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»

(МТУСИ)

Кафедра «Математическая кибернетика и информационные технологии»

Курсовая работа по дисциплине

«Структуры и алгоритмы обработки данных»

Выполнила: студентка группы БФИ2001

Вострикова П.В.

Вариант №3

Проверил: Симонов С.Е.

Москва, 2022

**ОГЛАВЛЕНИЕ**

[**Цель работы.** 3](#_Toc103104323)

[**Задание №1.** 3](#_Toc103104324)

[**Задание №2.** 4](#_Toc103104325)

[**Задание №3.** 5](#_Toc103104326)

[**Задание №4.** 6](#_Toc103104327)

[**Задание №5.** 8](#_Toc103104328)

[**Задание №6.** 11](#_Toc103104329)

[**Задание №7.** 14](#_Toc103104330)

[**Задание №8.** 15](#_Toc103104331)

[**Задание №9.** 17](#_Toc103104332)

[**Задание №10.** 18](#_Toc103104333)

[**Вывод.** 20](#_Toc103104334)

[ПРИЛОЖЕНИЕ А 21](#_Toc103104335)

[Листинг 1 – Решение задачи №1 21](#_Toc103104336)

[Листинг 2 – Решение задачи №2 22](#_Toc103104337)

[Листинг 3 – Решение задачи №3 23](#_Toc103104338)

[Листинг 4 – Решение задачи №4 24](#_Toc103104339)

[Листинг 5 – Решение задачи №5 25](#_Toc103104340)

[Листинг 6 – Решение задачи №6 26](#_Toc103104341)

[Листинг 7 – Решение задачи №7 28](#_Toc103104342)

[Листинг 8 – Решение задачи №8 28](#_Toc103104343)

[Листинг 9 – Решение задачи №9 29](#_Toc103104344)

[Листинг 10 – Решение задачи №10 29](#_Toc103104345)

# **Цель работы.**

Необходимо выполнить 10 заданий в соответствии с вариантом №3.

# **Задание №1.**

**Условие:** Ехаб известен своей любовью к определенным операциям. Сейчас у него есть массив 𝑎 длины 𝑛, и он может делать на нем следующую операцию:

Он выбирает 2 соседних элемента; Ехаб удаляет их из массива и вставляет на их место одно число: их исключающее ИЛИ. Обратите внимание, длина массива уменьшается на один.  
Теперь он спрашивает вас: возможно ли сделать все элементы массива равными? Так как он хочет сделать вашу жизнь сложнее, то вводит дополнительное ограничение: в массиве должно остаться хотя бы 2 элемента.

**Входные данные**  
В первой строке записано одно целое число 𝑡 (1≤𝑡≤15) — количество наборов входных данных

В первой строке каждого набора входных данных записано целое число 𝑛 (2≤𝑛≤2000) — размер массива 𝑎.

Во второй строке каждого набора входных данных записано 𝑛 целых чисел 𝑎1, 𝑎2, …, 𝑎𝑛 (0≤𝑎𝑖<230) — элементы массива 𝑎.

**Выходные данные**  
Если Ехаб может сделать все элементы массива равными, применяя описанную операцию, так, чтобы в массиве осталось хотя бы 2 элемента, выведите «YES». Иначе выведите «NO».

**Пример**

входные данные  
2  
3  
0 2 2  
4  
2 3 1 10

выходные данные  
YES  
NO  
Примечание: В первом примере мы можем удалить первые 2 элемента, 0 и 2, и вставить вместо них 0⊕2=2. Получится массив [2,2], в котором все элементы равны. Во втором примере ответа не существует.

**Решение:**

Код для решения задания представлен в листинге 1 (приложение A).

На рисунке 1 продемонстрирована работа программы:

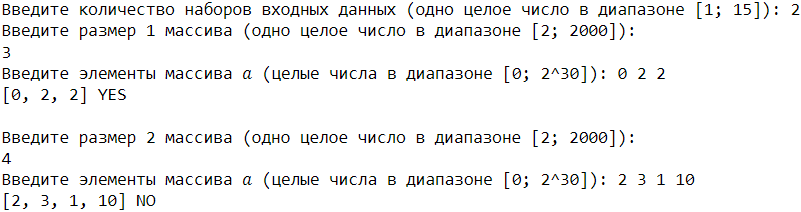


Рисунок 1 – Выполнение задания №1

# **Задание №2.**

**Условие:** Школьник Вася получил в подарок на день рождения строку длины n, состоящую из букв «a» и «b». Вася называет привлекательностью строки максимальную длину подстроки (последовательности соседних символов), состоящей из одинаковых символов.

Вася может поменять в исходной строке не более k символов. Какой максимальной привлекательности данной строки он сможет добиться?

**Входные данные**  
В первой строке входных данных содержатся два числа n и k (1 ≤ n ≤ 100 000, 0 ≤ k ≤ n) — длина строки Васи и максимальное разрешённое количество изменений.

Во второй строке следует строка Васи, состоящая из букв «a» и «b».

**Выходные данные**  
Выведите единственное целое число — максимальная привлекательность строки, которую Вася может получить, изменив в исходной строке не более k символов.

**Примеры**  
входные данные  
4 2  
abba  
выходные данные  
4  
входные данные  
8 1  
aabaabaa  
выходные данные  
5  
Примечание  
В первом примере Вася может получить как строку «aaaa», так и строку «bbbb». Во втором примере оптимальный ответ достигается на строке «aaaaabaa» или на строке «aabaaaaa».

**Решение:**

Код для решения задания представлен в листинге 2 (приложение A).

На рисунке 2 продемонстрирована работа программы:

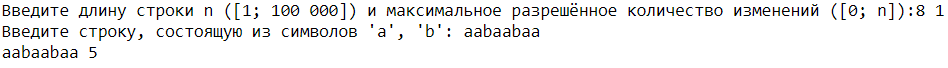


Рисунок 2 – Выполнение задания №2

# **Задание №3.**

**Условие:** Вам заданы два целых числа 𝑙 и 𝑟, где 𝑙<𝑟. Будем прибавлять 1 к 𝑙 до тех пор, пока результат не окажется равным 𝑟. Таким образом, будет сделано ровно 𝑟−𝑙 прибавлений единицы. Для каждого такого прибавления посмотрим на количество цифр, которые будут изменены при этом прибавлении. Например:

если 𝑙=909, то прибавление единицы приведёт к результату 910 и будут изменены 2 цифры;  
если к 𝑙=9 прибавить единицу, то результат будет равен 10 и будут изменены тоже 2 цифры;  
если к 𝑙=489999 прибавить единицу, то результат будет равен 490000 и будут изменены 5 цифр.  
Изменённые цифры всегда образуют некоторый суффикс десятичной записи результата.

