# Introduction to Data Science and Programming

Assignment 7

Due November 11, 2022, 11:59

## Description

This week you are being provided with the function `search_pascal_multiples_slow(row_limit)` that performs a task in a slow way. The implemented task is the following, closely related to an unsolved problem in mathematics:

Consider Pascal's triangle:

```
              1
            1   1
          1   2   1
        1   3   3   1
      1   4   6   4   1
    1   5   10  10  5   1
  1   6   15  20  15  6   1  ....
```

Which numbers in Pascal's triangle, without its outermost 2 numbers in each row (shown in red), appear more than 3 times?

The function has one parameter, `row_limit`, which denotes the number of rows to search through in Pascal's triangle. The output is a sorted list. For `row_limit = 250`, which we keep fixed, the expected output is: `[120, 210, 1540, 3003, 7140, 11628, 24310, 61218182743304701891431482520]`.

## Submission

Your task is to write the function `search_pascal_multiples_fast(row_limit)` which implements the same problem but runs at least 5 times faster. In this assignment you are allowed to use *any* Python library, method, or trick, that you can come up with, even if not covered in a lecture, but hardcoding the solution or using a language other than Python is not allowed. For example, just returning the expected list is not ok. For your fast implementation, feel free to copy-paste and change the slow function, or to re-implement the function from scratch.

Submit the provided skeleton file `pascal.py` with your filled in solution. The file already contains a timer which will tell you how many times faster your function runs: Running `python pascal.py` will print the correct solution from the slow function, your function's solution, which must be identical, and the factor by which your function is faster. This factor must be at least 5.00. Hint: `from collections import Counter`

## Prize

The student with the fastest implementation submitted by the due date will win a prize and will get the chance to explain their implementation in the lecture (if they want to).

## Advanced material

If you are interested to learn more about this mathematical problem, see also:
https://www.youtube.com/watch?v=Z3xq4ODNeZs
https://en.wikipedia.org/wiki/Singmaster%27s_conjecture