**Задачи на Comprehensions**

1. Сгенерировать список нечётных двузначных чисел.
2. Сгенерировать список из 10 чисел степени 2. От 1 до 10.
3. Сгенерировать список всех трёхзначных чисел кратных 5 и 3.
4. Дан список, упорядоченный по не убыванию элементов в нем. Определите, сколько в нем различных элементов. Set не использовать.
5. Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящий на противоположном конце этого списка. Например, если на вход подаётся список "1 3 5 6 10", то на выход ожидается список "13 6 9 15 7" (без кавычек).Если на вход пришло только одно число, надо вывести его же. Вывод должен содержать одну строку с числами нового списка, разделёнными пробелом.

***Sample Input 1:***

***1 3 5 6 10***

***Sample Output 1:***

***13 6 9 15 7***

***Sample Input 2:***

***10***

***Sample Output 2:***

***10***

***Sample Input 3:***

***10 2***

***Sample Output 3:***

***4 20***

1. Создать список используя comprehension. Продублировать все неповторяющиеся элементы.

**Задачи на функции**

1. Дана функция, которая выводит все простые числа в промежутке от 1 до 100. Написать декоратор который будет проверять время работы этой функции. Задекорировать функцию. Вывести время работы этой функции, а так же сами простые числа.
2. Дана функция которая запрашивает у пользователя определённые данные для регистрации на портале и запоминает их. Напишите декоратор, который будет засекать время проведённое пользователем на портале при регистрации.

**Дополнительные задачи**

1. Напишите рекурсивную функцию is\_power(n), которая возвращает True, если переданное натуральное число является степенью двойки, и False иначе.

***Sample Input:***

***1048576***

***Sample Output:***

***True***

1. Сгенерируйте список как показано ниже при помощи генератора:

***[1, 2 , 3 , 4 , 5]***

***6 7 8 9 10***

***11 12 13 14 15***

***16 17 18 19 20***

1. Сгенерировать список всех простых чисел до 100 с помощью генератора.
2. Создать список используя comprehension. Преобразуйте исходный список, вставив 0 между числами. Дополнительный список не создавать!

***Sample Input:***

***7 4 1***

***Sample Output:***

***7 0 4 0 1***