

Lab 4 - นิพจน์ตรรกศาสตร์ เงื่อนไข การทำซ้ำ

1. นิพจน์ทางตรรกศาสตร์

ผลลัพธ์ที่เกิดจากนิพจน์ทางตรรกศาสตร์มี 2 ค่า คือ 1.จริง (True) 2. เท็จ(False) เท่านั้น ในภาษา python เขียนนิพจน์ทางตรรกศาสตร์โดยอาศัยเครื่องหมายดังต่อไปนี้

เครื่องหมาย	ตัวอย่างการใช้งาน	ความหมาย
==	x == 6	ค่าในตัวแปร x เท่ากับ 6 จริงหรือไม่
!=	z != y	ค่าในตัวแปร z ไม่เท่ากับค่าในตัวแปร y จริงหรือไม่
>	x > 9	ค่าในตัวแปร x มากกว่า 9 จริงหรือไม่
>=	m >= n	ค่าในตัวแปร m มากกว่าหรือเท่ากับค่าในตัวแปร n จริงหรือไม่
<	w < 10	ค่าในตัวแปร w น้อยกว่า 10 จริงหรือไม่
<=	3 <= x	3 น้อยกว่าหรือเท่ากับค่าในตัวแปร x จริงหรือไม่

เครื่องหมายสำหรับเชื่อมนิพจน์ทางตรรกศาสตร์ที่มากกว่า 1 นิพจน์เข้าด้วยกันมีดังต่อไปนี้

เครื่องหมาย	สัญลักษณ์	ตัวอย่างการใช้งาน	ความหมาย
และ	and	(x > 1) and (x < 10)	จะมีค่าเป็นจริงเมื่อค่าในตัวแปร x มีค่าอยู่ระหว่าง 1 ถึง 10
หรือ	or	(x < 1) or (x > 10)	จะมีค่าเป็นจริงเมื่อค่าในตัวแปร x มีค่าน้อยกว่า 1 หรือมากกว่า 10
นิเสธ	not	not (x == 5)	จะมีค่าเป็นจริงเมื่อค่าในตัวแปร x มีค่าไม่เท่ากับ 5

แบบฝึกหัดที่ 1.1 : นิพจน์ทางตรรกศาสตร์ต่อไปนี้มีความหมายเช่นไร

ข้อที่	นิพจน์ทางตรรกศาสตร์ในรูปของภาษาไพทอน	ความหมาย
1	(z != 4) and (y <= 5)	
2	(a < 2) or ((a >= 3) and (a < 4))	
3	st != "male"	
4	(member == True) or (price >= 500)	
5	(ch >= "A") and (ch <= "Z")	
6	(x*x + y*y) == (z*z)	
7	x % 2 != 0	

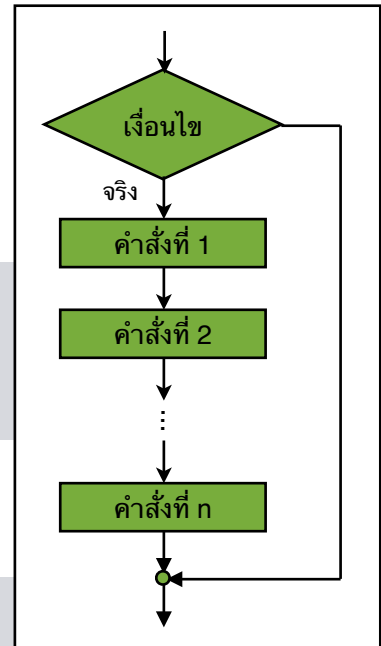
แบบฝึกหัดที่ 1.2 : จงแปลงข้อความต่อไปนี้เป็นนิพจน์ทางตรรกศาสตร์ในรูปแบบของภาษาไพทอน

ข้อที่	ข้อความ	นิพจน์ทางตรรกศาสตร์ในรูปของภาษาไพทอน
1	ค่าในตัวแปร y มากกว่า 30	y > 30
2	ค่าในตัวแปร x มากกว่า 15 และน้อยกว่า 50	
3	ค่าในตัวแปร m น้อยกว่าหรือเท่ากับ 10 หรือค่าในตัวแปร n มากกว่า 2	
4	ค่าในตัวแปร a หารด้วย 8 ลงตัว หรือหารด้วย 5 ลงตัว	

2. คำสั่งเงื่อนไข (If statements)

คำสั่ง if จะคำนวณค่าของนิพจน์ตรรกศาสตร์ที่เป็น เงื่อนไข (condition) โดยหากเงื่อนไขเป็นจริง โปรแกรมจะทำคำสั่งที่อยู่ในความควบคุมของ if นั้น ทั้งนี้ คำสั่งที่อยู่ในความควบคุมของ if นั้นสังเกตได้โดยคำสั่งจะถูกเขียนไว้ในตำแหน่งที่ขยับไปทางขวาจากตำแหน่งปกติของ if นั้น ๆ 1 ย่อหน้า (tab) ดังนี้

```
if (เงื่อนไข) :
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ 1
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ 2
    :
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ n
```



ตัวอย่างที่ 2.1 : หากต้องการให้โปรแกรมพิมพ์คำว่า Excellent ให้กับนักเรียนที่ทำการบ้านได้คะแนน (score) เต็ม 10 คะแนน สามารถใช้คำสั่ง if ได้ดังนี้

```
if (score == 10) :
    print("Excellent")
```

แบบฝึกหัดที่ 2.1 : ส่วนลดร้านชายขนม

ร้านชายขนมจะลดราคา 5% ให้กับคนที่ซื้อขนมมากกว่า 100 บาท ให้เขียนโปรแกรมรับราคาสินค้าจากนั้นคำนวณเงินที่ต้องจ่าย ให้แสดงผลเป็นทศนิยม 2 ตำแหน่ง

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>p = int(input("Enter total price: ")) if _____: _____ _____ print("You have to pay %0.2f" % p)</pre>	<p>ตัวอย่างที่ 1 Enter total price: 90 You have to pay 90.00</p> <hr/> <p>ตัวอย่างที่ 2 Enter total price: 101 You have to pay 95.95</p>

แบบฝึกหัดที่ 2.2 : ขนาดผลไม้

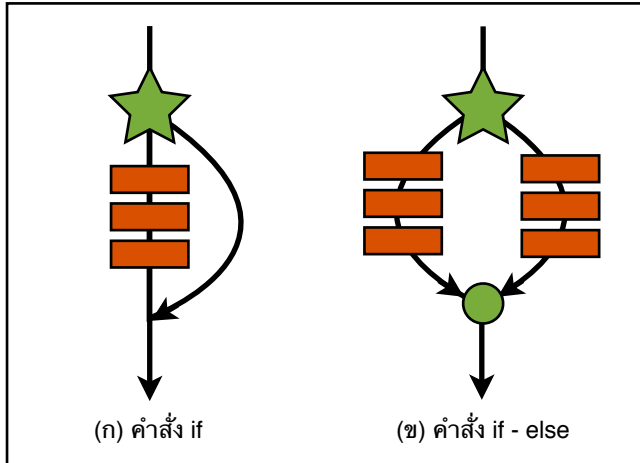
ลูกทุเรียนถูกแบบขนาดออกเป็น 3 กลุ่มตามน้ำหนัก ดังนี้

- น้อยกว่า 1 กิโลกรัม เรียกว่าเป็นขนาดเล็ก (small)
- มากกว่าหรือเท่ากับ 1 กิโลกรัม แต่น้อยกว่า 3 กิโลกรัม เรียกว่าเป็นขนาดกลาง (medium)
- มากกว่าหรือเท่ากับ 3 กิโลกรัม เรียกว่าเป็นขนาดใหญ่ (large)

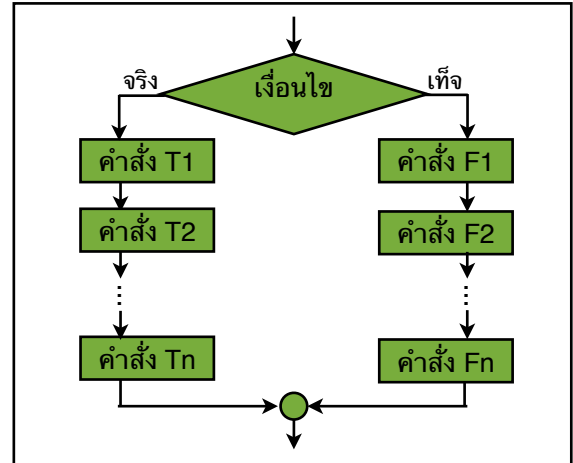
ให้เขียนโปรแกรมสำหรับคัดขนาดของลูกทุเรียน ดังตัวอย่าง

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>_____ if _____: _____ _____ if _____: _____ _____ if _____: _____ _____</pre>	<p>ตัวอย่างที่ 1 Enter weight: 1.5 It is medium.</p> <hr/> <p>ตัวอย่างที่ 2 Enter weight: 3 It is large.</p> <hr/> <p>ตัวอย่างที่ 3 Enter weight: 0.5 It is small.</p>

3. คำสั่งเงื่อนไขแบบ if - else (If - else statements)



ภาพที่ 3.1 : ลักษณะของคำสั่ง if และคำสั่ง if - else



ภาพที่ 3.2 : แผนภาพเส้นทางการทำงานของคำสั่ง if - else

คำสั่ง if - else จะต่างจากคำสั่ง if ทั่วไปตรงที่คำสั่ง if จะใช้ในการตัดสินใจว่าจะทำชุดคำสั่งกลุ่มหนึ่งหรือไม่ (ถ้าไม่ทำก็จะข้ามชุดคำสั่งนี้ไปทั้งหมดดังภาพที่ 3.1(ก)) แต่ในขณะที่คำสั่ง if - else จะใช้ในการตัดสินใจที่ต้องเลือกทางใดทางหนึ่งจากสองทาง (ดังภาพที่ 3.1(ข)) โดยหากเงื่อนไข (condition) เป็นจริง โปรแกรมจะทำคำสั่งที่อยู่ในความควบคุมของ if แต่หากเงื่อนไขเป็นเท็จ โปรแกรมจะทำคำสั่งที่อยู่ในความควบคุมของ else แทนที่ ดังภาพที่ 3.2

```

if (เงื่อนไข) :
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ 1 (T1)
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ 2 (T2)
    :
    คำสั่งที่อยู่ในความควบคุมของ if คำสั่งที่ n (Tn)
else :
    คำสั่งที่อยู่ในความควบคุมของ else คำสั่งที่ 1 (F1)
    คำสั่งที่อยู่ในความควบคุมของ else คำสั่งที่ 2 (F2)
    :
    คำสั่งที่อยู่ในความควบคุมของ else คำสั่งที่ n (Fn)
    
```

ตัวอย่างที่ 3.1 : หากต้องการให้โปรแกรมพิมพ์คำว่า Passed ให้กับนักเรียนที่ได้คะแนน (score) มากกว่าหรือเท่ากับ 50 คะแนน และพิมพ์คำว่า Failed ให้กับนักเรียนที่ได้คะแนนต่ำกว่า 50 คะแนนสามารถใช้คำสั่ง if ได้ดังนี้

```

if (score >= 50) :
    print("Passed")
if (score < 50) :
    print("Failed")
    
```

หรือเขียนโดยใช้คำสั่ง if - else ได้ดังนี้

```

if (score >= 50) :
    print("Passed")
else :
    print("Failed")
    
```

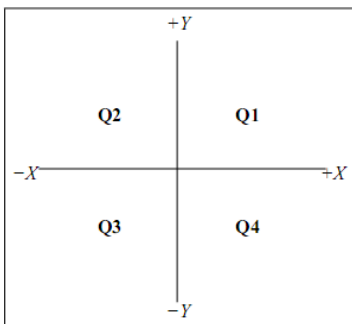
แบบฝึกหัดที่ 3.1 : เข้าประตู

ประตูปราสาทมีขอบด้านล่างสูง 40 เมตร ขอบด้านบนสูง 90 เมตร เด็กน้อยเตะลูกบอลสูง h เมตร อยากทราบว่าเตะเข้าประตูหรือไม่ (ถ้าลูกบอลสูงพอดีกับขอบให้ถือว่าไม่เข้า)

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>h = int(input("Enter height: ")) if _____ : _____ else : _____</pre>	<p>ตัวอย่างที่ 1 Enter height: 60 The ball went in.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter height: 100 The ball did not go in.</p>

แบบฝึกหัดที่ 3.2 : จุดในระนาบ(ง่าย)

ระนาบถูกแบ่งออกเป็น 4 ควอดแดรนต์ จุดจะอยู่ในควอดแดรนต์ที่หนึ่ง (Q1) ถ้าจุดไม่ได้อยู่บนแกน x หรือแกน y และมีพิกัดในทั้งสองแกนเป็นบวกทั้งหมด จงเขียนโปรแกรมรับพิกัดของจุด แล้วระบุว่าจุดดังกล่าวอยู่ใน Q1 หรือไม่



โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>x = float(input("Enter X: ")) y = _____ if _____ : _____ else: print("The point is not in Q1.")</pre>	<p>ตัวอย่างที่ 1 Enter X: 10 Enter Y: 5 The point is in Q1.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter X: -9 Enter Y: 0 The point is not in Q1.</p>

แบบฝึกหัดที่ 3.3 : พหุคูณบวกของ 5

จงเขียนโปรแกรมเพื่อตรวจสอบว่าตัวเลขที่ input เข้ามานั้นเป็นจำนวนเต็มบวกที่หารด้วย 5 ลงตัวหรือไม่ดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter number: 40 40 is positive multiple of 5.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter number: -8 -8 is not positive multiple of 5.</p> <p>_____</p> <p>ตัวอย่างที่ 3 Enter number: 12 12 is not positive multiple of 5.</p>

4. คำสั่งเงื่อนไขที่ซ้อนกัน (Nested If)

ในโปรแกรมบางโปรแกรมที่มีความซับซ้อนยิ่งขึ้นคำสั่งควบคุมที่อยู่ในคำสั่ง if ตัวหนึ่งก็อาจเป็นคำสั่ง if อีกตัวหนึ่งได้เช่นเดียวกัน เช่นหากเราตรวจสอบว่าจำนวนที่เก็บอยู่ไหนตัวแปร n เป็นจำนวนเต็มบวก เต็มลบ หรือเต็มศูนย์ ได้ตั้งตัวอย่างโปรแกรมที่ 4.1 หรือ 4.2 ดังนี้

	ตัวอย่างโปรแกรมที่ 4.1	ตัวอย่างโปรแกรมที่ 4.2
1	if (n >= 0):	if (n > 0):
2	if (n > 0):	print("It's positive number")
3	print("It's positive number")	else :
4	else :	if (n<0):
5	print("It's zero")	print("It's negative number")
6	else :	else:
7	print("It's negative number")	print("It's zero")

แบบฝึกหัดที่ 4.1 : ร้านหนังสือ SE-ED

ร้านหนังสือ SE-ED มอบส่วนลดพิเศษให้กับสมาชิก โดยถ้าสมาชิกซื้อหนังสือจากสำนักพิมพ์ SE-ED จะมอบส่วนลดให้ 10% และหากเป็นสำนักพิมพ์อื่นมอบส่วนลดให้ 5% จงเขียนโปรแกรมที่ใช้ในการคำนวณหาราคาสินค้าที่ลูกค้าต้องจ่ายในการซื้อหนังสือหนึ่งเล่มดังตัวอย่าง

โปรแกรมที่กำหนดให้ (เดิมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre> member = input("Do you be a member? ") pub = input("Name of Publishers : ") price = float(input("Price of Book : ")) if _____: if _____: print("Your price is", price*0.9) _____ print("Your price is", price*0.95) _____ print("Your price is", price) </pre>	<p>ตัวอย่างที่ 1 Do you be a member? Yes Name of Publishers : SE-ED Price of Book : 200 Your price is 180.0</p> <hr/> <p>ตัวอย่างที่ 2 Do you be a member? Yes Name of Publishers : JAMSAI Price of Book : 150 Your price is 142.5</p> <hr/> <p>ตัวอย่างที่ 3 Do you be a member? No Name of Publishers : NAMEE Price of Book : 175 Your price is 175.0</p>

5. คำสั่งเงื่อนไขแบบ elif

จากหัวข้อที่ 4 การใช้คำสั่ง if แบบซ้ำซ้อนในกรณีของ else ตามด้วย if นั้น สามารถเขียนย่อรวมได้เป็น elif ดังเช่นคำสั่ง else ตามด้วย if ในบรรทัดที่ 3 - 4 ของโปรแกรมในตัวอย่างที่ 4.2 สามารถเขียนได้โดยใช้ elif ได้ดังตัวอย่างโปรแกรมที่ 5.1

โปรแกรมตัวอย่างที่ 5.1
<pre> if (n > 0): print("It's positive number") elif (n<0): print("It's negative number") else: print("It's zero") </pre>

