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»

Выполнил: студент группы БСТ2001

Коцич Лазарь

Вариант 9

Руководитель:

Чайка Андрей Дмитриевич

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ3

Глава 1. Теоретическая часть5

1.1 Задача 82А8

1.2 Задача 108А9

1.3 Задача 456А10

1.4 Задача 43А11

1.5 Задача 131А12

Глава 2. Практическая часть13

2.1 Решение задачи 82А13

2.2 Решение задачи 108А15

2.3 Решение задачи 456А18

2.4 Решение задачи 43А20

2.5 Решение задачи 131А23

ЗАКЛЮЧЕНИЕ26

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ27

ПРИЛОЖЕНИЕ А28

**ВВЕДЕНИЕ**

**Актуальность работы.** На актуальность темы указывает такой фактор, как развитие информационных технологий в нашей стране. Студентам, поступившим в технический ВУЗ на направление, связанное с данными технологиями, необходимо развиваться и расширять свои знания в области программирования. ВУЗ дает базовые знания и ставит задачи перед студентами, а те в свою очередь определяют какой из языков программирования их интересует, что дает возможность им определить свои цели на будущее. Для данной работы была поставлена задача, заключающаяся в решении заданий с сайта «Codeforces», которые имеют свои формулировки, критерии и баллы, выдающиеся пользователям за решение заданий. Такая работа повышает самостоятельность студентов в решении задач, а также дает возможность самим выбрать для себя те задания, которые являются наиболее интересными и понятными для них. Таким образом, решая задания, студент повышает свое понимание в программировании на языке Python, а также развивает свои способности в данной области.

**Предмет и объект исследования.** Предметом и объектом изучения являются задания, представленные на сайте «Codeforces».

**Цель и задачи.** Цель работы – решить задачи, которые будут представлены дальше в данной работе. Следует отметить, что целью еще является и изучения дополнительных возможностей в языке программирования Python, которые помогут в решении тех или иных задач. Также можно выделить несколько пунктов для выполнения работы:

1. Выбрать задания, за которые дают 1000 баллов и более;
2. Собрать необходимый теоретический материал для их решения;
3. Реализовать все программы по каждому заданию.

**Методы исследования.** В ходе выполнения данной курсовой работы был использован теоретический метод исследования, позволяющий более явно продемонстрировать актуальность работы. Все решения заданий производились в программе «PyCharm».

**ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

В теоретической части работы будут разобраны базовые понятия, необходимые для выполнения заданий, функции, которые будут использованы, а также будут определены задачи, которые в дальнейшем будут решены. К каждой задаче будут приложены ее условия, входные и выходные данные, а также те способы решения, которые были предприняты для их решения.

Переходя к цели работы, требуется найти те задачи, которые оцениваются в 1000 баллов и более. Для этого необходимо открыть архив на сайте «Codeforces», представленный на рисунке 1.1, и выбрать задачу, удовлетворяющую данному критерию. В конечном итоге были определены следующие задания: 82А, 108A, 456A, 43A и 131А.

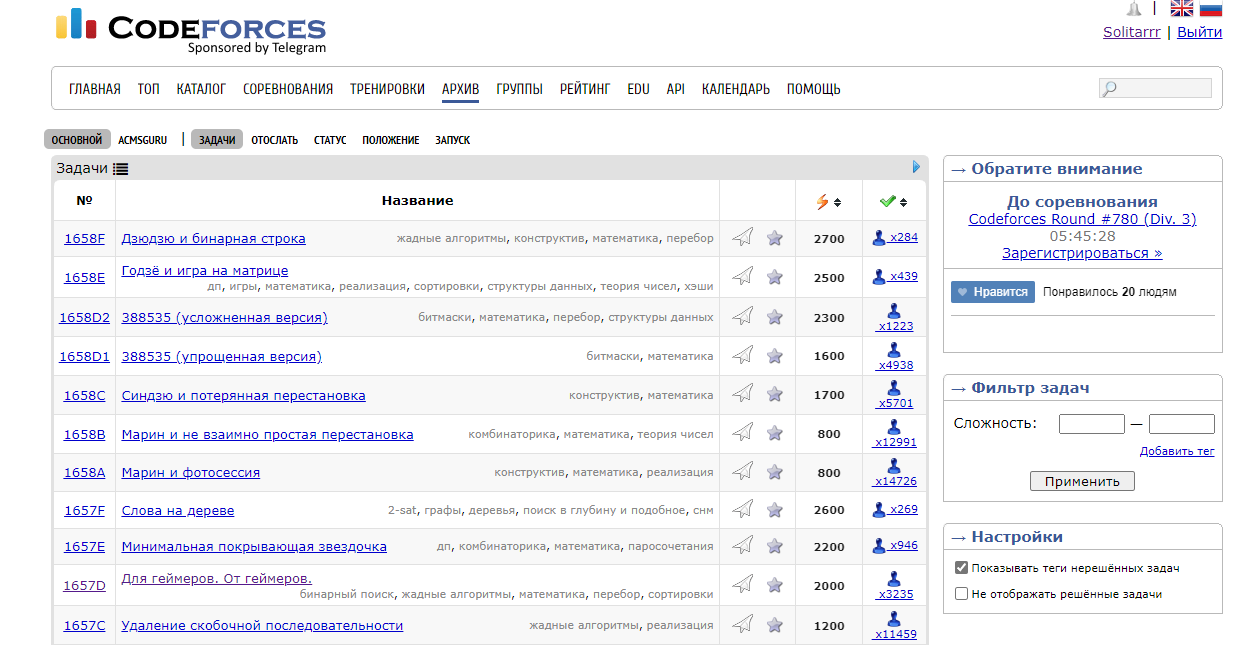


Рисунок 1.1 – Архив задач

Многие задачи, которые определены и указаны выше, используют те или иные функции, возможности Python-а. Для того, чтобы лучше понимать, что требуется от пользователя, к каждой задаче пишут так называемые «теги», в которых указаны примерные знания в Python, которыми должен обладать решающий. Пример тегов представлен на рисунке 1.2. Также на рисунке продемонстрировано то, как именно выглядят задачи, как выглядят их выходные и входные данные.

