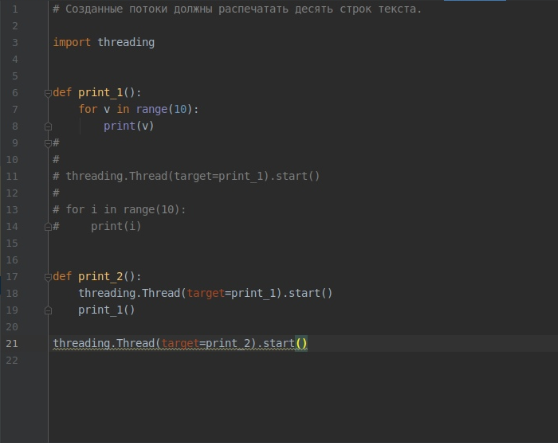
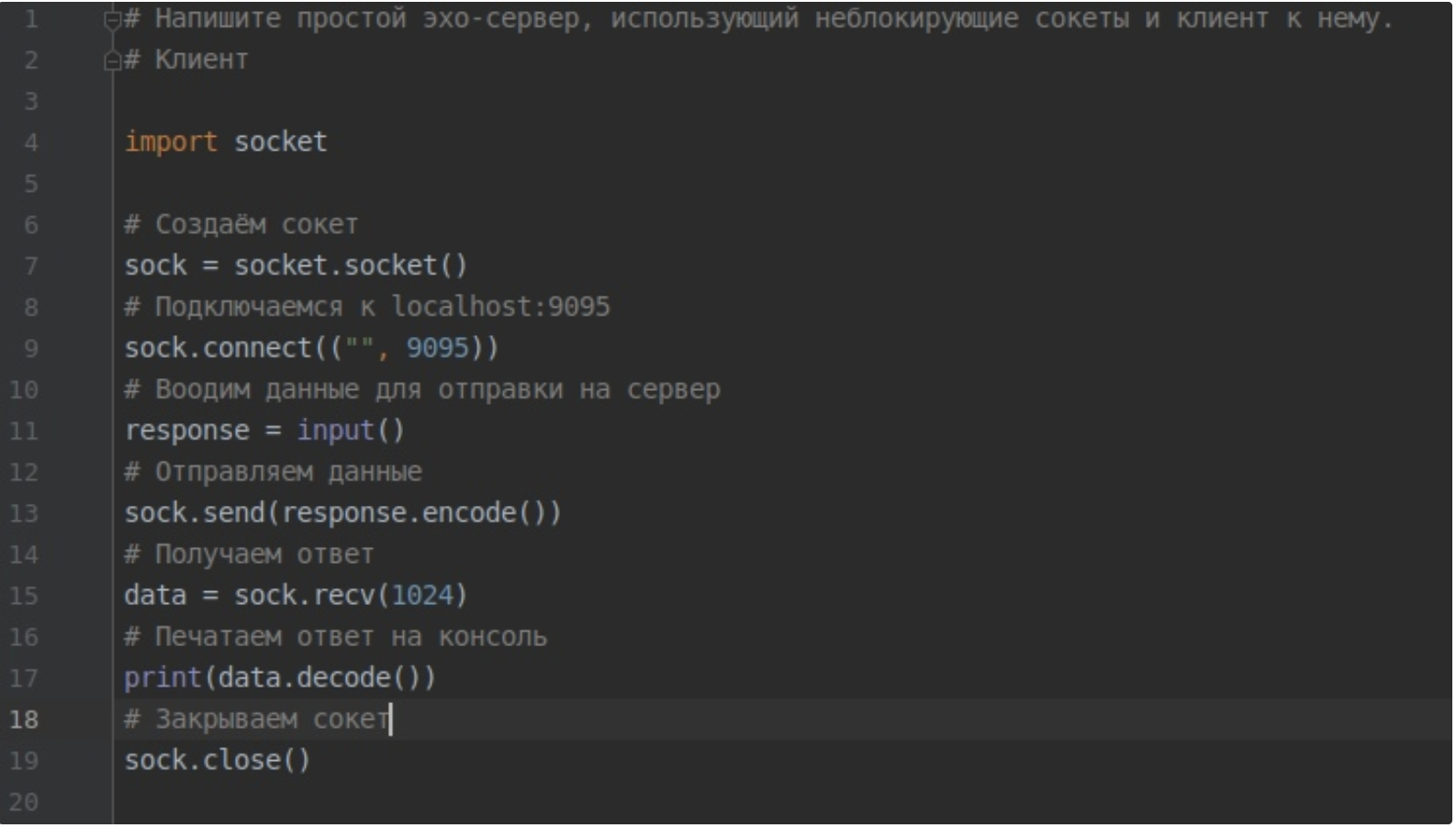
