1. Finding GCD and LCM
2. Determine if Prime Number
3. Sliding Window
   1. Use Deque to store the index of the maximum (or potential maximum element)
   2. Leverage head and tail to determine the range of the deque
   3. First find the maximum of the first k elements
   4. For the rest (n – k) elements:
      1. Remove the previous maximum outside the current window
      2. Remove the smaller element (compared to the new element) and add the new element to from the tail of the deque.
   5. Do remember that it is the head and tail determinde what the actual range of deque is. (Which means the “removed” element might still in the deque, but it is ok as long as it is outside the [head, tail)
   6. Code:  
      A white screen with black and green text

      AI-generated content may be incorrect.  
      A screenshot of a computer code

      AI-generated content may be incorrect.­­
4. Print Pascal’s Triangle (I call it as Pascal’s Triangle just because I am in English now, it should actually be called as Yanghui’s Triangle or Jiaxian’s Triangle)
   1. One Array Version
      1. Define and array
      2. Explicitly set the first element as 1
      3. For each row, update the elements from left to right.
      4. Explicitly set the tail element as 1, namely a[r – 1] = 1
      5. Code:

A white background with black and white text

AI-generated content may be incorrect.

* 1. Two Array Version
     1. Define two array – one for the previous row and one for the current
     2. Define two pointers point to one separately.
     3. For each row, c[j] = p[j – 1] + p[j]
     4. Explicitly set the border elements as 1
     5. Swap the two pointers
     6. Code:

A screenshot of a computer program

AI-generated content may be incorrect.