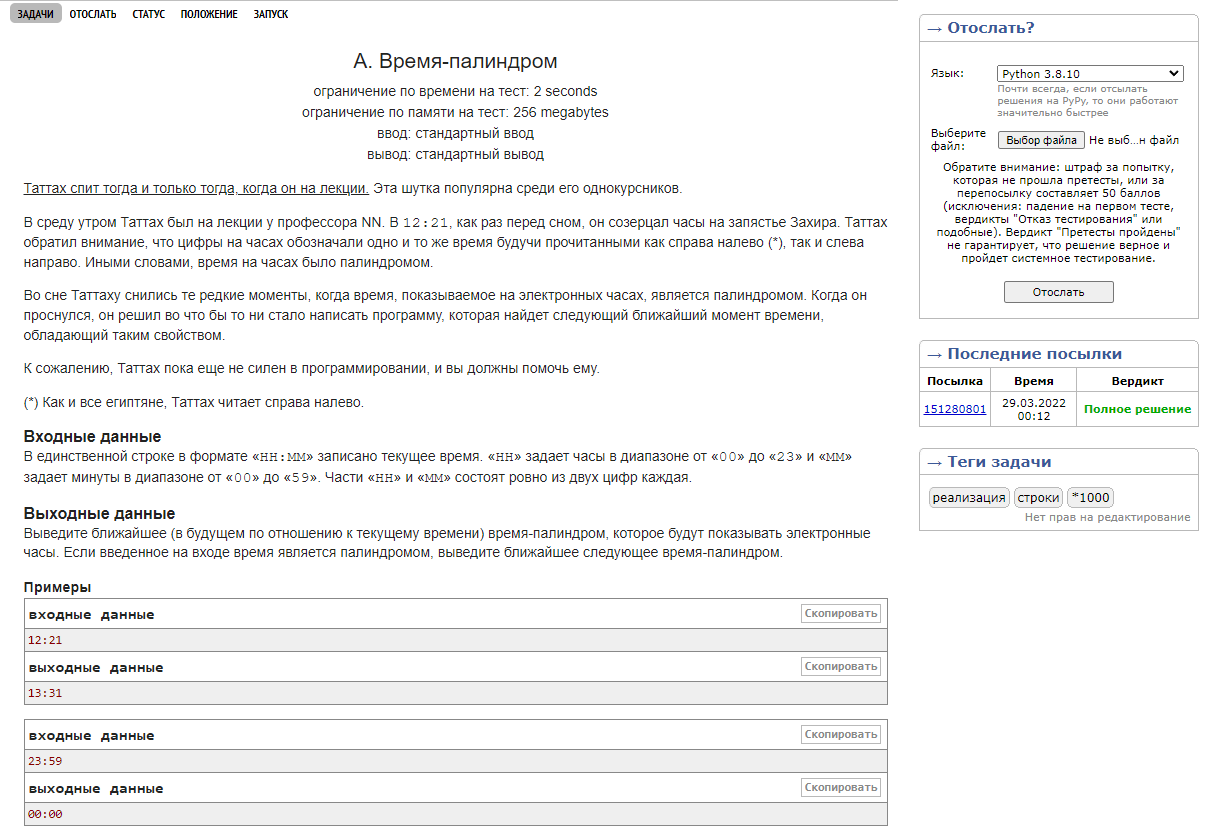


Рисунок 1.2 – Пример задачи и ее теги

Говоря о теоретическом материале, можно начать с термина реализация. Языки программирования в целом являются одним из наиболее быстро развивающихся элементов компьютерных наук, но их развитие происходит неравномерно. К нашему примеру взят Python, являющийся высокоуровневым языком программирования, который очень вырос с времени своего появления, а в настоящее время он используется в самых разных сферах нашей жизни. Решая задачи, можно с уверенностью сказать, что для них в нашем случае в PyCharm используется Virtualenv, то есть виртуальная среда, позволяющая устанавливать библиотеки локально в изолированном каталоге для конкретного проекта. Следующим термином являются строки. Строка – это последовательность символов произвольной длины. Еще в данной работе используются массивы и он них можно сказать следующее: массив в Python – это набор элементов, которые хранятся в непрерывных ячейках памяти, это идея хранения нескольких элементов одного типа вместе, упрощает вычисление позиции каждого элемента, просто добавляя смещение к базовому значению. Это контейнер, который может содержать фиксированное количество элементов, и эти элементы должны быть одного типа.

Также есть циклы while и for и о них можно сказать, что while – это цикл, который используется в Python для неоднократного исполнения определенной инструкции до тех пор, пока заданное условие остается истинным. Этот цикл позволяет программе перебирать блок кода. Цикл же for - есть средство для перебора последовательностей. С его помощью можно совершать обход строк, списков, кортежей и описанных выше итерируемых объектов. Оператор if else в языке Python — это типичная условная конструкция, которую можно встретить и в большинстве других языков программирования. После if следует условие, а за else идет другое условие, которое выполняется в случае того, если первое не выполняется.

Существуют такие функции, как swapcase() и upper(). Метод swapcase() строки преобразует все символы верхнего регистра в нижний регистр и все символы нижнего регистра в символы верхнего регистра данной строки и возвращает ее. Функция swapcase() в Python возвращает строку, в которой все символы верхнего регистра преобразованы в нижний регистр, а символы нижнего регистра преобразованы в верхний регистр. Метод upper() в Python не принимает никаких параметров. Метод возвращает строку в верхнем регистре из данной строки. Он преобразует все символы нижнего регистра в верхний регистр. Если строчные символы отсутствуют, возвращается исходная строка. Следует упомянуть также метод append(), который добавляет элемент в конец списка. Данный метод принимает единственный аргумент – item. Это элемент, который нужно добавить в конец списка. Элементом могут быть числа, строки, словари, другой список и т.д.

* 1. **Задача 82А**

Первой задачей является 82А под названием «Double Cola». Ограничение по времени на тест составляет секунду, а ограничение по памяти на тест 256 мегабайт. Сама задача звучит следующим образом:

Шелдон, Леонард, Пенни, Раджеш и Говард стоят в очереди к автомату по продаже баночек с напитком «Double Cola», других людей в очереди нет. Первый в очереди (Шелдон) покупает баночку, выпивает ее содержимое и раздваивается! Получившиеся два Шелдона встают в конец очереди. Затем следующий в очереди (Леонард) покупает баночку, выпивает и встает в конец очереди в двойном экземпляре, и так далее. Этот процесс продолжается до бесконечности.

Например, третью баночку колы выпьет Пенни, и очередь будет выглядеть так: Раджеш, Говард, Шелдон, Шелдон, Леонард, Леонард, Пенни, Пенни.

Напишите программу, которая выведет имя человека, выпившего n-ую баночку.

Обратите внимание, что в самом начале очередь выглядит так: Шелдон, Леонард, Пенни, Раджеш, Говард. Первым человеком является Шелдон.

Входными данными является:

Входные данные состоят из единственного целого числа n (1 ≤ n ≤ 109).

Гарантируется, что в претестах проверяется правильность написания всех пяти имен, то есть в них встречаются все пять возможных ответов.

Выходными данными являются:

Выведите единственную строку — имя человека, который выпьет n-ую баночку колы. Баночки нумеруются от 1. Обратите внимание, что следует выводить имена в следующем написании: "Sheldon", "Leonard", "Penny", "Rajesh", "Howard" (без кавычек). Именно в этом порядке друзья стоят в очереди изначально.

