Введение

Задача 1. Вывести на экран текст "Hello World".

Задача 2. Вывести на экран текущее название дня недели, название месяца и свое имя. Каждое слово должно быть в отдельной строке.

Задача 3. Вывести на экран пять строк из нулей, причем количество нулей в каждой строке равно номеру строки.

Задача 4. Вывести на экран прямоугольник, заполненный буквами А. Количество строк в прямоугольнике равно 5, количество столбцов равно 8.

Задача 5. Вывести на экран букву "W" из символов "*"

Задача 6. Пользователь вводит время в минутах и расстояние в километрах. Найдите скорость в м/с.

Задача 7. Пользователь вводит количество недель, месяцев, лет и получает количество дней за это время. Считать, что в месяце 30 дней.

Задача 8. Даны две переменных с некоторыми значениями. Поменять местами значения этих переменных

Задача 9. Поменяйте местами значения двух переменных, не используя дополнительных переменных.

Упражнение 1. Почтовый адрес. Напишите несколько строк кода, выводящих на экран ваше имя и почтовый адрес. Адрес напишите в формате, принятом в вашей стране. Никакого ввода от пользователя ваша первая программа принимать не будет, только вывод на экран и больше ничего.

Упражнение 2. Приветствие. Напишите программу, запрашивающую у пользователя его имя. В ответ на ввод на экране должно появиться приветствие с обращением по имени, введенному с клавиатуры ранее.

Упражнение 3. Площадь комнаты. Напишите программу, запрашивающую у пользователя длину и ширину комнаты. После ввода значений должен быть произведен расчет площади комнаты и выведен на экран. Длина и ширина комнаты должны вводиться в формате числа с плавающей запятой. Дополните ввод и вывод единицами измерения, принятыми в вашей стране. Это могут быть футы или метры.

Упражнение 4. Площадь садового участка. Создайте программу, запрашивающую у пользователя длину и ширину садового участка в футах. Выведите на экран площадь участка в акрах. Подсказка. В одном акре содержится 43 560 квадратных футов

Упражнение 5. Налоги и чаевые. Программа, которую вы напишете, должна начинаться с запроса у пользователя суммы заказа в ресторане. После этого должен быть произведен расчет налога и чаевых официанту. Вы можете использовать принятую в вашем регионе налоговую ставку для подсчета суммы сборов. В качестве чаевых мы оставим 18 % от стоимости заказа без учета налога. На выходе программа должна отобразить отдельно налог, сумму чаевых и итог, включая обе составляющие. Форматируйте вывод таким образом, чтобы все числа отображались с двумя знаками после запятой.

Упражнение 6. Сувениры и безделушки. Интернет-магазин занимается продажей различных сувениров и безделушек. Каждый сувенир весит 75 г, а безделушка – 112 г. Напишите программу, запрашивающую у пользователя количество тех и других покупок, после чего выведите на экран общий вес посылки.

Упражнение 7. Сложные проценты. Представьте, что вы открыли в банке сберегательный счет под 4 % годовых. Проценты банк рассчитывает в конце года и добавляет к сумме счета. Напишите программу, которая запрашивает у пользователя сумму первоначального депозита, после чего рассчитывает и выводит на экран сумму на счету в конце первого, второго и третьего годов. Все суммы должны быть округлены до двух знаков после запятой.

Упражнение 8. Арифметика. Создайте программу, которая запрашивает у пользователя два целых числа a и b, после чего выводит на экран результаты следующих математических операций: сумма a и b; разница между a и b; произведение a и b; частное от деления a на b; остаток от деления a на b; десятичный логарифм числа a; результат возведения числа a в степень b. Подсказка. Функцию log10 вы найдете в модуле math.

Упражнение 9. Рост. Многие люди на планете привыкли рассчитывать рост человека в футах и дюймах, даже если в их стране принята метрическая система. Напишите программу, которая будет запрашивать у пользователя количество футов, а затем дюймов в его росте. После этого она должна пересчитать рост в сантиметры и вывести его на экран. Подсказка. Один фут равен 12 дюймам, а один дюйм – 2,54 см.

Упражнение 10. Расстояние. Для этого упражнения вам необходимо будет написать программу, которая будет запрашивать у пользователя расстояние в футах. После этого она должна будет пересчитать это число в дюймы, ярды и мили и вывести на экран. Коэффициенты для пересчета единиц вы без труда найдете в интернете.

