



October 2016

Fundamental IT Engineer Examination (Afternoon)

ตอบคำถามตามข้อกำหนดด้านล่างนี้:

หมายเลขคำถาม	Q1 – Q6	Q7 , Q8
การเลือกคำถาม	บังคับ	เลือก 1 จาก 2
เวลาสอบ	13:30 - 16:00 (150 นาที)	

ข้อปฏิบัติ:

1. ให้ใช้ดินสอตอบ ถ้าต้องการเปลี่ยนคำตอบ ให้ลบคำตอบเก่าให้สะอาดก่อนโดยไม่ให้มีคราบยางลบเหลือ
2. ให้ทำเครื่องหมายบอกข้อมูลผู้สอบและคำตอบของแบบทดสอบ ตามคำสั่งด้านล่างอย่างเคร่งครัด หากทำเครื่องหมายไม่เหมาะสม คำตอบของท่านอาจไม่ได้รับการตรวจ ห้ามทำเครื่องหมายหรือเขียนตอบนอกพื้นที่ที่กำหนดไว้

(1) หมายเลขผู้สอบ (Examinee Number)

ให้เขียนหมายเลขผู้สอบลงในช่องที่เตรียมไว้ให้ และทำเครื่องหมายในช่องว่างที่เหมาะสมใต้ตัวเลขแต่ละตัว

(2) วันเกิด (Date of Birth)

ให้เขียนวันเกิดของผู้สอบ (เป็นตัวเลข) ลงในช่องที่เตรียมไว้ ให้ตรงกับที่พิมพ์อยู่ในบัตรเข้าห้องสอบ และทำเครื่องหมายในช่องว่างที่เหมาะสมที่อยู่ใต้ตัวเลขแต่ละตัว

(3) เลือกคำถาม (Q7-Q8)

ให้ทำเครื่องหมายตรง ⑤ ของคำถามที่ท่านเลือกตอบใน "คอลัมน์ที่เลือก" ในกระดาษคำตอบของท่าน

(4) คำตอบ (Answers)

ให้ทำเครื่องหมายตรงคำตอบที่เลือก ตามตัวอย่างคำถามที่อยู่ข้างล่าง

[คำถามตัวอย่าง]

ข้อสอบวัดระดับ Fundamental IT Engineer Examination รอบฤดูใบไม้ร่วงจัดในเดือนใด

- a) กันยายน b) ตุลาคม c) พฤศจิกายน d) ธันวาคม

เนื่องจากคำตอบที่ถูกเป็น "b) ตุลาคม" ดังนั้นให้ทำเครื่องหมายที่ข้อ b) ในกระดาษคำตอบ ดังตัวอย่างต่อไปนี้

[ตัวอย่างคำตอบ]






Sample	a	<input checked="" type="radio"/>	c	d	e	f	g	h	i	j
--------	---	----------------------------------	---	---	---	---	---	---	---	---

ห้ามเปิดดูข้อสอบก่อนได้รับอนุญาต
ข้อสงสัยที่เกี่ยวข้องกับคำถามในข้อสอบนี้ ผู้คุมอาจไม่ตอบ

สัญลักษณ์สำหรับ pseudo-language

ในคำถามที่ใช้ pseudo-language ให้ใช้สัญลักษณ์และเครื่องหมายต่อไปนี้ ยกเว้นว่าจะระบุเป็นอย่างอื่น

[การประกาศ คอมเมนต์ และกระบวนการ]

สัญลักษณ์	รายละเอียด
○	ประกาศชื่อ ชนิด และอื่นๆ สำหรับโปรแกรมย่อย และตัวแปร
/* text */	ระบุคอมเมนต์
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div> <ul style="list-style-type: none"> • variable ← expression </div> </div>	กำหนดค่าของนิพจน์ให้กับตัวแปร
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div> <ul style="list-style-type: none"> • procedure(argument, ...) </div> </div>	เลือกใช้โปรแกรมย่อย โดยผ่านและรับอาร์กิวเมนต์
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div>  </div> </div>	แสดงทางเลือกทางเดียว ถ้า conditional expression เป็นจริง process จะทำงาน.
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div>  </div> </div>	แสดงทางเลือกสองทาง ถ้า conditional expression เป็นจริง process 1 จะทำงาน ถ้าไม่จริง process 2 จะทำงาน..
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div>  </div> </div>	แสดงการวนซ้ำแบบตรวจสอบก่อน ในขณะที่ conditional expression เป็นจริง จะทำ process ซ้ำไปเรื่อยๆ
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div>  </div> </div>	แสดงการวนซ้ำแบบตรวจสอบทีหลัง process จะถูกทำงานก่อน และในขณะที่ conditional expression เป็นจริง จะทำ process ซ้ำไปเรื่อยๆ
<div style="display: flex;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg); font-weight: bold; margin-right: 5px;">Process</div> <div>  </div> </div>	แสดงการวนซ้ำ ค่าเริ่มต้นกำหนดโดย init (อยู่ในรูปนิพจน์) เก็บในตัวแปร variable ในตอนเริ่มต้น และในขณะที่เงื่อนไข cond เป็นจริง process จะถูกทำซ้ำไปเรื่อยๆ การเพิ่มค่า incr (อยู่ในรูปนิพจน์) จะเพิ่มค่าให้กับตัวแปร ในแต่ละรอบของการทำงาน

[ค่าคงที่ตรรกะ]

true, false

[ตัวดำเนินการ (operators) และลำดับความสำคัญระหว่างมัน (their priority)]

Type of operation	Operator	Priority
Unary operation	+, -, not	<div style="text-align: center;"> High ↑ ↓ Low </div>
Multiplication, division	*, /, %	
Addition, subtraction	+, -	
Relational operation	>, <, ≥, ≤, =, ≠	
Logical product	and	
Logical sum	or	

หมายเหตุ: การหารเลขจำนวนเต็ม จะได้ผลหารเป็นเลขจำนวนเต็ม เครื่องหมาย % แสดงการหาเศษของการหาร

Company names and product names appearing in the test questions are trademarks or registered trademarks of their respective companies. Note that the ® and TM symbols are not used within the text.

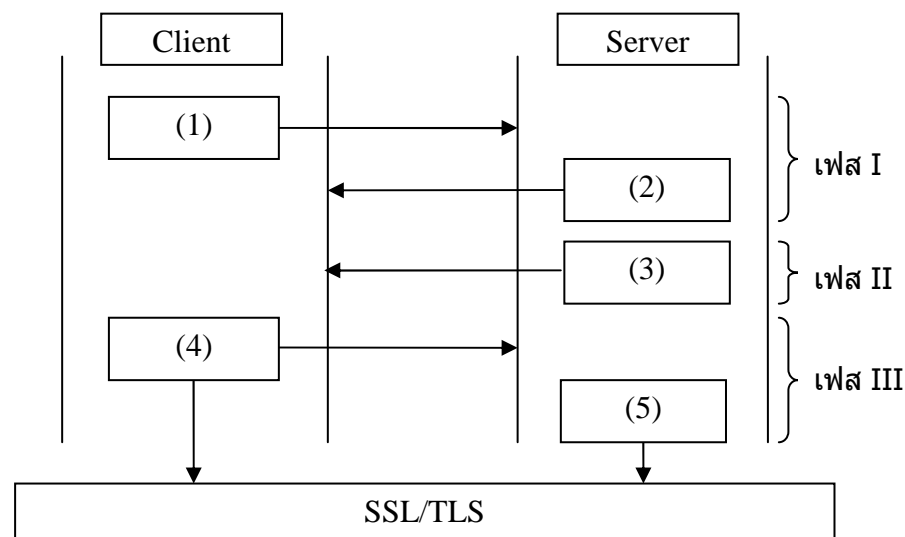
Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. จงอ่านคำอธิบายเกี่ยวกับช่องโหว่ความปลอดภัยและตอบคำถามย่อย 1 และ 2

TLS (Transport Layer Security) และ SSL (Secure Sockets Layer) เป็นโพรโทคอลการเข้ารหัสที่รักษาความปลอดภัยในการสื่อสารระหว่างไคลเอนต์ (client) กับเซิร์ฟเวอร์ (server) ผ่านอินเทอร์เน็ต แม้ว่า TLS เป็นถูกพัฒนาต่อเนื่องมาจาก SSL แต่โดยปกติทั้งคู่จะถูกอ้างถึงร่วมกันเป็น SSL/TLS

[แฮนด์เชคโพรโทคอล (handshake protocol)]

รูปที่ 1 แสดงการไหลของข้อมูลและกระบวนการระหว่างไคลเอนต์ (client) กับเซิร์ฟเวอร์ (server) ก่อนที่เซสชัน (session) การรักษาความปลอดภัย SSL/TLS จะเกิดขึ้น



รูปที่ 1 ภาพรวมของแฮนด์เชคโพรโทคอล SSL/TLS

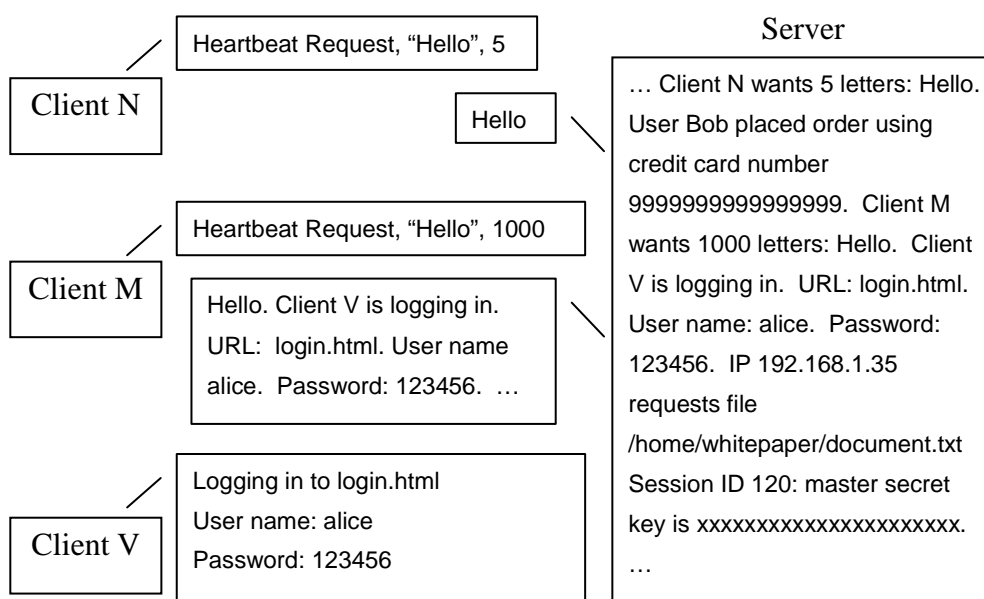
- (1) ในเฟส I, ไคลเอนต์ส่งข้อความฮัลโลไปยังไคลเอนต์พร้อมทั้งระบุเวอร์ชัน SSL รุ่นสูงที่สุดที่ไคลเอนต์รองรับ ตัวเลขสุ่มที่จะถูกใช้สำหรับการสร้างค่าความลับตัวหลัก (master secret) และรายการการเข้ารหัสและอัลกอริทึมการบีบอัดที่ไคลเอนต์รองรับ
- (2) ต่อมา เซิร์ฟเวอร์ตอบสนองด้วยข้อความฮัลโลของเซิร์ฟเวอร์ที่ระบุเวอร์ชัน SSL สูงสุดที่รองรับ โดยทั้งไคลเอนต์และเซิร์ฟเวอร์ ตัวเลขสุ่มอีกค่าหนึ่งสำหรับการสร้างความลับตัวหลัก พร้อมทั้งวิธีการเข้ารหัสที่เลือกและอัลกอริทึมการบีบอัดที่จะถูกใช้
- (3) ในเฟส II โดยส่วนใหญ่แล้ว เซิร์ฟเวอร์จะส่งใบรับรองของเซิร์ฟเวอร์ (certificate) หรือสายโซ่ใบรับรอง (certificate chain) ไปยังไคลเอนต์ ในขั้นตอนนี้ ไคลเอนต์จึงมี **A** ที่สามารถนำไปใช้ได้ ทั้งนี้ ทางเซิร์ฟเวอร์ก็สามารถส่งคำร้องขอใบรับรองไปยังไคลเอนต์ได้เช่นกันหากจำเป็นต้องใช้

