**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени**

**федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

Курсовая работа

По дисциплине «Структуры и алгоритмы обработки данных»

Выполнила: студент группы БФИ2001

Аржанова Анастасия

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2022

**Оглавление**

[Введение 3](#_Toc102777917)

[Задание 1: 4](#_Toc102777918)

[Решение задачи 1: 4](#_Toc102777919)

[Задание 2: 5](#_Toc102777920)

[Решение задачи 2 5](#_Toc102777921)

[Задание 3: 6](#_Toc102777922)

[Решение задачи 3: 7](#_Toc102777923)

[Задание 4: 9](#_Toc102777924)

[Решение задачи 4 9](#_Toc102777925)

[Задание 5: 11](#_Toc102777926)

[Решение задачи 5 12](#_Toc102777927)

[Задание 6: 14](#_Toc102777928)

[Решение задачи 6 15](#_Toc102777929)

[Задание 7: 16](#_Toc102777930)

[Решение задачи 7 17](#_Toc102777931)

[Задание 8: 18](#_Toc102777932)

[Решение задачи 8 19](#_Toc102777933)

[Задание 9: 20](#_Toc102777934)

[Решение задачи 9 21](#_Toc102777935)

[Задание 10: 21](#_Toc102777936)

[Решение задачи 10 22](#_Toc102777937)

[Вывод 24](#_Toc102777938)

# Введение

Общее задание: необходимо найти решение для 10 задач заданный в соответствие с первым вариантом.

# **Задание 1:**

Леша не любит скучать. Поэтому, когда ему скучно, он придумывает игры. Как-то раз Леша придумал следующую игру.

Задана последовательность a, состоящая из n целых чисел. Игрок может сделать несколько ходов. За один ход игрок может выбрать некоторый элемент последовательности (обозначим выбранный элемент ak) и удалить его, при этом из последователости также удаляются все элементы, равные ak+1 и ak-1. Описанный ход приносит игроку ak очков.

Леша максималист и поэтому хочет набрать как можно больше очков. Какое максимальное количество очков он сможет набрать?

Входные данные: В первой строке задано целое число n (1 ≤ n ≤ ) — количество элементов последовательности. Во второй строке записаны n целых чисел a1, a2, ..., an (1 ≤ ai ≤ ) — элементы последовательности.

Выходные данные: Выведите целое число — максимальное количество очков, которые может набрать Леша.

## Решение задачи 1:

Ввод производиться с клавиатуры

def record(a):

f = open("answer\_1.txt","w")

f.write(str(a)+"\n")

f.close

t=""

q = input("Введите число ")

while q.isalpha():

if not q.isdigit():

q = input("Введите число ")

n = int(q)

a = input("Введите числа через пробел ").split()

flag = True

while flag:

flag = False

if len(a) != n:

flag = True

a = input("Введите только "+ str(n)+" чисел").split()

else:

for i in a:

if not i.isdigit():

flag = True

a = input("Введите только числа через пробел ").split()

break

s = {}

ball = 0

ball2 = 0

maxi = 0

maxi2 = 0

for i in a:

if maxi < int(i):

maxi = int(i)

if maxi > int(i) and int(i)> maxi2:

maxi2 = int(i)

for i in range(n):

if maxi%2==0 and int(a[i])%2==0:

ball +=int(a[i])

elif int(a[i])%2==1 and maxi%2==1 :

ball +=int(a[i])

if maxi2%2==0 and int(a[i])%2==0:

ball2 +=int(a[i])

elif int(a[i])%2==1 and maxi2%2==1 :

ball2 +=int(a[i])

if ball > ball2:

print(ball)

record(str(ball))

else:

print(ball2)

record(str(ball2))

Результат:

Тест 1, рисунок 1

Входные первая строка- «2», вторая строка – «1 2»

Вывод на третьей строке – «2»

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Результат теста 1

Тест 2, рисунок 2

Входные первая строка- «3», вторая строка – «1 2 3»

Вывод на третьей строке – «4»

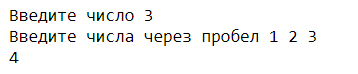


Рисунок 2 – Результат теста 2

Тест 1, рисунок 3

Входные первая строка- «9», вторая строка – «1 2 1 3 2 2 2 2 3»

Вывод на третьей строке – «10»

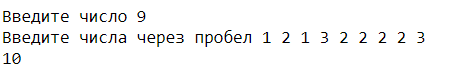


Рисунок 3 – Результат

# **Задание 2:**

Дана строка s. Требуется определить, существуют ли в данной строке s две непересекающиеся подстроки "AB" и "BA" (подстроки могут идти в любом порядке).

Входные данные: на вход подаётся строка s длиной от 1 до символов, состоящая из заглавных букв латинского алфавита.

Выходные данные: Выведите "YES" (без кавычек), если строка s содержит две непересекающиеся подстроки "AB" и "BA", и "NO" иначе.

## Решение задачи 2

def record(a):

f = open("answer\_2.txt","w")

f.write(str(a)+"\n")

f.close

word = input("Введите строку ")

flag=True

while flag:

flag = False

ch = 1

for i in range(len(word)):

ch+=1

if word[i:ch].isdigit():

word = input("Введите строку ")

flag = True

break

word = word.upper()

s = list(word)

f = 1

flag = 0

if "AB" not in word:

f = 0

elif "BA" not in word:

f = 0

else:

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

if s[i] == "A" and s[i+1] == 'B':

if i !=0 and s[i-1] == 'B':

f = 0

break

elif s[i] == 'B' and s[i+1] == 'A':

if i !=0 and s[i-1] == 'A':

f = 0

break

if f == 0:

print("NO")

record("NO")

else:

print("YES")

record("YES")

Результат:

Тест 1, рисунок 4

Входные первая строка- «ABA»

Вывод на второй строке – «NO»

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Результат теста 1

Тест 2, рисунок 5

Входные первая строка- «BACFAB»

Вывод на второй строке – «YES»



Рисунок 5 – Результат теста 2

Тест 3, рисунок 6

Входные первая строка- «AXBYBXA»

Вывод на второй строке – «NO»



Рисунок 6 – Результат теста 3

# Задание 3:

Кефа решил отпраздновать свой первый крупный заработок походом в ресторан.

Он живет возле необычного парка. Парк представляет из себя подвешенное дерево из n вершин c корнем в вершине 1. В вершине 1 также находится дом Кефы. К сожалению для нашего героя, в парке также находятся коты. Кефа уже выяснил номера вершин, в которых находятся коты.

В листовых вершинах парка находятся рестораны. Кефа хочет выбрать ресторан, в который он пойдет, но, к сожалению, он очень боится котов, поэтому он ни за что не пойдёт в ресторан, на пути к которому от его дома найдётся более m подряд идущих вершин с котами.

Ваша задача — помочь Кефе посчитать количество ресторанов, в которые он может сходить.

Входные данные  
В первой строке записаны два целых числа n и m (2 ≤ n ≤ , 1 ≤ m ≤ n) — количество вершин дерева и максимальное количество подряд идущих вершин с котами, которое способен перенести Кефа.

Во второй строке содержится n целых чисел a1, a2, ..., an, где каждое ai либо равняется 0 (тогда в вершине i нет кота), либо равняется 1 (тогда в вершине i есть кот).

В следующих n - 1 строках записаны ребра дерева в формате «xi yi»(без кавычек) (1 ≤ xi, yi ≤ n, xi ≠ yi), где xi и yi — вершины дерева, соединенные очередным ребром.