Упражнение 11. Единицы времени. Создайте программу, в которой пользователь сможет ввести временной промежуток в виде количества дней, часов, минут и секунд и узнать общее количество секунд, составляющее введенный отрезок

Упражнение 12. Текущее время. Модуль time в Python включает в себя несколько очень полезных функций для работы со временем. Одна из таких функций – asctime – считывает текущее системное время компьютера и возвращает его в удобном для восприятия виде. Используйте эту функцию для отображения на экране текущей даты и времени. Никакого ввода от пользователя на этот раз вам не потребуется.

Упражнение 13. Цельсий в Фаренгейт и Кельвин. Напишите программу, которая будет запрашивать у пользователя значение температуры в градусах Цельсия и отображать эквивалентный показатель по шкалам Фаренгейта и Кельвина. Необходимые коэффициенты и формулы для проведения расчетов нетрудно найти на просторах интернета.

Упражнение 14. Сумма цифр в числе. Разработайте программу, запрашивающую у пользователя целое четырехзначное число и подсчитывающую сумму составляющих его цифр. Например, если пользователь введет число 3141, программа должна вывести следующий результат: 3 + 1 + 4 + 1 = 9.

Упражнение 15. Сортировка трех чисел. Напишите программу, запрашивающую у пользователя три целых числа и выводящую их в упорядоченном виде – по возрастанию. Используйте функции min и max для нахождения наименьшего и наибольшего значений. Оставшееся число можно найти путем вычитания из суммы трех введенных чисел максимального и минимального.

Упражнение 16. Вчерашний хлеб. Пекарня продает хлеб по $3,49 за буханку. Скидка на вчерашний хлеб составляет 60 %. Напишите программу, которая будет запрашивать у пользователя количество приобретенных вчерашних буханок хлеба. В вывод на экран должны быть включены обычная цена за буханку, цена со скидкой и общая стоимость приобретенного хлеба. Все значения должны быть выведены на отдельных строках с соответствующими описаниями. Используйте для вывода формат с двумя знаками после запятой и выровненным разделителем.

Принятие решений if-elif-else

Упражнение 17. Чет или нечет? Напишите программу, запрашивающую у пользователя целое число и выводящую на экран информацию о том, является введенное число четным или нечетным.

Упражнение 18. Собачий возраст. Считается, что один год, прожитый собакой, эквивалентен семи человеческим годам. При этом зачастую не учитывается, что собаки становятся абсолютно взрослыми уже к двум годам. Таким образом, многие предпочитают каждый из первых двух лет жизни собаки приравнивать к 10,5 года человеческой жизни, а все последующие – к четырем. Напишите программу, которая будет переводить человеческий возраст в собачий с учетом указанной выше логики. Убедитесь, что программа корректно работает при пересчете возраста собаки меньше и больше двух лет. Также программа должна выводит

Упражнение 19. Гласные и согласные. 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

Упражнение 20. Сколько дней в месяце? Количество дней в месяце варьируется от 28 до 31. Очередная ваша программа должна запрашивать у пользователя название месяца и отображать количество дней в нем. Поскольку годы мы не учитываем, для февраля можно вывести сообщение о том, что этот месяц может состоять как из 28, так и из 29 дней, чтобы учесть фактор високосного года.

Упражнение 21. Счет за телефон. Тарифный план мобильной связи включает в себя 50 минут разговоров и 50 смс-сообщений за $15,00 в месяц. Каждая дополнительная минута стоит $0,25, а каждое дополнительное сообщение – $0,15. Все счета за телефон включают налог на поддержку кол-центров 911 в размере $0,44, и общая сумма, включающая сумму отчислений кол-центрам, облагается налогом в размере 5 %. Напишите программу, которая будет запрашивать у пользователя количество израсходованных за месяц минут разговора и смс-сообщений и отображать базовую сумму тарификации, сумму за

дополнительные минуты и сообщения, сумму отчислений кол-центрам 911, налог, а также итоговую сумму к оплате. При этом дополнительные звонки и сообщения необходимо выводить на экран только в случае их расходования. Убедитесь в том, что все суммы отображаются в формате с двумя знаками после запятой.