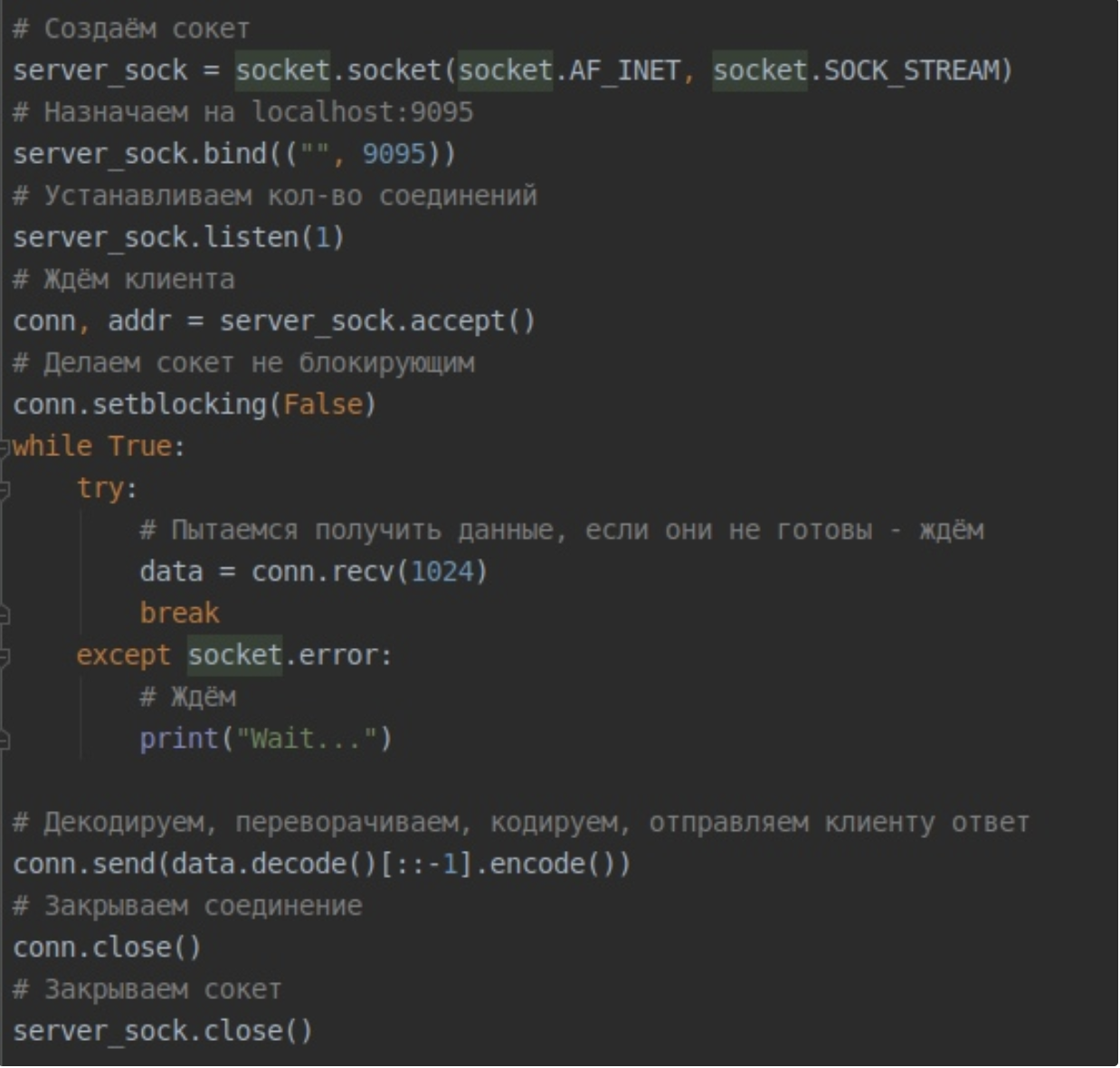
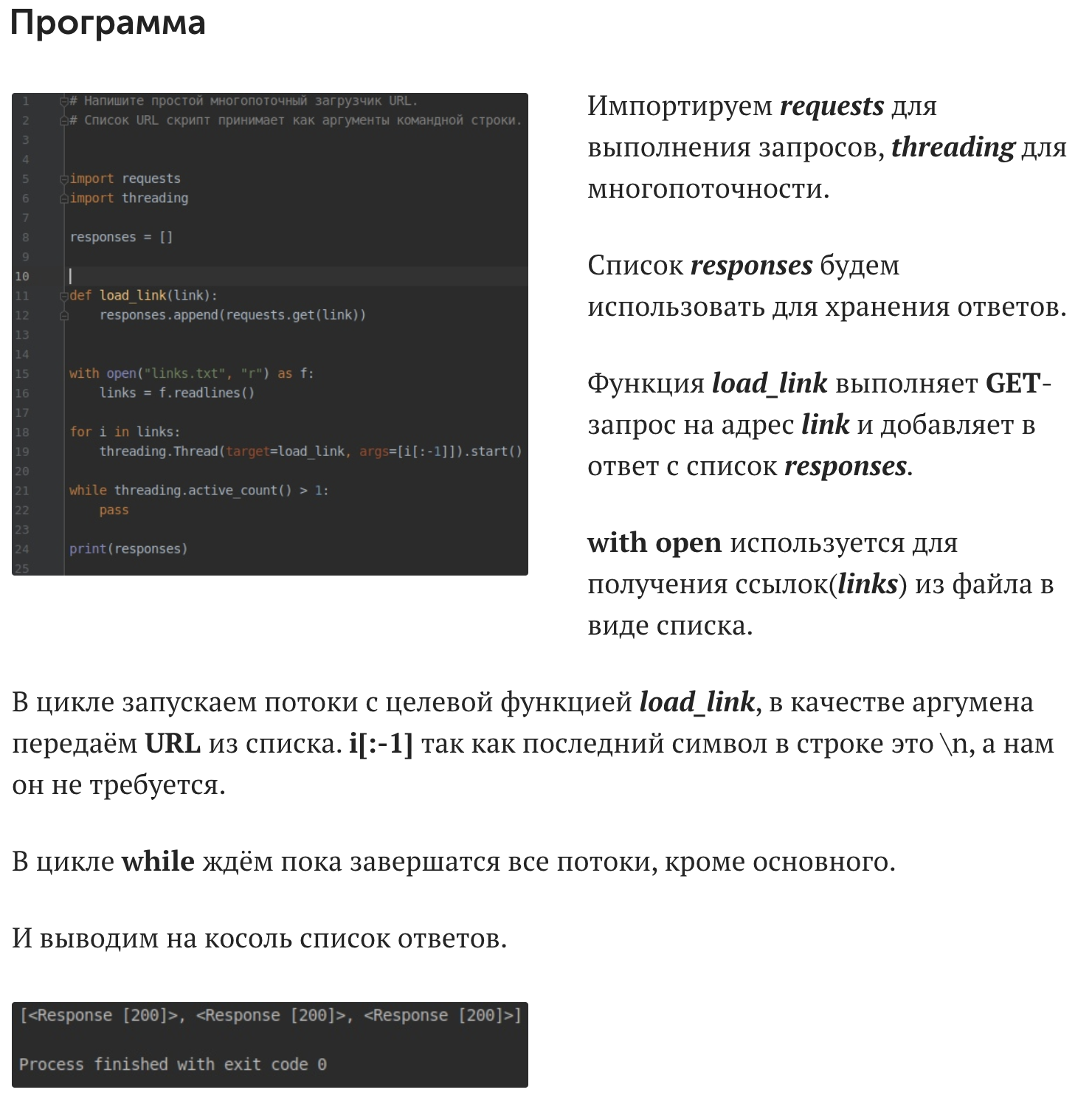
**1. Созданные нити должны распечатать десять строк текста.**

