Лабораторная работа 5

(25 баллов)

Для сдачи лабораторной необходимо решить, как минимум по одной задаче из каждого раздела.

map, filter

№1 **(2 балла)**

При помощи функций filter и map выполните задания:

1. Выведите все элементы списка, которые меньше 5.
2. Выведите все элементы списка, разделенные на два.
3. Выведите результат деления на два всех элементов списка, которые больше 17.
4. Напишите программу, которая подсчитает и выведет сумму квадратов всех двузначных чисел, делящихся на 9. При решении задачи используйте комбинацию функций filter, map, sum. Обратите внимание: на 9 должно делиться исходное двузначное число, а не его квадрат.

Генераторы

№2 **(1 балл)**

Напишите функцию генератор factorials(n), генерирующую последовательность факториалов натуральных чисел.

Ввод	Вывод
7	1 2 6 24 120 720 5040

№3 **(1 балл)**

Напишите функцию генератор square_fibonacci(n), генерирующую последовательность квадратов чисел Фибоначчи.

Ввод	Вывод
7	1 1 4 9 25 64 169

№4 **(1 балл)**

Напишите функцию генератор, генерирующую буквы русского алфавита по порядку от а до я. Дополнительные модули использовать нельзя.

№5 **(1 балл)**

Напишите генераторное выражение, генерирующее буквы русского алфавита по порядку от а до я. Дополнительные модули использовать нельзя.

Функция как объект

№6 **(1 балл)**

Напишите функцию `arithmetic_operation(operation)`, которая принимает символ одной из четырех арифметических операций, а возвращает функцию двух аргументов для соответствующей операции.

Пример

Ввод	Вывод
`operation = arithmetic_operation('+')` `print(operation(1, 4))`	`5`

№7 **(1 балл)**

Напишите функцию `same_by(characteristic, objects)`, которая проверяет, все ли объекты имеют одинаковое значение некоторой характеристики, и возвращает **True**, если это так. Если значение характеристики для разных объектов отличается – то **False**. Для пустого набора объектов, функция должна возвращать True. Аргумент characteristic – это функция, которая принимает объект и вычисляет его характеристику.

Пример 1

Ввод	Вывод

`values = [0, 2, 10, 6]`
`if same_by(lambda x: x % 2, values):`
`  print('same')`
`else:`
`  print('different')`
	`same`

Пример 2

Ввод	Вывод

`values = [1, 2, 3, 4]`
`if same_by(lambda x: x % 2, values):`
`  print('same')`
`else:`
`  print('different')`
	`different`

№8 **(1 балл)**

Напишите функцию *print_operation_table(operation, num_rows=9, num_columns=9)*, которая принимает в качестве аргумента функцию, реализующую бинарную операцию вычисления элемента по номеру строки и столбца. Аргументы num_rows и num_columns указывают число строк и столбцов таблицы, которые должны быть распечатаны. Нумерация строк и столбцов идёт с единицы.

Примечание: бинарной операцией называется любая операция, у которой ровно два аргумента, как, например, у операции умножения.

Ввод
`print_operation_table(lambda x, y: x * y)`

Вывод								
1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Ввод
`print_operation_table(lambda x, y: x * y, 5)`

Вывод								
1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45

№9 **(1 балл)**

Напишите функцию ask_password(login, password, success, failure), которая получает логин и пароль пользователя и проверяет их правильность. Пароль считается правильным, если в нём содержится ровно три английские гласные буквы (гласными считать буквы a,e,i,o,u,y) и ровно такой же набор согласных (все буквы, кроме перечисленных шести), как в логине. Порядок и количество согласных также должно совпадать. Логин и пароль приводятся к нижнему регистру и передаются.

Пример: для логина “login” подойдут пароли “aaalgn” и “luagon“.

success и failure – коллбэки. Коллбэком называется специальная функция, которая вызывается, когда ваше вычисление завершилось. Для программы, запустившей запрос или долгое вычисление, коллбэк – это способ сообщить, что надлежит сделать, когда вычисление завершится. Коллбэк обычно передают как аргумент функции запуска вычисления.

Если пароль правильный, функция должна вызвать коллбэк success, передав ему в качестве аргумента логин. А если пароль был неверный - failure, передав ему в качестве аргументов логин и сообщение об ошибке (в таком порядке). Сообщение об ошибке должно быть одним из трёх вариантов:

- “Wrong number of vowels”, если в пароле неверное число гласных;
- “Wrong consonants”, если в пароле набор согласных отличается от набора согласных логина;
- “Everything is wrong”, если оба условия нарушены.

Также напишите функцию main(login, password), которая вызывает написанную функцию ask_password так, чтобы в случае успеха она печатала “Привет, {логин}!”, а в случае ошибки – “Кто-то пытался притвориться пользователем {логин}, но в пароле допустил ошибку: {текст ошибки, большими буквами}.”.