Упражнение 22. Високосный год? В большинстве случаев год насчитывает 365 дней. Но на самом деле нашей планете требуется чуть больше времени, чтобы полностью пройти по своей орбите вокруг Солнца. В результате для компенсации этой разницы был введен дополнительный день в феврале для особых годов, называемых високосными. Определить, високосный год или нет, можно, следуя такому алгоритму: если год делится на 400 без остатка, он високосный; если год (из оставшихся) делится на 100 без остатка, он НЕ високосный; если год (из оставшихся) делится на 4 без остатка, он високосный; все остальные года не являются високосными. Напишите программу, запрашивающую год у пользователя и выводящую сообщение о том, високосный ли он.

Упражнение 23. Следующий день. Разработайте программу, принимающую на вход дату и выводящую на экран дату, следующую за ней. Например, если пользователь введет дату, соответствующую 18 ноября 2019 года, на экран должен быть выведен следующий день, то есть 19 ноября 2019 года. Если входная дата будет представлять 30 ноября, то на выходе мы должны получить 1 декабря. И наконец, если ввести последний день года – 31 декабря 2019-го, пользователь должен увидеть на экране дату 1 января 2020-го. Дату пользователь должен вводить в три этапа: год, месяц и день. Убедитесь, что ваша программа корректно обрабатывает високосные годы.

Упражнение 24. Действительный номерной знак машины? Допустим, в нашей стране старый формат номерных знаков автомобилей состоял из трех заглавных букв, следом за которыми шли три цифры. После того как все возможные номера были использованы, формат был изменен на четыре цифры, предшествующие трем заглавным буквам. Напишите программу, запрашивающую у пользователя номерной знак машины и оповещающую о том, для какого формата подходит данная последовательность символов: для старого или нового. Если введенная последовательность не соответствует ни одному из двух форматов, укажите это в сообщении.

Циклы

Упражнение 25. Среднее значение. В данном упражнении вы должны 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.

Подсказка. Поскольку ноль является индикатором окончания ввода, его не нужно учитывать при расчете среднего.

Упражнение 26. Таблица со скидками. 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.

Упражнение 27. Таблица соотношения температур. Напишите программу для вывода таблицы соотношения температур, выраженных в градусах Цельсия и Фаренгейта. В таблице должны размещаться все температуры между 0 и 100 градусами Цельсия, кратные 10. Дополните таблицу подходящими заголовками. Формулу для перевода температуры из градусов Цельсия в градусы Фаренгейта можно легко найти на просторах интернета.

Упражнение 28. Никаких центов. 4 февраля 2013 года Королевским канадским монетным двором была выпущена последняя монета номиналом в один цент. После вывода центов из обращения все магазины вынуждены были изменить цены на товары таким образом, чтобы они стали кратны пяти центам (расчеты по банковским картам по-прежнему ведутся с учетом центов). И хотя продавцы вольны сами определять политику преобразования цен, большинство из них просто округлили цены до ближайших пяти центов. Напишите программу, запрашивающую у пользователя цены, пока не будет введена пустая строка. На первой строке выведите сумму всех введенных пользователем сумм, а на второй – сумму, которую покупатель должен заплатить наличными. Эта сумма должна быть округлена до ближайших пяти центов. Вычислить сумму для оплаты наличными можно, получив остаток от деления общей суммы в центах на 5. Если он будет меньше 2,5, следует округлить сумму вниз, а если больше – вверх

Упражнение 29. Билеты в зоопарк. 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. В качестве индикатора окончания ввода можно по традиции использовать пустую строку. Общую цену билетов стоит выводить в формате с двумя знаками после запятой.

Упражнение 30. Игра Fizz-Buzz. Fizz-Buzz – это известная игра, помогающая детям освоить в игровой форме правила деления. Участники садятся в круг, чтобы игра теоретически могла продолжаться бесконечно. Первый игрок говорит «Один» и передает ход тому, кто слева. Каждый следующий игрок должен мысленно прибавить к предыдущему числу единицу и произнести либо его, либо одно из ключевых слов: Fizz, если число без остатка делится на три, или Buzz, если на пять. Если соблюдаются оба этих условия, он произносит Fizz-Buzz. Игрок, не сумевший сказать правильное слово, выбывает из игры. Последний оставшийся игрок признается победителем. Разработайте программу, реализующую алгоритм игры Fizz-Buzz применительно к первым 100 числам. Каждый следующий ответ должен отображаться на новой строке.

