НИУ «МЭИ»

Кафедра «Релейной защиты и автоматизации энергосистем»

Вычислительные комплексы в электроэнергетике

ЛАБОРАТОРНАЯ РАБОТА №1

«*Основы языка Python»*

|  |  |
| --- | --- |
| Группа: | Э-12-20 |
| Вариант: | 8 |
| ФИО студента: | Кетова Т.В. |
| ФИО преподавателя: | Данилов С.А. |

Москва, 2024

1. Создать список заполнив его значениями от 45 до 7 с шагом 3. Вывести

результат.

Код программы:

my\_list = list(range(45,7,-3))

print ('Список', my\_list)

Результат:

Список [45, 42, 39, 36, 33, 30, 27, 24, 21, 18, 15, 12, 9]

1. На основе списка создать множество значений от 0 до 10, использовать при этом цикл break. Вывести результат.

Код программы:

b = set() # создаем пустое множество

for i in my\_list[::-1]: # все элементы списка в обратном порядке

if i < 11: # задаем условие

b.add(i)

else:

break

print(b)

Результат:

Множество {9}

1. Создать список и добавить в него значения словаря. Вывести результат. Отсортировать список. Вывести результат.

Код программы:

my\_dict = {'a': 1, 'b': 2, 'c': 3}

my\_new\_list = list(my\_dict.values())

print('Новый список',my\_new\_list)

my\_new\_list.sort()

print('Отсортированный список',my\_new\_list)

Результат:

Новый список [1, 2, 3]

Отсортированный список [1, 2, 3]

1. Среди двух произвольных списков найти есть ли хотя бы один общий элемент.

Код программы:

first\_list = [1,12,15,16,17,19,123]

print ('Первый список', first\_list)

second\_list = [15, 13,33,14,19]

print ('Второй список', second\_list)

itog\_list = []

for i in first\_list : #задаем условие для нового листа

if i in second\_list :

itog\_list.append(i)

print ('Общие элементы', itog\_list)

Результат:

Первый список [1, 12, 15, 16, 17, 19, 123]

Второй список [15, 13, 33, 14, 19]

Общие элементы [15, 19]

1. 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().

Код программы:

dlina = [4,15,17,19]

shir = [2,3,14,15]

def calculate\_plosh(input\_dlina, input\_shir):

plosh = {length \* width for length, width in zip(input\_dlina, input\_shir)} # Записываем площади прямоугольников в множество

median = sorted(plosh)[len(plosh)//2]

filtered\_areas = filter(lambda x: x >= median, plosh)

sum\_filtered\_areas = sum(filtered\_areas)

result = sum\_filtered\_areas / len(input\_dlina) # Делим сумму на длину входного списка

return result

final\_answer = calculate\_plosh(dlina, shir)

print("Итоговый ответ:", final\_answer)

Результат:

Итоговый ответ: 130.75

1. 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().

Код программы:

my\_dict = {1: 'one', 3: 'three', 11: 'eleven', 20: 'twenty'}

def dictionary(input\_dict):

max\_value = max(input\_dict.keys())

min\_value = min(input\_dict.keys())

middle = (max\_value + min\_value) / 2

filtered\_dict = dict(filter(lambda x: int(x[0]) >= middle, input\_dict.items()))

print(filtered\_dict)

return filtered\_dict

final\_dict = dictionary(my\_dict)

Результат:

{11: 'eleven', 20: 'twenty'}

1. Написать функцию перевода секунд в дни, часы и минуты. На вход подается число секунд от начала точки отсчета, на выходе дата в формате d/h/m/s.

Код программы:

def convert\_seconds(seconds):

days = seconds // (24 \* 3600)

seconds %= 24 \* 3600 # остаток от деления

hours = seconds // 3600

seconds %= 3600

minutes = seconds // 60

seconds %= 60

result = "{:d}d {:02d}h {:02d}m {:02d}s".format(days, hours, minutes, seconds)

return result

total\_seconds = 123456789

time\_string = convert\_seconds(total\_seconds)

print(time\_string)

Результат:

Результат 1428d 21h 33m 09s

Вывод: в результате работы с помощью языка Python было решено 7 задач. В результате каждого решения были получены ожидаемые результаты Пояснения к работе кода представлены в виде построчного комментирования.