# ПРАКТИЧЕСКАЯ РАБОТА №1

## «Условные конструкции и операторы выбора»

Цель работы: Научиться работать с условными конструкциями и операторами выбора.

### Программа № 1

Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.

Сначала пользователь вводит число и в зависимости от того, больше оно нуля или нет, мы выведем его таким же или увеличенным на единицу.

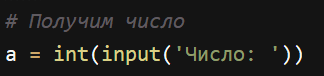


Рисунок – Ввод

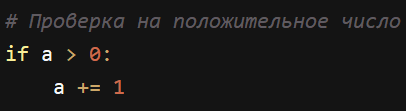


Рисунок – Блок проверки переменной

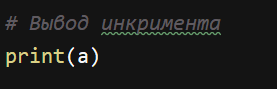


Рисунок – Вывод на консоль

Результат работы:



Рисунок – Условие выполнилось



Рисунок – Условие не выполнилось

### Программа № 2

Дано целое число. Если оно является положительным, то прибавить к

нему 1; если отрицательным, то вычесть из него 2; если нулевым, то

заменить его на 10. Вывести полученное число.

Пользователь вводит число. Если число положительное, то мы увеличиваем введённое число на 1, если оно меньше нуля, то выведется число, из которого вычли 2, а в ином же случае – заменим значение переменной на 10.

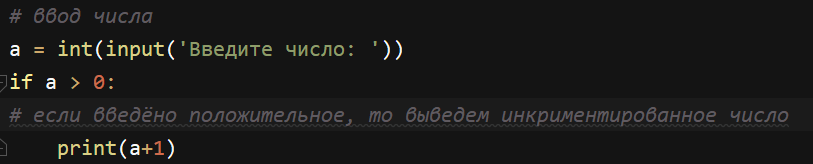


Рисунок – Ввод и проверка числа на положительное

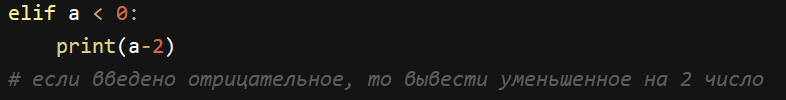


Рисунок – Условие для отрицательного числа

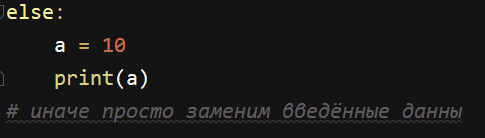


Рисунок – Блок для всех остальных случаев

Результат работы:



Рисунок – Условие 1



Рисунок – Условие 2

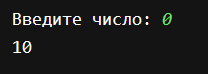


Рисунок – Все остальные случаи

### Программа № 3

Даны три переменные вещественного типа: A, B, C. Если их значения упорядочены по возрастанию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных A, B, C.

Реализуем стандартный ввод. Если значения переменных упорядочены по возрастанию, то выведем их умноженными на 2, иначе вывести их инвертированными.

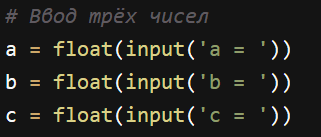


Рисунок – Пользовательский ввод

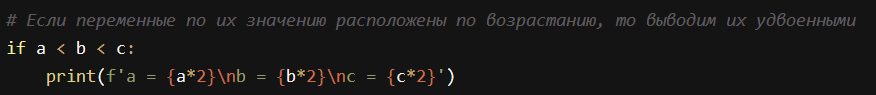


Рисунок – Вывод при условии задачи

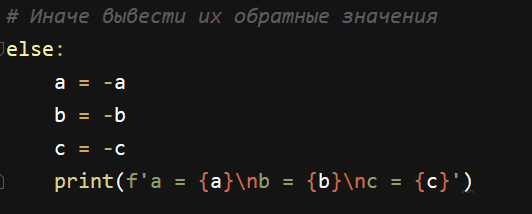


Рисунок – Блок иначе

Результат работы:

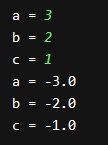
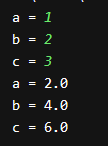


Рисунок – Результат работы в обоих условиях

### Программа № 4

Разработайте программу, запрашивающую у пользователя букву латинского алфавита. Если введенная буква входит в следующий список (a, e, i, o или u), необходимо вывести сообщение о том, что эта буква гласная. Если была введена буква y, программа должна написать, что эта буква может быть как гласной, так и согласной. Во всех других случаях должно выводиться сообщение о том, что введена согласная буква.

Пользователь вводит число. Если это гласные латинского алфавита, то вывести сообщение «это гласная», для y «эта буква может быть как гласной, так и согласной», а для всех остальных «это согласная».

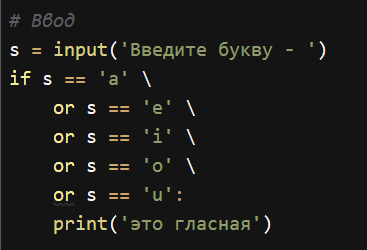


Рисунок – Ввод буквы и сравнение их с заданными

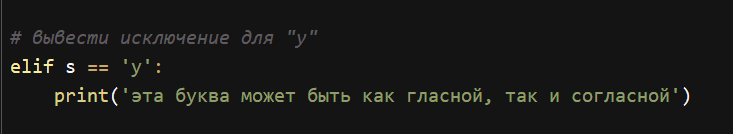


Рисунок – Исключение для y



Рисунок – Для согласных

Результат работы:

Рисунок – a Рисунок – o Рисунок – e



Рисунок – u Рисунок – i



Рисунок – y



Рисунок – f(или любая другая буква помимо вышеперечисленных)

### Программа № 5

Разработайте программу, запрашивающую у пользователя букву латинского алфавита. Если введенная буква входит в следующий список (a, e, i, o или u), необходимо вывести сообщение о том, что эта буква гласная. Если была введена буква y, программа должна написать, что эта буква может быть как гласной, так и согласной. Во всех других случаях должно выводиться сообщение о том, что введена согласная буква.

Пользователь вводит вариант единиц измерения, затем число этой массы. В итоге мы в зависимости выбранной единицы массы выводим нужное число килограмм(конвертируем).

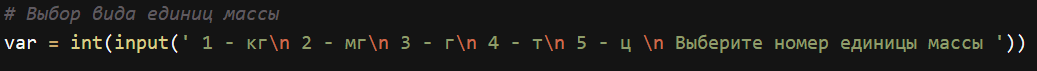


Рисунок – Вариант ЕМ

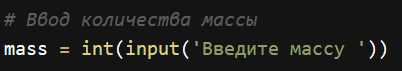


Рисунок – Масса

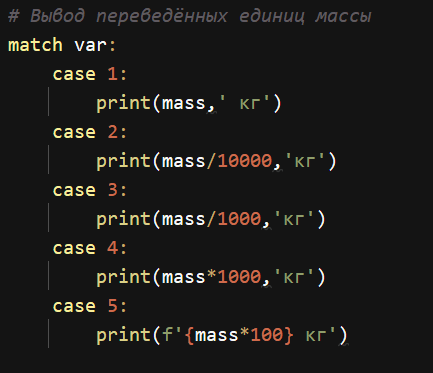


Рисунок - Вывод введённой массы в кг

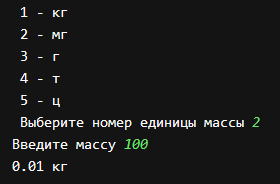
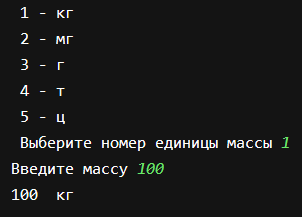
Результат работы:

Рисунок – мг в кг

Рисунок – кг в кг

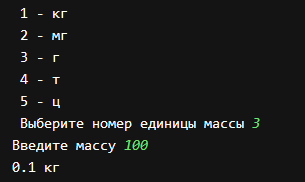


Рисунок – г в кг

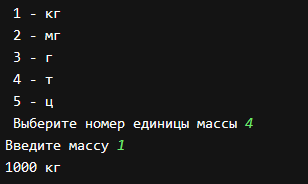


Рисунок – т в кг

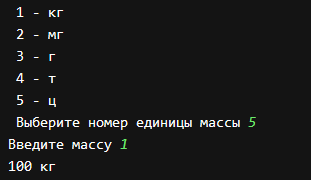


Рисунок – ц в кг

### Программа № 6

Необходимо реализовать консольный калькулятор со всеми основными

арифметическими операциями. Пользователь вводит 2 числа с клавиатуры и выбирает операцию из списка. Если пользователь выбрал не существующую операцию необходимо вывести соответствующее сообщение.

Сначала запросим у пользователя два числа для проведения действия над ними, затем с помощью функции знак соответствующей арифметической операции. Присваиваем работу функции char переменной ch(это позволит нам запрашивать знак снова), а потом сравниваем введённый знак со списком возможных операций. В противном случае, если пользователь ввёл не +-\*/, а какой-то другой символ, то запросит его снова и покажет корректный ответ, иначе ошибка.

