



October / November 2021

Fundamental IT Engineer Examination (Afternoon)

ให้ทำข้อสอบตามรายละเอียดต่อไปนี้

หมายเลขคำถาม	Q1	Q2 – Q5	Q6	Q7, Q8
การเลือกคำถาม	คำถามบังคับ	เลือก 2 ใน 4	คำถามบังคับ	เลือก 1 ใน 2
เวลาสอบ	13:30 – 16:00 (150 นาที)			

ข้อปฏิบัติ:

1. ให้ใช้ดินสอตอบ ถ้าต้องการเปลี่ยนคำตอบ ให้ลบคำตอบเก่าให้สะอาดก่อนโดยไม่ให้มีคราบยางลบหลงเหลือ
2. ให้ทำเครื่องหมายบอกข้อมูลผู้สอบและคำตอบของแบบทดสอบ ตามคำสั่งด้านล่างอย่างเคร่งครัด หากทำเครื่องหมายไม่เหมาะสม คำตอบของท่านอาจไม่ได้รับการตรวจ ห้ามทำเครื่องหมาย หรือเขียนตอบนอกพื้นที่ที่กำหนดไว้

(1) หมายเลขผู้สอบ (Examinee Number)

ให้เขียนหมายเลขผู้สอบลงในช่องที่เตรียมไว้ให้ และทำเครื่องหมายในช่องว่างที่เหมาะสมที่อยู่ใต้ตัวเลขแต่ละตัว

(2) วันเกิด (Date of Birth)

ให้เขียนวันเกิดของผู้สอบ (เป็นตัวเลข) ลงในช่องที่เตรียมไว้ ให้ตรงกับที่พิมพ์อยู่ในบัตรเข้าห้องสอบ และทำเครื่องหมายในช่องว่างที่เหมาะสมที่อยู่ใต้ตัวเลขแต่ละตัว

(3) การเลือกคำตอบ

สำหรับ Q2 ถึง Q5 และ Q7 กับ Q8 ให้เลือก (S) ของข้อที่คุณเลือกที่จะตอบในช่อง "Selection Column" บนกระดาษคำตอบของคุณ

(4) คำตอบ

ให้ทำเครื่องหมายตรงคำตอบที่เลือกตามตัวอย่างที่แสดงอยู่ด้านล่าง

[คำถามตัวอย่าง]

ข้อใดต่อไปนี้เป็นสิ่งที่ควรใช้ทำเครื่องหมายเพื่อเลือกข้อที่ต้องการในกระดาษคำตอบ

กลุ่มคำตอบ

- a) ปากกาลูกลื่น b) สีเทียน c) ปากกาหมึกซึม d) ดินสอ

เนื่องจากคำตอบที่ถูกคือ "d)" (ดินสอ), ดังนั้นให้ทำเครื่องหมายดังแสดงด้านล่างนี้:

[ตัวอย่างคำตอบ]

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

ห้ามเปิดดูข้อสอบก่อนได้รับอนุญาต

ข้อสงสัยที่เกี่ยวข้องกับคำถามในข้อสอบอาจจะไม่ถูกตอบ

Notations used in the pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated:

[Declaration, comment, and process]

Notation		Description
<i>type</i> : <i>var1</i> , ..., <i>array1</i> [], ...		Declares variables <i>var1</i> , ..., and/or arrays <i>array1</i> [], ..., by data <i>type</i> such as INT and CHAR.
FUNCTION: <i>function</i> (<i>type</i> : <i>arg1</i> , ...)		Declares a <i>function</i> and its arguments <i>arg1</i> ,
/* comment */		Describes a comment.
Process	<i>variable</i> ← <i>expression</i> ;	Assigns the value of the <i>expression</i> to the <i>variable</i> .
	<i>function</i> (<i>arg1</i> , ...);	Calls the <i>function</i> by passing / receiving the arguments <i>arg1</i> ,
	IF (<i>condition</i>) { <i>process1</i> } ELSE { <i>process2</i> }	Indicates the selection process. If the <i>condition</i> is true, then <i>process1</i> is executed. If the <i>condition</i> is false, then <i>process2</i> is executed, when the optional ELSE clause is present.
	WHILE (<i>condition</i>) { <i>process</i> }	Indicates the “WHILE” iteration process. While the <i>condition</i> is true, the <i>process</i> is executed repeatedly.
	DO { <i>process</i> } WHILE (<i>condition</i>);	Indicates the “DO - WHILE” iteration process. The <i>process</i> is executed once, and then while the <i>condition</i> is true, the <i>process</i> is executed repeatedly.
	FOR (<i>init</i> ; <i>condition</i> ; <i>incr</i>) { <i>process</i> }	Indicates the “FOR” iteration process. While the <i>condition</i> is true, the <i>process</i> is executed repeatedly. At the start of the first iteration, the process <i>init</i> is executed before testing the <i>condition</i> . At the end of each iteration, the process <i>incr</i> is executed before testing the <i>condition</i> .

[Logical constants]

true, false

[Operators and their precedence]

[illegible]

Note: With division of integers, an integer quotient is returned as a result.

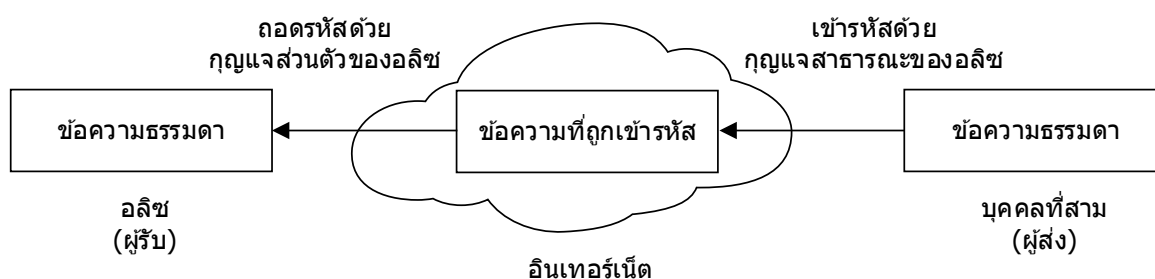
The “%” operator indicates a remainder operation.

คำถาม Q1 เป็นคำถามบังคับ

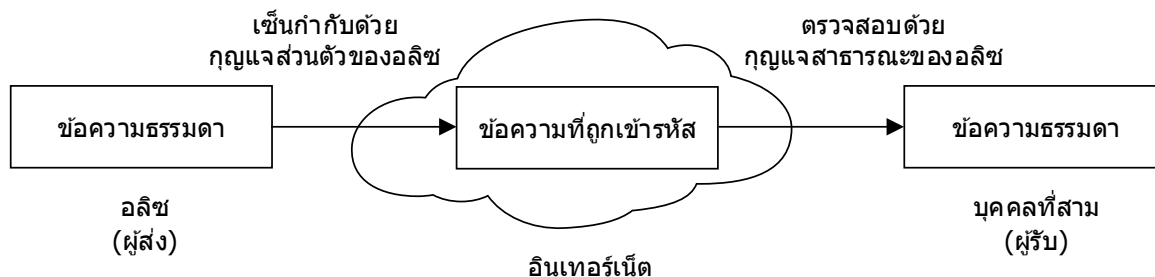
Q1. อ่านคำอธิบายเกี่ยวกับการเข้ารหัสลับกุญแจสาธารณะต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

การเข้ารหัสลับกุญแจสาธารณะ (public key cryptography) เป็นวิธีหนึ่งที่ถูกใช้เพื่อสร้างความปลอดภัยในการสื่อสารทางอีเมลอย่างกว้างขวาง วิธีนี้ใช้คู่ของกุญแจ (key) เพื่อเข้ารหัสและเข้ารหัสซ้ำกับข้อความ โดยกุญแจคู่เดียวกันนั้นสามารถใช้ในทางกลับกันเพื่อถอดรหัสและยืนยันข้อความอีเมลได้เช่นกัน การเข้ารหัสลับกุญแจสาธารณะใช้กุญแจสองชนิด: กุญแจส่วนตัว (private key) ที่ต้องถูกเก็บรักษาไว้เป็นความลับ และกุญแจสาธารณะ (public key) ที่สามารถถูกเผยแพร่ต่อสาธารณะ รูปที่ 1 แสดงตัวอย่างการทำงานโดยทั่วไปของการเข้ารหัสลับกุญแจสาธารณะ การเข้ารหัสลับข้อความ และการเข้ารหัสซ้ำเป็นดิจิทัล

[การเข้ารหัสลับข้อความ]



[ลายเซ็นดิจิทัล]



รูปที่ 1 ตัวอย่างการทำงานโดยทั่วไปของการเข้ารหัสลับกุญแจสาธารณะ

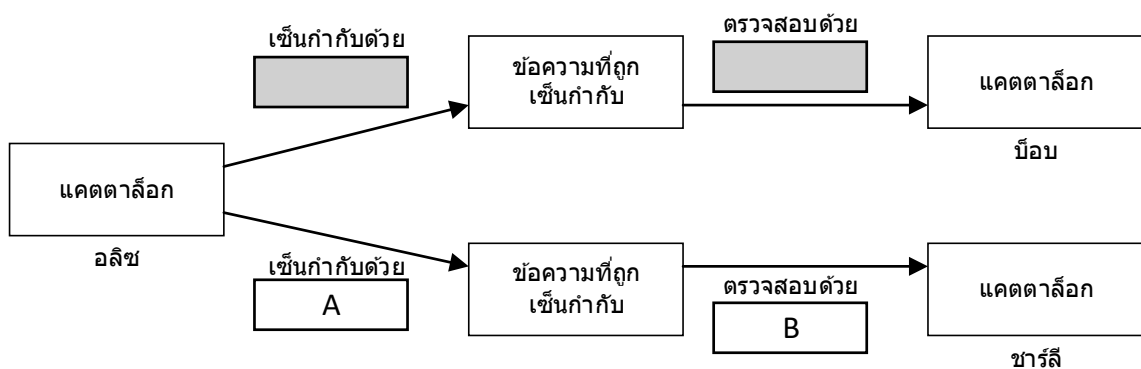
ในสถานการณ์ที่แสดงในรูปที่ 1 อลิซได้เผยแพร่กุญแจสาธารณะพร้อมกับที่อยู่อีเมลของเธอไว้บนอินเทอร์เน็ต เมื่อบุคคลที่สามต้องการสื่อสารกับอลิซ ก็สามารถใช้กุญแจสาธารณะของอลิซมาเข้ารหัสอีเมลแล้วส่งให้อลิซ ซึ่งสามารถใช้กุญแจส่วนตัวของเธอเพื่อถอดรหัสลับอีเมลให้ได้ข้อความต้นฉบับกลับมาได้ และในทางกลับกัน อลิซก็สามารถใช้กุญแจส่วนตัวของเธอเพื่อเซ็นกำกับข้อความก่อนส่งให้บุคคลที่สามได้ จากนั้น บุคคลที่สามดังกล่าว ก็สามารถใช้กุญแจสาธารณะของอลิซเพื่อตรวจสอบความถูกต้องของข้อความนั้นได้

กระบวนการเซ็นกำกับนี้เป็นการยืนยันความถูกต้องสมบูรณ์แต่ไม่ได้ปกป้องความลับ เนื่องจากกุญแจสาธารณะของอลิซได้ถูกเผยแพร่ไว้บนอินเทอร์เน็ตเพื่อให้ทุกคนที่เข้าถึงกุญแจสาธารณะของอลิซสามารถตรวจสอบข้อความที่ถูกเซ็นกำกับมาได้

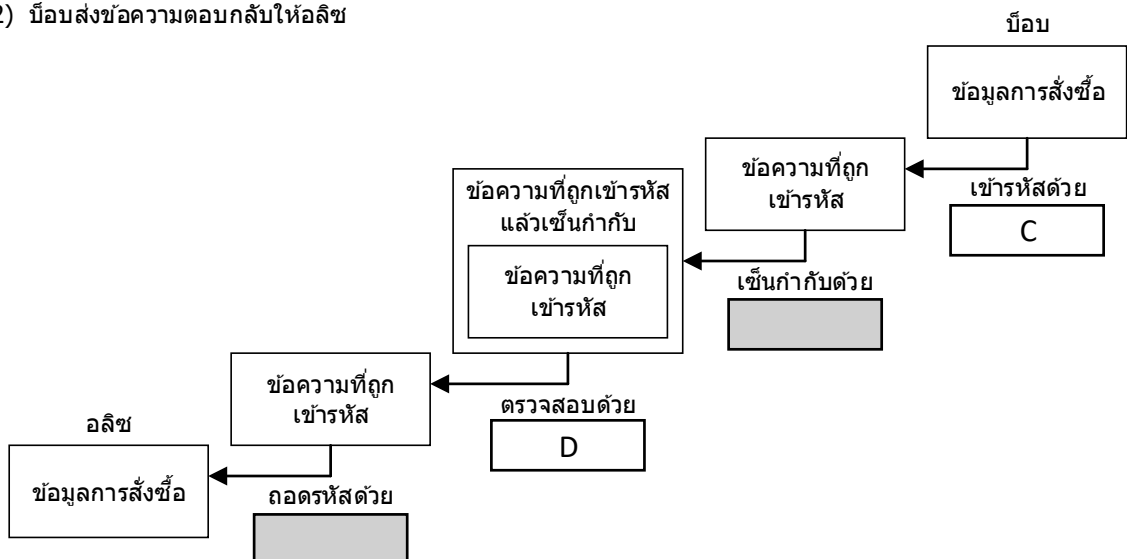
รูปที่ 2 แสดงขั้นตอนการสื่อสารที่ใช้การเข้ารหัสลับกุญแจสาธารณะระหว่างอลิซ บ๊อบ และชาร์ลี

