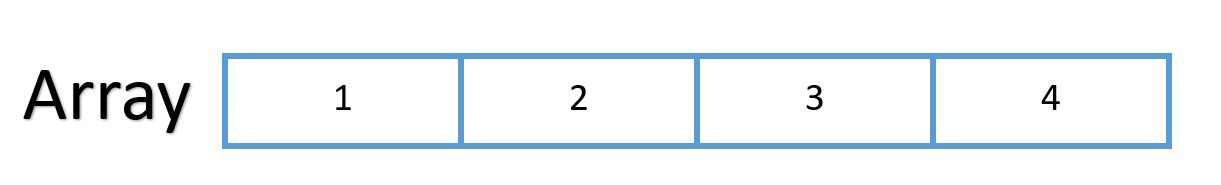
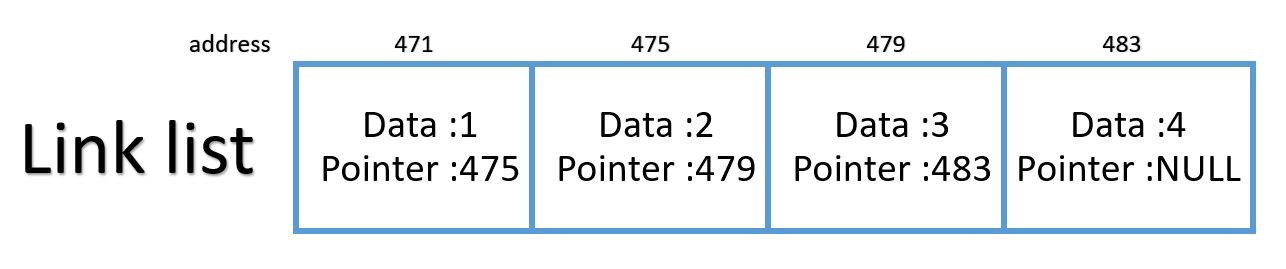
**Link list คืออะไร**

