Задание 1. Напишите программу, которая берет из параметров командной строки два целочисленных аргумента и выводит на экран их сумму.  
В случае любой ошибки (некорректное количество аргументов, аргументы не того типа и т.д.) следует вывести число **0**.

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py 3 5 | 8 |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py 3 | 0 |

**Пример 3**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py vasya 5 | 0 |

**Примечания**

Необходимо обязательно использовать sys.argv

Задание 2. Модифицируем наш целочисленный калькулятор-сумматор.  
Теперь он получает не два, а **произвольное**количество параметров. При этом параметры могут быть ошибочными, как и ранее.  
  
Вам необходимо посчитать сумму всех **целых**параметров, причем с чередующимися знаками. Это означает, что первое встретившееся целое число будет учтено в сумме с коэффициентом **1**, второе — с **-1**, третье — опять с **1**и т.д.

Если параметры не было переданы, то следует вывести фразу **NO PARAMS**, а в случае других ошибок — имя класса исключения, которое Python выбросит в этом случае.

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py 3 5 | -2 |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py 3 | 3 |

**Пример 3**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py | NO PARAMS |

**Примечания**

Для решения задачи необходимо **обязательно**использовать sys.argv  
Имейте в виду, что **05**— это тоже целое число.

Задание 3. Модифицируйте сумматор из первого задания. Теперь программа должна принимать последовательно два вещественных числа и знак арифметической операции + - \* / и ^. Вывод – результат операции. В случае неверных параметров – сообщать, какая ошибка произошла. Необходимо обязательно использовать sys.argv.

Задание 4. Используя sys.argv напишите программу, которая разбирает аргументы командной строки вида "ключ=значение" и выводит их на отдельной строке в виде "Key: ключ Value: значение".  
Кроме описанных аргументов программа может принимать необязательную опцию --sort, которая позволяет отсортировать выводимые значения по ключу.

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py --sort name=Vasya age=42 | Key: age Value: 42  Key: name Value: Vasya |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py --sort surname=Ivanov name=Vasya age=42 | Key: age Value: 42  Key: name Value: Vasya  Key: surname Value: Ivanov |

**Примечания**

Ваша программа может иметь произвольное имя.  
Проверка валидности входных параметров не требуется.  
Необходимо обязательно использовать sys.argv.

Задание 5. Напишите программу, которая в качестве аргумента принимает имя файла (не указан файл или указан несуществующий — ошибка) и выводит его содержимое на экран.  
В добавок, программа может принимать дополнительные агрументы:

* «--count» для вывода кол-ва строк в конце сообщения,
* «--num» для вывода порядкового номера с пробелом в начале каждой строки,
* «--sort» для сортировки строк в алфавитном порядке перед выводом.

Пусть файл **text1.txt**содержит строки:

Houston  
we have  
a problem

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py --num text1.txt | 0 Houston  1 we have  2 a problem |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py --count --sort text1.txt | Houston  a problem  we have  rows count: 3 |

**Пример 3**

| Ввод | Вывод |
| --- | --- |
| python3 solution.py --count --sort textX.txt | ERROR |

**Примечания**

Необходимо использовать sys.argv  
При наличии ошибки необходимо вывести слово **ERROR**