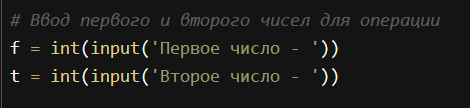


Рисунок – Ввод чисел

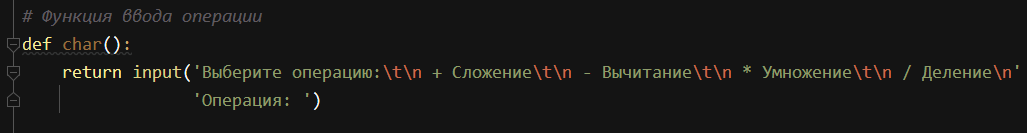


Рисунок – Функция запроса операции

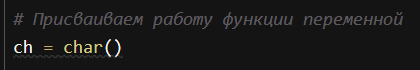


Рисунок – Присвоение работы функции переменной

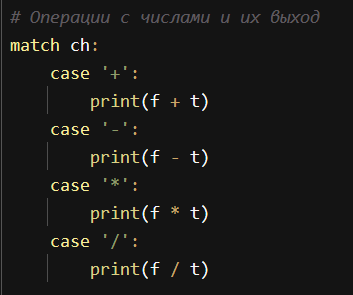


Рисунок – Список операций

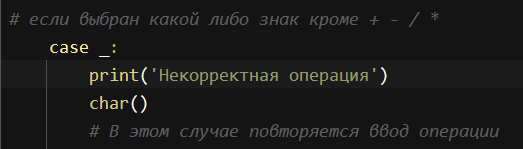


Рисунок – Дефолт для остальных знаков и повторный запрос знака

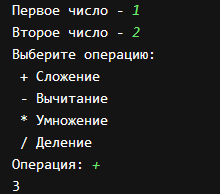
Результат работы:

Рисунок – Сложение

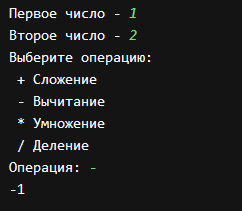


Рисунок – Вычитание

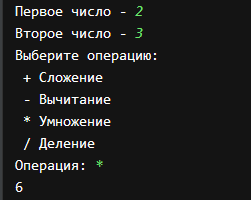


Рисунок – Умножение

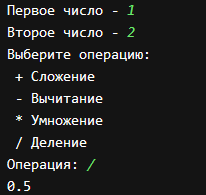


Рисунок – Деление

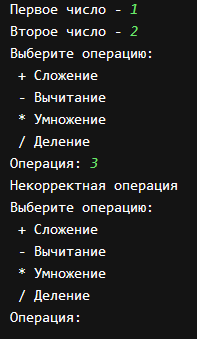


Рисунок – Ошибочный ввод операции

### Программа № 7

Дано целое число в диапазоне 1–7. Вывести строку — название дня недели, соответствующее данному числу (1 — «понедельник», 2 — «вторник» т. д.).

Реализуем ввод номера дня недели, а затем выведем соответствующий день недели.

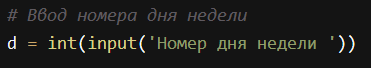


Рисунок – Ввод

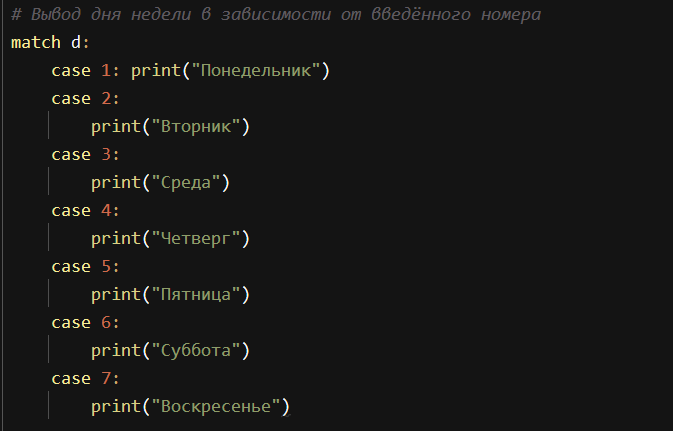


Рисунок – Вывод дня

Результат работы:



Рисунок – Понедельник



Рисунок - Вторник



Рисунок – Среда



Рисунок – Четверг



Рисунок – Пятница



Рисунок – Суббота



Рисунок – Воскресенье

Вывод: Научились работать с условными конструкциями, их выборным вариантом и операторами ввода/вывода.

# ПРАКТИЧЕСКАЯ РАБОТА №2

## «Циклы»

Цель работы: Научиться работать с циклами.

### Программа № 1

Необходимо написать программу для подсчета  
среднего значения всех введенных пользователем чисел. Индикатором  
окончания ввода будет служить ноль. При этом программа должна выдавать соответствующее сообщение об ошибке, если первым же введенным  
пользователем значением будет ноль.

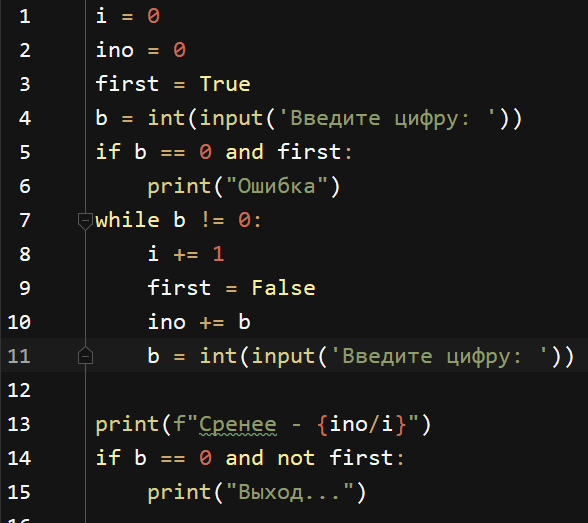


Рисунок - Решение

Создаём переменные счётчик и мешочек, переменную первого запуска. В случае того, если программа получит первый раз ноль приводе выведено сообщение об ошибке. В другом же случае будет повторяться цикл.

Результат работы:

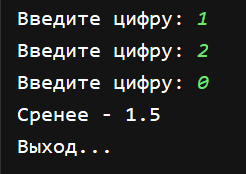


Рисунок - Вывод

### Программа № 2

В зоопарке цена входного билета зависит от возраста посетителя. Дети до двух лет допускаются бесплатно. Дети в возрасте от трех до 12 лет могут  
посещать зоопарк за $14,00. Пенсионерам старше 65 лет вход обойдется  
в $18,00, а обычный взрослый билет стоит $23,00.  
Напишите программу, которая будет запрашивать возраст всех посетителей в группе (по одному за раз) и выводить общую цену билетов для посещения зоопарка этой группой. Общую цену билетов  
стоит выводить в формате с двумя знаками после запятой.

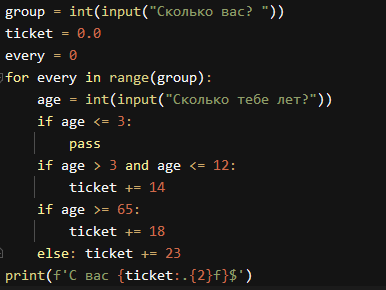


Рисунок – Решение

Вот количество человек в группе для итерации цикла for. Для каждого человека в группе мы прокручиваем цикл. Сколько тебе лет-то переменную, которую записываем и проверяем в дальнейшем при определённых условиях.

Результат работы:

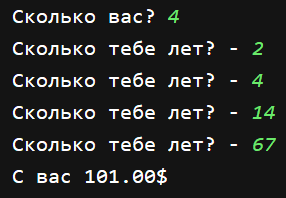


Рисунок - Вывод

### Программа № 3

Строка называется палиндромом, если она пишется одинаково в обоих  
направлениях. Например, палиндромами в английском языке являются  
слова «anna», «civic», «level», «hannah». Напишите программу, запрашивающую у пользователя строку и при помощи цикла определяющую, является ли она палиндромом.

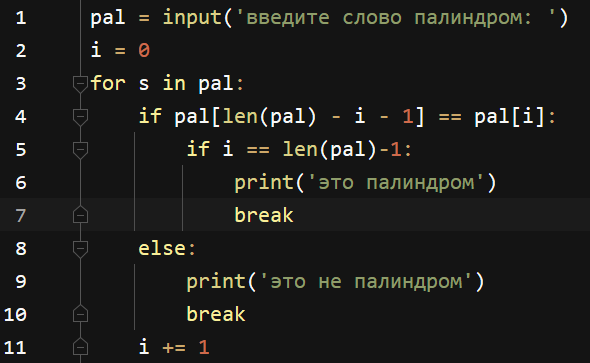


Рисунок – Решение

Сначала вводим возможный палиндром, затем создаём и который будет служить индексом нашего слова. Зачем в цикле прокручивает слово с двух сторон, не учитывая букву посередине. В случае, если 1 часть и 2 равны то мы вводим соответствующие сообщения.

Результат работы:



Рисунок - Вывод

### Программа № 4

В магазине была объявлена скидка размером 60 % на ряд товаров, и для  
того чтобы покупатели лучше ориентировались, владелец торговой точки  
решил вывесить отдельную таблицу со скидками с указанием уцененных  
товаров и их оригинальных цен. Используйте цикл для создания подобной  
таблицы, в которой будут исходные цены, суммы скидок и новые цены для  
покупок на сумму $4,95, $9,95, $14,95, $19,95 и $24,95. Убедитесь в том,  
что суммы скидки и новые цены отображаются с двумя знаками после  
запятой.

