



October 2018

## Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

### Instructions:

1. Use a pencil. If you need to change an answer, erase your previous answer completely and neatly. Wipe away any eraser debris.
2. Mark your examinee information and test answers in accordance with the instructions below. Your answer will not be graded if you do not mark properly. Do not mark or write on the answer sheet outside of the prescribed places.

(1) **Examinee Number**

Write your examinee number in the space provided, and mark the appropriate space below each digit.

(2) **Date of Birth**

Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.

(3) **Question Selection**

For **Q7** and **Q8**, mark the (S) of the question you select to answer in the “Selection Column” on your answer sheet.

(4) **Answers**

Mark your answers as shown in the following sample question.

[Sample Question]

In which month is the spring Fundamental IT Engineer Examination conducted?

Answer group

- a) September      b) October      c) November      d) December

Since the correct answer is “b) October”, mark your answer sheet as follows:

[Sample Answer]

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

**Do not open the exam booklet until instructed to do so.**

**Inquiries about the exam questions will not be answered.**

### Notations used for 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 priorities]

Type of operation	Unary	Arithmetic		Relational	Logical	
Operators	+, -, not	×, ÷, %	+, -	>, <, ≥, ≤, =, ≠	and	or
Precedence	<div style="display: flex; align-items: center; justify-content: space-between;"> <span>High</span> <span>→</span> <span>←</span> <span>Low</span> </div>					

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

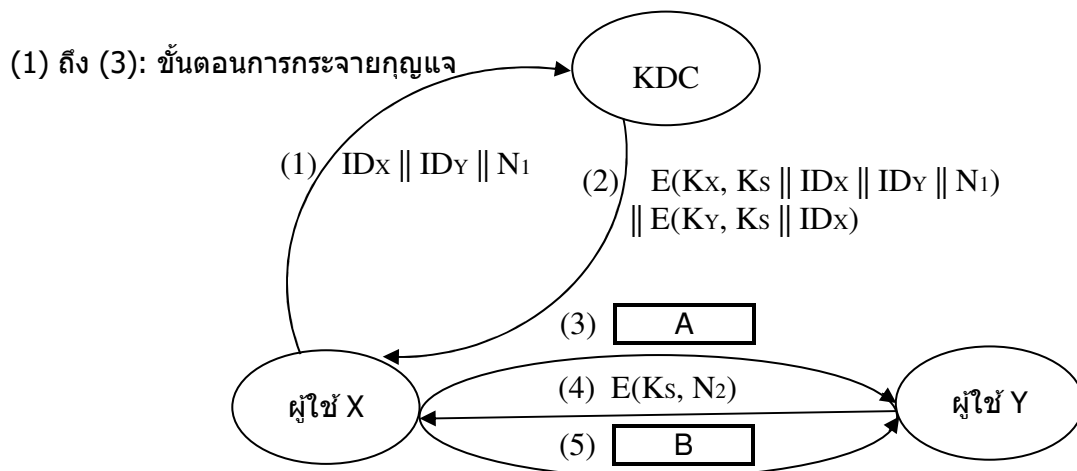
The “%” operator indicates a remainder operation.

**Q1.** อ่านคำอธิบายเกี่ยวกับการเข้ารหัสลับด้วยกุญแจแบบสมมาตร (symmetric key encryption) จากนั้นให้ตอบคำถามย่อย 1 และ 2

เมื่อผู้ใช้ต้องการส่งข้อความสื่อสารกับผู้ใช้รายอื่น อันดับแรกต้องสร้างเซสชัน (session) จากนั้นต้องใช้เทคนิคการเข้ารหัสลับเพื่อปกป้องข้อความจากผู้บุกรุก ในการเข้ารหัสลับด้วยกุญแจแบบสมมาตร ผู้ใช้ทั้งสองต้องแบ่งปันกุญแจเดียวกันเพื่อใช้ในการแลกเปลี่ยนสารสนเทศ

ศูนย์กระจายกุญแจ (key distribution center) ซึ่งจากนี้จะเรียกว่า KDC ส่งกุญแจหลัก (master key) ไปยังผู้ใช้ที่เกี่ยวข้องโดยใช้วิธีที่ไม่ใช้วิทยาการเข้ารหัสลับ เช่นการส่งมาทางพัสดุแบบกายภาพ ดังนั้น จึงมีเพียง KDC และผู้ใช้ที่เกี่ยวข้องเท่านั้นที่รู้กุญแจหลัก ในการสื่อสารแต่ละครั้งระหว่างผู้ใช้ ข้อความถูกเข้ารหัสด้วยกุญแจชั่วคราว (temporary key) ที่ถูกเรียกว่ากุญแจเซสชัน (session key) ที่ได้รับมาจาก KDC กุญแจเซสชันนี้ถูกส่งมาให้ผู้ใช้ผ่านการเข้ารหัสลับด้วยกุญแจหลัก (master key) ที่ KDC ส่งให้ไว้ก่อนหน้านี้

รูปที่ 1 แสดงขั้นตอนการกระจายกุญแจ (key distribution) และการพิสูจน์ตัวตน (authentication) เมื่อผู้ใช้ X ที่มีกุญแจหลัก  $K_x$  ต้องการสร้างการเชื่อมต่อทางตรรกะกับผู้ใช้ Y ที่มีกุญแจหลัก  $K_y$



(3) ถึง (5): ขั้นตอนการพิสูจน์ตัวตน

สัญลักษณ์:  $E(k, m)$ : ฟังก์ชันที่เข้ารหัสลับข้อความ  $m$  โดยใช้กุญแจ  $k$

$||$ : ตัวดำเนินการเชื่อมต่อกัน (Concatenation operator)

IDX: ตัวตน (ที่อยู่เครือข่าย) ของ X

IDY: ตัวตน (ที่อยู่เครือข่าย) ของ Y

KX: กุญแจหลักสำหรับ X

KY: กุญแจหลักสำหรับ Y

N1, N2: นอนซ์ (ตัวระบุรายการ)

KS: กุญแจเซสชันสำหรับเซสชัน S

รูปที่ 1 ขั้นตอนการกระจายกุญแจและการพิสูจน์ตัวตน

คำอธิบายต่อไปนี้อธิบายถึงขั้นตอนที่ (1) ถึง (5) ในรูปที่ 1 สังเกตว่าขั้นตอนที่ (1) ถึง (3) เป็นขั้นตอนกระจายกุญแจ (key distribution) และขั้นตอนที่ (3) ถึง (5) เป็นขั้นตอนพิสูจน์ตัวตน

- (1) ผู้ใช้ X ส่งคำขอกระจายกุญแจไปยัง KDC โดยข้อความที่ถูกส่งไปที่ KDC ประกอบด้วยตัวตน (ที่อยู่เครือข่าย) ของ X และ Y ซึ่งแทนด้วย IDx และ IDy และตัวเลขที่ใช้ระบุรายการ N1 สำหรับรายการ (transaction) นั้น ตัวเลขนี้รู้จักกันในชื่อนอนซ์ (nonce) ตัวเลขที่ถูกสุ่มขึ้นมา (random number) เป็นตัวเลือกที่ดีสำหรับนอนซ์ เนื่องจากยากต่อการคาดเดาโดยผู้ไม่หวังดี
- (2) KDC ส่งข้อความตอบกลับไปยังผู้ใช้ X โดยข้อความแบ่งออกเป็นสองส่วน: ส่วนแรกสำหรับผู้ใช้ X และส่วนที่สองสำหรับผู้ใช้ Y  
  
