**Задание 1.** Напишите функцию arithmetic\_operation(operation), которая принимает символ одной из четырех арифметических операций, а возвращает функцию двух аргументов для соответствующей операции.

Пример проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| operation = arithmetic\_operation('+')  print(operation(1, 4)) | 5 |

**Задание 2.** Сконструируйте свою функцию simple\_map(transformation, values), которая из одного списка делает другой. Каждый элемент результирующего списка должен получаться применением функции преобразования transformation к соответствующему элементу списка values.

Использовать стандартную функцию map не получится.

Пример проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| values = [1, 3, 1, 5, 7]  operation = lambda x: x + 5  print(\*simple\_map(operation, values)) | 6 8 6 10 12 |

**Задание 3.** В двух глобальных переменных one, two находятся два натуральных числа: первое слагаемое и второе слагаемое. Напишите функцию roman(), которая считает их сумму и записывает ее в глобальную переменную three, переводит каждое слагаемое и сумму в римскую запись и печатает их, как показано в примере.

Формат ввода

one = 5

two = 4

roman()

Формат вывода

V + IV = IX

Полезным будет использовать словарь для перевода цифр

RomanNumbers = {'M': 1000, 'CM': 900, 'D': 500,

'CD': 400, 'C': 100, 'XC': 90,

'L': 50, 'XL': 40, 'X': 10,

'IX': 9, 'V': 5, 'IV': 4, 'I': 1}

**Задание 4.** У некоторого банка есть короткий номер, на который клиенты отправляют смс сообщения для совершения транзакций. Каждая транзакция имеет вид: 111111111-перевод:1000, где набор цифр до символа "-" является номером телефона клиента, далее следует тип транзакции, а после двоеточия следует сумма в копейках для ее совершения. Для сбора статистики по клиентам требуется написать функцию get\_transactions(t), которая подсчитывает, сколько транзакций какого вида было сделано и сколько денег было потрачено.

Если вместо транзакции передана фраза "print\_it", то функция печатает информацию обо всех типах поступивших транзакций в том порядке, в котором впервые встретились их наименования. Информация о каждой транзакции выводится на отдельной строке и состоит из количества транзакций данного типа, наименования типа транзакции и количества потраченных денег.

Вывод заключается в следующем В каждой строке через пробел печатается информация обо всех типах поступивших транзакций: количество транзакций данного типа, наименование транзакции данного типа, количество потраченных денег на данную транзакцию.

Пример проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| get\_transactions('880005553535-перевод:100')  get\_transactions('111111111-перевод:1000')  get\_transactions('880005553535-оплата\_жкх:10000')  get\_transactions('89065664312-перевод:50000000')  get\_transactions('print\_it') | 3 перевод 50001100  1 оплата\_жкх 10000 |

**Задание 5.** 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 проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| values = [0, 2, 10, 6]  if same\_by(lambda x: x % 2, values):  print('same')  else:  print('different') | same |