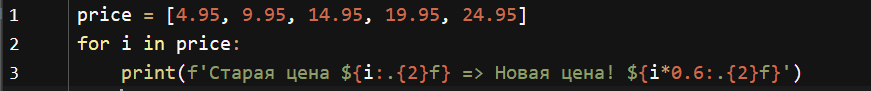


Рисунок – Решение

Создаём список с оценками, затем в цикле прокручиваем эту эти цены. Сначала обычную цену, потом со скидкой. В итоге мы водим стоимость билета с двумя знаками после запятой.

Результат работы:

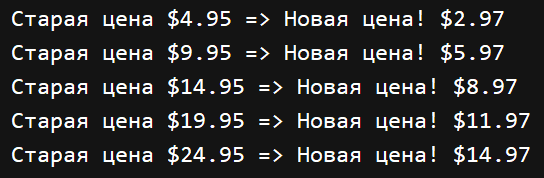


Рисунок - Вывод

### Программа № 5

Необходимо от пользователя принять строку. Если элемент строки является четным, то перевести символ в нижний регистр. Если элемент строки является не четным перевести символ в верхний регистр

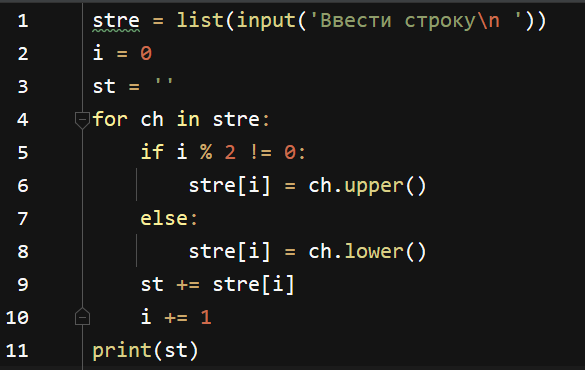


Рисунок 61 – Решение

Вводим строчку создаём переменную итератора и строчки, затем запускаем цикл. И если элемент чётный и нечётный, то сделаем его в верхнем регистре или в Нижнем регистре, затем всё складываем в одну строчку все элементы и вводим.

Результат работы:

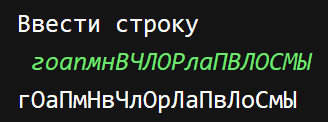


Рисунок - Вывод

Вывод: Научились работать с циклами.

# ПРАКТИЧЕСКАЯ РАБОТА №3

## «Списки, словари, кортежи, множества»

Цель работы: Научиться работать со словарями, списками, кортежами и множествами.

### Программа № 1

Дан массив, состоящий из n кортежей. найдите максимальное число повторений элементов в кортежах.

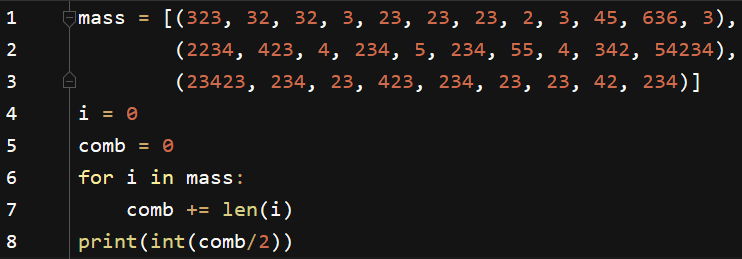


Рисунок – Решение

Если мы хотим найти максимальное количество повторений, а повторение топора, значит, он нужно прокрутить наш массив в цикле и найти половину от него так мы узнаем округленную округлённое количество повторений в нашем массиве.

Результат работы:

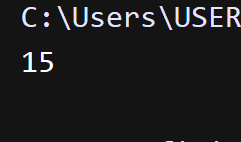


Рисунок - Вывод

### Программа № 2

Дан словарь данных шаблона "фамилия: {дата рождения, email}", а также дана дата проверки. вывести все фамилии пользователей, которые родились позже этой даты.

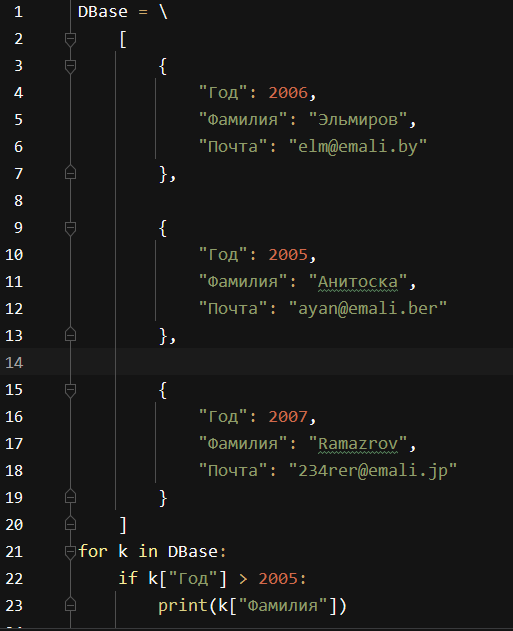


Рисунок – Решение

Создаём список из нескольких словарей в словаре пишем Вот фамилию и почту пользователя затем прокручиваем данные список со словарями и если год больше 2005 выведем фамилию всех остальных кто младше 2005.

Результат работы:

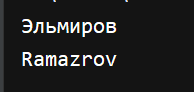


Рисунок - Вывод

### Программа № 3

Дана строка, состоящая из некоторого количества слов. вывести слова из этой строки без их повторений.

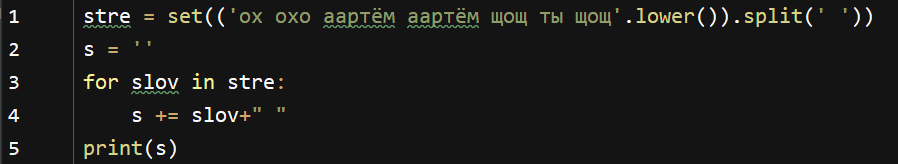


Рисунок - Решение

Результат работы:



Рисунок - Вывод

### Программа № 4

Напишите программу, которая будет запрашивать у пользователя целочисленные значения и сохранять их в виде списка. Индикатором окончания ввода значений должен служить ноль. Затем программа должна вы- вести на экран все введенные пользователем числа (кроме нуля) в порядке возрастания – по одному значению в строке. Используйте для сортировки  
либо метод sort, либо функцию sorted.

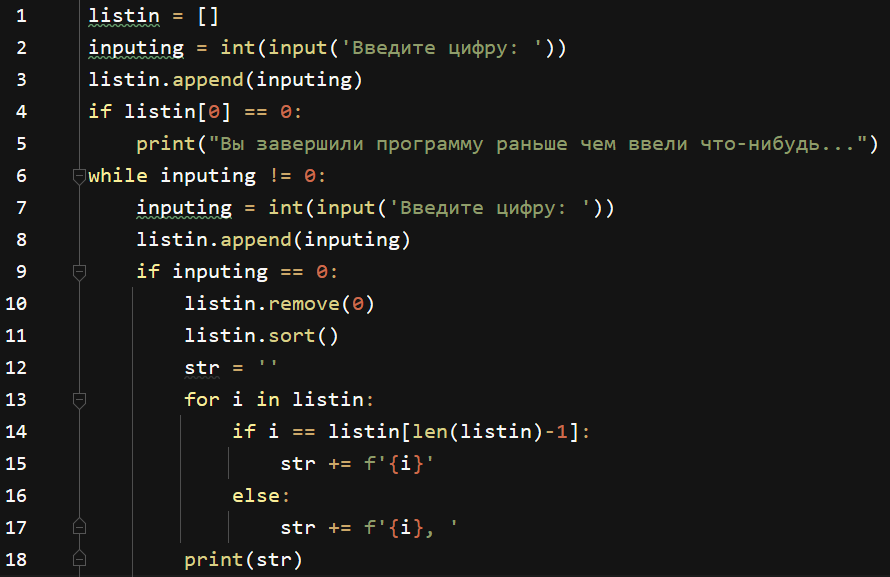


Рисунок – Решение

Создаю список ввожу цифру добавляют цифру в массив, если первая цифра этого массива будет 0, то выйдет сообщение о преждевременное завершение программы. Иначе мы вводим цифры, которые мы сразу же, добавляя в наш список и Если наша переменная равна нулю, то мы убираем, текущий 0 сортирует массив И добавляем строчку для вывода, которую в цикле мы заполняем и потом же выводим.

Результат работы:

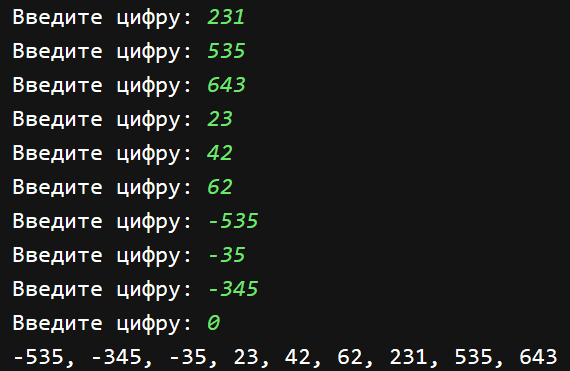


Рисунок – Вывод

### Программа № 5

Напишите программу, которая, как и в предыдущем случае, будет запрашивать у пользователя целые числа и сохранять их в виде списка. Индикатором окончания ввода значений также должен служить ноль. На этот раз необходимо вывести на экран введенные значения в порядке убывания.