Link list จริงๆแล้วก็คล้ายๆ array ซึงมีข้อแตกต่างกันอยู่เล็กน้อย





**Array**ในแต่ละกล่องจะเก็บแค่ค่าที่เราใส่ลงไป

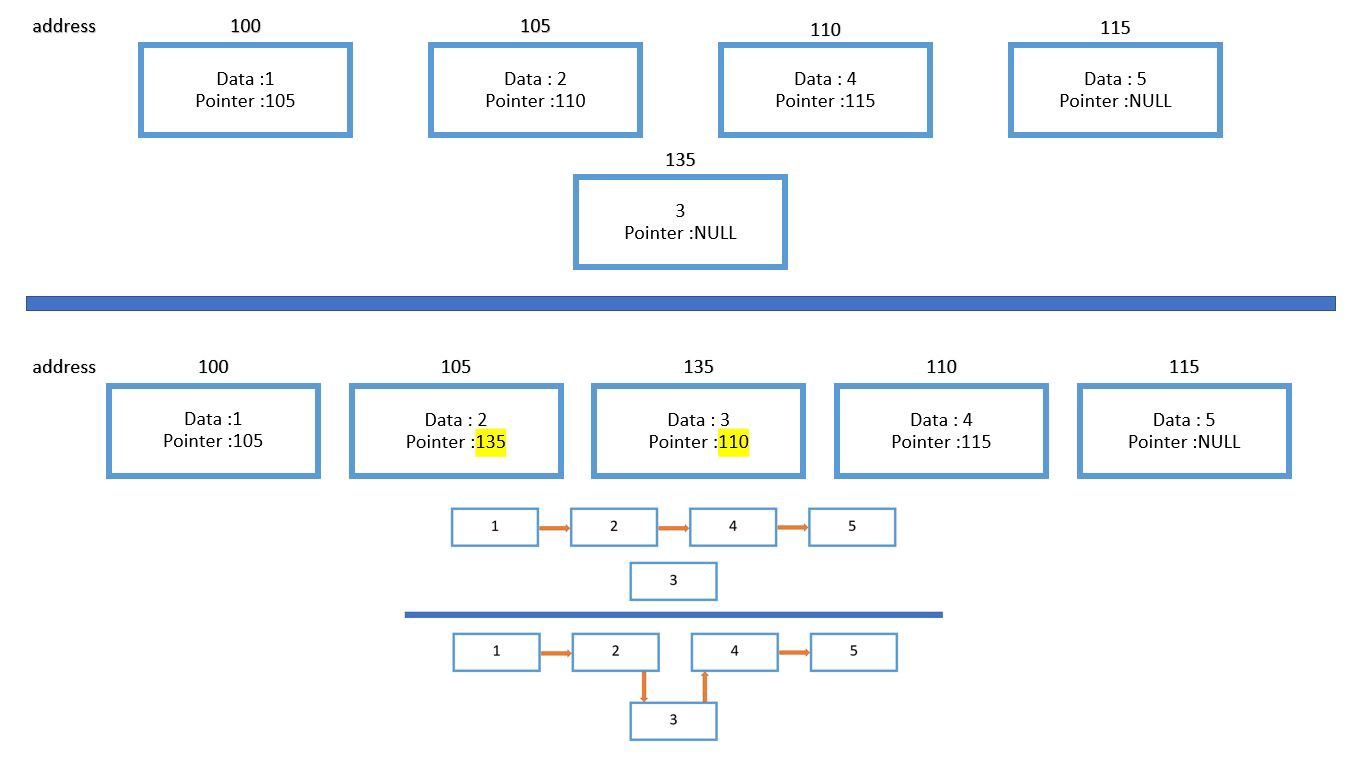
**Link list** นั้นจะเก็บ address ขอช่องที่ติดกันอยู่ด้วย สามารถมีทั้งแบบ ทางเดียวและแบบไปกลับได้

**อธิบายเพิ่มเติมเพื่อให้เข้าใจภาพได้ง่ายขึ้น**

ตัวเลขบนกล่องคือ address(ค่าตำแหน่งที่เก็บข้อมูลบน memory)  
กล่อง คือ struct ที่เก็บตัวแปรสองชนิด int , pointer(ตัวแปรที่เก็บค่า address)  
ค่า address ที่แสดงเป็นเพียงค่าสมมุติ

**ทำไมมี array แล้วต้องใช้ link list**

การที่เราเก็บข้อมูลเป็น array ในบางเหตุการที่เราต้องการที่จะแทนข้อมูลลงไปใน array นั้นเราต้องทำการขยับข้อมูลที่มีอยู่ให้ถอยหลังออกไปทำให้เราต้องเสียเวลาเขียน loop เพื่อขยับข้อมูลใน array แต่ link list สามารถแก้ปัญหานี้ได้เพราะการสามารถนำข้อมูลใหม่เข้ามาแทรกได้โดยไม่จำเป็นต้องขยับข้อมูลที่มีอยู่เลย เพียงแค่เปลียน pointer เท่านั้น



จะเห็นได้ว่าการแทรกข้อมูลของ link list นั้นเพียงแค่เป็นการแก้ไข pointer ให้ไปยังกล่องใหม่แล้วให้กล่องใหม่ ชี้ไปยังกล่องถัดไป

นอกจากนี้เรายังสามารถนำ link list ไปใช้ทำ stack หรือ queue

Link list ทางเดียว กับสองทางต่างกันยังไง

Link list ทางเดียว

นั้นเราจะไม่สามารถเรียกข้อมูลก่อนหน้าได้เนื่องจากเรามีเพียง pointer ที่ชี้ไปยังกล่องถัดไปเท่านั้น ทำให้ถ้าต้องการข้อมูลก่อนหน้าจำเป็นต้องไปไล่จากหัวใหม่เสมอ

