**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра “Математическая кибернетика и информационные технологии”

**Отчет по курсовой работе**

по дисциплине «Структуры и алгоритмы обработки данных»

Вариант №12.

Выполнила: студент группы БВТ2004

Готовко Антон

Проверил:

Мкртчян Грач Маратович

Москва, 2022

# Задачи

## Задача №1

Задан массив 𝑎1,𝑎2,…,𝑎𝑛, состоящий из 𝑛 положительных целых чисел.

Изначально вы находитесь в позиции 1, и ваше количество очков равно 𝑎1. Можно совершать два типа шагов:

шаг вправо — перейти из текущей позиции 𝑥 в 𝑥+1 и получить 𝑎𝑥+1 очков. Этот шаг можно делать только если 𝑥<𝑛. шаг влево — перейти из текущей позиции 𝑥 в 𝑥−1 и получить 𝑎𝑥−1 очков. Этот шаг можно делать только если 𝑥>1. Также нельзя совершать два или более шагов влево подряд. Вам требуется совершить ровно 𝑘 шагов. Не более 𝑧 из них могут быть шагами влево.

Какое наибольшее количество очков можно получить?

Входные данные  
В первой строке записано одно целое число 𝑡 (1≤𝑡≤104) — количество наборов входных данных.

В первой строке каждого набора входных данных записаны три целых числа 𝑛,𝑘 и 𝑧 (2≤𝑛≤105, 1≤𝑘≤𝑛−1, 0≤𝑧≤𝑚𝑖𝑛(5,𝑘)) — количество элементов в массиве, суммарное количество шагов, которое вы должны сделать, и максимальное количество шагов влево, которое вы можете сделать.

Во второй строке каждого набора входных данных записаны 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (1≤𝑎𝑖≤104) — данный массив.

Сумма 𝑛 по всем наборам входных данных не превосходит 3⋅105.

Выходные данные  
Выведите 𝑡 целых чисел — для каждого набора входных данных выведите наибольшее количество очков, которое можно получить, если требуется сделать ровно 𝑘 шагов, не более 𝑧 из них могут быть шагами влево и не должно быть двух шагов влево подряд.

## Задача №2

Вам дан массив 𝑎 из 𝑛 целых чисел. Вы хотите сделать все элементы 𝑎 равными нулю, применив следующую операцию ровно три раза:

Выберите отрезок, к каждому числу на этом отрезке добавьте число кратное 𝑙𝑒𝑛, где 𝑙𝑒𝑛 это длина этого отрезка (добавленные числа могут быть разными). Можно доказать, что таким образом всегда можно превратить все элементы 𝑎 в нули.

Входные данные  
В первой строке записано одно целое число 𝑛 (1≤𝑛≤100000): количество элементов массиве.

Во второй строке записаны 𝑛 элементов массива 𝑎, разделенные пробелами: 𝑎1,𝑎2,…,𝑎𝑛 (−109≤𝑎𝑖≤109).

Выходные данные  
Выведите шесть строк, описывающих три операции.

Для каждой операции, выведите две строки:

В первой строке выведите два числа 𝑙, 𝑟 (1≤𝑙≤𝑟≤𝑛): границы выбранного отрезка. Во второй строке выведите 𝑟−𝑙+1 целых чисел 𝑏𝑙,𝑏𝑙+1,…,𝑏𝑟 (−1018≤𝑏𝑖≤1018): числа, которые нужно прибавить к 𝑎𝑙,𝑎𝑙+1,…,𝑎𝑟, соответственно; 𝑏𝑖 должно делиться на 𝑟−𝑙+1.

## Задача №3

Вам задан массив 𝑎, состоящий из 𝑛 целых чисел. Индексы элементов массива начинаются с нуля (то есть первый элемент — это 𝑎0, второй — 𝑎1, и так далее). Вы можете развернуть не более одного подмассива (последовательного отрезка) этого массива. Напомним, что подмассив 𝑎 с границами 𝑙 и 𝑟 равен 𝑎[𝑙;𝑟]=𝑎𝑙,𝑎𝑙+1,…,𝑎𝑟.

Ваша задача — развернуть такой подмассив, чтобы сумма элементов на четных позицияx получившегося массива была максимально возможной (то есть сумма элементов 𝑎0,𝑎2,…,𝑎2𝑘 для целого числа 𝑘=⌊𝑛−12⌋ должна быть максимально возможной).

Вам необходимо ответить на 𝑡 независимых наборов тестовых данных.

Входные данные  
Первая строка входных данных содержит одно целое число 𝑡 (1≤𝑡≤2⋅104) — количество наборов тестовых данных. Затем следуют 𝑡 наборов тестовых данных.

Первая строка набора содержит одно целое число 𝑛 (1≤𝑛≤2⋅105) — длину 𝑎. Вторая строка набора содержит 𝑛 целых чисел 𝑎0,𝑎1,…,𝑎𝑛−1 (1≤𝑎𝑖≤109), где 𝑎𝑖 — это 𝑖-й элемент 𝑎.

