Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

Национальный исследовательский университет “МИЭТ”

Институт Системной и программной инженерии и информационных технологий

Дисциплина: Проектирование информационных систем

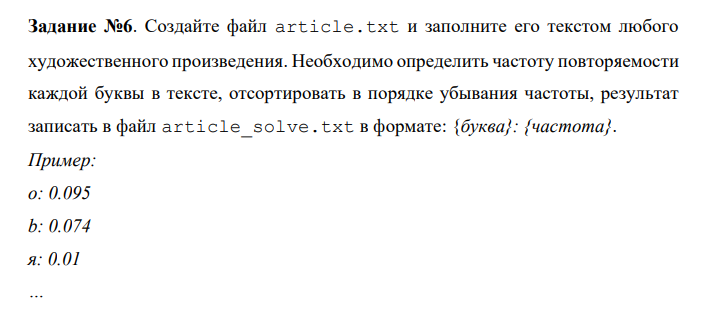
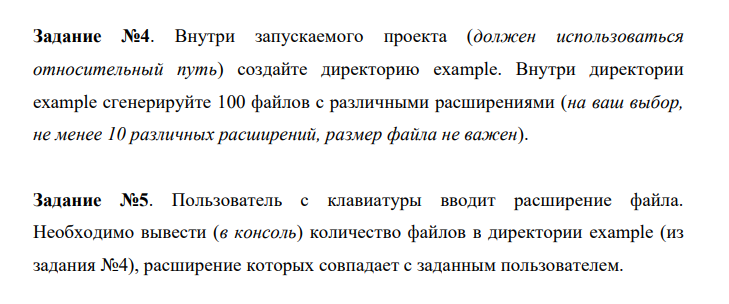
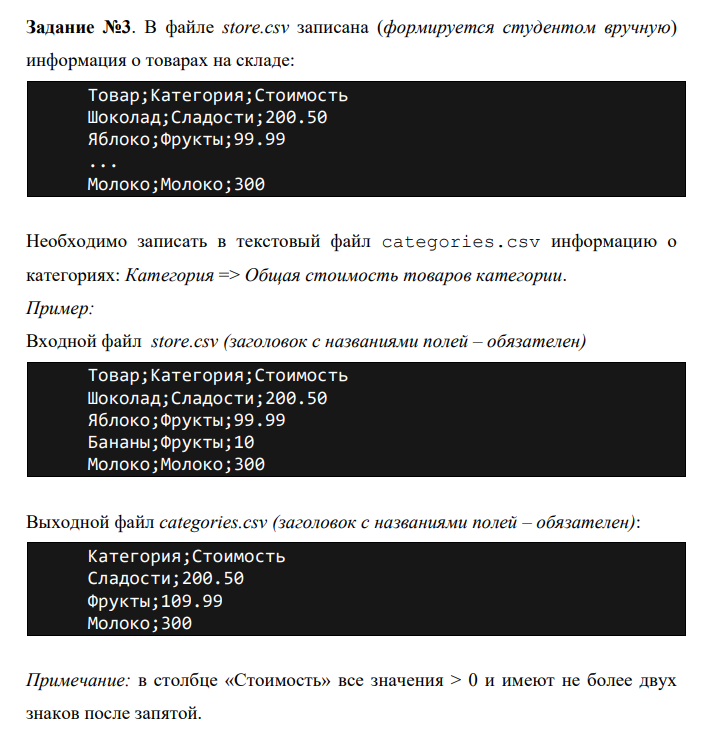
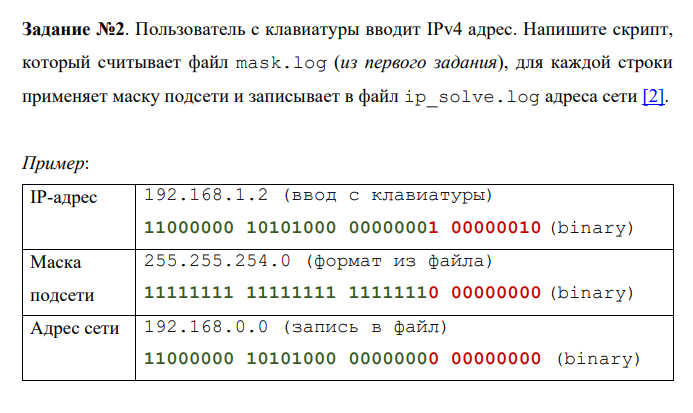
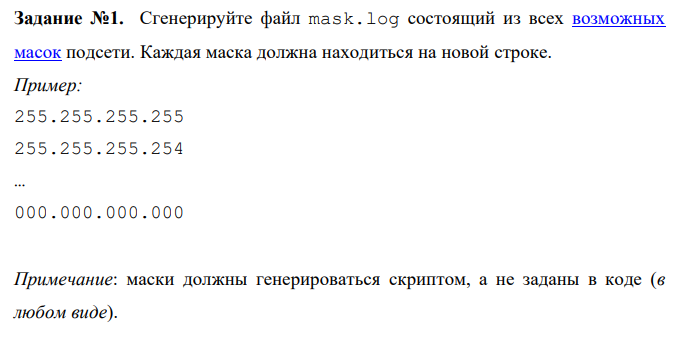
Отчет по Лабораторной работе №2   
«Работа со встроенными типами данных, числовые типы, строки, кортежи, измененяемые последовательности. Применение основных арифметических операций, определение приоритетов»  
для варианта № 2

