Лабораторная работа №1

|  |
| --- |
| **Вариант №1**  **Задание №1.** Объявите функцию, которая принимает три стороны треугольника и проверяет, можно ли из переданных аргументов составить треугольник. (Напомню, что у любого треугольника длина третьей стороны должна быть меньше суммы двух других). Если проверка проходит, вернуть значение True, иначе - значение False. Вызовите функцию и выведите на экран ДА, если функция возвратит True и НЕТ, если возвращается значение False.  **Задание № 2.** Вводится целое положительное число N. Необходимо написать рекурсивную функцию, которая отображает на экране последовательность целых чисел от 1 до N (включительно). Каждое число выводится с новой строки. |
| **Вариант №2**  **Задание №1.** Объявите функцию для проверки числа на четность (возвращается True, если переданное число четное и False, если число нечетное). После объявления функции в цикле необходимо считывать целое числовое значение (функцией input), пока не поступит число 1. Если введенное значение четное (проверяется с помощью заданной функции), то оно выводится на экран.  **Задание №2.** Вводится список целых чисел в одну строчку через пробел. Необходимо вычислить сумму введенных значений, используя рекурсивную функцию (для перебора элементов списка). Функция должна возвращать значение суммы. (Выводить на экран она ничего не должна). Вызвать рекурсивную функцию для введенного с клавиатуры списка и отобразить значение вычисленной суммы на экране. |
| **Вариант №3**  **Задание №1.** Объявите функцию для проверки числа на нечетность (возвращается True, если переданное число нечетное и False, если число четное). После объявления функции прочитайте (с помощью функции input) список целых значений, записанных в одну строку через пробел. И, используя генератор списков и объявленную функцию, сформируйте список из нечетных значений на основе введенного исходного списка. Результат отобразите на экране.  **Задание № 2.** Вводится натуральное число n. Необходимо с помощью рекурсивной функции сформировать последовательность чисел Фибоначчи по правилу: первые два числа равны 1 и 1, а каждое следующее значение равно сумме двух предыдущих. Пример такой последовательности для первых 7 чисел: 1, 1, 2, 3, 5, 8, 13, ... Функция должна возвращать список сформированной последовательности длиной n. Вызовите эту функцию и отобразите на экране полученный список. |
| **Вариант №4**  **Задание №1.** Вводится слово. Если это слово RECT, то следует объявить функцию с именем get\_sq, двумя параметрами, вычисляющую площадь прямоугольника и возвращающую вычисленное значение. Если же введенное слово не RECT (любое другое), то объявляется функция с тем же именем get\_sq, с двумя параметрами для вычисления периметра прямоугольника. Вычисленное значение возвращается функцией. В конце программы вызовите функцию с двумя аргументами и выведите результат на экран.  **Задание № 2.** Вводится натуральное число n. Необходимо с помощью рекурсивной функции вычислить факториал числа n. Напомню, что факториал числа, равен: n! = 1 \* 2 \* 3 \*...\* n. Функция должна возвращать вычисленное значение. Вызовите эту функцию и отобразите на экране результат ее работы. |
| **Вариант №5**  **Задание №1.** Объявите функцию, которая принимает строку (в качестве аргумента) и возвращает False, если длина строки меньше трех символов. Иначе возвращается значение True. После объявления функции прочитайте (с помощью функции input) список названий городов, записанных в одну строку через пробел. Затем, используя генератор списка и созданную функцию, сформируйте список из названий городов длиной не менее трех символов на основе введенного исходного списка. Результат отобразите на экране.  **Задание № 2.** Имеется следующий многомерный список: d = [1, 2, [True, False], ["Москва", "Уфа", [100, 101], [True, [1, -1]]], 7.89] С помощью рекурсивной функции создать на его основе одномерный список из значений элементов списка d. Функция должна возвращать новый созданный одномерный список. Вызовите эту функцию и отобразите на экране результат ее работы. |