Выведите суммарное количество изменённых цифр, если требуется из 𝑙 получить 𝑟, прибавляя каждый раз 1.

**Входные данные**  
В первой строке находится целое число 𝑡 (1≤𝑡≤104). Далее следуют 𝑡 наборов входных данных.

Каждый дабор входных данных характеризуется двумя целыми числами 𝑙 и 𝑟 (1≤𝑙<𝑟≤109).

**Выходные данные**  
Для каждого набора входных данных вычислите суммарное количество изменённых цифр, если требуется из 𝑙 получить 𝑟, прибавляя каждый раз единицу.

**Пример**  
входные данные  
4  
1 9  
9 10  
10 20  
1 1000000000  
выходные данные  
8  
2  
11  
1111111110

**Решение:**

Код для решения задания представлен в листинге 3 (приложение A).

На рисунке 3 продемонстрирована работа программы:

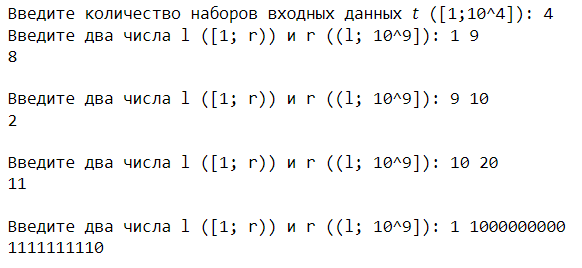


Рисунок 3 – Выполнение задания №3

# **Задание №4.**

**Условие:** Слово 𝑠 длины 𝑛 называется 𝑘-полным, если

𝑠 — палиндром, то есть 𝑠𝑖=𝑠𝑛+1−𝑖 для всех 1≤𝑖≤𝑛; 𝑠 имеет период 𝑘, то есть 𝑠𝑖=𝑠𝑘+𝑖 для всех 1≤𝑖≤𝑛−𝑘. Например, «abaaba» — это 3-полное слово, а «abccba» нет.

Бобу вручили слово 𝑠 длины 𝑛, состоящее только из строчных букв латинского алфавита, и целое число 𝑘 такое, что 𝑛 делится на 𝑘. Он хочет превратить слово 𝑠 в любое 𝑘-полное слово.

Для этого Боб может выбирать некоторую позицию 𝑖 (1≤𝑖≤𝑛) и заменять букву на позиции 𝑖 на любую другую строчную букву латинского алфавита.

Поэтому теперь Боба интересует минимальное количество позиций, буквы на которых ему необходимо заменить, чтобы превратить 𝑠 в любое 𝑘-полное слово.

Обратите внимание, что Боб может сделать ноль изменений, если слово 𝑠 уже 𝑘-полное.

Требуется ответить на 𝑡 независимых наборов входных данных.

**Входные данные**  
В первой строке записано одно целое число 𝑡 (1≤𝑡≤105) — количество наборов входных данных.

В первой строке каждого набора входных данных записаны два целых числа 𝑛 и 𝑘 (1≤𝑘<𝑛≤2⋅105, 𝑛 делится на 𝑘).

Во второй строке записано слово 𝑠 длины 𝑛.

Гарантируется, что слово 𝑠 состоит только из строчных букв латинского алфавита. Также гарантируется, что сумма 𝑛 по всем наборам входных данных не превосходит 2⋅105.

**Выходные данные**  
Для каждого набора входных данных выведите одно целое число, обозначающее минимальное количество позиций, буквы на которых ему придется заменить, чтобы превратить 𝑠 в любое 𝑘-полное слово.

**Пример**  
входные данные  
4  
6 2  
abaaba  
6 3  
abaaba  
36 9  
hippopotomonstrosesquippedaliophobia  
21 7  
wudixiaoxingxingheclp  
выходные данные  
2  
0  
23  
16  
Примечание  
В первом наборе входных данных одно из оптимальных решений — это «aaaaaa».Во втором наборе входных данных слово уже 𝑘-полное.

**Решение:**

Код для решения задания представлен в листинге 4 (приложение A).

На рисунке 4 продемонстрирована работа программы:

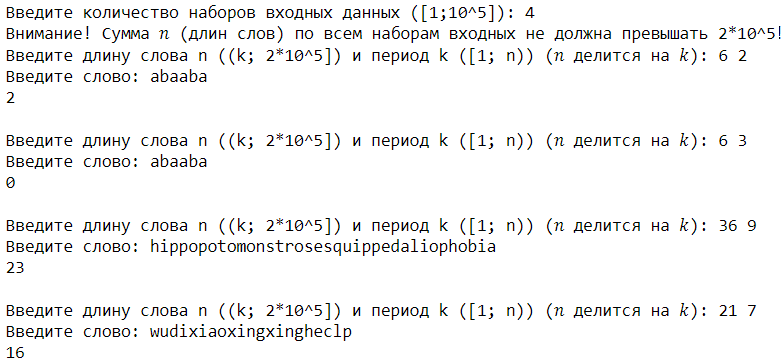


Рисунок 4 – Выполнение задания №4

# **Задание №5.**

**Условие:**

Гильдонг — владелец ресторана Пулькоги. Ресторан пользуется большим спросом, и часто посетители хотят забронировать столик до посещения.

Гильдонг так старается угодить всем посетителям, что он даже запомнил предпочитаемые диапазоны температуры всех посетителей! Смотря на список бронирований, он хочет удовлетворить всех посетителей, контролируя температуру ресторана.

В ресторане стоит кондиционер, у которого есть три состояния: выключенное, нагрев, и охлаждение. Когда кондиционер выключен, температура в ресторане не изменяется. Когда включен нагрев, температура увеличивается на 1 каждую минуту. Наконец, когда включено охлаждение, температура уменьшается на 1 каждую минуту. Гильдонг может переключать состояние сколько угодно раз, в любые целочисленные минуты. Кондиционер исходно выключен.