Решив данную задачу, пользователь получает 1100 баллов, а тегами задачи являются математика и реализация. В данном задании достаточно понять принцип происходящего для того, чтобы написать соответствующую программу.

* 1. **Задача 108А**

Второй задачей является 108А под названием «Время-палиндром». За ее решение пользователь получает 1000 баллов, ограничение по времени на тестирование программы составляет не более 2 секунд, а ограничение по памяти составляет 256 мегабайтов. Требуется стандартный ввод и стандартный вывод. Далее представлена суть задачи, а также ее входные и выходные данные.

Таттах спит тогда и только тогда, когда он на лекции. Эта шутка популярна среди его однокурсников.

В среду утром Таттах был на лекции у профессора NN. В 12:21, как раз перед сном, он созерцал часы на запястье Захира. Таттах обратил внимание, что цифры на часах обозначали одно и то же время будучи прочитанными как справа налево (\*), так и слева направо. Иными словами, время на часах было палиндромом.

Во сне Таттаху снились те редкие моменты, когда время, показываемое на электронных часах, является палиндромом. Когда он проснулся, он решил во что бы то ни стало написать программу, которая найдет следующий ближайший момент времени, обладающий таким свойством.

К сожалению, Таттах пока еще не силен в программировании, и вы должны помочь ему.

(\*) Как и все египтяне, Таттах читает справа налево.

Входными данными является: в единственной строке в формате «HH:MM» записано текущее время. «HH» задает часы в диапазоне от «00» до «23» и «MM» задает минуты в диапазоне от «00» до «59». Части «HH» и «MM» состоят ровно из двух цифр каждая.

Выходными данными является: выведите ближайшее (в будущем по отношению к текущему времени) время-палиндром, которое будут показывать электронные часы. Если введенное на входе время является палиндромом, выведите ближайшее следующее время-палиндром.

Для решения, задачи требуется понимание того, что такое реализация, а также то, что обозначают строки на языке Python. Ее решения представлено в следующей главе, которая содержит в себе решения и последующих задач.

* 1. **Задача 456А**

Третья задача – 456А с названием «Ноутбуки». По ограничениям – 1 секунда на тест и 256 мегабайт памяти на тест. Данная задача дает 1100 баллов за ее решение, а для нее требуется понимать, что такое сортировки и как они работают. Сама задача звучит следующим образом:

Как-то раз Дима и Леша поспорили о цене и качестве ноутбуков. Дима считает, что чем дороже ноутбук, тем он лучше. Леша говорит, что это не так. Леша считает, что существуют два таких ноутбука, что цена первого меньше, чем цена второго, но первый качественнее второго.

Ваша задача — проверить гипотезу Леши. Заданы характеристики n ноутбуков. Выясните, существуют ли два таких ноутбука, что цена первого строго меньше, чем цена второго, но первый строго качественнее второго.

Входные данные:

Первая строка содержит целое число n (1 ≤ n ≤ 105) — количество ноутбуков.

Каждая из следующих n строк содержит два целых числа ai и bi (1 ≤ ai, bi ≤ n): ai — цена i-го ноутбука, bi — число, обозначающее качество i-го ноутбука (чем больше это число, тем качественнее ноутбук).

Все ai различны. Все bi различны.

Выходные данные:

Если Леша прав, то выведите строку «Happy Alex», иначе выведите «Poor Alex» (кавычки выводить не нужно).

* 1. **Задача 43А**

Предпоследней задачей является задача 43А – «Футбол» с ограничениями по времени в 2 секунды и по памяти на тест в 256 мегабайт. Данная задача оценивается в 1000 баллов, а для ее решения необходимо знать, что такое строки в Python. Задача звучит следующим образом:

Однажды Вася решил посмотреть результаты финала чемпионата Берляндии по футболу 1910 года. К сожалению, он не нашел итоговый счет матча, зато нашел подробное текстовое описание хода всей игры. Всего в этом описании оказалось n строк, каждая из которых описывала один гол. Каждый гол обозначался названием забившей его команды. Помогите Васе — узнайте название команды, победившей в финале. Гарантируется, что матч не закончился вничью.

Входные данные:

В первой строке записано целое число n (1 ≤ n ≤ 100) — количество строк в описании. Далее следует n строк — для каждого мяча записано название забившей его команды. Названия — непустые строки из больших латинских букв, длины которых не превосходят 10 символов. Гарантируется, что матч не закончился в ничью, и в описании встречается не более двух различных команд.

Выходные данные:

Выведите название победившей команды. Напоминаем, что в футболе победившей считается команда, забившая больше мячей.

* 1. **Задача 131А**

Последняя задача под названием «cAPS lOCK» - 131А. Время на тест составляет 0,5 секунд, а память на тест не более 256 мегабайт. За решение пользователь получает 1000 баллов, а для решения требуется понимать что такое реализация и строки.

зАЧЕМ НУЖНА КЛАВИША cAPS lOCK?

Caps Lock — клавиша компьютерной клавиатуры, предназначенная для автоматической (постоянной) смены регистра букв со строчных на прописные. Будучи случайно нажатой, она приводит к последствиям вроде первого абзаца в условии этой задачи.

Будем считать, что слово набрано с ошибочно нажатой клавишей Caps Lock, если:

* либо оно полностью состоит из прописных букв;
* либо прописными являются все его буквы, кроме первой.

В таком случае, нужно автоматически поменять регистр всех букв на противоположный. Например, регистр букв слов «hELLO», «HTTP», «z» должен быть изменен. Напишите программу, которая применяет описанное выше правило или оставляет слово без изменения, если оно не применимо.

Входные данные:

В первой строке входных данных записано слово, состоящее из прописных или строчных букв латинского алфавита. Длина слова — от 1 до 100 символов включительно.

Выходные данные:

Выведите результат обработки данного слова.

**ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

Все программы для курсовой работы были разработаны на языке программирования Python в виртуальной среде Virtualenv в таком ПО, как PyCharm Community Edition. Каждая программа открывается через «File – Open» и соответственно выбирается папка с названием программы. Для запуска и проведения тестов пользователю необходимо произвести нажатие кнопки «Run», либо же воспользоваться горячими клавишами «Shift+F10».

Суть каждой задачи, ее входные и выходные данные, а также требуемая теория для выполнения заданий представлена в предыдущей главе. Ниже будут продемонстрированы решения заданий, их код, а также будет объяснен принцип решения.

**2.1 Решение задачи 82А**

Первой программой является задача 82А с названием «Double Cola». Листинг программы представлен ниже, в нем также подписаны комментариями все моменты, происходящие в программе.