| **Вариант №6**  **Задание №1.** Объявите функцию, которая принимает строку (в качестве аргумента) и возвращает два значения в виде кортежа: переданная строка и ее длина. После объявления функции прочитайте (с помощью функции input) список названий городов, записанных в одну строку через пробел. Затем, используя генератор словарей и созданную функцию, сформируйте словарь d в формате: d = {<город>: <число символов>, ...,<город N> :<число символов> }  **Задание № 2**. Объявите анонимную (лямбда) функцию для вычисления модуля числа (то есть, отрицательные числа нужно делать положительными). Вызовите эту функцию для введенного с клавиатуры вещественного числа и отобразите результат на экране. |
| **Вариант №7**  **Задание №1.** Вводится список целых чисел в одну строчку через пробел. Необходимо задать функцию с именем get\_sq, которая принимает два аргумента (максимальное и минимальное значения из списка) и возвращает их произведение. Вызовите эту функцию и отобразите на экране полученное числовое значение.  Подсказка: для передачи аргументов функции используйте функции max и min для введенного списка чисел.  **Задание № 2.** Объявите обычную функцию, которая первым параметром принимает список, а вторым – ссылку на функцию. Она должна возвращать отфильтрованный список. Делается это по следующему алгоритму. В цикле переберите элементы списка, для каждого вызовите, переданную вторым параметром, функцию и если она возвращает True, то элемент должен быть добавлен в возвращаемый основной функцией список. Вызовите объявленную функцию, первым параметром передайте список, а вторым запишите лямбда-функцию для отбора элементов из передаваемого списка. Результат работы функции выведите на экран. |
| **Вариант №8**  **Задание №1.** Напишите функцию, которая проверяет корректность переданного ей email-адреса в виде строки. Будем полагать, что адрес верен, если он обязательно содержит символы '@' и '.', а все остальные символы могут принимать значения: 'a-z', 'A-Z', '0-9' и '\_'. Если email верен, то функция возвращает True, иначе – False. После объявления функции прочитайте (с помощью функции input) строку с email-адресом и вызовите функцию с этим аргументом. Результат отобразите на экране.  **Задание № 2.** Объявите анонимную (лямбда) функцию для определения вхождения в строку фрагмента "ra". То есть, функция должна возвращать True, если такой фрагмент присутствует в строке и False – в противном случае. Вызовите эту функцию для введенной строки и отобразите результат ее работы на экране. |
| **Вариант №9**  **Задание №1.** Объявите функцию, которая принимает список, находит максимальное, минимальное и сумму значений этого списка и возвращает все вычисленные результаты в виде кортежа. Вызовите объявленную функцию со списком чисел, введенными с клавиатуры и отобразите результат работы функции на экране.  **Задание № 2.** С помощью рекурсивной функции найдите минимальное значение элементов, переданного ей кортежа. Функция должна возвращать найденное минимальное значение. Вызовите эту функцию и отобразите результат ее работы на экране. |
| **Вариант №10**  **Задание №1.** Запишите функцию без параметров, которая считывает с клавиатуры фамилию, имя и отчество, записанные в одну строку через пробел, и возвращает три строки: фамилия, имя, отчество (в виде кортежа). Вызовите объявленную функцию и отобразите на экране строку в формате: «Уважаемый, Имя Фамилия! Вы верно выполнили данную работу!»  **Задание №2.** С помощью рекурсивной функции сформируйте список из отфильтрованных значений переданного ей кортежа. Фильтрацию определить с помощью анонимной функции, которая передается рекурсивной функции вторым аргументом (первым аргументом передается кортеж). Вызовите эту функцию и отобразите результат ее работы на экране. |

Лабораторная работа №2

|  |
| --- |
| **Вариант №1**  **Задание №1.** Используя замыкания функций, определите вложенную функцию, которая бы увеличивала значение переданного параметра на 5 и возвращала бы вычисленный результат. Вызовите внешнюю функцию для получения ссылки на внутреннюю функцию и присвойте ее переменной с именем cnt. Затем, вызовите внутреннюю функцию через переменную cnt со значением k, введенным с клавиатуры.  **Задание № 2.** Объявите функцию с именем get\_sq, которая вычисляет площадь прямоугольника по двум параметрам: width и height – ширина и высота прямоугольника и возвращает результат. Определите декоратор для этой функции с именем (внешней функции) func\_show, который отображает результат на экране в виде строки (без кавычек): "Площадь прямоугольника: " Вызовите декорированную функцию get\_sq. |