Каждый посетитель характеризуется тремя значениями: 𝑡𝑖 — момент визита (в минутах) 𝑖-го посетителя, 𝑙𝑖 — нижняя граница предпочитаемого диапазона температур, and ℎ𝑖 — верхняя граница предпочитаемого диапазона температур.

Каждый посетитель удовлетворен, если температура находится в их предпочтительном диапазоне в момент их посещения ресторана. Формально, 𝑖-й посетитель удовлетворён, если и только если в минуту 𝑡𝑖 температура находится в отрезке от 𝑙𝑖 до ℎ𝑖, включительно.

Вам дана исходная температура, список времен посещения посетителей и их предпочитаемые диапазоны температуры, помогите ему удовлетворить всех посетителей.

**Входные данные**

Каждый тест состоит из одного или большего числа наборов входных данных. В первой строке записано количество наборов входных данных 𝑞 (1≤𝑞≤500). Затем следуют их описания.

В первой строке каждого набора входных данных записаны два целых числа 𝑛 и 𝑚 (1≤𝑛≤100, −109≤𝑚≤109), где 𝑛 это число посетителей и 𝑚 это изначальная температура ресторана.

Затем следуют 𝑛 строк. 𝑖-я из них содержит три целых числа 𝑡𝑖, 𝑙𝑖, и ℎ𝑖 (1≤𝑡𝑖≤109, −109≤𝑙𝑖≤ℎ𝑖≤109), где 𝑡𝑖 это момент визита 𝑖-го посетителя, 𝑙𝑖 это нижняя граница предпочитаемого диапазона температур, а ℎ𝑖 это верхняя граница предпочитаемого диапазона температур. Все границы диапазонов — включительны.

Посетители даны в порядке неубывания момента посещения, а текущее время — 0.

**Выходные данные**  
Для каждого набора входных данных выведите «YES» если возможно удовлетворить потребности всех посетителей. В противном случае, выведите «NO».

Вы можете выводить все буквы в любом регистре (верхнем или нижнем).

**Пример**  
входные данные  
4  
3 0  
5 1 2  
7 3 5  
10 -1 0  
2 12  
5 7 10  
10 16 20  
3 -100  
100 0 0  
100 -50 50  
200 100 100  
1 100  
99 -100 0  
выходные данные  
YES  
NO  
YES  
NO  
Примечание  
В первом наборе входных данных примера Гильдонг может контролировать кондиционер, чтобы удовлетворить всех посетителей, следующим образом:

В 0-ю минуту, поменяйте состояние на нагрев (температура 0).  
Во 2-ю мунуту, поменяйте состояние на выключенное (температура 2).  
В 5-ю минуту, поменяйте состояние на нагрев (температура 2, 1-й посетитель уже удовлетворен).  
В 6-ю минуту, поменяте состояние на выключенное (температура 3).  
В 7-ю минуту, поменяйте состояние на охлаждение (температура 3, 2-й посетитель удовлетворен).  
В 10-ю минуту, температура будет 0, что удовлетворит последнего посетителя.  
В третьем наборе входных данных примера поменяйте состояние на выключенное в 0-ю муинуту и оставьте его таким. Тогда все посетители будут удовлетворены. Обратите внимание, что момент посещения 1-го посетителя совпадает с моментом посещения 2-го посетителя.

Во втором и четвертом наборах входных данных примера Гильдонгу придется оставить хотя бы одного посетителя неудовлетворенным.

**Решение:**

Код для решения задания представлен в листинге 5 (приложение A).

На рисунке 5 продемонстрирована работа программы:

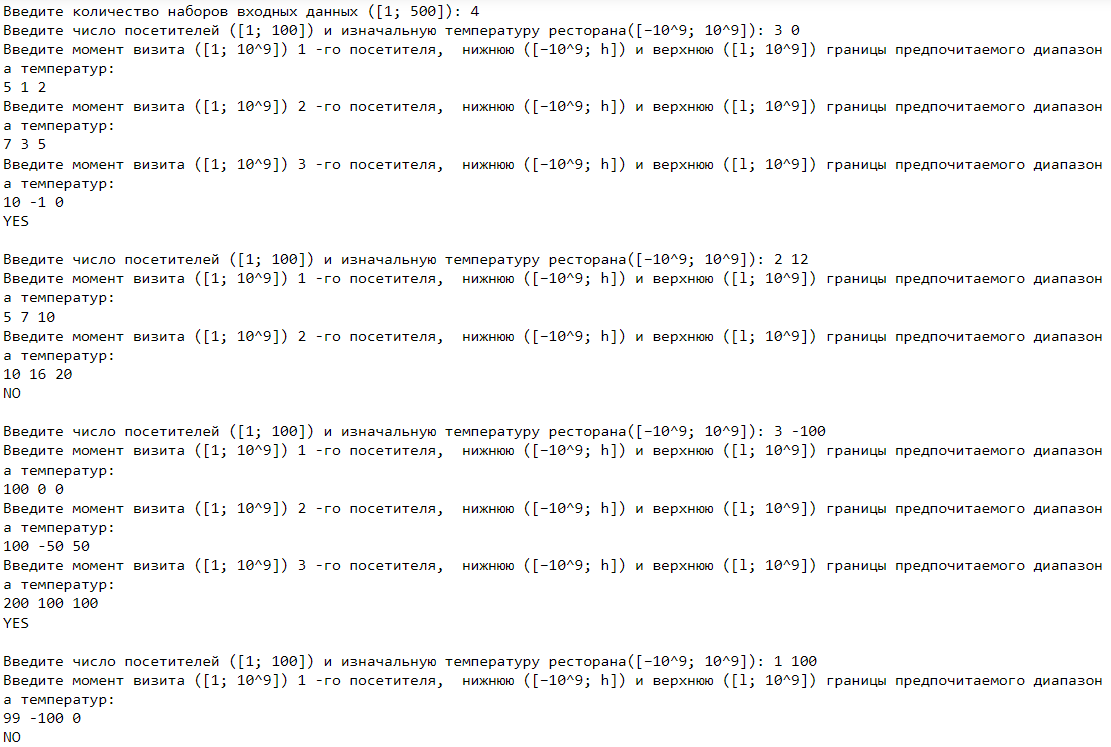


Рисунок 5 – Выполнение задания №5

# **Задание №6.**

**Условие:**

Бинарная строка это строка, состоящая только из символов 0 и 1. Бинарная строка называется 𝑘-сбалансированной, если каждая подстрока длины 𝑘 этой бинарной строки содержит равное количество символов 0 и 1 (𝑘/2 каждого).