- (4) ในเฟส III ถ้าได้รับการร้องขอ ไคลเอนต์จะส่งใบรับรองของตนและทำการพิสูจน์ตัวตนกับเซิร์ฟเวอร์ หลังจากนั้น ขึ้นอยู่กับอัลกอริทึมการแลกเปลี่ยนกุญแจ (key exchange algorithm) ไคลเอนต์จะสร้างค่าก่อนที่จะเป็นความลับตัวหลัก (pre-master secret) ในส่วนของตน และไคลเอนต์จะเข้ารหัสค่าดังกล่าวด้วย **A** ที่ได้มาจากใบรับรองของเซิร์ฟเวอร์ และส่งไปยังเซิร์ฟเวอร์ ขณะนี้ ไคลเอนต์จึงมีข้อมูลทั้งหมดที่จำเป็นสำหรับการสร้างกุญแจความลับตัวหลัก โดยการใช้ค่าที่ถูกสร้างทางฝั่งไคลเอนต์และข้อมูลที่ได้รับในขั้นตอนก่อนหน้านี้
- (5) ต่อมา เซิร์ฟเวอร์ใช้ **B** เพื่อที่จะถอดรหัสค่าที่ได้รับจากไคลเอนต์ ขณะนี้มันจึงสามารถสร้างกุญแจความลับตัวหลักได้เช่นกัน ดังนั้น แชนด์เชคบนฝั่งเซิร์ฟเวอร์จึงเสร็จสมบูรณ์

[ข้อผิดพลาดแบบฮาร์ทบีต (Heartbleed Bug)]

ฮาร์ทบีตเป็นช่องโหว่ใน OpenSSL ซึ่งเป็นการนำโพรโทคอล SSL/TLS ไปพัฒนาในรูปแบบโอเพนซอร์ส ข้อผิดพลาดได้มาจากส่วนขยายฮาร์ทบีต (heartbeat extension) ซึ่งหมายถึงการเตือนของหัวใจที่ใช้เพื่อคงการเชื่อมต่อเซสชัน SSL/TLS ไว้ เนื่องจากการสร้างเซสชัน SSL/TLS ใหม่เป็นภาระงานที่มาก ดังนั้นการดูแลให้เซสชันปัจจุบันทำงานได้อยู่ระหว่างเซิร์ฟเวอร์กับไคลเอนต์เป็นระยะเวลาที่สมเหตุสมผลถือว่าเป็นวิธีแก้ปัญหามิฉะนั้นทางปฏิบัติโดยเฉพาะอย่างยิ่งสำหรับเว็บไซต์และเซิร์ฟเวอร์ที่มีปริมาณการใช้ SSL/TLS สูง

โพรโทคอลฮาร์ทบีต (heartbeat protocol) ใช้ข้อความสองประเภท: คำร้องขอฮาร์ทบีต (Heartbeat Request) และคำตอบสนองฮาร์ทบีต (Heartbeat Response) ฝ่ายใดฝ่ายหนึ่งสามารถส่งคำร้องขอฮาร์ทบีตที่มีข้อความที่ส่งไปด้วย (payload message) พร้อมขนาดของข้อความดังกล่าว และอีกฝ่ายหนึ่งจะตอบสนองด้วยข้อความที่ถูกส่งมาเพื่อแสดงให้เห็นว่าอีกฝ่ายยังทำงานอยู่ รูปที่ 2 แสดงการร้องขอฮาร์ทบีตและการตอบสนองฮาร์ทบีต ที่มี "Hello" เป็นข้อความที่ส่งไปด้วย ของไคลเอนต์ที่ทำงานตามปกติ N, ไคลเอนต์ที่มึนร้าย M, และไคลเอนต์ที่อาจตกเป็นเหยื่อ V



รูปที่ 2 ตัวอย่างของคำร้องขอฮาร์ทบีตและคำตอบสนองฮาร์ทบีต

ข้อผิดพลาดสารทบทใช้ประโยชน์จากข้อความตอบสนองสารทบทที่ระบุขนาดสิ่งที่ส่งไปด้วยไม่ถูกต้อง สำหรับไคลเอนต์ M เซิร์ฟเวอร์ไม่ได้ทำการตรวจสอบขอบเขต (bound checking) และส่งข้อมูลในหน่วยความจำของตัวเองขนาด 1000 ไบต์โดยเริ่มต้นจากตำแหน่งที่อยู่ของสิ่งที่ส่งมาด้วยในหน่วยความจำแทนที่การส่งกลับมาเฉพาะตัวข้อความที่ส่งมาเท่านั้น เนื่องจากขนาดของสิ่งที่ส่งไปด้วยมีค่าเป็นเลขฐานสองขนาด 16 บิต ผู้โจมตีจึงสามารถได้รับหน่วยความจำของเซิร์ฟเวอร์ได้สูงสุดถึง ในการร้องขอสารทบทแต่ละครั้ง ดังนั้นจึงเป็นไปได้ที่ข้อมูลที่มีความอ่อนไหวจะรั่วไหลจากเซิร์ฟเวอร์ได้ ในกรณีที่เลวร้ายที่สุด ถ้า รั่วไหลออกไป ผู้โจมตีจะสามารถถอดรหัสการสื่อสารที่ถูกดักจับไว้ทั้งหมดทั้งในอดีตและอนาคตต่อไปได้โดยง่าย นอกจากนี้ มันยังทำให้ผู้โจมตีสามารถปลอมตัวเป็นเซิร์ฟเวอร์ได้แม้ว่าข้อผิดพลาดดังกล่าวได้ถูกแก้ไขเรียบร้อยแล้ว

ข้อผิดพลาดนี้ถูกค้นพบในปี 2014 และมีผลกับ OpenSSL เวอร์ชันที่เก่ากว่า และเวอร์ชันใหม่ 1.0.1g ซึ่งได้แก้ไขข้อผิดพลาดดังกล่าวได้ปล่อยให้ใช้ในเดือนเมษายน 2014 แต่อย่างไรก็ตาม ข้อผิดพลาดนี้ส่งผลกระทบอย่างแพร่หลายเนื่องจาก OpenSSL ได้ถูกรวมไว้ในระบบปฏิบัติการหลายเวอร์ชันซึ่งครอบคลุมตั้งแต่เซิร์ฟเวอร์ไปยังอุปกรณ์มือถือรวมทั้งในซอฟต์แวร์เว็บเซิร์ฟเวอร์ที่ได้รับความนิยม ดังนั้นผู้ใช้จึงควรอัปเดต OpenSSL ให้เป็นเวอร์ชันล่าสุด

คำถามย่อย 1

จากกลุ่มคำตอบข้างล่างนี้ จงเลือกคำตอบที่ถูกต้องเติมลงในช่องว่าง ในคำอธิบายข้างต้น และถ้าจำเป็น สามารถเลือกคำตอบเดิมได้สองครั้งหรือมากกว่าก็ได้

กลุ่มคำตอบสำหรับ A, B และ D

- a) กุญแจส่วนตัวของไคลเอนต์ (client's private key)
- b) กุญแจสาธารณะของไคลเอนต์ (client's public key)
- c) ค่าก่อนเป็นความลับตัวหลัก (pre-master secret)
- d) กุญแจความลับตัวหลัก (master secret key)
- e) กุญแจส่วนตัวของเซิร์ฟเวอร์ (server's private key)
- f) กุญแจสาธารณะของเซิร์ฟเวอร์ (server's public key)

กลุ่มคำตอบสำหรับ C

- | | | |
|--------------|-------------|-------------|
| a) 256 ไบต์ | b) 32k ไบต์ | c) 64k ไบต์ |
| d) 256k ไบต์ | e) 1M ไบต์ | f) 4M ไบต์ |

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกสองข้อความที่ถูกต้องเกี่ยวกับข้อผิดพลาดฮาร์ทบีต (heartbleed bug)

กลุ่มคำตอบ

- a) ทั้งไคลเอนต์และเซิร์ฟเวอร์สามารถส่งคำร้องขอฮาร์ทบีท ดังนั้นมันจึงเป็นไปได้ที่ผู้มั่งรายทางฝั่งเซิร์ฟเวอร์จะอ่านข้อมูลจากหน่วยความจำของไคลเอนต์ได้ด้วย
- b) มันเป็นข้อบกพร่องในการออกแบบส่วนขยายฮาร์ทบีท (heartbeat extension) ดังนั้น เพื่อให้ไม่ให้อายุเสียข้อมูลที่มีความอ่อนไหว ผู้ใช้จึงควรหลีกเลี่ยงการส่งคำร้องขอฮาร์ทบีทไปยังเซิร์ฟเวอร์จนกว่าจะได้รับแจ้งในโอกาสต่อไป
- c) ตรวจสอบการจำใจด้วยวิธีนี้ทำได้ยาก เพราะผู้โจมตีเพียงแต่อ่านเนื้อหาในหน่วยความจำเท่านั้น และการจำใจไม่ทั้งร่องรอยของการความเสียหาย
- d) บนเซิร์ฟเวอร์ที่มีช่องโหว่นี้ ผู้บุกรุกสามารถส่งข้อความคำร้องขอฮาร์ทบีทที่ไม่ถูกต้องเพื่อเปลี่ยนเนื้อหาในหน่วยความจำและสามารถเข้าถึงระบบเป็นรูท (root) ของระบบปฏิบัติการได้
- e) ข้อผิดพลาดนี้ถูกค้นพบในปี 2014 ในเวอร์ชันเก่าของ OpenSSL ดังนั้นจึงสรุปได้ว่าไม่มีการจำใจด้วยวิธีนี้เกิดขึ้นก่อนการค้นพบ

Q2. อ่านคำอธิบายเกี่ยวกับวงจรคณิตศาสตร์สำหรับการคูณข้างล่างนี้ จากนั้นตอบคำถามย่อย 1 และ 2

วิศวกรฮาร์ดแวร์กำลังวางแผนออกแบบวงจรคณิตศาสตร์ซึ่งคูณตัวเลขฐานสองแบบจุดตรึง (fixed-point binary number) ที่มีการแทนจำนวนแบบมีเครื่องหมาย จำนวน 2 ตัว

[การแทนจำนวนแบบมีเครื่องหมาย (signed magnitude representation)]

ในการแทนจำนวนแบบมีเครื่องหมาย ตัวเลขฐานสองแบบจุดตรึงจะถูกแทนดังนี้

บิตแสดงเครื่องหมาย (sign bit)	บิตแสดงจำนวน (magnitude bits)
-------------------------------	-------------------------------

บิตที่อยู่ทางซ้ายมือสุดเป็นบิตแสดงเครื่องหมาย ซึ่ง 0 หมายถึงเครื่องหมาย บวก ส่วน 1 หมายถึงเครื่องหมายลบ บิตที่เหลือนับจากบิตแสดงเครื่องหมาย เป็นบิตแสดงจำนวนที่แทนค่าสัมบูรณ์ของตัวเลข

ตัวอย่าง:

01001 แสดงถึงเลข +9 เพราะว่ามีบิตที่แสดงเครื่องหมายเป็น 0 (+) และมีบิตที่แสดงจำนวนเป็น 1001 (9)