****

**2.Задание: Написать простой эхо-сервер и клиент к нему.**

**3. Написать потоковый загрузчик URL, взятых из файла**

**4. Реализуйте простой HTTP-клиент. Он принимает один параметр командной строки - URL. Клиент делает запрос по указанному URL и выдает тело ответа на терминал как текст.**

import urllib.request

# открываем соединение к URL-адресу с помощью urllib

print('Write URL')

url=input()

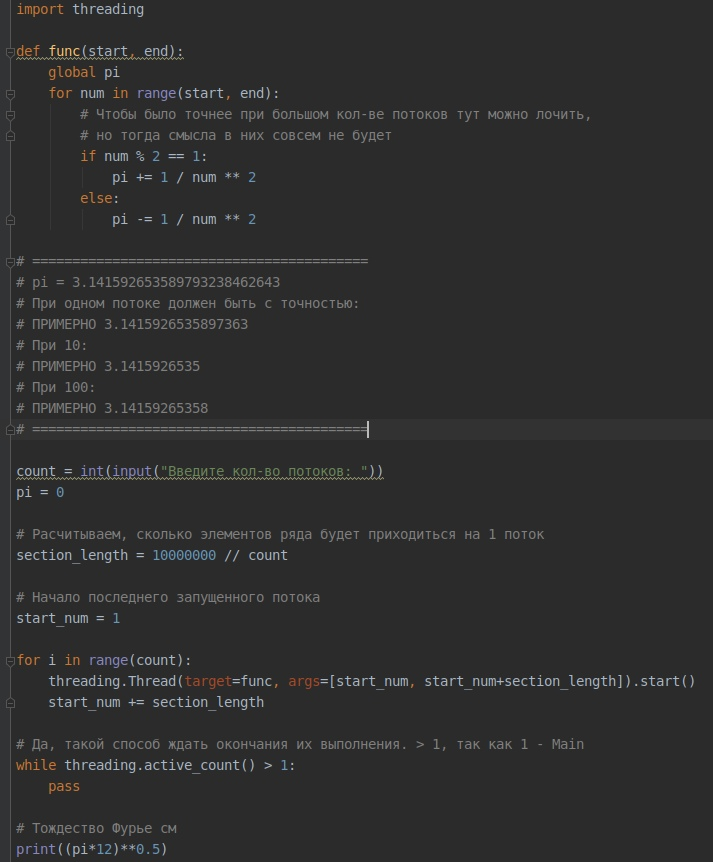
webUrl = urllib.request.urlopen(url)

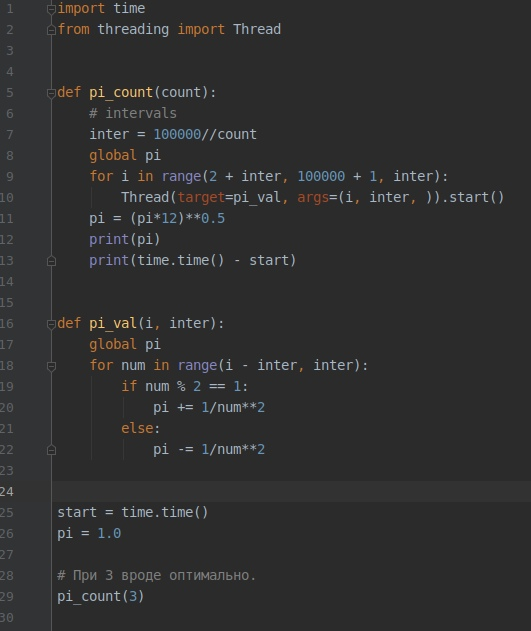
# получаем код результата и выводим его

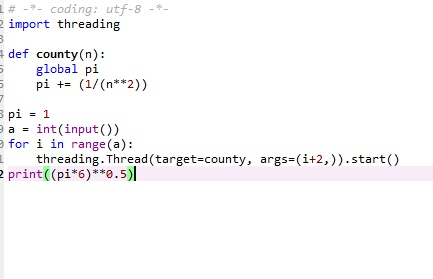
print ("result code: " + str(webUrl.getcode()))# читаем данные с URL-адреса и выводим их

data = webUrl.read()

print (data)