Упражнение 31. Код Цезаря. Одним из первых в истории примеров шифрования считаются закодированные послания Юлия Цезаря. Римскому полководцу необходимо было посылать письменные приказы своим генералам, но он не желал, чтобы в случае чего их прочитали недруги. В результате он стал шифровать свои послания довольно простым методом, который впоследствии стали называть кодом Цезаря. Идея

шифрования была совершенно тривиальной и заключалась в циклическом сдвиге букв на три позиции. В итоге буква A превращалась в D, B – в E, C – в F и т. д. Последние три буквы алфавита переносились на начало. Таким образом, буква X становилась A, Y – B, а Z – C. Цифры и другие символы не подвергались шифрованию. Напишите программу, реализующую код Цезаря. Позвольте пользователю ввести фразу и количество символов для сдвига, после чего выведите результирующее сообщение. Убедитесь в том, что ваша программа шифрует как строчные, так и прописные буквы. Также должна быть возможность указывать отрицательный сдвиг, чтобы можно было использовать вашу программу для расшифровки фраз.

Упражнение 32. Палиндром или нет? 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. Примечание. Яибофобия (Aibohphobia) – это безрассудный страх палиндромов. Эти слова в русском и английском сами по себе являются палиндромами, что и привело к их образованию. Напротив, яилифилия (ailihphilia) характеризуется любовью к палиндромам. Объяснять образование этого слова нет нужды.

Упражнение 33. Таблица умножения. В данном упражнении вы создадите программу для отображения стандартной таблицы умножения от единицы до десяти. При этом ваша таблица умножения должна иметь заголовки над первой строкой и слева от первого столбца. Возможно, для выполнения этого упражнения вам придется озаботиться тем, чтобы выводить значения на экран без принудительного перевода каретки на строку ниже. Этого можно добиться, если последним аргументом функции print передать end="". Например, инструкция print("A") выведет на экран букву A, после чего автоматически перейдет на новую строку, тогда как print("A", end="") не станет переводить каретку, что позволит произвести следующий вывод в той же строке.

Упражнение 34. Двоичное число в десятичное. Напишите программу, которая будет преобразовывать двоичные значения (по основанию 2) в десятичные (по основанию 10). Пользователь должен ввести число в двоичном виде как строку, а программа – преобразовать его посимвольно в десятичный вид и вывести на экран с соответствующим сообщением.

Функции

Упражнение 35. Плата за такси. Представьте, что сумма за пользование услугами такси складывается из базового тарифа в размере $4,00 плюс $0,25 за каждые 140 м поездки. Напишите функцию, принимающую в качестве единственного параметра расстояние поездки в километрах и возвращающую итоговую сумму оплаты такси. В основной программе должен демонстрироваться результат вызова функции. Подсказка. Цены на такси могут меняться со временем. Используйте константы для представления базового тарифа и плавающей ставки, чтобы программу можно было легко обновлять при изменении цен.

Упражнение 36. Расчет стоимости доставки (23 строки) Интернет-магазин предоставляет услугу экспресс-доставки для части своих товаров по цене $10,95 за первый товар в заказе и $2,95 – за все последующие. Напишите функцию, принимающую в качестве

единственного параметра количество товаров в заказе и возвращающую общую сумму доставки. В основной программе должны производиться запрос количества позиций в заказе у пользователя и отображаться на экране сумма доставки.

Упражнение 37. Простое число? Простое число представляет собой число, большее единицы, которое без остатка делится лишь на само себя и единицу. Напишите функцию для определения того, является ли введенное число простым. Возвращаемое значение должно быть либо True, либо False. В основной программе, как и ожидается, пользователь должен ввести целое число и получить ответ о том, является ли оно простым.

Упражнение 38. Следующее простое число. В данном упражнении вам нужно написать функцию с именем nextPrime, которая находит и возвращает первое простое число, большее введенного числа n. Само число n должно передаваться в функцию в качестве единственного параметра. В основной программе запросите у пользователя это значение и выведите на экран первое простое число, большее заданного. Для решения этой задачи импортируйте функцию, созданную в упражнении 98.