Вам дается целое число 𝑘 и строка 𝑠, состоящая только из символов 0, 1 и ?. Вам необходимо определить, можно ли получить 𝑘-сбалансированную бинарную строку, заменив каждый символ ? в 𝑠 либо на 0, либо на 1.

Строка 𝑎 является подстрокой 𝑏, если 𝑎 может быть получена из 𝑏 удалением нескольких (возможно, ни одного или всех) символов из начала и нескольких (возможно, ни одного или всех) символов из конца.

**Входные данные**  
Каждый тест содержит несколько наборов входных данных. В первой строке указано количество наборов входных данных 𝑡 (1≤𝑡≤104). Описание наборов входных данных приведено ниже.

Первая строка каждого набора входных данных содержит два целых числа 𝑛 и 𝑘 (2≤𝑘≤𝑛≤3⋅105, 𝑘 чётное) — длина строки и параметр для сбалансированной бинарной строки.

Следующая строка содержит строку 𝑠 (|𝑠|=𝑛). При этом 𝑠 состоит только из 0, 1 и ?.

Гарантируется, что сумма 𝑛 по всем наборам входных данных не превышает 3⋅105.

**Выходные данные**  
Для каждого набора входных данных выведите YES, если мы можем заменить каждый ? в 𝑠 на 0 или 1 так, чтобы получившаяся бинарная строка была 𝑘-сбалансированной, или NO если это невозможно.

**Пример**  
входные данные  
9  
6 4  
100110  
3 2  
1?1  
3 2  
1?0  
4 4  
????  
7 4  
1?0??1?  
10 10  
11??11??11  
4 2  
1??1  
4 4  
?0?0  
6 2  
????00  
выходные данные  
YES  
NO  
YES  
YES  
NO  
NO  
YES  
NO  
Примечание  
В первом наборе входных данных строка уже является 4-сбалансированной бинарной строкой.

Во втором наборе входных данных строка может быть преобразована в 101.

В третьем наборе входных данных строку можно преобразовать в 0110.

В четвертом наборе входных данных строку можно преобразовать в 1100110.

**Решение:**

Код для решения задания представлен в листинге 6 (приложение A).

На рисунке 6 продемонстрирована работа программы:

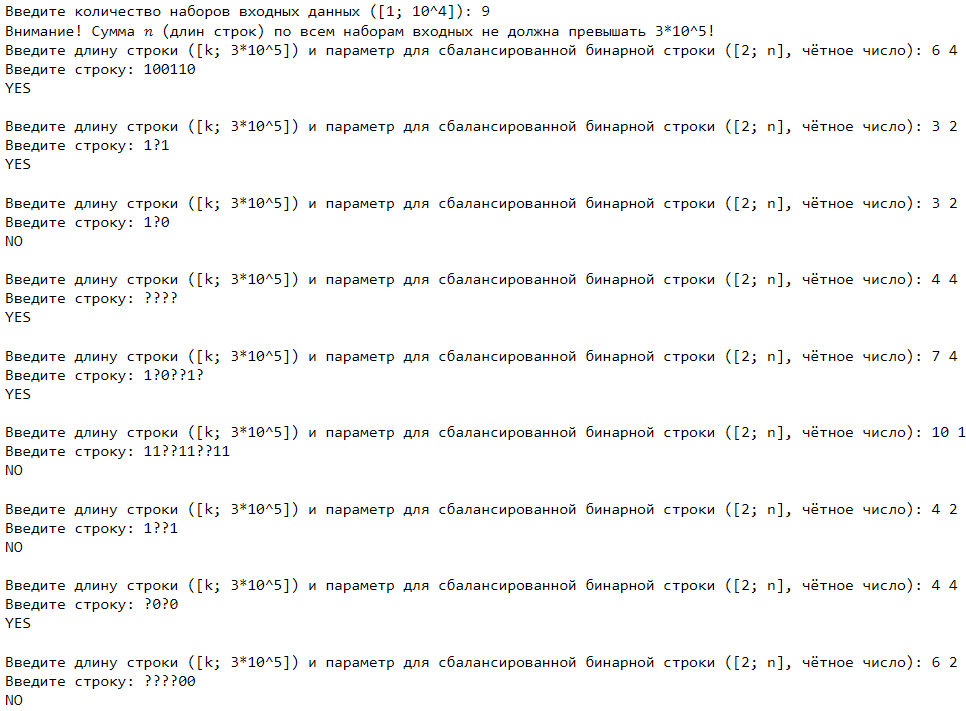


Рисунок 6 – Выполнение задания №6

# **Задание №7.**

**Условие:**

Перестановка длины 𝑛 — это массив, состоящий из 𝑛 различных целых чисел от 1 до 𝑛 в произвольном порядке. Например, [2,3,1,5,4] — это перестановка, но [1,2,2] — это не перестановка (2 встречается дважды в массиве), а [1,3,4] также не является перестановкой (𝑛=3, но в массиве встречается 4).

Рассмотрим перестановку 𝑝 длины 𝑛. Построим граф на 𝑛 вершинах, используя перестановку следующим образом:

Для каждого 1≤𝑖≤𝑛 найдите наибольшее значение 𝑗, для которого 1≤𝑗<𝑖 и 𝑝𝑗>𝑝𝑖, и добавьте неориентированное ребро между вершинами 𝑖 и 𝑗.

Для каждого 1≤𝑖≤𝑛 найдите наименьшее значение 𝑗, для которого 𝑖<𝑗≤𝑛 и 𝑝𝑗>𝑝𝑖, и добавьте неориентированное ребро между вершинами 𝑖 и 𝑗.