ส่วนแรกของข้อความถูกเข้ารหัสลับโดยใช้ Kx ส่งผลให้ผู้ใช้ X เป็นผู้เดียวที่อ่านข้อความได้ ข้อความนี้รวมกุญแจเซสชันสำหรับใช้ครั้งเดียว Ks ที่จะถูกใช้สำหรับเซสชันนั้น ๆ รวมถึงข้อความต้นฉบับที่มีนอนซ์เพื่อให้ผู้ใช้ X จับคู่การตอบกลับกับข้อความที่ส่งไปในขั้นตอนที่ (1) ได้  
  
ส่วนที่สองของข้อความถูกเข้ารหัสลับโดยใช้ Ky ส่งผลให้ผู้ใช้ Y เป็นเพียงผู้เดียวที่อ่านข้อความได้ ข้อความนี้มีกุญแจเซสชัน Ks และ IDx รวมอยู่ด้วย
- (3) ผู้ใช้ X เก็บกุญแจเซสชัน Ks สำหรับการใช้งานในเซสชันที่จะมาถึง และส่งต่อส่วนของข้อความที่ได้รับมาในขั้นที่ (2) ไปให้ผู้ใช้ Y ในตอนนี้ กุญแจเซสชันได้ถูกส่งไปให้กับผู้ใช้ X และ Y อย่างปลอดภัย และผู้ใช้ Y ได้ทราบว่าผู้ใช้ X ต้องการสื่อสารด้วยโดยใช้กุญแจเซสชัน Ks
- (4) ผู้ใช้ Y ส่งนอนซ์ N2 ไปให้ผู้ใช้ X โดยใช้กุญแจเซสชันที่ได้รับมาใหม่นี้สำหรับการเข้ารหัสลับ
- (5) ผู้ใช้ X ตอบกลับด้วยข้อความที่ถูกเข้ารหัสลับ ข้อความนี้ประกอบไปด้วยค่าของ N2 ที่ถูกเปลี่ยนแปลงไป ในที่นี้  $f(n)$  เป็นฟังก์ชันที่ทำการเปลี่ยนแปลงค่า  $n$  (ในที่นี้คือบวกด้วยหนึ่ง) เมื่อถึงขั้นตอนนี้ ผู้ใช้ Y ก็มั่นใจได้ว่าผู้ใช้ X มีกุญแจเซสชันสำหรับการสื่อสารในเซสชันนี้

### คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมในช่องว่าง  แต่ละช่องในรูปที่ 1

กลุ่มคำตอบ

- |                       |                       |
|-----------------------|-----------------------|
| a) $E(Ks, f(N2))$     | b) $E(Ks, N2)$        |
| c) $E(Kx, f(N2))$     | d) $E(Kx, Ks    IDx)$ |
| e) $E(Ky, Ks    IDx)$ | f) $E(Ky, Ks    IDy)$ |

## คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมในช่องว่าง  แต่ละช่องในคำอธิบายต่อไปนี้

KDC ส่ง  ให้ผู้ใช้ทั้งสองก่อนล่วงหน้า เมื่อผู้ใช้ X ต้องการสื่อสารด้วยข้อความกับผู้ใช้ Y KDC สร้าง  ก่อนการสื่อสารระหว่างผู้ใช้ทั้งสองจะเริ่มต้นขึ้น

เมื่อผู้ใช้ X ส่งคำขอกุญแจเซสชันไปให้ KDC แล้ว ทาง KDC จะส่งคืนข้อความเดิมไปพร้อมกับกุญแจเซสชันที่ผู้ใช้ร้องขอ สาเหตุหลักที่ข้อความดั้งเดิมที่มีnonceถูกส่งกลับไปให้ผู้ใช้ X คือ  และ

กลุ่มคำตอบสำหรับ C และ D

- |                                 |  |
|---------------------------------|--|
| a) กุญแจหลัก (master key(s))    | b) ที่อยู่เครือข่าย (network address(es))    |
| c) กุญแจเซสชัน (session key(s)) | d) ตัวระบุรายการ (transaction identifier(s)) |

กลุ่มคำตอบสำหรับ E และ F

- a) ผู้ใช้ X สามารถยืนยันได้ว่ากุญแจเซสชันยังคงปลอดภัย
- b) ผู้ใช้ X สามารถรู้ได้ว่าการตอบกลับเกิดจาก KDC จริง
- c) ผู้ใช้ X สามารถส่งข้อความดั้งเดิมที่ถูกเข้ารหัสแล้วไปให้ผู้ใช้ Y ได้
- d) ผู้ใช้ X สามารถใช้การตอบกลับนี้เป็นลายเซ็นดิจิทัลได้
- e) ผู้ใช้ X สามารถยืนยันได้ว่าคำขอไม่ได้ถูกแก้ไข
- f) ผู้ใช้ X สามารถยืนยันได้ว่าการตอบกลับนี้ไม่ใช่การโจมตีแบบทำซ้ำ (replay attack) ที่ใช้พื้นฐานข้อมูลจากเซสชันอื่น ๆ ก่อนหน้า

**Q2.** อ่านคำอธิบายที่เกี่ยวกับการแบ่งหน้า (paging) ต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

การแบ่งหน้า (paging) เป็นวิธีการจัดการหน่วยความจำที่จับคู่เลขที่อยู่ทางตรรกะ (logical address) เข้ากับที่อยู่ทางกายภาพ (physical address)

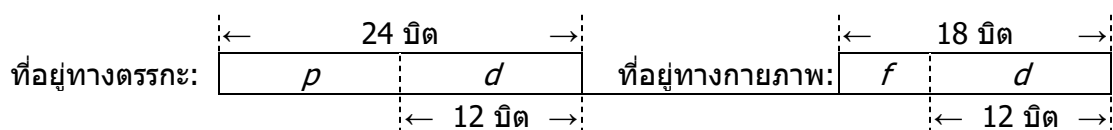
ในระบบที่จัดการหน่วยความจำด้วยวิธีการแบ่งหน้า พื้นที่ของเลขที่อยู่ทางตรรกะ (logical address space) ถูกแบ่งออกเป็นบล็อก ๆ เรียกว่าหน้า (page) และพื้นที่ของเลขที่อยู่ทางกายภาพ (physical address space) ก็ถูกแบ่งออกเป็นบล็อก ๆ เช่นกัน เรียกว่าเฟรม (frame) โดยทั่วไป หน้าจะถูกเก็บอยู่ในหน่วยความจำสำรองและจะถูกนำขึ้นไปใส่ในเฟรมที่ว่างเมื่อต้องการใช้งาน

เลขที่อยู่ตรรกะแบ่งออกเป็นสองส่วน: ส่วนของเลขหน้า  $p$  และการกระจัด (displacement)  $d$  และเช่นเดียวกันกับเลขที่อยู่กายภาพ ซึ่งแบ่งออกเป็นสองส่วนได้แก่ส่วนของเลขประจำเฟรม  $f$  และการกระจัด  $d$  โดยขนาดของ  $d$  คือขนาดของหน้า ซึ่งถูกกำหนดไว้ตายตัวและเป็นค่าเดียวกันทั้งสองที่เลขหน้าถูกใช้เป็นตัวชี้ (index) ในตารางหน้า (page table) โดยตารางหน้าจะรักษาความสอดคล้องกันระหว่างเลขหน้า (page number) และเลขประจำเฟรม (frame number) ด้วยการใช้ตารางหน้า ที่อยู่ทางตรรกะจะสามารถถูกแปลงไปเป็นที่อยู่ทางกายภาพได้โดยการแทนที่เลขหน้าด้วยเลขประจำเฟรมที่สอดคล้องกัน

## คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องใน  
คำอธิบายต่อไปนี้

รูปที่ 1 แสดงตัวอย่างรายละเอียดเลขที่อยู่ทางตรรกะและทางกายภาพ ที่อยู่ทางตรรกะมีความยาว 24 บิต ซึ่งครอบคลุมที่อยู่ทั้งสลิบ 16M ( $2^{24}$ ) ตำแหน่ง และที่อยู่ทางกายภาพมีความยาว 18 บิต ครอบคลุมที่อยู่ทั้งสลิบ 256k ( $2^{18}$ ) ตำแหน่ง ขนาดของหน้ามีความยาว 12 บิต ครอบคลุม 4,096 ( $=2^{12}$ ) ตำแหน่ง



รูปที่ 1 ตัวอย่างรายละเอียดของเลขที่อยู่ทางตรรกะและเลขที่อยู่ทางกายภาพ

ในรูปที่ 1 จำนวนของหน้าคือ 4,096 และจำนวนของเฟรมคือ 

A
---

โดยทั่วไป เมื่อเลขที่อยู่ทางตรรกะมีความยาว  $n$  บิต และขนาดของหน้ามีความยาว  $m$  บิต จำนวนของหน้าสามารถแทนได้ว่า 

B
---

กลุ่มคำตอบสำหรับ A

- a) 16                      b) 64                      c) 256                      d) 4,096

กลุ่มคำตอบสำหรับ B

- a)  $2^m$                       b)  $2^{n-m}$                       c)  $m^2$                       d)  $n - m$

## คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องในคำอธิบายต่อไปนี้

ในระบบหนึ่งที่ใช้กลไกการแบ่งหน้า รายละเอียดของเลขที่อยู่ทางตรรกะและทางกายภาพ เป็นไปตามที่แสดงในรูปที่ 1

รูปที่ 2 แสดงสถานการณ์ที่หน้าต่าง ๆ ในพื้นที่เลขที่อยู่ทางตรรกะถูกจับคู่เข้ากับเฟรมในพื้นที่เลขที่อยู่ทางกายภาพตามตารางหน้าในเวลานั้น ๆ ซึ่งขณะนั้น ตารางหน้าแสดงว่าข้อมูลที่อยู่ในพื้นที่หน่วยความจำทางตรรกะ 006C00 นั้น  C และข้อมูลที่อยู่ในพื้นที่หน่วยความจำทางตรรกะ 007E00 นั้น  D ในระบบจัดเก็บข้อมูลเสมือนที่ใช้ตารางหน้าที่แสดงในรูปที่ 2  E และจากนั้น ระบบจะเริ่มการทำงานทาง I/O เพื่อโหลดหน้าที่ต้องใช้ชั้นสู่พื้นที่เลขที่อยู่ทางกายภาพ เลขที่อยู่ทั้งหมดที่แสดงอยู่ในที่นี่ถูกแสดงในรูปแบบเลขฐานสิบหก (hexadecimal)

ตารางหน้า:	เลขหน้า	L	เลขเฟรม
	0	1	0
	1	1	1
	2	1	2
	3	1	3
	4	0	-
	5	1	7
	6	0	-
	7	1	6
	...	...	...

หมายเหตุ:

L แสดงสถานะการโหลดของหน้า

- ถ้า L เป็น 1 หน้านั้นถูกโหลดอยู่ในเฟรม
- ถ้า L เป็น 0 หน้านั้นไม่ได้ถูกโหลดไว้

รูปที่ 2 ข้อมูล (บางส่วน) ที่อยู่ในตารางหน้า

กลุ่มคำตอบสำหรับ C และ D

- อยู่ที่เลขที่อยู่ทางกายภาพ 05E00
- อยู่ที่เลขที่อยู่ทางกายภาพ 06C00
- อยู่ที่เลขที่อยู่ทางกายภาพ 06E00
- อยู่ที่เลขที่อยู่ทางกายภาพ 07C00
- ไม่ได้ถูกโหลดไว้ในพื้นที่เลขที่อยู่หน่วยความจำทางกายภาพ

กลุ่มคำตอบสำหรับ E

- จะเกิดการผิดหน้า (page fault) เมื่อบิต L ของหน้าที่ถูกเรียกใช้เป็น 0
- จะเกิดการผิดหน้าเมื่อบิต L ของหน้าที่ถูกเรียกใช้เป็น 1
- จะเกิดการขัดจังหวะ I/O (I/O interruption) เมื่อบิต L ของหน้าที่ถูกเรียกใช้เป็น 0
- จะเกิดการขัดจังหวะ I/O เมื่อบิต L ของหน้าที่ถูกเรียกใช้เป็น 1

**Q3. อ่านคำอธิบายต่อไปนี้เกี่ยวกับฐานข้อมูลสำหรับบริษัททาสี แล้วตอบคำถามย่อย 1 ถึง 3**

บริษัทยู (U) เป็น บริษัทบริการทาสีได้ให้ข้อเสนอบริการทาสีภายในและภายนอกขึ้นอยู่กับสัญญาของลูกค้า บริษัท U ให้บริการตามความต้องการของลูกค้า ลูกค้าบางรายต้องการทาสีอาคารทั้งหลังและบางรายต้องการทาสีแค่ในห้องน้ำ ห้องครัวหรือห้องนอนเท่านั้น ลูกค้าบางรายต้องการสีทนน้ำและบางรายต้องการสีที่มีการรับประกัน 3 ปี

บริษัทยู (U) มีสีทาที่มีคุณภาพสูงจากบริษัทที่จัดจำหน่าย (Supplier) หลายบริษัท และสีทาแต่ละชนิดสามารถสั่งได้จากผู้จัดจำหน่ายที่กำหนด

พนักงานของบริษัทยู (U) ทำงานเป็นรายชั่วโมง ลูกค้าสามารถทำสัญญาครั้งละหนึ่งงานหรือมากกว่า และสามารถมอบหมายพนักงานหนึ่งคนหรือมากกว่าให้กับแต่ละงาน

ฐานข้อมูลมี 5 ตาราง โครงสร้างตารางและข้อมูลตัวอย่างของแต่ละตารางดังแสดงด้านล่าง ในที่นี้ส่วนที่ขีดเส้นใต้คือ (\_\_\_) คีย์หลัก (primary key) และเส้นประ (....) คือคีย์นอก (Foreign key)

**(1) ตารางลูกค้า (Customer table)**

ตารางลูกค้าประกอบด้วยข้อมูลที่เกี่ยวข้องกับลูกค้า

<u>CustomerID</u>	CustomerName	CustomerAddress
1	Maung Maung	Yangon
2	Aung Aung	Mandalay
3	Hla Hla	Bagan

**(2) ตารางพนักงาน (Staff table)**

ตารางพนักงานประกอบด้วยข้อมูลที่เกี่ยวข้องกับพนักงาน ฟیلด์ "Rate" คือ อัตราค่าแรงของพนักงานต่อชั่วโมง

<u>StaffID</u>	StaffName	Rank	Rate
1	Kyaw Kyaw	In charge	80
2	Zaw Zaw	Painter	50
3	Min Min	Painter	50

**(3) ตารางผู้จัดจำหน่าย (Supplier table)**

ตารางผู้จัดจำหน่ายประกอบด้วยข้อมูลเกี่ยวกับบริษัทที่จัดจำหน่าย (supplier)

