

การแสดงขั้นตอนวิธีด้วยผังงานหรือรหัสเทียม

Flowchart

Nested Selection

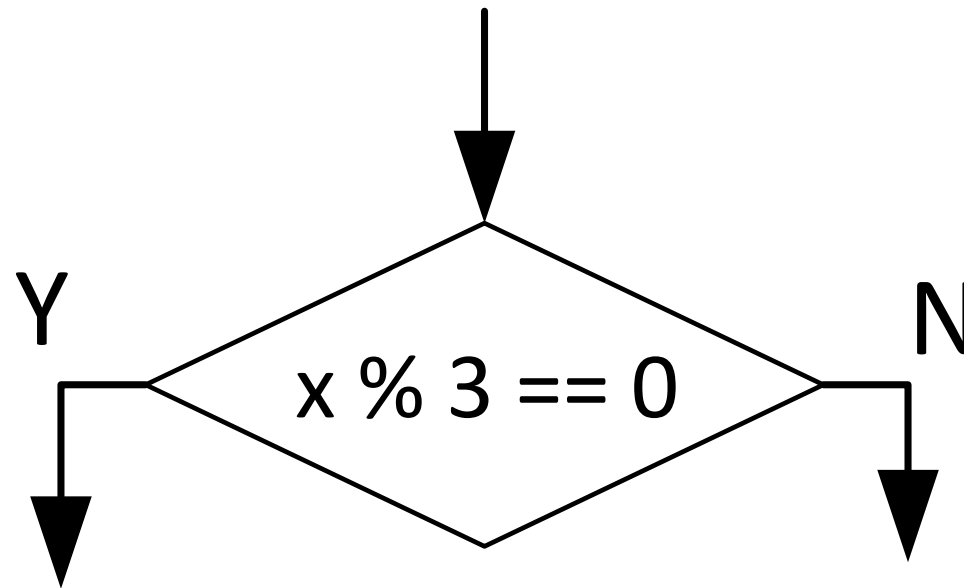
เมื่อ Decision ตัวเดียวมีทางเลือกมากกว่า 2 ทาง

- “จงรับเลขจำนวนเต็ม x เข้ามา 1 จำนวน แล้วหารเอาเศษ (%) ด้วย 3:
 - ถ้า เศษเป็นศูนย์ ให้เพิ่ม x เป็นสองเท่า
 - ถ้า เศษเป็นหนึ่ง ให้เพิ่ม x เป็นสามเท่า
 - ถ้า เศษเป็นสอง ให้บวก x ด้วย 5”

- จะเขียน Flowchart อย่างไร

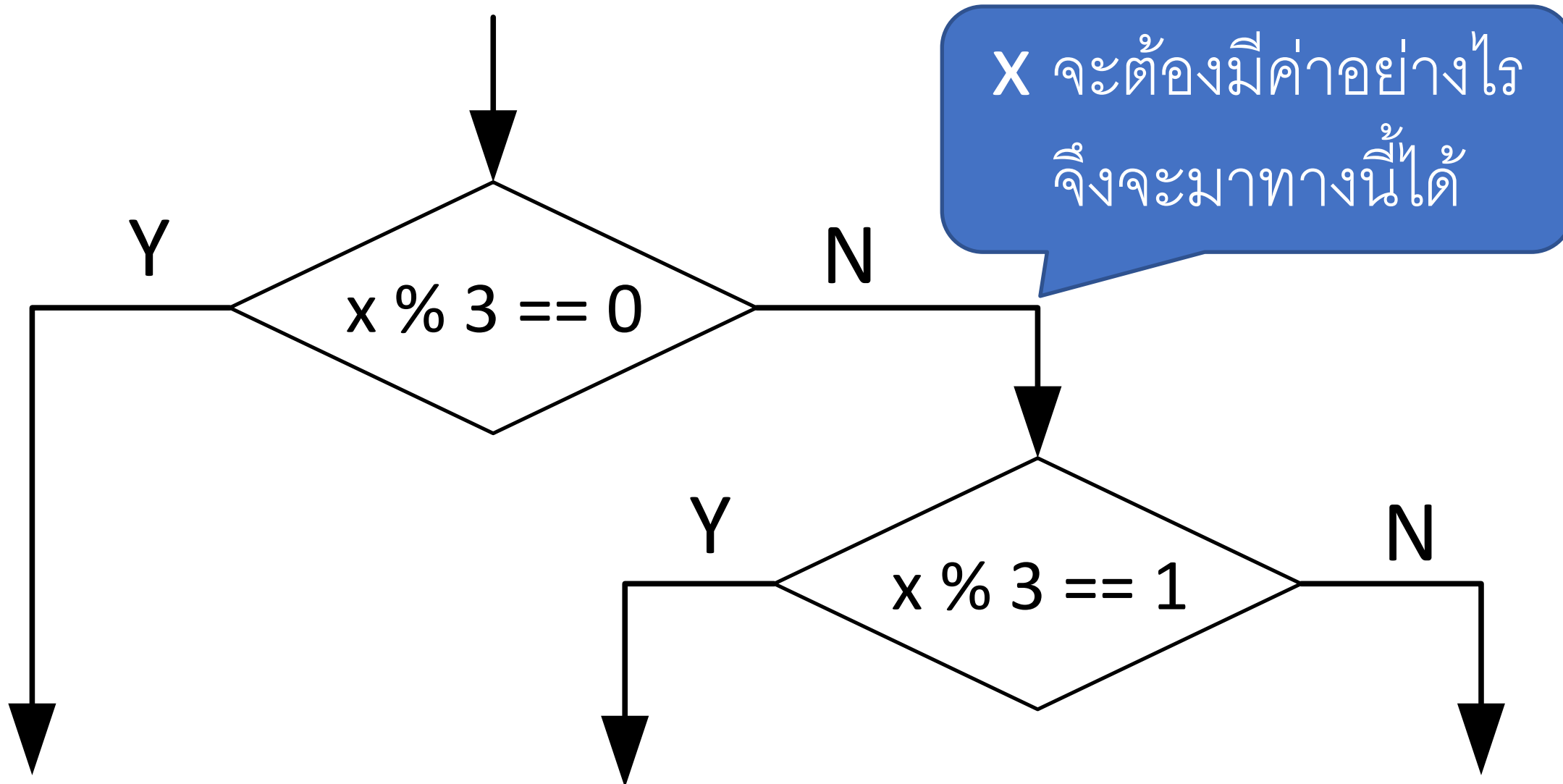
การเขียน Flowchart

- การตัดสินใจมี 3 ทางเลือก แต่ selection block จะมีทางเลือกได้แค่ 2 ทาง และเป็น Y/N เท่านั้น