Link list สองทาง

สามารถเรียกข้อมูลก่อนหน้าได้เพราะในตัวกล่องมีทั้ง pointer ที่ชี้ทั้งกล่องถัดไปและกล่องก่อนหน้านั้นเอง

สร้าง function

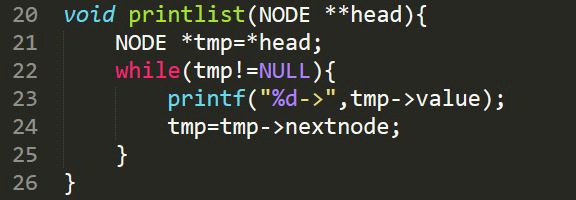
Function สร้าง กล่อง

เราจะสร้างฟังชั้นที่จะนำข้อมูลมาใส่ node และเชื่อม node เข้าด้วยกัน

เริ่มจากสร้าง node ขึ้นมา(10–12)บรรทัดที่13เซ็คว่ามีnodeหรือไม่ถ้าไม่มีก็ให้ใช้nodeที่สร้างขึ้นมาเลยแต่ถ้ามี node อยู่แล้วให้หากล่องสุดท้ายแล้วให้ pointer ชี้มายัง node ที่สร้างใหม่

**Function print**

การแสดงข้อมูลที่เรามีอยู่นั้นสามารถทำได้โดยเริ่มจากหัวแล้วขยับไปยัง node ต่อไปด้วย pointer ที่เก็บไว้



**Function delete**

เราสามารถลบ node ที่อยู่ภายใน link list ได้โดยเปลียน pointer ให้ข้ามไป node ถัดไปเลย แต่ในการลบนั้นมีหลายกรณีซึ่งต้องใช้วิธีในการลบที่แตกต่างกัน

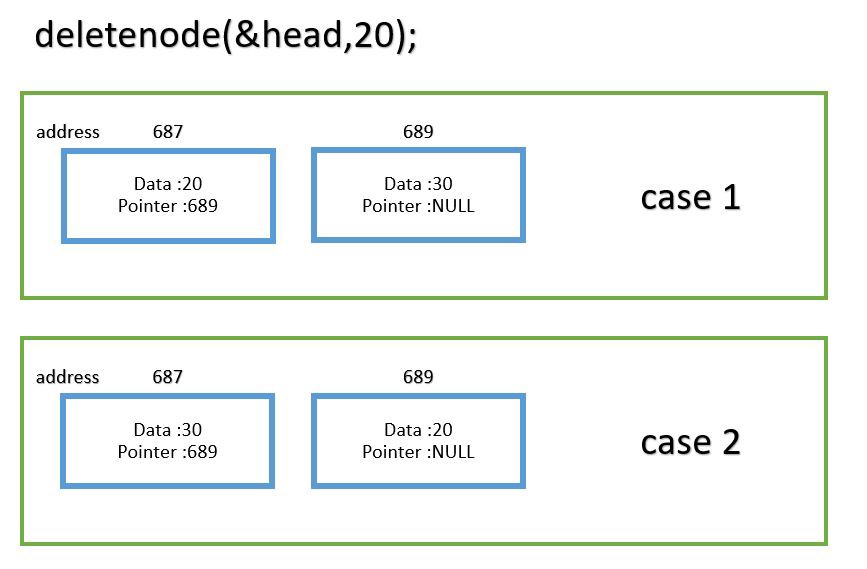
มีสองกรณีคือตัวที่จะลบนั้นเป็นหัวของ link list กับ เป็นตัวอื่นที่ไม่ใช่หัว

**Function delete**

เราสามารถลบ node ที่อยู่ภายใน link list ได้โดยเปลียน pointer ให้ข้ามไป node ถัดไปเลย แต่ในการลบนั้นมีหลายกรณีซึ่งต้องใช้วิธีในการลบที่แตกต่างกัน