<u>SupplierID</u>	SupplierName	PaintQuality	UnitPrice
1	Daw Saw	Water Resistant	100
2	U Win	Oil Resistant	200
3	Daw Thein	3-Year Guarantee	300



(4) ตารางงาน (Job table)

ตารางงานประกอบด้วยข้อมูลเกี่ยวกับงานที่ได้ทำสัญญาไว้แล้ว พื้นที่ทาสีทั้งหมด (TotalSurfaceArea) มีหน่วยเป็นตารางเมตร

<u>JobID</u>	TotalSurfaceArea	<u>CustomerID</u>	<u>SupplierID</u>
1	100	2	3
2	10	1	1
3	20	1	2
4	30	3	3
5	15	3	1

(5) ตารางงานของพนักงาน (JobStaff table)

ตารางงานของพนักงานเป็นตารางร่วม (joint table) ระหว่างตารางงาน (Job table) กับตารางพนักงาน (Staff table) งานหนึ่งงานสามารถมีพนักงานได้หลายคน และพนักงานหนึ่งคนสามารถทำงานได้หลายงาน

<u>JobStaffID</u>	<u>JobID</u>	<u>StaffID</u>	WorkingHours
1	1	1	2
2	1	2	4
3	1	3	4
4	2	1	5
5	3	2	8
6	4	3	5
7	5	2	8

**คำถามย่อย 1**

จากกลุ่มของคำตอบด้านล่าง จงเลือกคำตอบที่ถูกที่สุดเพื่อเติมในช่องว่าง  ในคำสั่ง SQL ต่อไปนี้

คำสั่ง SQL นี้เรียกว่า SQL1 จะแสดงผลลัพธ์เป็นชื่อพนักงานและจำนวนงานทั้งหมดของพนักงานที่มีงานมากกว่า 2 งาน

```
-- SQL1 --  
SELECT S.StaffID, S.StaffName,  A AS TotalJob  
FROM Job J, Staff S, JobStaff JS  
WHERE J.JobID = JS.JobID AND S.StaffID = JS.StaffID  
 B
```

จากตัวอย่างข้อมูลที่อยู่ในตารางข้างต้น ผลลัพธ์จากการรันคำสั่ง SQL1 เป็นดังนี้

StaffID	StaffName	TotalJob
2	Zaw Zaw	3

กลุ่มคำตอบสำหรับ A

- a) AVG(J.TotalSurfaceArea)
- b) COUNT(\*)
- c) J.JobID
- d) SUM(\*)

กลุ่มคำตอบสำหรับ B

- a) GROUP BY S.StaffID, S.StaffName
- b) GROUP BY S.StaffID, S.StaffName HAVING COUNT(\*) > 2
- c) HAVING COUNT(\*) > 2
- d) HAVING COUNT(\*) >= 2 GROUP BY S.StaffID, S.StaffName

## คำถามย่อย 2

จากกลุ่มของคำตอบด้านล่าง จงเลือกคำตอบที่ถูกที่สุดเพื่อเติมในช่องว่าง  ในคำสั่ง SQL ต่อไปนี้

คำสั่ง SQL นี้เรียกว่า SQL2 จะแสดงผลลัพธ์เป็น job IDs กับพื้นที่ทาสีทั้งหมด สำหรับลูกค้าที่รับประกัน 3 ปี (3-Year Guarantee)

```
-- SQL2 --
SELECT J.JobID, J.TotalSurfaceArea
FROM Job J, Supplier S
 C J.SupplierID = S.SupplierID
AND S.PaintQuality = '3-Year Guarantee'
```

จากตัวอย่างข้อมูลที่อยู่ในตารางข้างต้น ผลลัพธ์จากการรันคำสั่ง SQL2 เป็นดังนี้

JobID	TotalSurfaceArea
1	100
4	30

กลุ่มคำตอบสำหรับ C

- a) GROUP BY
- b) HAVING
- c) ORDER BY
- d) WHERE

### คำถามย่อย 3

จากกลุ่มของคำตอบด้านล่าง จงเลือกคำตอบที่ถูกที่สุดเพื่อเติมในช่องว่าง  ในคำสั่ง SQL ต่อไปนี้

คำสั่ง SQL นี้เรียกว่า SQL3 จะแสดงผลลัพธ์ของพื้นที่ทาสีทั้งหมดกับค่าแรงทั้งหมดของแต่ละงาน

```
-- SQL3 --  
SELECT JobID, TotalSurfaceArea,  AS TotalLaborCosts  
FROM (SELECT JS.JobID AS JobID,  
            J.TotalSurfaceArea,  AS LaborCosts  
      FROM Job J, Staff S, JobStaff JS  
     WHERE JS.StaffID = S.StaffID AND JS.JobID = J.JobID) LC  
GROUP BY JobID, TotalSurfaceArea
```

จากตัวอย่างข้อมูลที่อยู่ในตารางข้างต้น ผลลัพธ์จากการรันคำสั่ง SQL3 เป็นดังนี้

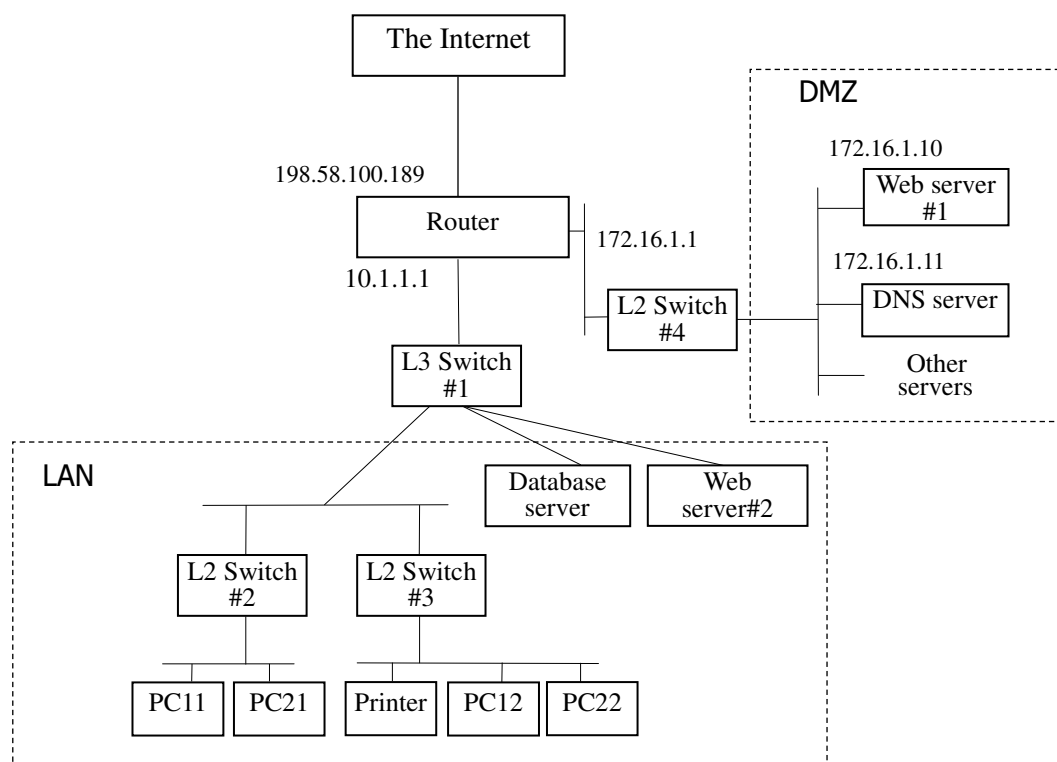
JobID	TotalSurfaceArea	TotalLaborCosts
1	100	560
2	10	400
3	20	400
4	30	250
5	15	400