Гарантируется, что данный набор рёбер задаёт дерево.

Выходные данные  
Одно целое число — количество различных листьев дерева, на пути от дома Кефы до которых не больше m подряд идущих вершин с котами.

## Решение задачи 3:

def read():

f = open("3.txt", "r")

s=""

record = []

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s)

s = ""

f.close()

n = int(record[0][0])

m = int(record[0][2])

way = record[1].split()

graph = []

for i in range(n-1):

k = record[2+i].split()

graph.append(k)

return n,m,way,graph

def record(a):

f = open("answer\_3.txt","w")

f.write(str(a)+"\n")

f.close

t = []

flag = 0

n,m,way,graph = read()

for i in range(len(graph),0,-1):

w = graph[i-1]

for ii in range(len(t)):

if w[-1] == t[ii][0]:

q = list(w[0])+t[ii]

t[ii] = q

flag = 1

if flag == 0:

t.append(graph[i-1])

flag = 0

cat = ""

answer = 0

for i in t:

for ii in i:

cat += way[int(ii)-1]

if "11" in cat:

pass

else:

answer+=1

cat = ""

print(answer)

record(str(answer))

Результат:

Данные читаются из файла 3.txt, Рисунок 7 – входные данные для теста 1, рисунок 9 – входные данные для теста 2

Вывод изображен на рисунках 8 и 10.

