**DZ\_m\_4\_l\_4**

**Подвиг 1.**

Вводится четырехзначное целое положительное число. Подумайте, как можно определить итератор для перебора его цифр. Выведите цифры этого введенного числа (с помощью итератора) в одну строчку через пробел.

**Sample Input:** 4387

**Sample Output:** 4 3 8 7

**Подвиг 2.**

 Вводится целое положительное число a. Необходимо определить генератор, который бы возвращал модули чисел в диапазоне [-a; a], а затем еще один, который бы вычислял кубы чисел (возведение в степень 3), возвращаемых первым генератором.

Вывести в одну строчку через пробел первые четыре значения. (Полагается, что генератор выдает, как минимум четыре значения).

**Sample Input:**

3

**Sample Output:**

27 8 1 0

**Подвиг 3.**

Используя символы малых букв латинского алфавита (строка ascii\_lowercase):

from string import ascii\_lowercase

запишите генератор, который бы возвращал все сочетания из двух букв латинского алфавита. Выведите первые 50 сочетаний на экран в строку через пробел.

Например, первые семь начальных сочетаний имеют вид:

aa ab ac ad ae af ag

**Подвиг 4.**

На вход программы поступает строка с названиями пунктов меню, записанные в одну строчку через пробел. Необходимо задать функцию с именем get\_menu, которая преобразует эту строку в список из слов и возвращает этот список. Сигнатура функции, следующая - def get\_menu(s): ... Определите декоратор для этой функции с именем show\_menu, который отображает список на экран в формате:

**Sample Input:** Главная Добавить Удалить Выйти

**Sample Output:**

1. Главная

2. Добавить

3. Удалить

4. Выйти

**Подвиг 5.**

Вводятся две строки из слов (слова записаны через пробел). Объявите функцию, которая преобразовывает эти две строки в два списка слов и возвращает эти списки.

Определите декоратор для этой функции, который попарно объединяет эти списки. Полученный список - должен возвращаться при вызове декоратора.

Примените декоратор к первой функции и вызовите ее для введенных строк. Результат отобразите на экране.

**Sample Input:**

house river tree car

дом река дерево машина

**Sample Output:**

[('house', 'дом'), ('river', 'река'), ('tree', 'дерево'), ('car', 'машина')]