มีสองกรณีคือตัวที่จะลบนั้นเป็นหัวของ link list กับ เป็นตัวอื่นที่ไม่ใช่หัว



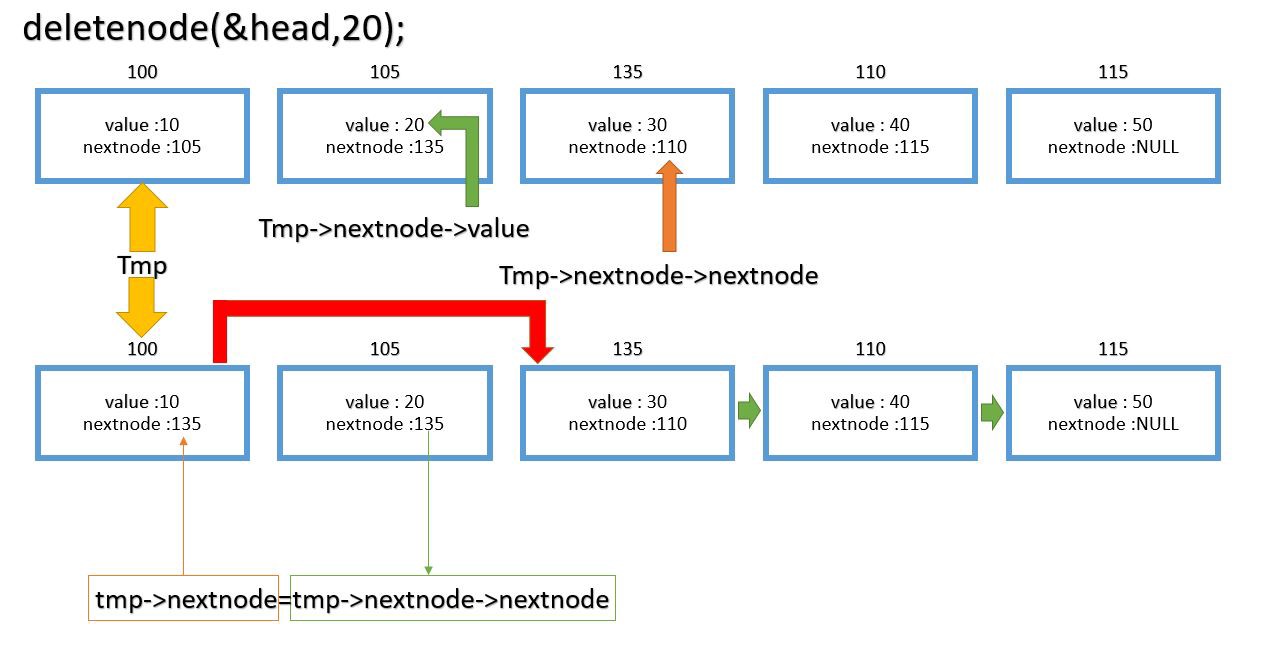


ในกรณีที่ 1 เราจำเป็นต้องแก้ pointer ของ head ที่เข้ามาให้เป็น node ถัดไป อีกกรณีเราจะแก้ไขแค่ address ของ node ให้ชี้ข้าม node ที่เราจะลบไป