В тех случаях, когда таких 𝑗 не существует, мы не добавляем ребер. Также обратите внимание, что мы проводим ребра между соответствующими индексами, а не значениями в этих индексах.

Например, рассмотрим случай 𝑛=4 и 𝑝=[3,1,4,2]; здесь ребрами графа являются (1,3),(2,1),(2,3),(4,3).

Перестановка 𝑝 является циклической, если граф, построенный с использованием 𝑝, имеет хотя бы один простой цикл.

Для данного 𝑛, найдите число циклических перестановок длины 𝑛. Поскольку число может быть очень большим, выведите его по модулю 109+7.

Пожалуйста, обратитесь к разделу Примечания для формального определения простого цикла.

**Входные данные**  
Первая и единственная строка содержит одно целое число 𝑛 (3≤𝑛≤106).

**Выходные данные**  
Выведите единственное целое число 0≤𝑥<109+7, количество циклических перестановок длины 𝑛 по модулю 109+7.

**Примеры**  
входные данные  
4  
выходные данные  
16  
входные данные  
583291  
выходные данные  
135712853  
Примечание  
Для 𝑛=4 существует 16 циклических перестановок. [4,2,1,3] — одна из таких перестановок, она содержит цикл длины четыре: 4→3→2→1→4.

Вершины 𝑣1, 𝑣2, …, 𝑣𝑘 образуют простой цикл, если выполняются следующие условия:

𝑘≥3. 𝑣𝑖≠𝑣𝑗 для любой пары индексов 𝑖 и 𝑗. (1≤𝑖<𝑗≤𝑘) Между 𝑣𝑖 и 𝑣𝑖+1 есть ребро для всех 𝑖 (1≤𝑖<𝑘), как и между 𝑣1 и 𝑣𝑘

**Решение:**

Код для решения задания представлен в листинге 7 (приложение A).

На рисунке 7 продемонстрирована работа программы:



Рисунок 7 – Выполнение задания №7

# **Задание №8.**

**Условие:**

Пифагорова тройка — это тройка целых чисел (𝑎,𝑏,𝑐) таких, что можно образовать прямоугольный треугольник с длинами первого катета, второго катета и гипотенузы, равными 𝑎, 𝑏 и 𝑐 соответственно. Примером пифагоровой тройки является (3,4,5).

Вася изучает свойства прямоугольных треугольников, и он использует формулу, которая определяет, является ли некоторая тройка целых чисел пифагоровой. К сожалению, он забыл точную формулу; он помнит только, что формула была каким-то уравнением с квадратами. Поэтому он придумал следующую формулу: 𝑐=𝑎^2−𝑏.

Очевидно, что это неправильная формула для проверки, является ли тройка чисел пифагоровой. Но, к удивлению Васи, его формула сработала на тройке (3,4,5): 5=3^2−4 поэтому по формуле Васи, это пифагорова тройка.

Когда Вася нашел правильную формулу (и понял, что его формула неверна), он задался вопросом: сколько существует троек целых чисел (𝑎,𝑏,𝑐) 1≤𝑎≤𝑏≤𝑐≤𝑛 таких, что они являются пифагоровыми как по его формуле, так и по настоящему определению? Он попросил вас посчитать количество таких троек.

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

Каждый набор состоит из одной строки, содержащей одно целое число 𝑛 (1≤𝑛≤109).

**Выходные данные**  
Для каждого теста выведите одно целое число — количество троек целых чисел (𝑎,𝑏,𝑐) 1≤𝑎≤𝑏≤𝑐≤𝑛 таких, чтобы они являются пифагоровыми как по настоящему определению, так и по формуле, которую придумал Вася.

**Пример**  
входные данные  
3  
3  
6  
9  
выходные данные  
0  
1  
1  
Примечание  
Единственная пифагорова тройка, удовлетворяющая 𝑐=𝑎2−𝑏 и 1≤𝑎≤𝑏≤𝑐≤9, является (3,4,5); поэтому ответ для 𝑛=3 равен 0, а ответ для 𝑛=6 (и для 𝑛=9) равен 1.

**Решение:**

Код для решения задания представлен в листинге 8 (приложение A).

На рисунке 8 продемонстрирована работа программы:

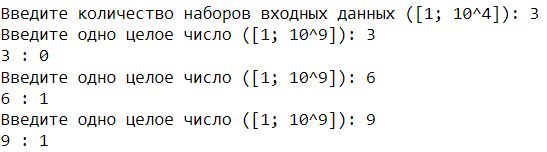


Рисунок 8 – Выполнение задания №8

# **Задание №9.**

**Условие:**

Назовем последовательность положительных чисел 𝑎0,𝑎1,...,𝑎𝑛−1 степенной последовательностью, если найдется такое положительное целое число 𝑐, что для всех 0≤𝑖≤𝑛−1, 𝑎𝑖=𝑐𝑖.

Вам дана последовательность из 𝑛 положительных чисел 𝑎0,𝑎1,...,𝑎𝑛−1, вам разрешается:

Переупорядочить последовательность (иначе говоря, выбрать перестановку 𝑝 из {0,1,...,𝑛−1} и заменить 𝑎𝑖 на 𝑎𝑝𝑖), и затем Выполнить следующую операцию любое количество раз: выбрать индекс 𝑖 и заменить 𝑎𝑖 на 𝑎𝑖−1 или 𝑎𝑖+1 (иначе говоря, уменьшить или увеличить 𝑎𝑖 на 1) за стоимость 1. Найдите минимальную стоимость, необходимую для превращения 𝑎0,𝑎1,...,𝑎𝑛−1 в степенную последовательность.

**Входные данные**  
В первой строке записано одно целое число 𝑛 (3≤𝑛≤105).

Во второй строке записаны 𝑛 целых чисел 𝑎0,𝑎1,...,𝑎𝑛−1 (1≤𝑎𝑖≤109).

**Выходные данные**  
Выведите минимальную стоимость, необходимую для превращения 𝑎0,𝑎1,...,𝑎𝑛−1 в степенную последовательность.

**Примеры**  
входные данные  
3  
1 3 2  
выходные данные  
1  
входные данные  
3  
1000000000 1000000000 1000000000  
выходные данные  
1999982505  
Примечание  
В первом примере сначала можно переупорядочить {1,3,2} в {1,2,3}, затем увеличить 𝑎2 до 4 за стоимость 1, чтобы получить степенную последовательность {1,2,4}.