| **Вариант №2**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая заключает строку s (s – строка, параметр внутренней функции) в произвольный тег, содержащийся в переменной tag – параметре внешней функции. Далее, на вход программы поступает две строки: первая с тегом, вторая с некоторым содержимым. Вторую строку нужно поместить в тег из первой строки с помощью реализованного замыкания. Результат выведите на экран.  **Задание №2.** На вход программы поступает строка из целых чисел, записанных через пробел. Напишите функцию get\_list, которая преобразовывает эту строку в список из целых чисел и возвращает его. Определите декоратор для этой функции, который сортирует список чисел, полученный из вызываемой в нем функции. Результат сортировки должен возвращаться при вызове декоратора. Вызовите декорированную функцию get\_list и отобразите полученный отсортированный список на экране. |
| **Вариант №3**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая преобразует строку из списка целых чисел, записанных через пробел, либо в список, либо в кортеж. Тип коллекции определяется параметром type внешней функции. Если type = 'list', то используется список, иначе – кортеж. Далее, на вход программы поступает две строки: первая – это значение для параметра type; вторая – список целых чисел, записанных через пробел. С помощью реализованного замыкания преобразовать эту строку в соответствующую коллекцию. Результат работы замыкания выведите на экран.  **Задание № 2.** Вводятся два списка (каждый с новой строки) из слов, записанных через пробел. Имеется функция, которая преобразовывает эти две строки в два списка слов и возвращает эти списки. Определите декоратор для этой функции, который из этих двух списков формирует словарь, в котором ключами являются слова из первого списка, а значениями – соответствующие элементы из второго списка. Полученный словарь должен возвращаться при вызове декоратора. Примените декоратор к первой функции и вызовите ее. Результат (словарь) отобразите на экране. |
| **Вариант №4**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая из переданного ей списка строк формирует многострочную строку вида:  <ol>  <li>строка\_1</li>  …  <li>строка\_N</li>  </ol>  и возвращает ее. Где строка\_1, строка\_2, … - это строки из переданного функции списка. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.  **Задание № 2.** Объявите функцию с именем to\_lat, которая принимает строку на кириллице и преобразовывает ее в латиницу, используя следующий словарь для замены русских букв на соответствующее латинское написание: t = {'ё': 'yo', 'а': 'a', 'б': 'b', 'в': 'v', 'г': 'g', 'д': 'd', 'е': 'e', 'ж': 'zh', 'з': 'z', 'и': 'i', 'й': 'y', 'к': 'k', 'л': 'l', 'м': 'm', 'н': 'n', 'о': 'o', 'п': 'p', 'р': 'r', 'с': 's', 'т': 't', 'у': 'u', 'ф': 'f', 'х': 'h', 'ц': 'c', 'ч': 'ch', 'ш': 'sh', 'щ': 'shch', 'ъ': '', 'ы': 'y', 'ь': '', 'э': 'e', 'ю': 'yu', 'я': 'ya'} Функция должна возвращать преобразованную строку. Замены делать без учета регистра (исходную строку перевести в нижний регистр – малые буквы). Все небуквенные символы "! ?:;.,\_" превращать в символ '-' (дефиса). Определите декоратор для этой функции, который несколько подряд идущих дефисов, превращает в один дефис. Полученная строка должна возвращаться при вызове декоратора. Примените декоратор к функции to\_lat и вызовите ее Результат работы декорированной функции отобразите на экране. |
| **Вариант №5**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве параметров фамилию и имя, а затем, заносит в шаблон эти данные. Сам шаблон – это строка, которая передается внешней функции и, например, может иметь такой вид: «Уважаемый %F%, %N%! Вы делаете работу по замыканиям функций.» Здесь %F% - это фрагмент куда нужно подставить фамилию, а %N% - фрагмент, куда нужно подставить имя. (Шаблон может быть и другим, вы это определяете сами). Здесь важно, чтобы внутренняя функция умела подставлять данные в шаблон, формировать новую строку и возвращать результат. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.  **Задание № 2.** Вводится строка целых чисел через пробел. Напишите функцию, которая преобразовывает эту строку в список чисел и возвращает их сумму. Определите декоратор для этой функции, который имеет один параметр start – начальное значение суммы. Примените декоратор со значением start=5 к функции и вызовите декорированную функцию. Результат отобразите на экране. |