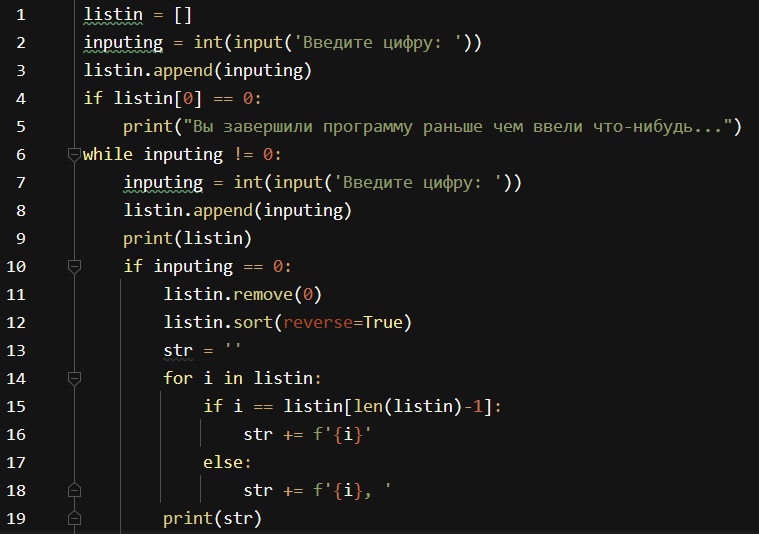


Рисунок – Решение

Делаем всё тоже самое, только пишем в 12 строчки скобочках от sort reverse = True Это означает, что мы выведем все значения в обратном порядке, то есть от самого большого до самого маленького.

Результат работы:

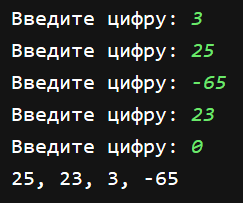


Рисунок - Вывод

### Программа № 6

Напишите программу, запрашивающую у пользователя целые числа, пока он не оставит строку ввода пустой. После окончания ввода на экран должны быть выведены сначала все отрицательные числа, которые были введены, затем нулевые и только после этого положительные. Внутри каждой группы числа должны отображаться в той последовательности, в которой были введены пользователем. Например, если он ввел следующие числа: 3, -4, 1, 0, -1, 0 и -2, вывод должен оказаться таким: -4, -1, -2, 0, 0, 3 и 1. Каждое значение должно отображаться на новой строке



Рисунок – Решение

Создаю список, внутри которого будут три пустых списка, потом я ввожу какое-либо значение в переменную, заполняя в другую переменную пустотой, если переменной вода в равно пустота, то она на вид сообщение о предварительном выполнение программы, иначе у нас преобразуется в число сравнивается с нулём равно это не больше 0 или меньше нуля и распределяет это всё по определённым осину внутри другого массива. Если у нас не равно пустоте, то мы вводим цифру, если у нас же Ведено цифра равна правила, то есть пустоте, то мы вводим всё в одну строчку и затем же вывозим без дополнительного знака запятой.

Результат работы:

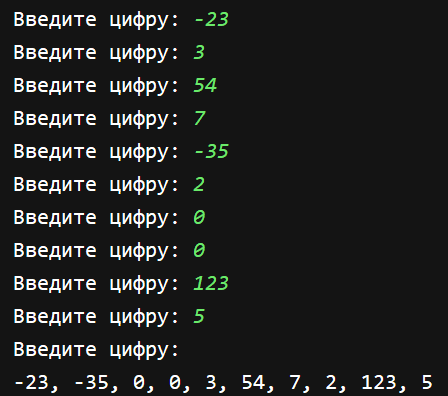


Рисунок - Вывод

### Программа № 7

Напишите функцию с названием reverseLookup, которая будет осуществлять поиск всех ключей в словаре по заданному значению. Функция должна принимать в качестве параметров словарь и значение для поиска и возвращать список ключей̆ (он может быть пустым) из этого словаря, соответствующих переданному значению. В основной программе продемонстрируйте работу функции путем создания словаря и поиска в нем всех ключей по заданному значению. Убедитесь, что функция работает корректно при наличии нескольких ключей для искомого значения, одного ключа и их отсутствии.

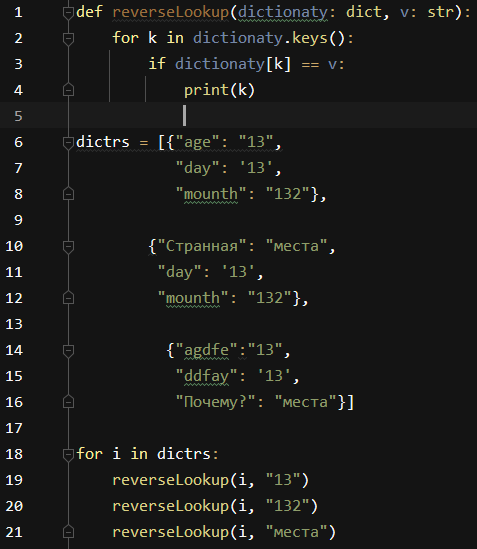


Рисунок – Решение

Создаём функцию, которая проверяет все текущие ключи и по значению В приведённом у мною внутри обращение к функции в Форе Затем прописываю все эти словари в списке и обращение к ним.

Результат работы:

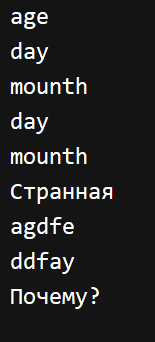


Рисунок - Вывод

### Программа № 8

Напишите программу, определяющую и выводящую на экран количество уникальных символов во введенной пользователем строке. Например, в строке Hello, World! содержится десять уникальных символов, а в строке zzz – один. Используйте словарь или набор для решения этой задачи.

Создаём множество, которое уберёт лишние знаки и выведем его в строку

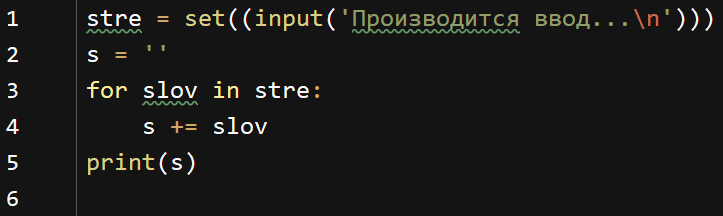


Рисунок - Решение

Результат работы:

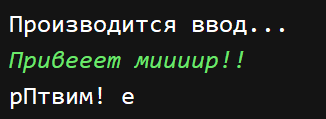


Рисунок - Вывод

Вывод: Научились работать с условными конструкциями, их выборным вариантом и операторами ввода/вывода.

# ПРАКТИЧЕСКАЯ РАБОТА №4

## «Функции»

### Программа № 1

Дан массив, состоящий из n кортежей. найдите максимальное число повторений элементов в кортежах.

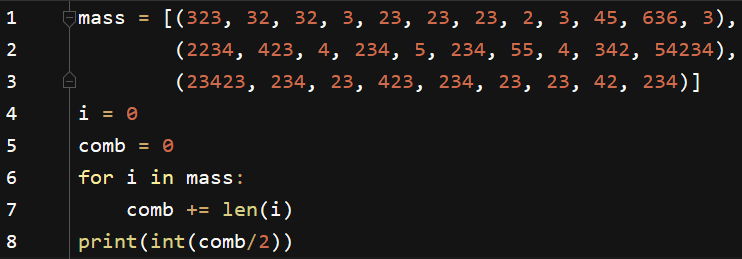


Рисунок – Решение

Если мы хотим найти максимальное количество повторений, а повторение топора, значит, он нужно прокрутить наш массив в цикле и найти половину от него так мы узнаем округленную округлённое количество повторений в нашем массиве.

Результат работы:

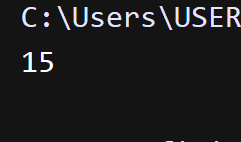


Рисунок - Вывод

# ПРАКТИЧЕСКАЯ РАБОТА №5

## «Обработка ошибок и исключений»

Цель работы: ознакомится со всеми видами исключения в Python и попробовать их на практике. Составить отчет по каждому виду исключения.

Ход работы:

ZeroDivisionError – ошибка с операцией на ноль… Не бывает бесконечных данных, как и оперативки.