10011 แสดงถึงเลข -3 เพราะว่ามีบิตที่แสดงเครื่องหมายเป็น 1 (-) และมีบิตที่แสดงจำนวนเป็น 0011 (3)

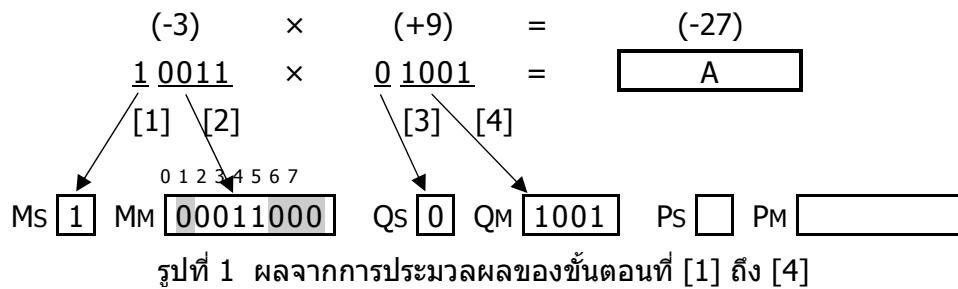
ในการแทนจำนวนแบบมีเครื่องหมาย การแทนค่าเลขศูนย์สามารถเป็นได้ 2 แบบ คือ +0 และ -0 เมื่อบิตที่แสดงจำนวนเป็น 0 ทั้งหมด ค่าของข้อมูลจะประเมินเป็น 0 โดยไม่สนใจค่าของบิตที่แสดงเครื่องหมาย

[วงจรคณิตศาสตร์สำหรับการคูณ]

- (1) การคำนวณกระทำกับตัวเลขที่มีการแทนจำนวนแบบมีเครื่องหมาย
- (2) ตัวตั้งหนึ่งตัวกับตัวคูณอีกหนึ่งตัว จะมีความยาว 5 บิต (รวมบิตที่แสดงเครื่องหมาย) และผลคูณจะมีความยาว 9 บิต (รวมบิตที่แสดงเครื่องหมาย).
- (3) มีรีจิสเตอร์สำหรับใช้งาน 6 ตัว ได้แก่ MS (รีจิสเตอร์ขนาด 1 บิต) QS (รีจิสเตอร์ขนาด 1 บิต) PS (รีจิสเตอร์ขนาด 1 บิต) MM (รีจิสเตอร์ขนาด 8 บิต) QM (รีจิสเตอร์ขนาด 4 บิต) และ PM (รีจิสเตอร์ขนาด 8 บิต).

ขั้นตอนต่อไปนี้ใช้การอธิบายวิธีการคูณที่ดำเนินการกับ $(-3) \times (+9) = (-27)$ ซึ่งแทนค่าภายในได้เป็น $10011 \times 01001 = \boxed{A}$ เป็นตัวอย่าง

- [1] โหลดบิตแสดงเครื่องหมายของตัวตั้งไปไว้ที่รีจิสเตอร์ MS
- [2] โหลดบิตแสดงจำนวนของตัวตั้งไปไว้ที่ บิตที่ 1 ถึง 4 ของรีจิสเตอร์ MM กำหนดค่า 0 ให้กับบิตที่เหลือของรีจิสเตอร์ MM (ส่วนที่เป็นแรงจูงใจในรูปที่ 1)
- [3] โหลดบิตแสดงเครื่องหมายของตัวคูณไปไว้ที่รีจิสเตอร์ QS
- [4] โหลดบิตแสดงจำนวนของตัวคูณไปไว้ที่รีจิสเตอร์ QM



- [5] ทดสอบบิตขวามือสุดของรีจิสเตอร์ QM ถ้ามีค่าเป็น 1 ให้คัดลอกข้อมูลของรีจิสเตอร์ MM ไปไว้ที่รีจิสเตอร์ PM แต่ถ้าค่าเป็น 0 ให้โหลดค่า 00000000 ไปไว้ที่รีจิสเตอร์ PM
- [6] ทำซ้ำขั้นตอน [6-1] ถึง [6-3] จำนวน 3 รอบ
- [6-1] เลื่อนข้อมูลของรีจิสเตอร์ PM ไปทางขวาจำนวน 1 บิต
- [6-2] เลื่อนข้อมูลของรีจิสเตอร์ QM ไปทางขวาจำนวน 1 บิต
- [6-3] ทดสอบบิตที่อยู่ขวาสุดของรีจิสเตอร์ QM ถ้ามีค่าเป็น 1 ให้นำข้อมูลของรีจิสเตอร์ MM เข้ากับข้อมูลที่อยู่ในรีจิสเตอร์ PM
- [7] กำหนดผลลัพธ์ของการดำเนินการทางตรรกศาสตร์ B ไปที่รีจิสเตอร์ Ps
- [8] ขั้นตอนสุดท้าย บิตที่แสดงเครื่องหมายของผลคูณจะดึงมาได้จากข้อมูลในรีจิสเตอร์ Ps และบิตที่แสดงจำนวนของผลคูณ (magnitude bits) จะดึงมาได้จากข้อมูลในรีจิสเตอร์ PM

หลังจากประมวลผลขั้นตอนที่ [5]: QM 1001 PM 00011000
 หลังจากทำงานรอบแรกของขั้นตอนที่ [6]: QM 0100 PM
 หลังจากทำงานรอบที่สองของขั้นตอนที่ [6]: QM 0010 PM C
 หลังจากทำงานรอบที่สามของขั้นตอนที่ [6]: QM 0001 PM

หมายเหตุ: ส่วนที่แรเงา ไม่ถูกแสดงออกมา

รูปที่ 2 ผลจากการประมวลผลของขั้นตอนที่ [5] และ [6]

ในการลงมือปฏิบัติจริง ถ้า PM และ QM ถูกรวมเข้าด้วยกันเป็นรีจิสเตอร์ที่มีความยาวดังแสดงข้างล่าง ขั้นที่ [1] ถึง [8] ยังคงทำงานได้ถูกต้องและจำนวนครั้งของการเลื่อนสามารถลดลงครึ่งหนึ่ง

PM	QM
----	----

คำถามย่อย 1

จากกลุ่มของคำตอบข้างล่างนี้ จงเลือกคำตอบที่ถูกต้องที่สุดเติมไปในช่องว่าง แต่ละช่อง ในคำอธิบายด้านบน ในที่นี้ เครื่องหมาย "&", "|", และ "^" แสดงถึงตัวดำเนินการทางตรรกศาสตร์ AND OR และ XOR ตามลำดับ

กลุ่มคำตอบสำหรับ A

- | | |
|--------------|--------------|
| a) 000011011 | b) 000110011 |
| c) 100011011 | d) 100110011 |

กลุ่มคำตอบสำหรับ B

a) $(Ms \& Qs)$

b) $(Ms | Qs)$

c) $(Ms \wedge Qs)$

กลุ่มคำตอบสำหรับ C

a) 00000011

b) 00000110

c) 00001100

d) 01100000

คำถามย่อย 2

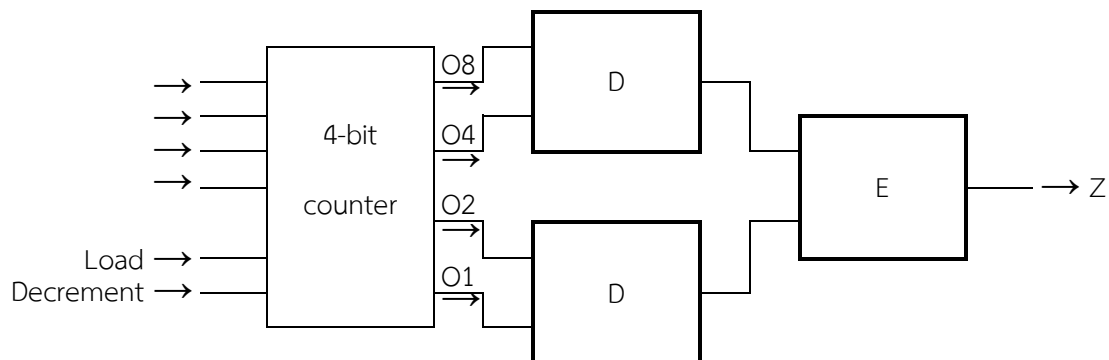
จากกลุ่มของคำตอบข้างล่างนี้ จงเลือกคำตอบที่ถูกต้องที่สุดเติมไปในช่องว่าง แต่ละช่อง ในรูปที่ 3

ในการดำเนินการคุณ ขั้นตอนที่ [6] ถูกประมวลผลจำนวน 3 รอบ งานของวิศวกรฮาร์ดแวร์คือพัฒนาแบบวงจรตรรกศาสตร์สำหรับควบคุมการวนรอบดังแสดงในรูปที่ 3

ในรูปที่ 3 ตัวนับขนาด 4 บิต (ต่อไปนี้จะเรียกตัวนับ) มีฟังก์ชันการทำงานดังต่อไปนี้

- ในขั้นแรก ด้วยสัญญาณ Load ค่านับ 3 ถูกกำหนดให้กับตัวนับ
- จากนั้น ในแต่ละครั้งที่ตัวนับได้รับสัญญาณ decrement ตัวนับจะลดค่าการนับทีละหนึ่ง และส่งค่าที่ลดลงแล้วออกไปที่เส้น O8, O4, O2 และ O1
 - ครั้งที่ 1, $(O8, O4, O2, O1) = (0, 0, 1, 0)$ เพราะค่าถูกลดลงเป็น 2 ($=0010$)
 - ครั้งที่ 2, $(O8, O4, O2, O1) = (0, 0, 0, 1)$ เพราะค่าถูกลดลงเป็น 1 ($=0001$)
 - ครั้งที่ 3, $(O8, O4, O2, O1) = (0, 0, 0, 0)$ เพราะค่าถูกลดลงเป็น 0 ($=0000$)

ในที่สุดวงจรตรรกศาสตร์ในรูปที่ 3 จะแสดงผลลัพท์สัญญาณ Z ซึ่งเป็นเลขฐานสอง (0 หรือ 1) ผลลัพท์ของวงจรตรรกศาสตร์ $Z = 1$ เมื่อ $(O8, O4, O2, O1) = (0, 0, 0, 0)$ ไม่เช่นนั้น $Z = 0$.



รูปที่ 3 วงจรตรรกศาสตร์สำหรับควบคุมการวนรอบ

กลุ่มคำตอบสำหรับ D และ E

a) (AND gate)

b) (OR gate)

c) (XOR gate)

d) (NAND gate)

e) (NOR gate)

Q3. อ่านคำอธิบายของระบบฐานข้อมูลต่อไปนี้ แล้วตอบคำถามย่อย 1 ถึง 3

ร้านโชคดีอาบนํ้าตัดขนสุนัขเป็นร้านดูแลสัตว์เลี้ยงที่ให้บริการอาบน้ำและตัดขนสัตว์เลี้ยงแบบครบวงจรโดยเฉพาะสำหรับสุนัข ทางร้านทำการจัดเก็บข้อมูลเกี่ยวกับสัตว์เลี้ยงในตารางชื่อ PetTable ที่มีแอตทริบิวต์ดังต่อไปนี้ รหัสสุนัข ชื่อสุนัข พันธุ์ ชื่อเจ้าของสัตว์เลี้ยง และยอดค่าใช้จ่ายบริการ โดยมีโครงสร้างตารางเป็นดังนี้:

PetTable (DogID, DogName, Breed, OwnerName, BalanceDue)

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเติมลงในช่องว่าง ในคำสั่ง SQL ดังต่อไปนี้

เจ้าของสัตว์เลี้ยงบางรายมีสุนัขมากกว่าหนึ่งตัว ผู้จัดการร้านต้องการสร้างรายงานที่แสดงรายชื่อของเจ้าของสัตว์เลี้ยงที่มีสุนัขมากกว่าหนึ่งตัว จึงสร้างชุดคำสั่ง SQL ตั้งชื่อว่า "SQL1" ดังต่อไปนี้

```
-- SQL statement "SQL1"
SELECT OwnerName, DogName, Breed, BalanceDue
FROM PetTable
WHERE  A
ORDER BY OwnerName
```

ตัวอย่างรายงานที่สร้างได้จาก "SQL1" เป็นดังนี้:

OwnerName	DogName	Breed	BalanceDue
Henry Chauncey	Buddy	Great Dane	1000
Henry Chauncey	Abe	Bulldog	300
Mike Barz	Baxter	Boxer	1000
Mike Barz	Fluffy	Poodle	0
Mike Barz	Love	Poodle	100

กลุ่มคำตอบสำหรับ A

- a) COUNT(OwnerName)>1
- b) DogID IN (SELECT DogID FROM PetTable HAVING(COUNT(DogID)>1))
- c) DogID IN (SELECT DogID FROM PetTable HAVING(COUNT(OwnerName)>1))
- d) HAVING(COUNT(DogID)>1)
- e) OwnerName IN (SELECT OwnerName FROM PetTable
GROUP BY OwnerName HAVING(COUNT(OwnerName)>1))

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเติมลงในช่องว่าง ในคำสั่ง SQL ดังต่อไปนี้ และถ้าจำเป็น สามารถเลือกคำตอบเดิมได้สองครั้งหรือมากกว่าก็ได้

ร้านโชคดีอานน้ำตัดขนสุนัข ต้องการเก็บข้อมูลเจ้าของสัตว์เลี้ยงแต่ละรายในตารางชื่อ OwnerTable โดยมีแอตทริบิวต์ได้แก่ รหัสเจ้าของ ที่อยู่ เมือง และเบอร์โทรศัพท์ โดยโครงสร้างตารางเป็นดังนี้:

OwnerTable (OwnerID, OwnerName, Address, Township, TelephoneNo)

เพื่อเป็นการสร้างความสัมพันธ์ระหว่างตาราง OwnerTable และ PetTable จึงแทนที่แอตทริบิวต์ OwnerName ในตาราง PetTable ด้วยแอตทริบิวต์ OwnerID เป็นผลให้โครงสร้างตาราง PetTable เป็นดังนี้:

PetTable (DogID, DogName, Breed, OwnerID, BalanceDue)

ผู้จัดการร้านค้าต้องการให้ส่วนลด 10% จากยอดค่าใช้จ่ายบริการสุนัข ให้แก่เจ้าของสัตว์เลี้ยงที่มาจากเมืองอื่นนอกเหนือจาก "Wild Rose" ซึ่งเป็นเมืองที่ร้านตั้งอยู่ ดังนั้นจึงสร้างชุดคำสั่ง SQL ตั้งชื่อว่า "SQL2" ดังต่อไปนี้

```
-- SQL statement "SQL2"
UPDATE  B
SET BalanceDue = BalanceDue - (BalanceDue * 0.1)
WHERE  C IN (SELECT  D
                        FROM  E
                        WHERE Township <> 'Wild Rose')
```

ตารางต่อไปนี้แสดงว่ายอดค่าใช้จ่ายบริการสุนัขจะถูกเปลี่ยนแปลงอย่างไรหลังรันคำสั่ง "SQL2"

DogID	DogName	Breed	BalanceDue ก่อนอัปเดต	BalanceDue หลังอัปเดต	Township
1	Buddy	Great Dane	1000	1000	Wild Rose
2	Abe	Bulldog	300	300	Wild Rose
3	Acridus	Great Dane	1500	1350	Schaumburg
4	Bam Bam	Bulldog	1000	900	Schaumburg
5	Baxter	Boxer	1000	900	Dubuque
6	Fluffy	Poodle	0	0	Dubuque
7	Love	Poodle	100	90	Dubuque

กลุ่มคำตอบสำหรับ B ถึง E

- a) BalanceDue b) DogID c) OwnerID
- d) OwnerTable e) PetTable f) Township

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเติมลงในช่องว่าง ในคำสั่ง SQL ดังต่อไปนี้

ผู้จัดการร้านต้องการสร้างรายงานที่แสดงรายการจำนวนสุนัข และยอดรวมค่าใช้บริการสุนัขที่ได้จากแต่ละเมือง ดังนั้นจึงสร้างชุดคำสั่ง SQL ดังชื่อว่า "SQL3" ดังต่อไปนี้

```
-- SQL statement "SQL3"
SELECT  F
      FROM  G
GROUP BY OwnerTable.Township
```

ตัวอย่างรายงานที่สร้างได้จาก "SQL3" เป็นดังนี้:

NoOfDogs	TotalAmount	Township
2	1300	wild Rose
2	2250	Schaumburg
3	990	Dubuque

กลุ่มคำตอบสำหรับ F และ G

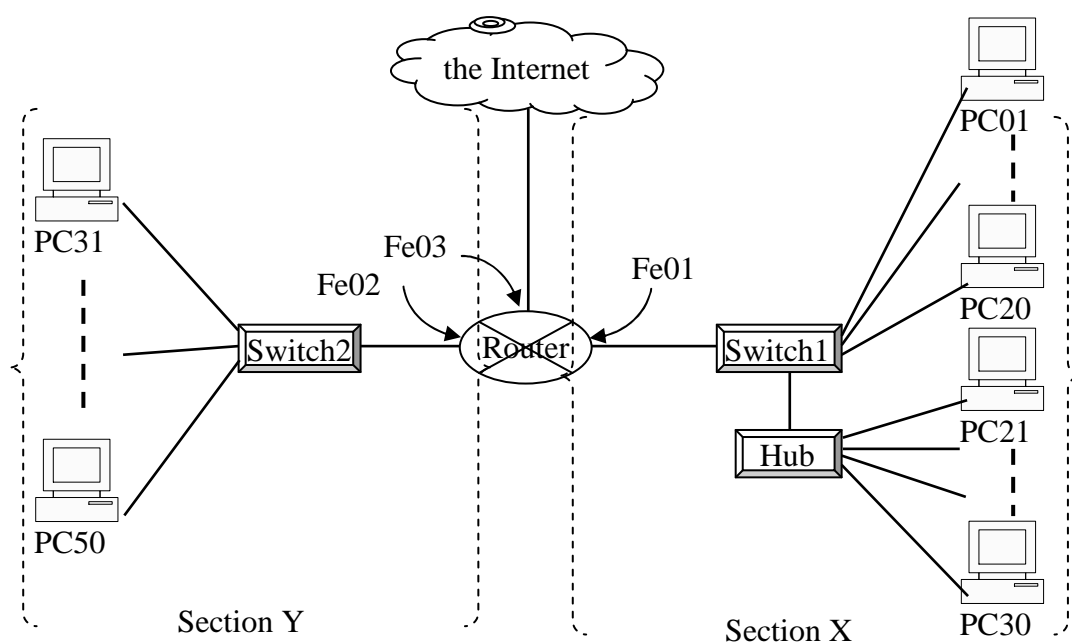
- a) COUNT(DogID) AS NoOfDogs,
SUM(BalanceDue) AS TotalAmount, Township
- b) COUNT(DogID) AS NoOfDogs,
TOTAL(BalanceDue) AS TotalAmount, Township
- c) OwnerTable
- d) OwnerTable INNER JOIN PetTable
ON OwnerTable.OwnerID = PetTable.OwnerID
- e) PetTable IN (SELECT OwnerID FROM OwnerTable
HAVING PetTable.OwnerID = OwnerTable.OwnerID)

Q4. อ่านคำอธิบายเกี่ยวกับปัญหาการเชื่อมต่อของเครือข่าย และตอบคำถามย่อย 1 และ 2

บริษัท ABC ได้มีการติดตั้งระบบ LAN ภายใน เพื่อให้มีการใช้ทรัพยากรร่วมกันภายในสำนักงานและสามารถเข้าถึงอินเทอร์เน็ตได้ สำนักงานแบ่งออกเป็นสองแผนกคือ X และ Y ซึ่งมีเครื่องพีซีทั้งหมด 50 เครื่อง นอกจากนี้ยังขอเช่าใช้เว็บเซิร์ฟเวอร์จากผู้ให้บริการอินเทอร์เน็ต (ISP) โดยวางไว้ที่บริษัทของผู้ให้บริการอินเทอร์เน็ต (ISP) นั้น เพื่อให้ประสิทธิภาพการรักษาความปลอดภัยและประสิทธิภาพการทำงานที่ดีที่สุด แผนกทั้งสองจะแยกกันโดยใช้เราเตอร์ ตารางที่ 1 แสดงรายการของเครื่องพีซีที่อยู่ในแผนก X และ Y และรูปที่ 1 แสดงให้เห็นแผนผังการเชื่อมต่อเครือข่าย

ตารางที่ 1 รายการเครื่องพีซีที่เชื่อมต่อกันผ่าน Switch/Hub

เครื่องพีซี	การเชื่อมต่อกับ Switch/Hub	แผนก
PC01 to PC20	Switch1	X
PC21 to PC30	Hub	X
PC31 to PC50	Switch2	Y



รูปที่ 1 แผนผังการเชื่อมต่อเครือข่าย

เครื่องพีซีในระบบ LAN ทั้งสองถูกกำหนดค่าโดยใช้โปรโตคอลไอพีแอดเดรส คลาส C โดยใช้ซับเน็ตมาสก์ตามปกติ ตารางที่ 2 แสดงข้อมูลบางส่วนเกี่ยวกับชื่ออุปกรณ์ ไอพีแอดเดรส ซับเน็ตมาสก์ แมคแอดเดรส (MAC address) และเกตเวย์ปริยาย (default gateway) ของแต่ละอุปกรณ์

ตาราง 2 แสดงข้อมูลของแต่ละอุปกรณ์ในเครือข่าย LANs

ชื่ออุปกรณ์	ไอพีแอดเดรส	แมคแอดเดรส	เกตเวย์ปริยาย	แผนก
Fe01	192.168.1.1/24	AA:AA:AA:12:34:5A		X
Switch1	192.168.1.11/24	AA:AA:12:5E:7D:6D		
PC01	192.168.1.101/24	AA:0F:3C:33:44:01	192.168.1.1	
...	192.168.1.1	
PC20	192.168.1.120/24	AA:0F:3C:33:44:20	192.168.1.1	
PC21	192.168.1.121/24	AA:0F:3C:33:44:21	192.168.1.11	
...	192.168.1.11	
PC30	192.168.1.130/24	AA:0F:3C:33:44:30	192.168.1.11	Y
Fe02	192.168.3.2/24	AA:AA:AA:12:34:5B		
Switch2	192.168.3.22/24	AA:AA:12:5E:7D:6E		
PC31	192.168.2.231/24	AA:0F:3C:33:44:31	192.168.3.2	
...	192.168.3.2	
PC50	192.168.2.250/24	AA:0F:3C:33:44:50	192.168.3.2	Internet
Fe03	203.129.30.57/30	AA:AA:AA:12:34:5C		
Web server web.example.com	198.51.100.189/28	AB:CD:EF:12:34:56		

คำถามย่อย 1

จากกลุ่มคำตอบที่อยู่ด้านล่าง จงเลือกคำตอบที่เหมาะสมเพื่อใส่ในช่องว่าง แต่ละช่องตามรายละเอียดต่อไปนี้

หลังจากกำหนดค่าเครือข่ายเสร็จสมบูรณ์แล้ว พนักงานของทั้งสองแผนกเริ่มใช้ทรัพยากรร่วมกันภายในแผนกของตัวเอง อย่างไรก็ตามพนักงานส่วนใหญ่ยังพบปัญหาเกี่ยวกับการเชื่อมต่อระหว่างแผนกและการเชื่อมต่ออินเทอร์เน็ต บริษัท ABC จึงว่าจ้างวิศวกรเครือข่ายเพื่อตรวจสอบปัญหาที่เกิดขึ้น

วิศวกรเครือข่ายได้จัดทำแผนในการทดสอบการเชื่อมต่อโดยใช้คำสั่ง ping โดยจัดกรณีทดสอบเป็นสามกรณีแล้วเริ่มการทดสอบ

ผลที่ได้จากกรณีทดสอบ (1) เป็นดังนี้

กรณีทดสอบ	ต้นทาง	ปลายทาง	คำสั่ง	ผลลัพธ์	สาเหตุ
(1)	PC01 – PC20	PC21	ping 192.168.1.121	ping สำเร็จ	
	PC21 – PC30	PC01	ping 192.168.1.101	ping สำเร็จ	
	PC01	Router Fe01	ping 192.168.1.1	ping สำเร็จ	
	PC21	Router Fe01	ping 192.168.1.1	ping สำเร็จ	
	PC01 – PC20	Web Server	ping 198.51.100.189	ping สำเร็จ	
	PC21 – PC30	Web Server	ping 198.51.100.189	ping ล้มเหลว	A

วิศวกรเครือข่ายได้แก้ไขสาเหตุ A แล้วดำเนินการทดสอบกับกรณีทดสอบ (2) ผลที่ได้จากกรณีทดสอบ (2) มีรายละเอียดดังนี้

กรณีทดสอบ	ต้นทาง	ปลายทาง	คำสั่ง	ผลลัพธ์	สาเหตุ
(2)	PC01 – PC20	PC31	ping 192.168.2.231	ping ล้มเหลว	B
	PC21 – PC30	PC31	ping 192.168.2.231	ping ล้มเหลว	
	PC31 – PC50	PC01	ping 192.168.1.101	ping ล้มเหลว	
	PC31 – PC50	PC21	ping 192.168.1.121	ping ล้มเหลว	
	PC31 – PC49	PC50	ping 192.168.2.250	ping สำเร็จ	
	PC31 – PC50	Web Server	ping 198.51.100.189	ping ล้มเหลว	B
	PC31 – PC50	Router Fe02	ping 192.168.3.2	ping ล้มเหลว	

วิศวกรเครือข่ายได้แก้ไขสาเหตุ B แล้วดำเนินการทดสอบกับกรณีทดสอบ (3) ผลที่ได้จากกรณีทดสอบ (3) มีรายละเอียดดังนี้

กรณีทดสอบ	ต้นทาง	ปลายทาง	คำสั่ง	ผลลัพธ์	สาเหตุ
(3)	PC01 – PC50	Web Server	ping web.example.com	ping ล้มเหลว	C

กลุ่มคำตอบสำหรับ A ถึง C

- a) เราเตอร์พอร์ต Fe02 ไม่ได้ถูกลงทะเบียนไว้กับ Fe01 และ Fe03
- b) เครื่องพีซีแต่ละเครื่องไม่ได้กำหนดค่าแอดเดรสของเครื่องแม่ข่าย DNS ไว้
- c) แอดเดรสของเกตเวย์บนเครื่องพีซีทุกเครื่องของบริษัท ABC ไม่ได้ให้ค่าเป็น 203.129.30.57
- d) แอดเดรสเกตเวย์ของ PC21 ถึง PC30 ถูกกำหนดให้เป็น 192.168.1.11/24 แทนที่จะเป็น 192.168.1.1/24
- e) ไอพีแอดเดรสของ Fe03 ไม่ได้ถูกกำหนดให้เป็น 192.168.2.2/24
- f) ไอพีแอดเดรสของ Switch1 ไม่ได้ถูกกำหนดให้เป็น 192.168.1.1/24
- g) ไอพีแอดเดรสของ Switch1 ไม่ได้ถูกกำหนดให้เป็น 192.168.3.2/24
- h) ไอพีแอดเดรสของ PC31 ถึง PC50 ถูกกำหนดค่าผิดเป็น 192.168.2.xxx แทนที่จะเป็น 192.168.3.xxx

คำถามย่อย 2

จากกลุ่มคำตอบที่อยู่ด้านล่าง จงเลือกคำตอบที่เหมาะสมเพื่อใส่ในช่องว่าง แต่ละช่องตามคำอธิบายต่อไปนี้

พนักงานคนหนึ่งทำงานในแผนก Y ได้ถูกย้ายมาทำงานในแผนก X ซึ่งเครื่อง PC32 จะต้องถูกย้ายไปกับเขาด้วยเพราะมีข้อมูลสารสนเทศที่เกี่ยวข้องกับผลประโยชน์ของบริษัท เขาได้เชื่อมต่อเครื่องคอมพิวเตอร์ของเขาเข้ากับ Switch1 แต่เขาไม่สามารถเรียกดูเว็บไซต์บนอินเทอร์เน็ตรวมทั้งไม่สามารถใช้ทรัพยากรในแผนก X ได้ เขาสังเกตเห็นว่ามีไฟกระพริบที่พอร์ต Ethernet ของเครื่องพีซีของเขาและสวิตช์พอร์ตที่ต่ออยู่ตามปกติเช่นเดียวกับที่เครื่องพีซีเครื่องอื่นกระพริบอยู่ เขาได้ขอให้วิศวกรเครือข่ายช่วยแก้ปัญหาที่เกิดขึ้น วิศวกรเครือข่ายแนะนำให้เขา D ซึ่งเขาพบว่าคำแนะนำนี้ช่วยแก้ปัญหาได้

กลุ่มคำตอบสำหรับ D

- a) เปลี่ยนค่า DNS แอดเดรสที่อยู่บนเครื่องพีซีเป็น 203.129.30.57 และเกตเวย์ปริยายเป็น 192.168.1.1
- b) เปลี่ยนค่าไอพีแอดเดรสบนเครื่องพีซีเป็น 192.168.1.132 และ DNS เป็น 203.129.30.57
- c) เปลี่ยนค่าไอพีแอดเดรสบนเครื่องพีซีเป็น 192.168.1.132 และเกตเวย์ปริยายเป็น 192.168.1.1
- d) เปลี่ยนสาย UTP แบบต่อตรง (straight through UTP cable) ให้เป็นสาย UTP แบบต่อไขว้ (crossover UTP cable)
- e) รีเซ็ต Switch1 เพื่อตั้งค่าตารางแมคแอดเดรส (MAC address table) ของ Switch1 ใหม่

Q5. อ่านคำอธิบายการออกแบบการทดสอบซอฟต์แวร์ต่อไปนี้ จากนั้นให้ตอบคำถามย่อย 1 ถึง 3

บริษัท N ซึ่งเป็นบริษัทผู้รวมระบบ (system integrator) กำลังทบทวนวิธีการทดสอบเพื่อลดจำนวนบั๊กที่ไม่ได้แก้ไขในโปรแกรมที่บริษัทเป็นผู้พัฒนา

[คำอธิบายวิธีการทดสอบที่ใช้ในบริษัท N]

บริษัท N ทดสอบโปรแกรมที่บริษัทพัฒนาขึ้นมาโดยใช้การทดสอบกระแสควบคุม (control flow testing) เป็นหลัก ซึ่งการทดสอบนี้เป็นหนึ่งในวิธีการทดสอบแบบกล่องขาว (white box testing method)

การทดสอบกระแสควบคุมมุ่งความสนใจไปที่หน่วยที่เล็กที่สุดที่ประกอบขึ้นมาเป็นโปรแกรม เช่น คำสั่งต่าง ๆ เส้นทางของโปรแกรม และเงื่อนไขที่ใช้ในการตัดสินใจ ซึ่งกรณีทดสอบ (test cases) และข้อมูลทดสอบ (test data) จะถูกเตรียมไว้ให้สอดคล้องกับเงื่อนไขครอบคลุม (coverage criteria) ที่กำหนดไว้ขณะวางแผนการทดสอบ จากนั้น พฤติกรรมของโปรแกรมที่ถูกพัฒนาขึ้นมาจึงถูกตรวจสอบ

เงื่อนไขครอบคลุมมีทั้งการครอบคลุมคำสั่ง (statement coverage) ที่คำสั่งทั้งหมดต้องถูกดำเนินการ (executed) อย่างน้อยหนึ่งครั้งในการทดสอบ และการครอบคลุมเงื่อนไขการตัดสินใจ (decision condition coverage) ซึ่งจากนี้ไปจะเรียกว่าการครอบคลุมทางแยก (branch coverage) ที่เส้นทางทั้งหมดของทุกทางแยกในโปรแกรมต้องถูกดำเนินการอย่างน้อยหนึ่งครั้ง

บริษัท N ใช้การครอบคลุมทางแยกเป็นเงื่อนไขครอบคลุมของบริษัท

[คำอธิบายเงื่อนไขการตัดสินใจของการครอบคลุมทางแยกที่ใช้โดยบริษัท N]

เงื่อนไขการตัดสินใจสำหรับทางแยกหนึ่งมีทั้งแบบเงื่อนไขเดียว (single condition) ที่ประเมินเงื่อนไขเพียงข้อเดียว และแบบหลายเงื่อนไข (multiple condition) ที่รวมการประเมินสองเงื่อนไขหรือมากกว่าเข้าด้วยกันโดยใช้ "and" หรือ "or" ซึ่งตัวอย่างด้านล่างนี้แสดงภาพทั้งแบบเงื่อนไขเดียวและแบบหลายเงื่อนไข

ตัวอย่าง	$(a > b)$	and	$(a < c)$
	Single condition		Single condition
	Multiple condition		

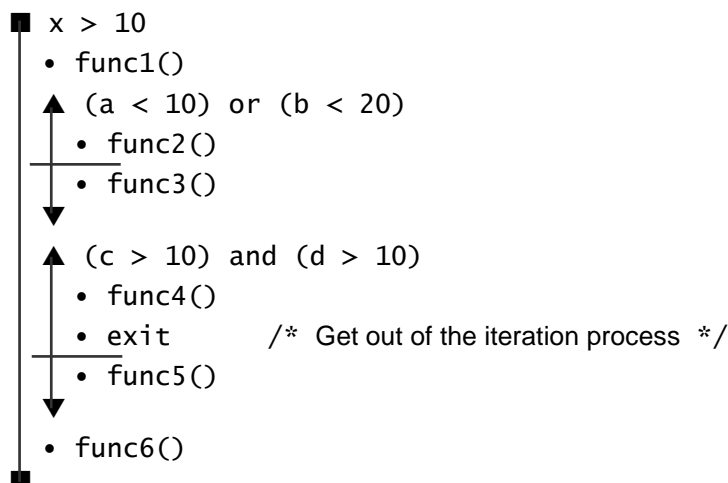
ในที่นี้ เมื่อโปรแกรมถูกประมวลผล การประเมินแบบทางลัด (short-cut evaluation) จะถูกใช้กับทางแยกแบบหลายเงื่อนไข ในการประเมินค่าแบบทางลัดนั้น เงื่อนไขเดียวต่าง ๆ ที่ประกอบกันขึ้นมาเป็นทางแยกหลายเงื่อนไขจะถูกนำไปประเมินตามลำดับจากซ้ายไปขวา เมื่อสามารถทราบถึงผลลัพธ์ของทางแยกหลายเงื่อนไขนั้นแล้ว เงื่อนไขเดียวที่เหลืออยู่จะไม่ถูกนำไปประเมิน ตัวอย่างเช่นในกรณีทางแยกแบบหลายเงื่อนไขที่มีสองเงื่อนไขเดียวมารวมกันด้วย "and" หากผลลัพธ์การประเมินค่าเงื่อนไขเดียวตัวแรกเป็นเท็จ ผลลัพธ์จากการประเมินค่าทางแยกหลายเงื่อนไขดังกล่าวจะเป็นเท็จโดยไม่ต้องคำนึงถึงเงื่อนไขเดียวตัวที่สอง ดังนั้นเงื่อนไขเดียวตัวที่สองจึงไม่ต้องถูกนำไปประเมิน

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อใส่ลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้ที่เกี่ยวข้องกับเงื่อนไขการตัดสินใจของการครอบคลุมทางแยกที่ใช้โดยบริษัท N

รูปที่ 1 แสดงโปรแกรมตัวอย่างที่จะถูกทดสอบ และตารางที่ 1 แสดงกรณีทดสอบตัวอย่างสำหรับโปรแกรมนี้ เมื่อโปรแกรมในรูปที่ 1 ถูกทดสอบตามเงื่อนไขการตัดสินใจของการครอบคลุมทางแยกที่ใช้โดยบริษัท N ตามกรณีทดสอบ ผลการทดสอบแสดงให้เห็นว่า A ในกรณีทดสอบ 1 และ B ในกรณีทดสอบ 2

- Program (Integer type: x, Integer type: a, Integer type: b, Integer type: c, Integer type: d)



รูปที่ 1 โปรแกรมตัวอย่างที่จะถูกทดสอบ

ตารางที่ 1 กรณีทดสอบตัวอย่าง

ตัวแปร	ข้อมูลทดสอบ (test data)				
	x	a	b	c	d
กรณีทดสอบ 1	11	9	19	10	10
กรณีทดสอบ 2	11	10	20	11	11

กลุ่มคำตอบสำหรับ A และ B

- a) $b < 20$ ไม่ถูกนำไปประเมิน
- b) $b < 20$ และ $c > 10$ ไม่ถูกนำไปประเมิน
- c) $b < 20$ และ $d > 10$ ไม่ถูกนำไปประเมิน
- d) $c > 10$ ไม่ถูกนำไปประเมิน
- e) $c > 10$ and $d > 10$ ไม่ถูกนำไปประเมิน
- f) $d > 10$ ไม่ถูกนำไปประเมิน
- g) เงื่อนไขเดียวทั้งหมดถูกนำไปประเมิน

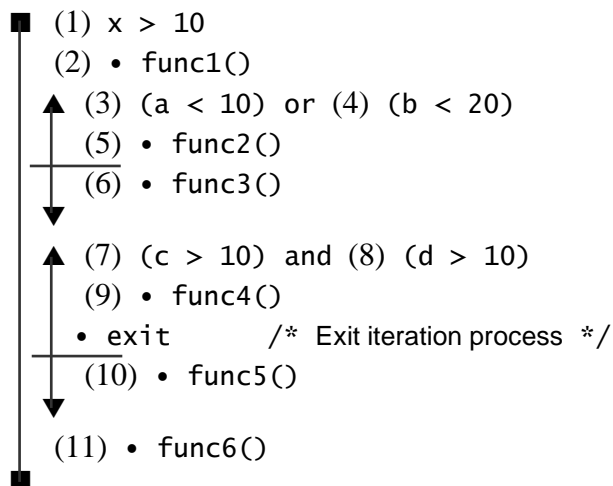
คำถามย่อย 2

โครงสร้างควบคุม (control structure) ของโปรแกรมสามารถเขียนอธิบายได้ด้วยกราฟกระแสควบคุม (control flow graph) ในกราฟกระแสควบคุม กระบวนการต่าง ๆ จะถูกแบ่งออกเป็นชุดคำสั่งที่เชื่อมต่อกันแบบอนุกรม ชุดคำสั่งสำหรับการวนซ้ำ และชุดคำสั่งสำหรับทางแยก โดยแต่ละชุดคำสั่งจะถูกบรรจุอยู่ในบล็อกกระบวนการ ซึ่งจากนี้ไปจะเรียกว่าโหนด (node) โดยเชื่อมต่อกันด้วยเส้นที่เชื่อมระหว่างแต่ละส่วนโดยตรงที่จากนี้ไปจะเรียกว่าเส้นเชื่อม (edge) ตามลำดับการประมวลผลของแต่ละกระบวนการ ในที่นี้ ทางแยกแบบหลายเงื่อนไขจะถูกแบ่งออกเป็นเงื่อนไขเดียวหลายตัวตามลำดับ และจะถูกนำไปใส่ไว้ในกราฟกระแสควบคุม

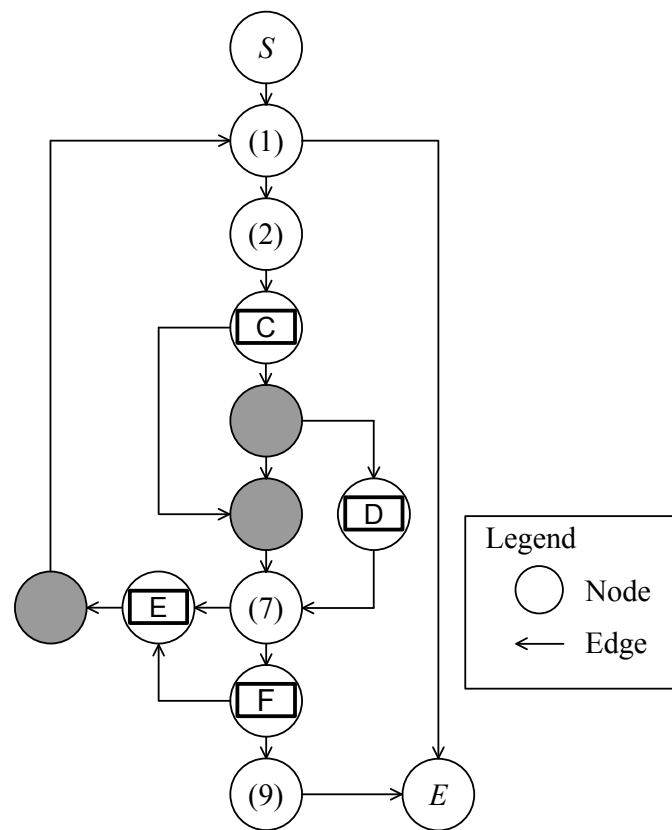
รูปที่ 2 ถูกจัดเตรียมขึ้นด้วยการกำหนดโหนดหมายเลข (1) ถึงหมายเลข (11) ให้กับโปรแกรมตัวอย่างในรูปที่ 1 และรูปที่ 3 แสดงกราฟกระแสควบคุมที่เป็นคู่กับรูปที่ 2 หมายเลขโหนดในรูปที่ 3 เป็นคู่กับหมายเลขโหนดในรูปที่ 2 โดยรูปที่ 3 มีโหนด S และ E เป็นโหนดพิเศษที่แสดงจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรมตามลำดับ และไม่มีกระบวนการที่เกี่ยวข้องอื่นใดในโปรแกรมตัวอย่างที่ต้องได้รับการทดสอบ

จากกลุ่มคำตอบด้านล่าง ให้เลือกหมายเลขโหนดที่ต้องเพื่อนำไปใส่ในช่องว่าง แต่ละช่องในกราฟกระแสควบคุมในรูปที่ 3

○ Program (Integer type: x, Integer type: a, Integer type: b, Integer type: c, Integer type: d)



รูปที่ 2 ตัวอย่างโปรแกรมจากรูปที่ 1 พร้อมหมายเลขโหนด



หมายเหตุ: ส่วนที่แรเงาคือส่วนที่ไม่แสดงรายละเอียดภายใน
รูปที่ 3 กราฟกระแสควบคุมที่เป็นคู่ของโปรแกรมตัวอย่างในรูปที่ 2

กลุ่มคำตอบสำหรับ C ถึง F

- | | | | |
|--------|---------|---------|--------|
| a) (3) | b) (4) | c) (5) | d) (6) |
| e) (8) | f) (10) | g) (11) | |

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อใส่ลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้

เพื่อทำการทดสอบโปรแกรมที่แสดงในรูปที่ 1 ตามกรณีของการครอบคลุมทางแยกที่ใช้โดยบริษัท N นั้น จะต้องใช้กรณีทดสอบอย่างน้อยที่สุดจำนวน G กรณี

นอกจากนี้ ยังมีวิธีการสร้างกรณีทดสอบด้วยการสกัด (extract) เส้นทางต่าง ๆ ออกมาจากกราฟกระแสควบคุม จำนวนเส้นทาง (S) ที่น้อยที่สุดที่ครอบคลุมเส้นขอบและโหนดทั้งหมดของกราฟกระแสควบคุมสามารถหาได้จากนิพจน์ต่อไปนี้

$$S = \text{จำนวนเส้นขอบ} - \text{จำนวนของโหนด} + 2$$

ด้วยการดำเนินการทดสอบสำหรับกรณีทดสอบ S ที่ใช้วิธีการสกัดเส้นทาง จะทำให้มั่นใจได้ว่าสามารถทดสอบได้ครอบคลุมมากกว่าการครอบคลุมทางแยก

เมื่อพิจารณาตามกราฟกระแสควบคุมในรูปที่ 3 ค่าของ S ที่ได้คือ H และเพื่อให้สามารถลดจำนวนบั๊กที่ไม่ได้รับการแก้ไขในโปรแกรมของบริษัทลง บริษัท N จึงตัดสินใจทดสอบโปรแกรมด้วยกรณีทดสอบที่มีพื้นฐานอยู่บนกราฟกระแสควบคุม

กลุ่มคำตอบสำหรับ G และ H

- | | | |
|------|------|------|
| a) 2 | b) 3 | c) 4 |
| d) 5 | e) 6 | f) 7 |

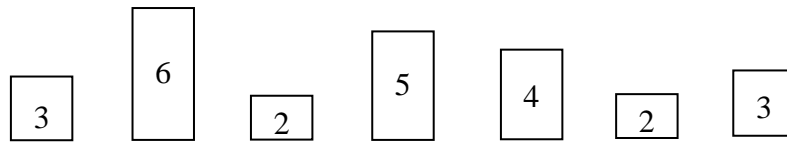
Q6. จงอ่านคำอธิบายของโปรแกรมและโค้ดของโปรแกรม แล้วตอบคำถามย่อยข้อที่ 1 และข้อที่ 2

ปัญหาการจัดของลงใส่ช่องเก็บ (bin-packing problem) เป็นปัญหาคลาสสิกในการหาค่าเหมาะที่สุดเชิงผสมผสาน (combinatorial optimization) โดยมีสินค้าหลายชิ้นที่มีขนาด (size) แตกต่างกันและช่องเก็บแต่ละใบมีความจุ (capacity) เท่ากัน ซึ่งปัญหาคือการกำหนดช่องเก็บสำหรับสินค้าแต่ละรายการ และสินค้าแต่ละรายการจะต้องสามารถบรรจุลงในช่องเก็บได้โดยขนาดรวมของสินค้าทั้งหมดในช่องเก็บใบนั้น ต้องไม่เกินความจุของช่องเก็บ และต้องให้มีจำนวนของช่องเก็บที่ไม่ว่าง (non-empty) น้อยที่สุดอีกด้วย เพื่อแก้ปัญหาดังกล่าว วิธีการแบบฮิวริสติก (heuristic method) จึงได้มีการพัฒนาขึ้นมา ซึ่งวิธีที่เป็นนิยม ได้แก่ Next-Fit, First-Fit, Best-Fit โดยตัวโปรแกรมและตัวอย่างได้ถูกแสดงด้านล่างนี้