Листинг 2.1.1 – Программа «Task82A»

n=int(input())-1 # Пользовательский ввод чисел в строку. Ожидается целочисленное значение  
while n>4: # Цикл while, где n>4  
 n=(n-5)//2 # Тело цикла, в котором происходит вычитание 5 из n и результат целочисленно делится на 2  
print(["Sheldon","Leonard","Penny","Rajesh","Howard"][n]) # Вывод имени в соответствии с n

Первым делом здесь идет ввод целочисленного значения пользователем, но из введенного значения сразу же вычитается единица. Далее идет цикл while с условием n>2, то есть пока введенное пользователем число больше 4, то выполняется тело цикла, а в случае, если число равно или меньше 4, то тело цикла не выполняется и сразу же программа переходит к следующему этапу после цикла. В самом теле цикла написано то, что n принимает новое значение, а именно сперва из предыдущего значения n вычитают 5, а затем с помощью целочисленного деления делят на два. Данное деление возвращает только целую часть результата, а часть после запятой отбрасывается. Например, если n будет равно 10, то первым делом из 10 вычитается 5, что равно 5, а затем 5 делят на 2 и получают 2. В конце программы используется функция print(), которая выводит заданные объекты на стандартное устройство вывода, то есть на экран. В самой функции принимается последнее значение n и в соответствии с ним выбирается имя человека, представленное в виде: ["Sheldon","Leonard","Penny","Rajesh","Howard"]. В данном списке имен нумерация происходит с 0, поэтому если n = 2, то будет выведено имя Penny. Проверим решение тем, что введем данные значение, указанные выше. Результат работы программы представлен на рисунке 2.1.1

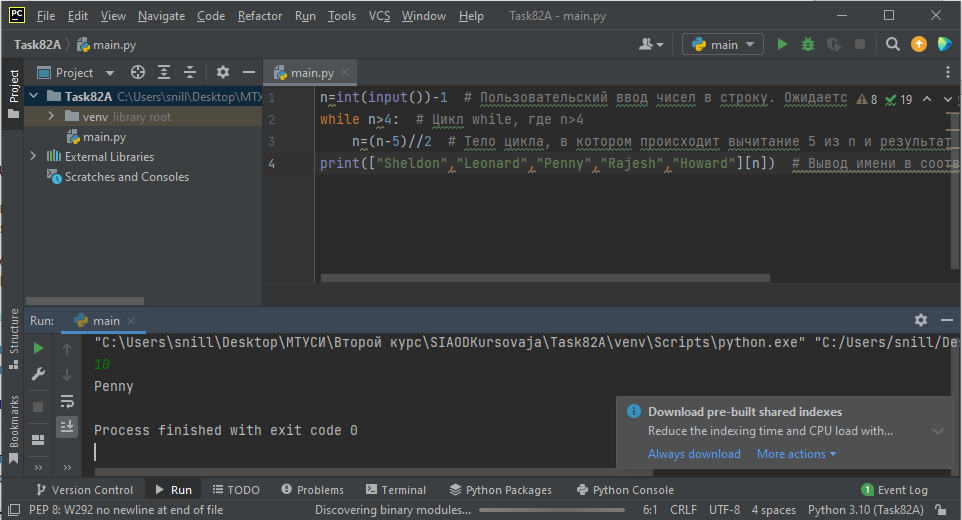


Рисунок 2.1.1 – Результат программы, а также ее код

Таким образом, данная задача решена, и это первые 1100 баллов, которые были получены за решение задач по данной курсовой работе. Ниже на рисунке 2.1.2 представлена отсылка задания, а также ее вердикт, то есть то, что задача имеет полное решение. Время составило 46 мс, что намного быстрее, чем время, выделенное как ограничение по времени на тест.

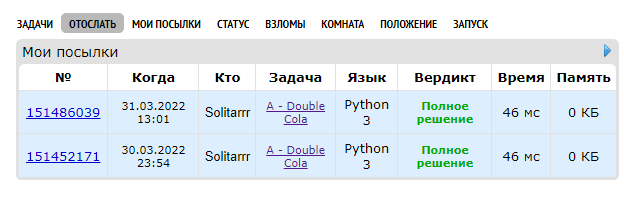


Рисунок 2.1.2 – Отсылка кода и проверка задачи на правильность

Ниже также представлен рисунок 2.1.3, на котором продемонстрировано тестирование программы на сайте, а также показаны несколько тестов и их выходные и выходные данные.

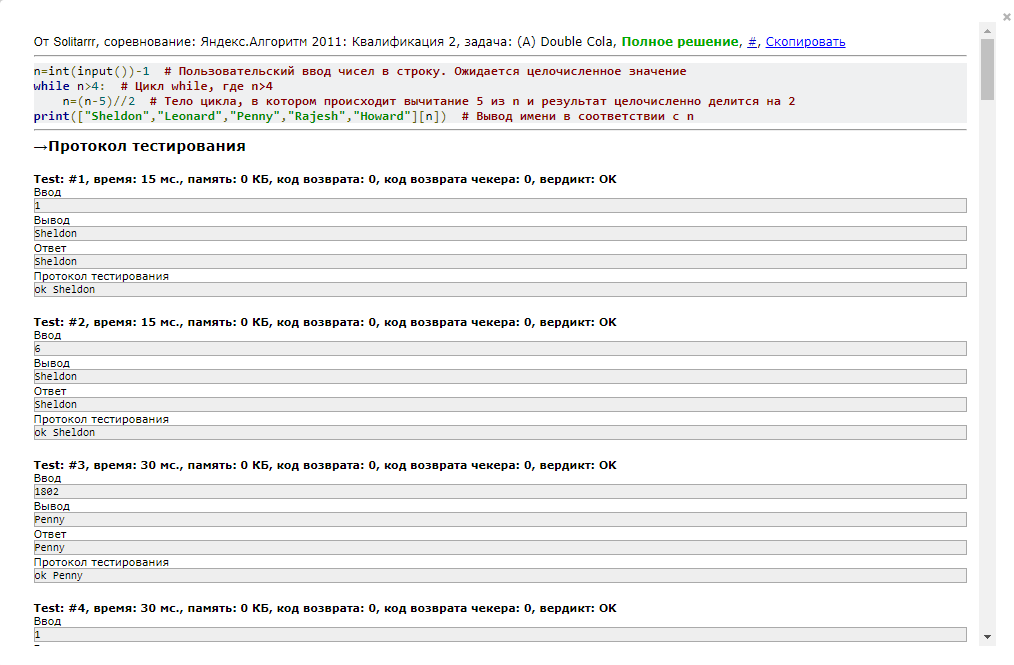


Рисунок 2.1.3 – Тестирования задачи с разными входными и выходными данными

**2.2 Решение задачи 108А**

Второй задачей является 108 под названием «Время-палиндром». Листинг программы представлен ниже, и каждая строчка прокомментирована для лучшего понимания того, что именно происходит в программе. За решение задачи дается 1000 баллов, ограничение по времени составляет 2 секунды, а память ограничена в размере 256 мегабайт.