- (1) อลิซส่งแคตตาล็อกสินค้าไปให้บ๊อบและชาร์ลี เนื่องจากข้อมูลในแคตตาล็อกไม่ได้เป็นความลับ ดังนั้นอีเมลจึงไม่ได้ถูกเข้ารหัสลับ อย่างไรก็ตาม ข้อความที่ส่งนั้นต้องถูกเซ็นกำกับเพื่อยืนยันความถูกต้องสมบูรณ์และเพื่อไม่ให้ปฏิเสธความรับผิดชอบ (non-repudiation) ต่อเนื้อหาและข้อมูลต่าง ๆ ในแคตตาล็อกได้
- (2) หลังจากได้รับแคตตาล็อกแล้ว บ๊อบต้องการสั่งซื้อสินค้า ดังนั้น เขาจึงตอบกลับข้อความถึงอลิซ และเนื่องจากข้อความนี้มีข้อมูลอ่อนไหวทางการเงิน ข้อความนี้จึงต้องถูกเข้ารหัสลับเพื่อให้มั่นใจได้ว่าอลิซจะเป็นเพียงผู้เดียวที่สามารถอ่านข้อความได้ และข้อความนี้ยังถูกเซ็นกำกับอีกด้วย เพื่อให้อลิซสามารถยืนยันได้ว่าบ๊อบเป็นผู้สั่งซื้อจริง

- (1) อลิซส่งแคตตาล็อกให้บ๊อบและชาร์ลี



- (2) บ๊อบส่งข้อความตอบกลับให้อลิซ



หมายเหตุ: ส่วนที่ถูกแรเงาถูกซ่อนไว้

รูปที่ 2 ขั้นตอนการสื่อสารที่ใช้การเข้ารหัสลับกุญแจสาธารณะ

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในรูปที่ 2 สามารถใช้คำตอบเดียวกันตอบได้มากกว่าหนึ่งครั้งหากจำเป็น

กลุ่มคำตอบสำหรับ A ถึง D

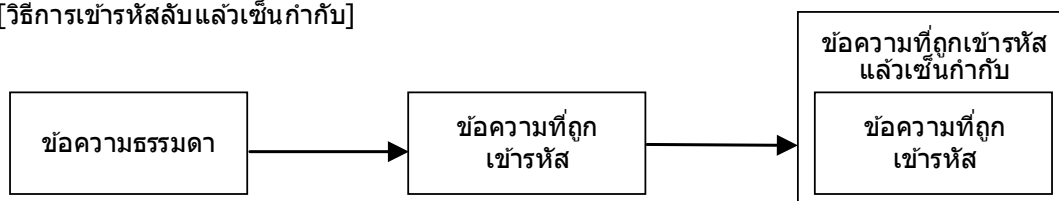
- | | |
|-------------------------|-------------------------|
| a) กฎแฉส่วนตัวของอลิซ | b) กฎแฉสาธารณะของอลิซ |
| c) กฎแฉส่วนตัวของบ๊อบ | d) กฎแฉสาธารณะของบ๊อบ |
| e) กฎแฉส่วนตัวของชาร์ลี | f) กฎแฉสาธารณะของชาร์ลี |

คำถามย่อย 2

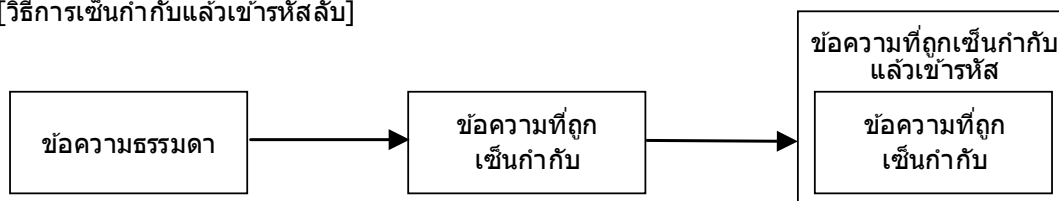
จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้

นอกจากการใช้กฎแฉให้ถูกต้องกับฟังก์ชันที่ต้องการแล้ว ลำดับในการประมวลผลก็มีความสำคัญเช่นกัน โปรแกรมอีเมลบางตัวอนุญาตให้ผู้ใช้เลือกได้ว่าต้องการเข้ารหัสลับข้อความก่อนแล้วจึงเซ็นกำกับในภายหลัง หรือให้ดำเนินการในทางกลับกัน รูปที่ 3 แสดงวิธีการเข้ารหัสลับแล้วเซ็นกำกับและวิธีการเซ็นกำกับแล้วเข้ารหัสลับ

[วิธีการเข้ารหัสลับแล้วเซ็นกำกับ]



[วิธีการเซ็นกำกับแล้วเข้ารหัสลับ]



รูปที่ 3 วิธีการเข้ารหัสลับแล้วเซ็นกำกับและวิธีการเซ็นกำกับแล้วเข้ารหัสลับ

ลำดับในการเซ็นกำกับและการเข้ารหัสลับในแต่ละวิธีมีข้อดีที่แตกต่างกัน ตัวอย่างเช่น หากข้อความถูกเข้ารหัสลับก่อนแล้วจึงเซ็นกำกับ ก็ทำให้ E และในทางกลับกัน หากข้อความถูกเซ็นกำกับก่อน ก็จะช่วย F ดังนั้น ผู้ใช้จึงควรเลือกวิธีการให้เหมาะสมตามความต้องการ

กลุ่มคำตอบสำหรับ E และ F

- a) ตรวจจับและละทิ้งข้อความที่ไม่ได้มาจากผู้ส่งตัวจริงได้แน่นอน ๆ
- b) มั่นใจได้ว่าจะสามารถกู้คืนกุญแจส่วนตัวได้หากเสียกุญแจต้นฉบับไป
- c) ซ่อนร่องรอยและลักษณะของกุญแจส่วนตัวจากผู้ไม่เกี่ยวข้องได้
- d) เพิ่มความเข้มแข็งของการเข้ารหัสลับได้
- e) ลดเวลาที่ต้องใช้ในการเซ็นกำกับข้อความได้อย่างมาก

สำหรับข้อสอบข้อที่ Q2 ถึง Q5 ให้เลือกทำเพียงสองในสี่ข้อ
 จากนั้น ให้ระบายทับในวงกลม (S) ในกระดาษคำตอบสำหรับข้อที่เลือกทำ
 หากเลือกตั้งแต่สามข้อขึ้นไป จะตรวจให้คะแนนเฉพาะสองข้อแรกเท่านั้น

- Q2.** อ่านคำอธิบายเกี่ยวกับรหัสเกรย์ (Gray code) ต่อไปนี้ แล้วตอบคำถามย่อย 1 ถึง 3
 ในคำถามนี้ สัญลักษณ์ "•", "+", "⊕", และ "−" คือ ตัวดำเนินการทางตรรกศาสตร์ AND, OR, exclusive OR, และ NOT ตามลำดับ

รหัสเกรย์ (Gray code) คือรหัสซึ่งมีระยะทางแฮมมิง (Hamming distance) ระหว่างสองรหัสใด ๆ ที่อยู่ติดกันมีค่าเป็น 1 และระยะทางแฮมมิงคือตัวเลขที่แสดงถึงจำนวนของตำแหน่งที่มีค่าต่างกันเมื่อนำเลขฐานสองจำนวน 2 ชุดที่มีจำนวนบิตเท่ากันมาเปรียบเทียบกัน ตัวอย่างเช่น ระยะทางแฮมมิงระหว่างเลขฐานสอง 1001 กับ 1010 คือ A จากตารางที่ 1 นั้น รหัสไบนารีสองค่าที่อยู่ติดกันที่มีระยะทางแฮมมิงมากที่สุดมีค่าเป็นเลขฐานสิบคือ B และมีระยะทางแฮมมิงเป็น C

สำหรับเลขฐานสิบ 0, 1, 2 และ 3 จะมีรหัสไบนารีเป็น 0000, 0001, 0010 และ 0011 และมีรหัสเกรย์เป็น 0000, 0001, 0011 และ 0010 ตามลำดับ ดังแสดงในตารางที่ 1

ตารางที่ 1 ตารางแสดงของรหัสไบนารีขนาด 4 บิต และรหัสเกรย์ขนาด 4 บิต

เลขฐานสิบ (Decimal value)	รหัสไบนารี (Binary code)	รหัสเกรย์ (Gray code)
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมในช่องว่าง แต่ละช่องในคำอธิบายด้านบน

กลุ่มคำตอบสำหรับ A และ C

- a) 1 b) 2 c) 3 d) 4

กลุ่มคำตอบสำหรับ B

- a) 1 และ 2 b) 3 และ 4 c) 5 และ 6
d) 7 และ 8 e) 9 และ 10 f) 11 และ 12

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ในคำอธิบายต่อไป

พิจารณาการดำเนินการทางตรรกศาสตร์ที่ใช้แปลงรหัสไบนารีให้เป็นรหัสเกรย์ ตารางที่ 2 เป็นตารางค่าความจริงสำหรับแปลงรหัสไบนารีเป็นรหัสเกรย์ มีอินพุตเป็นรหัสไบนารีขนาด 4 บิต (B1, B2 ,B3 และ B4) และเอาต์พุตเป็นรหัสเกรย์ขนาด 4 บิต (G1, G2, G3 และ G4)

ตารางที่ 2 ตารางค่าความจริงสำหรับแปลงรหัสไบนารีเป็นรหัสเกรย์

อินพุต (Inputs)				เอาต์พุต (Outputs)			
B1	B2	B3	B4	G1	G2	G3	G4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

การดำเนินการทางตรรกศาสตร์ที่ให้เอาต์พุต G1, G2, G3, และ G4 สามารถแสดงได้ดังนี้:

$$G1 = B1$$

$$G2 = B1 \oplus B2$$

$$G3 = B2 \oplus B3$$

$$G4 = \boxed{D}$$

กลุ่มคำตอบสำหรับ D

a) $B0 \oplus (B3 + B4)$

b) $B0 \oplus (B3 \oplus B4)$

c) $B3 + B4$

d) $B3 \oplus B4$

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ในคำอธิบายต่อไป

พิจารณาการดำเนินการทางตรรกศาสตร์ที่ใช้แปลงรหัสเกรย์ให้เป็นรหัสไบนารี ตารางที่ 3 เป็นตารางค่าความจริงสำหรับแปลงรหัสเกรย์เป็นรหัสไบนารี มีอินพุตเป็นรหัสเกรย์ขนาด 4 บิต (คือ G1, G2, G3 และ G4) และเอาต์พุตเป็นรหัสไบนารีขนาด 4 บิต (คือ B1, B2, B3 และ B4)

ตารางที่ 3 ตารางค่าความจริงสำหรับแปลงรหัสเกรย์เป็นรหัสไบนารี

อินพุต (Inputs)				เอาต์พุต (Outputs)			
G1	G2	G3	G4	B1	B2	B3	B4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

การดำเนินการทางตรรกศาสตร์ที่ให้เอาต์พุต B1, B2 และ B3 (ยกเว้น B4) สามารถแสดงได้ดังนี้:

$$B1 = G1$$

$$B2 = \overline{G1} \cdot G2 + G1 \cdot \overline{G2} = G1 \oplus G2$$

$$B3 = \overline{G1} \cdot \overline{G2} \cdot G3 + \overline{G1} \cdot G2 \cdot \overline{G3} + G1 \cdot \overline{G2} \cdot \overline{G3} + G1 \cdot G2 \cdot G3 = \text{E}$$

กลุ่มคำตอบสำหรับ E

- a) $(G1 + \overline{G2}) \cdot G3 + (\overline{G1} + G2) \cdot G3 + G1 \cdot G2 \cdot G3$
- b) $(G1 + \overline{G2}) \cdot G3 + (\overline{G1} + G2) \cdot \overline{G3} + G1 \cdot G2 \cdot G3$
- c) $(G1 \oplus G2) \cdot G3$
- d) $(G1 \oplus G2) + G3$
- e) $G1 \oplus G2 \oplus G3$

สำหรับข้อสอบข้อที่ Q2 ถึง Q5 ให้เลือกทำเพียงสองในสี่ข้อ

Q3. อ่านรายละเอียดของฐานข้อมูลสำหรับร้านเช่าดีวีดีต่อไปนี้ แล้วตอบคำถามย่อย 1 ถึง 3

