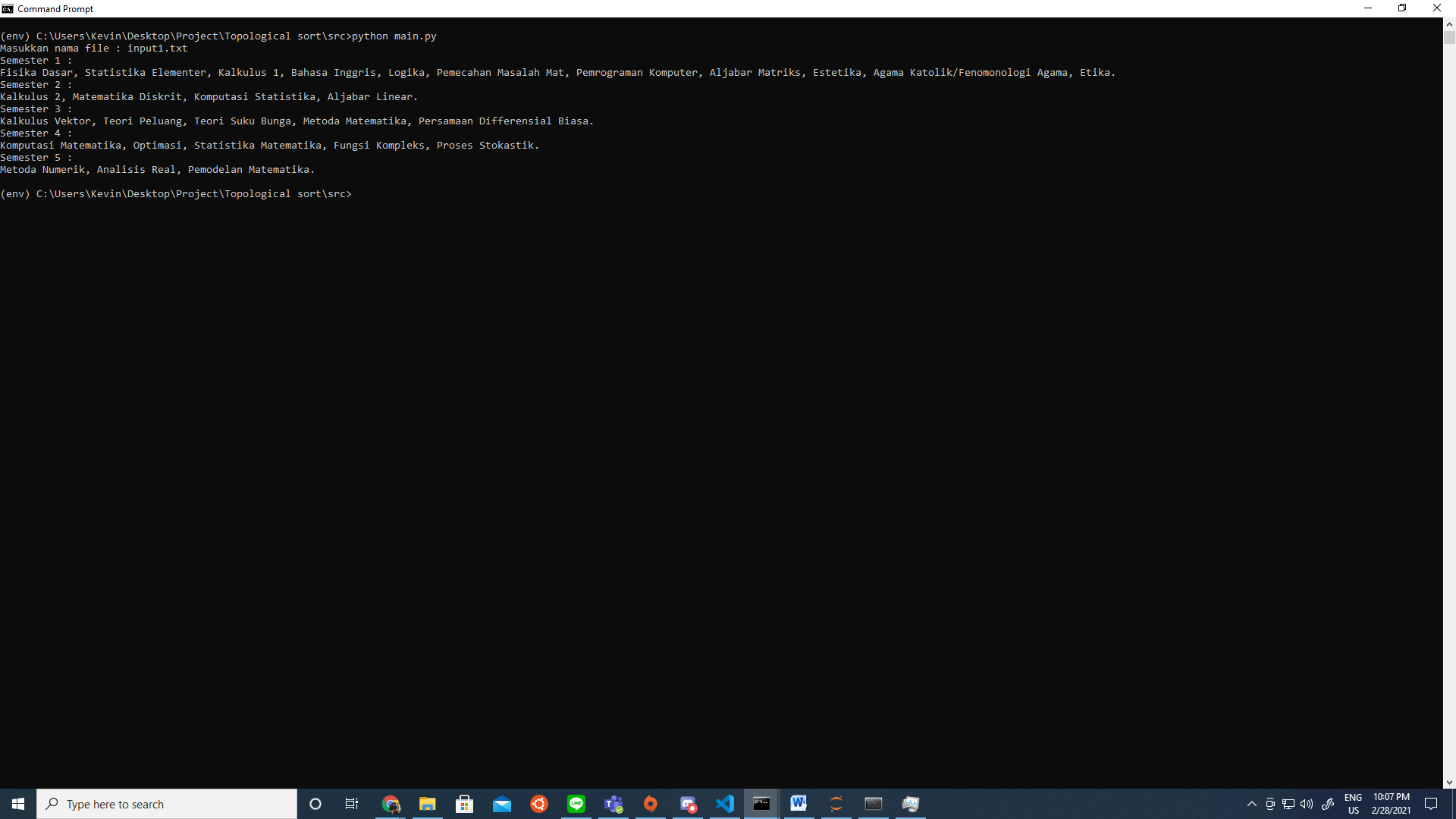
print("%s." % listmatkul[len(listmatkul)-1].name)

