//week 2 solution code:

import re

name=input('Enter filename:')

x=open(name)

sum=0

for line in x:

    line=line.rstrip()

    stuff=re.findall('[0-9]+',line)

    if len(stuff)<1:

        continue

    for nums in stuff:

        num=int(nums)

        sum=sum+num

print(sum)

//week 4(2nd one)

from bs4 import BeautifulSoup

import urllib.request, urllib.parse, urllib.error

import ssl

import re

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

url = input('url:')

rangetill=int(input('enter count='))

position=int(input('enter position='))

#count here

for i in range(rangetill):

    html = urllib.request.urlopen(url, context=ctx).read()

    soup = BeautifulSoup(html, 'html.parser')

    tags = soup('a')

    count = 0

    for tag in tags:

        count = count +1

        if count>position:

            break

        url = tag.get('href', None)

    print(url)

#week 3

import urllib.request, urllib.parse, urllib.error

import xml.etree.ElementTree as ET

import ssl

#Ignore SSL certificate errors

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

count = 0

sum = 0

address = input('Enter location: ')

print('Retrieving ' + address)

data = urllib.request.urlopen(address, context=ctx).read()

tree = ET.fromstring(data)

lst = tree.findall('.//count')

for item in lst:

    count += 1

    sum = sum + int(item.text)

print('Count: ', count)

print('Sum: ', sum)

#week 6 1st assignment

import urllib.request, urllib.parse, urllib.error

import json

import ssl

#Ignore SSL certificate errors

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

count = 0

sum = 0

address = input('Enter location: ')

print('Retrieving ' + address)

data = urllib.request.urlopen(address, context=ctx).read()

info = json.loads(data)

for item in info["comments"]:

    count += 1

    sum += item["count"]

print('Count: ', count)

print('Sum: ', sum)

#week 6 2nd assignment

if api\_key is False:

    api\_key = 42

    serviceurl = 'http://py4e-data.dr-chuck.net/json?'

else :

    serviceurl = 'https://maps.googleapis.com/maps/api/geocode/json?'

#Ignore SSL certificate errors

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

address = input('Enter location: ')

parms = dict()

parms['address'] = address

if api\_key is not False: parms['key'] = api\_key

url = serviceurl + urllib.parse.urlencode(parms)

print('Retrieving', url)

uh = urllib.request.urlopen(url, context=ctx)

data = uh.read().decode()

print('Retrieved', len(data), 'characters')

try:

    js = json.loads(data)

except:

    js = None

if not js or 'status' not in js or js['status'] != 'OK':

    print('==== Failure To Retrieve ====')

    print(data)

#print(json.dumps(js, indent=4))

place\_id = js['results'][0]['place\_id']

print('Place ID: ', place\_id)