ร้าน U Store เป็นร้านเช่าดีวีดีที่ใช้ฐานข้อมูลเชิงสัมพันธ์เพื่อจัดการการให้เช่าและการคืนดีวีดีของร้าน

ตารางดิสก์ (Disk table)

DiskID	FilmName	DirectorName	YearOfProduction	DateEntered
D0280	Alice in AI wonderland	A. I. Carroll	2019	2019-10-25
D0480	Super IT man	Clark Kent Jr.	2020	2021-02-26
D0860	Hello database world!	I. T. Pec	2021	2021-10-10

ตารางดิสก์จัดเก็บข้อมูลเกี่ยวกับดีวีดีแต่ละแผ่น ได้แก่ รหัสดิสก์ (DiskID) ชื่อเรื่อง (FilmName) ชื่อผู้กำกับ (DirectorName) ปีที่สร้าง (YearOfProduction) และวันที่เข้าร้าน (DateEntered)

ตารางลูกค้า (Customer table)

CustomerID	CustomerName	Address	Phone
C001	John	401 North street, East city	1234567
C003	Kathy	44 South street, East city	2345678
C005	Tom	109 Central street, West city	3456789

ตารางลูกค้าจัดเก็บข้อมูลลูกค้าแต่ละราย ได้แก่ รหัสลูกค้า (CustomerID) ชื่อลูกค้า (CustomerName) ที่อยู่ (Address) และเบอร์โทรศัพท์ (Phone)

ตารางใบเสร็จ (Bill table)

BillID	CustomerID	RentDateTime	RentalCharge	ReturnDateTime	Penalty
B1680	C005	2021-10-16 18:30	1.00	2021-10-18 17:50	NULL
B2120	C003	2021-10-21 12:10	2.00	NULL	NULL
B2180	C005	2021-10-21 18:40	1.00	NULL	NULL

ตารางใบเสร็จจัดเก็บข้อมูลใบเสร็จแต่ละใบ ได้แก่ เลขที่ใบเสร็จ (CustomerID) รหัสลูกค้า (CustomerID) วันเวลาเช่า (RentDateTime) ค่าเช่า (RentalCharge) วันเวลาคืน (ReturnDateTime) และค่าปรับ (Penalty)

ตารางรายละเอียดใบเสร็จ (BillDetail table)

BillDetailID	BillID	DiskID
B1680-1	B1680	D0280
B2120-1	B2120	D0860
B2120-2	B2120	D0280
B2180-1	B2180	D0480

ตารางรายละเอียดใบเสร็จจัดเก็บข้อมูลลำดับเลขที่ใบเสร็จ (BillDetailID) เลขที่ใบเสร็จ (BillID) และรหัสดิสก์ (DiskID)

เมื่อลูกค้าแต่ละรายเข้าแผนดีวีดี ข้อมูลใบเสร็จจะถูกสร้างขึ้น โดยมีค่าเข้าดีวีดีแผ่นละ 1 เหรียญ ในขณะนี้ ค่าว่าง (NULL) จะถูกกำหนดไว้ในคอลัมน์วันเวลาดิ้น (ReturndateTime) และค่าปรับ (Penalty)

เมื่อลูกค้านำแผ่นดีวีดีที่เข้าไปมาคืน ค่าของวันเวลาดิ้นจะถูกบันทึก รวมทั้งจะบันทึกค่าปรับหากลูกค้าส่งคืนเกินกำหนดและต้องจ่ายค่าปรับ ทั้งนี้ กำหนดส่งคืนคือ 72 ชั่วโมงนับจากวันที่เข้าแผ่นดีวีดี และค่าปรับอยู่ที่ 1 เหรียญต่อวัน (24 ชั่วโมง) ต่อแผ่น

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำสั่ง SQL1 ต่อไปนี้

คำสั่ง SQL1 ต่อไปนี้ จะแสดงรายการลูกค้าที่เกินกำหนดส่งคืนแผ่นดีวีดี (ณ เวลารันคำสั่ง SQL1) รวมถึงจำนวนของแผ่นดีวีดีที่เข้าไปด้วย

```
-- SQL1 --  
SELECT C.CustomerID, CustomerName, COUNT(*) AS DiskCount  
FROM Customer C, Bill B, BillDetail D  
WHERE C.CustomerID = B.CustomerID  
AND B.BillID = D.BillID  
AND  A  
AND  B  
GROUP BY C.CustomerID, CustomerName
```

โดยตัวอย่างข้อมูลที่แสดงในตารางด้านล่างนี้ เป็นผลลัพธ์การทำงานของคำสั่ง SQL1 ที่ถูกสั่งรัน ณ 13:30 น. ของวันที่ 2021-10-24

CustomerID	CustomerName	DiskCount
C003	Kathy	2

หมายเหตุ

- ฟังก์ชัน GETDATE() จะคืนค่าวันเวลาในรูปแบบ 'YYYY-MM-DD HH:MI:SS.sss'
- ฟังก์ชัน DATEDIFF(*part*, *start*, *end*) คืนค่าเป็นระยะเวลาที่แตกต่างกันของ 2 ช่วงเวลา คือวันเวลาเริ่มต้น *start* และวันเวลาสิ้นสุด *end* โดยมีตัวแปรหรือพารามิเตอร์ *part* ระบุว่าคืนค่าเป็น ปี เดือน วัน ชั่วโมง นาที วินาที (YY: year, MM: month, DD: day, HH: hour, MI: minute, SS: second) ตัวอย่างเช่น DATEDIFF(HH, '2021-10-23 23:00', '2021-10-24 01:20') จะคืนค่าเป็น 3 (ปัดเศษขึ้นให้เป็นตัวเลขจำนวนเต็ม)

กลุ่มคำตอบสำหรับ A

- a) Penalty IS NOT NULL
- b) Penalty IS NULL
- c) ReturnDateTime IS NOT NULL
- d) ReturnDateTime IS NULL

กลุ่มคำตอบสำหรับ B

- a) DATEDIFF(HH, RentDateTime, GETDATE()) < 72
- b) DATEDIFF(HH, RentDateTime, GETDATE()) > 72
- c) DATEDIFF(HH, ReturnDateTime, GETDATE()) < 72
- d) DATEDIFF(HH, ReturnDateTime, GETDATE()) > 72

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ในคำสั่ง SQL2 ต่อไปนี้

ลูกค้าที่เกินกำหนดส่งคืนแผ่นดีวีดี (ณ เวลารันคำสั่ง SQL1) รวมถึงจำนวนของแผ่นดีวีดีที่เข้าไปด้วย

คำสั่ง SQL2 ต่อไปนี้ จะแสดงรายการลูกค้าพร้อมข้อมูลใบเสร็จ แต่ถ้าลูกค้ารายใดไม่มีข้อมูลใบเสร็จที่เชื่อมโยงกันอยู่ เลขที่ใบเสร็จ (BillID) และวันเวลาเช่า (RentDateTime) จะเป็นค่าว่าง (NULL)

```
-- SQL2 --  
SELECT C.CustomerID, CustomerName, BillID, RentDateTime  
FROM CUSTOMER C  
     C Bill B  
ON C.CustomerID = B.CustomerID
```

โดยตัวอย่างข้อมูลที่แสดงในตารางด้านล่างนี้ เป็นผลลัพธ์การทำงานของคำสั่ง SQL2

CustomerID	CustomerName	BillID	RentDateTime
C001	John	NULL	NULL
C003	Kathy	B2120	2021-10-21 12:10
C005	Tom	B1680	2021-10-16 18:30
C005	Tom	B2180	2021-10-21 18:40

หมายเหตุ:

- INNER JOIN: เป็นการเลือกรายการที่มีค่าตรงกันทั้งสองตาราง
- FULL OUTER JOIN: เป็นการแสดงทุกรายการเมื่อมีค่าตรงกันอยู่ในตารางด้านซ้ายหรือในตารางด้านขวา
- LEFT OUTER JOIN: เป็นการแสดงทุกรายการของตารางด้านซ้าย และรายการที่มีค่าตรงกันจากตารางด้านขวา ซึ่งผลลัพธ์จากตารางด้านขวาจะเป็นค่าว่าง (NULL) หากไม่มีค่าที่ตรงกัน

- RIGHT OUTER JOIN: เป็นการแสดงทุกรายการของตารางด้านขวา และรายการที่มีค่าตรงกันจากตารางด้านซ้าย ซึ่งผลลัพธ์จากตารางด้านซ้ายจะเป็นค่าว่าง (NULL) หากไม่มีค่าที่ตรงกัน

กลุ่มคำตอบสำหรับ C

- a) FULL OUTER JOIN
- b) INNER JOIN
- c) LEFT OUTER JOIN
- d) RIGHT OUTER JOIN

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำสั่ง SQL3 ต่อไปนี้

ผู้จัดการร้านต้องการพิจารณาว่ามีดีวีดีเรื่องใดบ้างที่เข้าร้านมาตั้งแต่ช่วงเวลาหนึ่งแล้ว แต่ยังคงอยู่ในความสนใจของลูกค้าอยู่ คำสั่ง SQL3 ต่อไปนี้ แสดงรายการดีวีดีที่เข้าร้านมาก่อนวันที่ 2019-10-31 และถูกเช่าสองครั้งหรือมากกว่านั้นในช่วง 12 เดือนที่ผ่านมา ทั้งนี้สมมติว่า ตารางใบเสร็จ (Bill table) ได้เก็บข้อมูลใบเสร็จในช่วง 12 เดือนที่ผ่านมาไว้แล้ว

```
-- SQL3 --
SELECT D.DiskID, FilmName, COUNT(*) AS RentCount
FROM Disk D, BillDetail B
WHERE D.DiskID = B.DiskID
AND DateEntered <= '2019-10-31'
```

D
E

โดยตัวอย่างข้อมูลที่แสดงในตารางด้านล่างนี้ เป็นผลลัพธ์การทำงานของคำสั่ง SQL3

DiskID	FilmName	RentCount
D0280	Alice in AI wonderland	2

กลุ่มคำตอบสำหรับ D และ E

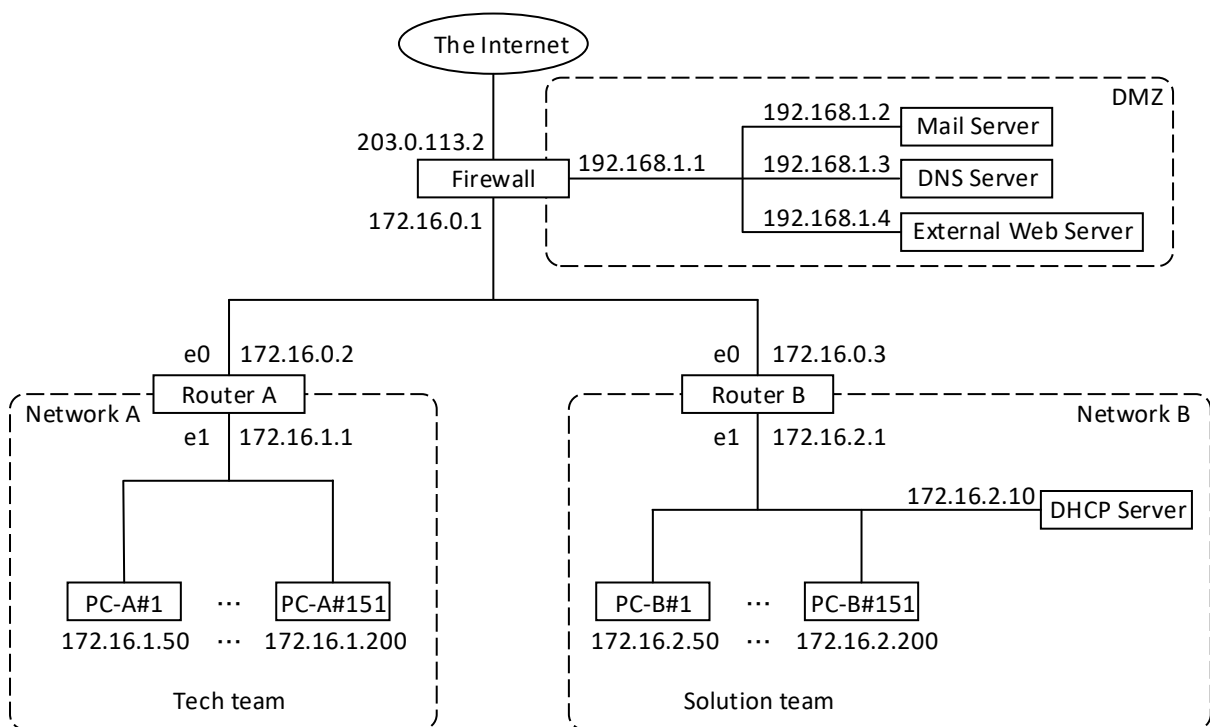
- a) AND COUNT(*) >= 2
- b) HAVING COUNT(*) >= 2
- c) GROUP BY D.DiskID
- d) GROUP BY D.DiskID, FilmName
- e) ORDER BY D.DiskID
- f) ORDER BY D.DiskID, FilmName

สำหรับข้อสอบข้อที่ Q2 ถึง Q5 ให้เลือกทำเพียงสองในสี่ข้อ

Q4. อ่านคำอธิบายเกี่ยวกับเครือข่ายภายในของบริษัทแห่งหนึ่ง แล้วตอบคำถามย่อย 1 ถึง 3

บริษัท V ได้ออกแบบโครงสร้างพื้นฐานเครือข่ายภายในโดยใช้ที่อยู่เครือข่าย 172.16.0.0/24, 172.16.1.0/24 และ 172.16.2.0/24 เครือข่ายเหล่านี้เชื่อมต่อระหว่างกันด้วยไฟร์วอลล์ (Firewall), เราเตอร์ A (Router A) และเราเตอร์ B (Router B) บริษัท V ใช้ฟังก์ชัน NAT (Network Address Translation) ของไฟร์วอลล์เพื่อให้เครือข่ายภายในสามารถเข้าใช้อินเทอร์เน็ตได้ และมีเมลเซิร์ฟเวอร์ (Mail Server) เซิร์ฟเวอร์ DNS (DNS Server) และเว็บเซิร์ฟเวอร์สำหรับภายนอก (External Web Server) ติดตั้งไว้ใน DMZ เครื่องพีซี (PC) แต่ละเครื่องที่ถูกใช้โดยทีมเทคนิค (Tech team) ถูกเชื่อมต่ออยู่กับเครือข่าย A (Network A) และเครื่องพีซีต่าง ๆ ที่ถูกใช้โดยทีมโซลูชัน (Solution team) ถูกเชื่อมต่ออยู่กับเครือข่าย B (Network B) โดยไฟร์วอลล์จะบล็อกการสื่อสารขาเข้าทั้งหมดจากอินเทอร์เน็ตที่เข้ามายังเครือข่ายภายในรวมทั้ง DMZ เว้นเฉพาะการสื่อสารที่ได้รับอนุญาตเท่านั้น ในที่นี้ DMZ ใช้ซับเน็ต 192.168.1.0/27

รูปที่ 1 แสดงเครือข่ายภายในของบริษัท V



รูปที่ 1 เครือข่ายภายในของบริษัท V

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้

ในเครือข่าย B มีเซิร์ฟเวอร์ DHCP (DHCP Server) ช่วยจัดสรรที่อยู่ไอพีให้กับเครื่องพีซีทั้งหมดได้อย่างมีประสิทธิภาพ โดยพีซีทั้งหมดจะได้รับข้อมูลที่อยู่ไอพี (IP address) มาจากเซิร์ฟเวอร์ DHCP เครื่องนี้ แต่อุปกรณ์อื่น ๆ ด้านการบริหารจัดการเช่นเราเตอร์และเครื่องเซิร์ฟเวอร์จะใช้ที่อยู่ไอพีที่กำหนดไว้ตายตัว เมื่อเครื่องพีซีต่าง ๆ ในเครือข่าย A ต้องการร้องขอข้อมูล DHCP ก็ส่งคำขอ DHCP ไปยังเซิร์ฟเวอร์ DHCP ในเครือข่าย B โดยเซิร์ฟเวอร์ DHCP จะแจกจ่ายที่อยู่ไอพีได้ก็ต่อเมื่อได้รับคำขอ DHCP ที่ถูกกระจายมาเท่านั้น ซึ่งโดยปกติแล้ว กรณีเช่นนี้จะถูกจำกัดไว้ในขอบเขตทางกายภาพของเครือข่ายเดียวกัน คือ A เท่านั้น เพื่อให้สามารถรับข้อมูล DHCP สำหรับเครือข่าย A ได้นั้น จำเป็นต้องใช้คุณสมบัติของตัวส่งผ่าน DHCP (DHCP relay agent) บนเราเตอร์ A โดยตัวส่งผ่าน DHCP นี้คือบริการที่ทำหน้าที่ส่งต่อกลุ่มข้อมูล DHCP (DHCP packet) ระหว่างไคลเอนต์กับเซิร์ฟเวอร์ โดยจะส่งต่อคำขอจากไคลเอนต์ไปยังเซิร์ฟเวอร์ DHCP ในแบบยูนิแคสต์ (unicast) ซึ่งในโครงสร้างเครือข่ายนี้ ที่อยู่ปลายทางของคำขอที่ถูกส่งต่อมาก็คือ B และตัวส่งผ่าน DHCP นี้ ยังทำหน้าที่ส่งต่อการตอบกลับจากเซิร์ฟเวอร์ DHCP ไปยังไคลเอนต์ด้วย

ในการติดตั้งเครือข่ายนี้ ผู้ดูแลเครือข่ายเปิดใช้ตัวส่งผ่าน DHCP บนอินเทอร์เฟซ e1 ของเราเตอร์ A ซึ่งเชื่อมต่ออยู่กับไคลเอนต์ DHCP หลังจากตัวส่งผ่าน DHCP ถูกกำหนดค่าเรียบร้อยแล้ว คำขอ DHCP ที่ถูกกระจายสัญญาณ (broadcast) มาจะถูกส่งต่อไปยังเซิร์ฟเวอร์ DHCP เพื่อให้สามารถตอบกลับด้วยที่อยู่ไอพีที่ต้องไปยังไคลเอนต์ที่อยู่ต่างเครือข่ายได้

กลุ่มคำตอบสำหรับ A

- | | | |
|------------------|------------------|------------------|
| a) 172.16.0.0/16 | b) 172.16.0.0/24 | c) 172.16.1.0/24 |
| d) 172.16.2.0/24 | e) 172.16.3.0/24 | |

กลุ่มคำตอบสำหรับ B

- | | | | |
|-----------------|---------------|-----------------|-----------------|
| a) 172.16.0.2 | b) 172.16.0.3 | c) 172.16.0.255 | d) 172.16.1.1 |
| e) 172.16.1.255 | f) 172.16.2.1 | g) 172.16.2.10 | h) 172.16.2.255 |

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้

ตารางเส้นทาง (routing table) ของเครือข่ายภายในของบริษัท V ได้ถูกกำหนดไว้อย่างถูกต้อง โดยผู้ดูแลเครือข่ายได้ตรวจสอบและยืนยันว่าพีซีแต่ละเครื่องในเครือข่าย A สามารถสื่อสารระหว่างกัน รวมทั้งสามารถสื่อสารกับเซิร์ฟเวอร์ต่าง ๆ ใน DMZ ได้ อย่างไรก็ตาม พบว่าเครื่องพีซีที่อยู่ในเครือข่าย B ไม่สามารถเชื่อมต่อไปยังเครือข่ายอื่น ๆ

ผู้ดูแลระบบได้ตรวจสอบและยืนยันว่าการสื่อสารระหว่างพีซีแต่ละเครื่องในเครือข่าย B สามารถทำได้ตามปกติ อย่างไรก็ตาม เขาพบว่าการส่งข้อความ ping ไปยังไฟร์วอลล์และเราเตอร์ A นั้นไม่สามารถทำได้ เขาจึงตรวจสอบข้อมูลที่อยู่ที่ไอพีของพีซีหลายเครื่องในเครือข่าย B โดยตารางที่ 1 แสดงข้อมูลที่อยู่ที่ไอพีของโฮสต์ PC-B#1

ตารางที่ 1 ข้อมูลที่อยู่ที่ไอพีของโฮสต์ PC-B#1

ที่อยู่ที่ไอพี	ซับเน็ตมาสก์	เกตเวย์เริ่มต้น
172.16.2.50	255.255.255.0	172.16.1.1

จากข้อมูลในตารางที่ 1 เขาจึงสามารถระบุได้ว่า C และหลังจากที่ได้เปลี่ยนแปลงการกำหนดค่าของเซิร์ฟเวอร์ DHCP ให้เป็นดังที่แสดงในตารางที่ 2 แล้ว เครื่องพีซีต่าง ๆ ในเครือข่าย B ก็สามารถเข้าถึงเซิร์ฟเวอร์ใน DMZ และเครือข่าย B ได้

ตารางที่ 2 การกำหนดค่าที่ได้รับการแก้ไขบนเซิร์ฟเวอร์ DHCP

เครือข่าย	ซับเน็ตมาสก์	ช่วง	เกตเวย์เริ่มต้น
172.16.2.0	255.255.255.0	172.16.2.50 to 172.16.2.200	<input type="text"/> D

กลุ่มคำตอบสำหรับ C

- a) โฮสต์ไม่สามารถติดต่อกับเซิร์ฟเวอร์ DHCP ได้ จึงกำหนดที่อยู่ที่ไอพีขึ้นมาด้วยตนเอง
- b) โฮสต์ได้รับข้อมูลเกตเวย์เริ่มต้นที่ไม่ถูกต้องมาจากเซิร์ฟเวอร์ DHCP
- c) โฮสต์ใช้ที่อยู่ส่วนตัว (private address) ที่น่าจะได้รับการจัดสรรมาจากการใช้ DHCP
- d) โฮสต์ใช้ที่อยู่ที่ไอพีของเซิร์ฟเวอร์ DHCP เป็นเกตเวย์เริ่มต้นของตน

กลุ่มคำตอบสำหรับ D

- a) 172.16.0.2
- b) 172.16.1.1
- c) 172.16.2.1
- d) 172.16.2.10
- e) 192.168.1.1

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในตารางที่ 3

เพื่อป้องกันเซิร์ฟเวอร์ที่อยู่ใน DMZ ให้ปลอดภัย ผู้ดูแลระบบได้กำหนดกฎการสื่อสาร (traffic rule) ดังแสดงในตารางที่ 3 ไว้บนไฟร์วอลล์ตามกฎเกณฑ์ต่อไปนี้:

- (1) บริการส่งเมล (SMTP) บนที่อยู่ไอพีภายนอก (external IP address) ของไฟร์วอลล์จะถูกจับคู่เข้ากับเมลเซิร์ฟเวอร์ใน DMZ
- (2) บริการรับเมล (IMAP) บนที่อยู่ไอพีภายนอกของไฟร์วอลล์จะถูกจับคู่เข้ากับเมลเซิร์ฟเวอร์ใน DMZ
- (3) อนุญาตการสื่อสารจาก DMZ ไปยังอินเทอร์เน็ตผ่าน NAT (การแปลงที่อยู่ IP) ซึ่งจำเป็นสำหรับบริการที่จับคู่ไว้ให้สามารถทำงานได้อย่างถูกต้อง
- (4) อนุญาตการสื่อสารจาก LAN ต่าง ๆ ไปยัง DMZ ซึ่งช่วยให้เซิร์ฟเวอร์เข้าถึงได้จากผู้ใช้ภายใน
- (5) ปิดกั้นการสื่อสารจาก DMZ ไปยัง LAN เพื่อป้องกันการบุกรุกเครือข่ายต่าง ๆ ที่มาจาก DMZ

โปรดทราบว่ากลุ่มข้อมูลที่เป็นการตอบกลับ (return traffic) จากการสื่อสารที่ได้รับอนุญาตตามกฎหมายที่ระบุไว้ข้างต้นจะได้รับอนุญาตให้ผ่านได้โดยปริยาย

ตารางที่ 3 กฎการสื่อสารบนไฟร์วอลล์ (ไม่แสดงการกำหนดค่าอื่น ๆ ที่ไม่เกี่ยวข้อง)

ต้นทาง	ปลายทาง	บริการ	เวอร์ชันไอพี	แอคชัน	การแปลงที่อยู่
Internet	Firewall	<input type="text" value="E"/>	any	Allow	MAP 192.168.1.2
Internet	Firewall	IMAP	any	Allow	MAP 192.168.1.2
DMZ	Internet	any	any	Allow	<input type="text" value="F"/>
LANs	DMZ	any	any	Allow	
DMZ	LANs	any	any	deny	
...