Листинг 2.2.1 – Программа «Task108A»

a = ["00:00","01:10","02:20","03:30","04:40","05:50","10:01","11:11","12:21","13:31","14:41","15:51","20:02","21:12","22:22","23:32"] # Массив А со значениями времени, которые являютя палиндромами  
b = input() # Пользователь вводит время  
for i in range(len(a)): # Цикл for от i до значения в рамках, равной длине a  
 if a[i] > b: exit(print(a[i])) # Если элемент i в массиве a больше введеного значения, то выйти из цикла с выводом i   
print("00:00") # Если же значение i в массиве a меньше b, то вывести 00:00

Первым этапом программы является внесение в переменную а значений, которые являются палиндромами, то есть тем самым создать массив. Время-палиндром – это то время, которое читается и с одной, и, с другой стороны, одинаково. Далее идет ввод пользователем значения времени, а за этим цикл от элемента i до значения в рамках, равной длине а. В данном цикле имеется условие, что если элемент i в массиве а больше введённого пользователем значения, то выйти из цикла с последующим выводом i. Но в случае если значение i в массиве а меньше b, то вывести время 00:00. На рисунке 2.2.1 представлен ввод пользователем времени и вывод результата программы.

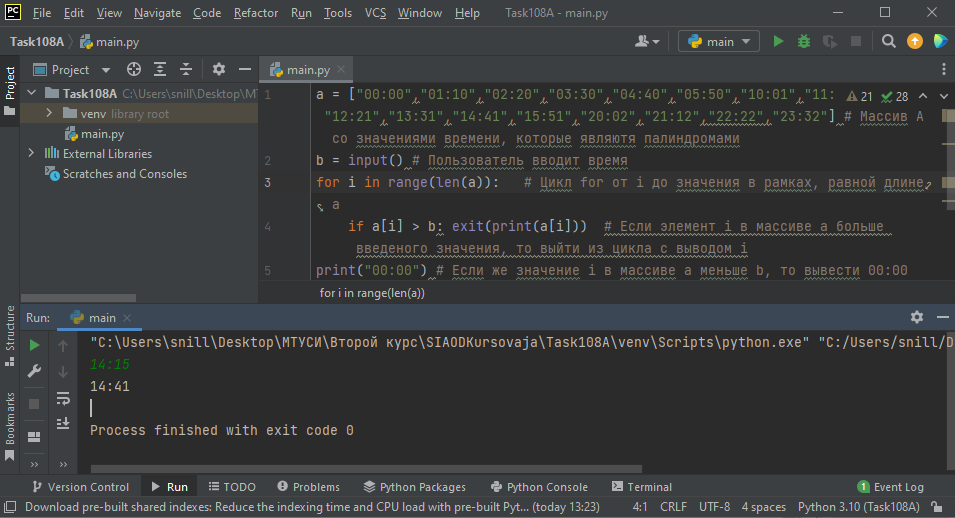


Рисунок 2.2.1 – Результат программы, а также ее код

Данная задача в конечном счете была решена, а за ее решение были получены 1000 баллов. Суммарно за две задачи было получено 2100 баллов. Задача проходит все критерии, а подтверждением этому ниже представлен рисунок с отсылкой задания.



Рисунок 2.2.2 – Отсылка кода и проверка задачи на правильность

На рисунке 2.2.3 представлено тестирование программы на сайте, а также показаны несколько тестов и их выходные и выходные данные.

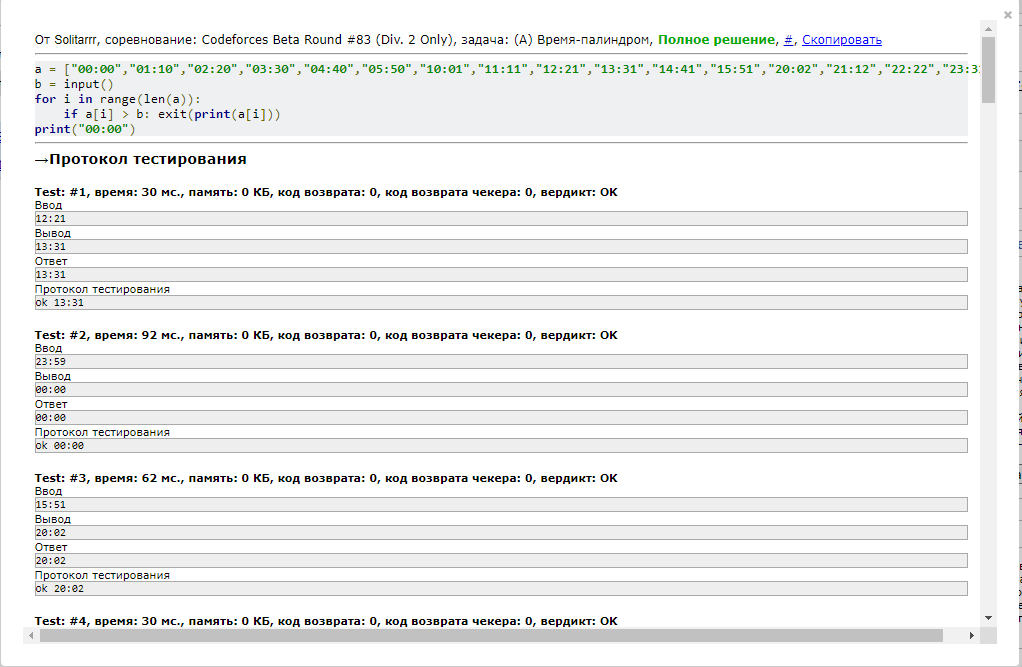


Рисунок 2.2.3 – Тестирования задачи с разными входными и выходными данными

**2.3 Решение задачи 456А**

Третьей задачей является 456А под названием «Ноутбуки». За данную задачу можно получить 1100 баллов, а ее ограничение по времени на тест составляет 1 секунда, а ограничение по памяти составляет 256 мегабайт. Листинг данной задачи представлен ниже, а также каждая строчка кода имеет свои комментарии, которые объясняют, что происходит в той или иной части программы.

Листинг 2.3.1 – Программа «Task456A»

n = int(input()) # Ввод значения в n  
for i in range(n): # Цикл for от элемента i до n  
 a, b = input().split() # Делит начальную строку на подстроки  
 if a != b: # Если а не равно значению b  
 print("Happy Alex") # То вывести Happy Alex  
 exit() # И выйти  
print("Poor Alex") # Либо же вывести Poor Alex

Сначала пользователь вводит значение, обозначающее количество ноутбуков. Далее приведен цикл for от элемента i до n в котором идет деление начальной строки на подстроки, а затем идет условие с помощью if заключающееся в том, что если а не равно значению b, то вывести сообщение «Happy Alex» и происходит выход. Либо же происходит вывод сообщения «Poor Alex». Ниже представлен рисунок 2.3.1, на котором продемонстрирован код программы, а также результат вводы входных данных, представленных на сайте «Codeforces».