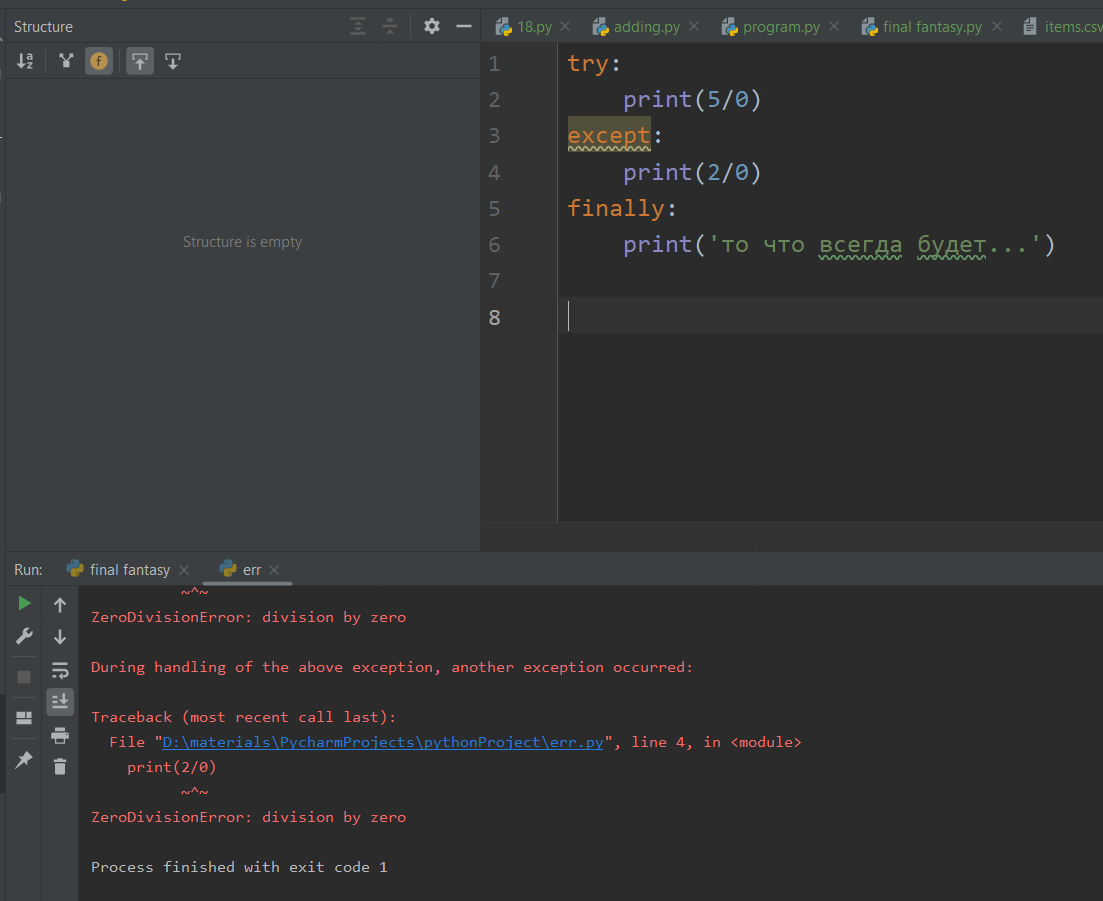


Рисунок 81 – ZeroDivisionError

ValueError – неправильно введено значение.

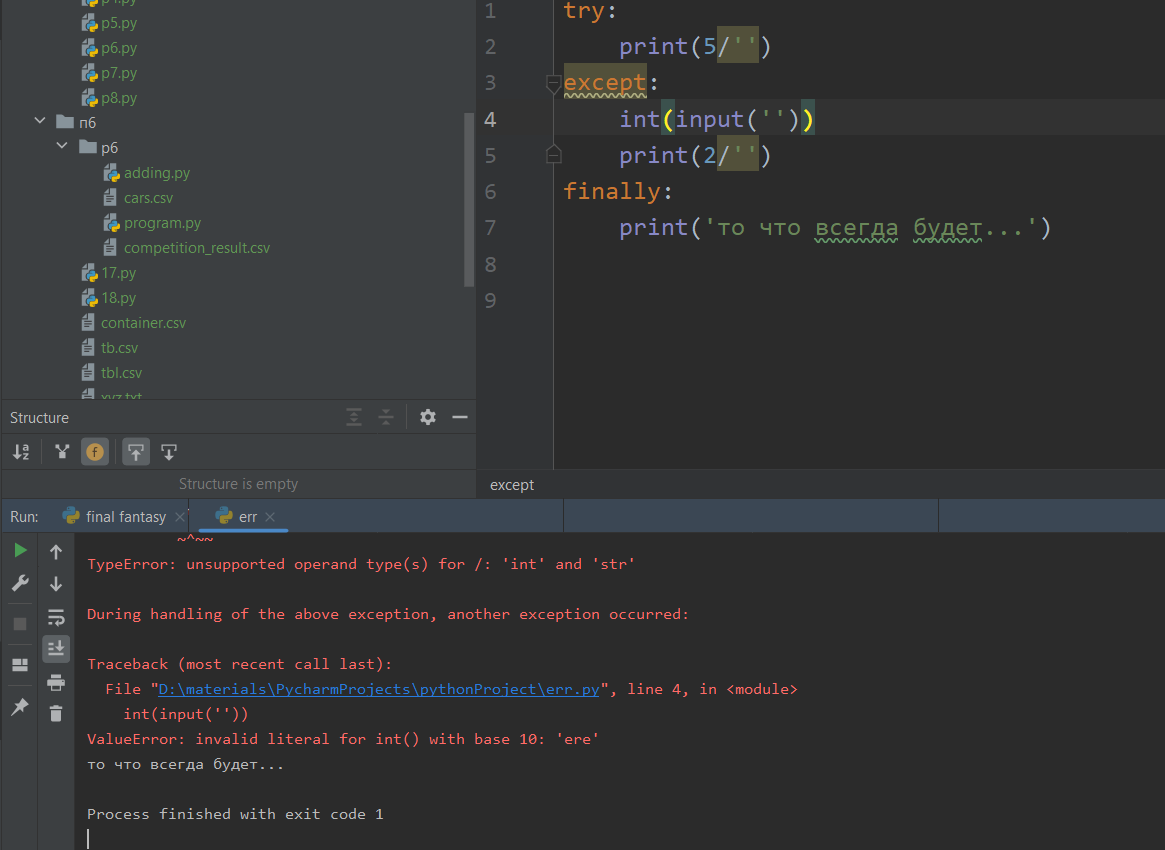


Рисунок 82 – ValueError

TypeError – Неправильный тип переменной.