Выполнил:

Студент 5 группы

Панфилов В.А.

Москва, 2024

from random import randint,choice

from ipaddress import IPv4Network

import csv

import os

import re

MASK = "./Lab2/mask.log"

IP\_ADDR = "./Lab2/ip\_solve.log"

STORE = "./Lab2/store.csv"

CATEGORIES = "./Lab2/categories.csv"

ARTICLE\_SOLVE = "./Lab2/article\_solve.txt"

ARTICLE = "./Lab2/article.txt"

def generate\_mask():

    mask = IPv4Network((0,randint(0,32)),strict=False)

    return str(mask.netmask)

def task1():

    try:

        with open(MASK,"w",) as mask\_file:

            for i in range(20):

                mask = generate\_mask()

                mask\_file.write(mask+"\n")

        print("Task 1 completed !")

    except Exception as Ex:

        print(Ex)

def apply\_mask(ip,mask):

    network = IPv4Network(ip+"/"+mask,strict=False)

    return str(network.network\_address)

def task2():

    ip\_addr = input("Введите ip адрес:\n")

    with open(MASK,"r+") as mask\_file:

        masks = mask\_file.readlines()

        for mask in masks:

            mask = mask.replace("\n","")

            with open(IP\_ADDR,"a+") as ip\_file:

                ip\_file.write(apply\_mask(ip\_addr,mask)+"\n")

    print("Task 2 completed !")

def task3():

    with open(STORE,"r",encoding="utf-8",newline="") as products\_file:

        products = csv.reader(products\_file)

        categories = dict()

        for product in products:

            product[0] = product[0].split(";")

            if product[0][1] in "Категория":

                categories[product[0][1]] = product[0][2]

                continue

            if product[0][1] in categories:

                categories[product[0][1]] += float(product[0][2])

                continue

            categories[product[0][1]] = float(product[0][2])

    print(categories)

    categories\_list = list(categories)

    with open(CATEGORIES,"a+",encoding="utf-8",newline="") as categories\_file:

        writer = csv.writer(categories\_file,delimiter=";")

        for i in categories\_list:

            writer.writerow([i,categories[i]])

    print("Task 3 completed !")

def task4():

    project\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))#определение нынешней директории

    example\_dir = os.path.join(project\_dir, 'example')#путь до папки

    extensions = ['.txt', '.py', '.jpg', '.csv', '.xml', '.json', '.html', '.css', '.js', '.pdf', '.doc']

    if not os.path.exists(example\_dir):

        os.mkdir(example\_dir)#создание папки

    for i in range(100):

        rand\_ext = choice(extensions)

        filename = f'my\_file{i}{rand\_ext}'

        file\_path = os.path.join(example\_dir, filename)

        open(file\_path, 'a').close()

def task5():

    file\_type = input("Введите расширение файла: ")

    example\_dir = 'example'

    count = 0

    for filename in os.listdir(example\_dir):

        if filename.endswith(file\_type):

            count += 1

    print(f"Количество файлов с расширением {file\_type}: {count}")

def task6():

    with open(ARTICLE,"r", encoding='utf-8') as f:

        text = f.read()

    text = re.sub(r'[^а-яA-Za-zА-Яа-я]', ' ', text).lower()# Удаление всех за исключением букв русского и английского алфавитов

    # Подсчитываем частоту каждой буквы

    counts = {}

    for letter in text:

        counts[letter] = counts.get(letter, 0) + 1

    # Вычисляем частоту в долях

    total = sum(counts.values())

    for letter in counts:

        counts[letter] = counts[letter] / total

    letters = sorted(counts, key=counts.get, reverse=True)

    with open(ARTICLE\_SOLVE, 'w', encoding='utf-8') as f:

        for letter in letters:

            f.write(f'{letter}: {counts[letter]:.3f}\n')

def main():

    task1()

    task2()

    task3()

    task4()

    task5()

    task6()

main()