Упражнение 39. Случайный пароль. Напишите функцию, которая будет генерировать случайный пароль. В пароле должно быть от 7 до 10 символов, при этом каждый символ должен быть случайным образом выбран из диапазона от 33 до 126 в таблице ASCII. Ваша функция не должна принимать на вход параметры, а возвращать будет сгенерированный пароль. В основной программе вы должны просто вывести созданный случайным образом пароль. Программа должна запускаться только в том случае, если она не импортирована в виде модуля в другой файл.

Упражнение 40. Случайный номерной знак. Представьте, что в вашем регионе устаревшим является формат номерных автомобильных знаков из трех букв, следом за которыми идут три цифры. Когда все номера такого шаблона закончились, было решено обновить формат, поставив в начало четыре цифры, а за ними три буквы. Напишите функцию, которая будет генерировать случайный номерной знак. При этом номера в старом и новом форматах должны создаваться примерно с одинаковой вероятностью. В основной программе нужно сгенерировать и вывести на экран случайный номерной знак.

Упражнение 41. Проверка пароля на надежность. В данном упражнении вам необходимо написать функцию, проверяющую введенный пароль на надежность. Определим как надежный пароль, состоящий минимум из восьми символов и включающий хотя бы по одной букве в верхнем и нижнем регистрах и как минимум одну цифру. Функция должна возвращать True, если переданный в качестве параметра пароль отвечает требованиям надежности. В противном случае возвращаемым значением должно быть False. В основной программе необходимо запросить у пользователя пароль и оповестить его о том, является ли он достаточно надежным.

Упражнение 42. Шестнадцатеричные и десятичные числа. Напишите две функции с именами hex2int и int2hex для конвертации значений из шестнадцатеричной системы счисления (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E и F) в десятичную (по основанию 10) и обратно. Функция hex2int должна принимать на вход строку с единственным символом в шестнадцатеричной системе и преобразовывать его в число от нуля до 15 в десятичной

системе, тогда как функция int2hex будет выполнять обратное действие – принимать десятичное число из диапазона от 0 до 15 и возвращать шестнадцатеричный эквивалент. Обе функции должны принимать единственный параметр со входным значением и возвращать преобразованное число. Удостоверьтесь, что функция hex2int корректно обрабатывает буквы в верхнем и нижнем регистрах. Если введенное пользователем значение выходит за допустимые границы, вы должны вывести сообщение об ошибке.

Упражнение 43. Дни в месяце. Напишите функцию для определения количества дней в конкретном месяце. Ваша функция должна принимать два параметра: номер месяца в виде целого числа в диапазоне от 1 до 12 и год, состоящий из четырех цифр. Убедитесь, что функция корректно обрабатывает февраль високосного года. В основной программе запросите у пользователя номер месяца и год и отобразите на экране количество дней в указанном месяце. При решении этой задачи вам может пригодиться написанная вами функция из упражнения 58.

Упражнение 44. Магические даты. Магическими называются даты, в которых произведение дня и месяца составляет последние две цифры года. Например, 10 июня 1960 года – магическая дата, поскольку 10 * 6 = 60. Напишите функцию, определяющую, является ли введенная дата магической. Используйте написанную функцию в главной программе для отображения всех магических дат в XX веке. Возможно, вам пригодится здесь функция, разработанная в упражнении 106.

Списки

Упражнение 45. Порядок сортировки. 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 вывести 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

Упражнение 46. Обратный порядок. 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.

Упражнение 47. Избавляемся от дубликатов. В данном упражнении вам предстоит разработать программу, в которой у пользователя будет запрошен список слов, пока он не оставит строку ввода пустой. После этого на экране должны быть показаны слова, введенные пользователем, но без повторов, – каждое по одному разу. При этом слова должны быть отображены в том же порядке, в каком их вводили с клавиатуры. Например, если пользователь на запрос программы введет следующий список слов: first second first third second программа должна вывести: first second third

Упражнение 48. Отрицательные, положительные и нули. 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.