index += 1

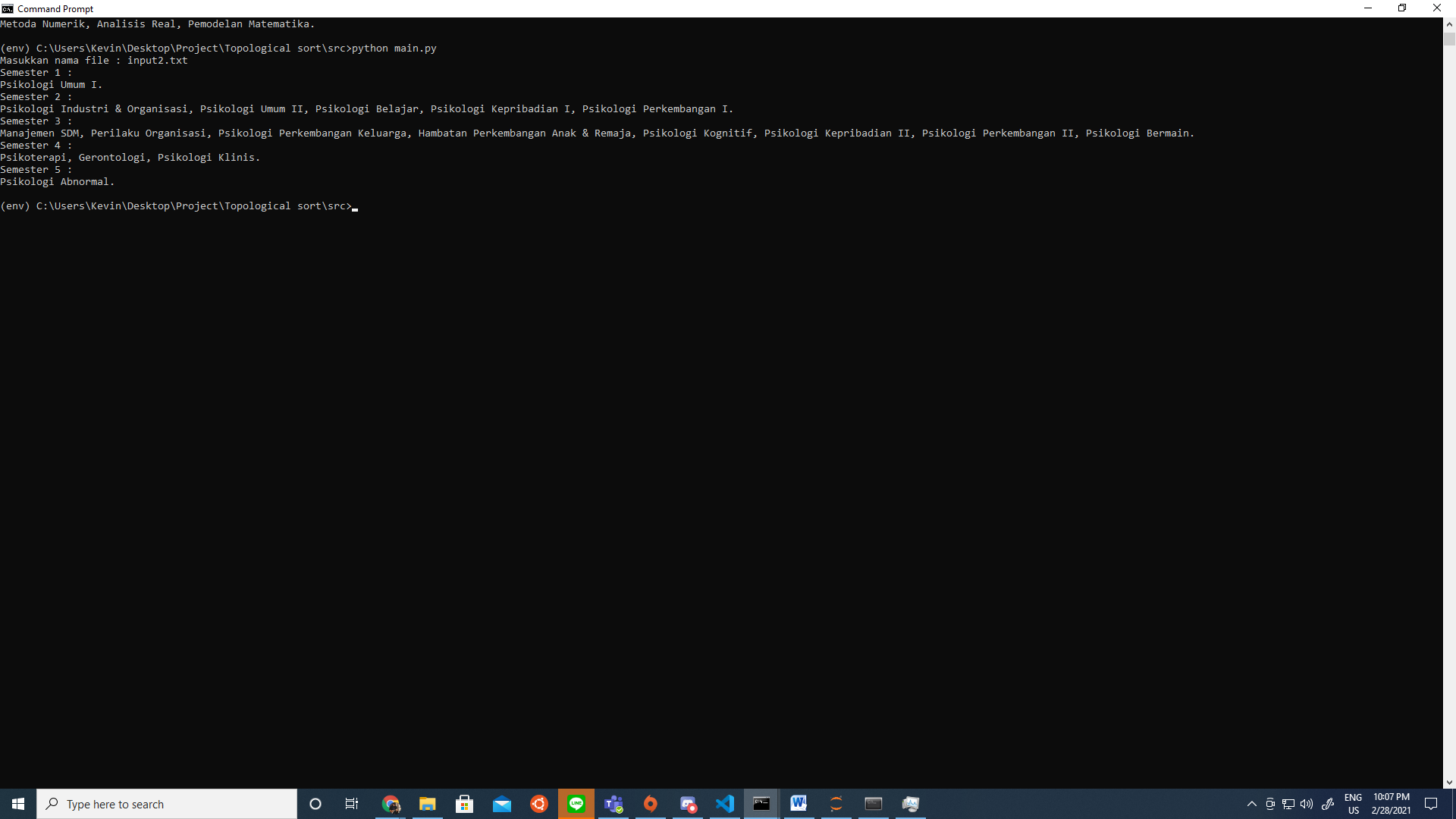
# Bab III

## Screenshots

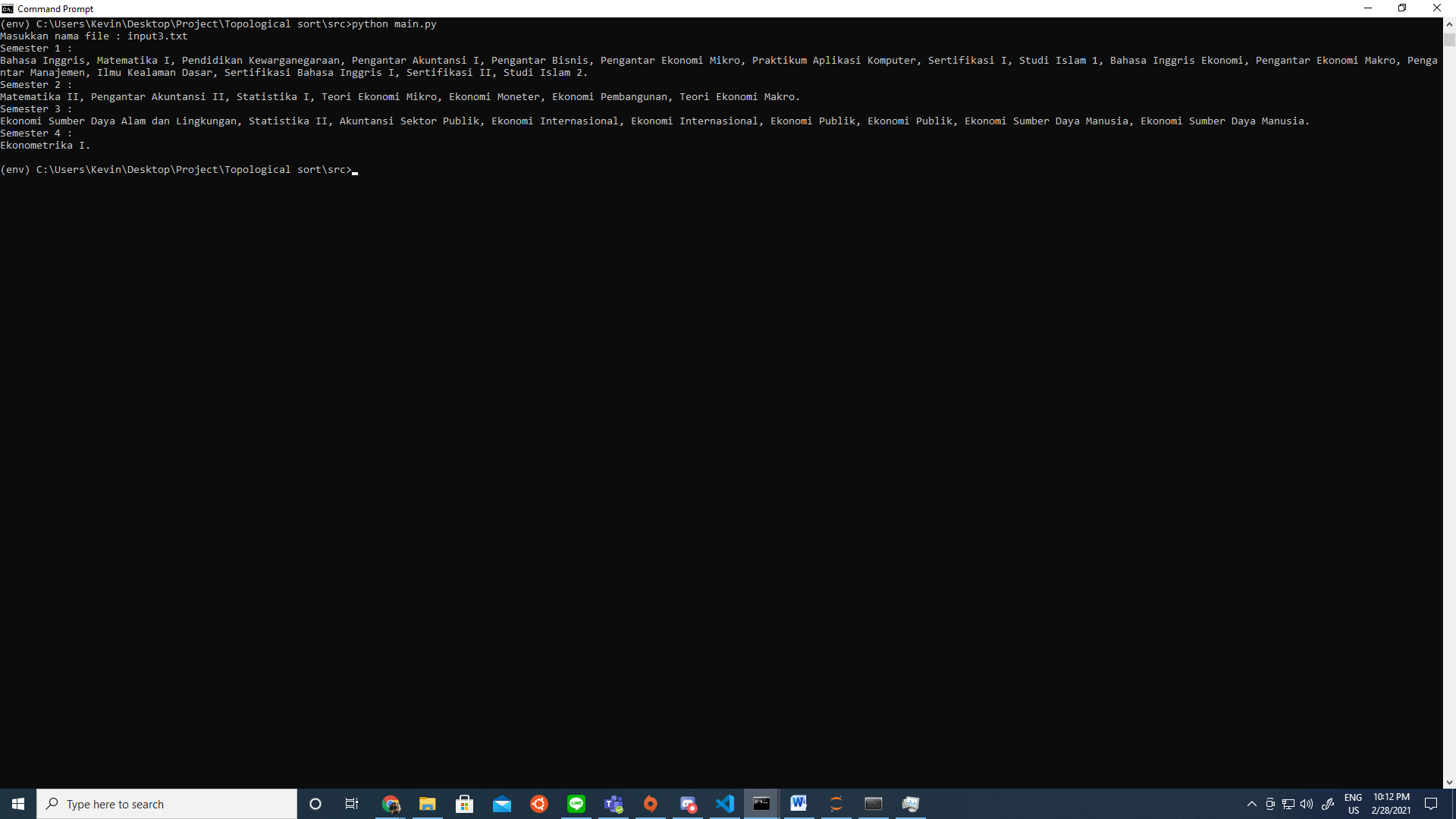