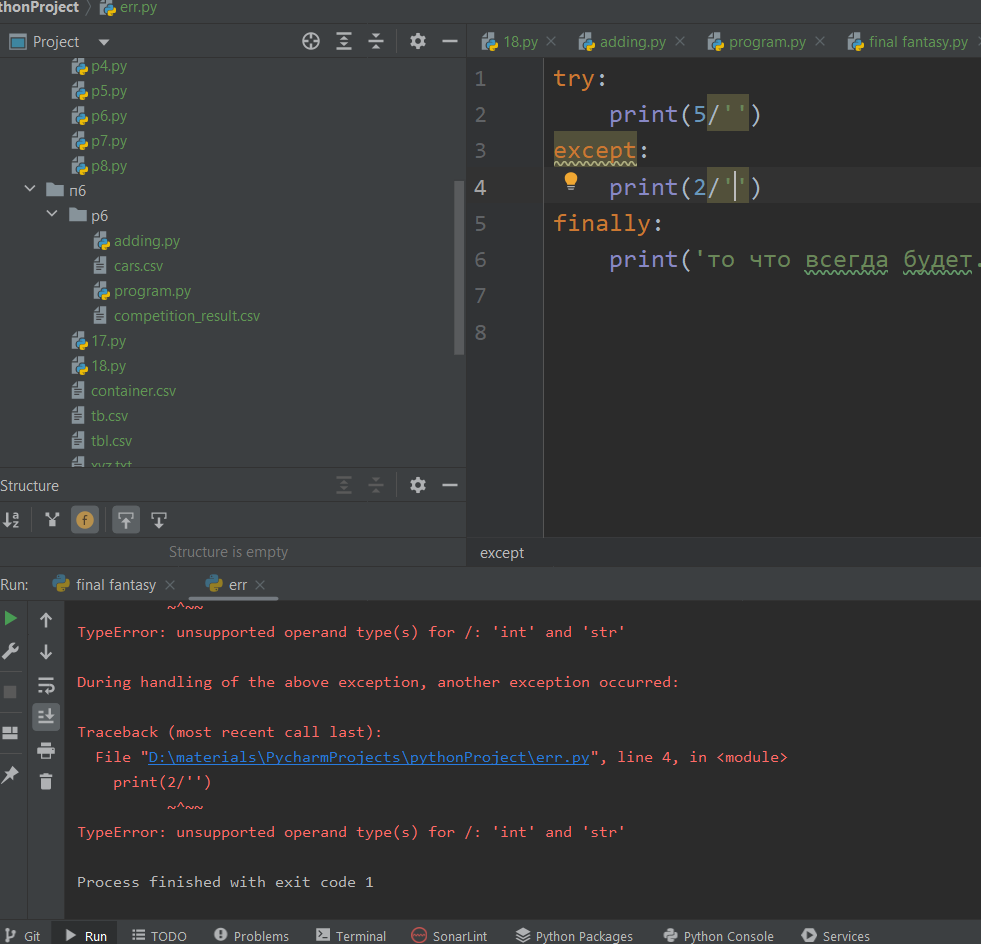


Рисунок 83 – TypeError

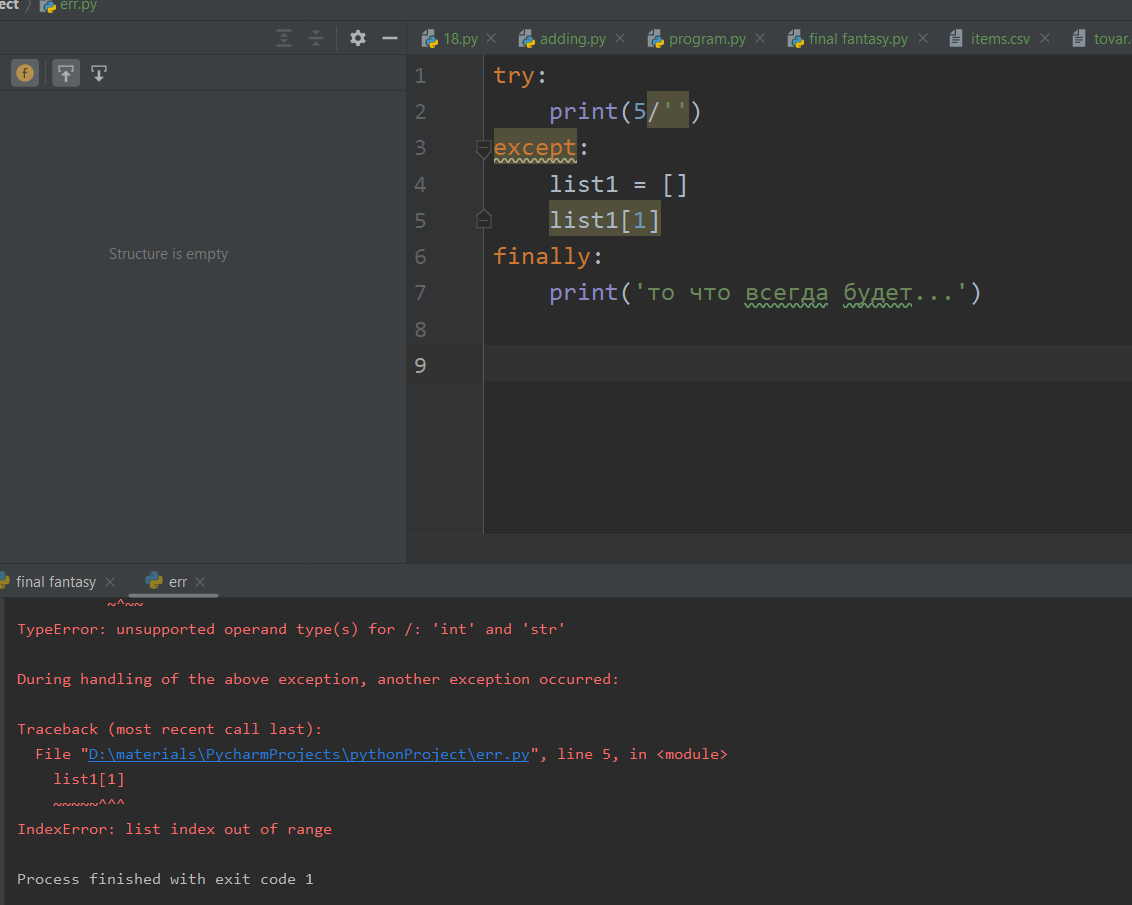
IndexError – Ошибка определения индекса массива, не туда попал, **слишком много просишь!**

Рисунок 84 – IndexError

KeyError – Ошибка подбора ключа, когда мы указываем ключ в словаре, которого нет.

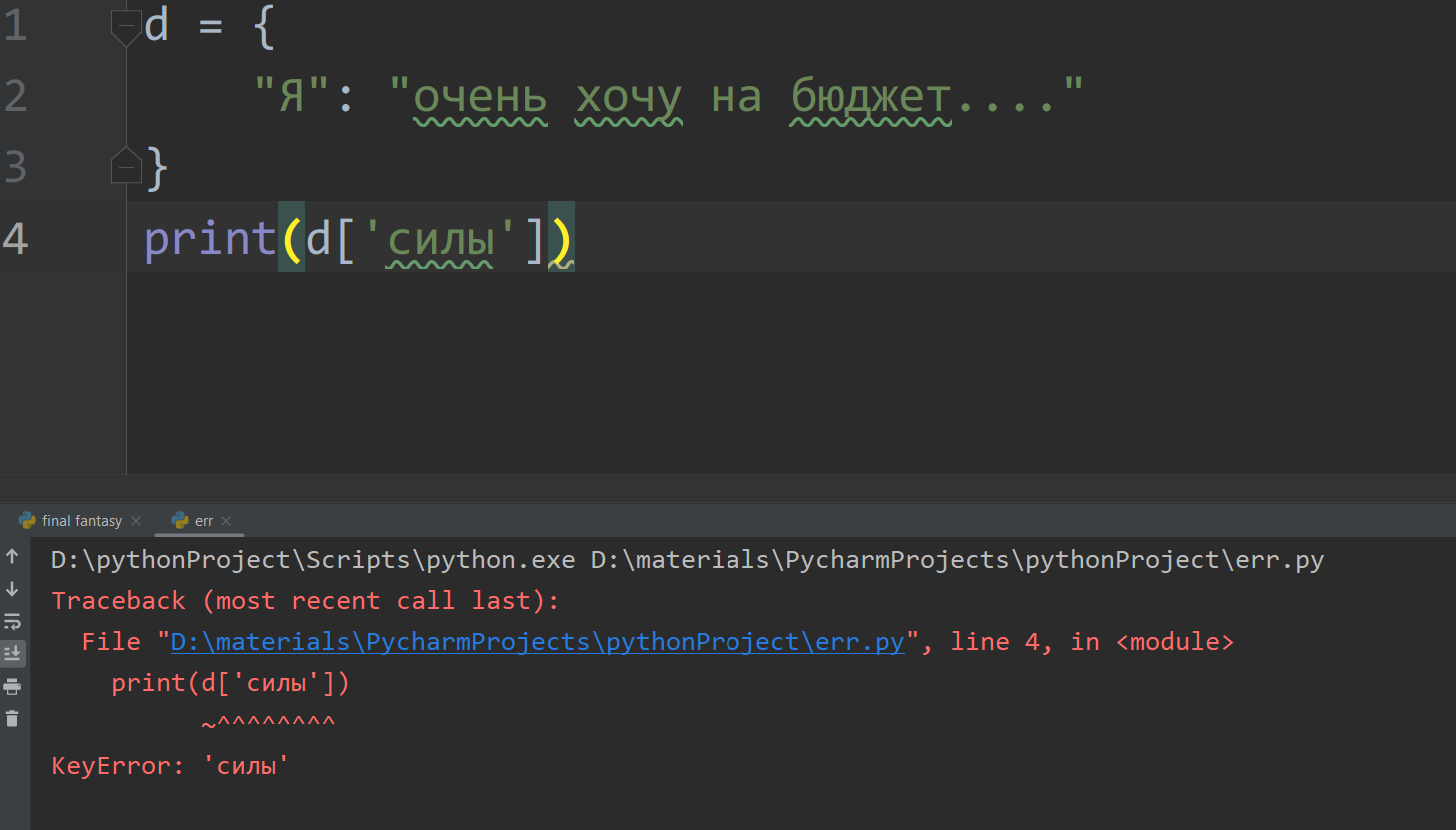


Рисунок 85 – KeyError

NameError – ошибка выведения необъявленной переменной, как вызвать то, чего нет???

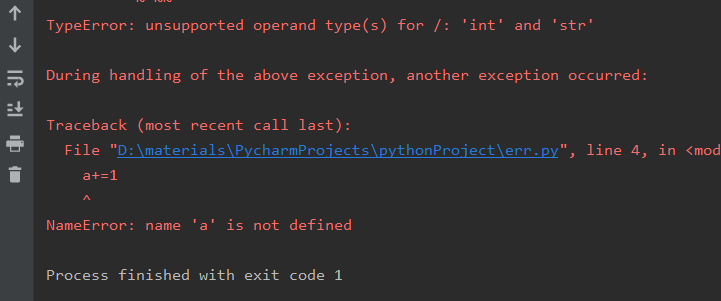


Рисунок 86 – NameError

SyntaxError – синтаксическая ошибка, написать неправильно функцию или пробел не в том месте…

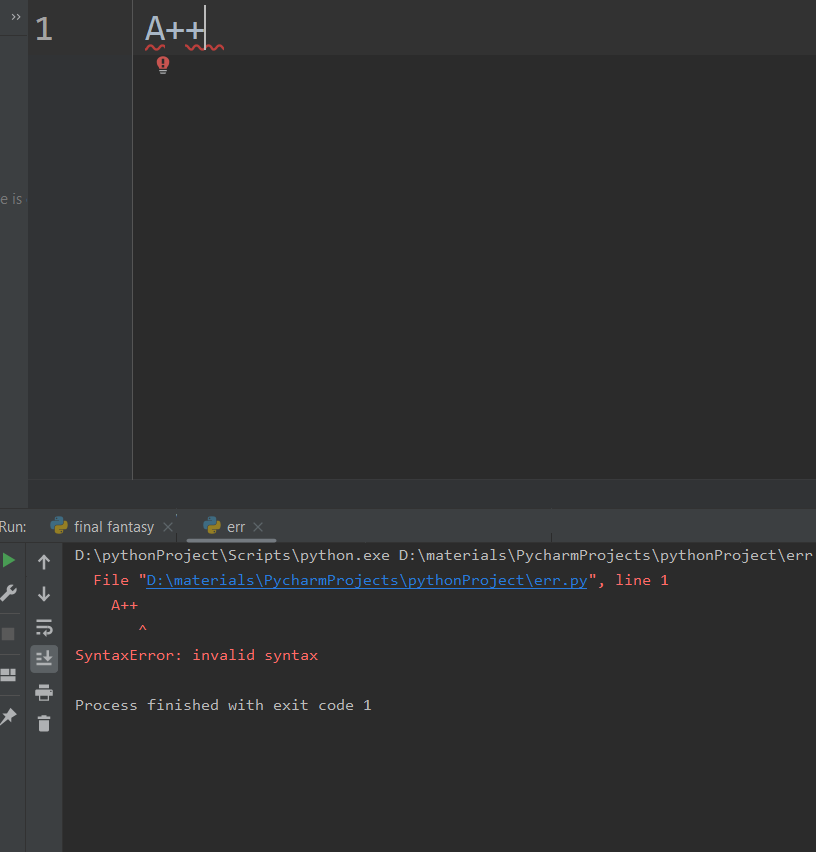


Рисунок 87 – SyntaxError

ModuleNotFoundError – ошибка модуля или же когда ты не разобрался как правильно импортировать модули.