Тест 1

Изображение выглядит как текст, компьютер, снимок экрана, ноутбук

Автоматически созданное описание

Рисунок 7 – Входные данные для теста 1 задачи 3

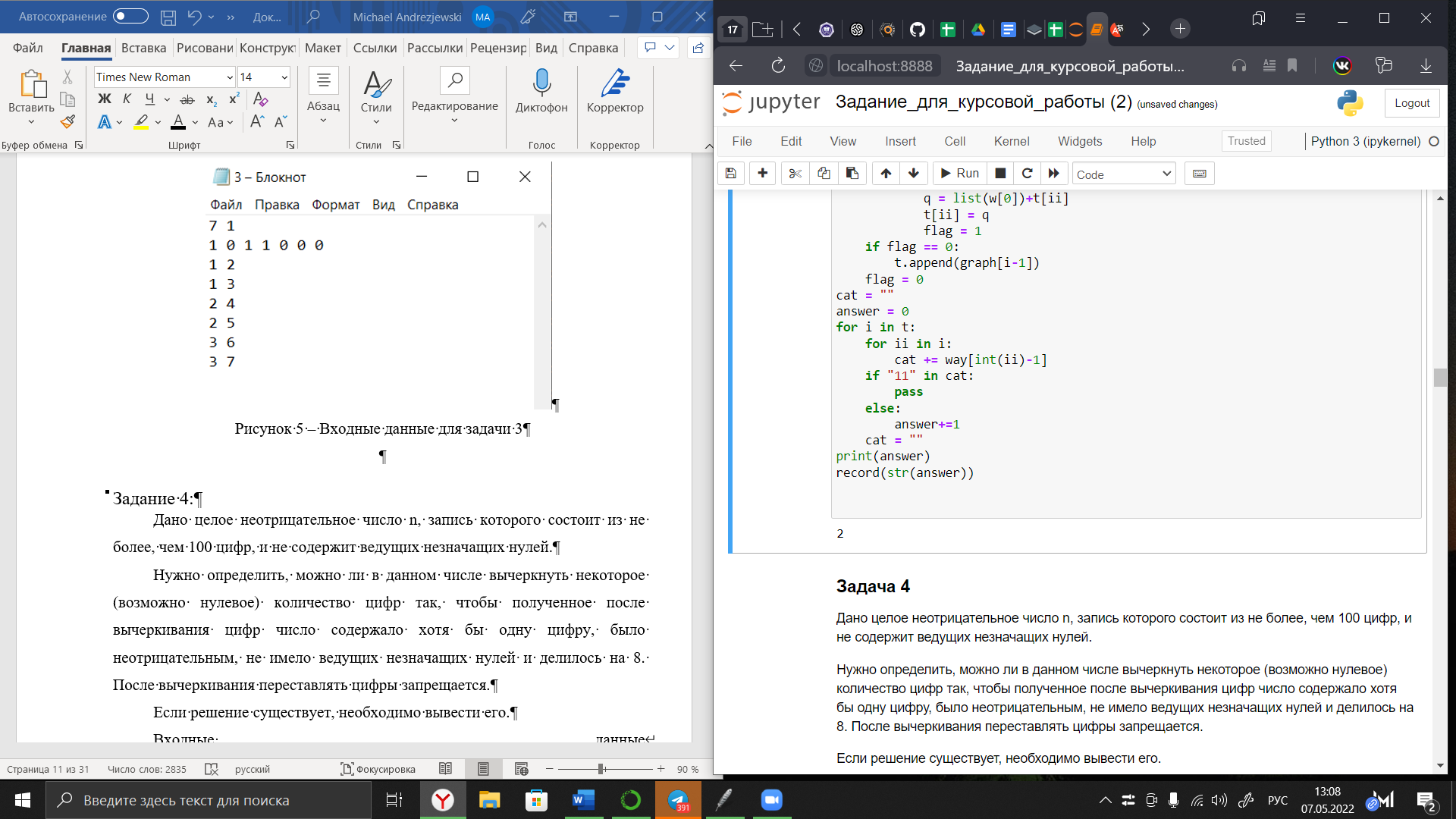


Рисунок 8 – Результат теста 1 задачи 3

Изображение выглядит как текст, снимок экрана, монитор, компьютер

Автоматически созданное описание

Рисунок 9 – Входные данные для теста 2 задачи 3

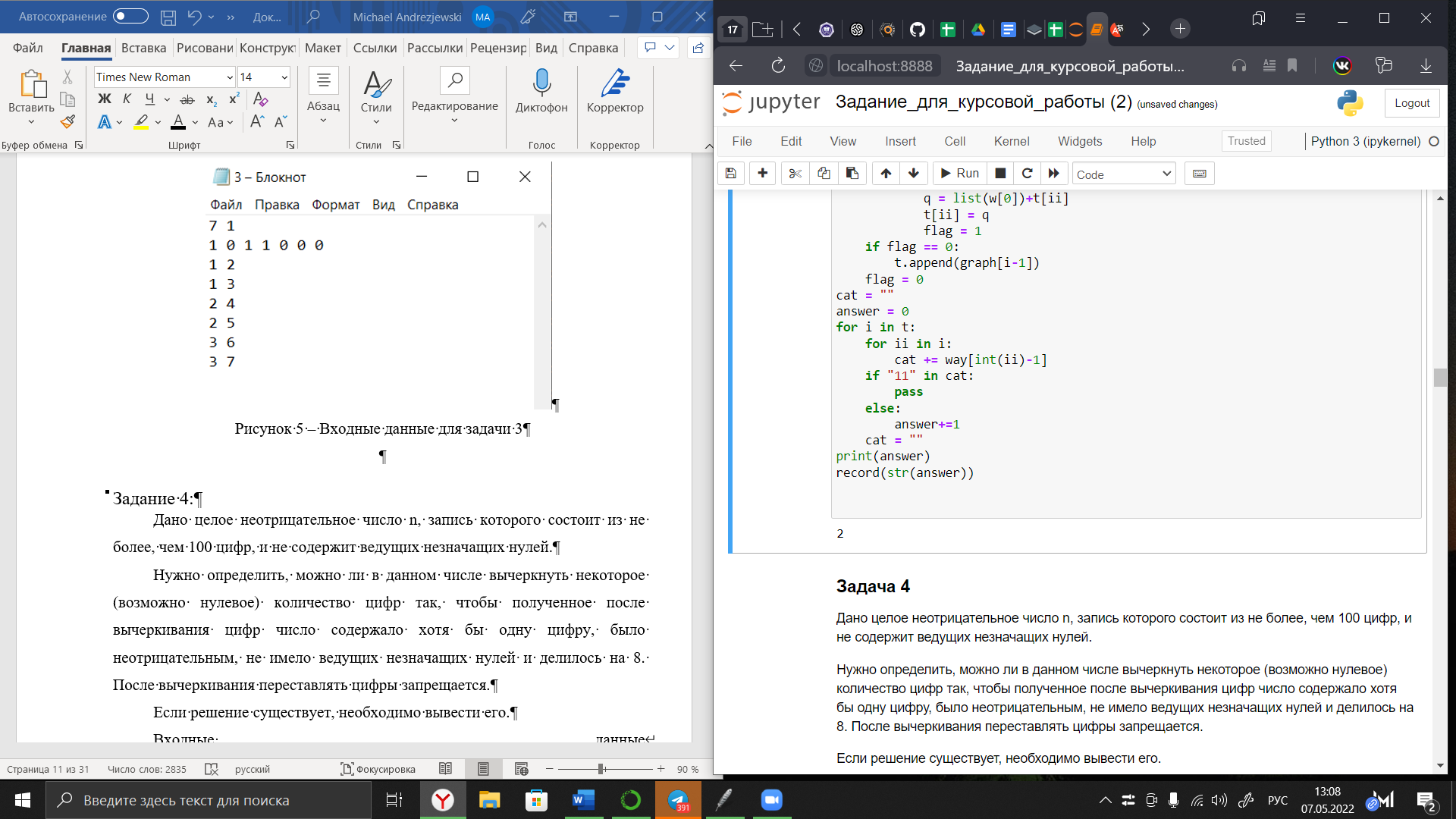


Рисунок 10 – Результат теста 2 задачи 3

# Задание 4:

Дано целое неотрицательное число n, запись которого состоит из не более, чем 100 цифр, и не содержит ведущих незначащих нулей.

Нужно определить, можно ли в данном числе вычеркнуть некоторое (возможно нулевое) количество цифр так, чтобы полученное после вычеркивания цифр число содержало хотя бы одну цифру, было неотрицательным, не имело ведущих незначащих нулей и делилось на 8. После вычеркивания переставлять цифры запрещается.

Если решение существует, необходимо вывести его.

Входные данные  
В единственной строке входных данных содержится целое неотрицательное число n. Запись числа n не содержит ведущих незначащих нулей и ее длина не превосходит 100 цифр.

Выходные данные  
Выведите "NO" (без кавычек), если искомого способа вычеркнуть некоторые цифры из числа n не существует.

Иначе выведите "YES" в первой строке и число, получившееся в результате вычеркивания некоторых цифр числа n, во второй строке. Выведенное число должно делиться на 8.

## Решение задачи 4

def delit(n,w):

if len(str(n)) == 1:

w.append(-1)

else:

for i in range(len(str(n))):

if int(str(n)[:i]+str(n)[i+1:]) % 8 == 0:

w.append(int(str(n)[:i]+str(n)[i+1:]))

else:

delit(int(str(n)[:i]+str(n)[i+1:]),w)

def record(a):

f = open("answer\_4.txt","w")

f.write(str(a)+"\n")

f.close

n\_st = input("Введите число ")

while n\_st.isalpha():

if not n\_st.isdigit():

n\_st = input("Введите число ")

n = int(n\_st)

w = []

delit(n,w)

flag = False

answer = 0

for i in w:

if i != -1:

answer = str(i)

flag = True

break

if flag:

print("YES\n"+answer)

record("YES\n"+answer)

else:

print("NO")

record("NO")

Результат:

Тест 1, рисунок 11

Входные первая строка- «3454»

Вывод на второй строка – «YES» и третьей строке – «344»

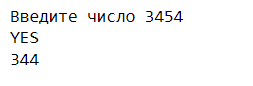


Рисунок 11 – Результат теста 1

Тест 2, рисунок 12

Входные первая строка- «10»

Вывод на второй строка – «YES» и третьей строке – «0»

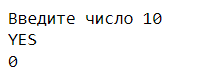


Рисунок 12 – Результат теста 2

Тест 3, рисунок 13

Входные первая строка- «111111»

Вывод на второй строка – «NO»



Рисунок 13 – Результат теста 3

# Задание 5:

Маленькая девочка Сьюзи каждый день слушает сказки перед сном. Сегодняшняя сказка была про дровосеков, поэтому маленькая девочка сразу же начала представлять себе, как дровосеки рубят деревья, и в её мыслях родилась ситуация, описанная ниже.

Есть n деревьев, расположенных вдоль дороги в точках с координатами x1, x2, ... , xn. Каждое дерево имеет свою высоту hi. Дерево можно срубить и повалить влево или вправо. Тогда оно будет занимать отрезки [xi - hi, xi] и [xi;xi + hi] соответственно. Пока дерево не срублено, оно занимает точку с координатами xi. Дерево можно повалить, если на отрезке, который оно должно занимать после сваливания, нет ни одной занятой точки. Дровосеки хотят заготовить как можно большее число деревьев, поэтому Сьюзи стало интересно, какое наибольшее количество деревьев можно повалить.

Входные данные  
В первой строке находится целое число n (1 ≤ n ≤ ) — количество деревьев.

В последующих n строках находятся пары целых чисел xi, hi (1 ≤ xi, hi ≤ ) — координата и высота і-го дерева.

Пары заданы в порядке возрастания xi. Никакие два дерева не находятся в точке с одинаковой координатой.

Выходные данные  
Требуется вывести одно число — максимальное количество деревьев, которое можно срубить по указанным правилам.

## Решение задачи 5

def read():

f = open("5.txt", "r")

record=[]

s=""

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s)

s = ""

f.close()

n = int(record[0])

x= []

for i in range(n):

t =record[1+i].split()

x.append([int(t[0]),int(t[1])])

return n,x

def record(a):

f = open("answer\_5.txt","w")

f.write(str(a)+"\n")

f.close

n,x = read()

ch=0

for i in range(len(x)):

if i == 0 or i == len(x)-1:

ch+=1

else:

if x[i][0]+x[i][1]<x[i+1][0]:

ch+=1

elif x[i][0]-x[i][1]>x[i-1][0]:

ch+=1

print(ch)

record(str(ch))

Результат:

Тест 1

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 14 – Входные данные для теста 1 задачи 5

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 15 – Результат теста 1 задачи 5