| **Вариант №6**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая бы все повторяющиеся символы заменяла одним другим указанным символов. Какие повторяющиеся символы искать и на что заменять, определяются параметрами внешней функции. Внутренней функции передается только строка для преобразования. Преобразованная (сформированная) строка должна возвращаться внутренней функцией. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы. **Задание № 2.** Объявите функцию, которая возвращает переданную ей строку в нижнем регистре (с малыми буквами). Определите декоратор для этой функции, который имеет один параметр tag, определяющий строку с названием тега (начальное значение параметра tag равно "h1"). Этот декоратор должен заключать возвращенную функцией строку в тег tag и возвращать результат. Пример заключения строки "python" в тег h1: <h1>python<h1>  Примените декоратор со значением tag="div" к функции и вызовите декорированную функцию. Результат отобразите на экране. |
| **Вариант №7**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая на основе двух параметров вычисляет площадь фигуры. Какой именно фигуры: треугольника или прямоугольника, определяется параметром type внешней функции. Если type принимает значение 0, то вычисляется площадь треугольника, а иначе – прямоугольника. По умолчанию параметр type должен быть равен 0. Вычисленное значение должно возвращаться внутренней функцией. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.  **Задание № 2.** Объявите функцию, которая вычисляет периметр многоугольника и возвращает вычисленное значение. Длины сторон многоугольника передаются в виде коллекции (списка или кортежа). Определите декоратор для этой функции, который выводит на экран сообщение: «Периметр фигуры равен = » Примените декоратор к функции и вызовите декорированную функцию. |
| **Вариант №8**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая принимает два параметра a, b, а затем, возвращает строку в формате: «Для значений a, b функция f(a,b) = » где число – это вычисленное значение функции f. Ссылка на f передается как аргумент внешней функции. Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы. Функцию f придумайте самостоятельно (она должна что то делать с двумя параметрами a, b и возвращать результат).  **Задание № 2.** Объявите функцию, которая вычисляет площадь круга и возвращает вычисленное значение. В качестве аргумента ей передается значение радиуса. Определите декоратор для этой функции, который выводит на экран сообщение: «Площадь круга равна = <число>» В строке выведите числовое значение с точностью до сотых. Примените декоратор к функции и вызовите декорированную функцию. |
| **Вариант №9**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента коллекцию (список или кортеж) и возвращает или минимальное значение, или максимальное, в зависимости от значения параметра type внешней функции. Если type равен «max», то возвращается максимальное значение, иначе – минимальное. По умолчанию type должно принимать значение «max». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.  **Задание № 2.** Объявите функцию, которая принимает строку на кириллице и преобразовывает ее в латиницу, используя следующий словарь для замены русских букв на соответствующее латинское написание: t = {'ё': 'yo', 'а': 'a', 'б': 'b', 'в': 'v', 'г': 'g', 'д': 'd', 'е': 'e', 'ж': 'zh', 'з': 'z', 'и': 'i', 'й': 'y', 'к': 'k', 'л': 'l', 'м': 'm', 'н': 'n', 'о': 'o', 'п': 'p', 'р': 'r', 'с': 's', 'т': 't', 'у': 'u', 'ф': 'f', 'х': 'h', 'ц': 'c', 'ч': 'ch', 'ш': 'sh', 'щ': 'shch', 'ъ': '', 'ы': 'y', 'ь': '', 'э': 'e', 'ю': 'yu', 'я': 'ya'} Функция должна возвращать преобразованную строку. Замены делать без учета регистра (исходную строку перевести в нижний регистр – малые буквы). Определите декоратор с параметром chars и начальным значением " !?", который данные символы преобразует в символ "-" и, кроме того, все подряд идущие дефисы (например, "--" или "---") приводит к одному дефису. Полученный результат должен возвращаться в виде строки. Примените декоратор со значением chars="?!:;,. " к функции и вызовите декорированную функцию. Результат отобразите на экране. |