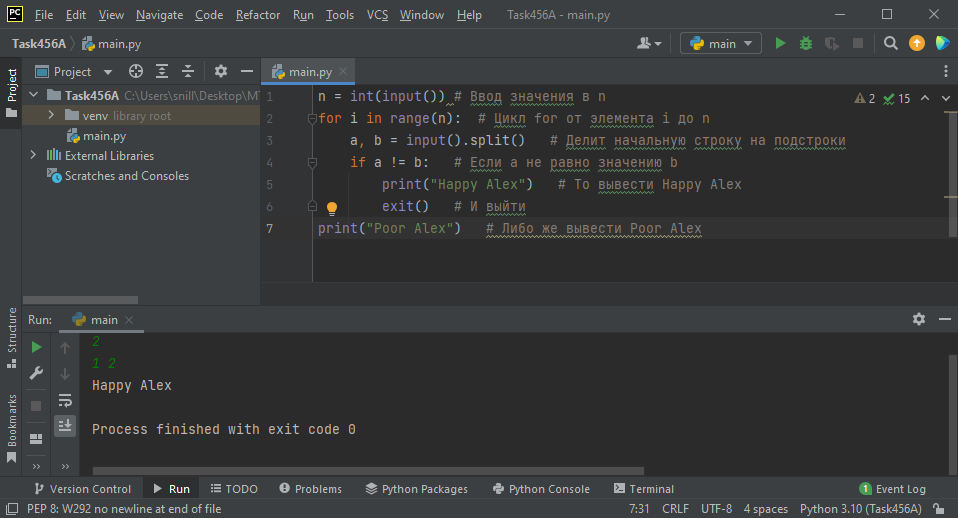


Рисунок 2.3.1 – Результат программы, а также ее код

Решение задачи представлено выше, а за нее были получены 1100 баллов, что в конечном счете дает 3200 баллов за три задачи. Задача проходит все критерии, а подтверждением этому ниже представлен рисунок с отсылкой задания.

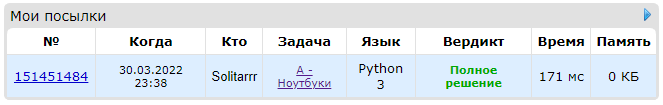


Рисунок 2.3.2 – Отсылка кода и проверка задачи на правильность

На рисунке 2.3.3 продемонстрировано тестирование программы на сайте. Также показаны несколько тестов и их выходные и выходные данные.

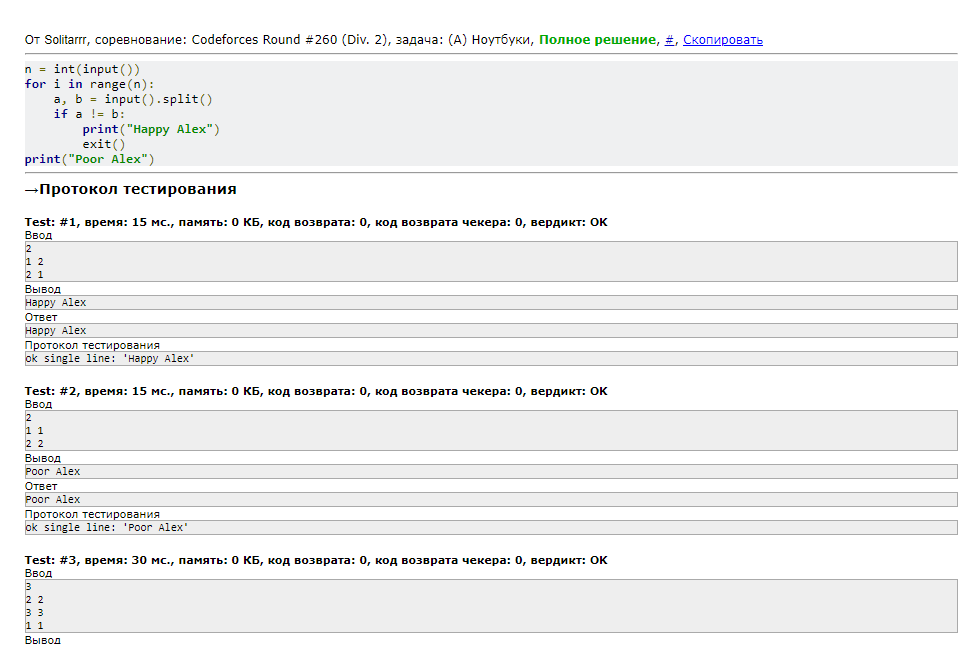


Рисунок 2.3.3 – Тестирования задачи с разными входными и выходными данными

**2.4 Решение задачи 43А**

Четвертой задачей является 43А под названием «Футбол» за которую можно получить 1000 баллов, а ее ограничение по времени на тест составляет 2 секунды, а ограничение по памяти составляет 256 мегабайт. Листинг задачи представлен ниже. Каждая строчка кода содержит комментарии, объясняющие что происходит в той или иной части программы.

Листинг 2.4.1 – Программа «Task43A»

n = int(input()) # Ввод пользователем данных, а именно кол-во строк  
arr = [] # Массив  
for i in range(n): # Цикл for от i до n  
 val = input() # Ввод назвиний команд  
 arr.append(val) # Append добавляет элемент в конец списка  
a = arr[0]  
a\_count = 0  
b\_count = 0  
for i in range(n): # Цикл for от i до n  
 if arr[i] == a: # Если элемент i равен а  
 a\_count += 1  
 else:  
 b = arr[i]  
 b\_count += 1  
if b\_count > a\_count: # В случае если b\_count больше a\_count  
 print(b) # Вывести b  
else:  
 print(a) # Или же вывести а

Ввод пользователем данных является первым этапом данной программы. Далее создается пустой массив для того, чтобы в дальнейшем заполнить его. Следующим является цикл for от i до n, а в нем происходит ввод названий команд и далее добавляются элементы в конец списка. Следующий цикл в себе добавляет единицу, либо же добавляет к другой команде единицу. Далее, в случае если b\_count больше a\_count, вывести либо b, либо а. Ниже представлен рисунок 2.4.1, на котором продемонстрирован код программы, а также результат вводы входных данных, представленных на сайте «Codeforces».