Упражнение 49. Только слова. В данном упражнении вы напишете программу, которая будет выделять слова из строки, введенной пользователем. Начните с создания функции, принимающей на вход единственный строковый параметр. В качестве результата она должна возвращать список слов из строки с удаленными знаками препинания, в число которых должны входить точки, запятые, восклицательный и вопросительный знаки, дефисы, апострофы, двоеточия и точки с запятыми. При этом не нужно избавляться от знаков препинания, стоящих внутри слова, таких как апостроф, служащий в английском языке для обозначения сокращений. Например, если на вход функции дать строку "Contractions include: don’t, isn’t, and wouldn’t.", функция должна вернуть следующий список: ["Contractions", "include", "don’t", "isn’t", "and", "wouldn’t"]. В основной программе, как обычно, должна происходить демонстрация вашей функции. Запросите у пользователя строку и выведите на экран все составляющие ее слова с удаленными знаками препинания.

Упражнение 50. Список уже отсортирован? Напишите функцию, показывающую, отсортирован ли переданный ей в качестве параметра список (по возрастанию или убыванию). Функция должна возвращать True, если список отсортирован, и False в противном случае. В основной программе запросите у пользователя последовательность чисел для списка, после чего выведите сообщение о том, является ли этот список отсортированным изначально. Примечание. Убедитесь в том, что вы правильно обрабатываете пустые списки, а также списки, состоящие из единственного элемента.

Упражнение 51. Содержит ли список подмножество элементов? Подмножеством элементов, или подсписком (sublist), мы будем называть список, являющийся составной частью большего списка. Подсписок может содержать один элемент, множество элементов, а также быть пустым. Например, [1], [2], [3] и [4] являются подсписками списка [1, 2, 3, 4]. Список [2, 3] также входит в состав [1, 2, 3, 4], но при этом список [2, 4] не является подсписком [1, 2, 3, 4], поскольку в исходном списке числа 2 и 4 не соседствуют друг с другом. Пустой список может быть рассмотрен как подсписок для любого списка. Таким образом, список [] является подсписком [1, 2, 3, 4]. Также список является подсписком самого себя, то есть [1, 2, 3, 4] – это подсписок для [1, 2, 3, 4]. В рамках данного упражнения вам необходимо написать функцию isSublist, определяющую, является ли один список подсписком другого. На вход функции должны поступать два списка – larger и smaller. Функция должна возвращать значение True только в том случае, если список smaller является подсписком списка larger. Напишите также основную программу для демонстрации работы функции.

Упражнение 52. Все подсписки заданного списка. Используя определение подсписка из упражнения 133, напишите функцию, возвращающую список, содержащий все возможные подсписки заданного. Например, в число подсписков списка [1, 2, 3] входят следующие: [], [1], [2], [3], [1, 2], [2, 3] и [1, 2, 3]. Заметьте, что ваша функция должна вернуть как минимум один пустой список, гарантированно являющийся подсписком для

любого списка. Напишите основную программу, демонстрирующую работу функции применительно к нескольким исходным спискам.

Словари

Упражнение 53. Азбука Морзе. Азбука Морзе зашифровывает буквы и цифры при помощи точек и тире. В данном упражнении вам необходимо написать программу, в которой соответствие символов из азбуки Морзе будет храниться в виде словаря. В табл. 6.3 приведена та часть азбуки, которая вам понадобится при решении этого задания. В основной программе вам необходимо запросить у пользователя строку. После этого программа должна преобразовать его в соответствующую последовательность точек и тире, вставляя пробелы между отдельными символами. Символы, не представленные в таблице, можно игнорировать. Например, сообщение Hello, World! может быть представлено следующей последовательностью: .... . .–.. .–.. ––– .–– ––– .–. .–.. –..

Упражнение 54. Анаграммы. Анаграммами называются слова, образованные путем взаимной перестановки букв. В английском языке, например, анаграммами являются слова «live» и «evil», а в русском – «выбор» и «обрыв». Напишите программу, которая будет запрашивать у пользователя два слова, определять, являются ли они анаграммами, и выводить на экран ответ.

Упражнение 55. Карточка лото. Карточка для игры в лото состоит из пяти колонок, в каждой из которых – пять номеров. Колонки помечены буквами B, I, N, G и O. Под каждой буквой могут быть номера в своем диапазоне из 15 чисел. А именно под буквой B могут присутствовать числа от 1 до 15, под I – от 16 до 30, под N – от 31 до 45 и т. д. Напишите функцию, которая будет создавать случайную карточку лото и сохранять ее в словаре. Ключами словаря будут буквы B, I, N, G и O, а значениями – списки из пяти чисел, располагающихся в колонке под каждой буквой. Создайте еще одну функцию для отображения созданной карточки лото на экране со столбцами с заголовками.

