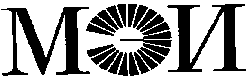
Национальный исследовательский университет "МЭИ"



mpei_1

Кафедра РЗиАЭ

Лабораторная работа №1

«Основы Python»

|  |  |
| --- | --- |
| Выполнила: | Попов Д.Г. |
| Вариант: | 10 |
| Группа: | Э-12-20 |
| Преподаватель: | Данилов С.А. |

Москва

2024

**Задание на лабораторную работу**

**Вариант 10**

1. Создать множество и заполнить его элементами от 1 до 10. Вывести результат.

2. Создать кортеж и заполнить его элементами списка. Список очистить. Вывести результат.

3. Создать словарь и заполнить его следующим образом: ключ – элемент множества, значение – элемент списка. Вывести результат.

4. Из произвольного списка чисел посчитать количество четных и нечетных чисел.

5. Написать функцию, на вход которой подается словарь, содержащий ключ в виде целого числа и значение в виде строки. Найти значение с максимальным и минимальным ключом. Найти среднее значение между полученными числами и удалить все пары ключ-значение значение которых меньше полученного числа. Итоговый словарь вернуть и вывести на экран. При выполнении задания использовать функцию filter().

6. Написать функцию, на вход которой подается два списка в одном значение длины, а другом ширины. Найти площади этих прямоугольников и записать в множество. Найти медиану площадей и удалить все значения из списка меньше неё. Из полученных значений найти сумму и поделить её на длину входного списка и вывести ответ. Итоговый ответ вернуть и вывести на экран. При выполнении задания использовать функцию filter().

7. Написать функцию, которая генерирует пароль заданной пользователем длины и с заданным количество прописных букв и специальных символов.

**Выполнение лабораторной работы**

1. Создать множество и заполнить его элементами от 1 до 10. Вывести результат.

print('Task 1')

my\_set = set(range(1, 11)) # Создание множества

print('Created set: ' + str(my\_set))

Result:



2. Создать кортеж и заполнить его элементами списка. Список очистить. Вывести результат.

print('Task 2')

my\_list = [i \*\* 2 for i in range(0, 11)]  # Создание списка для последующего заполнения кортежа

print('My list: ' + str(my\_list))

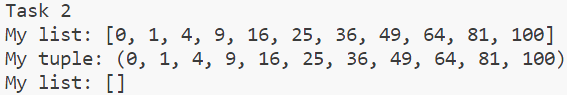
my\_tuple = tuple([j for j in my\_list])  # Заполнение

my\_list.clear()  # Очистка списка

print('My tuple: ' + str(my\_tuple))

print('My list: ' + str(my\_list))

Result:



3. Создать словарь и заполнить его следующим образом: ключ – элемент множества, значение – элемент списка. Вывести результат.

print('Task 3')

my\_list = [i + 5 for i in range(1, 11)]  # Заполняем новый список (генераторным методом)

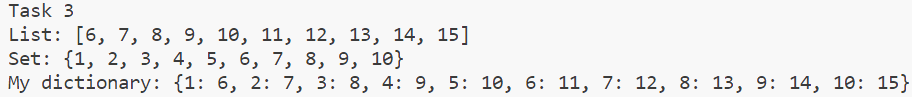
print('List: ' + str (my\_list))

print ('Set: ' + str (my\_set))

my\_dict = dict(zip(my\_set, my\_list))

print('My dictionary: ' + str (my\_dict))

Result:



4. Из произвольного списка чисел посчитать количество четных и нечетных чисел.

print('Task 4')

new\_list = list()

for i in range(0, 10):

    new\_list.append(randint(-10, 100))  # Заполняем список случайными числами

print('New list: ' + str (new\_list))

even\_num = 0

uneven\_num = 0  # В эти переменные записываем числа чётных и нечётных элементов

for j in new\_list:  # Считаем количество чётных и нечётных чисел

    if j != 0 and j % 2 == 0:

        even\_num += 1

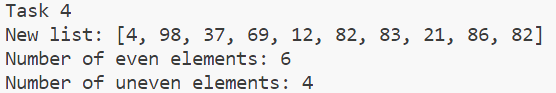
    elif j != 0:

        uneven\_num += 1

print('Number of even elements: ' + str(even\_num))

print('Number of uneven elements: ' + str (uneven\_num))

Result:



5. Написать функцию, на вход которой подается словарь, содержащий ключ в виде целого числа и значение в виде строки. Найти значение с максимальным и минимальным ключом. Найти среднее значение между полученными числами и удалить все пары ключ-значение значение которых меньше полученного числа. Итоговый словарь вернуть и вывести на экран. При выполнении задания использовать функцию filter().