ทั้งนี้คำสั่ง elif นี้จึงเหมาะสมกับโปรแกรมที่ต้องการแยกทำงานหลาย ๆ กรณีดังตัวอย่างที่ 5.2

ตัวอย่างที่ 5.2 : เกรดวิชาหนึ่งตัดคะแนน (score) ตามเกณฑ์ดังตารางที่ 5.2 สามารถเขียนโปรแกรมในส่วนของแสดงเกรดได้ดังตัวอย่างโปรแกรมที่ 5.2

คะแนน	เกรด
score \geq 80	A
70 \leq score < 80	B
60 \leq score < 70	C
50 \leq score < 60	D
score < 50	F

ตารางที่ 5.2 : เกณฑ์การตัดเกรดของวิชาหนึ่ง

ตัวอย่างโปรแกรมที่ 5.2
<pre> if (score >= 80) : print("A") elif (score >= 70) : print("B") elif (score >= 60) : print("C") elif (score >= 50) : print("D") else : print("F") </pre>

แบบฝึกหัดที่ 5.1 : กลุ่มของตัวอักษร

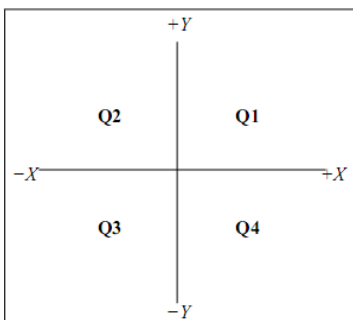
จงเขียนโปรแกรมสำหรับตรวจสอบว่าตัวอักษรที่ผู้ใช้ป้อนเข้ามานั้นเป็นกลุ่มตัวอักษรพิมพ์เล็ก (Lower case letter) กลุ่มอักษรพิมพ์ใหญ่ (Upper case letter) กลุ่มตัวเลข (Digit) หรือกลุ่มอื่น ๆ (Other)

*หมายเหตุ การตรวจสอบว่าอักษรในตัวแปร ch เป็นกลุ่มอักษรพิมพ์ใหญ่หรือไม่ สามารถใช้คำสั่ง `if((ch >= "A") and (ch <= "Z"))`

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter char: <u>p</u> Output: Lower case letter.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter char: <u>5</u> Output: Digit.</p> <p>_____</p> <p>ตัวอย่างที่ 3 Enter char: <u>\$</u> Output: Other.</p>

แบบฝึกหัดที่ 5.3 : จุดในระนาบ (ยาก)

ให้นักเรียนเขียนโปรแกรมที่รับจุด (x,y) บนระนาบ จากนั้นตรวจสอบว่าจุดดังกล่าว อยู่ที่ควอดแดรนต์ใด หรือว่าอยู่บนแกน X, แกน Y หรือ อยู่ที่จุดกำเนิด (Origin point) รูปของควอดแดรนต์ต่าง ๆ แสดงด้านล่าง



โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 Enter X: <u>10</u> Enter Y: <u>5</u> The point is in Q1.</p> <p>_____</p> <p>ตัวอย่างที่ 2 Enter X: <u>-9</u> Enter Y: <u>0</u> The point is on the X axis.</p> <p>_____</p> <p>ตัวอย่างที่ 3 Enter X: <u>0</u> Enter Y: <u>0</u> The point is at the origin.</p>

แบบฝึกหัดที่ 5.4 : รายได้

บริษัทเล็กๆ แห่งหนึ่งต้องการโปรแกรมคำนวณหาเงินที่ต้องจ่ายให้แก่พนักงาน โดยมีรายละเอียดดังต่อไปนี้

- INPUT - ชื่อพนักงาน และจำนวนชั่วโมงที่ทำงานใน 1 สัปดาห์
- OUTPUT - จำนวนเงินที่ต้องจ่ายให้แก่พนักงาน (บาท)