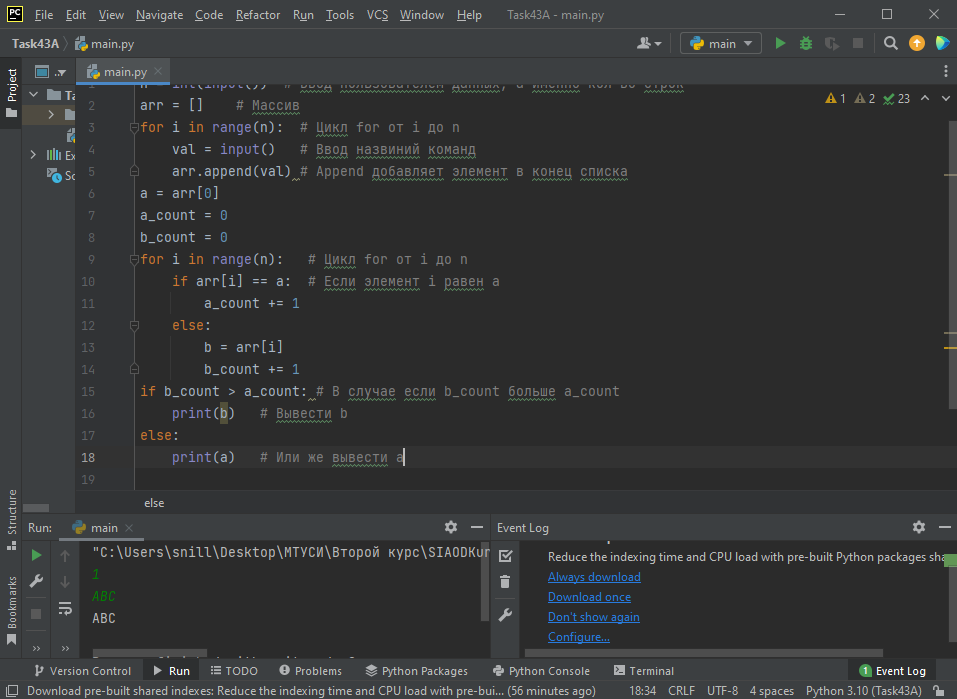


Рисунок 2.4.1 – Результат программы, а также ее код

Решение задачи указано на рисунке 2.4.1 и в листинге 2.4.1. Данная задача дает 1000 баллов, что в конечном счете дает 4200 баллов за четыре задачи. Задача проходит все критерии, а подтверждением этому ниже представлен рисунок с отсылкой задания.

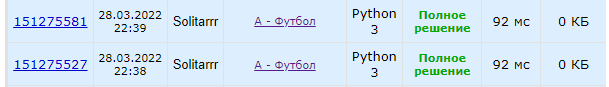


Рисунок 2.4.2 – Отсылка кода и проверка задачи на правильность

На рисунке 2.4.3 продемонстрировано тестирование программы на сайте. Также показаны несколько тестов и их выходные и выходные данные.

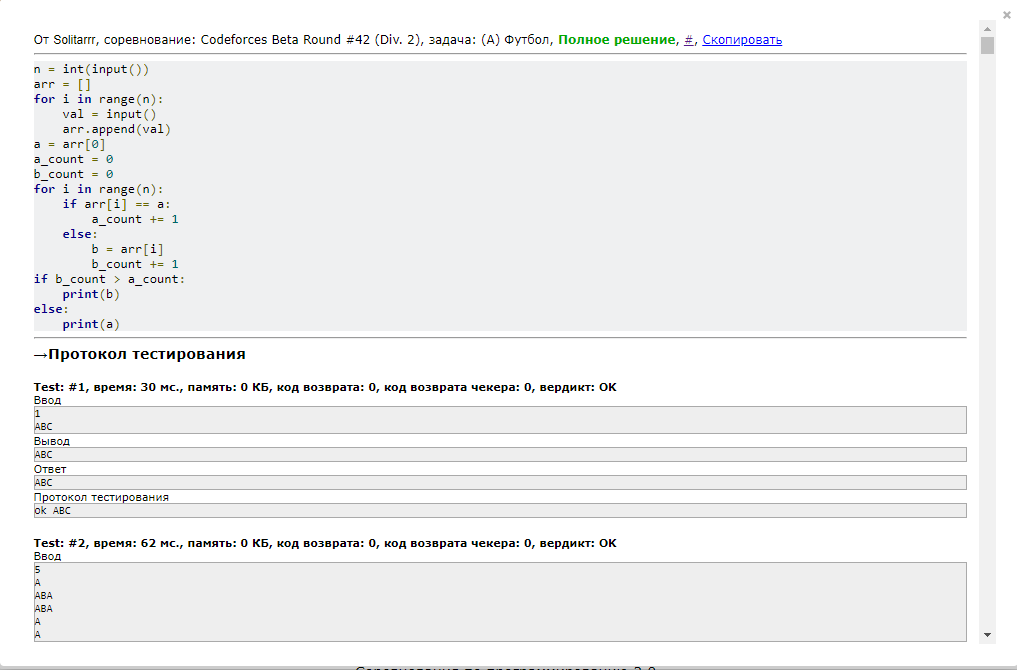


Рисунок 2.4.3 – Тестирования задачи с разными входными и выходными данными

**2.5 Решение задачи 131А**

Последней задачей является 131А под названием «cAPS lOCK» за которую можно получить 1000 баллов, а ее ограничение по времени на тест составляет 0.5 секунды, а ограничение по памяти составляет 256 мегабайт. Листинг задачи представлен ниже.

Листинг 2.5.1 – Программа «Task131A»

n= input()  
print(n.swapcase() if (n[1:]==n[1:].upper()) else n)

Сначала пользователь вводит слово, которое принимается в переменную n. А затем идет вывод результата. Используется метод swapcase() преобразует все символы верхнего регистра в нижний регистр и все символы нижнего регистра в символы верхнего регистра данной строки и возвращает ее. И также используется метод upper(), который возвращает строку в верхнем регистре из данной строки, то есть преобразует все символы нижнего регистра в верхний регистр. Таким образом если пользователь вводит слово, начинающееся с буквой нижнего регистра, а затем остальная часть с верхним регистром, то благодаря имеющимся функциям слово поменяется таким образом, что первая буква станет большой, а оставшиеся – маленькими. Результат программы представлен на рисунке 2.5.1.

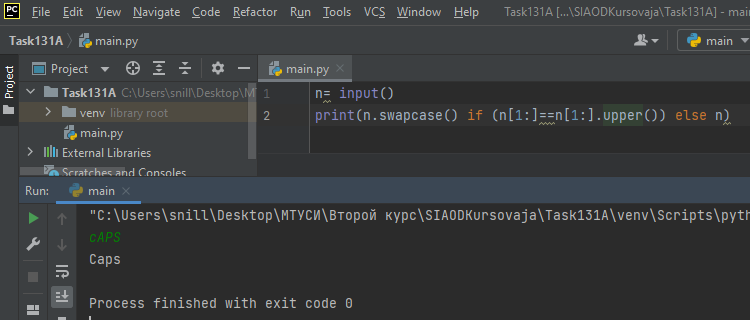


Рисунок 2.5.1 – Результат программы, а также ее код

Решение задачи указано на рисунке 2.5.1 и в листинге 2.5.1. Данная задача дает 1000 баллов, что в конечном счете дает 5200 баллов за пять задач. Задача проходит все критерии, а подтверждением этому ниже представлен рисунок с отсылкой задания.



