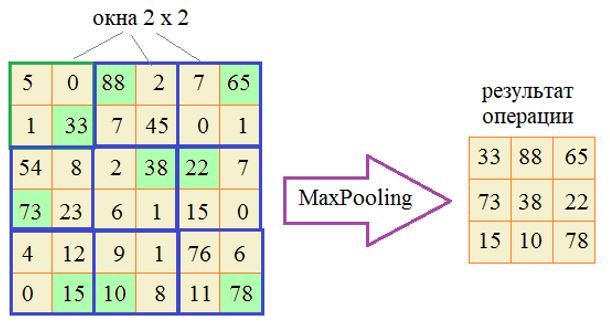
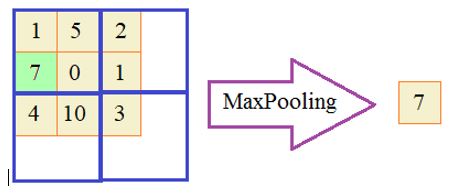
Видео-разбор подвига (решение смотреть только после своей попытки): <https://youtu.be/9wNoWmEHdfo>

**Подвиг 10 (на повторение).** В нейронных сетях использую операцию под названием Max Pooling. Суть ее состоит в сканировании прямоугольной таблицы чисел (матрицы) окном определенного размера (обычно, 2x2 элемента) и выбора наибольшего значения в пределах этого окна:



 Или, если окна выходят за пределы матрицы, то они пропускаются (игнорируются):



Мы повторим эту процедуру. Для этого в программе нужно объявить класс с именем MaxPooling, объекты которого создаются командой:

mp = MaxPooling(step=(2, 2), size=(2,2))

где step - шаг смещения окна по горизонтали и вертикали; size - размер окна по горизонтали и вертикали.

Параметры step и size по умолчанию должны принимать кортеж со значениями (2, 2).

Для выполнения операции Max Pooling используется команда:

res = mp(matrix)

где matrix - прямоугольная таблица чисел; res - ссылка на результат обработки таблицы matrix (должна создаваться новая таблица чисел.

Прямоугольную таблицу чисел следует описывать вложенными списками. Если при сканировании таблицы часть окна выходит за ее пределы, то эти данные отбрасывать (не учитывать все окно).

Если matrix не является прямоугольной таблицей или содержит хотя бы одно не числовое значение, то должно генерироваться исключение командой:

raise ValueError("Неверный формат для первого параметра matrix.")

Пример использования класса (эти строчки в программе писать не нужно):

mp = MaxPooling(step=(2, 2), size=(2,2))

res = mp([[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 8, 7, 6],

[5, 4, 3, 2]])    # [[6, 8], [9, 7]]

Результатом будет таблица чисел:

6 8  
9 7

P.S. В программе достаточно объявить только класс. Выводить на экран ничего не нужно.

# TEST  
# mp = MaxPooling(step=(2, 2), size=(2,2))  
# m1 = [[1, 10, 10], [5, 10, 0], [0, 1, 2]]  
# m2 = [[1, 10, 10, 12], [5, 10, 0, -5], [0, 1, 2, 300], [40, -100, 0, 54.5]]  
# res1 = mp(m1)  
# res2 = mp(m2)  
#  
# assert res1 == [[10]], "неверный результат операции MaxPooling"  
# assert res2 == [[10, 12], [40, 300]], "неверный результат операции MaxPooling"  
#  
# mp = MaxPooling(step=(3, 3), size=(2,2))  
# m3 = [[1, 12, 14, 12], [5, 10, 0, -5], [0, 1, 2, 300], [40, -100, 0, 54.5]]  
# res3 = mp(m3)  
# assert res3 == [[12]], "неверный результат операции при MaxPooling(step=(3, 3), size=(2,2))"  
#  
# try:  
# res = mp([[1, 2], [3, 4, 5]])  
# except ValueError:  
# assert True  
# else:  
# assert False, "некорректо отработала проверка (или она отсутствует) на не прямоугольную матрицу"  
#  
# try:  
# res = mp([[1, 2], [3, '4']])  
# except ValueError:  
# assert True  
# else:  
# assert False, "некорректо отработала проверка (или она отсутствует) на не числовые значения в матрице"