Гарантируется, что сумма 𝑛 не превосходит 2⋅105 (∑𝑛≤2⋅105).

Выходные данные  
Для каждого набора тестовых данных выведите ответ на отдельной строке — максимально возможная сумма элементов на четных позициях после разворота не более одного подмассива (последовательного отрезка) 𝑎.

## Задача №4

У Пети есть прямоугольная доска размера 𝑛×𝑚. Изначально на доске размещено 𝑘 фишек, 𝑖-я из них находится в клетке, на пересечении 𝑠𝑥𝑖-й строки и 𝑠𝑦𝑖-го столбца.

За одно действие Петя может сдвинуть все фишки влево, вправо, вниз или вверх на 1 ячейку.

Если фишка была в клетке (𝑥,𝑦), то после операции:

влево, ее координаты будут (𝑥,𝑦−1); вправо, ее координаты будут (𝑥,𝑦+1); вниз, ее координаты будут (𝑥+1,𝑦); вверх, ее координаты будут (𝑥−1,𝑦). Если фишка находится у стенки доски, и действие, выбранное Петей, двигает ее в направлении стены, то фишка остается на своей текущей позиции.

Обратите внимание, что несколько фишек могут располагаться в одной и той же клетке.

Для каждой фишки Петя выбрал позицию, в которой она должна побывать. Обратите внимание, что фишка не обязательно заканчивает в этой позиции.

Так как у Пети не очень много свободного времени, он готов сделать не более 2𝑛𝑚 действий, описанных выше.

Вам предстоит выяснить, какие действия должен делать Петя, чтобы все фишки побывали хотя бы раз в выбранных для них клетках. Или определить, что за 2𝑛𝑚 действий выполнить это невозможно.

Входные данные  
Первая строка содержит три целых числа 𝑛,𝑚,𝑘 (1≤𝑛,𝑚,𝑘≤200) — количество строк и столбцов доски и количество фишек соответственно.

Следующие 𝑘 содержат по два целых чисел 𝑠𝑥𝑖,𝑠𝑦𝑖 (1≤𝑠𝑥𝑖≤𝑛,1≤𝑠𝑦𝑖≤𝑚) — начальная позиция 𝑖-й фишки.

Следующие 𝑘 содержат по два целых чисел 𝑓𝑥𝑖,𝑓𝑦𝑖 (1≤𝑓𝑥𝑖≤𝑛,1≤𝑓𝑦𝑖≤𝑚) — позиция, которую должна посетить 𝑖-я фишка хотя бы раз.

Выходные данные  
В первой строке выведите количество операций, чтобы каждая фишка посетила хотя бы раз позицию, которую выбрал для нее Петя.

Во второй строке выведите последовательность операций. Для обозначения операций влево, вправо, вниз, вверх используйте символы 𝐿,𝑅,𝐷,𝑈 соответственно.

Если искомой последовательности не существует, в единственной строке выведите -1.

## Задача №5

Летние каникулы начались, поэтому Алиса и Боб хотят играть и веселиться, но... Их мама не согласна с этим. Она говорит, что они должны прочитать какое-то количество книг перед всеми развлечениями. Алиса и Боб прочитают каждую книгу вместе, чтобы быстрее закончить это задание.

В семейной библиотеке есть 𝑛 книг. 𝑖-я книга характеризуется тремя целыми числами: 𝑡𝑖 — количество времени, которое Алиса и Боб должны потратить, чтобы прочитать ее, 𝑎𝑖 (равное 1, если Алисе нравится 𝑖-я книга, и 0, если не нравится), и 𝑏𝑖 (равное 1, если Бобу нравится 𝑖-я книга, и 0, если не нравится).

Поэтому им нужно выбрать какие-то книги из имеющихся 𝑛 книг таким образом, что:

Алисе нравятся не менее 𝑘 книг из выбранного множества и Бобу нравятся не менее 𝑘 книг из выбранного множества; общее время, затраченное на прочтение этих книг минимизировано (ведь они дети и хотят начать играть и веселиться как можно скорее). Множество, которое они выбирают, одинаковое и для Алисы и для Боба (они читают одни и те же книги), и они читают все книги вместе, таким образом, суммарное время чтения равно сумме 𝑡𝑖 по всем книгам, которые находятся в выбранном множестве.

Ваша задача — помочь им и найти любое подходящее множество книг или определить, что такое множество найти невозможно.

Входные данные  
Первая строка теста содержит два целых числа 𝑛 и 𝑘 (1≤𝑘≤𝑛≤2⋅105).

Следующие 𝑛 строк содержат описания книг, по одному описанию в строке: 𝑖-я строка содержит три целых числа 𝑡𝑖, 𝑎𝑖 и 𝑏𝑖 (1≤𝑡𝑖≤104, 0≤𝑎𝑖,𝑏𝑖≤1), где:

𝑡𝑖 — количество времени, необходимое для прочтения 𝑖-й книги;  
𝑎𝑖, равное 1, если Алисе нравится 𝑖-я книга, и 0 в обратном случае;  
𝑏𝑖, равное 1, если Бобу нравится 𝑖-я книга, и 0 в обратном случае.  
Выходные данные  
Если подходящего решения не существует, выведите число -1. Иначе выведите целое число 𝑇 — минимальное суммарное время, необходимое для прочтения подходящего множества книг.