| **Вариант №10**  **Задание №1.** Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента список целых чисел и удаляет из него все четные или нечетные значения в зависимости от значения параметра type. Если type равен «even», то удаляются четные значения, иначе – нечетные. По умолчанию type должно принимать значение «even». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы. **Задание №2.** Объявите функцию, которая принимает строку, удаляет из нее все подряд идущие пробелы и переводит ее в нижний регистр – малые буквы. Результат (строка) возвращается функцией. Определите декоратор, который строку, возвращенную функцией, переводит в азбуку Морзе, используя следующий словарь для замены русских букв и символа пробела на соответствующие последовательности из точек и тире: morze = {'а': '.-', 'б': '-...', 'в': '.--', 'г': '--.', 'д': '-..', 'е': '.', 'ё': '.', 'ж': '...-', 'з': '--..', 'и': '..', 'й': '.---', 'к': '-.-', 'л': '.-..', 'м': '--', 'н': '-.', 'о': '---', 'п': '.--.', 'р': '.-.', 'с': '...', 'т': '-', 'у': '..-', 'ф': '..- .', 'х': '....', 'ц': '-.-.', 'ч': '---.', 'ш': '----', 'щ': '--.-', 'ъ': '--.--', 'ы': '-.--', 'ь': '-..-', 'э': '..-..', 'ю': '..--', 'я': '.-.-', ' ': '-···-'} Преобразованная строка возвращается декоратором. Примените декоратор к функции и вызовите декорированную функцию. Результат работы отобразите на экране. |

Лабораторная работа №3

|  |
| --- |
| **Вариант №1**  **Задание №1.** Импортируйте в программу стандартный модуль time и, используя одноименную функцию time из этого модуля, определите время работы программы, которая сначала сохраняет в файл с именем out.txt строку: «Работа №3. Импорт модулей и пакетов. Работа с файлами» а, затем, читает из этого файла его содержимое и отображает результат (чтения из файла) на экран. Также на экран нужно вывести время работы программы. |
| **Вариант №2**  **Задание №1.** Импортируйте в программу стандартный модуль math и, используя функцию log2, вычислите логарифм по основанию 2 для целых чисел от 1 до 10. Результат представьте в виде списка, который сформируйте с помощью генератора списков. Полученный список сохраните в файл в бинарном режиме доступа. А, затем, прочитайте его содержимое в другой список. Выведите на экран прочитанные данные (список). |
| **Вариант №3**  **Задание №1.** Импортируйте в программу стандартный модуль random и, используя функцию randint, сгенерируйте 100 случайных чисел в диапазоне [-2; 5]. Все эти числа следует представить в виде списка, сформированного с помощью генератора списков. Сохраните полученные данные в файл в текстовом режиме доступа. Затем, прочитайте из него данные и поместите в другой список. Выведите на экран полученный список. |
| **Вариант №4**  **Задание №1.** Импортируйте в программу стандартный модуль math и, используя функцию log10, вычислите логарифм по основанию 10 для целых чисел от 1 до 10. Результат представьте в виде множества, который сформируйте с помощью генератора множеств. Полученный список сохраните в файл в текстовом режиме доступа в виде строки вещественных чисел с точностью до тысячных, записанных через пробел. Затем, прочитайте содержимое этого файла и поместите в другое множество. Выведите на экран прочитанные данные (множество). |
| **Вариант №5**  **Задание №1.** Импортируйте в программу стандартный модуль time и, используя одноименную функцию time из этого модуля, определите время работы программы, которая сначала сохраняет в файл с именем out.dat кортеж (в бинарном режиме доступа): books = (('Пушкин А. С.', 'Пикова дама'), ('Булгаков М. А.', 'Мастер и Маргарита'), ('Конан Дойль', 'Собака Баскервилей')) а, затем, читает из этого файла (также в бинарном режиме) его содержимое и отображает прочитанный кортеж на экран. Также на экран нужно вывести время работы программы. |
| **Вариант №6**  **Задание №1.** Импортируйте в программу стандартный модуль random и, используя одноименную функцию random, сгенерируйте 100 случайных чисел. Все эти числа следует представить в виде списка, сформированного с помощью генератора списков. Сохраните полученные данные в файл в бинарном режиме доступа. Затем, прочитайте из него данные и поместите в другой список. Выведите на экран полученный список. |
| **Вариант №7**  **Задание №1.** Импортируйте в программу стандартный модуль math и, используя функцию factorial, вычислите факториал для целых чисел от 1 до 10. Результат представьте в виде словаря, ключами которого являются аргументы функции (числа от 1 до 10), а значениями – вычисленный факториал. Сформируйте словарь с помощью генератора словарей. Полученный словарь сохраните в файл в текстовом режиме доступа (формат данных придумайте сами). Затем, прочитайте содержимое этого файла и поместите в другой словарь. Выведите на экран прочитанные данные (словарь). |