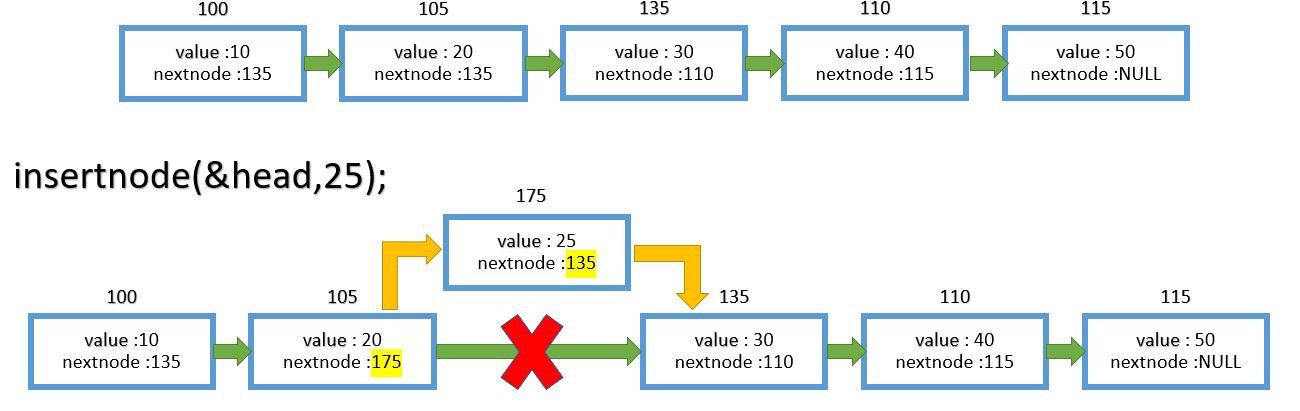


**Function insert**

ใน link list เราสามารถแทรงข้อมูลลงไปได้โดยแค่เปลียน pointer เท่านั้น

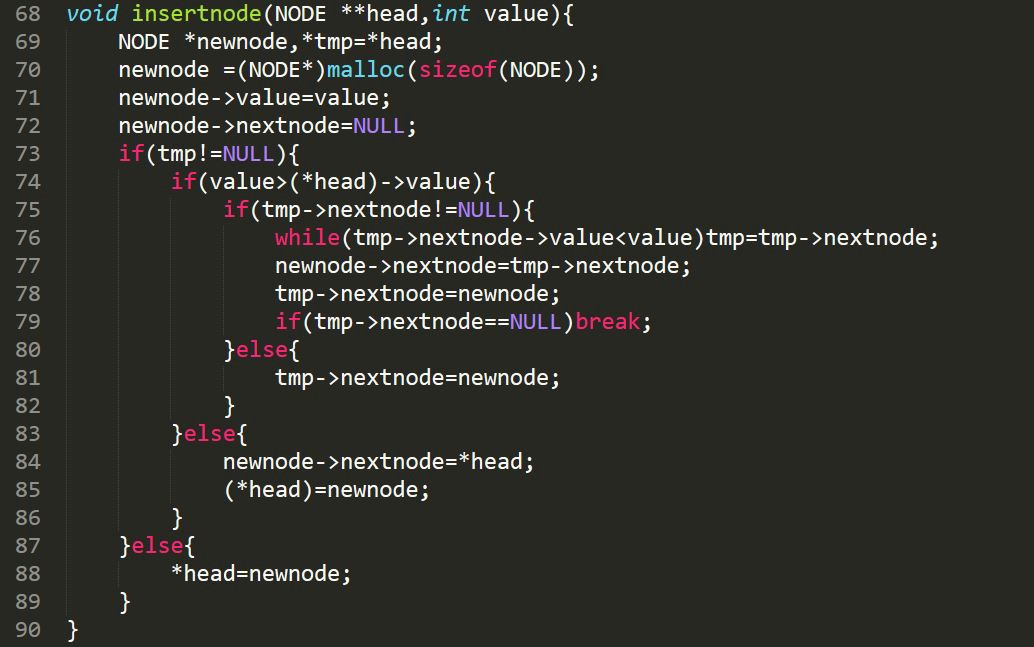
ตัวอย่างจะเป็นการแทรงข้อมูลให้มีค่าเรียงกันจากมากไปน้อย ซึ่งการที่เราเอาตัวเลขไปแทรกเพื่อเรื่องลำดับมีชื่อเรียกการจัดเรียงนี้ว่า insertion sort