กลุ่มคำตอบสำหรับ E และ F

- | | | |
|--------------------|----------|--------|
| a) any | b) DNAT | c) DNS |
| d) HTTP | e) HTTPS | f) NAT |
| g) Port Forwarding | h) SMTP | i) udp |

Q5. อ่านคำอธิบายเกี่ยวกับแอปพลิเคชันสั่งอาหารต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

บริษัท W วางแผนที่จะให้บริการสั่งอาหารกลับบ้านแก่ลูกค้าด้วยการพัฒนาแอปพลิเคชันสั่งอาหาร เมื่อใช้แอปพลิเคชันนี้ ลูกค้าสามารถเลือกร้านอาหาร เลือกอาหาร และสั่งซื้ออาหารได้ ลูกค้าสามารถระบุเวลาที่ต้องการรับอาหารได้ เพื่อให้ร้านอาหารสามารถเตรียมอาหารได้ในเวลาที่ถูกต้อง

[ฟังก์ชันหลัก]

(1) การสมัครสมาชิก

- มีสมาชิกสองประเภท: สมาชิกที่เป็นลูกค้า (ต่อไปนี้จะเรียกว่า ลูกค้า) และสมาชิกที่เป็นร้านอาหาร (ต่อไปนี้จะเรียกว่า ร้านอาหาร)
- ในการเป็นสมาชิกของแอปพลิเคชัน ลูกค้าและร้านอาหารต้องสมัครสมาชิกของแอปพลิเคชัน
- ข้อมูลการสมัครประกอบด้วยประเภทสมาชิก ชื่อ และหมายเลขโทรศัพท์
- แอปพลิเคชันจะลงทะเบียนรหัสลูกค้า (customer ID) หรือ รหัสร้านอาหาร (restaurant ID) ด้วยรหัสผ่านสำหรับสมาชิกแต่ละราย

(2) ฟังก์ชันสำหรับร้านอาหาร

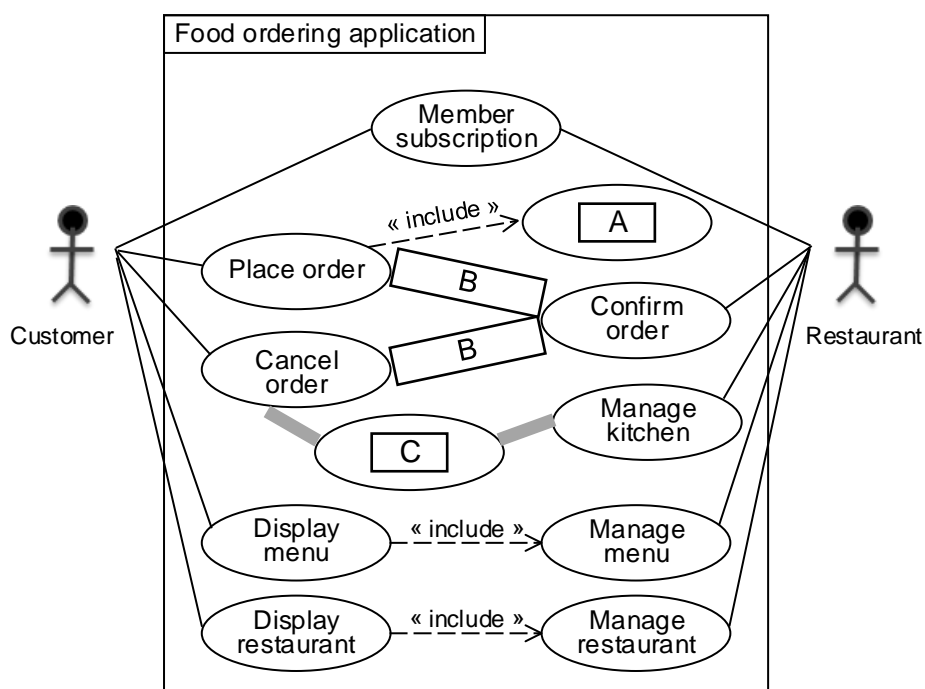
- การสร้างข้อมูลร้านอาหาร
 - ร้านอาหารลงทะเบียนข้อมูลร้านอาหารที่มีชื่อ ที่อยู่ และหมายเลขโทรศัพท์
 - หลังจากสร้างข้อมูลร้านอาหารแล้ว ร้านอาหารสามารถเพิ่มเมนูไปยังข้อมูลร้านอาหารได้ แต่ละเมนูจะถูกระบุด้วยรหัสเมนู (menu ID)
 - สำหรับแต่ละรายการอาหารในเมนูนั้น ร้านอาหารจะระบุชื่ออาหาร ราคา คำอธิบาย และรูปภาพของรายการอาหาร โดยแต่ละรายการจะถูกระบุด้วยรหัสอาหาร (food ID)
 - ข้อมูลร้านอาหารรวมถึงเมนูเหล่านี้สามารถแก้ไขหรือลบได้
- ยืนยันการสั่งซื้อของลูกค้า
 - หลังจากที่ลูกค้าได้ทำการสั่งซื้อ รายการสั่งซื้อจะปรากฏขึ้นบนหน้าจอของร้านอาหาร
 - ร้านอาหารตรวจสอบรายการสั่งซื้อ ยืนยันว่าอาหารแต่ละรายการนั้นพร้อมจำหน่าย และส่งการแจ้งเตือนไปยังลูกค้า
 - เริ่มทำอาหารเมื่อถึงเวลาที่กำหนด (deadline time) โดยร้านอาหารจะส่งข้อความ "ไม่สามารถยกเลิกคำสั่งซื้อ" ให้ลูกค้า ลูกค้าจะไม่สามารถยกเลิกคำสั่งซื้อได้หลังจากเวลานี้

(3) ฟังก์ชันสำหรับลูกค้า

- การสั่งซื้อ
 - ลูกค้าสามารถสั่งอาหารจากร้านอาหารได้ครั้งละหนึ่งร้าน
 - ต้องสั่งอาหารในวันเดียวกับวันที่รับอาหาร
 - ในการสั่งอาหาร ลูกค้าจะดูเมนู เลือกรายการอาหาร และกรอกจำนวนอาหารแต่ละรายการที่จะสั่ง

- หลังจากที่คุณยืนยันการสั่งซื้อ แอปพลิเคชันจะแสดงรายการสรุปคำสั่งซื้อบนหน้าจอของลูกค้า รายการสรุปคำสั่งซื้อประกอบด้วยชื่ออาหาร จำนวนอาหาร และราคาสำหรับอาหารแต่ละรายการ
- ลูกค้าส่งคำสั่งซื้อ (place order) ไปที่ร้านอาหารและแจ้งเวลาที่จะมารับให้ทางร้านอาหารทราบ
- ลูกค้าต้องรอรับการแจ้งเตือนจากทางร้าน
- การยกเลิกคำสั่งซื้อ
 - ลูกค้าสามารถยกเลิกคำสั่งซื้อได้ก่อนถึงเวลาที่กำหนดไว้ (deadline time)

รูปที่ 1 แสดงแผนภาพยูสเคส (use case diagram) ของแอปพลิเคชันสั่งอาหาร ลูกค้าต้องทำการสมัครสมาชิก (member subscription) หลังจากนั้นลูกค้าสามารถเข้าสู่ระบบของแอปพลิเคชันได้ โดยแอปพลิเคชันจะแสดงรายชื่อร้านอาหาร ลูกค้าสามารถเลือกร้านอาหาร แสดงรายการและเลือกอาหาร สั่งอาหาร (หรือยกเลิก) รายการอาหารบางรายการ ยืนยันการสั่งซื้อ และระบุเวลาที่ลูกค้าจะไปร้านเพื่อรับอาหาร



หมายเหตุ: ไม่แสดงส่วนที่แรเงา

รูปที่ 1 แผนภาพยูสเคส (use case diagram) ของแอปพลิเคชันสั่งอาหาร

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในรูปที่ 1

กลุ่มคำตอบสำหรับ A

- a) Check acceptability (ตรวจสอบการยอมรับ)
- b) Send notification (ส่งการแจ้งเตือน)
- c) Set pickup time (ระบุเวลารับอาหาร)
- d) Verify member subscription (ยืนยันการสมัครสมาชิก)

กลุ่มคำตอบสำหรับ B

- a) <-- « extend »
- b) -- « extend »-->
- c) <-- « include »
- d) -- « include »-->

กลุ่มคำตอบสำหรับ C

- a) Change order (เปลี่ยนแปลงการสั่งซื้อ)
- b) Change restaurant (เปลี่ยนแปลงร้านอาหาร)
- c) Confirm login information (ยืนยันข้อมูลการเข้าสู่ระบบ)
- d) No cancellation (ไม่สามารถยกเลิกได้)

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในตารางที่ 1

เพื่อให้เข้าใจความต้องการของแต่ละฟังก์ชัน ผู้วิเคราะห์ระบบได้อธิบายรายละเอียดด้วยคำอธิบายยูสเคส (use case description) ตารางที่ 1 แสดงคำอธิบายยูสเคสของ “การสั่งอาหาร (Place order)” ซึ่งหลังจากที่ลูกค้าเข้าสู่ระบบแอปพลิเคชันสำเร็จแล้วก็สามารถเริ่มสั่งอาหารได้

ตารางที่ 1 คำอธิบายยูสเคส (use case description) ของ “การสั่งอาหาร (Place order)”

ชื่อยูสเคส	การสั่งอาหาร (Place order)
ผู้กระทำหลัก (primary actor)	ลูกค้า
ผู้กระทำรอง (secondary actor)	-
ข้อมูลนำเข้า (input)	รหัสลูกค้า, รหัสร้านอาหาร, รหัสเมนู, <input type="text"/> D, จำนวนที่สั่ง, เวลารับอาหาร
ข้อมูลส่งออก (output)	สรุปคำสั่งซื้อเป็นรายการอาหาร
ทริกเกอร์ (Trigger)	ลูกค้าคลิกปุ่ม “Place order” บนหน้าจอของลูกค้า
ความพร้อมขาย (availability)	ตลอดเวลา
เงื่อนไขเบื้องต้น (pre-condition)	ลูกค้าเข้าสู่ระบบแอปพลิเคชันได้สำเร็จ
เงื่อนไขภายหลัง (post-condition)	รายการสั่งซื้อจะปรากฏขึ้นบนหน้าจอของร้านอาหาร

ตารางที่ 1 คำอธิบายยูสเคส (use case description) ของ "การสั่งอาหาร (Place order)" (ต่อ)

การไหล (Basic flow)	ผู้กระทำ (Actor)	แอปพลิเคชัน (Application)
	1. เลือกร้านอาหาร ลูกค้าเลือกร้านอาหารจากรายชื่อ ร้านอาหาร	
	2. เลือกเมนู ลูกค้าเลือกจากเมนูที่พร้อมขาย เช่น อาหารกลางวันและอาหารเย็น	
	3. เลือกอาหาร ลูกค้าเลือกอาหารจากรายการอาหาร ต่าง ๆ โดยในแต่ละรายการที่เลือก นั้น ลูกค้าจะกรอก <input type="text" value="E"/> และ คลิกปุ่ม "เพิ่มอาหาร" เพื่อเพิ่ม อาหารเข้าไปในรายการสั่งอาหาร	
		4. ตรวจสอบรายการอาหารที่เลือก แอปพลิเคชันตรวจสอบว่า <input type="text" value="E"/> สำหรับแต่ละรายการอาหารที่ถูกเลือก นั้นสามารถยอมรับได้หรือไม่ และคืน คำตอบกลับว่า "ยอมรับ (accepted)" หรือ "ไม่ยอมรับ (not accepted)" ตามความเหมาะสม
	5. ส่งคำสั่งซื้อ หลังจากลูกค้ายืนยันรายการอาหารที่ ต้องการสั่งซื้อทั้งหมดที่ถูกเพิ่มลงใน ตะกร้าแล้ว ลูกค้าจะกรอก <input type="text" value="F"/> และคลิกปุ่ม "ส่ง (submit)" เพื่อส่งคำสั่งซื้อ	
		6. ตรวจสอบ <input type="text" value="F"/> หากมีการขัดต่อกฎ ระบบจะคืนคำตอบกลับ มาว่า "ไม่ยอมรับ (not acceptable)" และแจ้งให้ลูกค้า กรอกข้อมูลใหม่อีกครั้ง หากไม่ขัดต่อกฎ แอปพลิเคชันจะส่ง คำขอยืนยันไปที่ร้านอาหารและรอการ ตอบกลับจากร้านอาหาร
		7. เสร็จสิ้นการสั่งซื้อ หลังจากที่ร้านอาหารได้ยืนยันคำสั่ง ซื้อแล้ว แอปพลิเคชันจะส่งการแจ้ง เตือนไปยังลูกค้า

กลุ่มคำตอบสำหรับ D

- | | |
|--|------------------------------------|
| a) Chef ID (รหัสพ่อครัว) | b) Deadline time (เวลาที่กำหนด) |
| c) Delivery address (ที่อยู่เพื่อจัดส่ง) | d) Food ID (รหัสอาหาร) |
| e) Service charge (ค่าบริการ) | f) Total amount (จำนวนเงินทั้งหมด) |

กลุ่มคำตอบสำหรับ E และ F

- | | |
|--|------------------------------------|
| a) amount of tip (จำนวนทิป) | b) deadline time (เวลาที่กำหนด) |
| c) delivery address (ที่อยู่เพื่อจัดส่ง) | d) number of dishes (จำนวนอาหาร) |
| e) pickup time (เวลารับอาหาร) | f) total amount (จำนวนเงินทั้งหมด) |

ข้อสอบข้อที่ Q6 เป็นคำถามบังคับ

Q6. อ่านรายละเอียดโปรแกรมสำหรับเข้ารหัสลับและถอดรหัสลับต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

[รหัสซีซาร์ (Caesar cipher)]