| **Вариант №8**  **Задание №1.** Импортируйте в программу стандартный модуль random и, используя функцию randint, сформируйте двумерный список, размером 5x5 элементов, со случайными значениями в диапазоне [-5; 5]. Двумерный список следует сформировать с помощью вложенных генераторов списков. Сохраните полученные данные в файл в бинарном режиме доступа. Затем, прочитайте из него данные и поместите в другой список. Выведите на экран полученный двумерный список. |
| **Вариант №9**  **Задание №1.** Импортируйте в программу стандартный модуль math и, используя функцию sin, вычислите синус для значений от -π до π с шагом 0.1. Результат представьте в виде списка. Сформируйте список с помощью генератора списков. Полученный список сохраните в файл в бинарном режиме доступа. Затем, прочитайте содержимое этого файла и поместите в другой список. Выведите на экран прочитанные данные (список). |
| **Вариант №10**  **Задание №1.** Импортируйте в программу стандартный модуль time, используя одноименную функцию time из этого модуля, определите время работы программы, которая сохраняет в файл таблицу умножения для целых чисел от 1 до 10 (каждое умножается от 1 до 10). Затем, открывает этот же файл на чтение и формирует из прочитанных данных двумерный кортеж в формате: ((2, 1, 2), (2, 2, 4), (2, 3, 6), …,) Выведите на экран кортеж и время работы программы. |

Лабораторная работа №4

|  |
| --- |
| **Вариант №1**  **Задание №1.** Запишите выражение для генератора, который бы возвращал числа от 2 до 100 000 с шагом 0.5 (то есть, 2.0, 2.5, 3, 3.5, ..., 100 000). Выберите первые пять значений из этого генератора и отобразите их на экране.  **Задание № 2.** Запишите функцию-генератор для формирования последовательности Фибоначчи. Это последовательность чисел, в которой каждое последующее равно сумме двух предыдущих. Первые два значения равны 1 и 1. Например, первые семь членов последовательности Фибоначчи, имеют значения: 1, 1, 2, 3, 5, 8, 13, ... Вывести первые 15 чисел этой последовательности, сформированной функцией- генератором. |
| **Вариант №2**  **Задание №1.** Используя символы малых букв латинского алфавита (строка ascii\_lowercase): from string import ascii\_lowercase запишите генератор, который бы возвращал все сочетания из двух букв латинского алфавита. Выведите первые 50 сочетаний на экран.  **Задание №2.** Имеется функция f(x) = 0,1x2 + 5x - 2. Необходимо записать функцию-генератор, которая бы выдавала значения этой функции в диапазоне [a; b] с шагом 0,01. Параметры a, b – это параметры, определенные у функции- генератора. Вызовите эту функцию со значениями a=-5, b = 7 и выведите на экран первые 20 значений, которые она возвратит. Какие-либо коллекции в программе не использовать. |
| **Вариант №3**  **Задание №1.** Вводится два целых числа a, b (a < b). Необходимо определить генератор, который бы возвращал модули чисел, в диапазоне [a; b]. Вывести первые четыре значения (или меньше, если значений меньше четырех) этого генератора на экран. (Программа должна быть универсальной и не меняться для разного количества генерируемых данных).  **Задание № 2.** Задайте функцию-генератор, которая на основе символов строки chars: from string import ascii\_lowercase, ascii\_uppercase chars = ascii\_lowercase + ascii\_uppercase + "0123456789!?@#$\*" формирует и выдает случайно сгенерированные пароли длиной 12 символов. Количество выдаваемых паролей функцией должно быть неограниченным. Случайный выбор символа из последовательности chars можно реализовать с помощью функции choice модуля random. Выведите первые пять сгенерированных паролей на экран. |
| **Вариант №4**  **Задание №1.** Используя модуль random и его функцию randint(a, b) для генерации целых чисел в диапазоне [a; b], а также символы малых букв латинского алфавита (строка ascii\_lowercase): from string import ascii\_lowercase Запишите генератор, который бы возвращал 1 000 000 строк из случайно выбранных символов. Длина каждой строки должна быть 10 символов. Выведите на экран первые пять сгенерированных строк.  **Задание № 2.** Написать функцию-генератор, которая должна выдавать значения арифметической последовательности: an = a n-1 + d. Значения a0, d, - указываются как параметры этой функции. Вызовите эту функцию со значениями аргументов a0 = −0,5, d = 0.1 и выведите на экран первые 50 значений, возвращенных этой функцией. Какие-либо коллекции в программе не использовать. |