**Решение:**

Код для решения задания представлен в листинге 9 (приложение A).

На рисунке 9 продемонстрирована работа программы:

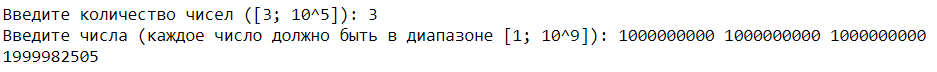


Рисунок 9 – Выполнение задания №9

# **Задание №10.**

**Условие:**

Вы и ваш друг играете в Mortal Kombat XI. Вы пытаетесь пройти башню испытаний. Всего в башне есть 𝑛 боссов, пронумерованных от 1 до 𝑛. Тип 𝑖-го босса равен 𝑎𝑖. Если 𝑖-й босс является легким, то его тип равен 𝑎𝑖=0, иначе этот босс является сложным и его тип равен 𝑎𝑖=1.

В течение одной игровой сессии вы или ваш друг можете убить одного или двух боссов (ни вы, ни ваш друг не можете пропускать сессию, поэтому минимальное количество боссов, убитых в течение сессии, равно хотя бы одному). После сессии вашего друга начинается ваша сессия, затем опять сессия вашего друга, затем опять ваша, и так далее. Первая сессия — сессия вашего друга.

Вашему другу надо научиться играть лучше, потому что на самом деле он не может убивать сложных боссов. Чтобы убивать их, он использует очки пропуска. Одно очко пропуска может быть использовано для того, чтобы убить одного сложного босса.

Ваша задача — найти минимальное количество очков пропуска, которое ваш друг должен использовать для того, чтобы вы с вашим другом убили всех 𝑛 боссов в заданном порядке.

Например: предположим, что 𝑛=8, 𝑎=[1,0,1,1,0,1,1,1]. Тогда лучшей последовательностью действий является следующая:

ваш друг убивает первых двух боссов, используя одно очко пропуска для первого босса;  
вы убиваете третьего и четвертого боссов;  
ваш друг убивает пятого босса;  
вы убиваете шестого и седьмого боссов;  
ваш друг убивает последнего босса, используя одно очко пропуска, таким образом, башня проходится с использованием двух очков пропуска. 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 входных данных содержит одно целое число 𝑛 (1≤𝑛≤2⋅105) — количество боссов. Вторая строка набора входных данных содержит 𝑛 целых чисел 𝑎1,𝑎2,…,𝑎𝑛 (0≤𝑎𝑖≤1), где 𝑎𝑖 равно типу 𝑖-го босса.

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

**Выходные данные**  
Выведите ответ на каждый набор тестовых данных: минимальное количество очков пропуска, которое ваш друг должен использовать для того, чтобы вы с вашим другом убили всех 𝑛 боссов в заданном порядке.

**Пример**  
входные данные  
6  
8  
1 0 1 1 0 1 1 1  
5  
1 1 1 1 0  
7  
1 1 1 1 0 0 1  
6  
1 1 1 1 1 1  
1  
1  
1  
0  
выходные данные  
2  
2  
2  
1  
0

**Решение:**

Код для решения задания представлен в листинге 10 (приложение A).

На рисунке 10 продемонстрирована работа программы:

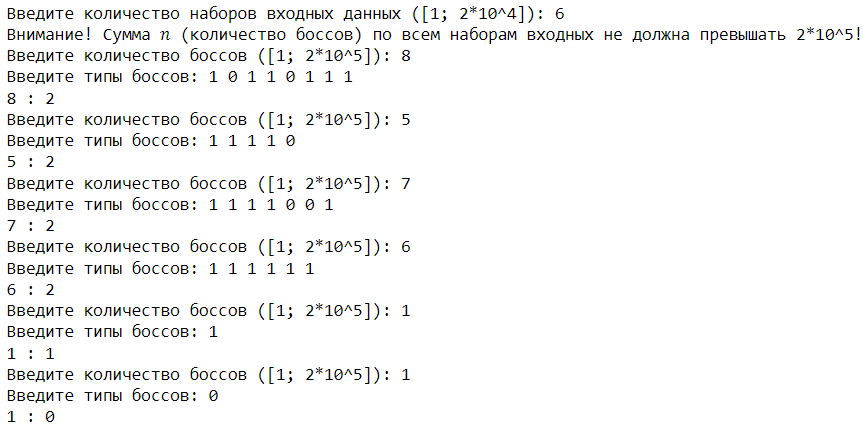


Рисунок 10 – Выполнение задания №10

# **Вывод.**

Были написаны решения для 10 задач на языке программирования python согласно варианту №3.

# ПРИЛОЖЕНИЕ А

## Листинг 1 – Решение задачи №1

def task1(a):

# делаем XOR сразу всех элементов

array\_xor = 0

for i in a:

array\_xor ^= i

# для прохождения по XOR всех эл. по очереди

temporary = 0

result = 0

# числа по очереди проходят по операции XOR (temporary) и сравниваются с array\_xor

for j in a:

temporary ^= j

# по сути, каждый раз делим список на части, при этом заменяя их XOR, и нам нужно, чтобы эти XOR были равны

if temporary == array\_xor:

result += 1

temporary = 0

if result > 1 or array\_xor == 0:

return("YES")

else:

return("NO")

try: # ловим ошибки

t = int(input('Введите количество наборов входных данных (одно целое число в диапазоне [1; 15]): '))

if t < 1 or t > 15: exit("введённое количество наборов не входит в диапазон [1; 15]")

for i in range (t):

print('Введите размер', i+1, 'массива (одно целое число в диапазоне [2; 2000]): ')

n = int(input())

if n < 2 or n > 2000: exit("введённый входные размер массива не входит в диапазон [2; 2000]")

a = list(map(int, input("Введите элементы массива 𝑎 (целые числа в диапазоне [0; 2^30]): ").split())) # принимает строку и возвращает список подстрок

if len(a) != n: exit("количествово элементов не равно заявленному.")