## Задача №6

Вы — амбициозный король, который хочет быть Императором Действительных чисел. Но перед этим вам нужно стать Императором Целых чисел. Рассмотрим числовую ось. Столица вашей империи изначально находится в точке 0. На прямой есть 𝑛 незахваченных королевств в позициях 0<𝑥1<𝑥2<…<𝑥𝑛. Вы хотите захватить все эти королевства.

Вы можете делать два вида действий:

Вы можете переместить свою столицу (пусть ее текущая координата 𝑐1) в любое захваченное королевство (пусть его позиция 𝑐2), стоимость такого действия 𝑎⋅|𝑐1−𝑐2|. Вы можете из текущей столицы (пусть ее текущая координата 𝑐1) захватить королевство (пусть его позиция 𝑐2), стоимость такого действия 𝑏⋅|𝑐1−𝑐2|. Вы не можете захватывать королевство, если между вашей столицей и целью есть другие незахваченные королевства. Обратите внимание, что вы не можете расположить столицу в точке, где нет королевства. Другими словами, в любой момент времени ваша столица может быть только в точке 0 или в 𝑥1,𝑥2,…,𝑥𝑛. Также обратите внимание, что захват королевства не изменяет положение вашей столицы.

Выведите минимальную суммарную стоимость захвата всех королевств. Ваша столица может оказаться в итоге в любой точке.

Входные данные Первая строка содержит одно целое число 𝑡 (1≤𝑡≤1000) — количество наборов входных данных. Далее следуют описания наборов.

Первая строка каждого набора содержит 3 целых числа 𝑛, 𝑎 и 𝑏 (1≤𝑛≤2⋅105; 1≤𝑎,𝑏≤105).

Вторая строка каждого набора содержит 𝑛 целых чисел 𝑥1,𝑥2,…,𝑥𝑛 (1≤𝑥1<𝑥2<…<𝑥𝑛≤108).

Гарантируется, что сумма значений 𝑛 по всем наборам входных данных не превосходит 2⋅105.

Выходные данные Для каждого набора входных данных выведите одно число: минимальную стоимость захвата всех королевств.

## Задача №7

Вам даны две последовательности целых чисел 𝑎1,…,𝑎𝑛 и 𝑏1,…,𝑏𝑚. Для каждого 𝑗=1,…,𝑚 найдите наибольший общий делитель чисел 𝑎1+𝑏𝑗,…,𝑎𝑛+𝑏𝑗.

Входные данные  
В первой строке записано два целых числа 𝑛 и 𝑚 (1≤𝑛,𝑚≤2⋅105).

Во второй строке записано 𝑛 целых чисел 𝑎1,…,𝑎𝑛 (1≤𝑎𝑖≤1018).

В третьей строке записано 𝑚 целых чисел 𝑏1,…,𝑏𝑚 (1≤𝑏𝑗≤1018).

Выходные данные  
Выведите 𝑚 целых чисел. 𝑗-е из этих чисел должно быть равно НОД(𝑎1+𝑏𝑗,…,𝑎𝑛+𝑏𝑗).

## Задача №8

Вам заданы два массива целых чисел 𝑎 и 𝑏 длины 𝑛. Вы можете развернуть не более одного подмассива (последовательного отрезка) массива 𝑎.

Ваша задача состоит в том, чтобы перевернуть такой подмассив, чтобы сумма ∑𝑖=1𝑛𝑎𝑖⋅𝑏𝑖 была максимально возможной.

Входные данные  
Первая строка содержит одно целое число 𝑛 (1≤𝑛≤5000).

Вторая строка содержит 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (1≤𝑎𝑖≤107).

Третья строка содержит 𝑛 целых чисел 𝑏1,𝑏2,…,𝑏𝑛 (1≤𝑏𝑖≤107).

Выходные данные  
Выведите одно целое число — максимально возможную сумма после разворота не более одного подмассива (последовательного отрезка) 𝑎.

## Задача №9

Жил-был в одной стране Царь по имени Цопа. После очередной царской реформы полномочия Царя стали настолько широкими, что, в частности, он стал собственноручно заниматься финансовой отчётностью.

Известен суммарный доход A его Царства по итогам 0-го года, и суммарный доход B по итогам n-го года (оба числа могут быть отрицательными, что означает — в этот год экономика Царства была убыточной).

Царь хочет продемонстрировать финансовую стабильность, для этого ему надо подобрать единый коэффициент X — коэффициент роста дохода Царства за один год. Этот коэффициент должен удовлетворять уравнению:

A·Xn = B. Разумеется, Царь не собирается делать такую работу вручную, и требует от вас написать программу, ищущую этот коэффициент X.

Следует отметить, что дробные числа крайне не любят в экономических структурах Царства, поэтому как входные данные, так и искомый коэффициент должны быть целыми. Искомый коэффициент X может оказаться равным нулю или даже быть отрицательным.

