

## #4: СИСТЕМЫ СЧИСЛЕНИЯ

Сложность: 1 из 10

Написать программу **digit.py**, которая получает из первого аргумента командной строки целое число, а после печатает его в разных системах счисления.

Результаты должны быть разделены между собой пробелами и идти в следующем порядке: десятичное число, двоичное число, восьмиричное число, шестнадцатиричное число.

**Пример использования**

```
$ python digit.py 10
$ 10 1010 12 a
```

## #1: ВСЁ БОЛЬШИМИ

Сложность: 1 из 10

Написать программу **upper.py**, которая получает из первого аргумента командной строки слово, а после печатает это слово большими буквами.

**Пример использования**

```
$ python upper.py word
$ WORD
```

## #8: БОЛЬШИЕ НА МАЛЕНЬКИЕ И НАОБОРОТ

Сложность: 1 из 10

Написать программу **swapcase.py**, которая получает из первого аргумента командной строки слово, а после заменяет в нем заглавные буквы на строчные и наоборот.

**Пример использования**

```
$ python swapcase.py Python
$ pYTHON
```

## #2: КИНЕТИЧЕСКАЯ ЭНЕРГИЯ

Сложность: 1 из 10

Написать программу **energy.py**, которая получает из первого аргумента командной строки массу тела, а из второго — скорость, а после печатает кинетическую энергию этого тела.

**Пример использования**

```
$ python energy.py 10 10
500.0
```

## #9: ПОВТОРЕНИЕ СТРОКИ

Сложность: 2 из 10

Написать программу **repeat.py**, которая получает из первого аргумента командной строки слово, а из второго число N, а после повторяет полученное слово N раз.

**Пример использования**

```
$ python repeat.py AA 5
$ AAAAAAAAAA
```

## #15: САМОЕ БОЛЬШОЕ ЧИСЛО

Сложность: 2 из 10

Написать программу **max.py**, которая получает пять чисел из аргументов командной строки, а затем выводит самое большое из них.

**Пример использования**

```
$ python max.py 2 42 3 56 4
$ 56
```

## #35: МУЖЧИНЫ И ЖЕНЩИНЫ

Сложность: 2 из 10

В очереди за билетами в кино стоят мужчины и женщины. Нужно посчитать количество мужчин и количество женщин.

Написать программу **mw.py**, которая получает из первого аргумента командной строки набор символов **m** (мужчины) и **w** (женщины). Затем программа должна посчитать количество мужчин и женщин и вывести результаты на экран в виде импровизированной диаграммы (см. пример использования).

**Пример использования**

```
$ python mw.py mmwmwmmwwmmw
$ m (6) ******
$ w (7) ******
```

## #14: ФАМИЛИЯ ИМЯ ОТЧЕСТВО

Сложность: 2 из 10

Написать программу **fio.py**, которая принимает три аргумента командной строки: фамилию, имя и отчество. А затем выводит их в правильной форме: первая буква заглавная, остальные строчные.

**Пример использования**

```
$ python fio.py иванов иван иванович
$ Ивано Иван Иванович
```

## #7: В ОБРАТНОМ ПОРЯДКЕ

Сложность: 2 из 10

Написать программу **reverse.py**, которая получает из первого аргумента командной строки слово, а после печатает его задом наперед.

**Пример использования**

```
$ python reverse.py Spartak
$ katrapS
```

## #11: КОЛИЧЕСТВО ПОВТОРЕНИЙ

Сложность: 2 из 10

Написать программу **count.py**, которая получает из первого аргумента командной строки слово, а из второго букву и выводит сколько раз эта буква встречается в переданном слове.

**Пример использования**

```
$ python count.py programming r
$ 2
```

## #12: ОТСОРТИРОВАТЬ ПО АЛФАВИТУ

Сложность: 2 из 10

Написать программу **sort.py**, которая получает из первого аргумента командной строки слово, а затем выводит все буквы слова в алфавитном порядке.

