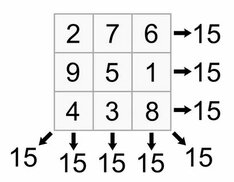
**Задания**

***1. Проверить валидность скобок***

Вам будет дана последовательность скобок проверьте, что все скобки закрыты.

***2. Определение магического квадрата***

***Магический квадрат*** *(или волшебный квадрат) — это квадратная таблица размером n × n, заполненная n2 различными числами таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова.*



***3. Проверка на арифметическую \ геометрическую прогрессию***

**Арифметическая прогрессия** — это последовательность, каждый член которой, начиная со второго, равен предыдущему, к которому прибавляется одно и то же число.

**Геометрическая прогрессия** — это последовательность, каждый член которой, начиная со второго, равен предыдущему, умноженному на одно и то же число.

***4. Определить тип треугольника по длинам сторон***

***5. Проверка на совершенное число***

Натуральное **число** считается **совершенным**, если оно равно сумме своих делителей, отличных от самого **числа**.

6 = 1+2+3

***6. Последовательность Фибоначчи***

**Последовательность Фибоначчи** - это последовательность, в которой каждое число является суммой двух предыдущих.

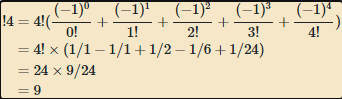
1 1 2 3 5…

***7. Нахождение факториала \ субфакториала***

**Факториал** числа n — это произведение всех натуральных чисел от единицы до n.

4! = 1 \*2 \* 3 \*4 = 24

Субфакториал числа n (обозначается как !n) — это количество беспорядков порядка n, то есть перестановок порядка n без неподвижных точек.



***8. Сжатие строки (rle)***

Необходимо реализовать функцию, принимающую в аргументах строку, состоящую из букв и вернуть новую строку, в которой повторяющиеся буквы заменены количеством повторений.

AAAABBCAA --> A4B2C1A2

***9. Бинарный поиск***

**Бинарный поиск** — это алгоритм, используемый для поиска элемента в отсортированном массиве путем многократного деления интервала поиска пополам. Принцип алгоритма бинарного поиска очень прост. Мы всегда имеем дело с упорядоченным массивом. Это позволяет прибегать к хитрости - на каждой итерации цикла, который мы запускаем, мы вычисляем индекс среднего элемента. Этот элемент сравниваем с искомым. Если элемент из середины массива оказался меньше, чем тот, что мы ищем, значит нужный нам находится правее по массиву, ведь массив упорядочен. Соответственно, нам нужно перейти в следующую итерацию цикла, "оставляя" ей лишь правую часть массива - в ней снова будет найден средний элемент и алгоритм повторится. Если же элемент из середины массива оказался больше искомого, то мы переходим к следующей итерации, отбрасывая правую часть массива, а оставляя левую. Если же элемент совпадает с искомым, мы выходим из цикла.