Входные данные  
В единственной строке записаны три целых числа A, B, n, удовлетворяющих условиям: |A|, |B| ≤ 1000, 1 ≤ n ≤ 10.

Выходные данные  
Выведите единственное целое число — искомый целый коэффициент X, или фразу «No solution», если такого коэффициента не существует, или он не целый. Если ответов несколько, выведите любой.

## Задача №10

У Алисы есть торт, и она собирается его разрезать. Она выполнит следующую операцию 𝑛−1 раз: выберет кусочек пирога (изначально весь торт является одним куском) с весом 𝑤≥2 и разрежем его на два более маленьких кусочка с весами ⌊𝑤2⌋ и ⌈𝑤2⌉ (⌊𝑥⌋ и ⌈𝑥⌉ обозначают округление вниз и вверх, соответственно).

После того как Алиса разрежет торт на 𝑛 кусочков, она расположит эти 𝑛 кусочков на столе в произвольном порядке. Обозначим за 𝑎𝑖 вес 𝑖-го кусочка.

Вам дан массив 𝑎. Определите, существует ли некоторый начальный вес торта и последовательность операций разрезания такие, чтобы веса кусочков получались равными 𝑎.

Входные данные  
Первая строка содержит одно целое число 𝑡 (1≤𝑡≤104) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число 𝑛 (1≤𝑛≤2⋅105).

Вторая строка содержит 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (1≤𝑎𝑖≤109).

Гарантируется, что сумма 𝑛 для всех наборов входных данных не превышает 2⋅105.

Выходные данные  
Для каждого набора входных данных выведите одну строку: выведите YES, если массив 𝑎 может быть получен с помощью действий Алисы, иначе выведите NO.

Вы можете выводить буквы в любом регистре (например, YES, Yes, yes, yEs будут приняты как положительный ответ).

# Выполнение

## Листинг кода задачи №1

**def** solve(t: int,n: int, k: int, z: int, a: list[int]) **->** int:

ans **=** 0

s **=** 0

mx **=** 0

**for** t **in** range(z **+** 1):

pos **=** k **-** 2 **\*** t

**if** pos **<** 0:

**continue**

mx **=** 0

s **=** 0

**for** i **in** range(pos **+** 1):

**if** i **<** n **-** 1:

mx **=** max(mx, a[i] **+** a[i **+** 1])

s **+=** a[i]

ans **=** max(ans, s **+** mx **\*** t)

**return** max(ans, s **+** mx **\*** t)

**def** inp(n: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** n:

*# Проверка количества введенных данных*

**if** number\_of\_input **==** 3:

final\_error **+=** str(" "**.**join([str(i **+** 1) **for** i **in**

range(n)]))

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 10 **\*\*** 4):

*# Проверка диапазонов ввода числа t*

final\_error **=** "Поддерживаемый диапазон входных данных: (1 ≤

t ≤ 10^4)"

**raise** BaseException

**elif** number\_of\_input **==** 2 \

**and** (**not** (2 **<=** result[0] **<** 10 **\*\*** 5)

**or** **not** (1 **<=** result[1] **<=** result[0] **-** 1)

**or** **not** (0 **<=** result[2] **<=** min(5, result[1]))):

*# Проверка диапазонов ввода для чисел n, k, z*

final\_error **=** ""

**if** **not** (2 **<=** result[0] **<** 10 **\*\*** 5):

final\_error **+=** "Диапазон n: 2 ≤ n ≤ 10^5"

**if** **not** (1 **<=** result[1] **<=** result[0] **-** 1):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон k: 1 ≤ k ≤ n-1"

**if** **not** (0 **<=** result[2] **<=** min(5, result[1])):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон z: 0 ≤ Z ≤ min(5,k)"

**raise** BaseException

**elif** number\_of\_input **==** 3 **and** \

(max(result) **>** 10 **\*\*** 4 **or**

min(result) **<** 1):

*# Проверка на содержание нуля в массиве*

final\_error **=** "Диапазон ai: 1 ≤ ai ≤ 10^4"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** '\_\_main\_\_':

result **=** []

t **=** inp(1, "Введите t: ", "Введите данные согласно примеру: 1", 1)[0]

**for** \_ **in** range(t):

n, k, z **=** inp(3, "Введите n, k, z (через пробел): ", "Введите данные согласно примеру: 3 2 1", 2)

a **=** inp(n, "Введите массив a: ", "Введите данные согласно примеру:

", 3)

result**.**append(solve(t,n, k, z, a))

print("Выходные данные\n"**+**"\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №2

**def** solve(n: int, a: list[int]):

result **=** ""

result **+=** "1 1\n"**+**str((n**-**1)**\***a[0])**+**"\n"

**if** n**>=**2:

result **+=** str(2)**+** " " **+** str(n) **+** "\n"

result **+=** " "**.**join([str(int((n**-**1)**\***a[i])) **for** i **in** range(1,n)]) **+** "\n"

**else**:result **+=**("1 1\n0")

result **+=** str(1) **+** " " **+** str(n) **+** "\n"

result **+=** " "**.**join([str(int(**-**n**\***a[i])) **for** i **in** range(n)]) **+** "\n"