[คำอธิบายโปรแกรม]

- (1) N เก็บ จำนวนทั้งหมดของสินค้า (total number of items)
- (2) C เก็บ ความจุของช่องเก็บ (capacity of the bin) โดยช่องเก็บแต่ละใบมีความจุเท่ากัน
- (3) A[i] เก็บขนาด (size) ของสินค้าตัวที่ i และขนาดของสินค้าแต่ละรายการจะไม่ใหญ่กว่าขนาดความจุของช่องเก็บ (bin capacity) C โดยในรูปที่ 1 ได้มีการแสดงตัวอย่างขนาดของสินค้า
- (4) ดัชนี (Index) ของอาร์เรย์จะเริ่มต้นที่ 1
- (5) 3 วิธีการทั้งหมดดังต่อไปนี้ เป็นวิธีการในการนำสินค้าไปวางลงในช่องเก็บ เมื่อพวกมันเข้ามาถึง
 - (a) โปรแกรมย่อย NextFit จะวางสินค้าอันถัดไป (next item) ลงในช่องเก็บปัจจุบัน (current bin) ก็ต่อเมื่อมันมีขนาดสามารถใส่ได้ (fit) แต่ถ้าขนาดของสินค้าไม่สามารถใส่ไปได้ ช่องเก็บปัจจุบันก็จะถูกปิดลง และเริ่มต้นการจัดเก็บในช่องใหม่ก็จะเริ่มต้นขึ้น ในรูปที่ 2 ได้แสดงถึงผลลัพธ์ในการประมวลผลด้วย NextFit โดยใช้สินค้าจากรูปที่ 1
 - (b) โปรแกรมย่อย FirstFit จะวางสินค้าอันถัดไป (next item) ลงในช่องเก็บที่มีหมายเลขน้อยที่สุดที่สามารถวางสินค้าชิ้นนั้นได้ แต่ถ้าหากไม่มีที่ให้วางในช่องเก็บใด ๆ ที่เปิดอยู่ ก็จะเริ่มจัดเก็บในช่องใหม่ต่อไป ในรูปที่ 3 ได้แสดงถึงผลลัพธ์ในการประมวลผลด้วย FirstFit โดยใช้สินค้าจากรูปที่ 1
 - (c) โปรแกรมย่อย BestFit จะวางสินค้าอันถัดไป (next item) ลงในช่องเก็บที่จะทำให้เหลือพื้นที่ว่างหลังจากที่มีการวางสินค้านั้น ๆ ลงไปแล้ว น้อยที่สุด แต่ถ้าขนาดของสินค้าไม่สามารถใส่ลงในช่องเก็บใด ๆ ที่เปิดอยู่ได้ ก็จะเริ่มจัดเก็บในช่องใหม่ต่อไป รูปที่ 4 ได้แสดงถึงผลลัพธ์ในการประมวลผลด้วย BestFit โดยใช้สินค้าจากรูปที่ 1
- (6) B[i] เก็บค่าผลรวมของ (sum of the size) สินค้าทั้งหมดที่ถูกวางลงในช่องเก็บหมายเลข i
- (7) S[] คือ คำตอบของการแก้ไขปัญหา (solution) โดยที่ S[i] จะเก็บหมายเลขช่องเก็บที่สินค้า i นำไปวางไว้
- (8) โปรแกรมย่อย OutputResult จะแสดงค่าสมาชิกในอาร์เรย์ S[]

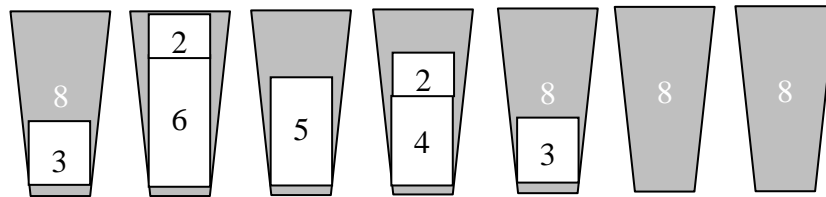
สินค้า:



	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
ค่า	3	6	2	5	4	2	3

Figure 1 ตัวอย่างของสินค้า (item) ใช้สำหรับโปรแกรม

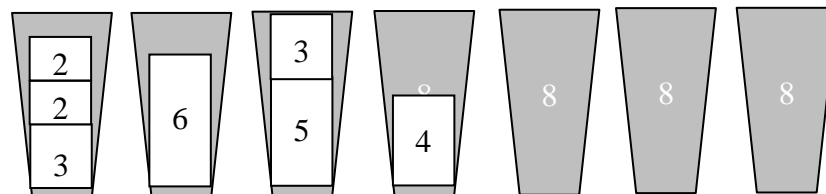
Next-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
ค่า	3	8	5	6	3	0	0

Figure 2 ผลลัพธ์จากการประมวลผล ของโปรแกรม nextFit (C=8)

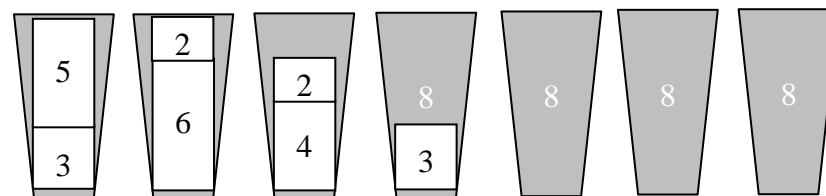
First-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
ค่า	7	6	8	4	0	0	0

Figure 2 ผลลัพธ์จากการประมวลผล ของโปรแกรม FirstFit (C=8)

Best-Fit:



	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]
ค่า	8	8	6	3	0	0	0

Figure 2 ผลลัพธ์จากการประมวลผล ของโปรแกรม BestFit (C=8)

[Program]

- Global: Integer type: $N \leftarrow 7, C \leftarrow 8$
- Global: Integer type: $A[N] \leftarrow \{3, 6, 2, 5, 4, 2, 3\}$
- Global: Integer type: $B[N], S[N]$

- Subprogram: OutputResult()
- Integer type: I

```

I: 1, I ≤ N, 1
• print(S[I]) /* output S[I] */
• print(" ") /* output " " */

```

- Subprogram: NextFit()
- Integer type: I, J

```

■ I: 1, I ≤ N, 1
  • B[I] ← 0
■
  • J ← 1
■ I: 1, I ≤ N, 1
  ▲ C - B[J] < A[I]
  │
  │ • J ← J + 1
  ▼
  • S[I] ← J
  • B[J] ← B[J] + A[I]
■
  • OutputResult()

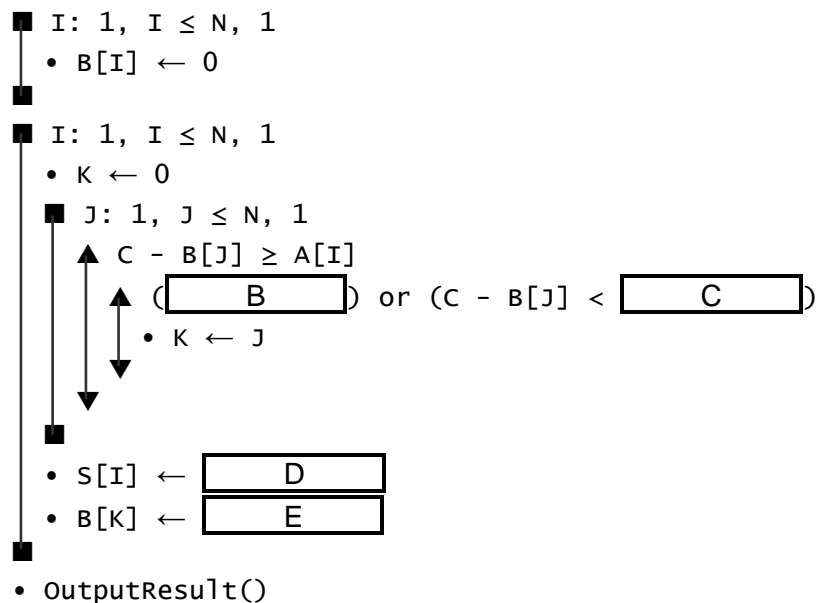
```

- Subprogram: FirstFit()
- Integer type: I, J

```
graph TD
    Node1["I: 1, I ≤ N, 1  
• B[I] ← 0"]
    Node2["I: 1, I ≤ N, 1"]
    Node3["J: 1, J ≤ N, 1"]
    Node4["C - B[J] ≥ A[I]  
• S[I] ← A  
• B[J] ← B[J] + A[I]  
• J ← N + 1"]
    Node5["OutputResult()"]

    Node1 --> Node2
    Node2 --> Node3
    Node3 --> Node4
    Node4 --> Node3
    Node4 --> Node5
```

- Subprogram: BestFit()
- Integer type: I, J, K



คำถามย่อย 1

จากกลุ่มคำตอบด้านล่างนี้ จงเลือกคำตอบที่ถูกต้องจะถูกเติมลงในแต่ละช่องว่าง ของโปรแกรมด้านบน

กลุ่มคำตอบสำหรับ A และ B

- | | | |
|----------------|-------------|----------------|
| a) A[I] | b) I | c) J |
| d) K | e) S[I] + 1 | f) S[I] + A[I] |
| g) S[I] + B[J] | h) S[I] + J | |

กลุ่มคำตอบสำหรับ B

- | | | |
|----------|----------|----------|
| a) I = J | b) I ≠ J | c) K = 0 |
| d) K = N | e) K > 0 | f) K > J |

กลุ่มคำตอบสำหรับ C

- | | | |
|-------------|-------------|-------------|
| a) A[J] | b) A[K] | c) B[J] |
| d) B[K] | e) C | f) C - A[I] |
| g) C - B[J] | h) C - B[K] | |

กลุ่มคำตอบสำหรับ E

- | | | |
|----------------|----------------|----------------|
| a) B[J] + A[I] | b) B[J] + B[K] | c) B[K] + A[I] |
| d) B[K] + A[J] | | |

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ในคำอธิบายต่อไปนี้
ในที่นี้สมมติว่าคำตอบที่ถูกต้องในคำถามย่อยที่ 1 ได้ถูกเติมลงในช่องว่างทั้งหมดแล้วใน [Program]
และถ้าจำเป็น สามารถเลือกคำตอบเดิมได้สองครั้งหรือมากกว่าก็ได้

เพื่อที่จะจำลอง (simulate) กรณีอื่น ๆ โปรแกรมใน 3 บรรทัดแรกจะถูกแทนที่ด้วย 3 บรรทัดต่อไปนี้
และหลังจากนั้นโปรแกรมย่อยต่าง ๆ (subprograms) จะถูกประมวลผล

- Global: Integer type: $N \leftarrow 10, C \leftarrow 10$
- Global: Integer type: $A[N] \leftarrow \{4, 8, 5, 7, 6, 1, 4, 2, 2, 1\}$
- Global: Integer type: $B[N], S[N]$

ตารางด้านล่างนี้ แสดงผลลัพธ์จากการประมวลผล (Execution Result)

โปรแกรมย่อย (subprogram)	ผลลัพธ์ที่แสดงผล (Printed result)
NextFit()	<input type="text" value="F"/>
FirstFit()	<input type="text" value="G"/>
BestFit()	<input type="text" value="H"/>

กลุ่มคำตอบสำหรับ F ถึง H

- a) 1 2 1 2 3 4 3 4 4 5
- b) 1 2 1 3 4 1 4 2 3 3
- c) 1 2 1 3 4 3 4 2 3 2
- d) 1 2 3 4 5 5 6 6 6 6
- e) 1 2 3 4 5 6 7 8 9 10
- f) 4 8 5 7 7 9 0 0 0 0
- g) 10 10 10 10 0 0 0 0 0 0