for el in a:

if el < 0 or el > 2\*\*30: exit("входные данные не входят в диапазон [0; 2^30]")

print(a, task1(a))

print()

except ValueError: # функция получает аргумент правильного типа, но некорректного значения (также когда вводим буквы вызывается)

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e: # исключение, поднимаемое sys.exit(), для остановки работы интерпретатора

print("Ошибка: %s" % e.code)

s = int(input('Введите количество наборов входных данных: '))

for i in range (s):

print('Введите размер', i+1, 'массива: ')

n = int(input())

a = []

for j in range (n):

print('Введите', j+1, 'элемент массива 𝑎: ')

a1 = input()

a.append(int(a1))

print(a, task1(a))

print()

## Листинг 2 – Решение задачи №2

def task2(k, s):

a = b = j = result = 0

for i in s:

# записываем сколько каких букв (a/b) на каждой итерации

if i == 'a': a += 1

else: b += 1

# больше ли максимально разрешённого кол-ва изменений?

if min(a,b) > k:

# чтобы не упустить момент, когда все допуст. изменения используем

if s[j] == 'a': a -= 1

else: b -= 1

j += 1

else:

# "позволяем" использовать это самое изменение при необходимости

result += 1

return(result)

try:

n, k = map(int, input("Введите длину строки n ([1; 100 000]) и максимальное разрешённое количество изменений ([0; n]):").split())

if n < 1 or n > 100000: exit("введённая длина строки не входит в диапазон [1; 100 000]")

elif k < 0 or k > n: exit("максимальное разрешённое количество изменений не входит в диапазон [0; n]")

s = input("Введите строку, состоящую из символов 'a', 'b': ")

if len(s) != n: exit("количествово элементов не равно заявленному.")

for el in s:

if el == 'a': continue

elif el == 'b': continue

else: exit("строка должна состоять только из символов 'a', 'b'")

print(s, task2(k, s))

except ValueError: # функция получает аргумент правильного типа, но некорректного значения (также когда вводим буквы вызывается)

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e: # исключение, поднимаемое sys.exit(), для остановки работы интерпретатора

print("Ошибка: %s" % e.code)

## Листинг 3 – Решение задачи №3

def task3(l, r):

# считаем кол-во изменений разряда единиц, каждый раз удаляя их после того, как посчитали

result = 0

while r > 0:

result += r - l

r //= 10

l //= 10

return result

try:

t = int(input('Введите количество наборов входных данных 𝑡 ([1;10^4]): '))

if t < 1 or t > 10\*\*4: exit("введённое количество наборов не входит в диапазон [1; 100 000]")

for i in range(t):

l, r = map(int, input("Введите два числа l ([1; r)) и r ((l; 10^9]): ").split())

if l < 1 or l >= r: exit("введённое число l не входит в диапазон [1; r)")

elif r <= l or r > 10\*\*9: exit("введённое число r не входит в диапазон (l; 10^9]")

print(task3(l, r))

print()

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 4 – Решение задачи №4

def task4(n, k, s):

result = 0

for i in range(k):

alphabet = [0]\*26

j = i

while j < n:

# считаем сколько каких символов у нас встречается на определённых позициях

alphabet[ord(s[j]) - 97] += 1

alphabet[ord(s[k-1-j]) - 97] += 1

j += k

# надо все буквы привести к той, которая встречается чаще всего, поэтому считаем макс кол-во вхождений и складываем

result += sum(alphabet) - max(alphabet)

return result // 2

try:

t = int(input('Введите количество наборов входных данных ([1;10^5]): '))

if t < 1 or t > 10\*\*5: exit("введённое количество наборов не входит в диапазон [1;10^5]")

print("Внимание! Сумма 𝑛 (длин слов) по всем наборам входных не должна превышать 2\*10^5!")

sum\_n = 0

for i in range(t):

n, k = map(int, input("Введите длину слова n ((k; 2\*10^5]) и период k ([1; n)) (𝑛 делится на 𝑘): ").split())

if n <= k or n > 2\*10\*\*5: exit("введённое число n не входит в диапазон (k; 2\*10^5]")

elif k < 1 or k >= n: exit("введённое число k не входит в диапазон [1; n)")

elif n%k != 0: exit("𝑛 не делится на 𝑘.")

sum\_n += n

if sum\_n > 2\*10\*\*5:

print("Внимание! Сумма 𝑛 (длин слов) по всем наборам входных достигла 2\*10^5! Дальнейшее выполнение программы невозможно.")

break

s = input("Введите слово: ")

alphabet = ["q", "w", "e", "r", "t", "y", "u", "i", "o", "p", "a", "s", "d", "f", "g", "h", "j", "k", "l", "z", "x", "c", "v", "b", "n", "m"]

if len(s) != n: exit("количествово элементов не равно заявленному.")

for el in s:

if el not in alphabet: exit("строка должна состоять только из строчных букв латинского алфавита")

print(task4(n, k, s))

print()

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 5 – Решение задачи №5

def task5(temp, array):

# текущее время

current\_time = 0

# представляет минимальную и максимальную возможную температуру, которую можно установить сейчас

low\_temp = temp

high\_temp = temp

for time, lower, high in array:

# возможный диапазон температур считаем

low\_temp -= (time - current\_time)

high\_temp += (time - current\_time)

# если предпочтительный диапазон пересекается с возможным диапазоном, то клиент удовлетворён

if lower > high\_temp or high < low\_temp: return "NO"

# обновляем

low\_temp = max(low\_temp, lower)

high\_temp = min(high, high\_temp)

current\_time = time

return "YES"

try:

q = int(input('Введите количество наборов входных данных ([1; 500]): '))

if q < 1 or q > 500: exit("введённое количество наборов не входит в диапазон [1; 500]")

for i in range(q):

n, m = map(int, input("Введите число посетителей ([1; 100]) и изначальную температуру ресторана([−10^9; 10^9]): ").split())

if n < 1 or n > 100: exit("введённое число посетителей не входит в диапазон [1; 100]")

elif m < -10\*\*9 or m > 10\*\*9: exit("введённое число изначальной температуры не входит в диапазон [−10^9; 10^9]")

array = []

for i in range(n):

print("Введите момент визита ([1; 10^9])", i+1, "-го посетителя, нижнюю ([−10^9; h]) и верхнюю ([l; 10^9]) границы предпочитаемого диапазона температур: ")

t, l, h = map(int, input().split())

if t < 1 or t > 10\*\*9: exit("введённый момент визита не входит в диапазон [1; 10^9]")

elif l < -10\*\*9 or l > h: exit("введённая нижняя граница температуры не входит в диапазон [−10^9; h]")

elif h < l or h > 10\*\*9: exit("введённая верхняя граница температуры не входит в диапазон [l; 10^9]")

array.append((t, l, h))

print(task5(m, array))

print()