**return** result

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 10 **\*\*** 4):

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 10^4"

**raise** BaseException

**elif** number\_of\_input **==** 2 **and** (min(result) **<** **-**109 **or**

max(result) **>** 109):

final\_error **+=** "Диапазон a: -109 ≤ a ≤ 109"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

n **=** inp(1, "Введите n: ", "Введите данные согласно примеру: 1", 1)[0]

a **=** inp(n, "Введите а, b: ", "Введите данные согласно примеру: 1 2 3

4", 2)

print("Выходные данные\n" **+** str(solve(n,a)))

## Листинг кода задачи №3

**def** solve(n: int, a: list[int]):

m**=**0

c**=**0

**for** i **in** range(0,n**-**1,2):

c**+=**a[i**+**1]**-**a[i]

c**=**max(0,c)

m**=**max(c,m)

c**=**0

**for** i **in** range(1,n**-**1,2):

c**+=**a[i]**-**a[i**+**1]

c**=**max(0,c)

m**=**max(c,m)

sum**=**0

**for** i **in** range(0,n,2):

sum**+=**a[i]

**return** sum**+**m

**def** inp(n: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых

элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

"""

(1≤𝑡≤2⋅104)

(1≤𝑛≤2⋅105)

(1≤𝑎𝑖≤109)

"""

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** n:

*# Проверка количества введенных данных*

**if** number\_of\_input **==** 3:

final\_error **+=** str(" "**.**join([str(i **+** 1) **for** i **in**

range(n)]))

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 2 **\*** 10 **\*\***

4):

*# Проверка диапазонов ввода числа t*

final\_error **=** "Поддерживаемый диапазон входных данных: (1 ≤

t ≤ 10^4)"

**raise** BaseException

**elif** number\_of\_input **==** 2 **and** **not** (2 **<=** result[0] **<** 2 **\*** 10 **\*\***

5):

*# Проверка диапазонов ввода для чисел n, k, z*

final\_error **+=** "Диапазон n: 2 ≤ n ≤ 10^5"

**raise** BaseException

**elif** number\_of\_input **==** 3 **and** \

(max(result) **>** 10 **\*\*** 9 **or**

min(result) **<** 1):

*# Проверка на содержание нуля в массиве*

final\_error **=** "Диапазон ai: 1 ≤ ai ≤ 104"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

result **=** []

t **=** inp(1, "Введите t: ", "Введите данные согласно примеру: 3", 1)[0]

**for** \_ **in** range(t):

n **=** inp(1, "Введите n: ", "Введите данные согласно примеру: 3",

2)[0]

a **=** inp(n, "Введите массив a: ", "Введите данные согласно примеру:

", 3)

result**.**append(solve(n,a))

print("Выходные данные\n"**+**"\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №4

**import** math

**def** find\_path(s\_pos: list[int], e\_pos: list[int]):

""" Вспомогательный метод для поиска пути от точки до точки"""

start **=** s\_pos**.**copy()

finish **=** e\_pos**.**copy()

result **=** []

**while** start **!=** finish:

**if** finish[0] **>** start[0]:

start[0] **+=** 1

result **+=** "D"

**elif** finish[0] **<** start[0]:

start[0] **-=** 1

result **+=** "U"

**if** finish[1] **>** start[1]:

start[1] **+=** 1

result **+=** "R"

**elif** finish[1] **<** start[1]:

start[1] **-=** 1

result **+=** "L"

**return** result

**def** solve(s\_pos: list[list[int]], e\_pos: list[list[int]],n,m):

result **=** ""

**while** len(result) **<** 2 **\*** m **\*** n **and** len(s\_pos) **>** 0:

tmp **=** list(map(''**.**join, map(sorted, map(find\_path, s\_pos, e\_pos))))

**for** move **in** tmp[tmp**.**index(max(tmp))]:

**match** move:

**case** "D":

**for** el **in** s\_pos: el[0] **+=** 1 **if** math**.**fabs(el[0] **+** 1) **<=** n **else** 0

**case** "U":

**for** el **in** s\_pos: el[0] **-=** 1 **if** math**.**fabs(el[0] **-** 1) **<=** n **else** 0

**case** "R":

**for** el **in** s\_pos: el[1] **+=** 1 **if** math**.**fabs(el[1] **+** 1) **<=** n **else** 0

**case** "L":

**for** el **in** s\_pos: el[1] **-=** 1 **if** math**.**fabs(el[1] **-** 1) **<=** n **else** 0

result **+=** tmp[tmp**.**index(max(tmp))]

**for** i **in** range(len(s\_pos)**-**1,**-**1,**-**1):

**if** s\_pos[i] **==** e\_pos[i]: s\_pos**.**pop(i); e\_pos**.**pop(i)

**if** len(s\_pos) **>** 0:

**return** **-**1

**return** str(len(result)) **+** "\n" **+** result

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 **and**\

(max(result) **>** 200 **or**

min(result) **<** 1):

final\_error **=** ""

