**Assignment-6 (Python)**

**1. Write a Python program to read a file line by line and store it into a list.**

Sol.

def file\_read(fname):

with open(fname) as f:

content\_list = f.readlines()

print(content\_list)

file\_read('test.txt')



**2. Write a Python program to read a file line by line store it into an array.**

Sol.

def file\_read(fname):

content\_array = []

with open(fname) as f:

for line in f:

content\_array.append(line)

print(content\_array)

file\_read('test.txt')



**3. Write a Python program to read a random line from a file.**

Sol.

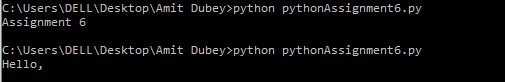
import random

def random\_line(fname):

lines = open(fname).read().splitlines()

return random.choice(lines)

print(random\_line('test.txt'))



**4. Write a Python program to combine each line from first file with the corresponding line in second file**

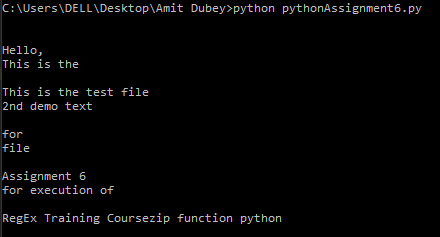
Sol.

print('\n')

with open('test.txt') as fh1, open('test2.txt') as fh2:

for line1, line2 in zip(fh1, fh2):

print(line1+line2)



**5. Write a Python program to generate 26 text files named A.txt, B.txt, and so on up to Z.txt.**

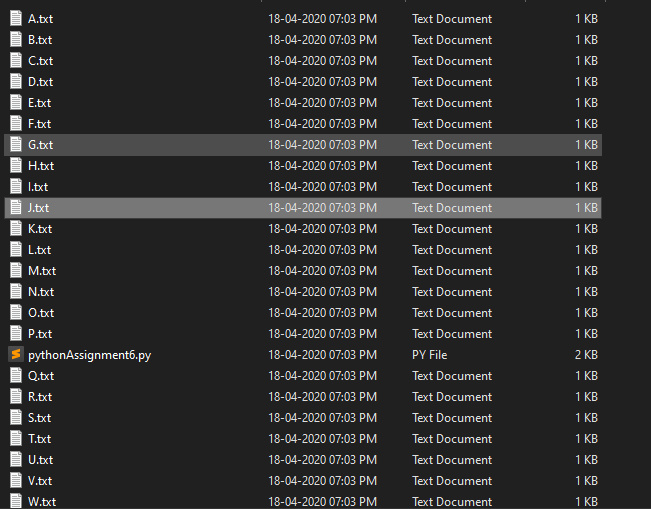
Sol.

import string

for letter in string.ascii\_uppercase:

with open (letter + ".txt", "w") as f:

f.writelines(letter)



**6. Write a Python program to create a file where all letters of English alphabet are listed by specified number of letters on each line.**

Sol.

import string

def letters\_file\_line(n):

with open("words1.txt", "w") as f:

alphabet = string.ascii\_uppercase

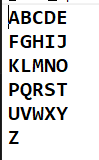
letters = [alphabet[i:i + n] + "\n" for i in range(0, len(alphabet), n)]

f.writelines(letters)

num=int(input("Enter the number: "))

letters\_file\_line(num)





**7. Scrap data from Worldometer example: INDIA Data and run it on live mode Print additionally total number of CORONAVIRUS Cases, Deaths and Recovered.**

**Code:**

import urllib.request

import time

from urllib.request import Request

from bs4 import BeautifulSoup

while True:

data=[]

total\_data=[]

url="https://www.worldometers.info/coronavirus/"

hdr = {'User-Agent': 'Chrome/80.0.3987.163'}

req = Request(url,headers=hdr)

print("Python Script to scrape Worldometer for Data of Covid19 India\n")

r=urllib.request.urlopen(req).read()

soup=BeautifulSoup(r,'lxml')

divs=soup.findAll("div", {"id": "maincounter-wrap"})

for tag in divs:

spanTags = tag.find\_all("span")

for tag in spanTags:

total\_data.append(tag.text)

print("{:>16}TOTAL WORLD DATA\nCORONAVIRUS Cases Total Deaths Total Recovered".format(" "))

print(" {}{:>15}{:>16}".format(total\_data[0],total\_data[1],total\_data[2]))

table=soup.find('table')

table\_rows=table.find\_all('tr')

for tr in table\_rows:

td=tr.find\_all('td')

row=[i.text for i in td]

data.append(row)

india=data[17]

print("\n{:>16}INDIA COVID19 DATA\nCountry Total Cases New Cases Total Deaths New Deaths Total Recovered Active Cases Serious,Critical Tot Cases/1M pop Deaths/1M pop Total Tests Tests/1M pop".format(" "))

print("{} {:>11} {:>10} {:>12} {:>20} {:>8} {:>13} {} {:>28} {:>18} {:>15} {:>10}".format(india[0],india[1],india[2],india[3],india[4],india[5],india[6],india[7],india[8],india[9],india[10],india[11],india[12]))

import urllib.request

from urllib.request import Request

from bs4 import BeautifulSoup

data=[]

total\_data=[]

url="https://www.worldometers.info/coronavirus/"

hdr = {'User-Agent': 'Chrome/80.0.3987.163'}

req = Request(url,headers=hdr)

print("Python Script to scrape Worldometer for Data of Covid19 India\n")

r=urllib.request.urlopen(req).read()

soup=BeautifulSoup(r,'lxml')

divs=soup.findAll("div", {"id": "maincounter-wrap"})

for tag in divs:

spanTags = tag.find\_all("span")

for tag in spanTags:

total\_data.append(tag.text)

print("{:>16}TOTAL WORLD DATA\nCORONAVIRUS Cases Total Deaths Total Recovered".format(" "))

print(" {}{:>15}{:>16}".format(total\_data[0],total\_data[1],total\_data[2]))

table=soup.find('table')

table\_rows=table.find\_all('tr')

for tr in table\_rows:

td=tr.find\_all('td')

row=[i.text for i in td]

data.append(row)

india=data[24]

print("\n{:>16}INDIA COVID19 DATA\nCountry Total Cases New Cases Total Deaths New Deaths Total Recovered Active Cases Serious,Critical Tot Cases/1M pop Deaths/1M pop Total Tests Tests/1M pop".format(" "))

print("{} {:>11} {:>10} {:>12} {:>20} {:>8} {:>13} {} {:>28} {:>18} {:>15} {:>10}".format(india[0],india[1],india[2],india[3],india[4],india[5],india[6],india[7],india[8],india[9],india[10],india[11],india[12]))

time.sleep(5)