Примечание. Как вы знаете, в настоящих карточках для игры в лото присутствуют пустые клетки в столбцах. Мы не будем реализовывать эту особенность в нашей программе.

Файлы и исключения

Упражнение 56. Отображаем начало файла. В операционных системах на базе Unix обычно присутствует утилита с названием head. Она выводит первые десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Напишите программу на Python, имитирующую поведение этой утилиты. Если файла, указанного пользователем, не существует, или не задан аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.

Упражнение 57. Отображаем конец файла. Продолжая тему предыдущего упражнения, в тех же операционных системах на базе Unix обычно есть и утилита с названием tail, которая отображает последние десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Реализуйте программу, которая будет делать то же самое. Так же, как и в упражнении 149, в случае отсутствия файла, указанного пользователем, или аргумента командной строки вам нужно вывести соответствующее сообщение. Данную задачу можно решить сразу несколькими способами. Например, можно все содержимое файла целиком загрузить в список и затем

выбрать из него последние десять элементов. А можно дважды прочитать содержимое файла: первый раз, чтобы посчитать количество строк, а второй – чтобы отобразить последние десять из них. При этом оба перечисленных подхода нежелательны, если речь идет о файлах достаточного объема. Существует решение, требующее единственного чтения файла и сохранения всех десяти строк за раз. В качестве дополнительного задания разработайте такой алгоритм

Упражнение 58. Сцепляем файлы. Продолжаем тему операционных систем на базе Unix, в которых обычно также есть утилита с названием cat, что является сокращением от concatenate (сцепить). Эта утилита выводит на экран объединенное содержимое нескольких файлов, имена которых передаются ей в качестве аргументов командной строки. При этом файлы сцепляются в том порядке, в котором указаны в аргументах. Напишите программу на Python, имитирующую работу этой утилиты. В процессе работы программа должна выдавать сообщения о том, какие файлы открыть не удается, и переходить к следующим файлам. Если программа была запущена без аргументов командной строки, на экран должно быть выведено соответствующее сообщение об ошибке.

Упражнение 59. Нумеруем строки в файле. Напишите программу, которая будет считывать содержимое файла, добавлять к считанным строкам порядковый номер и сохранять их в таком виде в новом файле. Имя исходного файла необходимо запросить у пользователя, так же, как и имя целевого файла. Каждая строка в созданном файле должна начинаться с ее номера, двоеточия и пробела, после чего должен идти текст строки из исходного файла.

Упражнение 60. Самое длинное слово в файле. В данном упражнении вы должны написать программу, которая будет находить самое длинное слово в файле. В качестве результата программа должна выводить на экран длину самого длинного слова и все слова такой длины. Для простоты принимайте за значимые буквы любые непробельные символы, включая цифры и знаки препинания.

Упражнение 61. Удаляем комментарии. Как вы знаете, в языке Python для создания комментариев в коде используется символ #. Комментарий начинается с этого символа и продолжается до конца строки – без возможности остановить его раньше. В данном упражнении вам предстоит написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Пройдите по всем строкам в файле на предмет поиска символа #. Обнаружив его, программа должна удалить все содержимое, начиная с этого символа и до конца строки. Для простоты не будем рассматривать ситуации, когда знак решетки встречается в середине строки. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.

Рекурсия

Упражнение 62. Сумма значений. Напишите программу, которая будет складывать числа, введенные пользователем. Сигналом к окончанию ввода должна служить пустая строка. Отобразите на экране сумму значений (или 0.0, если пользователь сразу же пропустил ввод). Решите эту задачу с использованием рекурсии. В вашей программе не должны

присутствовать циклы. Подсказка. В теле вашей рекурсивной функции должен производиться запрос одного числа у пользователя, после чего должно быть принято решение о том, производить ли еще один рекурсивный вызов. Ваша функция не должна принимать аргументов, а возвращать будет числовое значение.