Тест 2

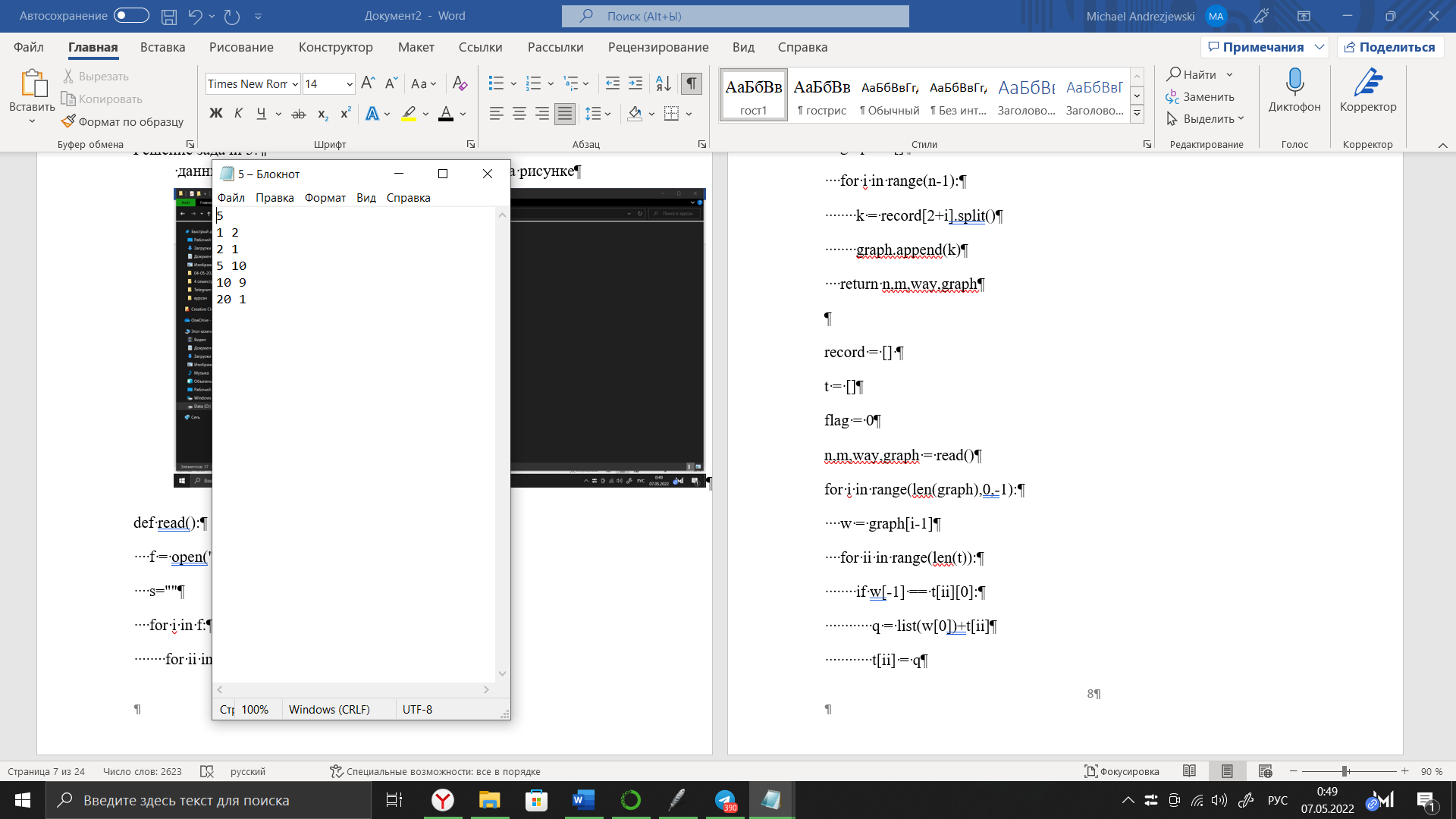


Рисунок 16 – Входные данные для теста 2 задачи 5

Изображение выглядит как текст, снимок экрана, компьютер, внутренний

Автоматически созданное описание

Рисунок 17 – Результат теста 2 задачи 5

# Задание 6:

 У вас есть массив 𝑎 длины 𝑛. Для каждого положительного числа 𝑥 в течение 𝑥-й секунды вы собираетесь выполнить следующую операцию:

Выберите несколько различных индексов 𝑖1,𝑖2,…,𝑖𝑘, которые находятся в диапазоне от 1 до 𝑛 включительно, и добавьте 2𝑥−1 к каждой соответствующей позиции в 𝑎. Формально 𝑎𝑖𝑗:=𝑎𝑖𝑗+2𝑥−1 для 𝑗=1,2,…,𝑘. Обратите внимание, что вы можете также не выбрать ни одного индекса. Вы должны сделать 𝑎 неубывающим как можно быстрее. Найдите наименьшее число 𝑇 такое, что вы можете сделать массив неубывающим не позднее, чем через 𝑇 секунд.

Массив 𝑎 называется неубывающим, если и только если 𝑎1≤𝑎2≤…≤𝑎𝑛.

Вы должны ответить на 𝑡 независимых тестовых случаев.

Входные данные Первая строка содержит одно целое число 𝑡 (1≤𝑡≤) — количество тестовых случаев.

Первая строка каждого тестового случая содержит одно целое число 𝑛 (1≤𝑛≤) — длину массива 𝑎. Гарантируется, что сумма значений 𝑛 по всем тестовым случаям не превышает 105.

Вторая строка каждого теста содержит 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (−≤𝑎𝑖≤).

Выходные данные Для каждого тестового примера выведите минимальное количество секунд, за которое вы можете сделать 𝑎 неубывающим.

## Решение задачи 6

from math import log2

def read():

f = open("6.txt", "r")

s=""

record = []

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s)

s = ""

f.close()

t = int(record[0])

n= []

a =[]

for i in range(0,t\*2,2):

n.append(int(record[1+i]))

a.append(record[2+i].split())

return t,n,a

def record(a):

f = open("answer\_6.txt","w")

f.write(str(a)+"\n")

f.close

t,n\_mas,a\_mas = read()

b = []

for k in range(t):

n=n\_mas[k]

a=a\_mas[k]

c=0

for i in range(1,n):

if(a[i]<a[i-1]):

if c > int(log2(int(a[i-1])-int(a[i])))+1:

c = c

else:

c = int(log2(int(a[i-1])-int(a[i])))+1

a[i]=a[i-1]

print(c)

b.append(c)

record(c)

Результат:

Тест 1

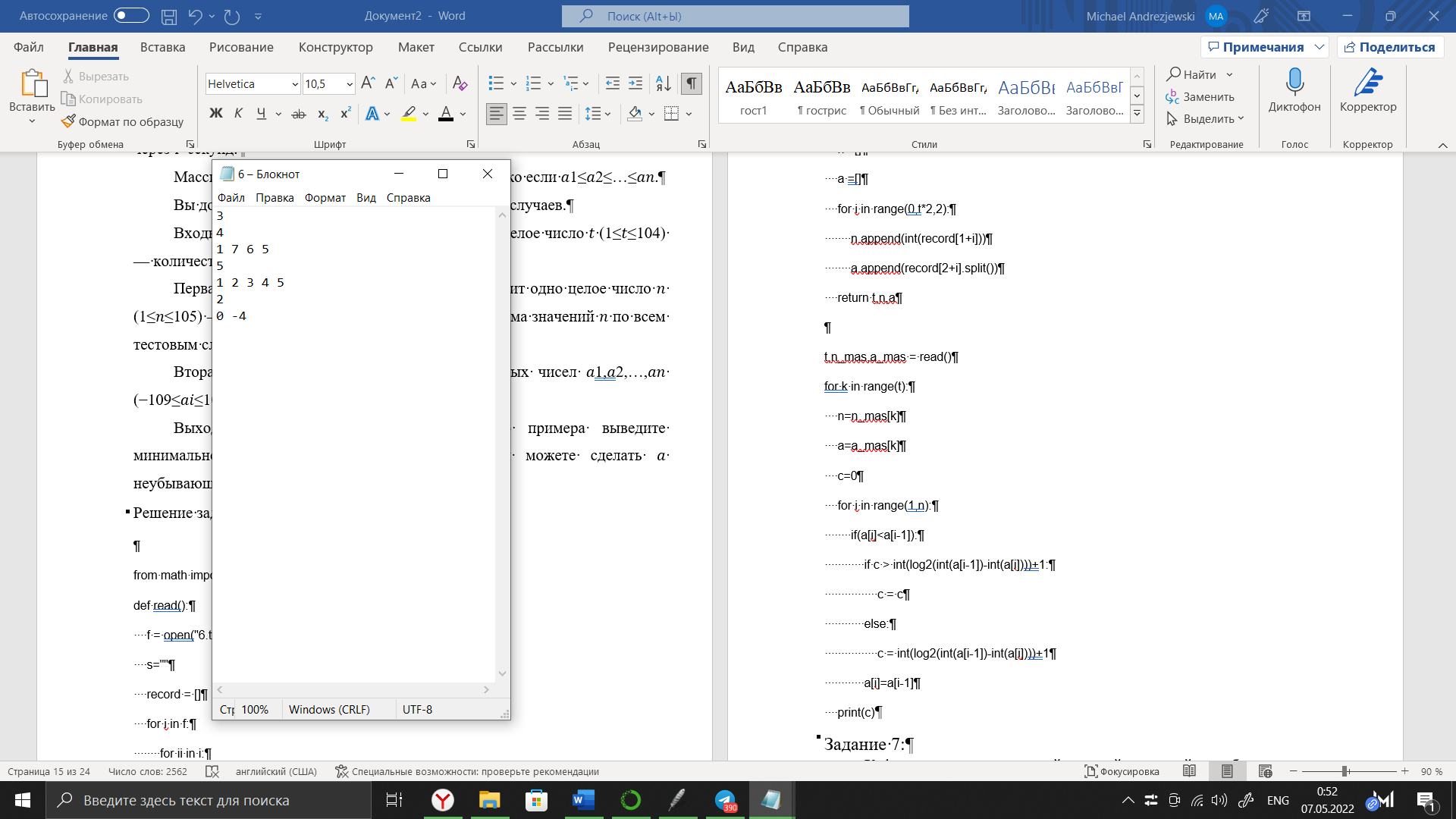


Рисунок 18 – Входные данные для теста 1 задачи 6

Изображение выглядит как текст, снимок экрана, компьютер, ноутбук

Автоматически созданное описание

Рисунок 19 – Результат теста 1 задачи 6

# Задание 7:

Кефа хочет отметить свой первый крупный заработок походом в ресторан. Однако ему нужна компания.

У Кефы есть n друзей, каждый из которых согласится пойти в ресторан, если Кефа попросит. Каждый друг характеризуется количеством денег у него и степенью дружбы с Кефой. Наш попугай не хочет, чтобы какой-то друг почувствовал себя бедным по сравнению с кем-то другим в компании (Кефа не в счет). Друг чувствует себя бедным, если в компании есть кто-то, у кого денег хотя бы на d единиц больше, чем у него. Также Кефа хочет, чтобы суммарная степень дружбы членов компании была максимальной. Помогите ему пригласить оптимальную компанию!

Входные данные Первая строка ввода содержит два целых числа, разделенных пробелом, n и d (1 ≤ n ≤ ) — количество друзей у Кефы и минимальная разница денег, приводящая к тому, что человек чувствует себя бедным.

В последующих n строках даны описания друзей Кефы, в (i + 1)-й строке содержится описание i-го друга вида mi, si (0 ≤ mi, si ≤ ) — количество денег и степень дружбы с Кефой соответственно.

Выходные данные  
Выведите максимальную суммарную степень дружбы, которой можно добиться.

## Решение задачи 7

def read():

f = open("7.txt", "r")

s=""

record = []

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s.split())

s = ""

f.close()

n = int(record[0][0])

d = int(record[0][1])

m =[]

s=[]

for i in range(0,n):

m.append(int(record[1+i][0]))

s.append(int(record[1+i][1]))

return n,d,m,s

def record(a):

f = open("answer\_7.txt","w")

f.write(str(a)+"\n")

f.close

n,d,m,s = read()

maxi = 0

for i in s:

if i > maxi:

maxi = i

answer = 0

for i in m:

t = m[s.index(maxi)]

if t+d >= i and i >= t-d:

answer+= s[m.index(i)]

print(answer)

record(str(answer))

Результат:

Тест 1

Изображение выглядит как текст, снимок экрана, компьютер, ноутбук

Автоматически созданное описание

Рисунок 20 – Входные данные для теста 1 задачи 7

Изображение выглядит как текст, снимок экрана, компьютер, внутренний

Автоматически созданное описание

Рисунок 21 – Результат теста 1 задачи 7

Тест 2

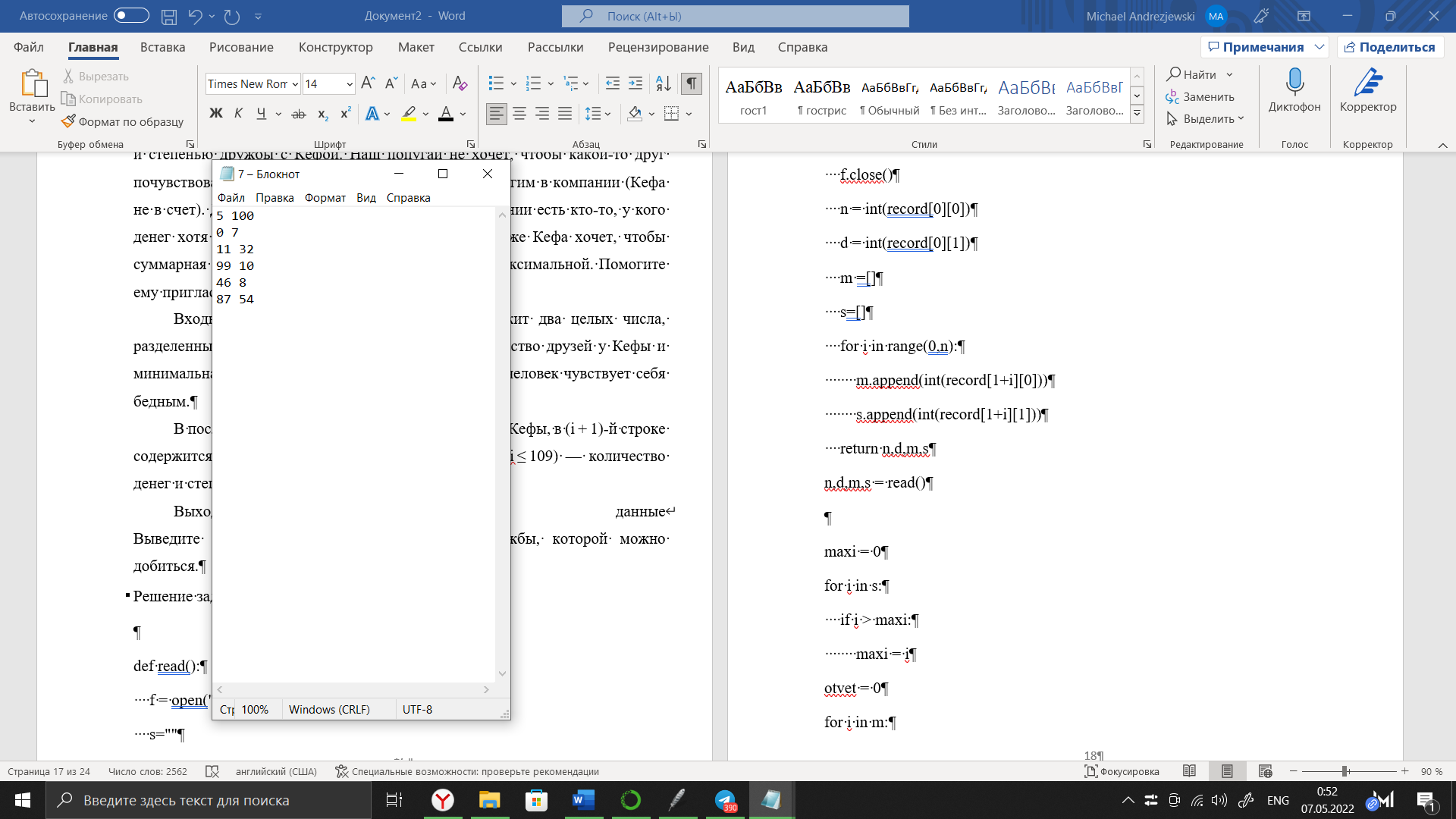


Рисунок 22 – Входные данные для теста 2 задачи 7

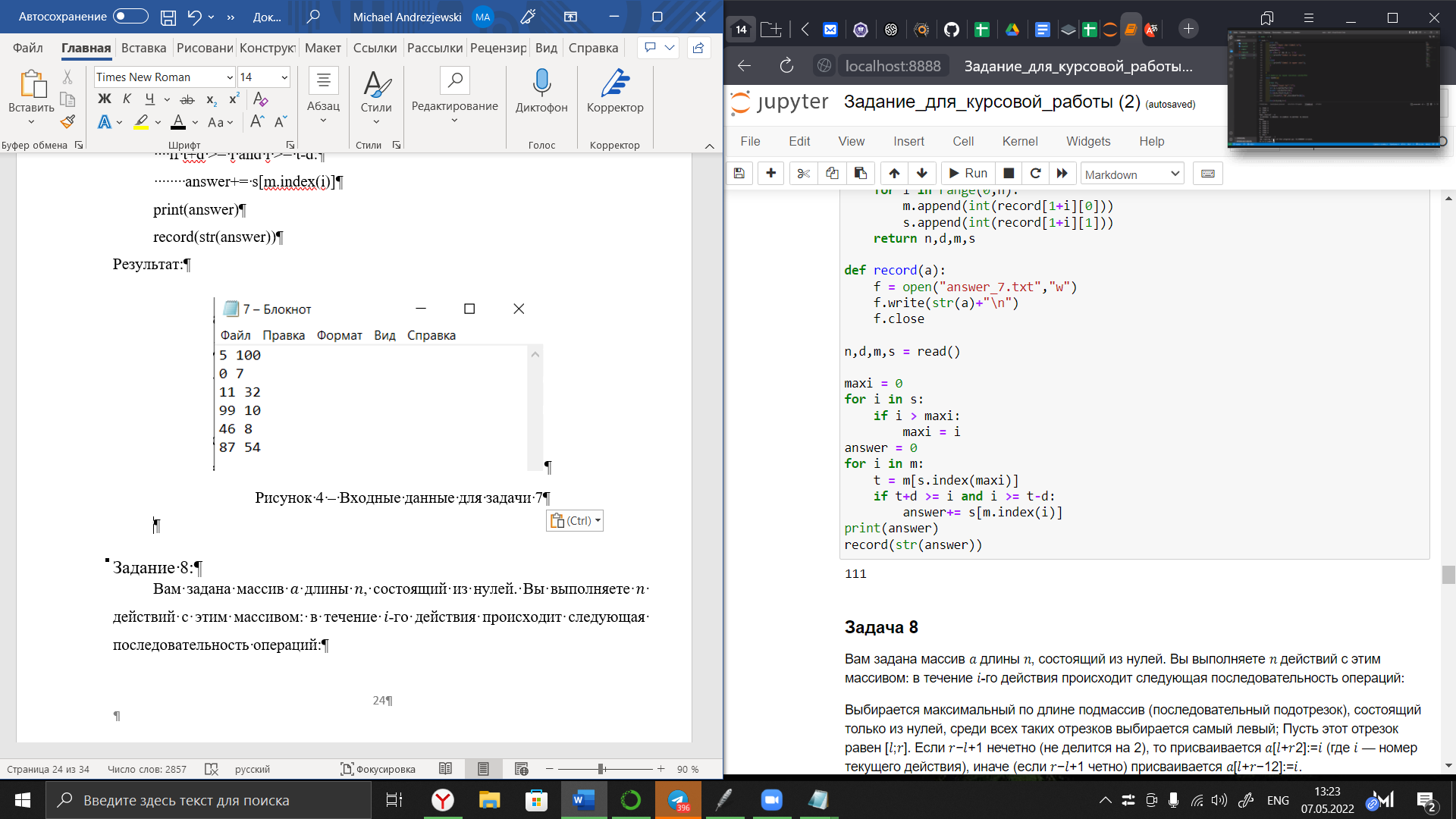


Рисунок 23 – Результат теста 2 задачи 7

# Задание 8:

Вам задана массив 𝑎 длины 𝑛, состоящий из нулей. Вы выполняете 𝑛 действий с этим массивом: в течение 𝑖-го действия происходит следующая последовательность операций:

Выбирается максимальный по длине подмассив (последовательный подотрезок), состоящий только из нулей, среди всех таких отрезков выбирается самый левый; Пусть этот отрезок равен [𝑙;𝑟]. Если 𝑟−𝑙+1 нечетно (не делится на 2), то присваивается 𝑎[𝑙+𝑟2]:=𝑖 (где 𝑖 — номер текущего действия), иначе (если 𝑟−𝑙+1 четно) присваивается 𝑎[𝑙+𝑟−12]:=𝑖.  
Рассмотрим массив 𝑎 длины 5 (изачально 𝑎=[0,0,0,0,0]). Тогда он меняется следующим образом:

Сначала мы выбираем отрезок [1;5] и присваиваем 𝑎[3]:=1, таким образом 𝑎 становится равен [0,0,1,0,0];  
затем мы выбираем отрезок [1;2] и присваиваем 𝑎[1]:=2, таким образом 𝑎 становится равен [2,0,1,0,0];  
затем мы выбираем отрезок [4;5] и присваиваем 𝑎[4]:=3, таким образом 𝑎 становится равен [2,0,1,3,0];  
затем мы выбираем отрезок [2;2] и присваиваем 𝑎[2]:=4, таким образом 𝑎 становится равен [2,4,1,3,0];  
и наконец мы выбираем отрезок [5;5] и присваиваем 𝑎[5]:=5, таким образом 𝑎 становится равен [2,4,1,3,5].  
Ваша задача — найти массив 𝑎 длины 𝑛 после выполнения всех 𝑛 действий. Заметьте, что ответ существует и единственен.

Вам необходимо ответить на 𝑡 независимых наборов тестовых данных.

Входные данные  
Первая строка входных данных содержит одно целое число 𝑡 (1≤𝑡≤) — количество наборов тестовых данных. Затем следуют 𝑡 наборов тестовых данных.