หลักการคิดเงินที่ต้องจ่ายให้แก่พนักงาน

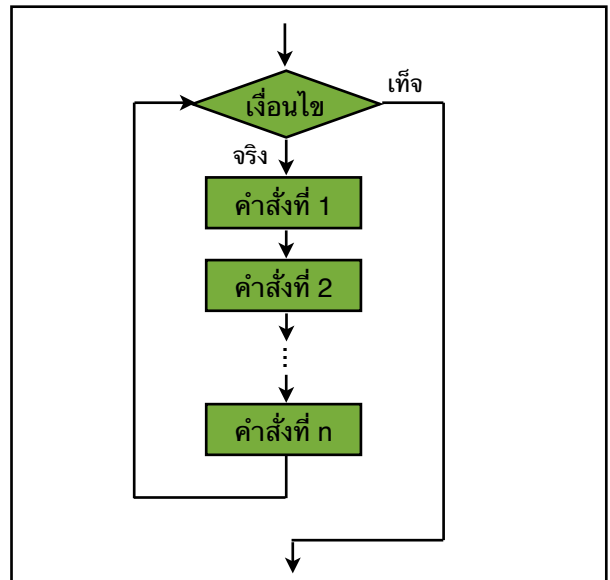
- 0 < จำนวนชั่วโมง <= 40 : รายได้พนักงาน = 1200 บาท
- จำนวนชั่วโมง > 40 : รายได้พนักงาน = 1200 + (จำนวนชั่วโมง - 40) x 55 บาท
- ถ้าจำนวนชั่วโมงไม่อยู่ในขอบเขตข้างต้น ให้พิมพ์ว่า INVALID WORK HOUR

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<p>ตัวอย่างที่ 1 What is your employee's name: DANGLON WORK HOUR(hours): 10 PAY DANGLON : 1200</p> <hr/> <p>ตัวอย่างที่ 2 What is your employee's name: NGOKUN WORK HOUR(hours): 41 PAY NGOKUN : 1255</p> <hr/> <p>ตัวอย่างที่ 3 What is your employee's name: HARRY WORK HOUR(hours): -5 INVALID WORK HOUR</p>

6. คำสั่งวนซ้ำแบบ while

ในการคำนวณบางโปรแกรมเรามีความจำเป็นที่จะต้องทำงานใดงานหนึ่งซ้ำ ๆ กันหลายรอบ ซึ่งในภาษาไพทอนได้ออกแบบคำสั่งที่ใช้ในการทำงานที่ซ้ำ ๆ กันหลาย ๆ ครั้งนี้ไว้หลายคำสั่ง หนึ่งในนั้นก็คือคำสั่ง while โดยคำสั่ง while จะทำงานในคำสั่งที่ถูกควบคุมโดย while (อยู่ใน block ของ while) ไปเรื่อย ๆ トラบไคที่เงื่อนไข (condition) ที่กำหนดไว้ด้านหลัง while ยังคงเป็นจริง ดังภาพที่ 6.1

รูปแบบของคำสั่ง while
<p>while (เงื่อนไข) : คำสั่งที่อยู่ในความควบคุมของ while คำสั่งที่ 1 คำสั่งที่อยู่ในความควบคุมของ while คำสั่งที่ 2 : : : คำสั่งที่อยู่ในความควบคุมของ while คำสั่งที่ n</p>



ภาพที่ 6.1 : แผนภาพการทำงานของคำสั่ง while

ตัวอย่างที่ 6.1 : โปรแกรมที่พิมพ์ข้อความ KU 1 ตามด้วย KU 2 ตามด้วย KU 3 ที่ละบรรทัดเป็นเช่นนี้ไปเรื่อย ๆ จำนวน n ครั้ง เขียนได้ดังนี้

```
i = 1
while (i<=n) :
    print("KU",i)
    i = i+1
```

แบบฝึกหัดที่ 6.1 : พิมพ์จาก 1 ถึง n

จงเขียนโปรแกรมเพื่อรับค่า n พิมพ์เลขบรรทัดละตัว ตั้งแต่ 1 ถึง n ให้เติมโปรแกรมให้สมบูรณ์

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>count = 0 n = int(input("Enter n: ")) while _____ : _____ _____ _____</pre>	Enter n: <u>5</u> 1 2 3 4 5

แบบฝึกหัดที่ 6.2 : พิมพ์จาก n ถึง 1

จงเขียนโปรแกรมเพื่อรับค่า n และพิมพ์เลขบรรทัดละตัว ตั้งแต่ n ถึง 1

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter n: <u>4</u> 4 3 2 1

แบบฝึกหัดที่ 6.3 : นับ

จงเขียนโปรแกรมรับเลขไปเรื่อย ๆ จนกว่าผู้ใช้จะพิมพ์ 0 ใ้รายงานจำนวนตัวเลขที่รับได้ (ไม่รวม 0) เติมโปรแกรมให้สมบูรณ์