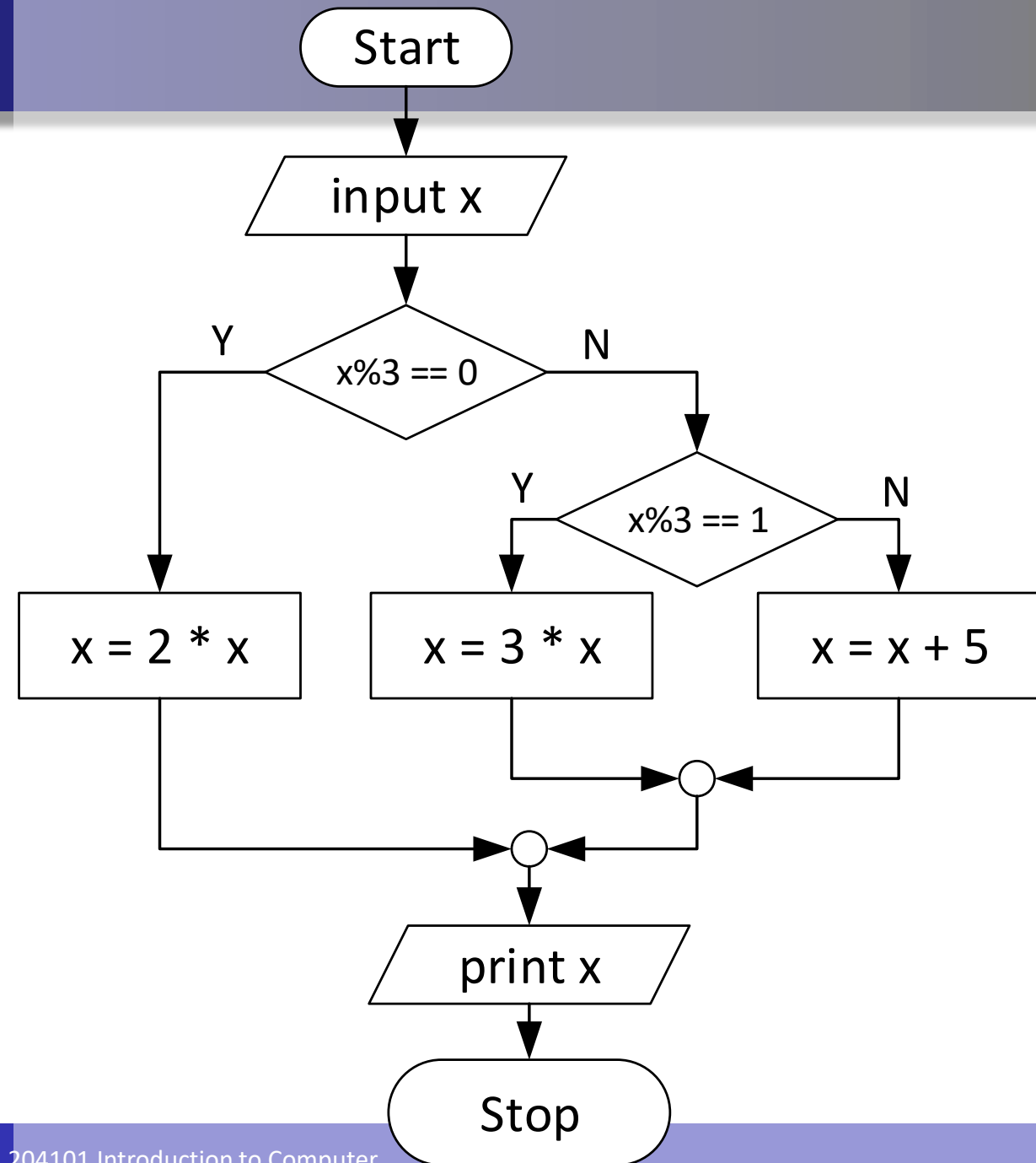
- ดังนั้น จะต้องใช้ selection มากกว่า หนึ่งตัว

การเขียน Flowchart (ต่อ)