input1.txt



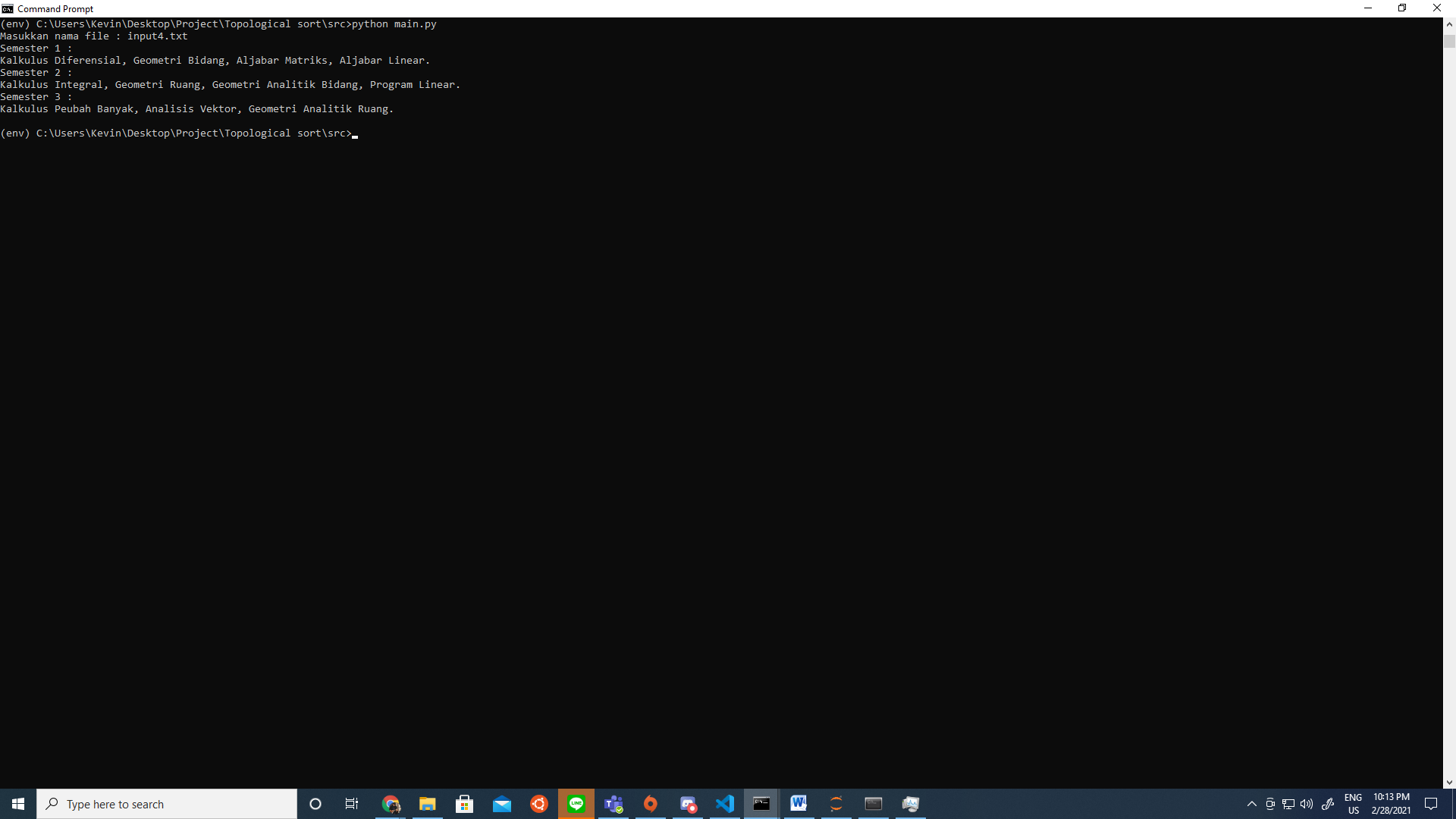
input2.txt



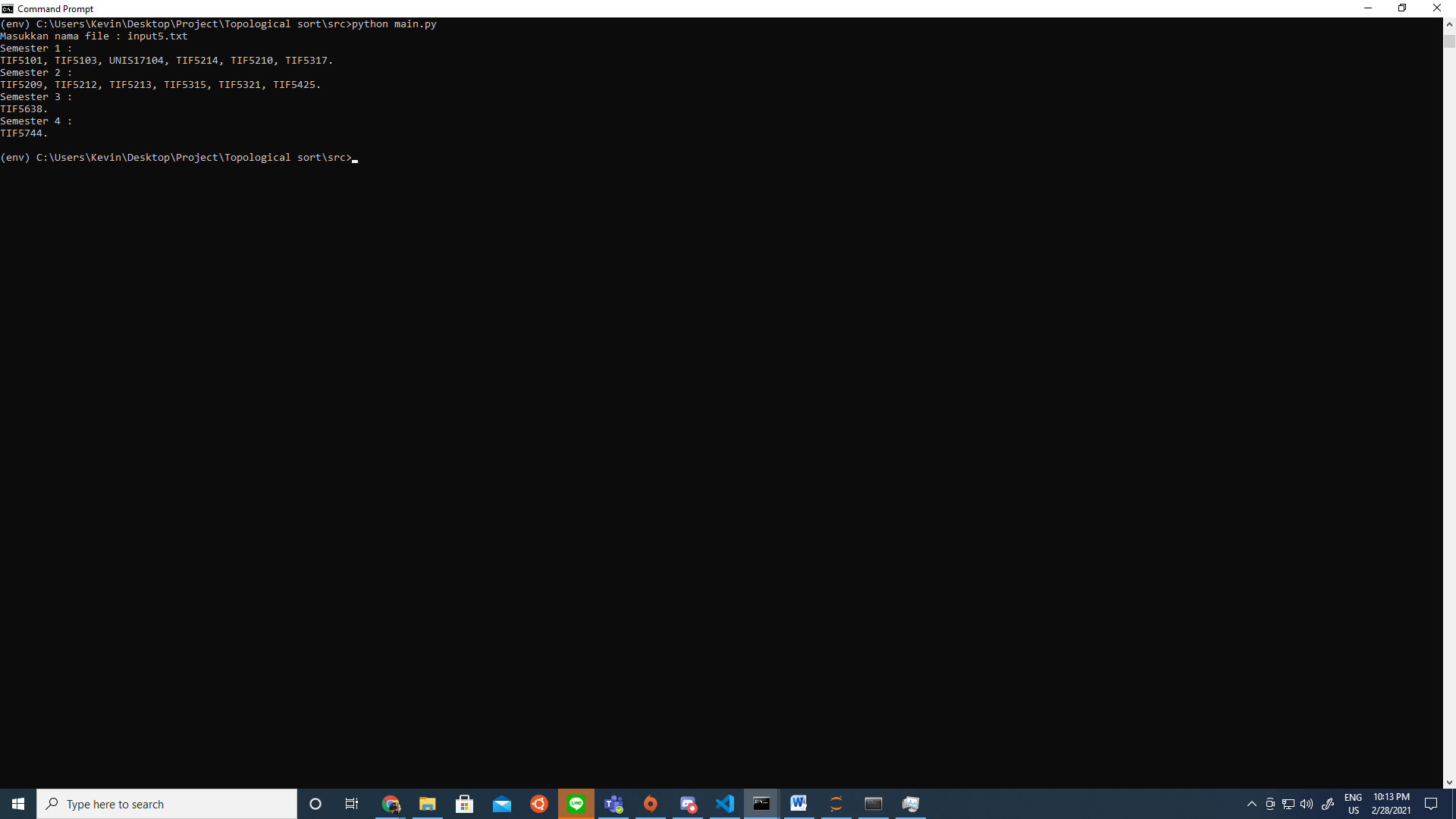
input3.txt