Рисунок 2.5.2 – Отсылка кода и проверка задачи на правильность

На рисунке 2.5.3 продемонстрировано тестирование программы на сайте. Также аналогичным образом показаны несколько тестов и их выходные и выходные данные.

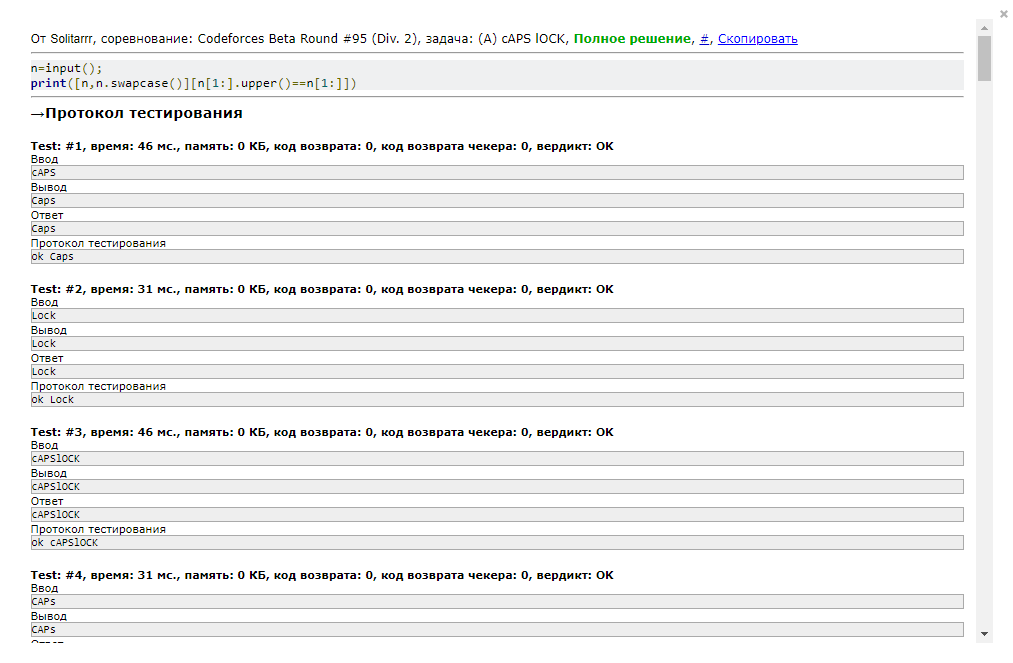


Рисунок 2.5.3 – Тестирования задачи с разными входными и выходными данными

Таким образом, все задачи были решены и суммарно было набрано 5200 баллов, что и требовалось изначально. Каждая задача имеет свое решение, свои критерии, которые были соблюдены, а также каждая задача была подробно разобрана и каждый шаг программ был прокомментирован.

**ЗАКЛЮЧЕНИЕ**

В рамках работы над курсовым проектом были разработаны 5 программ для решения задач, которые были ранее определены. Каждая задача была подобрана в соответствии с критериями и суммарный балл 5 задач составил 5200 баллов.

В результате выполненной работы были реализованы следующие задачи:

1. Выбрать задания, за которые дают 1000 баллов и более;
2. Собрать необходимый теоретический материал для их решения;
3. Реализовать все программы по каждому заданию.

Таким образом, поставленные в курсовой работе цели и задачи, были успешно выполнены.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) – URL: [http://docs.cntd.ru/document/1200157208](https://vk.com/away.php?to=http%3A%2F%2Fdocs.cntd.ru%2Fdocument%2F1200157208&cc_key=) (дата обращения: 29.10.2020). – Текст: электронный.  
2. ГОСТ 19.201-78. Межгосударственный стандарт. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению (введен в действие Постановлением Госстандарта СССР от 18.12.1978 N 3351) (ред. от 01.06.1981) – URL: [http://docs.cntd.ru/document/gost-19-201-78](https://vk.com/away.php?to=http%3A%2F%2Fdocs.cntd.ru%2Fdocument%2Fgost-19-201-78&cc_key=) (дата обращения: 29.10.2020). – Текст: электронный.

3. Codeforces: [Электронный ресурс]. URL: <https://codeforces.com/>. (Дата обращения: 25.03.2022

**ПРИЛОЖЕНИЕ А**

В приложении А указан код программы.

В листинге №1 представлен код программы 82А.

Листинг №1:

n=int(input())-1 # Пользовательский ввод чисел в строку. Ожидается целочисленное значение  
while n>4: # Цикл while, где n>4  
 n=(n-5)//2 # Тело цикла, в котором происходит вычитание 5 из n и результат целочисленно делится на 2  
print(["Sheldon","Leonard","Penny","Rajesh","Howard"][n]) # Вывод имени в соответствии с n

В листинге №2 представлен код программы 108А.

Листинг №2:

a = ["00:00","01:10","02:20","03:30","04:40","05:50","10:01","11:11","12:21","13:31","14:41","15:51","20:02","21:12","22:22","23:32"] # Массив А со значениями времени, которые являютя палиндромами  
b = input() # Пользователь вводит время  
for i in range(len(a)): # Цикл for от i до значения в рамках, равной длине a  
 if a[i] > b: exit(print(a[i])) # Если элемент i в массиве a больше введеного значения, то выйти из цикла с выводом i  
print("00:00") # Если же значение i в массиве a меньше b, то вывести 00:00

В листинге №3 представлен код программы 456А.

Листинг №3:

n = int(input()) # Ввод значения в n  
for i in range(n): # Цикл for от элемента i до n  
 a, b = input().split() # Делит начальную строку на подстроки  
 if a != b: # Если а не равно значению b  
 print("Happy Alex") # То вывести Happy Alex  
 exit() # И выйти  
print("Poor Alex") # Либо же вывести Poor Alex

В листинге №4 представлен код программы 43А.

Листинг №4:

n = int(input()) # Ввод пользователем данных, а именно кол-во строк  
arr = [] # Массив  
for i in range(n): # Цикл for от i до n  
 val = input() # Ввод назвиний команд  
 arr.append(val) # Append добавляет элемент в конец списка  
a = arr[0]  
a\_count = 0  
b\_count = 0  
for i in range(n): # Цикл for от i до n  
 if arr[i] == a: # Если элемент i равен а  
 a\_count += 1  
 else:  
 b = arr[i]  
 b\_count += 1  
if b\_count > a\_count: # В случае если b\_count больше a\_count  
 print(b) # Вывести b  
else:  
 print(a) # Или же вывести а

В листинге №5 представлен код программы 131А.

Листинг №5:

n= input()  
print(n.swapcase() if (n[1:]==n[1:].upper()) else n)