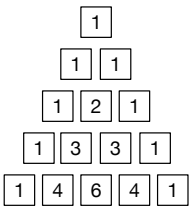
Homework Arrays

Problem 1 — The Sieve of Eratosthenes

Implement the Sieve of Eratosthenes to find all prime numbers between 1 and 1000000. The idea is to make a list of boolean values `is_not_prime`, such that `is_not_prime[i]` being True means that we know for sure that `i` is not a prime. Initially, we do not know that any number is not a prime, so the list is initialized to all values being `False`. The first two values of the array are irrelevant, since you can fight over whether zero and one are prime numbers. We start with the third entry, for 2. Its value is `False`, so we know that it is not not a prime, i.e. it is a prime. We then set the array to `True` for all true multiples of 2, that is `is_not_prime[4], is_not_prime[6], is_not_prime[8], is_not_prime[10],…` . The next index with False is 3. We again set the indices at all true multiples of 3 to `True`, i.e. `is_not_prime[6], is_not_prime[9], is_not_prime[12], …` . We then look for the next smallest index in the array that is False is 5. Again, we set the array to True at all multiples of 5. Then we find the next index in the array that is False, which will be 7 and do the same thing. We repeat until the index is larger than $\sqrt{1000000}$.

Problem 2 — Pascal's Triangle

Pascal's Triangle contains the combinations $\binom{n}{k}$ in form of a triangle. On the left and the right edge of the triangle, there are ones. An element in the middle is the sum of the two elements to the left and the right. We can implement this as a two dimensional array (a Python list of Python lists), where we move the rows to the left, so that we have




```
1, 0, 0, 0, 0, 0
1, 1, 0, 0, 0, 0
1, 2, 1, 0, 0, 0
1, 3, 3, 1, 0, 0
1, 4, 6, 4, 1, 0
1, 5,10,10, 5, 1
```

The definition of the array elements is

```
pascal[i+1][j] = pascal[i][j]+pascal[i][j+1].
```

You will have to supplement this recursive definition. When you are done, create the 21 by 21 Pascal triangle and derive $\binom{20}{10}$.