กลุ่มคำตอบสำหรับ D และ E

- a) LC.LaborCosts
- b) SUM(LC.LaborCosts)
- c) SUM(WorkingHours \* S.Rate)
- d) SUM(WorkingHours) \* S.Rate
- e) WorkingHours \* S.Rate

**Q4.** อ่านคำอธิบายต่อไปนี้เกี่ยวกับเครือข่ายในมหาวิทยาลัยแห่งหนึ่ง แล้วตอบคำถามย่อย 1 และ 2

เพื่อการปรับปรุงความมั่นคงปลอดภัยให้ดีขึ้น เครือข่ายของมหาวิทยาลัยแห่งหนึ่งจะถูกแบ่งเป็นสองเซกเมนต์ คือ Demilitarized Zone (DMZ) และ Local Area Network (LAN) โดยเครือข่าย LAN เป็นส่วนของเครือข่ายภายในที่ใช้จัดเก็บข้อมูลทุกชนิด และการเข้าถึงใด ๆ จากภายนอกไปยังข้อมูลของมหาวิทยาลัยจะกระทำผ่าน DMZ ผู้บริหารเครือข่าย (network administrator) ได้สร้างนโยบายความมั่นคงปลอดภัยมาจำนวนหนึ่งบนไฟร์วอลล์ตามมาตรฐานความมั่นคงปลอดภัยของมหาวิทยาลัย โดยข้อมูลที่ต้องการเปิดเผยสู่สาธารณะต้องส่งไปยัง DMZ เสียก่อน เราเตอร์สามารถทำงานในหน้าที่ของไฟร์วอลล์ได้ รูปที่ 1 แสดงการกำหนดค่าเครือข่ายของมหาวิทยาลัยแห่งนี้



รูปที่ 1 การกำหนดค่าเครือข่ายของมหาวิทยาลัย

### คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมในช่องว่าง  แต่ละช่องในตารางที่ 1

เพื่อป้องกันเซิร์ฟเวอร์ที่อยู่ใน DMZ ผู้บริหารเครือข่ายทำการสร้างกฎการรับส่งข้อมูล (traffic rule) ดังแสดงในตารางที่ 1 บนเราเตอร์ ตามกฎต่อไปนี้:

- (1) เว็บเซอร์วิสที่ปลอดภัย (secure web services) บนไอพีแอดเดรสภายนอกของเราเตอร์จะถูกแมปไปยังเว็บเซิร์ฟเวอร์ #1 ใน DMZ
- (2) เซิร์ฟเวอร์ทั้งหมดสามารถเข้าถึงอินเทอร์เน็ตจาก DMZ โดยใช้ NAT (Network Address Translation)
- (3) ทำให้เซิร์ฟเวอร์ทั้งหมดสามารถเข้าถึงได้จากผู้บริหารเครือข่ายที่อยู่ใน LAN
- (4) แม้ว่าจะมีเซิร์ฟเวอร์ใน DMZ ที่ถูกโจมตี แต่ผู้โจมตีจะไม่สามารถเข้าถึงอุปกรณ์ที่อยู่ใน LAN ได้

กำหนดให้ DMZ ใช้เครือข่ายย่อย (subnet) 172.16.1.0/26

ตารางที่ 1 กฎการรับส่งข้อมูลบนเราเตอร์ (การกำหนดค่าอื่น ๆ ไม่ได้ถูกแสดงไว้)

Source	Destination	Services	Action	Translation
the Internet	<input type="text" value="A"/>	https	<input type="text" value="B"/>	Map 172.16.1.10
DMZ	the Internet	<input type="text" value="C"/>	Allow	NAT
<input type="text" value="D"/>	DMZ	ssh	Allow	-

กลุ่มคำตอบ

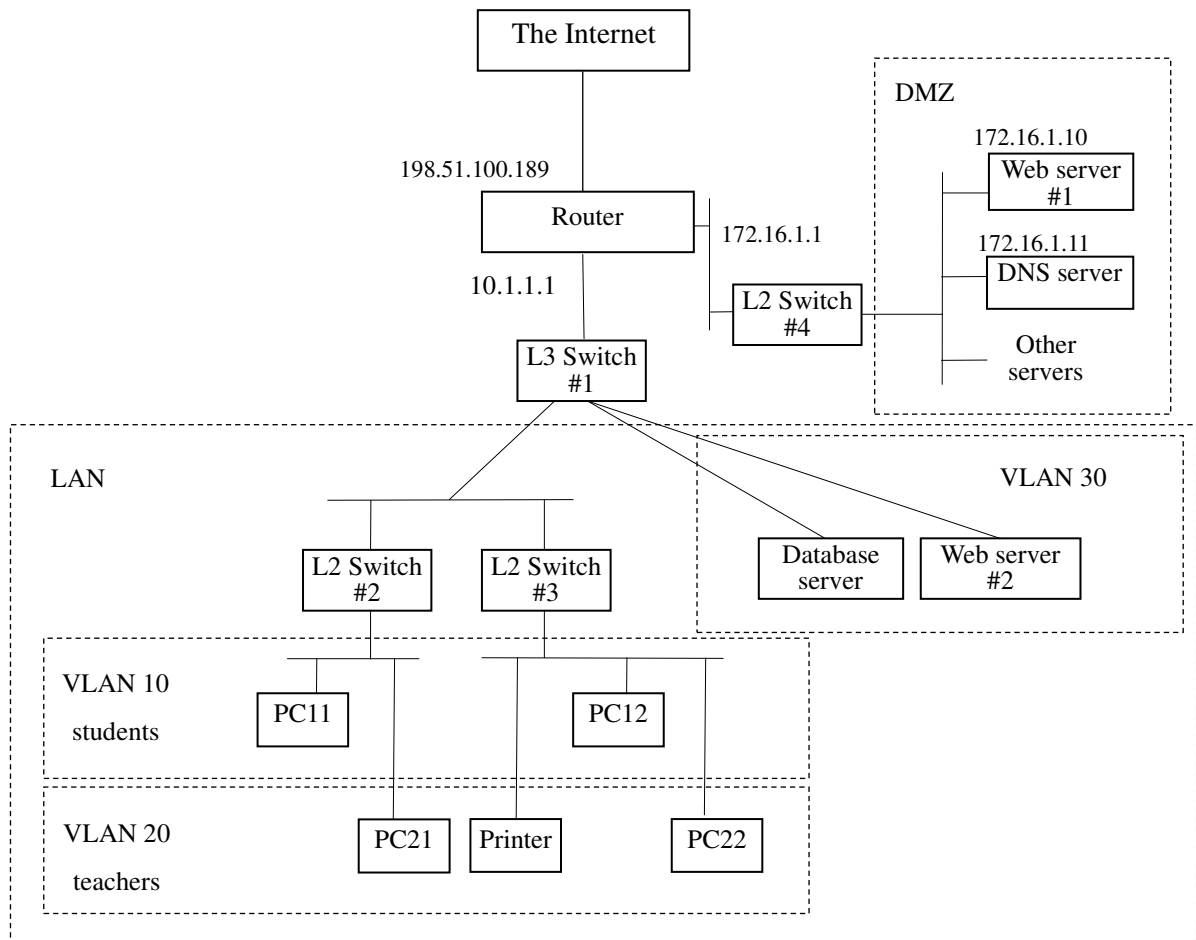
- |             |               |                |                   |
|-------------|---------------|----------------|-------------------|
| a) 10.1.1.1 | b) 172.16.1.1 | c) 172.16.1.10 | d) 198.58.100.189 |
| e) Allow    | f) Any        | g) Deny        | h) https          |
| i) LAN      | j) ssh        |                |                   |

### คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมในช่องว่าง  ในคำอธิบายต่อไปนี้

เพื่อเป็นการปรับปรุงความมั่นคงปลอดภัยของเครือข่ายให้ดีขึ้น ผู้บริหารเครือข่ายวางแผนเพื่อตั้ง VLANs สองวง ด้วยการแบ่งเซกเมนต์ LAN ที่มีอยู่โดยไม่ได้เคลื่อนย้ายอุปกรณ์ต่าง ๆ ทางกายภาพ VLAN วงแรกชื่อ VLAN 10 ถูกกำหนดไว้สำหรับนักศึกษา (students) และอีกวงหนึ่งชื่อ VLAN 20 สำหรับอาจารย์ (teachers) สวิตช์ทุกตัวมีคุณสมบัติทอดข้าม VLAN (VLAN spanning) และทุกอุปกรณ์ที่ถูกวางไว้ภายใต้ VLAN เดียวกัน สามารถข้ามสวิตช์เพื่อเข้าถึงซึ่งกันและกันได้

รูปที่ 2 แสดงการกำหนดโครงแบบเครือข่าย (network configuration) ของมหาวิทยาลัยที่ออกแบบใหม่ กำหนดให้ L3 สวิตช์ #1 (L3 Switch #1) สามารถเข้าถึงได้จากทุก VLANs



รูปที่ 2 การกำหนดโครงข่ายของมหาวิทยาลัยที่ออกแบบใหม่

เมื่อ PC21 ใน VLAN 20 พยายามเชื่อมต่อไปยังเว็บไซต์ <https://www.info.sample.edu> ที่รันอยู่บนเว็บเซิร์ฟเวอร์ #1 แล้วนั้น PC21 ส่งข้อความ (message) ไป 2 ข้อความ ก่อนที่จะส่งข้อความร้องขอ http (http request message)

- (1) PC21 ส่งข้อความ broadcast เพื่อให้ได้รับแมคแอดเดรสของ L3 Switch #1
  - (2) PC21 ส่งข้อความสอบถาม (query message) ในเครือข่ายท้องถิ่นเพื่อแปลง (resolve) URL
- Message (1) สามารถมองเห็นโดยอุปกรณ์  เท่านั้น  
 Message (2) ถูกส่งไปที่

กลุ่มคำตอบสำหรับ E

- a) ใน DMZ
- b) ใน VLAN 20
- c) ที่เชื่อมต่ออยู่กับ L2 Switch #2
- d) ที่เชื่อมต่ออยู่กับ L2 Switch #2 และ L2 Switch #3

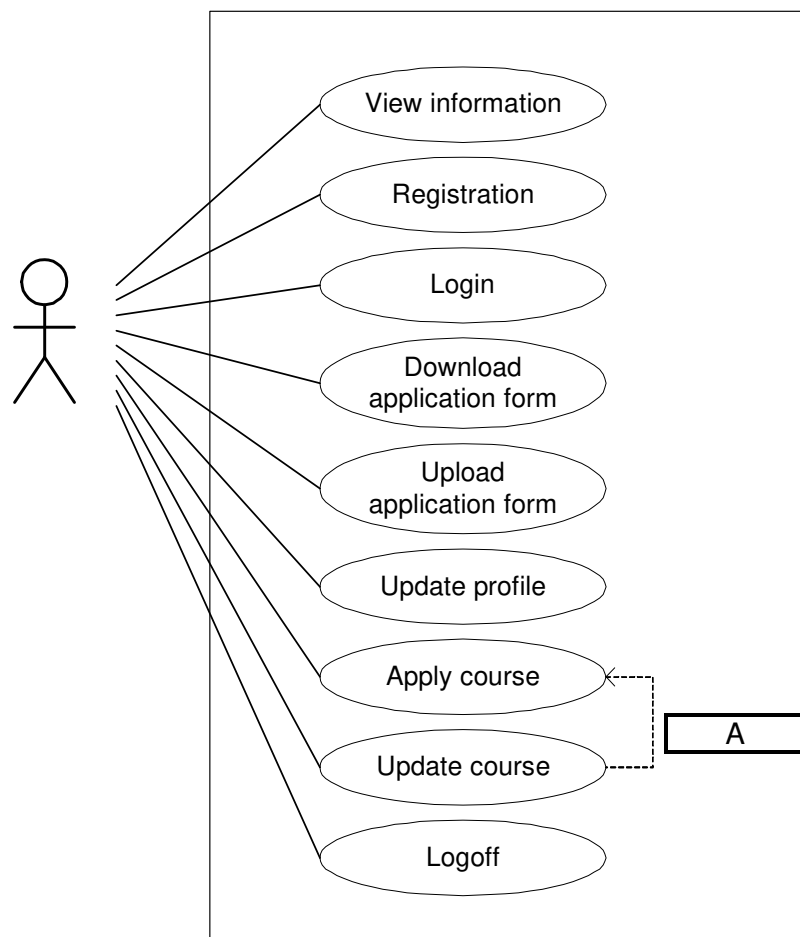
กลุ่มคำตอบสำหรับ F

- a) Database server
- b) DNS server
- c) Web server #1
- d) Web server #2

**Q5.** อ่านคำอธิบายของระบบบริหารจัดการฝึกอบรม แล้วตอบคำถามต่อไปนี้

บริษัท W เป็นศูนย์อบรมทางด้านเทคโนโลยีสารสนเทศ (IT) บริษัทนี้ใช้ระบบบริหารจัดการการฝึกอบรม (จากนี้ไปจะเรียกสั้น ๆ ว่า "ระบบ") เพื่อจัดการข้อมูลผู้เข้าอบรมและหลักสูตรฝึกอบรม ทุก ๆ คนสามารถเรียกดูข้อมูลการฝึกอบรมได้ผ่านเว็บไซต์ แต่ถ้าบุคคลใดต้องการเข้าร่วมอบรมหลักสูตรทางด้านเทคโนโลยีสารสนเทศ บุคคลนั้นจะต้องลงทะเบียนข้อมูลผู้เข้าร่วมอบรมผ่านระบบเสียก่อน หลังจากลงทะเบียนแล้ว บุคคลนั้น ๆ จะได้รับชื่อผู้ใช้และรหัสผ่านทางอีเมลจากระบบดังกล่าว เมื่อผู้เข้าอบรมเข้าสู่ระบบโดยชื่อผู้ใช้และรหัสผ่านของตนเอง ผู้เข้าร่วมอบรมต้องดาวน์โหลดใบสมัครกรอกข้อมูลที่จำเป็น แล้วอัปโหลดใบสมัครเพื่อเข้าเรียนในหลักสูตรที่ต้องการ บางหลักสูตรจะมีหลายชั้นเรียน: ชั้นเรียนคู่ขนาน เช่น เรียนวันจันทร์ และวันศุกร์ และชั้นเรียนตามช่วงเวลา เช่น เรียนช่วงเวลาฤดูหนาว และฤดูใบไม้ผลิ