**5. Напишите программу, которая вычисляет число Пи при помощи ряда Эйлера. Количество потоков программы должно определяться параметром командной строки.**

****

****

**6.Дана функция calculate(x, y). Напишите программу, которая создает пул из 5 процессов и распределяет в этом пуле вычисление функции на промежутке х от 0 до 1 с шагом 0,1. у равняется 2 всегда.**

from multiprocessing import Process, Queue,Pool

*#функция calculate была просто взята из головы*

def calculate(x,y=2):

return x+y

processes = []

pool = Pool(processes=5)

x = [i \* 0.1 for i in range(11)]

print(list(pool.map(calculate,x)))

**7.Напишите программу, которая проверяет все числа от 0 на простоту и выводит простые числа на экран по мере нахождения. Числа должны проверяться в различных потоках (или процессах, по выбору студента) Программа должна работать до тех пор, пока ее не остановит пользователь.**

*from queue import Queue*

*import threading*

*from os import abort*

*def easy(thread\_num):*

*while work:*

*t = True*

*if not que.empty():*

*x = int(que.get())*

*for i in range(2, x):*

*if x%i == 0:*

*t = False*

*break*

*if t:*

*print(f'Число {x} простое! {thread\_num}')*

*def put():*

*x = 0*

*while work:*

*que.put(x)*

*x += 1*

*que = Queue(10)*

*threads = []*

*work = True*

*put\_thread = threading.Thread(target=put)*

*put\_thread.start()*

*for i in range(10):*

*n\_thread = threading.Thread(target=easy, args=[i], name=f"thread {i}")*

*n\_thread.start()*

*threads.append(n\_thread)*

*c = input()*

*work = False*

*abort()*

**8. Напишите программу, которая обходит все файлы в директории, переданной ей как параметр и выводит на экран имена тех, чей размер задан как второй параметр. Реализовать рекурсивный обход поддиректорий.**

import os

def cuc(dirname, m):

tree = os.walk(dirname)

for i in tree:

path = i[0]

if len(i[2])>0:

for i in i[2]:

p = os.getcwd()

if os.path.getsize(p+'/'+path+'/'+i) == m:

print(i)

a = cuc('test',0)

без использования os.walk

def coc(dirname, m):

f = []

d = []

pa = os.path.abspath(dirname)

for i in os.listdir(pa):

if os.path.isfile(os.path.abspath(dirname)+'/'+i) == True:

f.append(i)

else:

d.append(i)

for i in f:

if os.path.getsize(dirname+'/'+i) == m:

print(i)

if len(d) == 0:

return

for i in d:

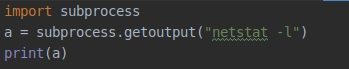
coc(dirname+'/'+i,m)

b = coc('test', 0)

**9. Напишите программу, которая выводит на экран список номеров открытых портов на данной машине. Использовать команду netstat**

**#!/bin/bash**

**netstat** -l



**10. Напишите программу, которая копирует файл с удаленного хоста в текущую папку по SSH. Имя файла и адрес хоста принимать как параметры.**

**#!/bin/bash**

**scp** “**$1**”:”**$2**” ~**/**yourcopy

Под этим понимается, что мы копируем файл (имя которого передается во втором параметре) из удаленного сервера (который передается в первом параметре) в текущую дерикторию локальной машины как файл с именем “yourcopy”

Пример вызова кода из терминала:

sh ./shfname.sh user@192.168.52.4 test

Далее спросят пароль от удаленного сервера.

Касаемо Пятона, сохраняем сие в условный p.py:

**import** os

a = **input**(**‘Введите пользователя и адрес: ’**)

b = **input**(**‘Введите имя файла: ’**)

os.system(f**‘scp {a}:{b} ~/yourcopy’**)

Из терминала вызываем:

python3 p.py

Далее спросят пароль от удаленного сервера.