Единственная строка набора тестовых данных содержит одно целое число 𝑛 (1≤𝑛≤2⋅) — длину 𝑎.

Гарантируется, что сумма 𝑛 по всем наборам тестовых данных не превосходит 2⋅105 (∑𝑛≤2⋅).

Выходные данные  
Для каждого набора тестовых данных выведите ответ — массив 𝑎 длины 𝑛 после выполнения 𝑛 действий, описанных в условии задачи. Заметьте, что ответ существует и единственен.

## Решение задачи 8

def way(l,r):

if l>r:

return

if (r-l+1)%2==1:

m = (l+r)//2

else:

m = (l+r-1)//2

c[m-1] = [l-r,m-1]

way(l,m-1)

way(m+1,r)

def record(a):

s=""

f = open("answer\_8.txt","w")

for i in a:

for y in i:

s+=str(y)+" "

f.write(s+"\n")

s=""

f.close

n\_st=input("Введите число ")

while n\_st.isalpha():

if not n\_st.isdigit():

n\_st = input("Введите число ")

n = int(n\_st)

b=[]

d=[]

for i in range(n):

b\_st=input("Введите число ")

while b\_st.isalpha():

if not b\_st.isdigit():

b\_st = input("Введите число ")

b.append(int(b\_st))

for i in range(n):

k = b[i]

a=[0]\*k

c = [[0,1]]\*k

way(1,k)

c.sort()

for y in range(k):

a[c[y][1]] = y+1

print(\*a)

d.append(a)

record(d)

Результат:

Входные первая строка- «6» и следующие 6 строк.

Вывод: 6 строк снизу

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 24 – Результат

# Задание 9:

Девочка очень любит задачи про запросы на массиве.

Однажды ей попалась довольно известная задача: дан массив из 𝑛 элементов (элементы массива проиндексированы от 1); также есть 𝑞 запросов, каждый из которых задается парой целых чисел 𝑙𝑖, 𝑟𝑖 (1≤𝑙𝑖≤𝑟𝑖≤𝑛). Для каждого запроса необходимо найти сумму всех элементов массива с индексами от 𝑙𝑖 до 𝑟𝑖 включительно.

Такая задача показалась Девочке довольно скучной. Она решила, что перед тем, как отвечать на запросы, она перемешает элементы массива, причем так, чтобы сумма ответов на все запросы была максимально возможной. Ваша задача — найти значение этой максимальной суммы.

Входные данные Первая строка содержит два целых числа 𝑛 (1≤𝑛≤2⋅) и 𝑞 (1≤𝑞≤2⋅), разделенных пробелом — количество элементов в массиве и количество запросов, соответственно.

Следующая строка содержит 𝑛 целых чисел 𝑎𝑖 (1≤𝑎𝑖≤2⋅), разделенных пробелами — элементы массива.

Каждая из следующих 𝑞 строк содержит два целых числа 𝑙𝑖 и 𝑟𝑖 (1≤𝑙𝑖≤𝑟𝑖≤𝑛), разделенных пробелом — 𝑖-й запрос.

Выходные данные В единственной строке выведите целое число — максимальную сумму ответов на запросы после перемешивания элементов массива.

## Решение задачи 9

def read():

f = open("9.txt", "r")

s=""

record = []

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s.split())

s = ""

f.close()

n = int(record[0][0])

q = int(record[0][1])

l =[]

r =[]

a = record[1]

for i in range(0,q):

l.append(int(record[2+i][0]))

r.append(int(record[2+i][1]))

return n,q,a,l,r

def record(a):

f = open("answer\_9.txt","w")

f.write(str(a)+"\n")

f.close

n,q,a,l,r = read()

d=[0]\*n

a.sort()

for i in range(q):

for i in range(l[i],r[i]+1):

d[i-1] += 1

d.sort()

ans=0

for i in range(n):

ans+=d[i]\*int(a[i])

print(ans)

record(ans)

Результат:

Тест 1

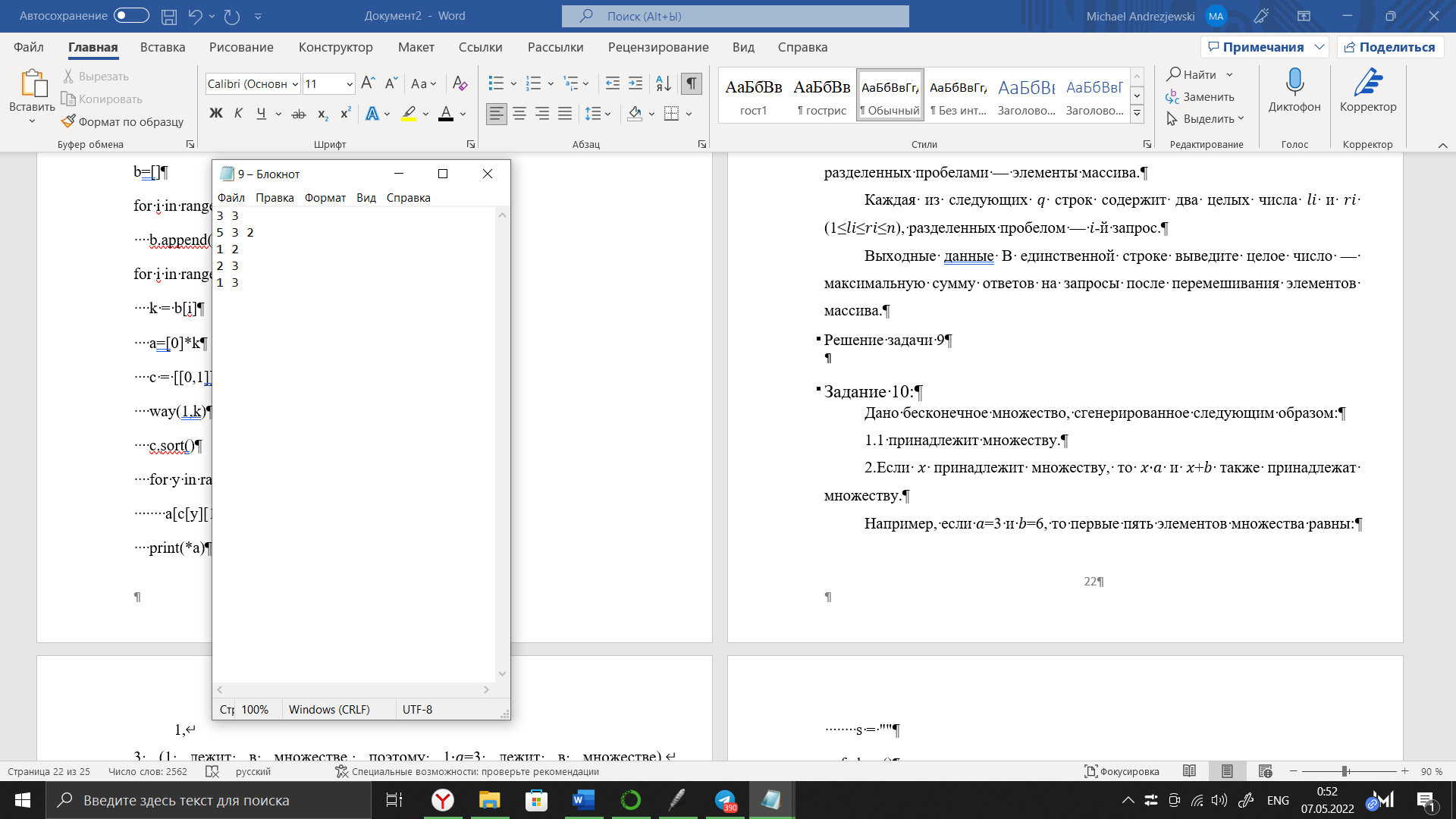


Рисунок 25 – Входные данные для теста 1 задачи 9

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

Рисунок 26 – Результат теста 1 задачи 9

Тест 2

Изображение выглядит как текст, снимок экрана, компьютер, внутренний

Автоматически созданное описание

Рисунок 27 – Входные данные для теста 2 задачи 9

Изображение выглядит как текст, компьютер, снимок экрана, ноутбук

Автоматически созданное описание

Рисунок 28 – Результат теста 2 задачи 9

# Задание 10:

Дано бесконечное множество, сгенерированное следующим образом:

1.1 принадлежит множеству.