รูปที่ 1 แสดงแผนภาพยูสเคส (use case diagram) ของระบบ



รูปที่ 1 แสดงแผนภาพยูสเคส (use case diagram) ของระบบ

ตารางที่ 1 แสดงรายละเอียดคำอธิบายของแต่ละยูสเคส (use case) ตามรูปที่ 1

ตารางที่ 1 คำอธิบายของแต่ละยูสเคส (use case) ตามรูปที่ 1

ลำดับ	รายละเอียดคำอธิบาย
01	View information ผู้เข้าร่วมอบรมสามารถเรียกดูข้อมูลทั้งหมดที่เกี่ยวกับการฝึกอบรมเทคโนโลยีสารสนเทศได้
02	Registration ถ้าผู้เข้าร่วมอบรมต้องการสมัครหลักสูตรฝึกอบรม ผู้เข้าร่วมอบรมจำเป็นต้องลงทะเบียนข้อมูลตนเองในระบบ จากนั้นระบบจะกำหนดชื่อผู้ใช้และรหัสผ่านให้
03	Login ถ้าผู้เข้าร่วมอบรมต้องการแก้ไขข้อมูลส่วนตัว สมัครหลักสูตรฝึกอบรม หรือแก้ไขหลักสูตรฝึกอบรมของตน ผู้เข้าร่วมอบรมจะต้องเข้าสู่ระบบ
11	Download application form ถ้าผู้เข้าร่วมอบรมต้องการสมัครหลักสูตรฝึกอบรม ผู้เข้าร่วมอบรมต้องดาวน์โหลดใบสมัคร
12	Upload application form หลังจากกรอกข้อมูลที่จำเป็นแล้ว ผู้เข้าร่วมอบรมจะต้องอัปโหลดใบสมัคร และหลังจากอัปโหลดแล้ว ผู้เข้าร่วมอบรมจะต้องยืนยันหลักสูตรฝึกอบรมที่ตนเองสมัคร
13	Update profile ผู้เข้าร่วมอบรมสามารถแก้ไขข้อมูลที่ลงทะเบียนไว้ได้
14	Apply course ผู้เข้าร่วมอบรมจะต้องยืนยันการสมัครเรียนหลักสูตรฝึกอบรมที่ต้องการได้สมัครไว้ หลังจากสมัครแล้ว ผู้เข้าร่วมอบรมจะได้รับข้อมูลหลักสูตรฝึกอบรมผ่านอีเมล
15	Update course ผู้เข้าร่วมอบรมสามารถยกเลิกการสมัครเรียน หรือเปลี่ยนชั้นเรียนของหลักสูตรที่สมัครไว้ได้ หลังจากแก้ไขหลักสูตรแล้ว ผู้เข้าร่วมอบรมจะได้รับการแจ้งเตือนการแก้ไขผ่านทางอีเมล
16	Logoff ออกจากระบบ

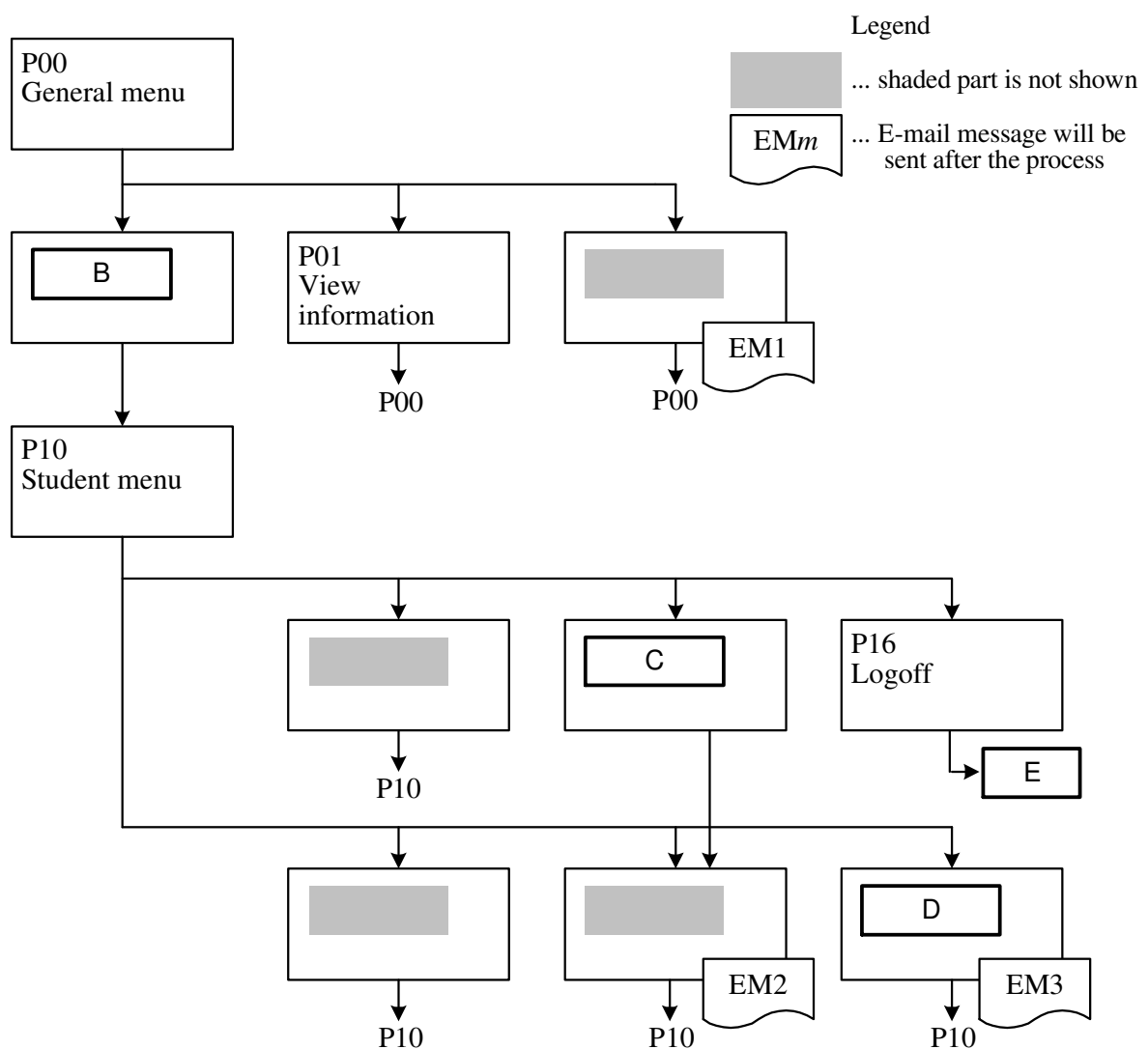
รูปที่ 2 แสดงแผนภาพการเปลี่ยนสถานะหน้าจอ (screen transition diagram) ของระบบ

ในรูปที่ 2 *Pnn* แสดงถึงหน้าจอหนึ่ง ๆ ในที่นี้ นอกเหนือจาก P00 และ P10 แล้ว *nn* จะสอดคล้องกับหมายเลขของยูสเคส (use case) ในตารางที่ 1 และ *EMnn* แสดงถึงข้อความในอีเมล

#### คำถามย่อย

จากชุดคำตอบข้างล่างนี้ ให้เลือกคำตอบที่เหมาะสมกับช่องว่าง  ในรูปที่ 1 และรูปที่ 2





รูปที่ 2 แผนภาพการเปลี่ยนสถานะหน้าจอ (screen transition diagram) ของระบบ

ชุดคำตอบสำหรับ A

a) << extend >>

b) << include >>

ชุดคำตอบสำหรับ B ถึง D

a) P02 Registration

b) P03 Login

c) P11 Download application form

d) P12 Upload application form

e) P13 Update profile

f) P14 Apply course

g) P15 Update course

ชุดคำตอบสำหรับ E

a) P01 หรือ P10 อย่างใดอย่างหนึ่ง

b) P00

c) P02

d) P10

**Q6.** อ่านคำอธิบายของโปรแกรมและโค้ดของโปรแกรม แล้วตอบคำถามย่อย 1 และ 2 (มีการปรับปรุงสัญลักษณ์ที่ใช้ในรหัสเทียม ซึ่งอยู่ที่ส่วนต้นของข้อสอบฉบับนี้)

