

## Лабораторная работа 2

(20 баллов)

### №1 (2 балла)

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.

Ваша программа может иметь произвольное имя.

Проверка валидности входных параметров не требуется.

Необходимо обязательно использовать `sys.argv`.

#### Пример 1

Ввод

```
python3 solution.py --sort name=Vasya age=42
```

Вывод

```
Key: age Value: 42
Key: name Value: Vasya
```

#### Пример 2

Ввод

```
python3 solution.py --sort surname=Ivanov name=Vasya age=42
```

Вывод

```
Key: age Value: 42
Key: name Value: Vasya
Key: surname Value: Ivanov
```

### №2 (2 балла)

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 аргументы:

- «`--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.

Необходимо использовать обязательно использовать `sys.argv`.

При наличии ошибки необходимо вывести слово `ERROR`

#### Пример 1

Ввод

```
python3 solution.py --num text1.txt
```

Вывод

```
0 Houston
1 we have
2 a problem
```

### Пример 2

Ввод

```
python3 solution.py --count --sort text1.txt
```

Вывод

```
Houston  
a problem  
we have  
rows count: 3
```

### Пример 3

Ввод

```
python3 solution.py --count --sort textX.txt
```

Вывод

```
ERROR
```

### №3 (2 балла)

Напишите скрипт, который принимает любое кол-во параметров через произвольное количество пробелов в качестве аргументов и выводит их в той же последовательности на экран, каждый на отдельной строке. В случае отсутствия параметров скрипт должен выводить «no args».

Необходимо использовать библиотеку argparse, без указания description.

В этой задаче в параметрах командной строки не могут быть переданы именованные параметры, например, не возможна ситуация: --param value.

### Пример 1

Ввод

```
python3 solution.py 5 string 1
```

Вывод

```
5  
string  
1
```

### Пример 2

Ввод

```
python3 solution.py
```

Вывод

```
no args
```

### №4 (2 балла)

Напишите программу, которая умеет копировать текстовый файл.

В качестве аргументов она принимает имя файла-источника и имя файла-приемника.

Кроме того, в аргументах может быть указаны опция «--upper» для приведения всех символов файла к верхнему регистру, а также опция «--lines N» для копирования только первых N строк. Под строкой понимается последовательность символов, которая заканчивается символом перевода строки — \n.

N может быть числом, превышающим количество строк в файле. Необходимо корректно обработать эту ситуацию. Если целевой файл не существует, то его нужно создать, если существует – перезаписать.

В программе необходимо обязательно использовать библиотеку argparse.

### №5 (2 балла)

Напишите программу калькулятор, которая принимает два целочисленных параметра и считает их сумму. При этом параметры могут быть ошибочными.

Если параметры не были переданы, то следует вывести фразу NO PARAMS, если целочисленный параметр только один, то выведите фразу TOO FEW PARAMS, если же целочисленных параметров больше двух — TOO MANY PARAMS, а в случае других ошибок — имя класса исключения, которое Python выбросит в этом случае.

В программе необходимо обязательно использовать библиотеку argparse.

#### Пример 1

Ввод  
`python3 solution.py 3 5`

Вывод  
8

#### Пример 2

Ввод  
`python3 solution.py 3`

Вывод  
TOO FEW PARAMS

### №6 (1 балл)

Реализуйте задачу №1, используя библиотеку argparse.

### №7 (1 балл)

Реализуйте задачу №2, используя библиотеку argparse.

### №8 (4 балла)

Представьте, что вашей компании требуется сформировать отчет P&L для расчета итоговых прибылей и убытков (например, расчет прибыли за год, если известен доход за день).

Вы решили написать программу, которая принимает на вход именованные аргументы с известными доходами/расходами (вещественные числа) в день/неделю/месяц/год (--per-day, --per-week, --per-month, --per-year) и период времени за который требуется рассчитать итоговый результат. Это может быть день, месяц или год (--get-by [day, month, year]). Других значений параметр --get-by принимать не может.

Программа должна выводить вычисленную сумму за указанный период расчета, при этом, если он явно не указан, то надо рассчитывать P&L за день.

При выводе суммы ее необходимо привести к целому числу с помощью int().

При расчете можно полагаться на следующие аксиомы: 1 год = 360 дней, 1 месяц = 30 дней, 1 неделя = 7 дней.

#### Пример 1

Ввод  
`python3 solution.py --per-day +1 --per-week -7 --get-by 0  
month`

Вывод  
0

## Пример 2

Ввод

```
python3 solution.py --per-year +360 --per-day -2
```

Вывод

```
-1
```

## Пример 3

Ввод

```
python3 solution.py --per-day 1 --per-week -14 --per-year 360 --per-month 7 --get-by what
```

Вывод

```
usage: solution.py [-h] [--per-day PER_DAY] [--per-week PER_WEEK]
                  [--per-month PER_MONTH] [--per-year PER_YEAR]
                  [--get-by {day,month,year}]
solution.py: error: argument --get-by: invalid choice: 'what' (choose fr
```

## №9 (4 балла)

Напишите программу, которая форматирует текстовый файл и выводит его содержимое в указанных рамках.

Программа принимает на вход высоту, ширину (`--frame-height`, `--frame-width`) блока в символах и имя файла, а на выходе выводит содержимое (начальную часть) файла которое помещается в указанные размеры. Если строка не умещается по длине, то ее нужно перенести.

Вся логика форматирования должна быть реализована в функции `format_text_block(frame_height, frame_width, file_name)`, которая возвращает отформатированный текст и в дальнейшем может быть использована в других скриптах.

В случае возникновения ошибок при работе с файлом необходимо вернуть текст исключения, которое выбрасывается в этой ситуации.

В программе необходимо обязательно использовать библиотеку `argparse`.

Файл для тестирования находится по адресу <https://cloud.mail.ru/public/kcos/WhjV4C7Cs>.

## Пример 1

Ввод

```
python3 solution.py --frame-height 10 --frame-width 30 filename.txt
```

Вывод

```
Chapter I: An Unexpected Party
```

```
In a hole in the ground there
lived a hobbit. Not a nasty, d
irty, wet hole, filled with th
e ends of worms and an oozy sm
ell, nor yet a dry, bare, sand
```

## Пример 2

Ввод

```
python3 solution.py --frame-height 10 --frame-width 30 file.txt
```

Вывод

```
[Errno 2] No such file or directory: 'file.txt'
```