**if** **not** (1 **<=** result[0] **<** 200):

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 200"

**if** **not** (1 **<=** result[1] **<=** 200):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон m: 1 ≤ m ≤ 200"

**if** **not** (1 **<=** result[2] **<=** 200):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон k: 1 ≤ k ≤ 200"

**raise** BaseException

**elif** number\_of\_input **in** range(2,4)\

**and** (**not** (1 **<=** result[0] **<=** n) \

**or** **not** (1 **<=** result[1] **<=** m)):

*# Проверка диапазонов ввода для чисел sxi, syi, fxi, fyi*

final\_error **=** ""

litera **=** "s" **if** number\_of\_input **==** 2 **else** "f"

**if** **not** (1 **<=** result[0] **<=** n):

final\_error **+=** "Диапазон "**+**litera**+**"xi: 1 ≤ " **+** litera **+** "xi ≤ n"

**if** **not** (1 **<=** result[1] **<=** m):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон "**+**litera**+**"yi: 1 ≤ " **+** litera **+** "yi ≤ m"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

**global** n,m,k

result **=** []

**for** \_ **in** range(2):

n, m, k **=** inp(3, "Введите n, m, k (через пробел): ", "Введите данные согласно примеру: 1 2 3", 1)

s\_pos **=** [inp(2,"Введите начальные координаты точки: ","Введите данные согласно примеру: 1 2",2) **for** \_ **in** range(k)]

e\_pos **=** [inp(2,"Введите конечные координаты точки: ","Введите данные согласно примеру: 1 2",3) **for** \_ **in** range(k)]

result**.**append(solve(s\_pos,e\_pos,n,m))

print("\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №5

**from** math **import** inf

**def** solve(times: list[int]):

sums **=** [[] **for** \_ **in** range(4)]

**for** i **in** range(4):

times[i]**.**sort()

sums[i]**.**append(0)

**for** it **in** times[i]:

sums[i]**.**append(sums[i][len(sums[i]) **-** 1] **+** it)

result **=** inf

**for** count **in** range(min([k **+** 1, len(sums[3])])):

**if** (k **-** count **<** len(sums[1]) **and** k **-** count **<** int(len(sums[2]))):

result **=** min([result, sums[3][count] **+** sums[1][k **-** count] **+** sums[2][k **-** count]])

**if** result **==** inf:

result **=** **-**1

**return** result

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 **and**\

**not** (1 **<=** result[1] **<=** result[0] **<** 10 **\*\*** 5):

*# Проверка ввода n и k*

final\_error **=** ""

**if** **not** (1 **<=** result[0] **<** 10 **\*\*** 5):

final\_error **+=** "Диапазон : 1 ≤ n ≤ 10^5"

**if** **not** (1 **<=** result[1] **<=** result[0]):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон k: 1 ≤ k ≤ n"

**raise** BaseException

**elif** number\_of\_input **==** 2\

**and** (**not** (1 **<=** result[0] **<=** 10 **\*\*** 4)

**or** **not** (0 **<=** result[1] **<=** 1)

**or** **not** (0 **<=** result[2] **<=** 1)):

*# Проверка диапазонов ввода для чисел ti, ai, bi*

final\_error **=** ""

**if** **not** (1 **<=** result[0] **<=** 10 **\*\*** 4):

final\_error **+=** "Диапазон ti: 1 ≤ ti ≤ 10 \*\* 4"

**if** **not** (0 **<=** result[1] **<=** 1):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон ai: 0 ≤ k ≤ 1"

**if** **not** (0 **<=** result[2] **<=** 1):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон bi: 0 ≤ k ≤ 1"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

result **=** []

**for** \_ **in** range(3):

n, k **=** inp(2, "Введите n, k (через пробел): ", "Введите данные согласно примеру: 1 2", 1)

times **=** [[] **for** \_ **in** range(4)]

**for** i **in** range(n):

t, a, b **=** inp(3, "Введите ti, ai, bi (через пробел): ", "Введите данные согласно примеру: 1 2 3", 2)

times[a **\*** 2 **+** b]**.**append(t)

result**.**append(solve(times))

print("\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №6

**def** solve(n: int, a: int, b: int, vh: list[int]) **->** int:

sm **=** sum(vh)

rs **=** sm **\*** b

s **=** 0

**for** j **in** range(n):

s **+=** vh[j]

rs **=** min(rs, (a **+** b) **\*** vh[j] **+** (sm **-** s **-** (n **-** j **-** 1) **\*** vh[j]) **\*** b)

**return** rs

**def** inp(n: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

*# t (1≤t≤10 ^ 3)*

*# n, a и b (1≤n≤2⋅105; 1≤a,b≤10^5).*

*# (1≤xn≤108).*

final\_error **=** mess\_error

**if** number\_of\_input **==** 3:

final\_error **+=** str(" "**.**join([str(i **+** 1) **for** i **in** range(n)]))

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** n:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 10 **\*\*** 3):

*# Проверка диапазонов ввода числа t*

final\_error **=** "Поддерживаемый диапазон входных данных: (1 ≤ t ≤ 10^3)"