| **Вариант №5**  **Задание №1.** Имеется список из названий городов: cities = ["Москва", "Ульяновск", "Самара", "Уфа", "Омск", "Тула"] Необходимо записать генератор, который бы используя этот список, выдавал 1 000 000 наименований городов по циклу. То есть, дойдя до конца списка, возвращался в начало и повторял перебор. И так, для выдачи миллиона названий. Вывести на экран первые 20 наименований городов с помощью генератора.  **Задание № 2**. Запишите функцию-генератор, которая бы выдавала кортежи длиной n, состоящие из случайных целых чисел в диапазоне [a; b]. Величины n, a, b должны определяться как параметры функции-генератора. Сгенерировать случайные целые величины в диапазоне [a; b] можно функцией randint(a, b) модуля random. Используя эту функцию-генератор, сформируйте с помощью генератора списков двумерную коллекцию размером 20x20 элементов, состоящую из случайных чисел в диапазоне [-5; 5]. Вывести полученную двумерную коллекцию на экран. |
| **Вариант №6**  **Задание №1.** Напишите генератор, который бы возвращал площади кругов с радиусами в диапазоне от 10 до 10 000. Вывести на экран первые пять значений с точностью до сотых.  **Задание № 2**. Задайте функцию-генератор, которая на основе символов строки to\_emails: from string import ascii\_lowercase, ascii\_uppercase to\_emails = ascii\_lowercase + ascii\_uppercase + "0123456789\_" формирует и выдает случайно сгенерированные email-адреса вида: xxxxxxxx@mail.ru где x – случайный символ из строки to\_emails. Количество выдаваемых email- адресов функцией должно быть неограниченным. Случайный выбор символа из последовательности to\_emails можно реализовать с помощью функции choice модуля random. Выведите первые семь сгенерированных адресов на экран. |
| **Вариант №7**  **Задание №1.** Имеется строка, содержащая цифры: d = "0123456789" Необходимо записать генератор, который бы формировал и выдавал все сочетания из трех цифр. Выдавать он должен именно целые числа, а не строки. Вывести первые 50 сгенерированных значений на экран. **Задание № 2.** Определите функцию-генератор, которая бы возвращала значения некоторой функции f(x) для значений x a b ∈[ ; ] с шагом 0,01. Величины a, b, а также функцию f, определить как параметры фукнции-генератора. Вызовите функцию-генератор со значениями a b = − = 20; 100 для функции f x x ( ) = − + 1,5 2, которую определите как лямбда-функцию непосредственно в аргументе функции-генератора. Выведите на экран первые 20 значений, возвращенные функцией-генератором. Какие-либо коллекции в программе не использовать. |
| **Вариант №8**  **Задание №1.** Напишите генератор, который бы возвращал площади прямоугольников с длиной, изменяющейся в диапазоне [200; 10 000], и шириной – в диапазоне [500; 100 000]. Вывести на экран первые 100 значений.  **Задание № 2.** Запишите функцию-генератор, которая бы выдавала списки длиной n, состоящие из случайно выбранных нот из списка p: p = ['до', 'ре', 'ми', 'фа', 'соль', 'ля', 'си'] Величины n, a, b должны определяться как параметры функции-генератора. Выбрать случайное значение из списка p можно функцией choice модуля random. Используя эту функцию-генератор, сформируйте с помощью генератора списков двумерную коллекцию размером 10x10 элементов. Выведите полученный двумерный список на экран. |
| **Вариант №9**  **Задание №1.** Имеется график линейной функции f(x) = 0,5x - 2. Записать генератор, который бы выдавал значения этой функции в диапазоне x∈[0;100] с шагом 0,01 (для x). Вывести первые 50 сгенерированных значений.  **Задание № 2.** Определите функцию-генератор, которая бы перебирала все комбинации из трех целых положительных чисел, каждое в диапазоне [2; 20]. Для каждой тройки чисел она должна определять, могут ли они образовывать длины сторон треугольника. (Критерий: длина любой стороны треугольника должна быть меньше суммы двух других). Функция должна возвращать текущие длины сторон и булево значение True – если числа образуют треугольник и False – в противном случае. Выведите с помощью этой функции-генератора первые 20 значений. Какие-либо коллекции в программе не использовать. |