การแทรงที่หัวของ link list จะต้องแยกกรณีไปเหมือน การ delete node

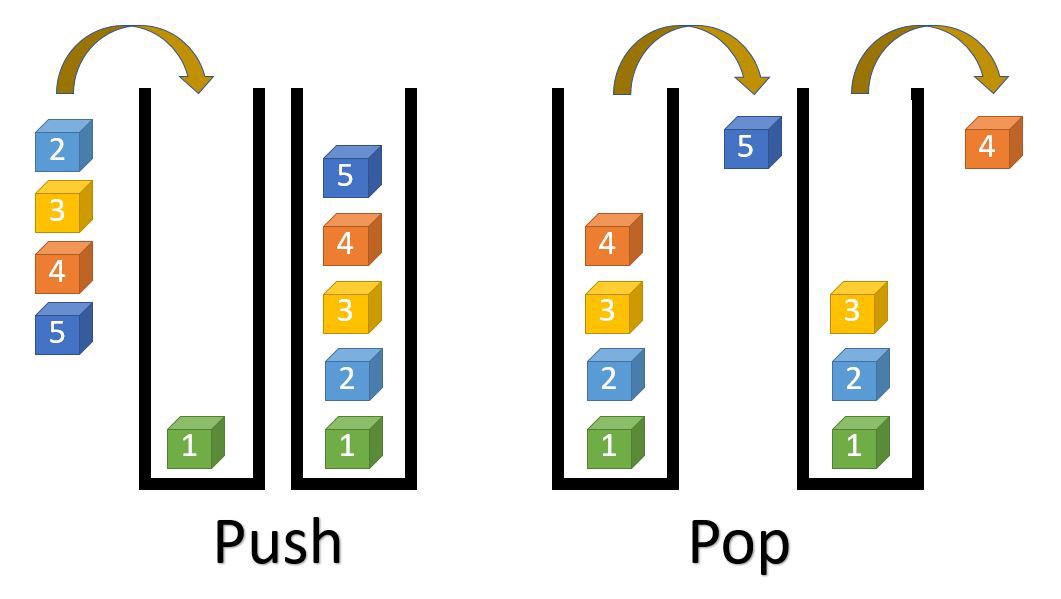




เราจะนำ link list ทางเดียวนี้มาทำเป็น stack และ queue กันแต่ก่อนอื่นเราต้องเข้าใจก่อนว่าสองอย่างนี้มันคืออะไร

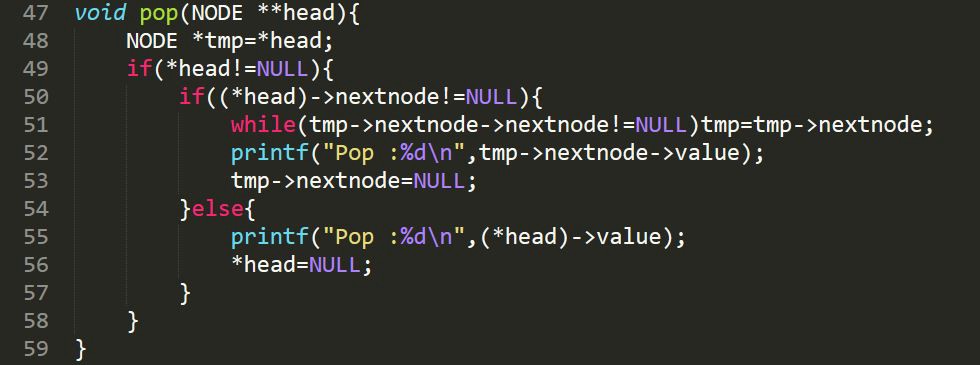
**Stack**

เข้าหลังออกก่อน



function push นั้นเหมือนกัน createnode ของ link list ปกติ

function pop เราจะขยับไป node สุดท้ายแล้วแสดงค่าออกมา และให้ node ก่อนหน้าชี้ไป NULL แทน



**Queue**

เข้าก่อนออกก่อน