Упражнение 63. Наибольший общий делитель. Евклид был греческим математиком, жившим около 2300 лет назад. Именно ему приписывается авторство эффективного рекурсивного алгоритма нахождения наибольшего общего делителя двух положительных чисел a и b. Этот алгоритм описывается так: Если b = 0, тогда Возвращаем a Иначе c = остаток от деления a на b Возвращаем наибольший общий делитель чисел b и c Напишите программу, реализующую алгоритм Евклида для определения наибольшего общего делителя двух положительных чисел, введенных пользователем. Проверьте программу на работоспособность с очень большими числами. Результат должен высчитываться очень быстро даже для огромных входных значений, состоящих из сотен чисел. Причина заключается в очень высокой эффективности данного алгоритма.

Упражнение 64. Римские цифры. Как ясно из названия, римские цифры появились еще в Древнем Риме. Но даже после падения Римской империи они продолжали использоваться на территории Европы вплоть до позднего Средневековья, а в определенных случаях применяются и сегодня. Римские цифры базируются на обозначениях M, D, C, L, X, V и I, соответствующих числам 1000, 500, 100, 50, 10, 5 и 1. В основном римские цифры в составляющих их числах располагаются в порядке убывания – от больших к меньшим. В этом случае итоговое число равно сумме всех составляющих его римских цифр. Если цифры меньшего номинала предшествуют цифрам большего номинала, то первые вычитаются из вторых и итог прибавляется к общей сумме. В такой манере могут использоваться римские цифры C, X и I. При этом вычитание производится только из чисел, максимум в десять раз превосходящих вычитаемое. Таким образом, цифра I может предшествовать V или X, но не может вычитаться из L, C, D или M. А значит, число 99 должно быть написано как XCIX, а не IC. Создайте рекурсивную функцию, которая будет переводить римскую запись чисел в десятичную. Функция должна обрабатывать один или два символа в начале строки, после чего будет производиться ее рекурсивный вызов для оставшихся символов. Используйте пустую строку с возвращаемым значением 0 в качестве базового случая. Также напишите основную программу, которая будет запрашивать у пользователя число, введенное римскими цифрами, и отображать его десятичный эквивалент. При этом можно сделать допуск о том, что пользователь всегда вводит корректное число, так что обработку ошибок вам реализовывать не нужно.

Упражнение 65. Возможный размен. Напишите программу, которая будет определять, можно ли составить конкретную сумму из определенного количества монет. Например, можно набрать доллар из четырех монет номиналом в 25 центов. Но при помощи пяти монет доллар никак не собрать. При этом из шести монет это снова возможно, если взять три монеты по 25 центов, две – по 10 и одну номиналом в 5 центов. Также возможно собрать сумму $1,25 из пяти или восьми монет, но не удастся это сделать с четырьмя, шестью или семью монетами. Ваша основная программа должна запрашивать у пользователя искомую сумму и количество монет. На выходе вы должны получить сообщение о том, можно или нет собрать введенную сумму при помощи заданного

количества монет. Представьте, что для решения этой задачи в вашем распоряжении есть монеты номиналом 1, 5, 10 и 25 центов. Также ваша программа должна включать рекурсивный алгоритм. Циклов в ней быть не должно.

Упражнение 66. Декодирование на основе длин серий. Кодирование на основе длин серий представляет собой простую технику сжатия информации, которая демонстрирует свою эффективность при наличии множества соседствующих друг с другом повторяющихся значений. Сжатие достигается за счет замены целой группы повторяющихся значений на однократное его упоминание с отдельно хранящимся счетчиком повторов. Например, список ["A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "B", "B", "B", "B", "A", "A", "A", "A", "A", "A", "B"] может быть закодирован в следующем виде: ["A", 12, "B", 4, "A", 6, "B", 1]. Процесс декодирования заключается в размножении каждого элемента в соответствии со счетчиком. Напишите рекурсивную функцию для декодирования списка, кодированного на основе длин серий. В качестве единственного аргумента функция должна принимать закодированный соответствующим образом список. На выходе должен получиться расшифрованный список элементов. В основной программе продемонстрируйте работу алгоритма декодирования на примере представленного здесь списка.

Упражнение 67. Кодирование на основе длин серий. Напишите рекурсивную функцию, реализующую алгоритм кодирования на основе длин серий, описанный в упражнении 185. На вход функции должен поступать список или строка, а на выходе будет закодированный список. В основной программе запросите у пользователя строку, сожмите ее при помощи своей функции и отобразите на экране кодированный список. Подсказка. Возможно, вам придется поместить цикл внутрь тела своей рекурсивной функции