**Пример использования**

```
$ python sort.py programming
$ aggimnopr
```

## #13: ОТСОРТИРОВАТЬ ПО АЛФАВИТУ - 2

Сложность: 3 из 10

Написать программу **sort.py**, которая получает из первого аргумента командной строки слово, а затем выводит все буквы слова в алфавитном порядке исключая повторные буквы.

**Пример использования**

```
$ python sort.py programming
$ agimnopr
```

## #3: РАЗБИЕНИЕ ЧИСЛА НА РАЗРЯДЫ

Сложность: 3 из 10

Написать программу **th.py**, которая получает из первого аргумента командной строки число, после разбивает это число пробелами на разряды и печатает его.

**Пример использования**

```
$ python th.py 12345678
$ 12 345 678
```

## #52: ДОРОГУ СТАРШИМ

Сложность: 3 из 10

В очереди на кассе стоит несколько человек разного возраста. Кассир решила обслужить старшего покупателя первым.

Написать программу **queue.py**, которая читает числа (возраст покупателей) из файла **queue.txt**, ставит самое большое число на первом место и выводит новую очередь на экран.

Файл **queue.txt** находится в папке с программой и содержит одну строку с числами, разделенными пробелами. Например: **34 12 53 14**

### Пример использования

```
$ python queue.py
$ 53 34 12 14
```

## #16: СПИСОК ПОКУПОК

Сложность: 4 из 10

Вы отправились в магазин и попросили друзей составить список покупок и прислать его вам. Каждый из друзей написал свой список и в итоге вам прислали три файла: shopping\_list\_1.txt, shopping\_list\_2.txt и shopping\_list\_3.txt. Когда вы открыли списки покупок, то сразу заметили, что некоторые товары повторяются, поэтому вы решили составить свой собственный нормализованный список, где продукты не повторяются и записаны по алфавиту.

Напишите программу **shopping\_list.py**, которая читает строки из трех файлов: shopping\_list\_1.txt, shopping\_list\_2.txt и shopping\_list\_3.txt, а затем создает новый файл shopping\_list.txt, в который помещает все прочитанные строки без повторов и в алфавитном порядке.

Последняя строка всех файлов должна быть пустой.

### Пример списков покупок

Файл 1	Файл 2	Файл 3	Итоговый файл
Пиво	Орешки	Водка	Алкозельце р
Чипсы	Пиво	Пиво	Водка

Сникерс	Сухарики	Пицца	Вяленое мясо
Чипсы	Сушеная рыба	Алкозельцер	Кефир
	Вяленое мясо	Чипсы	Орешки
			Пиво
			Сникерс
			Сухарики
			Сушеная рыба
			Чипсы

#### Пример использования

```
$ python shopping_list.py
```

## #36: БУЛОЧКИ И ПИРОЖКИ

Сложность: 4 из 10

Написать программу **bulka.py**, которая получает из первого аргумента командной строки список продуктов, разделенных запятой. Программа должна вывести этот список в удобном для человека виде:

1. Первое слово должно начинаться с заглавной буквы.
2. После запятых должны стоять пробелы.
3. Последние два слова должны быть разделены союзом «и».
4. Предложение должно заканчиваться точкой.

Примечание: если в программу передать набор слов заключенных в кавычки, то они будут восприняты как один аргумент.

#### Пример использования

```
$ python bulk.py "булочки, пирожки, печенье"
$ Булочки, пирожки и печенье.
```

## #37: КОЛИЧЕСТВО СЛОВ В ПРЕДЛОЖЕНИИ

Сложность: 6 из 10

Написать программу **count.py**, которая получает из первого аргумента командной строки предложение, а затем выводит количество слов в этом предложении.

В предложении могут быть двойные пробелы, а также неправильно расставленные знаки препинания (запятые, точки, восклицательный и вопросительный знаки).

Примечание: если в программу передать набор слов заключенных в кавычки, то они будут восприняты как один аргумент.

### **Пример использования**

```
$ python count.py "Мы с Андреем пошли на рыбалку , а затем  
вернулись домой ! "  
$ 10
```