จากคำถามข้อ Q7 และ Q8 จงเลือกตอบเพียงหนึ่งข้อ
และให้ระบายสีคำตอบที่เลือกทำลงในช่อง ๕ ของกระดาษคำตอบ
หากทำทั้งสองข้อ จะให้คะแนนเฉพาะข้อแรกเท่านั้น

Q7. จงอ่านรายละเอียดของโปรแกรมภาษา C และตัวโปรแกรมต่อไปนี้ แล้วตอบคำถามย่อย

โปรแกรมนี้เป็นการคำนวณแบบอาร์มสตรอง ที่ทำหน้าที่ตรวจสอบตัวเลขนำเข้าว่าเป็นตัวเลขอาร์มสตรองหรือไม่ ตัวเลขอาร์มสตรองคือ ตัวเลขจำนวน n หลัก (n -digit number) ที่มีค่าเท่ากับผลรวมของตัวเลขในแต่ละหลักที่ยกกำลัง n

ตัวอย่างเช่น

$$371 = 3^3 + 7^3 + 1^3$$

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

[รายละเอียดโปรแกรม]

- (1) โปรแกรมรับค่าตัวเลขที่เป็นเลขฐานสิบ
- (2) ตัวเลขนำเข้าเป็นจำนวนเต็มบวก ที่มีจำนวนหลักสูงสุด 8 หลัก หรือน้อยกว่า เนื่องจากผลรวมของตัวเลขแต่ละหลักยกกำลัง 8 ยังมีขนาดไม่เกินกว่าค่าสูงสุดที่ค่าจำนวนเต็มขนาดยาว (long integer) สามารถรองรับได้
- (3) โปรแกรมคำนวณจำนวนของหลัก
- (4) โปรแกรมคำนวณผลรวมของแต่ละหลักที่ยกกำลังด้วยเลขจำนวนของหลักทั้งหมด หากผลรวมมีค่าเท่ากับเลขนำเข้า ตัวเลขนั้นเป็นตัวเลขอาร์มสตรอง
- (5) ฟังก์ชัน 3 ฟังก์ชันต่อไปนี้ถูกใช้ในโปรแกรม
 - (i) long calc_power(int base, int power)
เป็นฟังก์ชันที่ทำหน้าที่คำนวณค่ายกกำลังของตัวเลข
 - (ii) int number_digits(long input_number)
เป็นฟังก์ชันที่ทำหน้าที่คำนวณจำนวนของหลัก
 - (iii) int check_armstrong (long n_input)
เป็นฟังก์ชันที่คำนวณผลรวมของตัวเลขที่นำเข้าและตรวจสอบว่าตัวเลขนั้นเป็นตัวเลขอาร์มสตรองหรือไม่
- (6) ต่อไปนี้เป็นตัวอย่างผลลัพธ์ของโปรแกรมการคำนวณจำนวนอาร์มสตรอง

```
Enter a number:
8208
8208 has 4 digits.
8208 is an Armstrong numbers.
```

[Program]

```
#include <stdio.h>

long calc_power(int, int);
int number_digits(long);
int check_armstrong(long);

int main () {
    long n_input;

    printf("Enter a number:\n");
    scanf("%ld", &n_input);

    printf("%ld has %d digits.\n", n_input, number_digits(n_input));

    if (check_armstrong(n_input))
        printf("%ld is an Armstrong number.\n", n_input);
    else
        printf("%ld is not an Armstrong number.\n", n_input);
    return 0;
}

long calc_power(int base, int power) {
    int i;
    long pw = 1;

    for (i = 0; ; i++)
        pw = pw * base;

    return pw;
}

int number_digits(long input_number) {
    int n_digits = 0;

    while () {
        n_digits++;
        ;
    }

    return n_digits;
}
```

```

int check_armstrong(long n_input) {
    long tmp, sum = 0;
    int remainder, digits;

    digits = number_digits(n_input);
    tmp = n_input;

    while (tmp != 0) {
        remainder = tmp % 10;
        sum = sum + ;
        tmp = tmp / 10;
    }

    return ;
}

```

คำถามย่อย

จากกลุ่มคำตอบต่อไปนี้ จงเลือกคำตอบที่ถูกต้อง เพื่อระบุลงในช่องว่าง แต่ละช่องในโปรแกรมด้านบน

กลุ่มคำตอบสำหรับ A

- a) $i < \text{base} + \text{power}$
- b) $i < \text{base} * \text{power}$
- c) $i < \text{power}$
- d) $i \leq \text{base} + \text{power}$
- e) $i \leq \text{base} * \text{power}$
- f) $i \leq \text{power}$

กลุ่มคำตอบสำหรับ B

- a) $\text{input_number} \neq 0$
- b) $\text{input_number} \neq 1$
- c) $\text{input_number} == 0$
- d) $\text{input_number} == 1$
- e) $\text{n_digits} < \text{input_number}$
- f) $\text{n_digits} \leq \text{input_number}$

กลุ่มคำตอบสำหรับ C

- a) `input_number = input_number + 10`
- b) `input_number = input_number - 10`
- c) `input_number = input_number * 10`
- d) `input_number = input_number / 10`
- e) `input_number = input_number % 10`

กลุ่มคำตอบสำหรับ D

- a) `calc_power(digits, remainder)`
- b) `calc_power(digits, remainder) * calc_power(n_input, remainder)`
- c) `calc_power(n_input, remainder)`
- d) `calc_power(remainder, digits)`
- e) `calc_power(remainder, n_input)`
- f) `calc_power(remainder, n_input) * calc_power(remainder, digits)`

กลุ่มคำตอบสำหรับ E

- a) `digits != sum`
- b) `digits == sum`
- c) `n_input != sum`
- d) `n_input - sum`
- e) `n_input == sum`
- f) `sum`
- g) `sum - n_input`

Q8. อ่านคำอธิบายของโปรแกรมภาษาจาวาและตัวโปรแกรมต่อไปนี้แล้วตอบคำถามย่อย

[คำอธิบายโปรแกรม]

วิชาที่เรียนออนไลน์วิชาหนึ่งจะมอบเครื่องหมายสัญลักษณ์ (Badge) ให้กับนักศึกษาตามคะแนนที่นักศึกษาได้รับเพื่อกระตุ้นให้นักศึกษาดังใจเรียน โดยนักศึกษาจะต้องทำแบบทดสอบและตอบคำถามปรนัย 4 ตัวเลือก (A, B, C และ D) จำนวน 10 ข้อ โดยแต่ละข้อที่ตอบถูกจะได้รับ 50 คะแนน

GOLD 500	SILVER 300	BRONZE 100
-------------	---------------	---------------

เครื่องหมายสัญลักษณ์แบ่งออกได้เป็น "GOLD" "SILVER" และ "BRONZE" ซึ่งมีคะแนนเป็น 500, 300, และ 100 ตามลำดับ ถ้านักศึกษาได้คะแนนอย่างน้อย 100 คะแนนแล้ว เครื่องหมายสัญลักษณ์ BRONZE จะปรากฏ ถ้าได้คะแนนอย่างน้อย 300 คะแนน เครื่องหมายสัญลักษณ์ SILVER ก็จะปรากฏ และถ้าได้คะแนน 500 คะแนน เครื่องหมายสัญลักษณ์ GOLD ก็จะปรากฏ

ขั้นตอนของการแสดงผลลัพธ์ของโปรแกรมเป็นดังนี้

Quiz:

Question 1...

Question 2...

Question 3...

Question 4...

Question 5...

Question 6...

Question 7...

Question 8...

Question 9...

Question 10...

Student: ID: 123, Name: Thomas Anderson

Answers: A B C A B B C D C A

Badge earned: SILVER 300

[โปรแกรม 1]

```
public  Badge {  
    GOLD(500), SILVER(300), BRONZE(100), NONE(0);  
    private int value;  
  
    private Badge(int value) {  
        this.value = value;  
    }  
  
    public int getValue(){  
        return value;  
    }  
  
    public static Badge getBadge(int score) {  
        for (Badge badge : values()) {  
            if (score >= badge.getValue())  
                return badge;  
        }  
        return NONE;  
    }  
}
```

[โปรแกรม 2]

```
public class Student {  
    private int id;  
    private String name;  
    private Badge earnedBadge;  
    private char[] answers;  
  
    public Student(int id, String name,  answers) {  
        this.id = id;  
        this.name = name;  
        this.answers = answers;  
    }  
  
    public  getAnswers() {  
        return answers;  
    }  
  
    public void setBadge(Badge earnedBadge) {  
        this.earnedBadge = earnedBadge;  
    }  
}
```

```

public Badge getBadge() {
    return earnedBadge;
}

public String toString() {
    String str = String.format("ID: %s, Name: %s%n Answers: ",
                               id, name);
    for (char answer : answers) {
        str += answer + " ";
    }
    str += String.format("%n Badge earned: %s %d", );
    return str;
}
}

```

[โปรแกรม 3]

```

public class Quiz {
    private String[] questions = new String[10];
    private char[] answers = new char[10];

    public Quiz(String[] questions, char[] answers) {
        this.questions = questions;
        this.answers = answers;
    }

    public void setAnswers(char[] answers) {
        this.answers = answers;
    }

    public char[] getAnswers() {
        return answers;
    }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < questions.length; i++)
            sb.append(questions[i]).append("\n");
        return sb.toString();
    }
}

```

[โปรแกรม 4]

```
public class Badges {

    public static int checkQuiz(char[] correctAnswer,
                                char[] studentAnswer ) {
        int score = 0;
        for (int i = 0; i<correctAnswer.length; i++)
            if (correctAnswer[i] == studentAnswer[i])
                ;
        return score;
    }

    public static void main(String[] args) {
        String[] questions = {"Question 1...", "Question 2...",
                               "Question 3...", "Question 4...",
                               "Question 5...", "Question 6...",
                               "Question 7...", "Question 8...",
                               "Question 9...", "Question 10..."};
        char[] answers = {'A', 'B', 'C', 'A', 'D',
                           'B', 'C', 'D', 'D', 'A'};
        Quiz quiz = new Quiz(questions, answers);
        System.out.println("Quiz:\n" + quiz.toString());

        int id = 123;
        String name = "Thomas Anderson";
        char[] studentAnswer = {'A', 'B', 'C', 'A', 'B',
                                'B', 'C', 'D', 'C', 'A'};
        Student student = new Student(id, name, studentAnswer);

        int score = ;
        student.setBadge(Badge.getBadge(score));
        System.out.println("Student: " + student.toString());
    }
}
```

คำถามย่อย

จากกลุ่มคำตอบด้านล่างนี้ จงเลือกคำตอบที่ถูกต้องที่จะเติมลงในช่องว่าง แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ A

- | | |
|-------------|--------------|
| a) abstract | b) class |
| c) enum | d) interface |

กลุ่มคำตอบสำหรับ B

- | | |
|------------|-----------|
| a) Answers | b) char |
| c) char[] | d) String |

กลุ่มคำตอบสำหรับ C

- a) earnedBadge
- b) earnedBadge, earnedBadge.getValue()
- c) earnedBadge.getValue()
- d) earnedBadge.getValue(), earnedBadge

กลุ่มคำตอบสำหรับ D

- | | |
|----------------|----------------|
| a) score *= 10 | b) score *= 50 |
| c) score += 10 | d) score += 50 |
| e) score ^= 10 | f) score ^= 50 |

กลุ่มคำตอบสำหรับ E

- a) checkQuiz(Quiz.getAnswers(), Student.getAnswers())
- b) checkQuiz(Quiz.getAnswers(), student.getAnswers())
- c) checkQuiz(quiz.getAnswers(), Student.getAnswers())
- d) checkQuiz(quiz.getAnswers(), student.getAnswers())