2.Если 𝑥 принадлежит множеству, то 𝑥⋅𝑎 и 𝑥+𝑏 также принадлежат множеству.

Например, если 𝑎=3 и 𝑏=6, то первые пять элементов множества равны:

1,  
3 (1 лежит в множестве, поэтому 1⋅𝑎=3 лежит в множестве),  
7 (1 лежит в множестве, поэтому 1+𝑏=7 лежит в множестве),  
9 (3 лежит в множестве, поэтому 3⋅𝑎=9 лежит в множестве),  
13 (7 лежит в множестве, поэтому 7+𝑏=13 лежит в множестве).  
Даны положительные целые числа 𝑎, 𝑏, 𝑛. \Выясните, принадлежит ли 𝑛 множеству.

Входные данные  
Каждый тест содержит несколько наборов входных данных. Первая строка содержит одно целое число 𝑡 (1≤𝑡≤) – количество наборов входных данных. Далее следует описание наборов входных данных.\

Строка, описывающая набор входных данных содержит три целых числа 𝑛, 𝑎, 𝑏 (1≤𝑛,𝑎,𝑏≤), разделенных пробелом.\

Выходные данные  
Для каждого набора входных данных выведите «Yes», если 𝑛 принадлежит множеству, и «No» иначе. Вы можете выводить каждую букву в любом регистре (строчную или заглавную).

## Решение задачи 10

def read():

f = open("10.txt", "r")

s=""

record = []

for i in f:

for ii in i:

if ii != "\n":

s += ii

record.append(s.split())

s = ""

f.close()

t = int(record[0][0])

n = []

a = []

b = []

for i in range(0,t):

n.append(int(record[1+i][0]))

a.append(int(record[1+i][1]))

b.append(int(record[1+i][2]))

return t,n,a,b

def record(a):

f = open("answer\_10.txt","w")

for i in a:

f.write(str(i)+"\n")

f.close

def enumeration(k,a,b,n,q):

if n<k:

q.append('NO')

return

if (n-k)%b==0:

q.append('YES')

return

else:

enumeration(k\*a,a,b,n,q)

le,n,a,b = read()

answer = []

for i in range(le):

q=[]

flag = ''

if a[i] != 1:

enumeration(1,a[i],b[i],n[i],q)

flag = " ".join(q)

else:

flag = "YES"

answer.append(flag)

print(flag)

record(answer)

Результат:

Тест 1

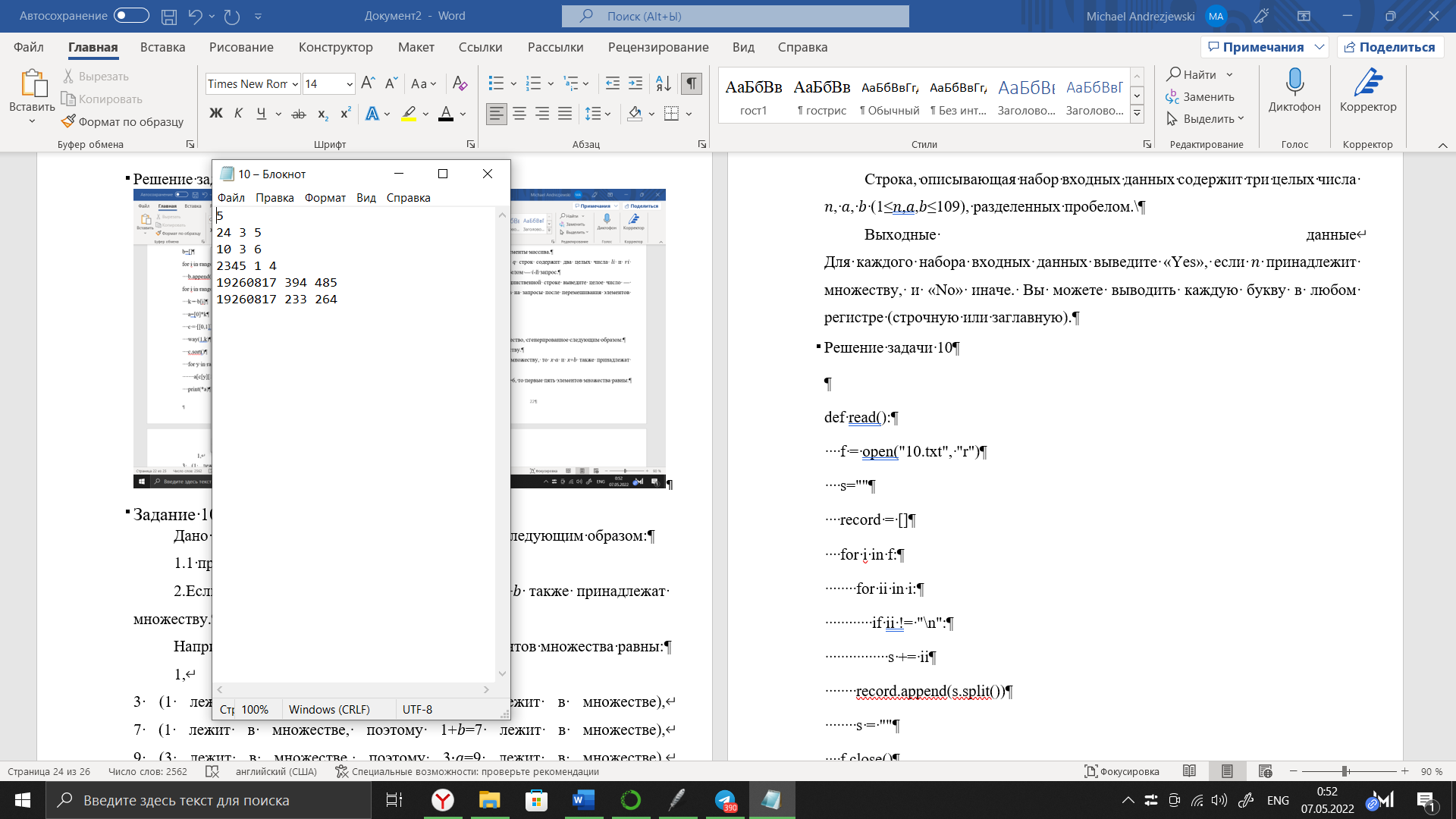


Рисунок 29 – Входные данные для теста 1 задачи 10

Изображение выглядит как текст, снимок экрана, компьютер, ноутбук

Автоматически созданное описание

Рисунок 30 – Результат теста 1 задачи 10

# Вывод

Были разработаны программы для решения заданных вариантом задач. Так же все задачи выводятся в текстовый файл