**raise** BaseException

**elif** number\_of\_input **==** 2 \

**and** (**not** (2 **<=** result[0] **<** 2 **\*** 10 **\*\*** 5)

**or** **not** (1 **<=** result[1] **<=** 10 **\*\*** 5)

**or** **not** (1 **<=** result[2] **<=** 10 **\*\*** 5)):

*# Проверка диапазонов ввода для чисел n, k, z*

final\_error **=** ""

**if** **not** (2 **<=** result[0] **<** 2 **\*** 10 **\*\*** 5):

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 2 \* 10^5"

**if** **not** (1 **<=** result[1] **<=** 10 **\*\*** 5):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон a: 1 ≤ a ≤ 10 \*\* 5"

**if** **not** (1 **<=** result[2] **<=** 10 **\*\*** 5):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон a: 1 ≤ a ≤ 10 \*\* 5"

**raise** BaseException

**elif** number\_of\_input **==** 3 **and** \

(max(result) **>** 108 **or**

min(result) **<** 1):

*# Проверка на содержание нуля в массиве*

final\_error **=** "Диапазон xi: 1 ≤ xi ≤ 108"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

result **=** []

t **=** inp(1, "Введите t: ", "Введите данные согласно примеру: 6", 1)[0]

**for** i **in** range(t):

n, a, b **=** inp(3, "Введите n, a, b (через пробел): ", "Введите данные согласно

примеру: 3 2 1", 2)

vh **=** inp(n, "Введите массив a: ", "Введите данные согласно примеру: ", 3)

result**.**append(solve(n,a,b,vh))

print("\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №7

**from** math **import\***

**def** gcd(x, y):

**if**(y **==** 0):

**return** x

**else**:

**return** gcd(y, x **%** y)

**def** solve(a: list[int], b: list[int]) **->** list[int]:

s **=** 0

**for** i **in** a[1:]:

s **=** gcd(s, a[0] **-** i)

result **=** []

**for** i **in** b:

result**.**append(gcd(s, a[0] **+** i))

**return** result

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 \

**and** **not** (1 **<=** result[0] **<=** 2 **\*** 10 **\*\*** 5)\

**or** **not** (1 **<=** result[1] **<=** 2 **\*** 10 **\*\*** 5):

final\_error **=** ""

**if** **not** (1 **<=** result[0] **<** 10 **\*\*** 5):

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 10^5"

**if** **not** (1 **<=** result[1] **<=** 10 **\*\*** 5):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон m: 1 ≤ m ≤ 10^5"

**raise** BaseException

**elif** number\_of\_input **in** range(2,4)\

**and** (**not** (1 **<=** result[0] **<=** 10 **\*\*** 18) \

**or** **not** (1 **<=** result[1] **<=** 10 **\*\*** 18)):

*# Проверка диапазонов ввода для чисел sxi, syi, fxi, fyi*

litera **=** "a" **if** number\_of\_input **==** 2 **else** "b"

**if** (min(result) **<** 1 **or** max(result) **>** 10 **\*\*** 19):

final\_error **=** "Диапазон "**+**litera**+**"i: 1 ≤ " **+** litera **+** "i ≤ 10^18"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

n, m **=** inp(2, "Введите n, m (через пробел): ", "Введите данные согласно примеру:

1 2", 1)

a **=** inp(n, "Введите массив a (через пробел): ", "Введите данные согласно примеру:

1 2 3", 2)

b **=** inp(m, "Введите массив b (через пробел): ", "Введите данные согласно примеру:

1 2 3", 3)

print("Выходные данные\n"**+**"\n"**.**join(str(el) **for** el **in** solve(a,b)))

## Листинг кода задачи №8

**def** solve(l, r):

**global** p, mx

d**=**p

**while** l **>=** 0 **and** r **<** n:

d **-=** (a[l] **-** a[r]) **\*** (b[l] **-** b[r])

mx **=** max(mx, d)

l **-=** 1

r **+=** 1

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**if** number\_of\_input **in** range(2,4):

final\_error **+=** str(" "**.**join([str(i **+** 1) **for** i **in** range(n)]))

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 5 **\*** 10 **\*\*** 3):

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 5000"

**raise** BaseException

**elif** number\_of\_input **in** range(2,4)\

**and** (**not** (1 **<=** result[0] **<=** 10 **\*\*** 7) \

**or** **not** (1 **<=** result[1] **<=** 10 **\*\*** 7)):

*# Проверка диапазонов ввода для чисел sxi, syi, fxi, fyi*

litera **=** "a" **if** number\_of\_input **==** 2 **else** "b"

**if** (min(result) **<** 1 **or** max(result) **>** 10 **\*\*** 7):

final\_error **=** "Диапазон "**+**litera**+**"i: 1 ≤ " **+** litera **+** "i ≤ 10^7"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

result **=** []

**for** \_ **in** range(int(input())):

n **=** inp(1, "Введите n: ", "Введите данные согласно примеру: 1", 1)[0]

a **=** inp(n, "Введите a: ", "Введите данные согласно примеру: 1 2 3", 2)

b **=** inp(n, "Введите b: ", "Введите данные согласно примеру: 1 2 3", 3)

mx **=** p **=** sum(a[i] **\*** b[i] **for** i **in** range(n))