| **Вариант №10**  **Задание №1.** Имеется список из нот: p = ['до', 'ре', 'ми', 'фа', 'соль', 'ля', 'си'] Необходимо записать генератор, который бы случайным образом выдавал ноты из этого списка 1 000 000 раз. Для случайного выбора нот, используйте функцию choice модуля random. Выведите на экран первые 20 значений этого генератора.  **Задание №2.** Определите функцию-генератор, которая бы возвращала целые числа кратные 3, начиная поиск от значения a, которое задается через параметр функции. Вызовите эту функции со значением аргумента a=-100 и выведите первые 20 чисел, возвращенных этой функцией-генератором. Какие-либо коллекции в программе не использовать. |

Лабораторная работа №5

|  |
| --- |
| **Вариант №1**  **Задание №1.** На вход поступает список из вещественных чисел, записанных в строку через пробел. С помощью функции map преобразовать числа в строке в их вещественное представление и отобразить первые три числа. (Полагается, что минимум три вещественных числа имеются). Реализовать извлечение чисел через функцию next. |
| **Вариант №2**  **Задание №1.** Вводятся названия городов в одну строчку через пробел. Необходимо определить функцию filter, которая бы возвращала только названия длиной более 5 символов. Извлеките первые три значения на выходе этой функции с помощью next и отобразите их на экране (Полагается, что минимум три значения имеются). |
| **Вариант №3**  **Задание №1.** Вводятся два списка целых чисел. Необходимо попарно перебрать их элементы и перемножить между собой. При реализации программы используйте функции zip и map. Выведите на экран первые три значения, используя функцию next. (Полагается, что три выходных значения всегда будут присутствовать). |
| **Вариант №4**  **Задание №1**. Вводятся названия городов в одну строчку через пробел. Необходимо определить функцию map, которая бы возвращала названия городов только длиной более 5 символов. Вместо остальных названий - строку с дефисом ("-"). Сформировать список из полученных значений и отобразить его на экране. |
| **Вариант №5**  **Задание №1.** Вводится строка из слов, записанных через пробел. Необходимо на их основе составить прямоугольную таблицу из трех столбцов и N строк (число строк столько, сколько получится). Лишнее (выходящее) слово - отбросить. Реализовать эту программу с использованием функции zip. Результат отобразить на экране в виде прямоугольной таблицы из слов, записанных через пробел (в каждой строчке). |
| **Вариант №6**  **Задание №1.** Вводится список целых чисел в одну строчку через пробел. Необходимо оставить в нем только двузначные числа. Реализовать программу с использованием функции filter. Результат отобразить на экране в виде последовательности оставшихся чисел. |
| **Вариант №7**  **Задание №1.** На вход функции с именем get\_sort поступает словарь, например, такой: d = {'cat': 'кот', 'horse': 'лошадь', 'tree': 'дерево', 'dog': 'собака', 'book': 'книга'} Необходимо отсортировать словарь d по убыванию ключей (лексикографическая сортировка строк) и возвратить список из соответствующих значений ключей словаря. Например, для указанного словаря d, результатом должен быть список: ['дерево', 'лошадь', 'собака', 'кот', 'книга'] Вызовите функцию get\_sort для заданного в программе произвольного словаря dсt и выведите возвращенный ею список на экран. |
| **Вариант №8**  **Задание №1.** Саша и Галя коллекционируют монетки. Каждый из них решил записать номиналы монеток из своей коллекции. Получилось два списка. Эти списки поступают на вход программы в виде двух строк из целых чисел, (числа записаны через пробел). Необходимо выделить значения, присутствующие в обоих списках и оставить среди них только четные. Результат вывести на экран. При реализации программы используйте функцию filter и кое-что еще (для упрощения программы), подумайте что. |
| **Вариант №9**  **Задание №1.** На вход программы поступает список целых чисел, записанных в одну строчку через пробел. Необходимо выбрать из них четыре наибольших уникальных значения. Результат вывести на экран. |
| **Вариант №10**  **Задание №1.** Вводится список email-адресов в одну строчку через пробел. Среди них нужно оставить только корректно записанные адреса. Будем полагать, что к таким относятся те, что используют латинские буквы, цифры и символ подчеркивания. А также в адресе сначала должен идти символ "@", а затем, точка "." (между ними, конечно же, могут быть и другие символы). Отобразить результат отбора корректных адресов на экране. |