รหัสซีซาร์เป็นวิธีการเข้ารหัสลับวิธีหนึ่งที่ไม่ซับซ้อน โดยในขณะทำการเข้ารหัส (ส่วนที่อยู่ในวงเล็บ กล่าวถึงการถอดรหัส) ตัวอักษรแต่ละตัวในข้อความต้นฉบับ (ข้อความที่เข้ารหัส) ถูกแทนด้วยตัวอักษรที่ต่อจากตัวอักษรนั้นตามลำดับที่ตามมา (ลำดับก่อนหน้า) โดยนับจำนวนตัวอักษรเป็นลำดับเท่ากับตัวเลขที่กำหนด ในที่นี้ อักษรตัวพิมพ์ใหญ่ครอบคลุมล้อมรอบตั้งแต่ "Z" ถึง "A" หรืออีกนัยหนึ่งคือ ... X, Y, Z, A, B, ... และอักษรตัวพิมพ์เล็กครอบคลุมล้อมรอบตั้งแต่ "z" ถึง "a"

ตัวอย่างการเข้ารหัส: ถ้าตัวเลขที่กำหนดคือ 2 ตัวอักษร "A" จะถูกแทนที่ด้วยตัวอักษร "C" ตัวอักษร "p" จะถูกแทนที่ด้วยตัวอักษร "r" และตัวอักษร "Z" จะถูกแทนที่ด้วยตัวอักษร "B"

ตัวอย่างการถอดรหัส: ถ้าตัวเลขที่กำหนดคือ 2 ตัวอักษร "c" จะถูกแทนที่ด้วยตัวอักษร "a" ตัวอักษร "R" จะถูกแทนที่ด้วยตัวอักษร "P" และตัวอักษร "b" จะถูกแทนที่ด้วยตัวอักษร "z"

[รายละเอียดโปรแกรม]

โปรแกรมประกอบด้วย 2 ฟังก์ชันคือ Encrypt และ Decrypt

```
FUNCTION: Encrypt(CHAR: input[], CHAR: output[],  
                  INT: textLen, INT: keyEven, INT: keyOdd)  
FUNCTION: Decrypt(CHAR: input[], CHAR: output[],  
                  INT: textLen, INT: keyEven, INT: keyOdd)
```

- (1) ฟังก์ชัน Encrypt ใช้เข้ารหัสลับข้อความต้นฉบับ (plain text) ที่ถูกเก็บอยู่ในตัวแปร input[] โดยใช้คีย์ keyEven และ keyOdd โดยผลที่ได้เป็นข้อความเข้ารหัส (cipher text) ที่ถูกเก็บอยู่ในตัวแปร output[]
- ฟังก์ชัน Decrypt ใช้ถอดรหัสข้อความเข้ารหัสที่ถูกเก็บอยู่ในตัวแปร input[] โดยใช้คีย์ keyEven และ keyOdd โดยผลที่ได้เป็นข้อความต้นฉบับที่ถูกเก็บอยู่ในตัวแปร output[]

- (i) input[] และ output[] เป็นอาร์เรย์ของตัวอักษร ตำแหน่งของอินเด็กซ์ของอาร์เรย์เริ่มที่ 0
- (ii) textLen ระบุถึงจำนวนตัวอักษรที่ถูกเก็บอยู่ในตัวแปร input[] และ output[]
- (iii) keyEven และ keyOdd ถูกใช้เพื่อพิจารณาจำนวนตำแหน่งของตัวอักษรตามลำดับที่ตามมา (ตามลำดับก่อนหน้า) เมื่อทำการเข้ารหัสข้อความต้นฉบับ (ถอดรหัสข้อความ) โดย keyEven และ keyOdd มีค่าอยู่ระหว่าง 0 ถึง 25

- (2) ขั้นตอนการเข้ารหัสลับข้อความต้นฉบับ (plain text) มีดังต่อไปนี้:
- พิจารณาแต่ละตัวอักษรใน $\text{input}[i]$ ($i: 0, 1, \dots, \text{textLen} - 1$), กระทำซ้ำขั้นตอน (ii).
 - ถ้า $\text{input}[i]$ เป็นตัวอักษร (“A” – “Z” หรือ “a” – “z”) ให้นำตัวอักษรที่ต่อจากตัวอักษร $\text{input}[i]$ เป็นจำนวน keyEven หรือ keyOdd ตัว และบันทึกลงใน $\text{output}[i]$ โดยถ้าหากค่า i เป็นเลขคู่ (0, 2, 4, ...) จะใช้ค่าคีย์ที่อยู่ในตัวแปร keyEven ถ้าหากค่า i เป็นเลขคี่ (1, 3, 5, ...) จะใช้ค่าคีย์ที่อยู่ในตัวแปร keyOdd
 มิฉะนั้นถ้า $\text{input}[i]$ ไม่ใช่ตัวอักษร ทำการคัดลอกค่า $\text{input}[i]$ ไปใส่ใน $\text{output}[i]$
- (3) ขั้นตอนการถอดรหัสลับข้อความเข้ารหัส (cipher text) มีดังต่อไปนี้:
- พิจารณาแต่ละตัวอักษรใน $\text{input}[i]$ ($i: 0, 1, \dots, \text{textLen} - 1$), กระทำซ้ำขั้นตอน (ii).
 - ถ้า $\text{input}[i]$ เป็นตัวอักษร (“A” – “Z” หรือ “a” – “z”) ให้นำตัวอักษรที่มาก่อนตัวอักษร $\text{input}[i]$ เป็นจำนวน keyEven หรือ keyOdd ตัว และบันทึกลงใน $\text{output}[i]$ โดย ถ้าหากค่า i เป็นเลขคู่ (0, 2, 4, ...) จะใช้ค่าคีย์ที่อยู่ในตัวแปร keyEven ถ้าหากค่า i เป็นเลขคี่ (1, 3, 5, ...) จะใช้ค่าคีย์ที่อยู่ในตัวแปร keyOdd
 มิฉะนั้นถ้า $\text{input}[i]$ ไม่ใช่ตัวอักษร ทำการคัดลอกค่า $\text{input}[i]$ ไปใส่ใน $\text{output}[i]$
- (4) ตัวอักษรถูกแสดงด้วยรหัส ASCII; โดยตัวอักษร “A” – “Z” สามารถแทนค่าด้วยตัวเลข 65 – 90 ตามลำดับตัวอักษร และ “a” – “z” สามารถแทนค่าด้วยตัวเลข 97 – 122 ตามลำดับ
- (5) ตัวดำเนินการ “%” ใช้เพื่อหาค่าเศษจากการหาร (remainder) ในคำถามนี้กำหนดให้เครื่องหมายของเศษที่ไม่ใช่ศูนย์จะเหมือนกับเครื่องหมายของตัวถูกดำเนินการทางซ้าย
 ตัวอย่างเช่น $2 \% 26 = 2$, $30 \% 26 = 4$, $(-2) \% 26 = -2$, and $(-30) \% 26 = -4$
- (6) ตัวอย่างข้อมูลนำเข้าและผลการทำงานของฟังก์ชัน

ฟังก์ชันที่ถูกเรียก	ข้อมูลนำเข้า (Input)			ผลการทำงาน (Output)
	$\text{input}[]$	keyEven	keyOdd	$\text{output}[]$
Encrypt	“ABCxyz”	2	2	“CDEzab”
Encrypt	“ABCxyz”	2	3	“CEEaac”
Encrypt	“non=no(on)”	3	2	“ A ”
Decrypt	“CDEzab”	2	2	“ABCxyz”
Decrypt	“CEEaac”	2	3	“ABCxyz”
Decrypt	“ B ”	3	2	“f(x)=x+1”

[โปรแกรม]

```
FUNCTION: Encrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd) {
    INT: i, key

    FOR (i ← 0; i < textLen; i ← i + 1) {
        IF (  )
            key ← keyEven;
        ELSE
            key ← keyOdd;
        IF (input[i] ≥ 'A' and input[i] ≤ 'Z')
            output[i] ← ;
        ELSE {
            IF (input[i] ≥ 'a' and input[i] ≤ 'z')
                output[i] ← ;
            ELSE
                output[i] ← input[i];
        }
    }
}
```

```
FUNCTION: Decrypt(CHAR: input[], CHAR: output[],
                  INT: textLen, INT: keyEven, INT: keyOdd) {
    INT: i, key

    FOR (i ← 0; i < textLen; i ← i + 1) {
        IF (  )
            key ← keyEven;
        ELSE
            key ← keyOdd;
        IF (input[i] ≥ 'A' and input[i] ≤ 'Z')
            output[i] ← ;
        ELSE {
            IF (input[i] ≥ 'a' and input[i] ≤ 'z')
                output[i] ← ;
            ELSE
                output[i] ← input[i];
        }
    }
}
```

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในตารางที่คำอธิบายและตัวโปรแกรมข้างต้น

ในที่นี้ คำตอบที่จะเติมลงในช่องว่าง D1 และ D2 เป็นคำตอบที่จับคู่กันได้อย่างถูกต้องจากกลุ่มคำตอบสำหรับ D และ คำตอบที่จะแทนที่ในช่องว่าง E1 และ E2 เป็นคำตอบที่จับคู่กันได้อย่างถูกต้องจากกลุ่มคำตอบสำหรับ E

กลุ่มคำตอบสำหรับ A

- a) $pqp=pq(qp)$
- b) $prp=pr(rp)$
- c) $prp=rp(pr)$
- d) $qqq=qq(qq)$

กลุ่มคำตอบสำหรับ B

- a) $h(z)=a+1$
- b) $h(z)=z+3$
- c) $i(a)=a+4$
- d) $i(a)=z+1$

กลุ่มคำตอบสำหรับ C

- a) $i \% 2 = 0$
- b) $i \% 2 = 1$
- c) $input[i] = 0$
- d) $input[i] = 1$

กลุ่มคำตอบสำหรับ D

	D1	D2
a)	$'A' + (input[i] - 'A' + key - 26) \% 26$	$'a' + (input[i] - 'a' + key - 26) \% 26$
b)	$'A' + (input[i] - 'A' + key) \% 26$	$'a' + (input[i] - 'a' + key) \% 26$
c)	$'A' + (input[i] - 'A' - key + 26) \% 26$	$'a' + (input[i] - 'a' - key + 26) \% 26$
d)	$'A' + (input[i] - 'A' - key) \% 26$	$'a' + (input[i] - 'a' - key) \% 26$
e)	$(input[i] + key) \% 26$	$(input[i] + key) \% 26$
f)	$(input[i] - key) \% 26$	$(input[i] - key) \% 26$

กลุ่มคำตอบสำหรับ E

	E1	E2
a)	$'A' + (input[i] - 'A' + key - 26) \% 26$	$'a' + (input[i] - 'a' + key - 26) \% 26$
b)	$'A' + (input[i] - 'A' + key) \% 26$	$'a' + (input[i] - 'a' + key) \% 26$
c)	$'A' + (input[i] - 'A' - key + 26) \% 26$	$'a' + (input[i] - 'a' - key + 26) \% 26$
d)	$'A' + (input[i] - 'A' - key) \% 26$	$'a' + (input[i] - 'a' - key) \% 26$
e)	$(input[i] + key) \% 26$	$(input[i] + key) \% 26$
f)	$(input[i] - key) \% 26$	$(input[i] - key) \% 26$

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้

อัลกอริทึมและโค้ดที่ใช้ในฟังก์ชัน Encrypt และ Decrypt มีความคล้ายคลึงกัน ดังนั้นฟังก์ชัน Encrypt สามารถกระทำการถอดรหัสลับข้อความได้ ถ้าได้รับอาร์กิวเมนต์ (arguments) นำเข้าที่เหมาะสม

พิจารณากรณีที่โปรแกรม main เรียกฟังก์ชัน Decrypt โดยใช้คำสั่งที่ให้มาด้านล่าง โดยค่าของ keyEven และ keyOdd เป็นไปตามเงื่อนไขที่ระบุไว้ในรายละเอียดโปรแกรมข้อ (1) (iii)

```
Decrypt(input, output, textLen, keyEven, keyOdd);
```

ถ้าคำสั่งนี้ถูกแทนที่ด้วยคำสั่งด้านล่าง:

```
Encrypt(input, output, textLen, 26 - keyEven, 26 - keyOdd);
```

ดังนั้น โดยส่วนใหญ่แล้ว การถอดรหัสลับจึงสามารถทำได้โดยใช้ฟังก์ชัน Encrypt อย่างไรก็ตาม ปัญหาของการใช้คำสั่งเช่นนี้คือค่าของอาร์กิวเมนต์ลำดับที่ 4 และ/หรือ 5 ไม่เป็นไปตามเงื่อนไขที่ระบุไว้ในรายละเอียดโปรแกรมข้อ(1) (iii) เมื่อ keyEven และ/หรือ keyOdd มีค่าเป็น F

นอกจากนี้ เพื่ออำนวยความสะดวกให้ผู้ใช้ ฟังก์ชัน Encrypt จึงถูกแก้ไขเพื่อให้สามารถรับค่าใดก็ได้สำหรับตัวแปร keyEven และ keyOdd