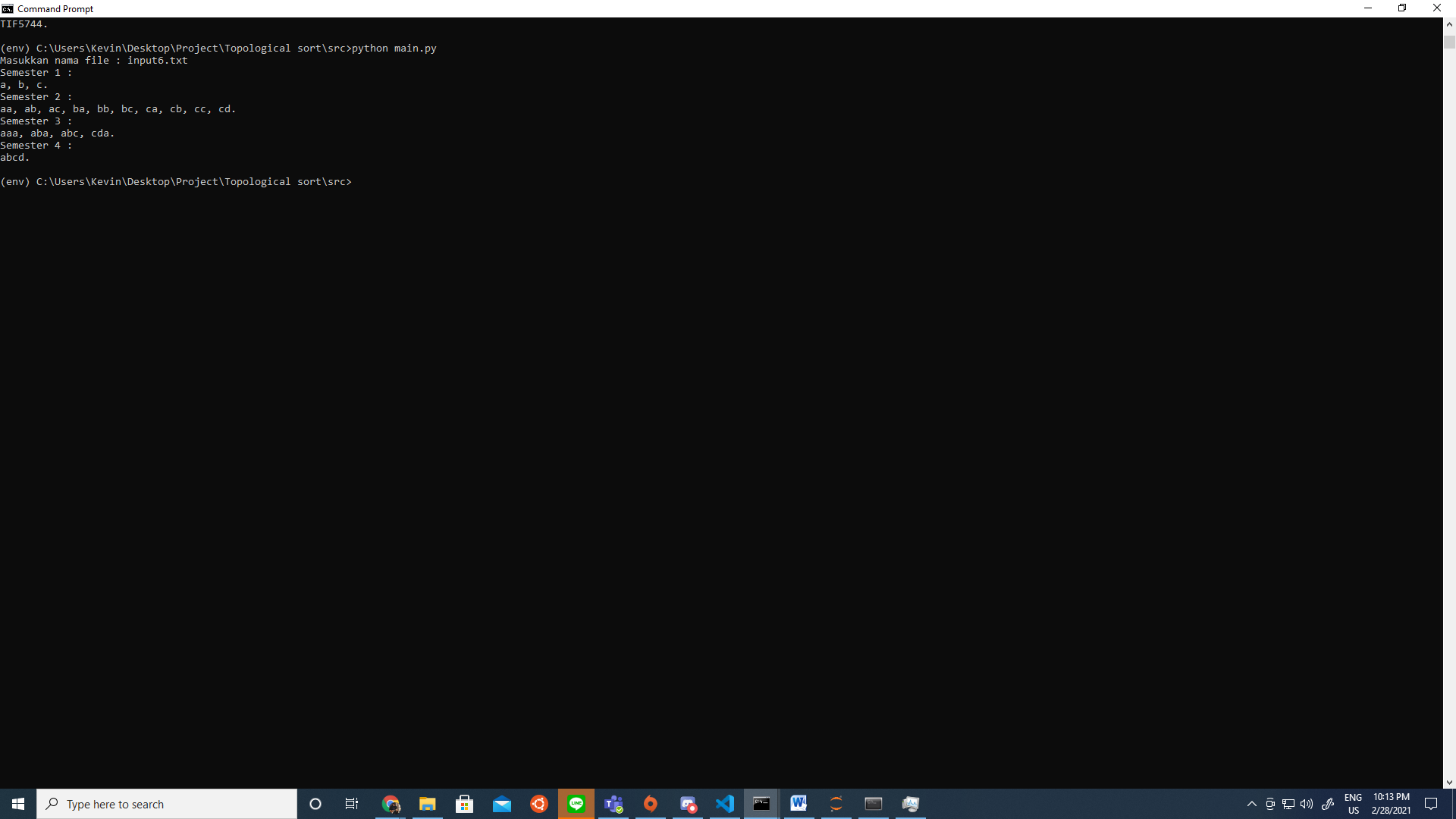
input4.txt



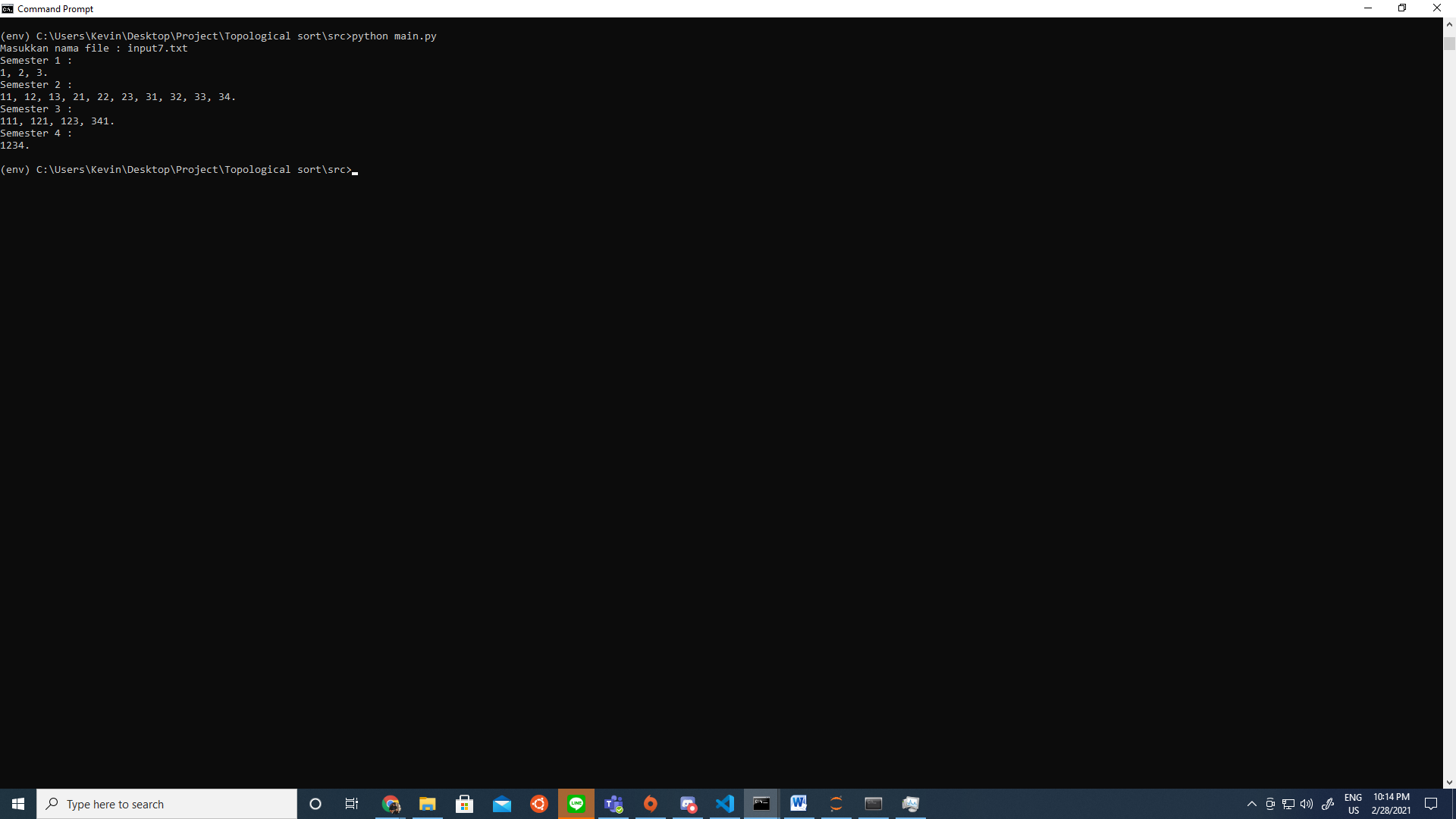
input5.txt



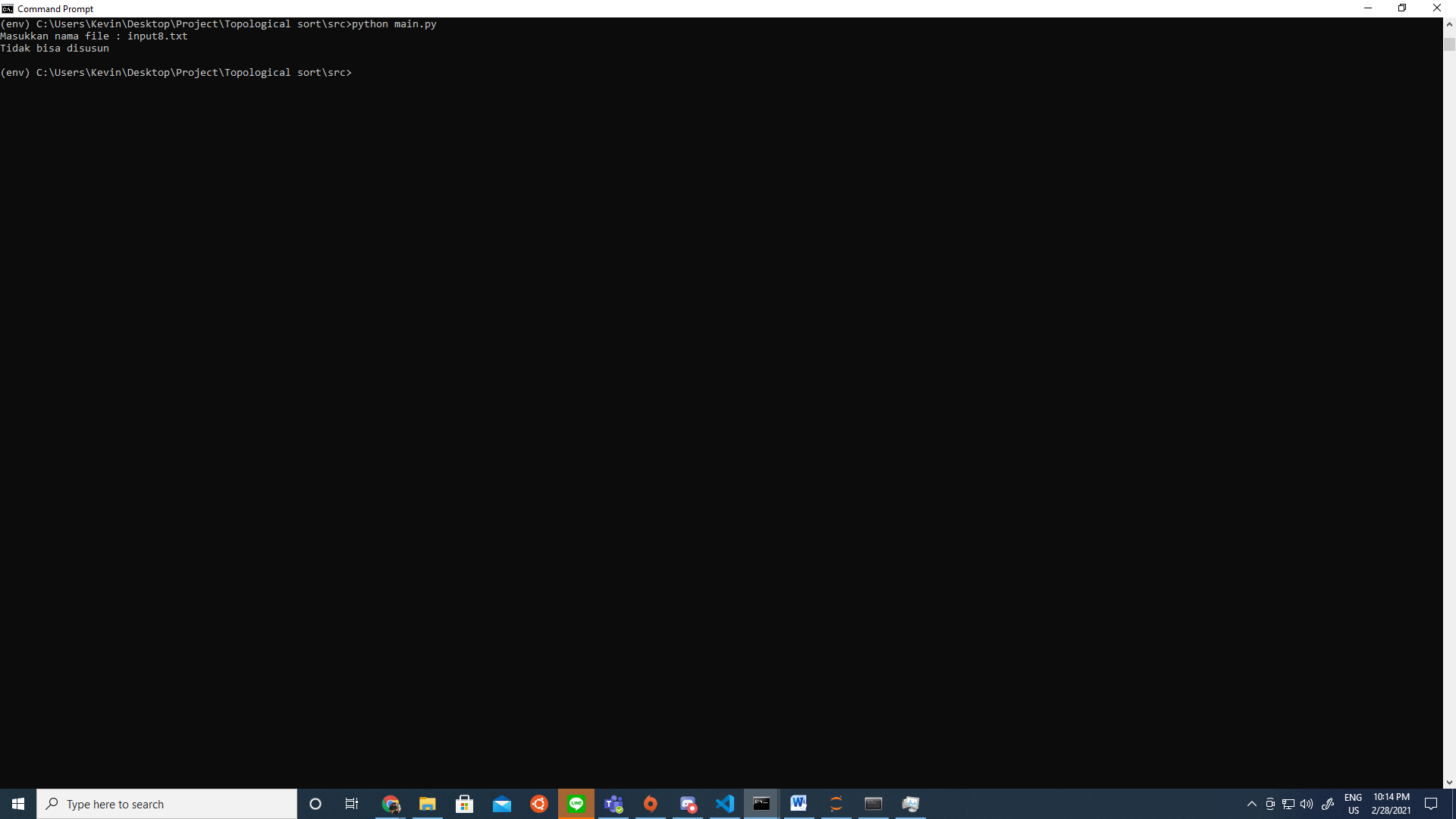