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>count = 0 x = int(input("Enter number: ")) while _____ : _____ _____ print("Received", count, "numbers")</pre>	Enter number: <u>4</u> Enter number: <u>3</u> Enter number: <u>5</u> Enter number: <u>1</u> Enter number: <u>3</u> Enter number: <u>5</u> Enter number: <u>0</u> Received 6 numbers

แบบฝึกหัดที่ 6.4 : หาผลรวม

จงเขียนโปรแกรมเพื่อรับจำนวนเต็มไปเรื่อย ๆ จนกว่าผู้ใช้จะป้อนค่าลบ แสดงผลรวม (ไม่รวมตัวเลขลบตัวสุดท้ายที่รับได้)

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter number: <u>54</u> Enter number: <u>12</u> Enter number: <u>1</u> Enter number: <u>-10</u> Sum = 67

แบบฝึกหัดที่ 6.5 : นับเลขคี่

เขียนโปรแกรมเพื่อรับจำนวนเต็มไปเรื่อย ๆ จนกว่าผู้ใช้จะป้อนค่าลบ แสดงจำนวนตัวเลขที่เป็นเลขคี่ (ไม่รวมตัวเลขลบตัวสุดท้ายที่ได้รับ) จงเขียนโปรแกรมเพื่อรับจำนวนเต็มไปเรื่อย ๆ จนกว่าผู้ใช้จะป้อนค่าลบ แสดงผลรวม (ไม่รวมเลขลบตัวสุดท้าย)

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter number: 23 Enter number: 4 Enter number: 1 Enter number: 46 Enter number: 765 Enter number: 234 Enter number: -90 Received 3 odd numbers

แบบฝึกหัดที่ 6.6 : ตัวประกอบ

จงเขียนโปรแกรมเพื่อรับจำนวนเต็ม n จากนั้นพิมพ์ตัวประกอบทั้งหมดของ n ออกมา บรรทัดละหนึ่งตัว

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter number: 70 1 2 5 7 10 14 35 70

แบบฝึกหัดที่ 6.7 : ฟังก์ชันคืนจำนวนตัวประกอบ

ให้เขียนฟังก์ชัน factor_count(n) ที่คืนจำนวนของตัวประกอบของ n โดยกำหนดให้ส่วนของโปรแกรมหลักที่ใช้เรียกฟังก์ชันดังกล่าว เป็นดังนี้

```
n = int(input("Enter n: "))  
c = factor_count(n)  
print(c)
```

ฟังก์ชันที่เขียนได้คือ	ตัวอย่างการทำงาน	
	ตัวอย่างที่ 1 Enter n: 5 2 _____	ตัวอย่างที่ 2 Enter n: 70 8

แบบฝึกหัดที่ 6.8 : หาห.ร.ม.

จงเขียนโปรแกรมเพื่อรับจำนวนเต็มบวก a และ b หา ห.ร.ม. ของ a และ b

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter a: 100 Enter b: 75 The gcd is 25

แบบฝึกหัดที่ 6.9 : ฟังก์ชันหาห.ร.ม.

จากแบบฝึกหัดที่ 6.8 ให้เขียนฟังก์ชัน gcd(a,b) ที่คืนค่าห.ร.ม. ของ a และ b โดยตัวอย่างของโปรแกรมหลักที่เรียกใช้ฟังก์ชันดังกล่าว แสดงดังด้านล่าง

```
a = int(input("Enter a: "))
b = int(input("Enter b: "))
print("The gcd is", gcd(a,b))
```

ฟังก์ชันที่เขียนได้คือ	ตัวอย่างการทำงาน
	ตัวอย่างที่ 1 Enter a: <u>48</u> Enter b: <u>32</u> The gcd is 16

แบบฝึกหัดที่ 6.10 : สถิติ: หาค่าเฉลี่ย, ค่ามากที่สุด, ค่าน้อยที่สุด

จงเขียนโปรแกรมเพื่อรับค่าตัวเลขจำนวนเต็มไปเรื่อยๆ จนกว่าจะได้เลขติดลบแล้วให้ทำการคำนวณเพื่อหาค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด (โดยไม่คิดตัวเลขสุดท้ายที่ติดลบ) ดังตัวอย่าง

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	Enter number: <u>4</u> Enter number: <u>5</u> Enter number: <u>6</u> Enter number: <u>-1</u> Average: 5.00 Max: 6 Min: 4