ลำดับคำสั่งต่อไปนี้ถูกเพิ่มเติมในฟังก์ชัน Encrypt ณ บรรทัดก่อนหน้าลูป FOR เพื่อแปลงค่าที่เก็บอยู่ในตัวแปร keyEven และ keyOdd ที่ได้รับให้เป็นค่าของ keyEven และ keyOdd ที่มีค่าระหว่าง 0 ถึง 25 ตามที่ได้กำหนดไว้ก่อนหน้านี้

```
/* แปลงค่าใด ๆ ในตัวแปร keyEven ให้อยู่ในช่วง 0 ถึง 25 ตามที่ได้กำหนดไว้ก่อนหน้านี้ */
```

```
 G
```

```
/* แปลงค่าใด ๆ ในตัวแปร keyOdd ให้อยู่ในช่วง 0 ถึง 25 ตามที่ได้กำหนดไว้ก่อนหน้านี้ */
```

```
 //หมายเหตุ: ไม่แสดงส่วนที่แรเงา
```

ในที่นี้ ลำดับคำสั่ง G สามารถถูกแทนที่ได้ด้วยคำสั่งสำหรับการกำหนดค่าดังนี้:

```
keyEven ←  H;
```

กลุ่มคำตอบสำหรับ F

- a) 0
- b) 0 or 25
- c) 25
- d) not 0

กลุ่มคำตอบสำหรับ G

- a) IF (keyEven < 0)
 keyEven \leftarrow - keyEven;
 keyEven \leftarrow keyEven % 26;
- b) IF (keyEven < 0)
 keyEven \leftarrow keyEven + 26;
 keyEven \leftarrow keyEven % 26;
- c) keyEven \leftarrow keyEven % 26;
 IF (keyEven < 0)
 keyEven \leftarrow - keyEven;
- d) keyEven \leftarrow keyEven % 26;
 IF (keyEven < 0)
 keyEven \leftarrow keyEven + 26;

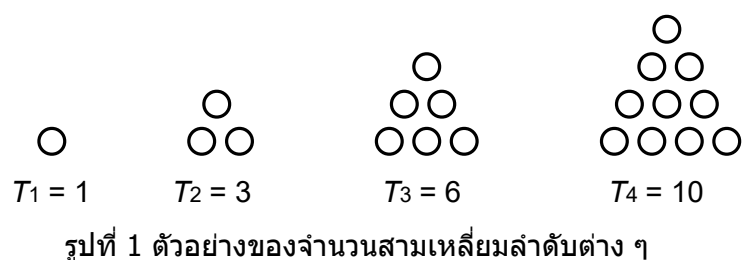
กลุ่มคำตอบสำหรับ H

- a) $((- \text{keyEven} \% 26) + 26) \% 26$
- b) $((\text{keyEven} \% 26) + 26) \% 26$
- c) $(- \text{keyEven} \% 26) + 26$
- d) $(26 - \text{keyEven}) \% 26$

สำหรับข้อสอบข้อที่ Q7 และ Q8 ให้เลือกทำหนึ่งในสองข้อ
 จากนั้น ให้ระบายทับในวงกลม (S) ในกระดาษคำตอบสำหรับข้อที่เลือกทำ
 หากเลือกทั้งสองข้อ จะตรวจให้คะแนนเฉพาะข้อแรกเท่านั้น

Q7. อ่านคำอธิบายของโปรแกรมและโค้ดโปรแกรมภาษา C จากนั้นให้ตอบคำถามย่อย 1 และ 2

จำนวนของจุดที่ประกอบกันขึ้นเป็นรูปสามเหลี่ยมด้านเท่า (equilateral) มีจำนวนในแต่ละด้านเท่ากัน คือ N จุด เรียกว่าจำนวนสามเหลี่ยมลำดับที่ N (T_N : N -th triangular number) ตัวอย่างเช่น 10 เป็นจำนวนสามเหลี่ยมลำดับที่ 4 เนื่องจากจุดจำนวน 10 จุด สามารถประกอบกันเป็นรูปสามเหลี่ยมด้านเท่าที่มีจำนวนจุด 4 จุด เท่ากันทุก ๆ ด้าน รูปที่ 1 เป็นการแสดงตัวอย่างของจำนวนสามเหลี่ยมลำดับต่าง ๆ



ทราบกันว่าผลรวมของจำนวนสามเหลี่ยมสองจำนวนที่อยู่ต่อเนื่องกัน เช่น $T_{N-1} + T_N$ จะมีค่าเท่ากับ N^2 ยกตัวอย่าง $T_3 + T_4 = 16$ ซึ่งมีค่าเท่ากับ 4^2

[คำอธิบายโปรแกรม]

กำหนดให้ช่วงของอินพุต N มีค่าระหว่าง $1 \leq N \leq 1000$, โปรแกรมจะวาดรูปสามเหลี่ยมโดยใช้อักขระโอ 'o' (แทนจุดในแต่ละด้าน) และอักขระช่องว่าง ' ' (อักขระเว้นวรรค) และหาค่าจำนวนสามเหลี่ยมด้วยการนับจำนวน 'o' ในรูป สำหรับสามเหลี่ยมลำดับที่ T_N ที่ $N \geq 2$ โปรแกรมจะวาดสามเหลี่ยมลำดับก่อนหน้าและหาค่าจำนวนสามเหลี่ยม รวมทั้งตรวจสอบผลรวมว่ามีค่าเท่ากับ N^2 หรือไม่ด้วย ตารางที่ 1 แสดงรูปสามเหลี่ยมที่ถูกวาดโดยโปรแกรมจำนวน 4 ลำดับแรก โดยใช้เครื่องหมายจุด (.) เป็นการแสดงแทนอักขระช่องว่าง (space)

ตารางที่ 1 ตัวอย่างของรูปสามเหลี่ยมที่ถูกวาดโดยโปรแกรม

N	T_N	Triangle drawn by the program
1	1	o.
2	3	.o. o.o.
3	6	..o. .o.o. o.o.o.
4	10	...o. ..o.o. .o.o.o. o.o.o.o.

โปรแกรมจะแสดงผลดังต่อไปนี้ เมื่อค่าที่อินพุตเป็น 1

```
Input value for generating a triangular number: 1
0
T(1) = 1
```

โปรแกรมจะแสดงผลดังต่อไปนี้ เมื่อค่าที่อินพุตเป็น 3

```
Input value for generating a triangular number: 3
0
0 0
0 0 0
T(3) = 6
```

```
The previous triangle
0
0 0
T(2) = 3
```

```
T(2) + T(3) = 3 ^ 2
Square property held
```

ตารางที่ 2 อธิบายถึงฟังก์ชันที่ถูกใช้ในโปรแกรม

ตารางที่ 2 ฟังก์ชันที่ถูกใช้ในโปรแกรม

ฟังก์ชัน	คำอธิบายฟังก์ชัน
int DrawTriangle(int n)	วาดรูปสามเหลี่ยมด้านเท่าที่มีจำนวนจุด n จุดในทุกด้าน และพิมพ์จำนวนสามเหลี่ยมลำดับที่ n (n-th triangular number)
void PrintSpace(int n)	พิมพ์อักขระช่องว่างจำนวน n ตัว

[โปรแกรม]

```
#include <stdio.h>

int DrawTriangle(int n);
void PrintSpace(int n); // α

int main() {
    int N, S, Ta, Tb;

    printf("Input value for generating a triangular number: ");
    scanf("%d", &N);
    Ta = DrawTriangle();
```

```

    if (N > 1) {
        printf("The previous triangle\n");
        Tb = DrawTriangle();
        S = N * N; // β
        if () {
            printf("T(%d) + T(%d) = %d ^ 2\n", N - 1, N, N); // γ
            printf("Square property held\n");
        } else {
            printf("Square property not held\n");
        }
        printf("\n");
    }
    return 0;
}

int DrawTriangle(int n) {
    int i, j, result = 0;

    for (i = 1; i <= n; i++) {
        PrintSpace();
        for (j = 1; ; j++) {
            printf("o ");
            result++;
        }
        printf("\n");
    }
    printf("T(%d) = %d\n\n", n, result);
    return result;
}

void PrintSpace(int n) {
    int i;

    for (i = 1; i <= n; i++) {
        printf(" ");
    }
}

```


คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในโปรแกรมด้านบน

กลุ่มคำตอบสำหรับ A และ B

- a) N b) $N + 1$ c) $N - 1$ d) T_a e) $T_a - 1$

กลุ่มคำตอบสำหรับ C

- a) $N > 2$ b) $T_a + T_b == S$
c) $T_a == S + T_b$ d) $T_b == S + T_a$

กลุ่มคำตอบสำหรับ D

- a) i b) $i - 1$ c) $n - 1$
d) $n - i$ e) $n - i + 1$

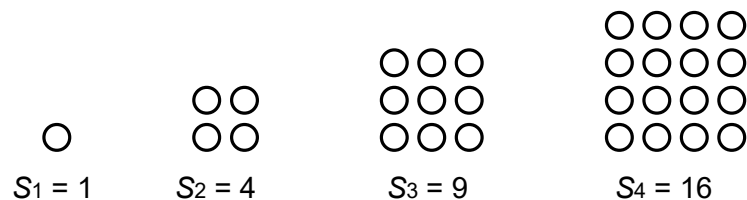
กลุ่มคำตอบสำหรับ E

- a) $j < i$ b) $j < i - 1$ c) $j <= i$ d) $j <= i + 1$

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในคำอธิบายต่อไปนี้ สมมติว่าคำตอบที่ถูกต้องได้ถูกเติมลงในช่องว่าง E ของคำถามย่อย 1 เรียบร้อยแล้ว

จำนวนของจุดที่ประกอบกันขึ้นเป็นรูปสี่เหลี่ยมจัตุรัส (Square) โดยมีจำนวนจุด N จุดเท่ากันในทุกด้าน เรียกว่าจำนวนจัตุรัสลำดับที่ N (S_N : N -th square number) รูปที่ 2 เป็นการแสดงตัวอย่างของจำนวนจัตุรัสลำดับต่าง ๆ



รูปที่ 2 ตัวอย่างของจำนวนจัตุรัสลำดับต่าง ๆ

จำนวนจัตุรัสนั้นมีคุณสมบัติที่สอดคล้องกับจำนวนสามเหลี่ยมสองจำนวนที่มีลำดับต่อเนื่องกัน ซึ่งสามารถแสดงเป็นสูตรได้ คือ $T_{N-1} + T_N = S_N$

มีฟังก์ชันใหม่หนึ่งฟังก์ชันที่ต้องถูกเพิ่มเข้าไปในโปรแกรมเพื่อให้สามารถวาดสี่เหลี่ยมจัตุรัสโดยใช้
อักขระโอ 'o', อักขระดอกจัน '*' และช่องว่าง ' ' (อักขระเว้นวรรค) และหาจำนวนจัตุรัสด้วยการนับ
จำนวนอักขระ 'o' และ '*' ในรูปสี่เหลี่ยมจัตุรัส

รูปสี่เหลี่ยมจัตุรัสที่แต่ละด้านมี N จุดนั้นประกอบขึ้นจากรูปสามเหลี่ยมจำนวนสองรูป โดยรูปแรกจะเป็น
รูปสามเหลี่ยมที่มีจุดจำนวน N จุดในแต่ละด้าน (วาดด้วย 'o') และรูปที่สองเป็นสามเหลี่ยมที่มีจุด
N - 1 จุดในแต่ละด้าน (วาดด้วย '*')

ตารางที่ 3 อธิบายถึงฟังก์ชันใหม่ที่ถูกเพิ่มเข้ามาในโปรแกรม

ตารางที่ 3 ฟังก์ชันใหม่ที่ถูกเพิ่มเข้ามาในโปรแกรม

ฟังก์ชัน	คำอธิบายฟังก์ชัน
int DrawSquare(int n)	วาดรูปสี่เหลี่ยมจัตุรัสที่มีจำนวนจุด n จุดในแต่ละด้าน และพิมพ์จำนวนจัตุรัสลำดับที่ n (n-th square number)

โปรแกรมจะมีดัดแปลงแก้ไขตามรายการดังต่อไปนี้:

- (1) แทนที่บรรทัดด้านล่างนี้ ด้วยบรรทัดที่มีเครื่องหมาย α

```
int DrawSquare(int n);
```
- (2) แทนที่สองบรรทัดด้านล่างนี้ ด้วย บรรทัดที่มีเครื่องหมาย β

```
printf("The square number\n");
S = DrawSquare(N);
```
- (3) แทนที่บรรทัดด้านล่างนี้ ด้วย บรรทัดที่มีเครื่องหมาย γ

```
printf("T(%d) + T(%d) = S(%d)\n", N - 1, N, N);
```

- (4) ให้แทรกฟังก์ชันใหม่ต่อไปนี้ ด้วยบรรทัดสุดท้ายของโปรแกรม

```
int DrawSquare(int n) {
    int i, j, result = 0;

    for (i = 1; i <= n; i++) {
        for (j = 1; E; j++) {
            printf("o ");
            result++;
        }
        for (j = F; j <= n; j++) {
            printf("* ");
            result++;
        }
        printf("\n");
    }
    printf("S(%d) = %d\n\n", n, result);
    return result;
}
```