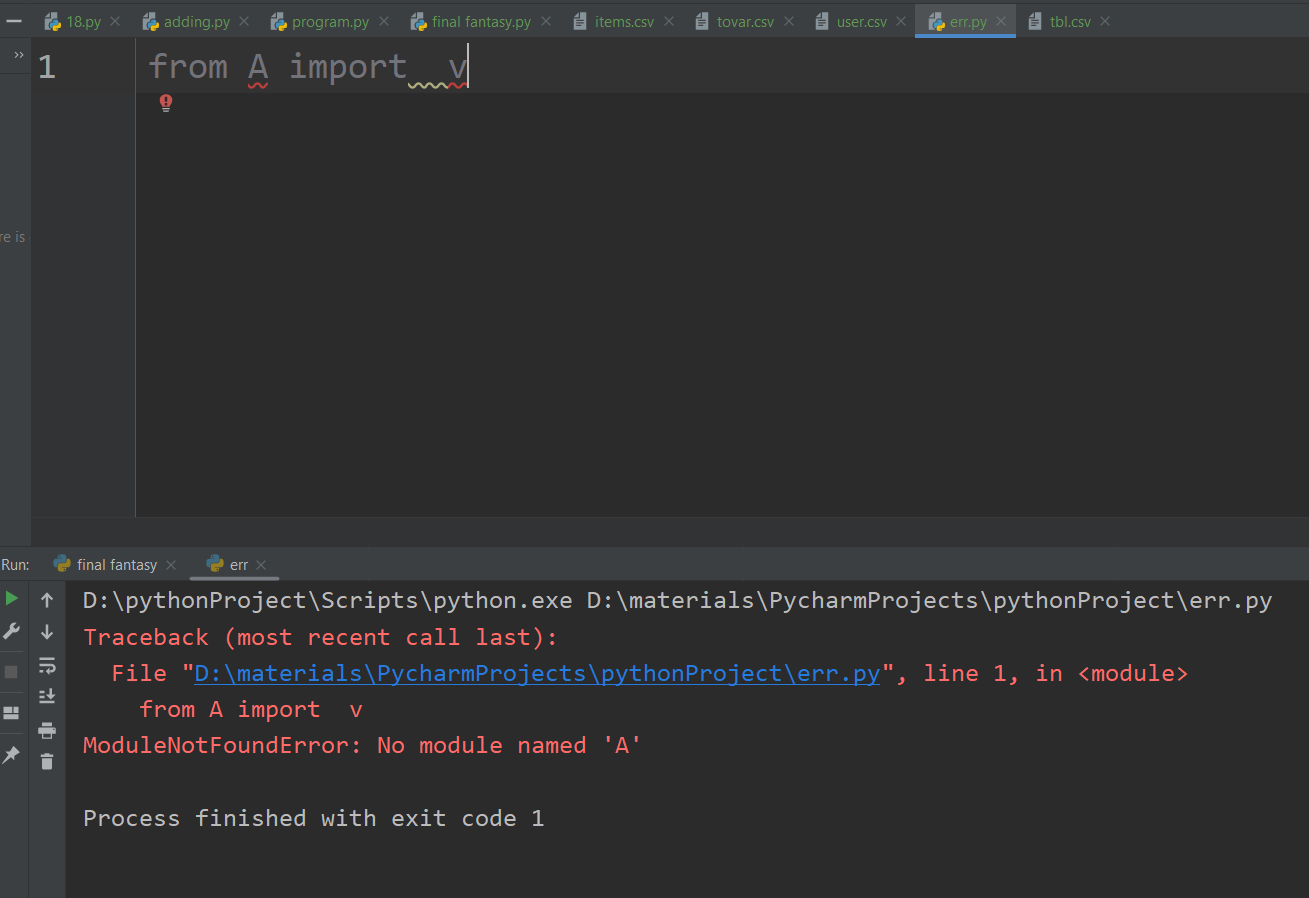


Рисунок 88 – ModuleNorFoundError

AttributeError: ошибка введения команды

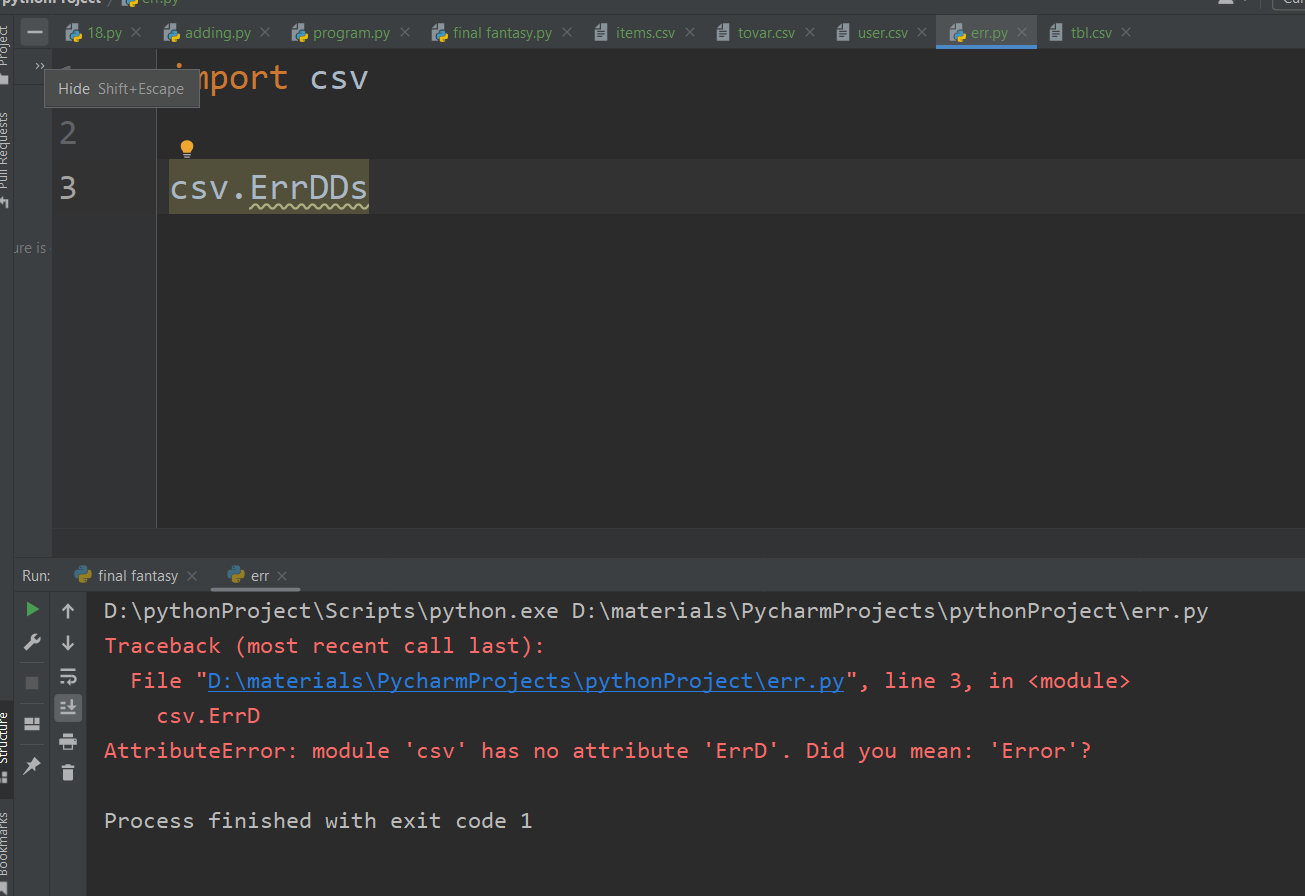


Рисунок 89 - AttributeError

IdentationError – Ошибка отступа, или же когда вы вы случайно поставили лишний пробел…

