**№1**

В строке найти первую последовательность цифр, прибавить к этому числу единицу и заменить начальное число в строке полученным результатом.

*При обработке числа в строке учесть тот факт, что оно может быть на столько большим, что не влезет ни в один числовой тип данных, поэтому, просто преобразовать его в число, прибавить 1 и помесить результат обратно в строку запрещается.*

Прим.

"dfgdf455dfg545ff" => "dfgdf456dfg545ff",

"aaaaa0123456999aaa" => "aaaaa0123457000aaa",

"aaaa999aaa9a" -> "aaaa1000aaa9a"

**№2**

Очеловечивание заголовка. Строка задана в CamelCase или для разделения слов используются нижние подчеркивания. Привести строку к следующему виду: слова должны быть разделены пробелами, каждое новое слово начинается с большой буквы.

Прим.

"\_user\_id" -> "User Id",

"categoryFullPath" -> "Category Full Path"

**№3**

Сложить 2 целых числа, заданных строками. Числа могут быть отрицательными. Ответ так же предоставить в виде строки.

Прим.

("-123456789", "-987654321") => "-1111111110"

("123456789", "-987654321") => "-864197532"

P.S. После прохождения классов, сделать тоже самое через перегрузку операторов.

**№4**

Заполнить массив n числами от 1 до n без повторений в случайном порядке - скорость алгоритмов.

**№5**

Дано прямоугольное поле размером *n*\**m* клеток. Можно совершать шаги длиной в одну клетку вправо или вниз. В каждой клетке записано некоторое натуральное число. Необходимо попасть из верхней левой клетки в правую нижнюю. Вес маршрута вычисляется как сумма чисел со всех посещенных клеток. Необходимо найти минимальный вес маршрута и отобразить сам маршрут в виде матрицы из нулей и единиц.

*Рекурсия*