ภายหลังมีการดัดแปลงแก้ไขโปรแกรมจะแสดงผลดังต่อไปนี้ เมื่ออินพุตมีค่าเป็น 3

Input value for generating a triangular number: 3

0
0 0
0 0 0
 $T(3) = 6$

The previous triangle

0
0 0
 $T(2) = 3$

The square number

G

$S(3) = 9$

H

Square property held

กลุ่มคำตอบสำหรับ F

- a) i b) $i + 1$ c) $i - 1$ d) $n - i$

กลุ่มคำตอบสำหรับ G

- a) $\begin{matrix} * * 0 \\ * 0 0 \\ 0 0 0 \end{matrix}$ b) $\begin{matrix} * 0 0 \\ * * 0 \\ * * * \end{matrix}$ c) $\begin{matrix} 0 * * \\ 0 0 * \\ 0 0 0 \end{matrix}$ d) $\begin{matrix} 0 0 * \\ 0 * * \\ * * * \end{matrix}$

กลุ่มคำตอบสำหรับ H

- a) $T(2) + T(3) = 3 \wedge 2$ b) $T(2) + T(3) = 9$
c) $T(2) + T(3) = S(3)$ d) $T(3) + T(2) = 3 \wedge 2$
e) $T(3) + T(2) = 9$ f) $T(3) + T(2) = S(3)$

สำหรับข้อสอบข้อที่ Q7 และ Q8 ให้เลือกทำเพียงหนึ่งในสองข้อ

Q8. อ่านคำอธิบายของโปรแกรมและโค้ดโปรแกรมภาษาจาวาต่อไปนี้ จากนั้นตอบคำถามย่อย 1 และ 2

[คำอธิบายโปรแกรม]

โปรแกรมนี้เป็นเครื่องคิดเลขที่สามารถทำการบวก ลบ คูณ และหาร ตัวเลขจำนวนเต็ม โดยเครื่องคิดเลขจะประกอบไปด้วยปุ่มตัวเลข (0-9) ปุ่มการคำนวณ ได้แก่ บวก ลบ คูณ และหาร ปุ่มเท่ากับและปุ่มล้างข้อมูล ทั้งนี้เมื่อมีการกดปุ่มดังใด ๆ โปรแกรมจะประมวลผลที่เกี่ยวข้องกับปุ่มนั้น ในที่นี้ ตัวเลขและค่าต่าง ๆ จะถูกแสดงผลบนหน้าจอด้วยประโยคคำสั่ง `System.out.println`

- (1) อินเทอร์เฟซ (Interface) `key` นิยามเมธอดที่จะถูกประมวลผลหากมีการกดปุ่มบนเครื่องคิดเลข เมธอด `operateOn` จะประมวลผลกระบวนการต่าง ๆ ที่เกี่ยวข้องกับปุ่มให้สอดคล้องกับการทำงานของอินสแตนซ์ของคลาส `java.util.Stack` (ต่อจากนี้จะเรียกว่า สแต็ก:stack) ที่ส่งเข้ามาเป็นพารามิเตอร์ ซึ่งคลาส `Stack` ยังสนับสนุนเมธอด `peek` ที่ส่งคืนออบเจกต์ที่ด้านบนสุดของสแต็กโดยไม่ลบออบเจกต์ออกจากสแต็ก เพิ่มขึ้นมาจากเมธอด `push` และเมธอด `pop` และมีเมธอด `clear` ที่จะดำเนินการลบออบเจกต์ทั้งหมดออกจากสแต็ก
- (2) อินิวเมอเรชัน (enumeration) `DigitKey` นิยามค่าคงที่ `DIGIT0` จนถึง `DIGIT9` เพื่อเป็นตัวแทนปุ่มตัวเลข 0 จนถึง 9
เมธอด `operationOn` จะประมวลผลปุ่ม (ตัวเลขฐานสิบ) ที่รับเป็นอินพุตเข้ามานี้ผ่านสแต็กที่เป็นพารามิเตอร์ โดยจะมีการเก็บค่า (Value) ไว้ที่ตำแหน่งบนสุด (Top) โดยมีค่าเริ่มต้นเป็น 0 และเมธอด `operationOn` จะมีการอัปเดตค่านี้โดยตลอด
- (3) อินิวเมอเรชัน `OperationKey` นิยามค่าคงที่ `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `EQUAL` และ `CLEAR` เพื่อเป็นตัวแทนปุ่มของการดำเนินการทางคณิตศาสตร์ ไว้ตามลำดับดังนี้คือการบวก การลบ การคูณ และการหาร ปุ่มเท่ากับ และปุ่มล้างข้อมูล
เมธอด `operationOn` ดำเนินการคำนวณการดำเนินการทางคณิตศาสตร์ดังกล่าวกับค่าที่ถูกเก็บไว้ในสแต็ก
ทั้งนี้ ค่าคงที่แต่ละตัว ยกเว้น `EQUAL` และ `CLEAR` นิยามการทำงานผ่านอินเทอร์เฟซ คือ `java.util.function.BinaryOperator<T>` และเมธอด `apply` ต้องการพารามิเตอร์ 2 ตัว และคืนค่าเป็นผลของการดำเนินการนั้น ๆ ทั้งนี้ ชนิดข้อมูลของพารามิเตอร์และการคืนค่าถูกกำหนดไว้ด้วยชนิด `T`
- (4) คลาส `Calculator` ถูกสร้างขึ้นเพื่อเป็นตัวแทนของเครื่องคิดเลข โดยมีฟิลด์ `stack` ซึ่งเป็นสแต็กที่แทนสถานะของค่าตัวเลขต่าง ๆ ภายในเครื่องคิดเลข และฟิลด์ `pendingKey` จะเก็บปุ่มการดำเนินการทางคณิตศาสตร์เอาไว้ชั่วคราวจนกว่าจะมีการป้อนตัวเลขที่จำเป็นต่อการดำเนินการทางคณิตศาสตร์ให้แล้วเสร็จ นอกจากนี้ ยังเก็บปุ่มเท่ากับเมื่อปุ่มเท่ากับถูกกดด้วยเช่นกัน

ตัวอย่างเช่น เมื่อค่าคงที่ของปุ่ม DIGIT2, ADD และ DIGIT4 ถูกนำมาประมวลผลตามลำดับแล้ว ตัวเลข 4 และ 2 จะถูกจัดเก็บไว้ในสแต็กตามลำดับ และค่าของ pendingKey จะเป็น ADD และสมมติว่าหลังจากนี้มีการประมวลผลปุ่ม EQUAL ก็จะมีการประมวลผลการดำเนินการทางคณิตศาสตร์ของปุ่ม ADD เกิดขึ้น ซึ่งค่า 6 จะถูกเก็บไว้ในสแต็กแทน และค่าของ pendingKey ก็จะกลายเป็น EQUAL ในที่นี้กำหนดให้ผลลัพธ์จากการดำเนินการทางคณิตศาสตร์ของจำนวนตัวเลขสองค่านั้น จะมีค่าเป็นไปตามประเภทข้อมูล int ของภาษา Java

เมธอด onKeyPressed ถูกเรียกเมื่อปุ่มเครื่องคิดเลขถูกกด ปุ่มกดจะถูกส่งผ่านเป็นพารามิเตอร์ กระบวนการที่จะถูกประมวลผลนั้นมีพื้นฐานจากปุ่มที่ถูกกด (พารามิเตอร์) และค่าสถานะต่าง ๆ ภายในของเครื่องคิดเลข

(5) คลาส CalculatorTest เป็นโปรแกรมเพื่อจะทดสอบคลาส Calculator

ในลำดับแรก เมธอด main เริ่มต้นสร้างการเชื่อมโยงระหว่างระหว่างแต่ละอักขระ (character) กับค่าคงที่ปุ่มที่มีความสัมพันธ์กัน และได้สร้างอินสแตนซ์ของคลาส ชื่อ Calculator ขึ้นมา ด้วย จากนั้น จะทำการแปลงอักขระแต่ละตัวในสตริงที่ถูกส่งเข้ามาเป็นพารามิเตอร์ให้เป็นค่าคงที่ของปุ่ม แล้วจึงเรียกใช้เมธอด onKeyPressed ของอินสแตนซ์ Calculator ด้วยค่าคงที่ของปุ่มที่ได้แปลงมาแล้วเหล่านั้น ตัวอย่างเช่น สมมติว่าสตริง "2*3=" ถูกส่งเข้ามาเป็นพารามิเตอร์ของเมธอด main แล้ว อักขระแต่ละตัวจะถูกแปลงเป็นค่าคงที่ของปุ่มที่สัมพันธ์กัน โดยจะมีถูกแปลงเป็น DIGIT2, MULTIPLY, DIGIT3 และ EQUAL และเมธอด onKeyPressed ก็จะถูกเรียกใช้งาน รูปที่ 1 แสดงผลลัพธ์เมื่อเมธอด main ถูกประมวลผล

DIGIT2
2
MULTIPLY
2
DIGIT3
3
EQUAL
6

รูปที่ 1 ผลลัพธ์เมื่อเมธอด main ถูกประมวลผล

[Program 1]

```
import java.util.Stack;

public interface Key {
    public void operateOn(Stack<Integer> stack);
}
```

[Program 2]

```
import java.util.Stack;

enum DigitKey  Key {
    DIGIT0, DIGIT1, DIGIT2, DIGIT3, DIGIT4,
    DIGIT5, DIGIT6, DIGIT7, DIGIT8, DIGIT9;

    public void operateOn(Stack<Integer> stack) {
        stack.push( * 10 + );
    }
}
```

[Program 3]

```
import java.util.Stack;
import java.util.function.BinaryOperator;

enum OperationKey  Key {
    ADD((val1, val2) -> val1 + val2),
    SUBTRACT((val1, val2) -> val1 - val2),
    MULTIPLY((val1, val2) -> val1 * val2),
    DIVIDE((val1, val2) -> val1 / val2),
    EQUAL(null),
    CLEAR(null);

    private final BinaryOperator<Integer> operator;

    OperationKey(BinaryOperator<Integer> operator) {
        this.operator = operator;
    }

    public void operateOn(Stack<Integer> stack) {
        if () {
            Integer val2 = stack.pop();
            Integer val1 = stack.pop();
            stack.push(operator.apply(val1, val2));
        }
    }
}
```

[Program 4]

```
import java.util.Stack;

public class Calculator {
    private final Stack<Integer> stack = new Stack<>();
    private Key pendingKey;

    public Calculator() {
        stack.push(0);
    }

    public void onKeyPressed(Key key) {
        System.out.println(key);
        if (key instanceof DigitKey) {
            if (pendingKey == OperationKey.EQUAL) {
                reset();
            }
            key.operateOn(stack);
            System.out.println(stack.peek());
        } else if (key == OperationKey.CLEAR) {
            reset();
            System.out.println(stack.peek());
        } else {
            try {
                if (pendingKey != null) {
                    pendingKey.operateOn(stack);
                }
                System.out.println(stack.peek());
                pendingKey = key;
                if (key != OperationKey.EQUAL) {
                    stack.push(0);
                }
            } catch (ArithmeticException e) {
                System.out.println("Error");
                reset();
            }
        }
    }

    private void reset() {
        stack.clear();
        stack.push(0);
        pendingKey = null;
    }
}
```

[Program 5]

```
import java.util.HashMap;
import java.util.Map;

public class CalculatorTest {
    public static void main(String[] args) {
        Map<Character, 
```

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ A

- | | | |
|-------------|---------------|------------|
| a) extends | b) implements | c) imports |
| d) inherits | e) requires | f) throws |

กลุ่มคำตอบสำหรับ B และ C

- | | |
|--------------------------|------------------|
| a) ordinal() | b) stack.peek() |
| c) stack.pop() | d) stack.push(0) |
| e) stack.push(ordinal()) | f) values() |

กลุ่มคำตอบสำหรับ D

- a) operator != null b) operator == null c) stack != null
d) stack == null e) this != operator f) this == operator

กลุ่มคำตอบสำหรับ E

- a) Calculator b) Character c) DigitKey
d) Integer e) Key f) OperationKey

คำถามย่อย 2

ตารางที่ 1 แสดงแถวสุดท้ายของเอาต์พุต (ถ้าในกรณีของรูปที่ 1 ก็คือ 6) เมื่อเมธอด main ถูกประมวลผลโดยมีสตริงเป็นพารามิเตอร์ จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง แต่ละช่องในตารางที่ 1 ในที่นี้ สมมติว่าคำตอบที่ถูกต้องทั้งหมดได้ถูกเติมลงในทุกช่องว่าง ในโปรแกรมเรียบร้อยแล้ว

ตารางที่ 1 สตริงอักขระ (พารามิเตอร์) และเอาต์พุต (แถวสุดท้าย)

สตริงอักขระ (พารามิเตอร์)	เอาต์พุต (แถวสุดท้าย)
2*6/3=	4
-2=	-2
2*4==	<input type="text" value="F"/>
2*4C2=	<input type="text" value="G"/>
8/2/=	<input type="text" value="H"/>

กลุ่มคำตอบสำหรับ F ถึง H

- a) 0 b) 2
c) 4 d) 8
e) 16 f) 32
g) 64 h) ArithmeticException
i) Error

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