input6.txt



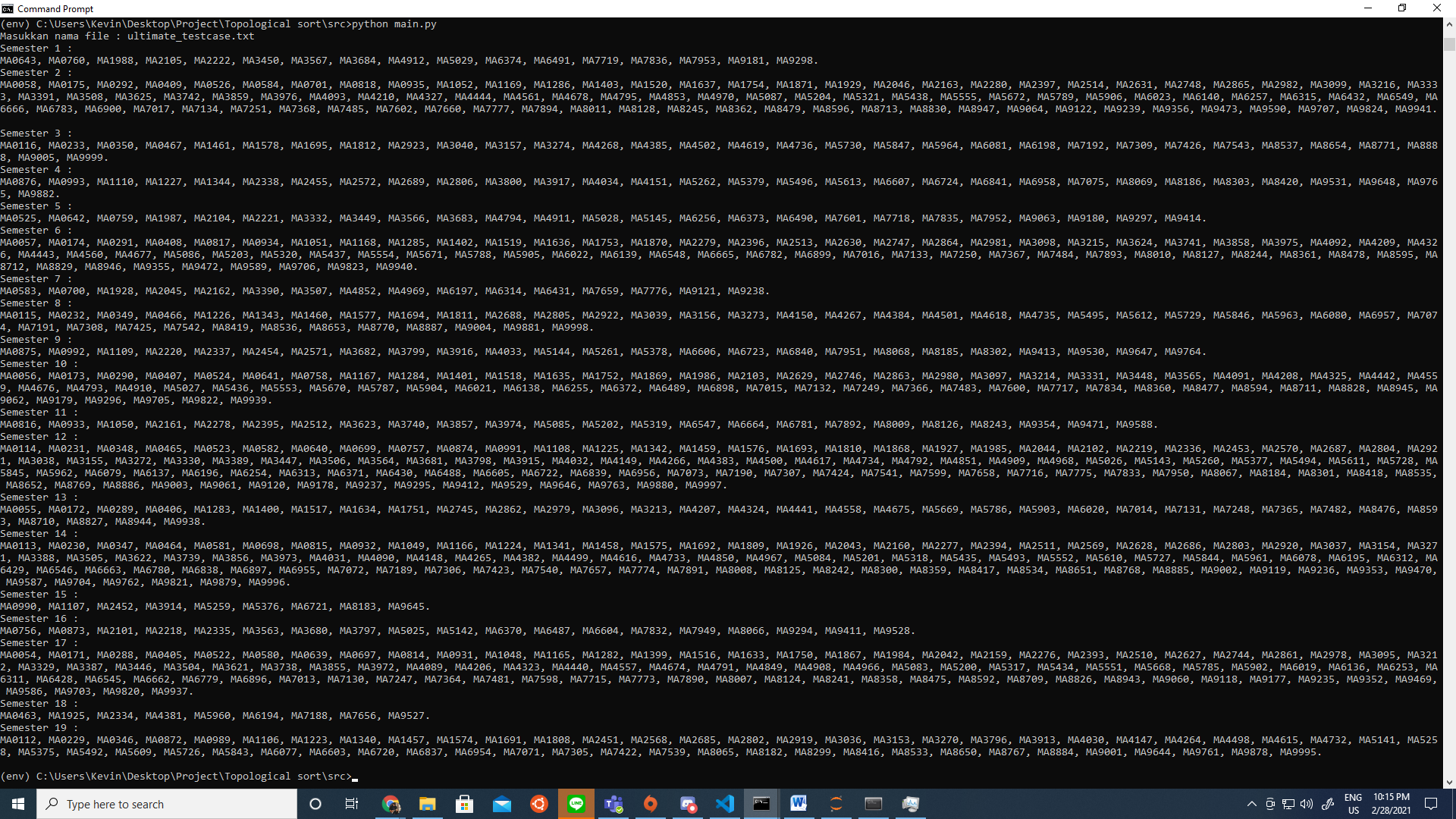
input7.txt



input8.txt (error testcase)



ultimate\_testcase.txt (1000 matkul)



# Bab IV

## Kode Sumber Program

# Bab V

## Tabel Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| No | Poin | Ya | Tidak |
| 1. | Program berhasil dikompilasi | V |  |
| 2. | Program berhasil running | V |  |
| 3. | Program dapat menerima berkas input dan menuliskan output | V |  |
| 4. | Luaran sudah benar untuk semua kasus input | V |  |