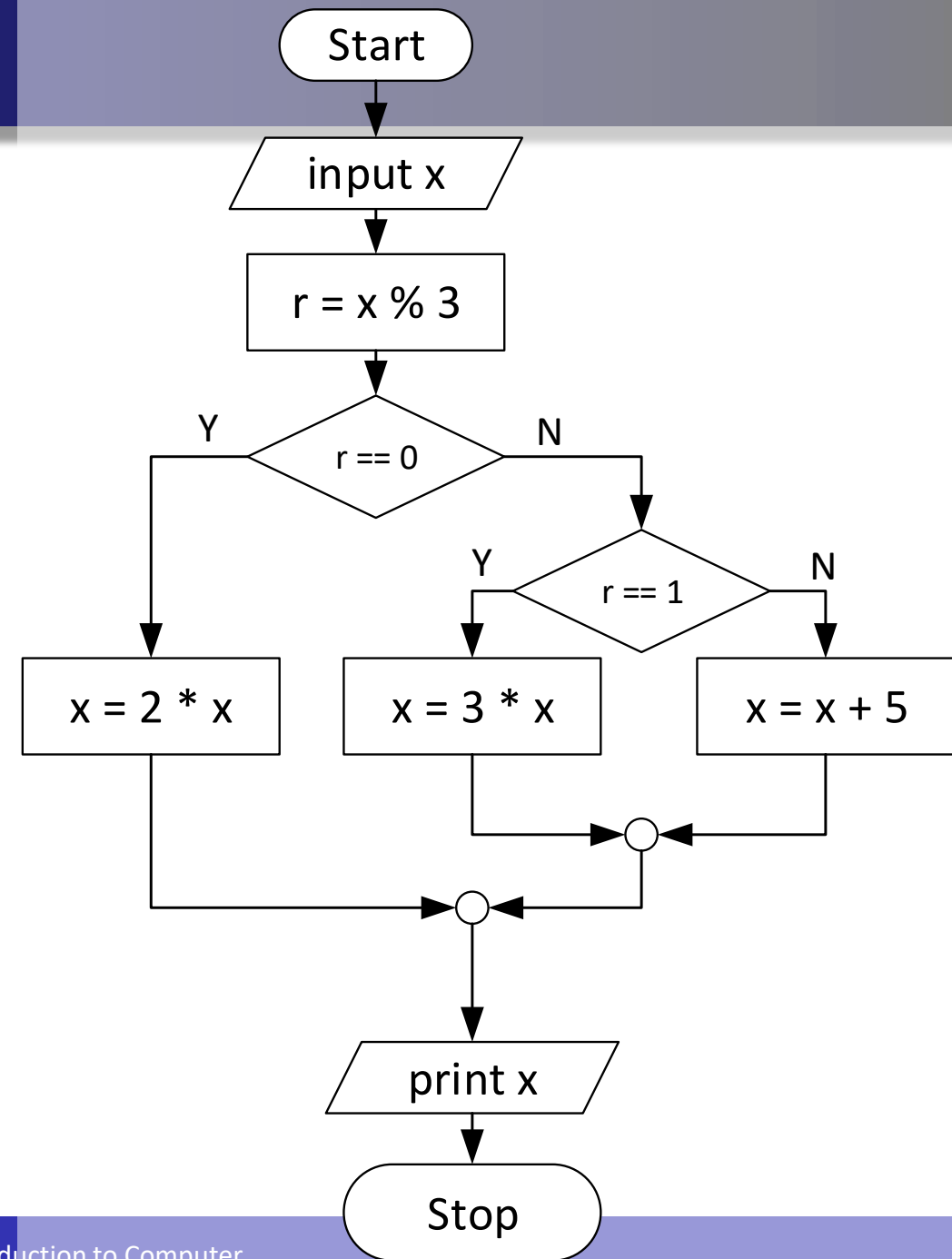
Flowchart #1

- จะเห็นว่า ต้องทำการหารเอาเศษสองครั้ง
- จำเป็นหรือไม่
- มีวิธีอื่นไหม



Flowchart #2

- เพื่อลดการคำนวณ (%) อาจใช้ตัวแปรมาช่วยได้
- ถ้าเป็นการคำนวณที่ซับซ้อนกว่านี้ ก็จะทำให้เพิ่มความเร็วของโปรแกรมได้มาก



Nested Selection

- การจัด selection blocks ต้องดูว่า
 - ครอบคลุมทุกกรณีหรือไม่
 - แต่ละกรณี เป็นไปตามที่เราต้องการจริง ($>$ กับ \geq)
- ลำดับการใช้เงื่อนไข อาจทำให้ความซับซ้อนต่างกัน
- การใช้ and, or สามารถลดความซับซ้อนของ flowchart ได้ด้วย

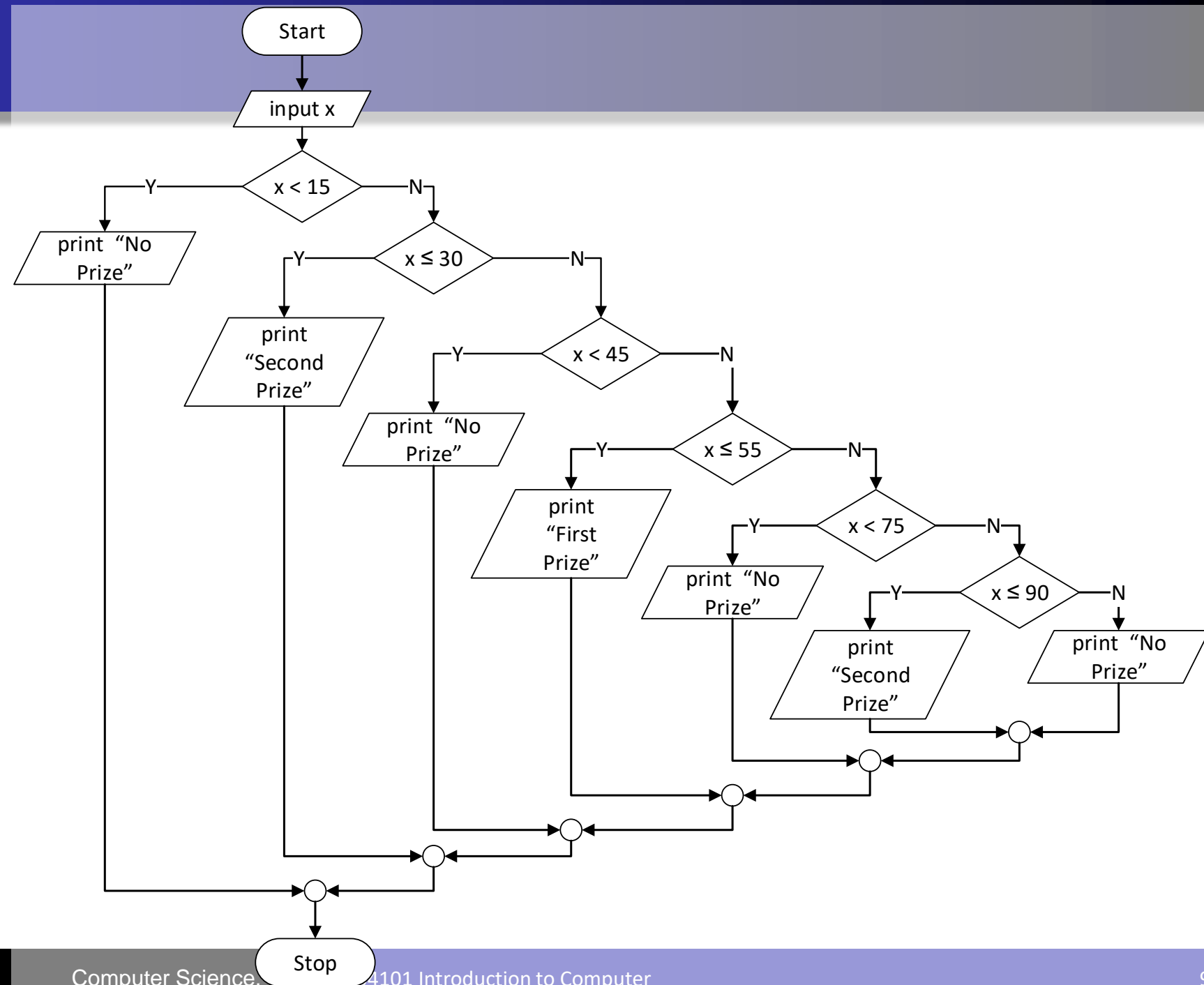
ตัวอย่าง

- ในเกมส์จับฉลาก ตัวฉลากจะมีเลข 1-99 อยู่ซึ่งอาจได้รางวัลดังนี้
 - ถ้าได้เลขในช่วง 45-55 จะได้รางวัลที่ 1
 - แต่ถ้าได้เลขในช่วง 15-30 หรือ 75-90 จะได้รางวัลที่ 2
 - นอกจากนั้น จะไม่ได้รางวัล
- ถ้าให้เขียนโปรแกรมที่รับเลขฉลาก แล้วให้บอกรางวัลที่ได้ จะเขียน flowchart อย่างไร

Flowchart

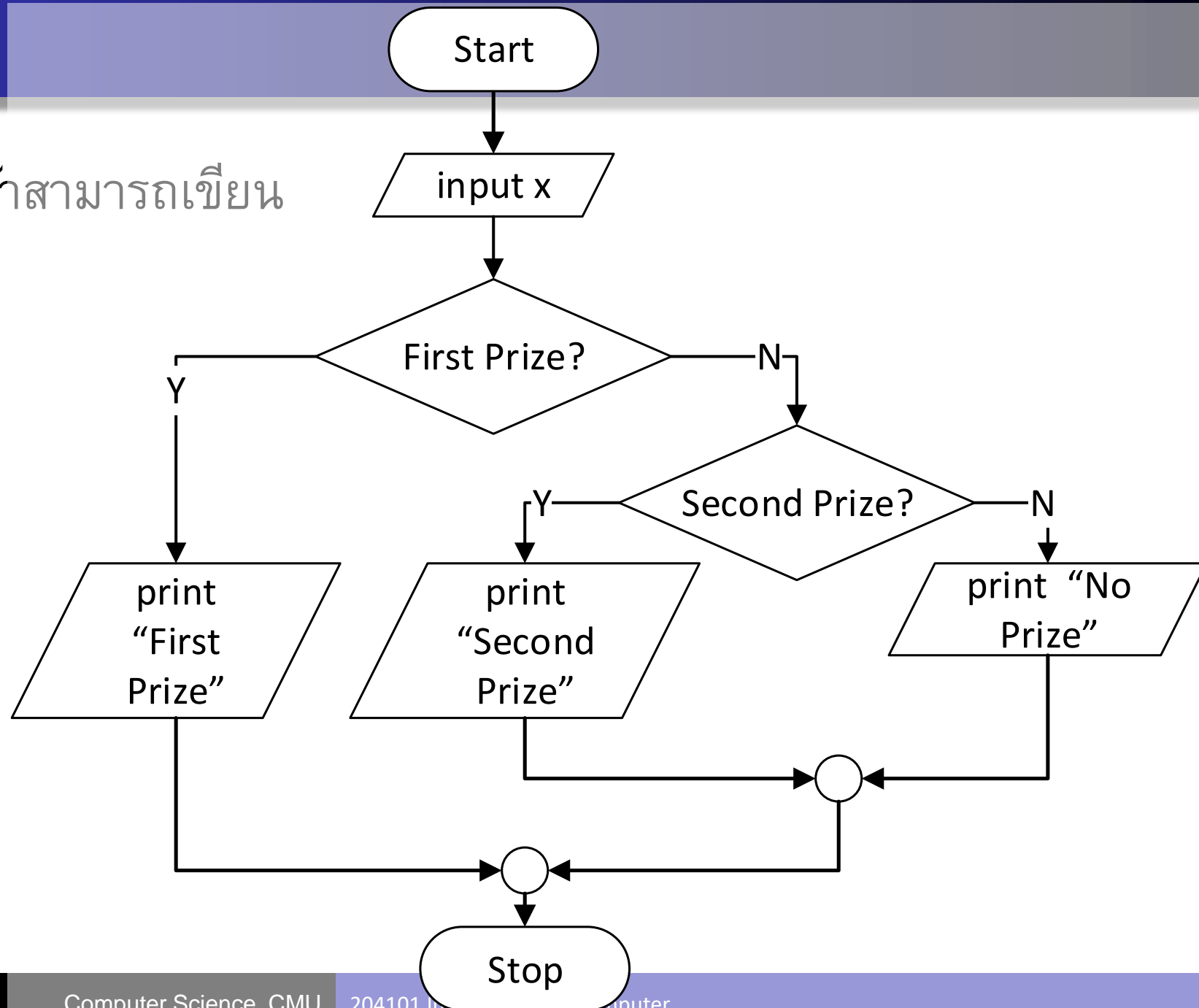
แบบ

ตรงไปตรงมา



ดูปัญหาอีกครั้ง

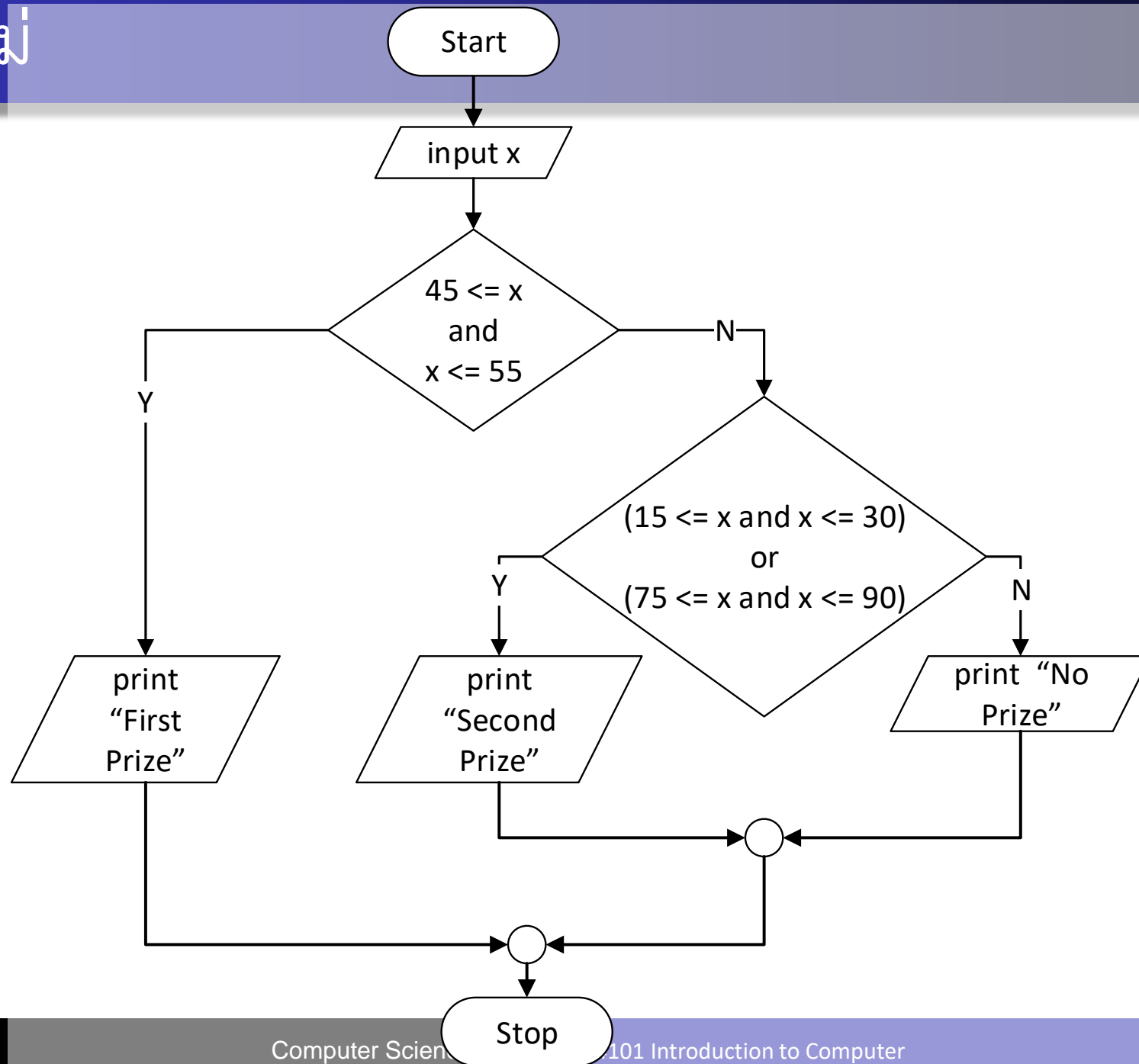
- จะซับซ้อนน้อยลงมาก ถ้าสามารถเขียนในรูปทางขวาได้
- จะทำได้อย่างไร?
- ต้องมาดูเงื่อนไขอีกที



ดูปัญหาอีกครั้ง (ต่อ)

- เปลี่ยนเงื่อนไขเป็นการเปรียบเทียบแบบตรรกะ
- ถ้าได้เลขในช่วง 45-55 จะได้รางวัลที่ 1
 - $45 \leq x \leq 55$, เมื่อเปลี่ยนให้ใช้กับภาษา python ได้ จะได้
 - $45 \leq x$ and $x \leq 55$
- แต่ถ้าได้เลขในช่วง 15-30 หรือ 75-90 จะได้รางวัลที่ 2
 - $15 \leq x \leq 30$ หรือ $75 \leq x \leq 90$
 - $(15 \leq x \text{ and } x \leq 30) \text{ or } (75 \leq x \text{ and } x \leq 90)$

Flowchart ใหม่

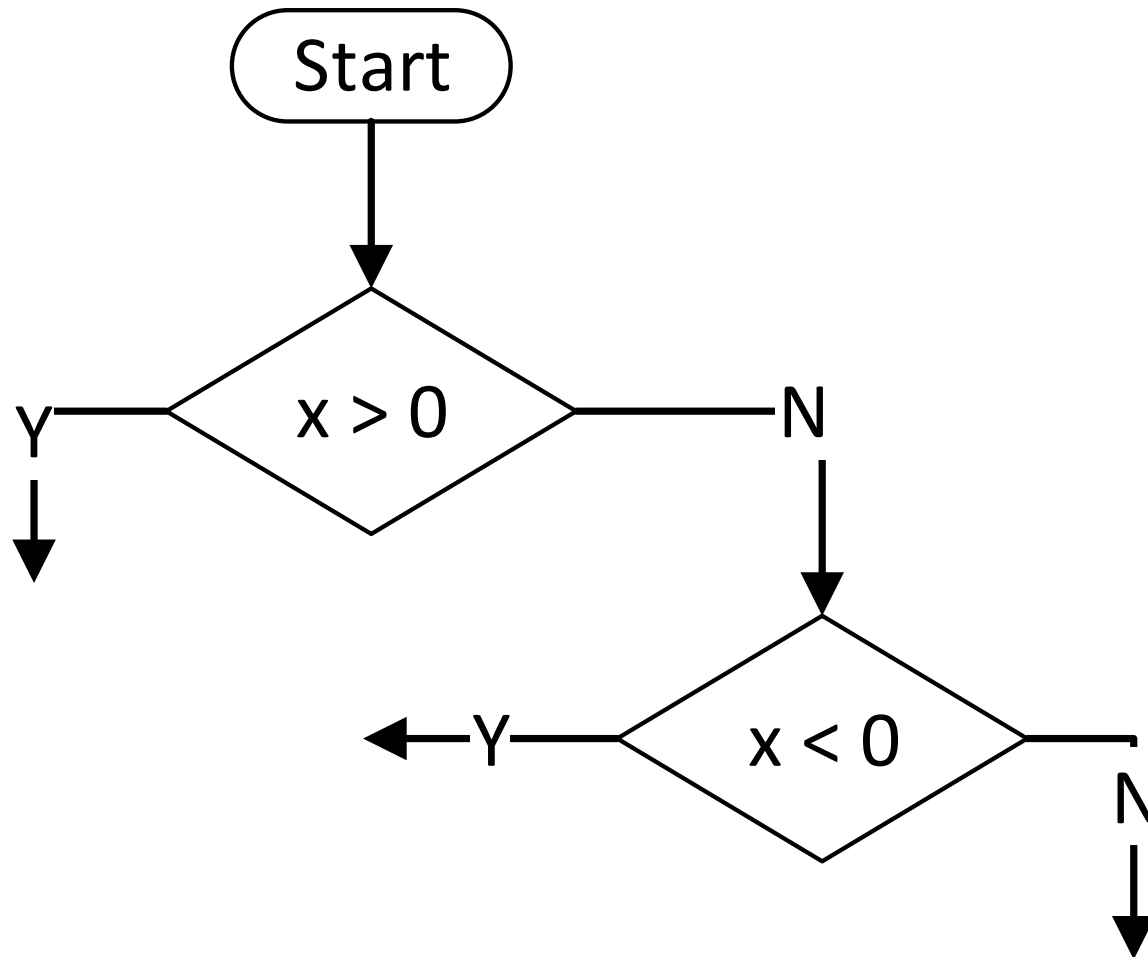


ถ้ามีการตัดสินใจที่ใช้การเปรียบเทียบมากกว่าหนึ่งครั้ง

- จงรับเลขจำนวนเต็มเข้ามา 1 จำนวน แล้ว:
 - ถ้า ตัวเลขมีค่ามากกว่าศูนย์
 - แล้วหารสองลงตัวให้ แสดงค่าตัวเลขออกมาทันที
 - แต่ถ้าหารสองไม่ลงตัว ให้บวกหนึ่งเข้ากับตัวเลขแล้วแสดงค่าใหม่
 - แต่ถ้า ตัวเลขมีค่าน้อยกว่าศูนย์
 - แล้วหารสองไม่ลงตัวให้ แสดงค่าตัวเลขออกมาทันที
 - แต่ถ้าหารสองลงตัว ให้บวกหนึ่งเข้ากับตัวเลขแล้วแสดงค่าใหม่
 - ในกรณีที่ไม่ใช่ทั้งคู่ ให้แสดงค่าตัวเลขออกมาทันที
- จะเขียน flowchart อย่างไร

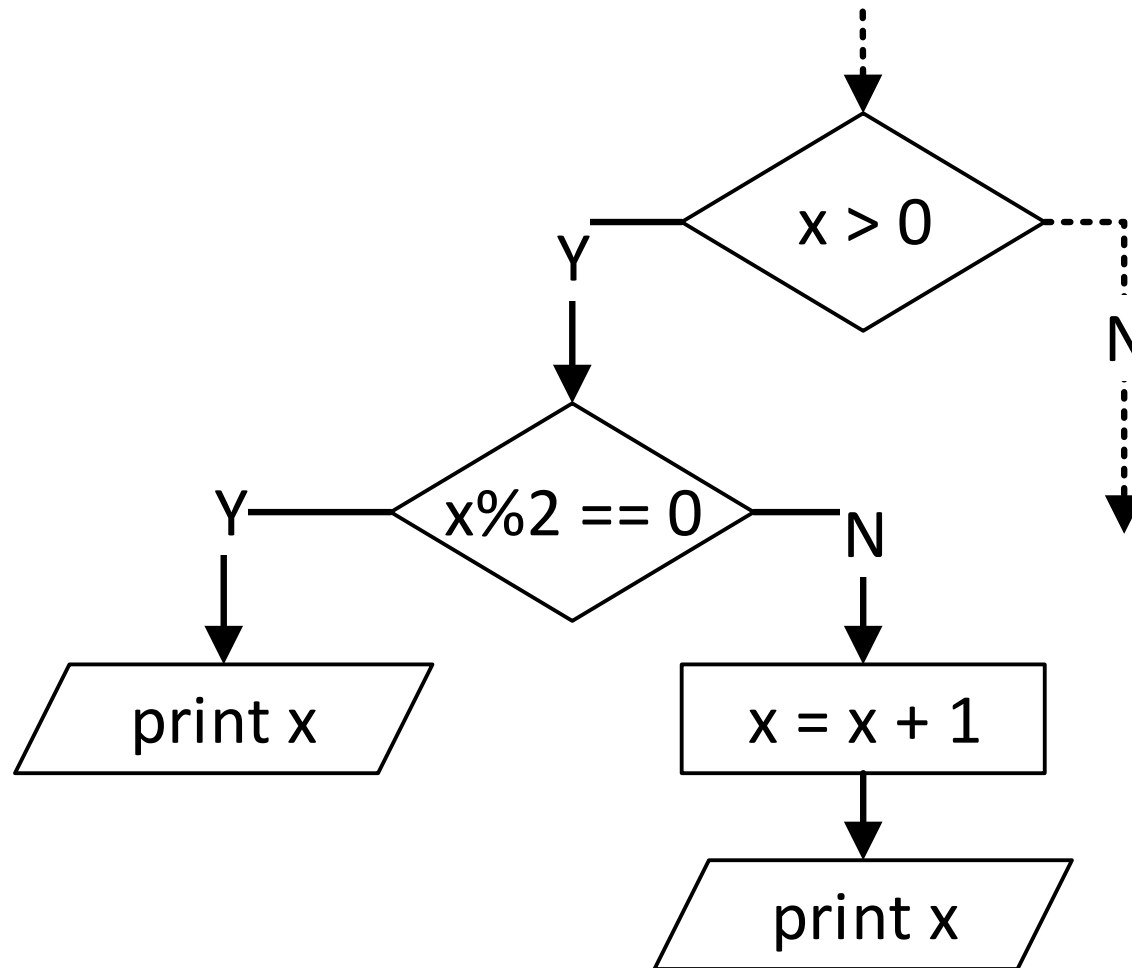
การเขียน Flowchart

1. เขียนเงื่อนไขส่วนที่จะต้องทำการเปรียบเทียบก่อน

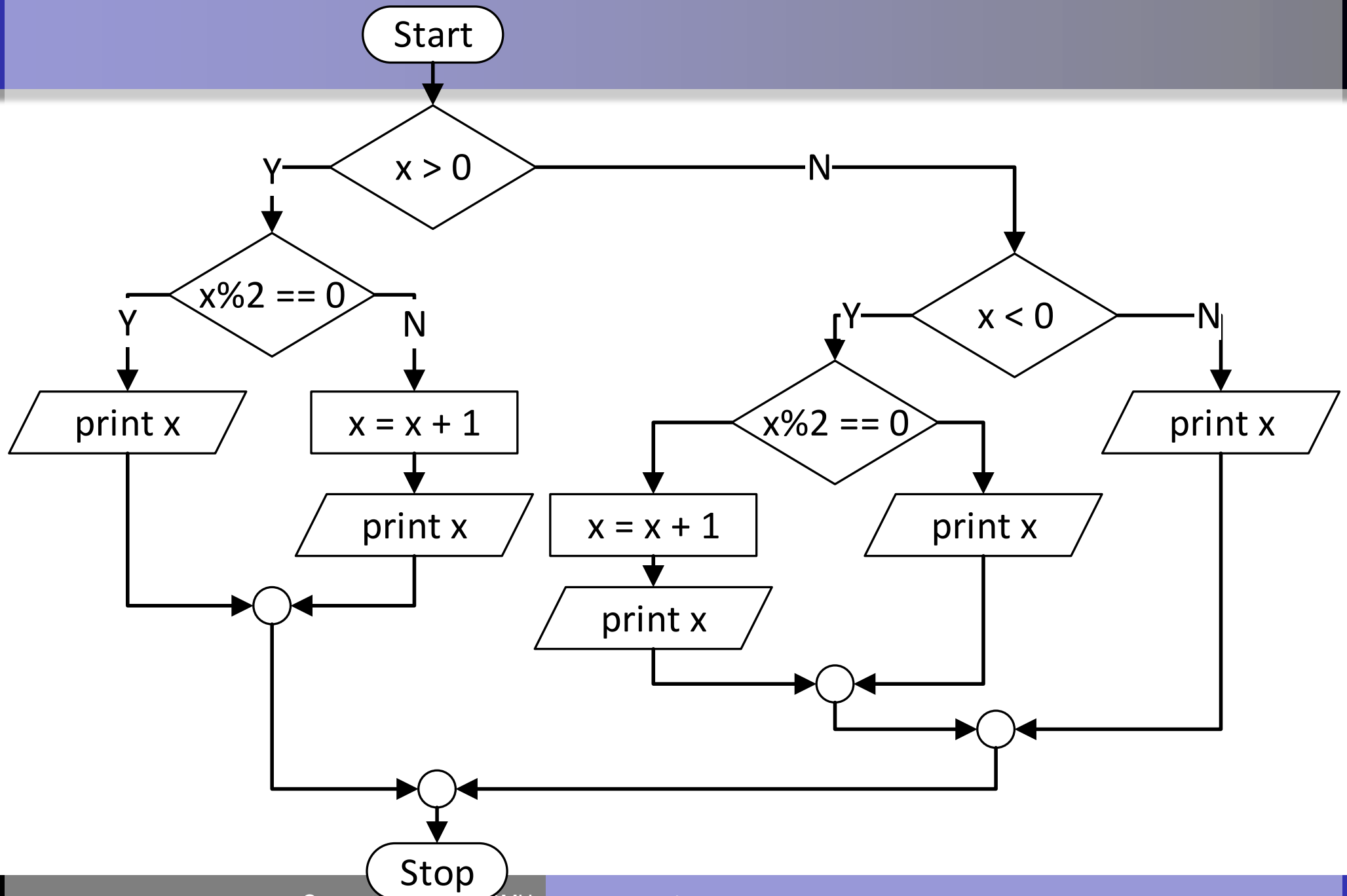


การเขียน Flowchart (ต่อ)

2. แล้วค่อยเขียนการเปรียบเทียบตัวถัดมา



Flowchart



การจัด selection blocks ต้องดูว่า

- เงื่อนไขครบถ้วนทุกกรณีหรือไม่
- แต่ละกรณี เป็นไปตามที่เราต้องการจริง หรือไม่
- สามารถลดความซับซ้อนได้หรือไม่
 - ตัวแปร
 - ลำดับ
 - ตัวเชื่อม (and or)

- “เมื่อไหร่จะถึงสิ้นเดือน?” (ปีนี้)
- Input: ???
- Output: ???
- Process: ???

แบบฝึกหัด 1

- “เมื่อไหร่จะถึงสิ้นเดือน?”
- Input: เลขจำนวนเต็มสองตัวแสดงวันที่ (day: 1-31) และเดือน (month: 1-12) ในปัจจุบัน
- Output: ตัวเลขแสดงจำนวนวันที่เหลือ ก่อนจะถึงสิ้นเดือน
- Process:

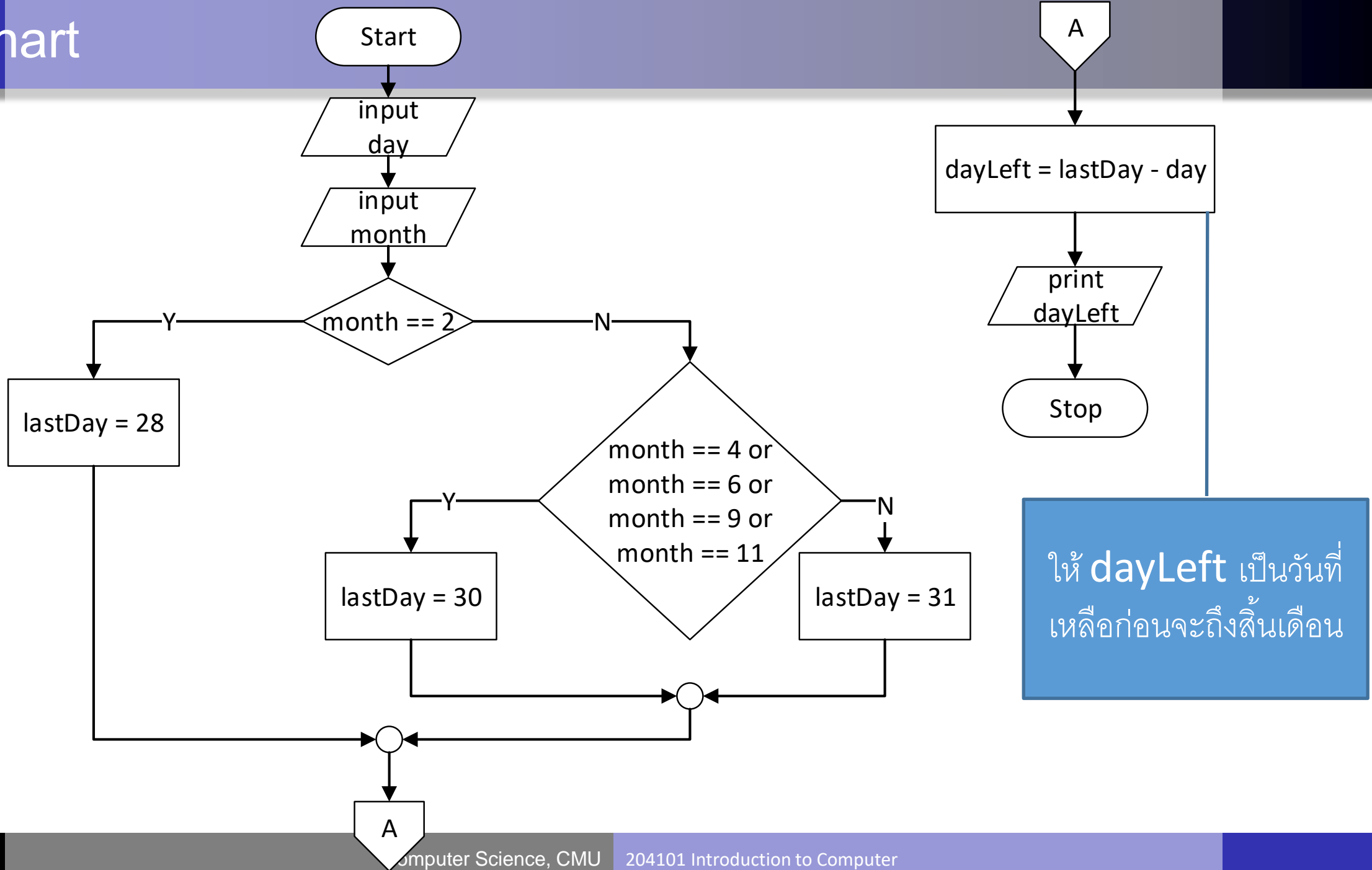
จะหาวันสุดท้ายของเดือนจากไหน?

หาวันที่เหลือ จาก วันที่เหลือ = วันสุดท้ายของเดือน - วันในปัจจุบัน

การเขียน Flowchart

- ค่าวันสุดท้ายของเดือน ขึ้นอยู่กับค่าเดือนที่ได้
 - ให้ lastDay เป็นวันสุดท้ายของเดือน
 - เดือน 1,3,5,7,8,10,12 lastDay = 31
 - เดือน 4,6,9,11 lastDay = 30
 - เดือน 2 lastDay = 28 (ปีนี้ไม่มี 29 กุมภาพันธ์)
- เมื่อรับค่าเข้ามา จะต้องตรวจเดือนก่อน จึงจะคำนวณวันที่เหลือได้

Flowchart



แบบฝึกหัด 2

- Input Validation
- จากแบบฝึกหัดที่ 1 จะตรวจสอบอย่างไรว่าตัวเลขที่ป้อนเข้าเป็นเดือนและวันที่อยู่ในช่วงที่ถูกต้อง
- ในเพื่อความง่ายต่อการตรวจสอบ ควรป้อนข้อมูลวันที่หรือเดือนก่อน?

ก่อนจะเขียน flowchart

- ช่วงข้อมูลที่ต้องการ
 - month: 1-12
 - day: 1-lastDay จะต้องรู้ lastDay ก่อน
- ดังนั้น จะต้องตรวจเดือนก่อน แล้วจึงจะตรวจวันได้
- สามารถแทรกการตรวจสอบ เข้าไปในการทำงานได้เลย

Flowchart