Пример 1

Ввод	Вывод
`main("anastasia", "nsyatos")`	`Привет, anastasia!`

Пример 2

Ввод	Вывод
`main("eugene", "aanig")`	`Кто-то пытался притвориться пользователем eugene, но в пароле допустил ошибку: WRONG CONSONANTS.`

Пример 3

Ввод	Вывод
`ask_password("anastasia", "nsyatos", lambda login: print('super'), lambda login, err: print('bad'))`	`super`

Стандартные функции для работы с итераторами и коллекциями

№10 **(1 балл)**

С клавиатуры вводится строка из разделённых пробелами слов. Выведите на экран в строку, разделённую пробелами, список слов, отсортированных в лексикографическом порядке.
Введённые слова могут быть написаны в различных регистрах. Сортироваться слова должны независимо от регистра, а выводиться на печать в том виде, в котором переданы на вход программы.

Напомним, что функция sorted сортирует элементы в лексикографическом порядке, но при этом по-умолчанию любая буква в верхнем регистре считается идущей раньше, чем буква в нижнем регистре (вам такой вариант не подходит).

Пример

Ввод	Вывод
cats dog CAT DOGGY monkey	CAT cats dog DOGGY monkey

№11 **(1 балл)**

Отсортировать последовательность целых чисел по убыванию модуля числа

Input	Output
3 6 -8 2 -78 1 23 -45 9	-78 -45 23 9 -8 6 3 2 1

№12 **(1 балл)**

Даны координаты точек. Отсортировать их по удалению от точки (0,0).

Если две точки находятся на одном расстоянии от начала координат, то сначала выведите точку с меньшим значением координаты x. Если и совпали и расстояния, и значения x-координаты, сначала выведите точку с меньшим значением y-координаты.

№13 **(1 балл)**

Напишите программу, которая ищет нули в таблице чисел и печатает $True$, если нули нашлись.
В противном случае надо напечатать $False$.

Эту задачу надо постараться решить «в одну строчку». В этом вам помогут функции any и all.

Пример

Ввод	Вывод
64 33 79 56 78 70 45 71 82 3 96 27 8 36 72 14 91 10 21 65 95 28 91 23 78 38 21 50 64 37 97 54 94 6 48 17 37 19 78 58 69 58 35 1 70 24 60 17 3 11 48 9 13 23 82 49 79 55 29 53 9 2 67 90 0 17 34 55 49 63 98 98 23 71 66 57 15 94 34 81 58 37 32 29 10 19 53 46 95 19 41 24 95 47 58 17 74 69 62 4	True

№14 **(2 балла)**

Напишите программу для обработки текста. На вход вашей программы подаётся многострочный текст, причём число строк заранее неизвестно.

Ваша задача – пронумеровать слова в нём, начиная с нуля, а затем вывести только те слова, которые начинаются с большой буквы. Перед словом необходимо вывести номер первого вхождения этого слова в текст.

Слова необходимо отсортировать в лексикографическом порядке.

Для решения этой задачи вам помогут итераторы sys.stdin и enumerate.

Пример

Ввод	Вывод
Ехал Грека через реку. Видит Грека в реке рак. Сунул в реку руку Грека. Рак за руку Греку цап.	4 - Видит 1 - Грека 17 - Греку 0 - Ехал 14 - Рак 9 - Сунул

№15 **(1 балл)**

Вычислите и выведите на экран минимальное значение (лексикографически) в итераторе, используя functools.reduce. В качестве итератора возьмите список строк из стандартного ввода.

Не забудьте обрезать символ перевода в конце строки. Обрезать вместе с символом новой строки пробелы по концам также допустимо: гарантируется, что во входных данных их не будет.

При решении этой задачи вам понадобится так называемый тернарный оператор (т.е. оператор с тремя аргументами):

<value_if_true> if <condition> else <value_if_false>

Значение тернарного оператора будет value_if_true, если условие condition верно, иначе будет равно value_if_false. Это позволяет выполнять вычисления, зависящие от условия, одной командой.

Например:

print("чётное" if (x % 2 == 0) else "нечётное")

напечатает, чётное ли число x.

Пример:

Ввод	Вывод
котик тюлень кашалот нарвал	кашалот

№16 **(1 балл)**

В модуле math есть функция math.gcd(a, b), возвращающая наибольший общий делитель (НОД) двух чисел. При помощи functools.reduce вычислите и напечатайте наибольший общий делитель для списка натуральных чисел, поступающих в стандартный поток ввода.

Ввод	Вывод
36 12 144 18	6

Декораторы

№17 **(1 балл)**

Напишите декоратор **check_password**, который запрашивает пароль, прежде чем вызвать функцию, и если он неверный — возвращает None и печатает «В доступе отказано».

№18 **(3 балла)**

Напишите генератор декораторов **check_password**, т. е. функцию, которая возвращает декоратор.
Генератор декораторов принимает в качестве параметра пароль, и получившийся декоратор должен закрыть функцию этим паролем.

Декоратор будет применяться следующим образом:

```
@check_password('password')
make_burger(typeOfMeat, withOnion=False, withTomato=True):
    # ...
```

Т.е при определении функции сначала вызывается функция **check_password()** с аргументом "password", получается декоратор, затем уже этот получившийся декоратор применяется к функции.

№19 **(3 балла)**

Напишите декоратор **cached**, который будет кэшировать результат вызова функции.
Пример того, как можно будет использовать ваш декоратор:

```
@cached
def fib(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fib(n - 1) + fib(n - 2)
```