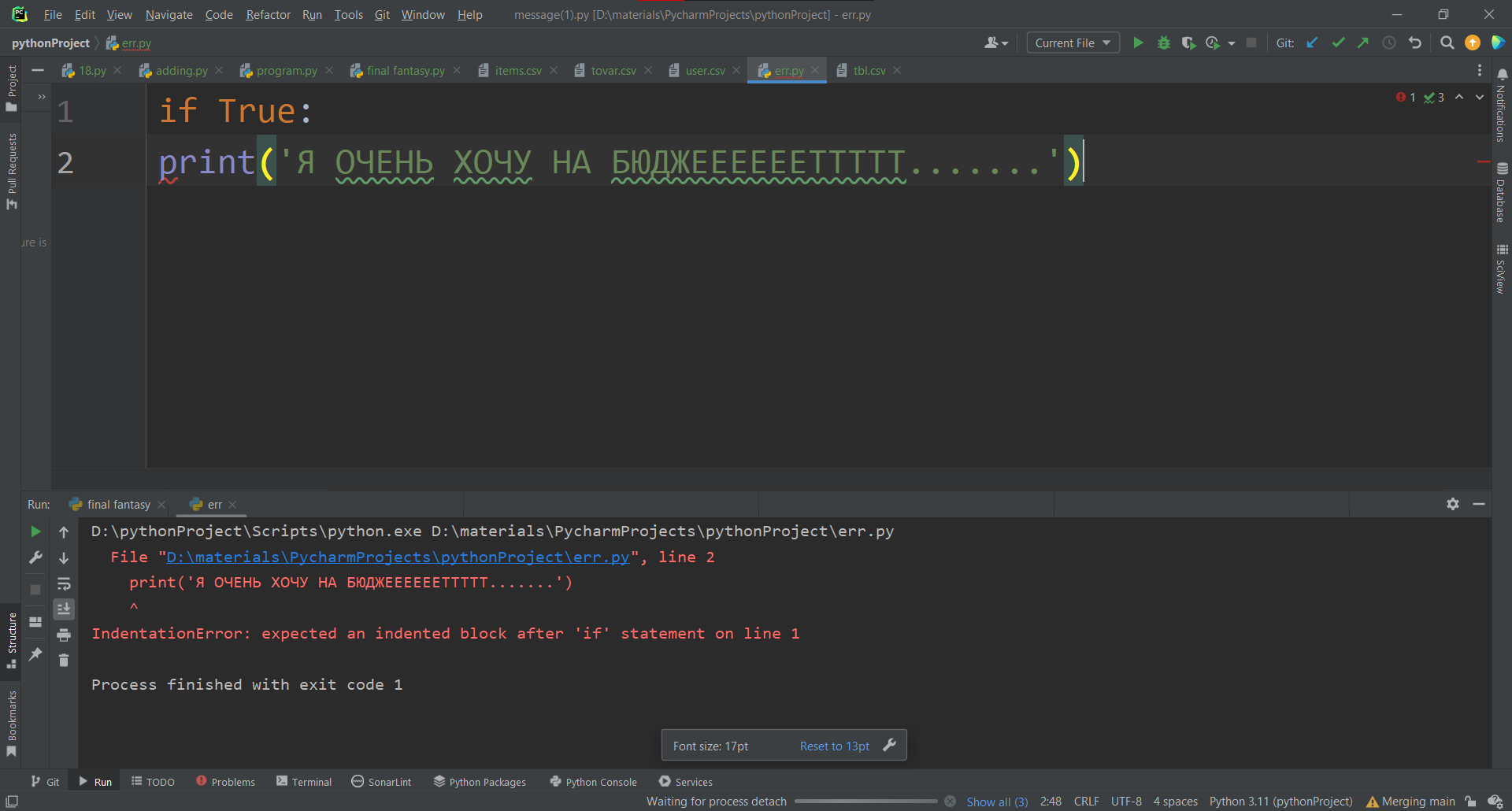


Рисунок 90 – IdentationError

Также некоторые другие ошибки, это не все, но их достаточно много…

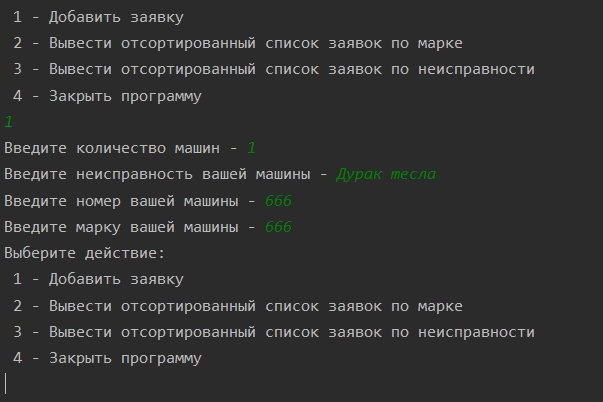
|  |  |
| --- | --- |
| AssertionError | Возникает при сбое утверждения утверждения. |
| AttributeError | Возникает при сбое присвоения атрибута или ссылки. |
| EOFError | Возникает при сбое присвоения атрибута или ссылки. |
| EOFError | Возникает, когда функция input() достигает условия конца файла. |
| FloatingPointError | Возникает при сбое операции с плавающей запятой. |
| KeyboardInterrupt | Возникает, когда пользователь нажимает клавишу прерывания (Ctrl + C или Delete). |

Вывод: познакомились со многими ошибками и исключениями на языке программирования «Python», а также поняли как обработать их в случае возникновения.

# ПРАКТИЧЕСКАЯ РАБОТА №6

## «Сохранение и таблицы tabulate»

Цель: решить задачи 17 и 18 с сортировками и выводом в файл.



17. Имеется N типов товаров (названия известны). Для каждого товара задано количество единиц этого товара, цена и вес единиц товара. Требуется загрузить контейнер (не превышая его известной грузоподъемности) товарами одного типа так, чтобы стоимость груза в контейнере была максимальной. Выполнить сортировку списка.

18. В автохозяйстве имеется N автомашин. Для каждого автомобиля заданы три характеристики: номер, марка машины и тип неисправности (или ее отсутствие). Необходимо составить сводные (по типам неисправностей) заявки на ремонт машин с указанием их номеров и марок. Выполнить сортировку списка.

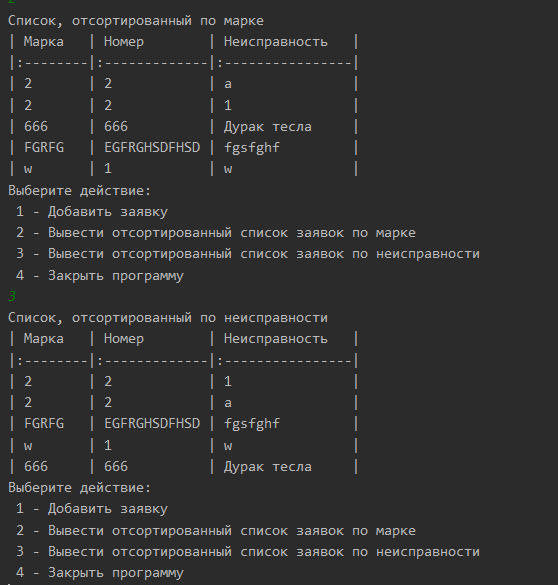


Рисунок – Задача 18

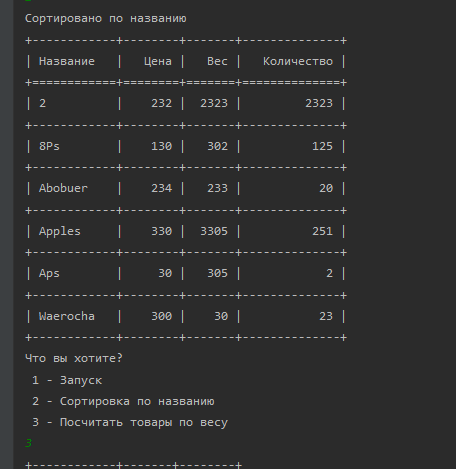


Рисунок – Задача 17

Вывод: решены задачи 17 и 18 с сортировками и выводом в файл.

# ИТОГОВАЯ РАБОТА

Цель работы:

Необходимо разработать консольный проект по работе с товарами

Все данные должны хранится в файлах. Всего должно быть 2 файла: 1 –

информация о пользователях(логин, пароль, роль), информация о

товарах(наименование, описание, поставщик, цена, доступное количество)

При запуске программы данные из файлов должны считываться и

помещаться в список/словарь. При запуске программы должно выводиться консольное меню:

1 – авторизироваться: При авторизации пользователь вводит логин и пароль.

Эти данные проверяются со значением в списке/словаре. Если у пользователя роль обычного пользователя то он попадает в личный кабинет

пользователя(Вывод всех товаров в библиотеку tabulate). Если у пользователя роль администратора то он попадает в личный кабинет

администратора(вывод всех товаров в библиотеку tabulate, редактирование товара, добавление товара, удаление товара)

2 – зарегистрироваться: Пользователь должен ввести логин, пароль,

повторный ввод пароля. Если первый ввод пароля и второй ввод пароля не совпадают необходимо выводить соответствующее сообщение и вызывать

функцию авторизации. Если логин уже занят необходимо выводить

соответствующее сообщение и вызывать функцию авторизации. По

умолчанию все пользователи регистрируются как обычные пользователи. В файле изначально должен быть 1 администрато

В данном приложение необходимо использовать функции,

документирующие комментарии к функциям, type hints, файлы. По

возможности необходимо использовать классы и объекты

классов(преобразование товаров и пользователей в объекты классов). В

зависимости от объема, реализованного в приложении будет зависеть оценка на зачете