**for** i **in** range(n):

solve(i **-** 1, i **+** 1)

solve(i, i **+** 1)

result**.**append(mx)

print("Выходные данные\n"**+**"\n"**.**join(str(el) **for** el **in** result))

## Листинг кода задачи №9

**from** math **import** fabs

**def** solve(a: int,b: int,n: int) **->** str:

**if** a **==** 0:

**return** 0 **if** b **==** 0 **else** 'No solution'

**elif** b **%** a:

**return** 'No solution'

**else**:

**for** x **in** range(**-**1000, 1001):

**if** x **\*\*** n **==** b **//** a:

**return** x

**return** 'No solution'

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**raise** BaseException

**if** number\_of\_input **==** 1 **and**\

(**not** (fabs(result[0]) **<=** 1000) **or**\

**not** (fabs(result[1]) **<=** 1000) **or**

**not** (1 **<=** result[2] **<=** 10)):

*# A, B, n, удовлетворяющих условиям: |A|,|B|≤1000, 1≤n≤10.*

final\_error **=** ""

**if** **not** (fabs(result[0]) **<=** 1000):

final\_error **+=** "Диапазон a: |A| ≤ 1000"

**if** **not** (fabs(result[1]) **<=** 1000):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон b: |B| ≤ 1000"

**if** **not** (1 **<=** result[2] **<=** 10):

final\_error **+=** "\n" **if** final\_error **!=** "" **else** ""

final\_error **+=** "Диапазон n: 1 ≤ n ≤ 10"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

**if** \_\_name\_\_ **==** "\_\_main\_\_":

result **=** []

**for** \_ **in** range(int(input())):

a, b, n **=** inp(3, "Введите a, b, n (через пробел): ", "Введите данные

согласно примеру: 1 2 3", 1)

result**.**append(solve(a,b,n))

print("Выходные данные\n" **+** "\n"**.**join([str(el) **for** el **in** result]))

## Листинг кода задачи №10

**from** collections **import** Counter

**def** solve(Sum: int) **->** bool:

**if** d[Sum] **>** 0:

d[Sum] **-=** 1

**return** **True**

**if** Sum **==** 1: **return** **False**

**return** solve(Sum **//** 2) **and** solve((Sum **+** 1) **//** 2)

**def** inp(count: int, mess: str, mess\_error: str, number\_of\_input: int):

""" Метод ввода данных. Принимает на вход количетсво вводимых элементов,

сообщение для ввода пользователем, стандартное сообщение об ошибке и номер ввода. """

result **=** []

**while** **True**:

**try**:

final\_error **=** mess\_error

result[:] **=** map(int, input(mess)**.**strip()**.**split())

**if** len(result) **!=** count:

*# Проверка количества введенных данных*

**if** number\_of\_input **==** 3:

final\_error **+=** str(" "**.**join([str(i **+** 1) **for** i **in** range(count)]))

**raise** BaseException

**if** number\_of\_input **==** 1 **and** **not** (1 **<=** result[0] **<=** 10 **\*\*** 4):

*# Проверка диапазонов ввода числа t*

final\_error **=** "Поддерживаемый диапазон входных данных: (1 ≤ t ≤

10^4)"

**raise** BaseException

**elif** number\_of\_input **==** 2 **and** **not** (1 **<=** result[0] **<** 2 **\*** 10 **\*\*** 4):

*# Проверка диапазонов ввода для чисел n, k, z*

final\_error **+=** "Диапазон n: 2 ≤ n ≤ 2 \* 10^5"

**raise** BaseException

**elif** number\_of\_input **==** 3 **and** \

(max(result) **>** 10 **\*\*** 9 **or**

min(result) **<** 1):

*# Проверка на содержание нуля в массиве*

final\_error **=** "Диапазон ai: 1 ≤ ai ≤ 10^9"

**raise** BaseException

**break** *# Если вводимые данные являются верными - выход из цикла*

**except** BaseException:

print(final\_error)

**return** result

"""Первая строка содержит одно целое число 𝑡 (1≤𝑡≤104) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число 𝑛 (1≤𝑛≤2⋅105).

Вторая строка содержит 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (1≤𝑎𝑖≤109)."""

**if** \_\_name\_\_ **==** "\_\_main\_\_":

**global** d

result **=** []

t **=** inp(1, "Введите t: ", "Введите данные согласно примеру: 1", 1)[0]

**for** \_ **in** range(t):

n **=** inp(1, "Введите n: ", "Введите данные согласно примеру: 1", 2)[0]

a **=** inp(n, "Введите массив a: ", "Введите данные согласно примеру: ", 3)

d **=** Counter(a)

result**.**append("YES" **if** solve(sum(a)) **else** "NO")

print("\n"**.**join(str(el) **for** el **in** result))

# Вывод

В ходе проделанной курсовой работы я научился решать более сложные задачи и выполнил 10 алгоритмических задач на данном языке.