print('Task 5')

def task5(task5\_dict: dict):  # Функция ожидает на вход тип данных - словарь.

    for key in task5\_dict.keys():  # Проверка словаря на соответствие условиям

        if not isinstance(key, int):

            return 'Ключи в словаре должны быть int'

    for elem in task5\_dict.values():

        if not isinstance(elem, str):

            return 'Элементы в словаре должны быть str'

    print('Введённый словарь ' + str (task5\_dict))

    keys = task5\_dict.keys()  #получаем коллекцию ключей из словаря

    max\_key = max(keys)

    min\_key = min(keys)  # Ищем наибольшее и наименьшее значения

    print('Наименьший ключ в словаре: ' + str (min\_key) + '\n' + 'Наибольший ключ в словаре: ' + str (max\_key))

    mid = (max\_key + min\_key) / 2  # Ищем среднее

    print('Среднее: ' + str (mid))

    elem\_to\_del = list(filter(lambda a: a if a >= mid else None, task5\_dict.keys()))  # Фильтрация словаря

    for elem in elem\_to\_del:

        del task5\_dict[elem]

    print('Полученный словарь: ' + str (task5\_dict))

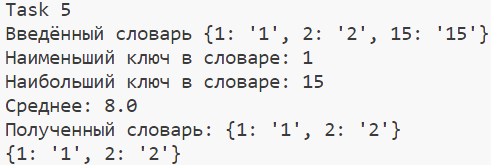
    return task5\_dict

my\_dict = {1: '1', 2: '2', 15: '15'} # Исходный словарь, подаваемый на вход

my\_new\_dict = task5(my\_dict)  # Выполняем функцию, получаем новый словарь

print(my\_new\_dict, '\n')

Result:



6. Написать функцию, на вход которой подается два списка в одном значение длины, а другом ширины. Найти площади этих прямоугольников и записать в множество. Найти медиану площадей и удалить все значения из списка меньше неё. Из полученных значений найти сумму и поделить её на длину входного списка и вывести ответ. Итоговый ответ вернуть и вывести на экран. При выполнении задания использовать функцию filter().

print('Task 6')

def task6(width: list, length: list):

    set\_of\_square = set()  # Считаем площади

    for i in range(len(length)):

        set\_of\_square.add(width[i] \* length[i])

    med = median(set\_of\_square)  # Медиана

    print('Множество посчитанных площадей: ' + str (set\_of\_square) + '\n' + 'Медиана: ' + str (med))

    filtered\_squares = list(filter(lambda a: a if a >= med else None, set\_of\_square))

    print('Отфильтрованный список: ' + str (filtered\_squares))

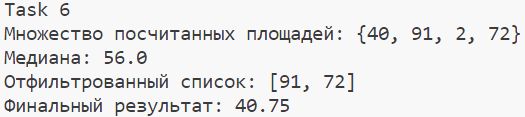
    final = sum(filtered\_squares) / len(width)

    print('Финальный результат: ' + str (final) + '\n')

    return final

z = task6([2, 4, 6, 7], [1, 10, 12, 13])

Result:



7. Написать функцию, которая генерирует пароль заданной пользователем длины и с заданным количество прописных букв и специальных символов.

print('Task 7')

def task7(length: int, uppercase\_num: int, special\_symbols\_num: int):

    if length < uppercase\_num + special\_symbols\_num:

        return 'Такой пароль невозможно сгенерировать'

    specials = []  # Специальные символы

    upper = []  # прописные буквы

    another = []  # Другие символы (строчные + цифры)

    for i in range(uppercase\_num):

        upper.append(choice(ascii\_uppercase))  # Выбираем нужное число спец символов (далее аналогично)

    for i in range(special\_symbols\_num):

        specials.append(choice(punctuation))

    for i in range(length - special\_symbols\_num - uppercase\_num):

        another.append(choice(ascii\_lowercase + digits))

    elem\_of\_password = specials+upper+another  # Список со всеми символами

    shuffle(elem\_of\_password)  # перемешиваем, чтобы не было последовательностей спец символов и строчных букв

    password = ''

    for j in elem\_of\_password:

        password += j  # записываем пароль в строку

    return password

a = int(input('Введите требуемое количество символов в пароле: '))

b = int(input('Введите количество прописных (заглавных) букв: '))

c = int (input('Введите количество специальных символов: '))

print(task7(a, b, c))

Result:

