

การแสดงขั้นตอนวิธีด้วยผังงานหรือรหัสเทียม

Flowchart

Nested Selection

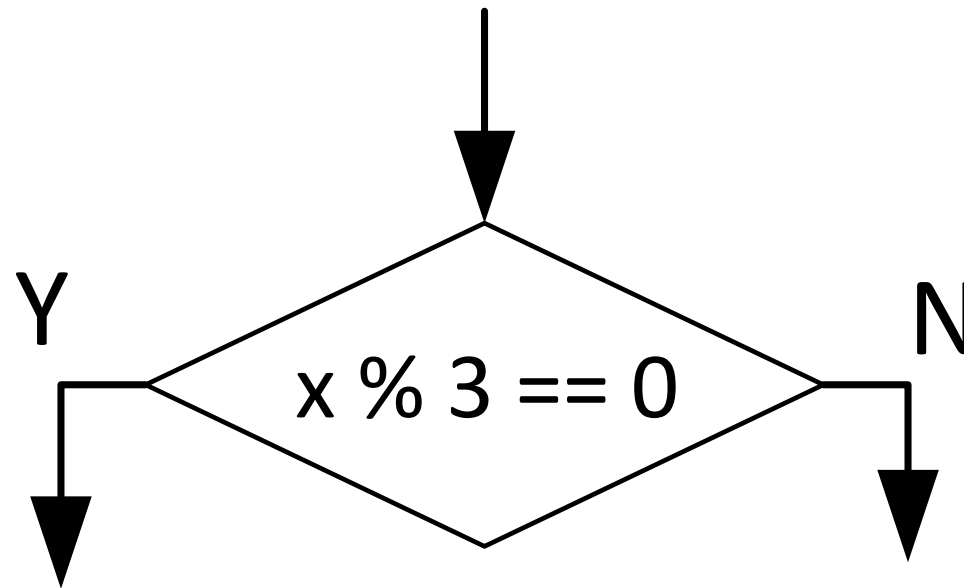
เมื่อ decision ตัวเดียวมีทางเลือกมากกว่า 2 ทาง

- “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

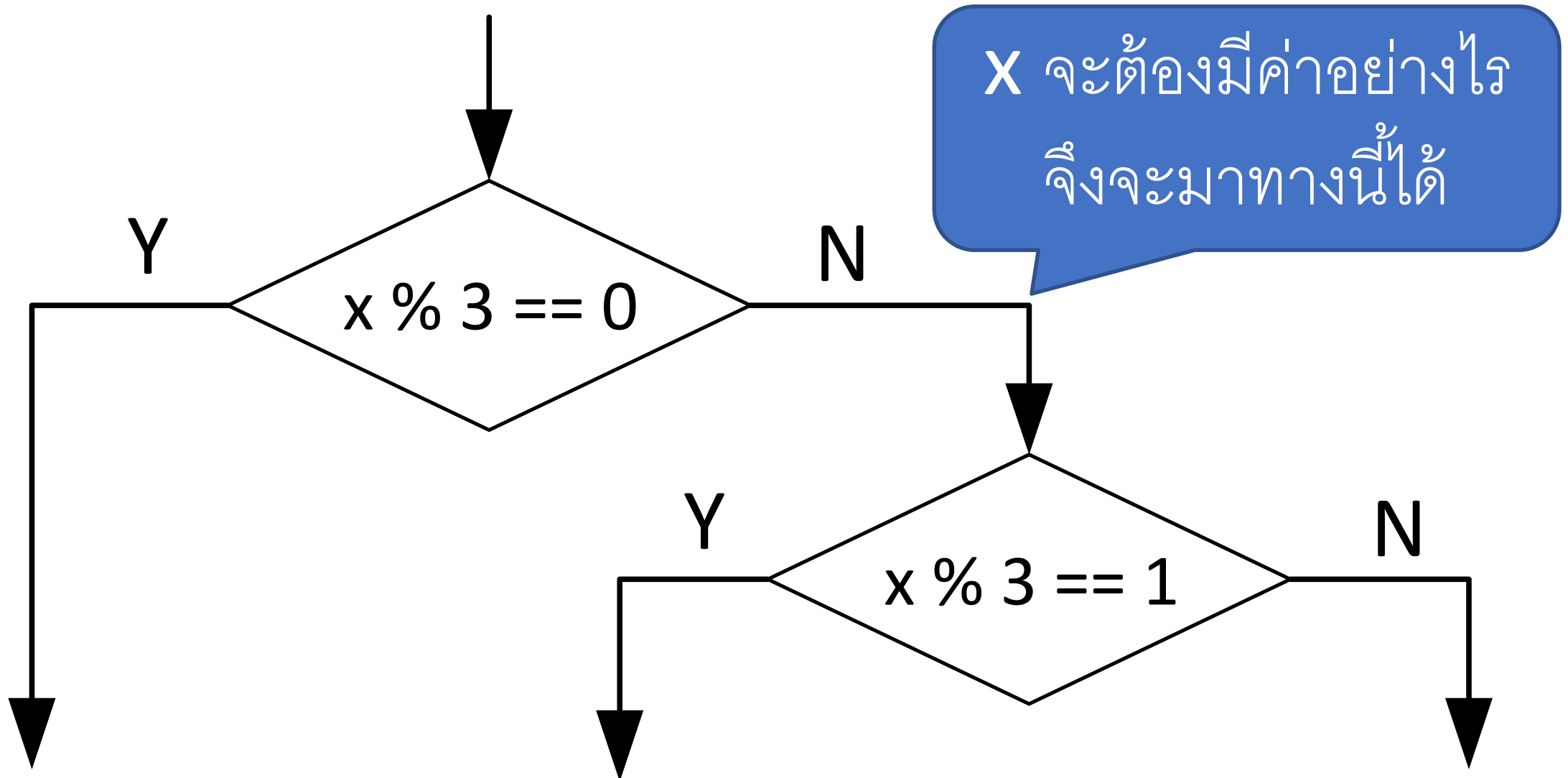
การเขียน Flowchart

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



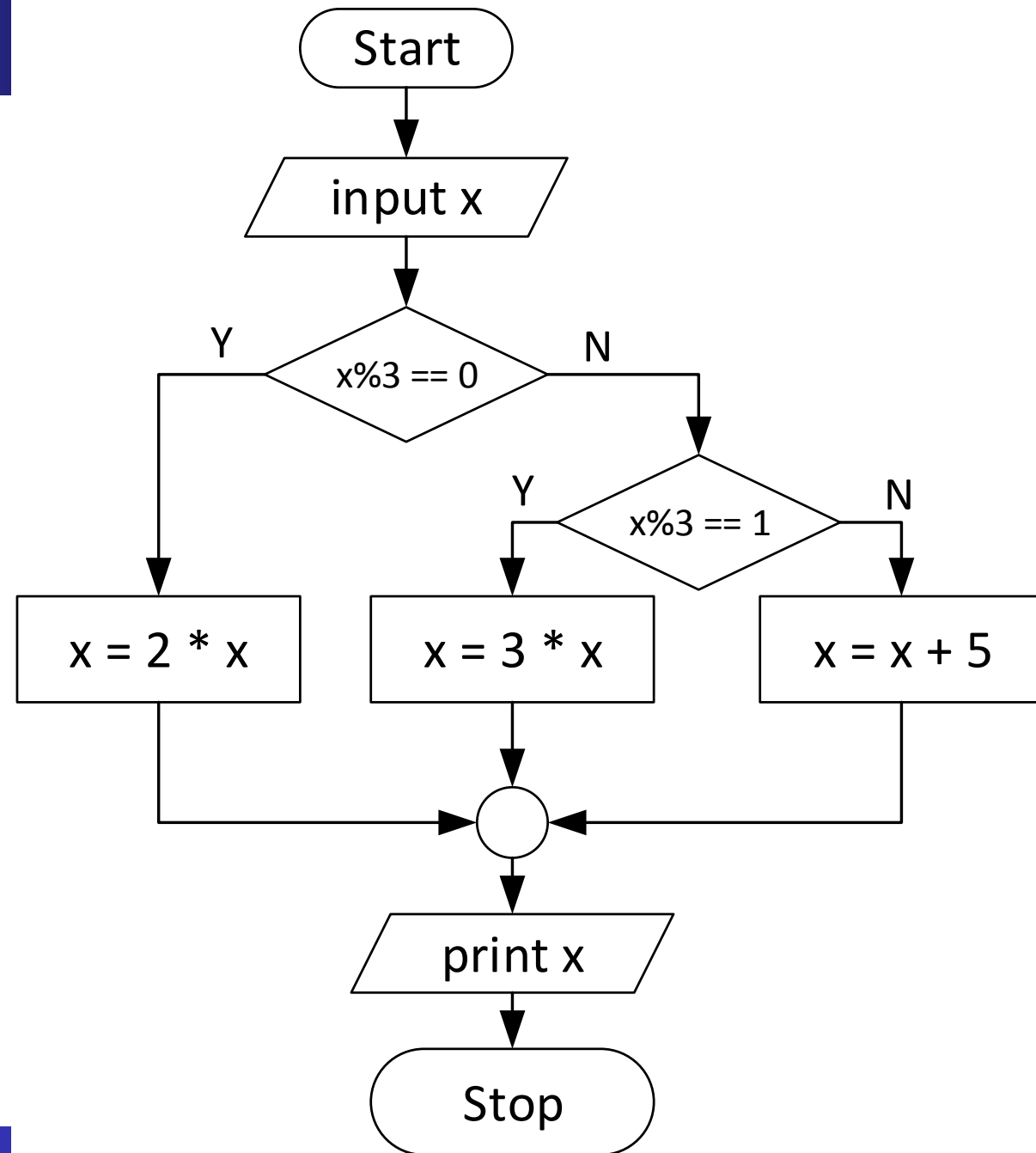
- ดังนั้น จะต้องใช้ selection มากกว่า หนึ่งตัว

การเขียน Flowchart (ต่อ)



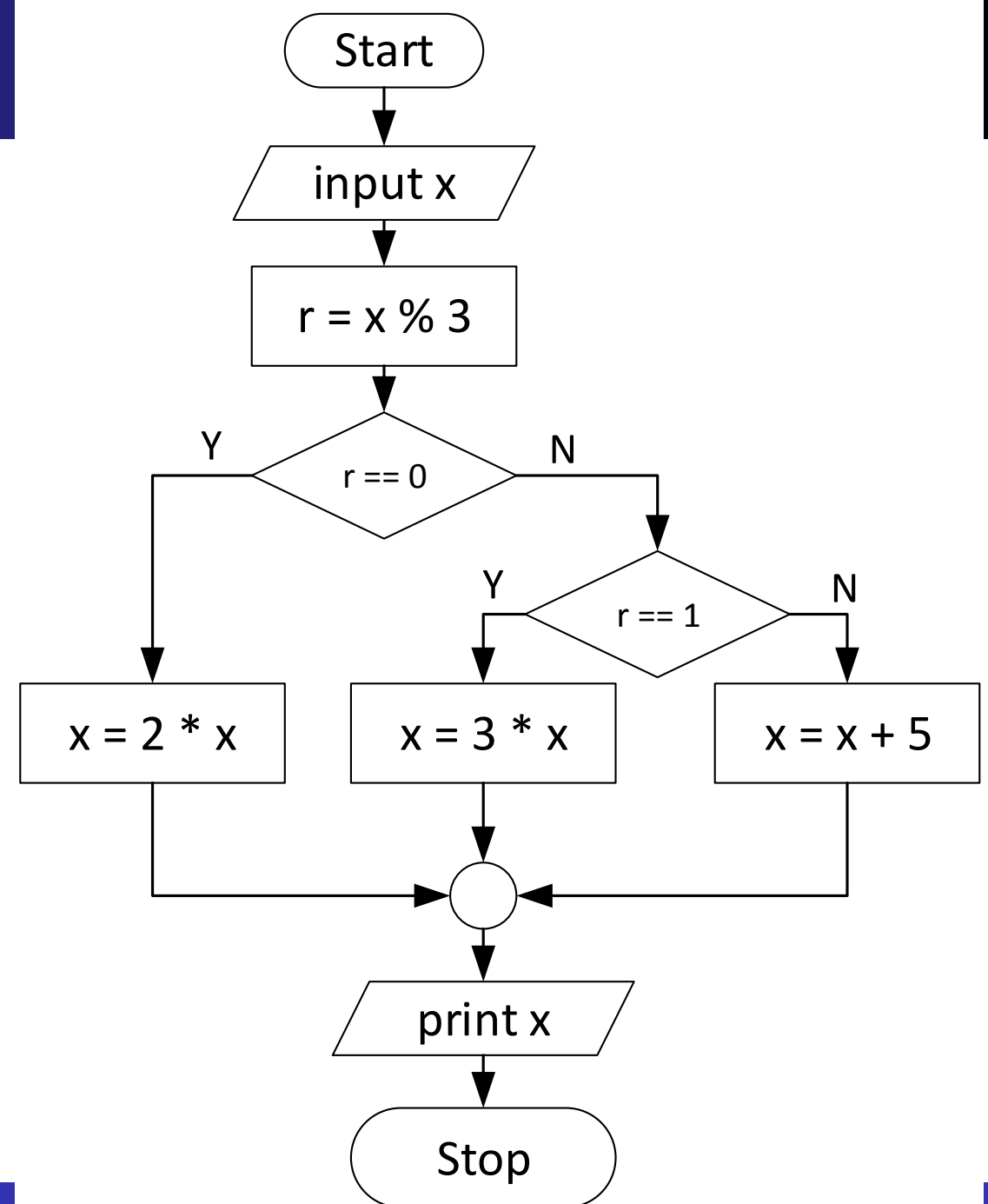
Flowchart #1

- 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

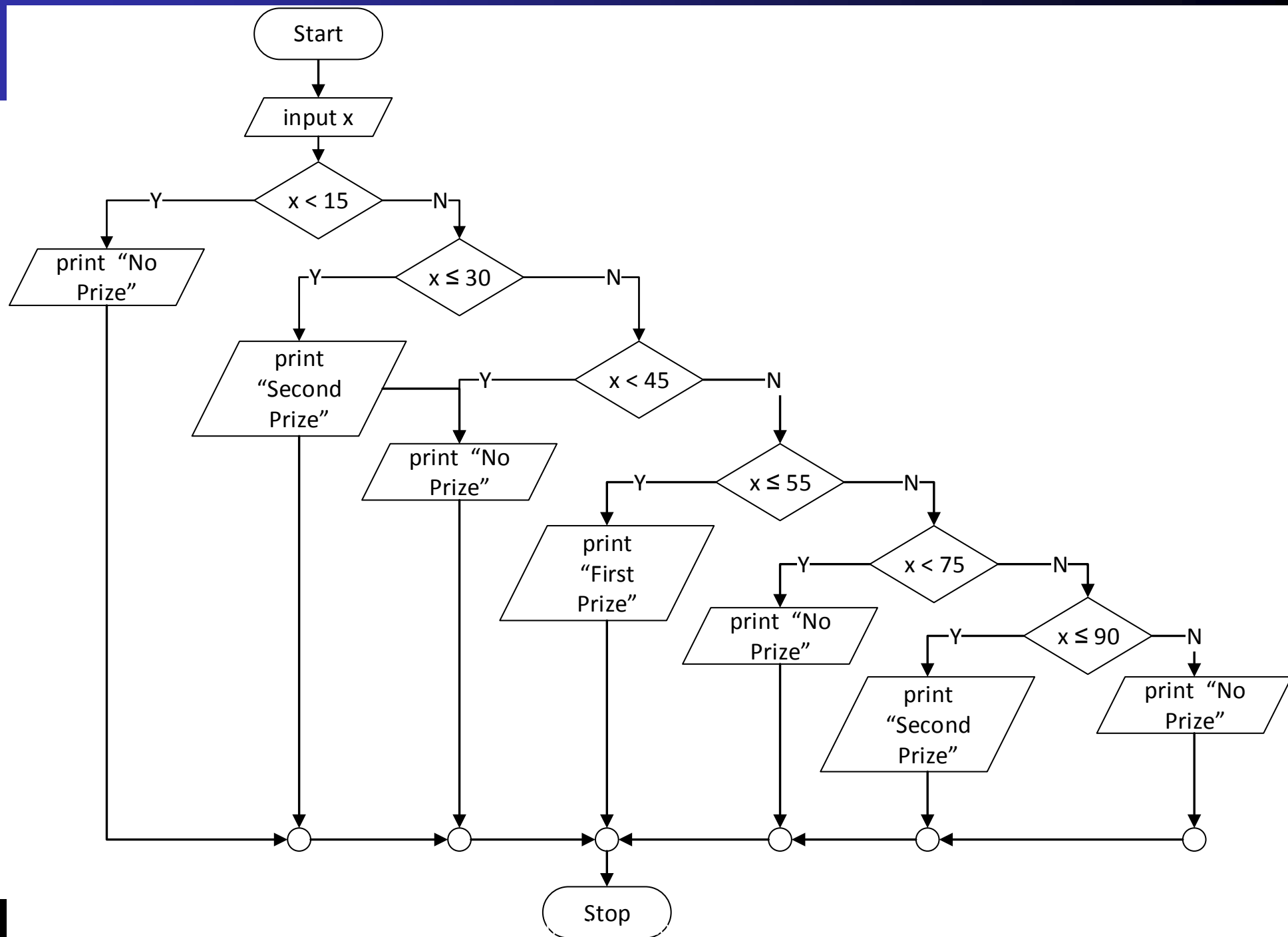
ตัวอย่าง

- 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

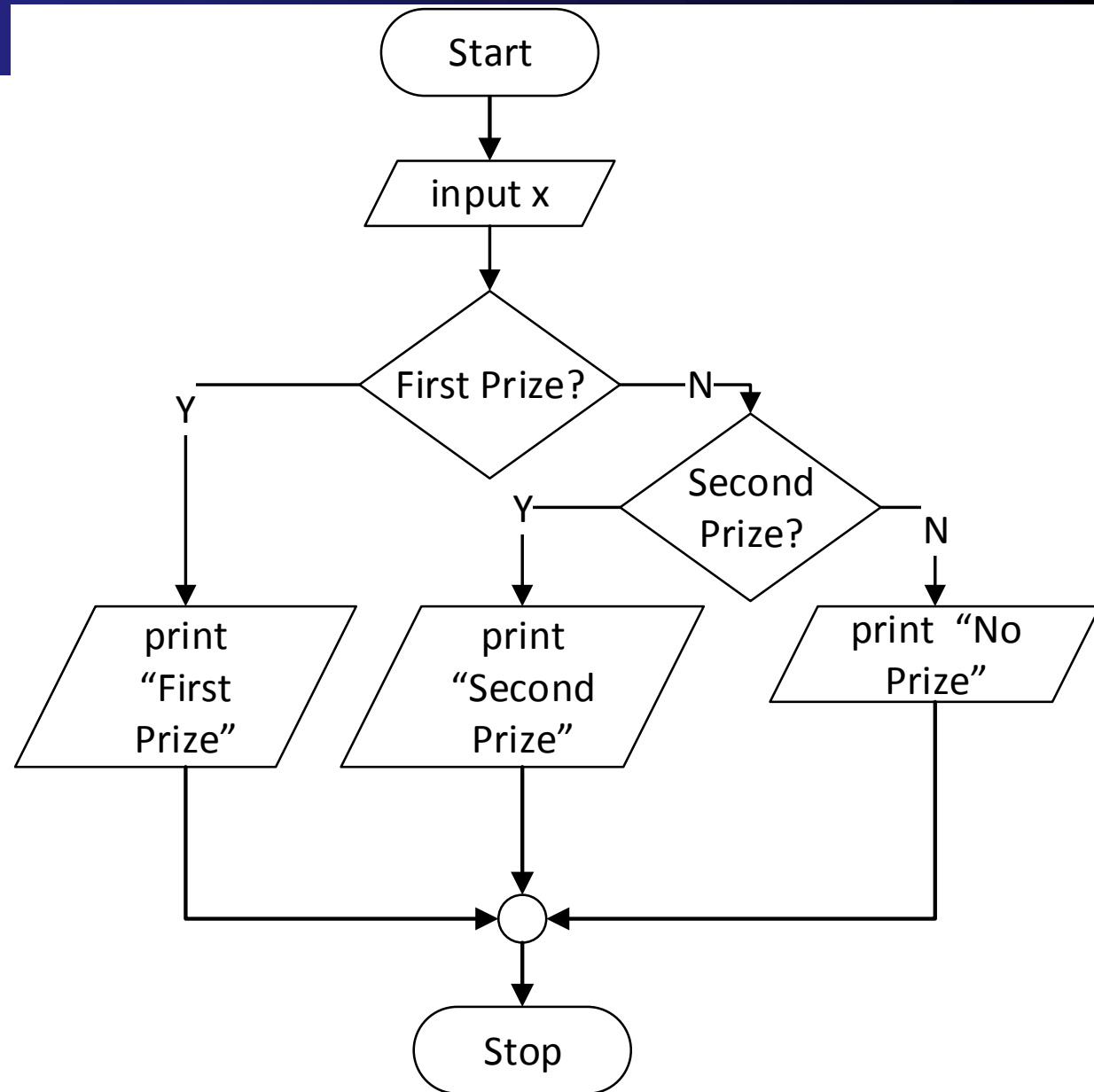
แบบ

ตรงไปตรงมา



ดูปัญหาอีกครั้ง

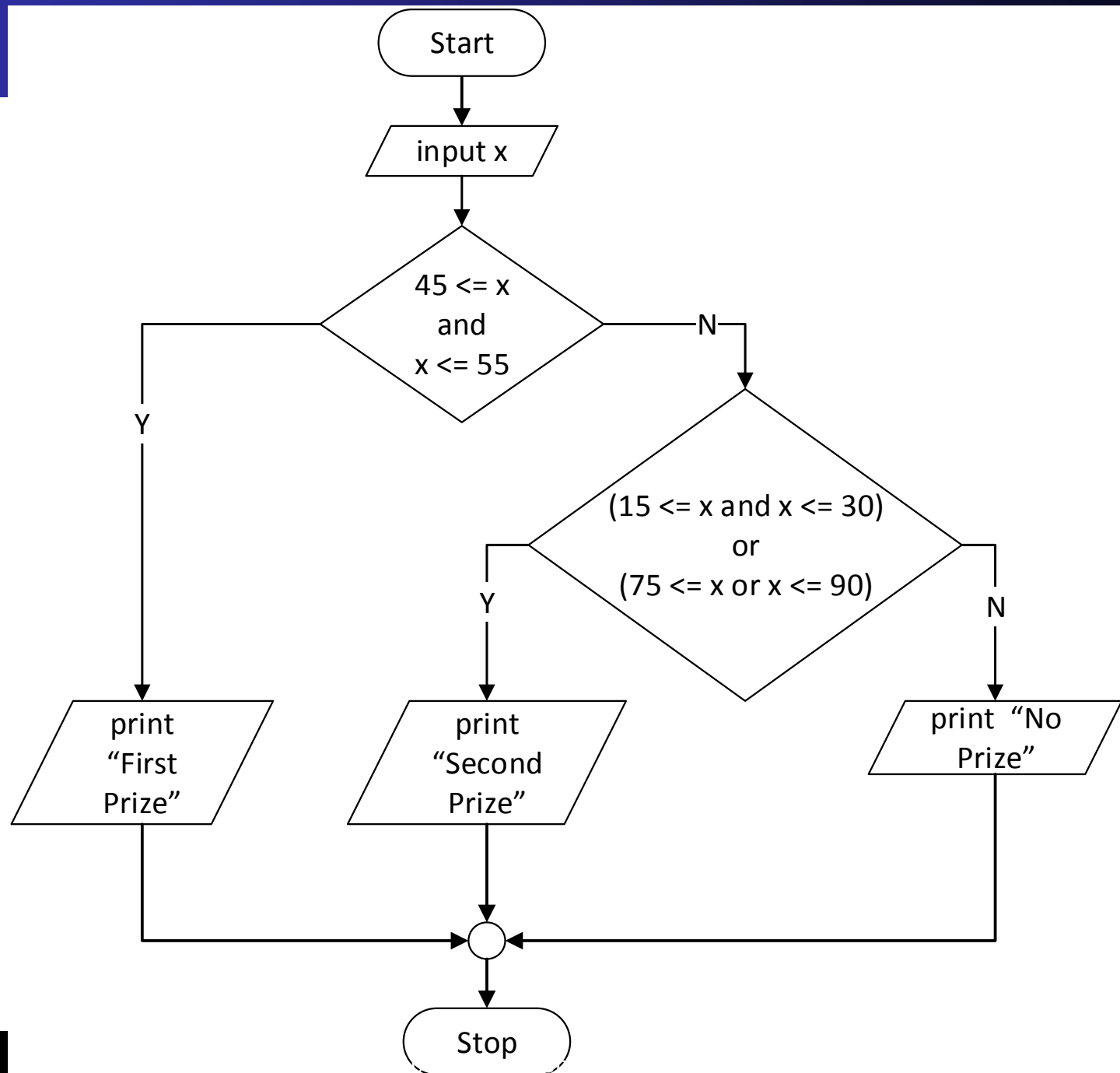
- จะซับซ้อนน้อยลงมาก ถ้าสามารถเขียนในรูปทางขวาได้
- จะทำได้อย่างไร?
- ต้องมาดูเงื่อนไขอีกที



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
 - $15 \leq x \leq 30$ และ $75 \leq x \leq 90$ หรือ
 - $(15 \leq x \text{ and } x \leq 30) \text{ or } (75 \leq x \text{ or } x \leq 90)$

Flowchart ใหม่

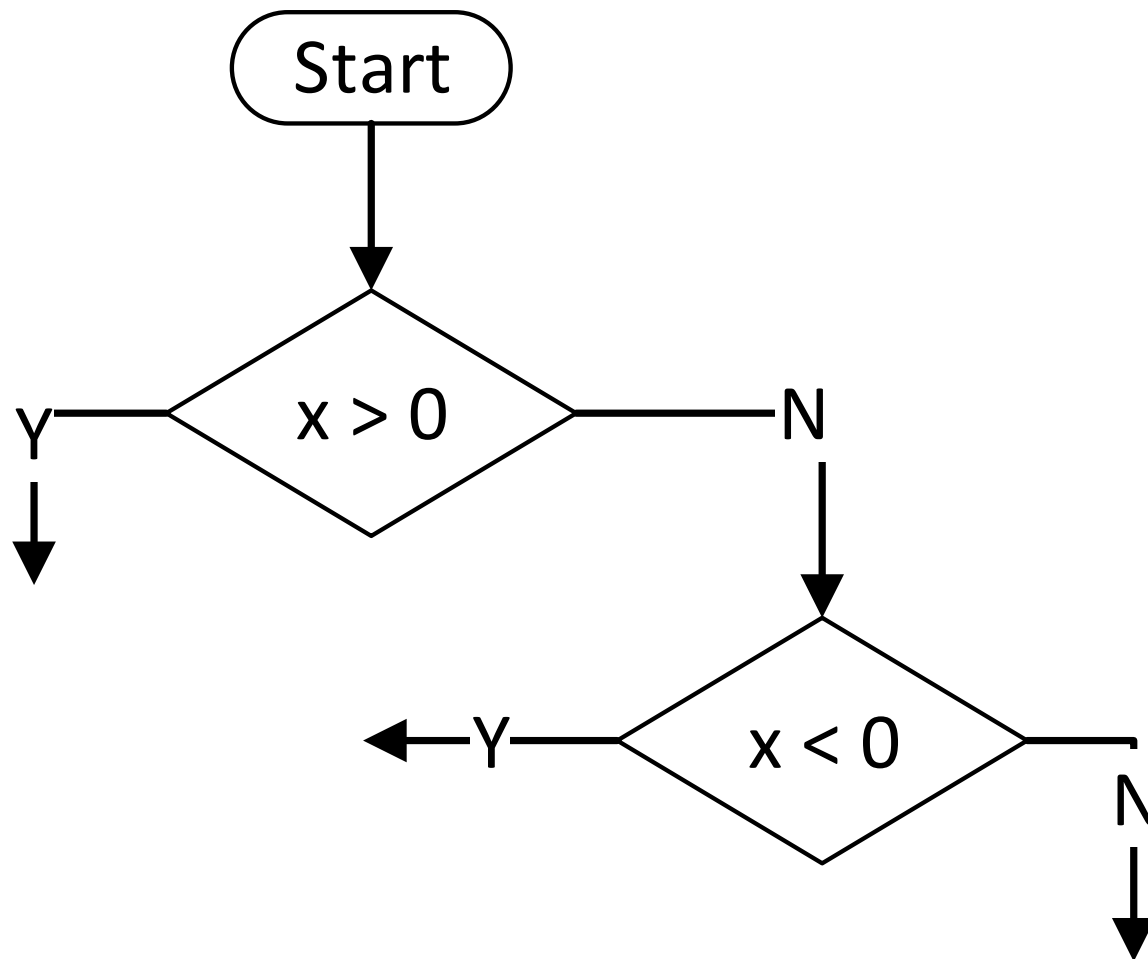


ถ้ามีการตัดสินใจที่ใช้การเปรียบเทียบมากกว่าหนึ่งครั้ง

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

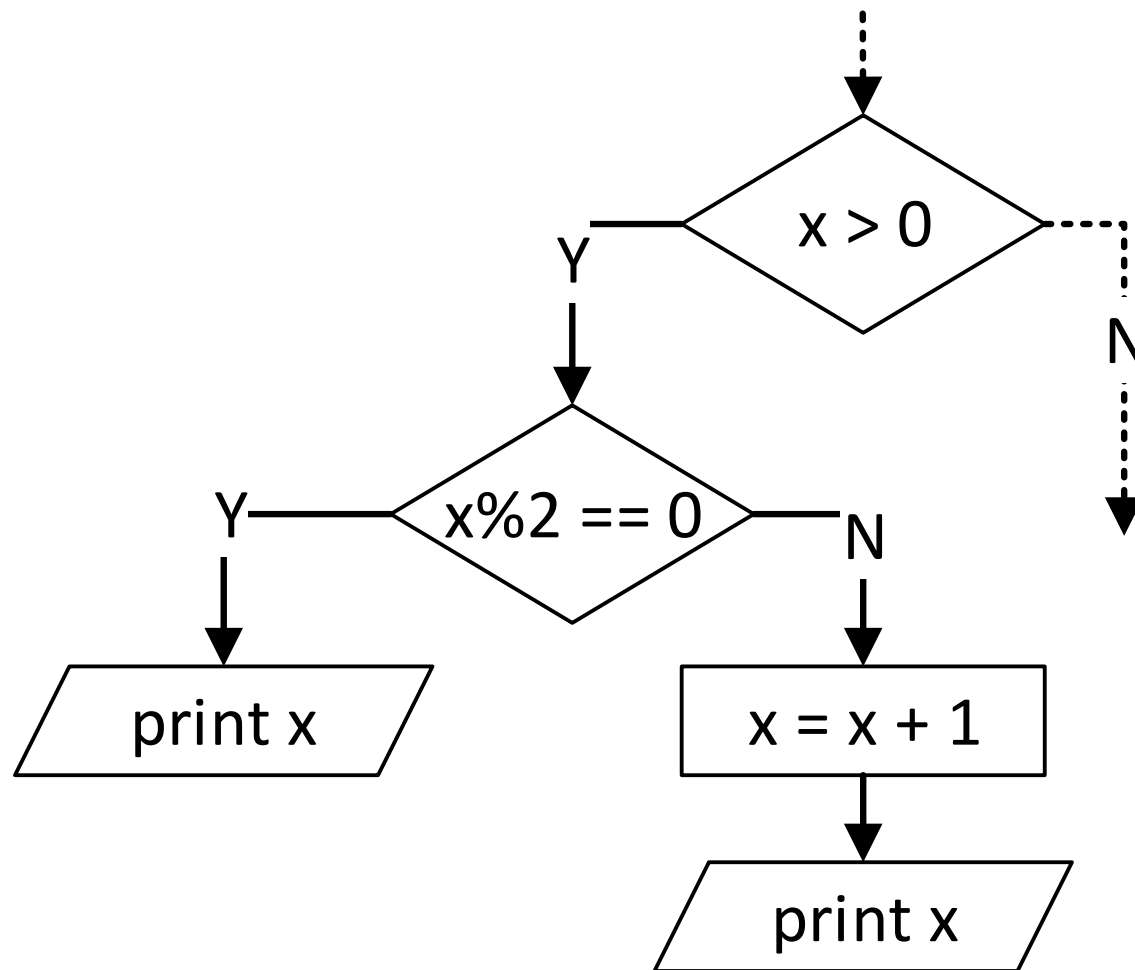
การเขียน Flowchart

1. เขียนเงื่อนไขส่วนที่จะต้องทำการเปรียบเทียบก่อน

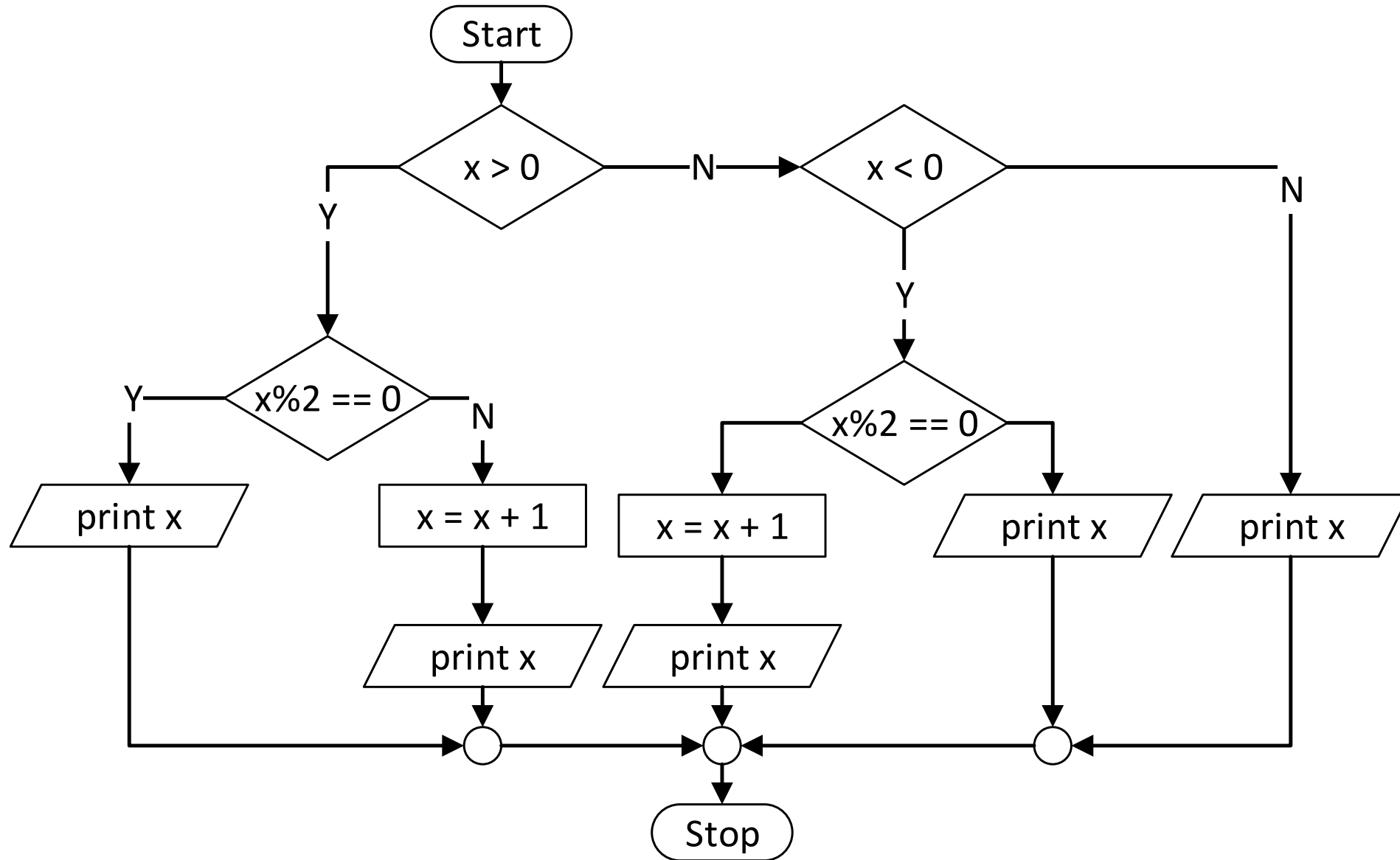


การเขียน Flowchart (ต่อ)

2. แล้วค่อยเขียนการเปรียบเทียบตัวถัดมา



Flowchart



การจัด selection blocks ต้องดูว่า

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

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

แบบฝึกหัด 2

- Input Validation
- จากแบบฝึกหัดที่ 1 จะตรวจสอบอย่างไรว่าตัวเลขที่ป้อนเข้าเป็นเดือนและวันที่อยู่ในช่วงที่ถูกต้อง
- ในเพื่อความง่ายต่อการตรวจสอบ ควรป้อนข้อมูลวันที่หรือเดือนก่อน?

แบบฝึกหัด 3

- รับตัวเลขสองหลักมา 1 จำนวน
 - นำหลักของตัวเลข มาบวกรวมกัน แล้วลบออกจากตัวเลขเดิม
 - ใช้ผลลบที่ได้ แสดงสัญลักษณ์ตามผลลบนั้น
- ตัวอย่าง

เลขตั้งต้น: 34

ผลลบ: $34 - (3+4) = 27$

แสดง: Ω

99	☐	79	♊	59	♊	39	♋	19	☐
98	♎	78	♎	58	♎	38	♌	18	♎
97	♊	77	♋	57	♌	37	♍	17	♍
96	♎	76	☐	56	♍	36	♎	16	♋
95	♋	75	☐	55	😊	35	♋	15	♎
94	♍	74	♍	54	♎	34	♎	14	♋
93	♌	73	☐	53	♏	33	♍	13	♎
92	☐	72	♎	52	♍	32	♍	12	♊
91	♋	71	♊	51	♌	31	♎	11	♎
90	♊	70	♍	50	☐	30	♎	10	♍
89	♏	69	♊	49	♎	29	♎	9	♎
88	♍	68	♌	48	♌	28	♎	8	♎
87	♎	67	♋	47	♋	27	♎	7	♋
86	♎	66	♎	46	♏	26	♋	6	♌
85	♎	65	♎	45	♎	25	♋	5	😊
84	♌	64	♏	44	♌	24	♊	4	♌
83	♎	63	♎	43	♎	23	♌	3	♋
82	♍	62	♋	42	♎	22	😊	2	♋
81	♎	61	♏	41	♌	21	😊	1	♌
80	♎	60	♏	40	♍	20	♏	0	♊