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 6 – Решение задачи №6

def task6(n, k, s):

temp = ['?'] \* k

for i in range(n):

if s[i] != '?':

if temp[i % k] == '?':

# присваиваем элементу (из списка temp) значение того эл. из строки, который сейчас обрабатывается по итерации

temp[i % k] = s[i]

elif temp[i % k] != s[i]:

# не выполняется условие того, что элементы в исходной строке через k элементов повторяются

return "NO"

# смотрим, чтобы в temp было равное кол-во 0 и 1

c0 = temp.count('0')

c1 = temp.count('1')

if max(c1, c0) > k // 2:

return "NO"

return "YES"

try:

t = int(input('Введите количество наборов входных данных ([1; 10^4]): '))

if t < 1 or t > 10\*\*4: exit("введённое количество наборов не входит в диапазон [1; 10^4]")

print("Внимание! Сумма 𝑛 (длин строк) по всем наборам входных не должна превышать 3\*10^5!")

sum\_n = 0;

for i in range(t):

n, k = map(int, input("Введите длину строки ([k; 3\*10^5]) и параметр для сбалансированной бинарной строки ([2; n], чётное число): ").split())

if n < k or n > 3\*10\*\*5: exit("введённая длина строки не входит в диапазон [k; 3\*10^5]")

elif k < 2 or k > n: exit("введённый параметр для сбалансированной бинарной строки не входит в диапазон [2; n]")

elif k%2 != 0: exit("введённый параметр для сбалансированной бинарной строки не является чётным числом.")

sum\_n += n

if sum\_n > 3\*10\*\*5:

print("Внимание! Сумма 𝑛 (длин строк) по всем наборам входных достигла 3\*10^5! Дальнейшее выполнение программы невозможно.")

break

s = input("Введите строку: ")

if len(s) != n: exit("количествово элементов не равно заявленному.")

alphabet = ["0", "1", "?"]

for el in s:

if el not in alphabet: exit("строка должна состоять только из \"0\", \"1\", \"?\"")

print(task6(n, k, s))

print()

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 7 – Решение задачи №7

def task7(n):

iterations = n

temp = 1

mod = 10\*\*9 + 7

# n! по модулю создаём

while iterations :

temp = temp \* iterations % mod

iterations -= 1

return((temp - 2\*\*(n - 1)) % mod)

try:

n = int(input('Введите одно целое число ([3; 10^6]): '))

if n < 3 or n > 10\*\*6: exit("введённое число не входит в диапазон [3; 10^6]")

print(n, ":", task7(n))

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 8 – Решение задачи №8

import math

def task8(n):

# вычтем одно уравнение из другого, подставим и получим следующее

a = int(math.sqrt(2\*n-1))

return((a-1)//2)

try:

t = int(input("Введите количество наборов входных данных ([1; 10^4]): "))

if t < 1 or t > 10\*\*4: exit("введённое количество наборов не входит в диапазон [1; 10^4]")

for i in range(t):

n = int(input("Введите одно целое число ([1; 10^9]): "))

if n < 1 or n > 10\*\*9: exit("введённое целое число не входит в диапазон [1; 10^9]")

print(n, ":", task8(n))

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 9 – Решение задачи №9

import math

def task9(n, a):

# нужна для того, чтобы на каждой итерации находить min значение среди всех

result = math.inf

# то самое целое положительное число

c = 1

mx = 10\*\*10

while c\*\*(n-1) < mx:

func = 0

# мин стоимость преобразования

for i in range(n): func += abs(a[i]-c\*\*i)

c+=1

result = min(result, func)

return result

try:

n = int(input("Введите количество чисел ([3; 10^5]): "))

if n < 3 or n > 10\*\*5: exit("введённое количество наборов не входит в диапазон [3; 10^5]")

a = sorted(map(int,input("Введите числа (каждое число должно быть в диапазоне [1; 10^9]): ").split()))

if len(a) != n: exit("количествово элементов не равно заявленному.")

for el in a:

if el < 1 or n > 10\*\*9: exit("введённые числа не входят в диапазон [1; 10^9]")

print(task9(n, a))

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)

## Листинг 10 – Решение задачи №10

def task10(a):

result = 0

# первого босса рассматриваем отдельно всегда

if a[0] == '1': result += 1

# чтобы другу использовать очко пропуска, необходимо, чтобы было три подряд идущих сложных боссов

result += a[1:].count('111')

return result

try:

t = int(input("Введите количество наборов входных данных ([1; 2\*10^4]): "))

if t < 1 or t > 2\*10\*\*4: exit("введённое количество наборов не входит в диапазон [1; 2\*10^4]")

print("Внимание! Сумма 𝑛 (количество боссов) по всем наборам входных не должна превышать 2\*10^5!")

sum\_n = 0;

for i in range(t):

n = int(input("Введите количество боссов ([1; 2\*10^5]): "))

if n < 1 or n > 2\*10\*\*5: exit("введённое количество боссов не входит в диапазон [1; 2\*10^5]")

sum\_n += n

if sum\_n > 2\*10\*\*5:

print("Внимание! Сумма 𝑛 (количество боссов) по всем наборам входных достигла 2\*10^5! Дальнейшее выполнение программы невозможно.")

break

a = input("Введите типы боссов: ")

a = a.replace(" ", "")

if len(a) != n: exit("количествово элементов не равно заявленному.")

alphabet = ["0", "1"]

for el in a:

if el not in alphabet: exit("строка должна состоять только из \"0\", \"1\" ")

print(n, ":", task10(a))

except ValueError:

print("Введены неверные данные, попробуйте ещё раз.")

except SystemExit as e:

print("Ошибка: %s" % e.code)