[คำอธิบายโปรแกรม 1]

ในงานปาร์ตี้แห่งหนึ่งมีผู้คนเข้าไปร่วมงานจำนวน  $N$  คน (กำหนดหมายเลขให้ตั้งแต่ 1 ถึง  $N$ ) โดยในตอนท้ายของงานปาร์ตี้ เค้กจำนวน  $M$  ก้อนได้ถูกจัดทำเป็นของขวัญเพื่อมอบให้กับคนที่มาร่วมงาน โดยเค้กก้อนที่  $i$  นั้นจะมีน้ำหนัก  $W[i]$  และได้มอบให้กับบุคคล  $P[i]$  โดยแต่ละคนจะได้รับเค้กเป็นของขวัญอย่างน้อยคนละหนึ่งก้อน และก่อนที่จะออกจากงานปาร์ตี้แต่ละคนก็สามารถแลกเปลี่ยนเค้กกับคนอื่นได้ ยกตัวอย่างเช่น นาย A สามารถมอบเค้กที่ได้รับให้กับนาย B และรับเค้กกลับมาจากนาย B และการแลกเปลี่ยนจะถูกเรียกว่าเป็นการแลกเปลี่ยน "มิตรภาพ" หากความแตกต่างของน้ำหนักรวมของของขวัญที่แต่ละคนเป็นเจ้าของระหว่างสองคนนั้นลดลงหลังจากการแลกเปลี่ยน โดยที่โปรแกรมย่อย Exchange จะแสดงวิธีการแลกเปลี่ยนแบบ "มิตรภาพ" ทั้งหมดที่เป็นไปได้ระหว่างผู้เข้าร่วมงาน

(1) ตารางที่ 1 แสดงตัวอย่างการจัดวางของขวัญ ทั้งนี้จากตารางที่ 1 แสดงให้เห็นว่า

- บุคคลคนที่ 1 (แสดงด้วย  $P[4]$ ) ได้รับเค้กก้อนที่ 4 ที่มีน้ำหนัก 8
- บุคคลคนที่ 2 (แสดงด้วย  $P[1]$  และ  $P[3]$ ) ได้รับเค้กก้อนที่ 1 และก้อนที่ 3 ซึ่งมีน้ำหนัก 10 และ 6 ตามลำดับ
- บุคคลคนที่ 3 (แสดงด้วย  $P[2]$ ) ได้รับเค้กก้อนที่ 2 ที่มีน้ำหนัก 3
- บุคคลคนที่ 4 (แสดงด้วย  $P[5]$ ) ได้รับเค้กก้อนที่ 5 ที่มีน้ำหนัก 9

ตารางที่ 1 ตัวอย่างการจัดสรรของขวัญ

$i$	1	2	3	4	5
น้ำหนักของเค้ก ( $W[i]$ )	10	3	6	8	9
บุคคลที่ได้รับเค้ก ( $P[i]$ )	2	3	2	1	4

(2) จากข้อมูลที่แสดงในตารางที่ 1 พบว่ามี 4 การแลกเปลี่ยนที่เป็นแบบมิตรภาพ ซึ่งรายการด้านล่างนี้เป็นผลลัพธ์ของการเรียกใช้ฟังก์ชัน Exchange (4, 5,  $W[]$ ,  $P[]$ ) โดยใช้พารามิเตอร์  $W[]$  และ  $P[]$  จากตารางที่ 1

Exchange between cake 1 and 2  
 Exchange between cake 1 and 4  
 Exchange between cake 1 and 5  
 Exchange between cake 3 and 2

(3) สมมติว่ามีการแลกเปลี่ยนเค้กที่มีน้ำหนัก  $x$  ของบุคคลหนึ่งซึ่งมีน้ำหนักของเค้กรวมเท่ากับ  $T_x$  กับเค้กน้ำหนัก  $y$  ของอีกบุคคลหนึ่งที่มีน้ำหนักของเค้กรวมเท่ากับ  $T_y$   
 ถ้า  $T_x > T_y$  และการแลกเปลี่ยนนั้นเป็นแบบ "มิตรภาพ" แล้ว:

$$T_x - T_y > |(T_x - x + y) - (T_y - y + x)| \Rightarrow x - y > 0 \text{ และ } T_x - T_y > x - y \quad (\alpha)$$

(4) ข้อกำหนดของอาร์กิวเมนต์ (argument) ของโปรแกรมย่อย Exchange ถูกแสดงในตารางที่ 2

ตารางที่ 2 ข้อกำหนดของอาร์กิวเมนต์ (Argument) สำหรับโปรแกรมย่อย Exchange

ตัวแปร	อินพุต/ เอาต์พุต	คำอธิบาย
N	อินพุต	จำนวนคนที่เข้าร่วมปาร์ตี้
M	อินพุต	จำนวนของเค้กที่มอบให้คนร่วมงาน (ในที่นี้ $M \geq N$ )
W[] P[]	อินพุต	อาร์เรย์สองอันที่แต่ละอันมีเอเลเมนต์จำนวน M ตัว โดยที่ค่าของ W[i] และ P[i] แสดงถึงน้ำหนักของเค้กชิ้นที่ i และบุคคลผู้ได้รับเค้กก่อนนั้น (ในที่นี้ $1 \leq P[i] \leq N$ )

(5) โปรแกรมย่อย Print จะแสดงข้อความเอาต์พุต ยกตัวอย่างเช่น Print("Exchange between cake ", I, " and ", J) จะแสดงเอาต์พุตเป็น "Exchange between cake 1 and 2" ถ้าตัวแปร I มีค่าเป็น 1 และตัวแปร J มีค่าเป็น 2

โปรแกรมย่อย Exchange คำนวณน้ำหนักรวมของเค้กของแต่ละคนก่อน แล้วจึงเก็บค่าที่ได้ไว้ในอาร์เรย์ T และจากข้อมูลเหล่านี้ การจัดกลุ่มที่เป็นไปได้ทั้งหมด (all combinations) ของเค้กสองก้อนที่สอดคล้องกับสมการ ( $\alpha$ ) จะถูกเอาต์พุตออกมา ในที่นี้ดัชนีของอาร์เรย์เริ่มต้นที่ 1

[โปรแกรม 1]

```

FUNCTION: Exchange(INT: N, INT: M, INT: W[], INT: P[]) {
    INT: T[], I, J

    FOR (I ← 1; I ≤ N; I ← I + 1) {
        T[I] ← A;
    }
    FOR (I ← 1; I ≤ M; I ← I + 1) {
        T[P[I]] ← B;
    }
    FOR (I ← 1; I ≤ M; I ← I + 1) {
        FOR (J ← 1; J ≤ M; J ← J + 1) {
            IF ((T[P[I]] > T[P[J]]) C1
                (W[I] > W[J]) C2
                ((T[P[I]] - T[P[J]]) > (W[I] - W[J]))) {
                Print("Exchange between cake ", I, " and ", J);
            }
        }
    }
}

```

### คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องในโปรแกรม 1 ในที่นี้ คำตอบที่จะเติมลงในช่อง C1 และ C2 จะต้องเลือกจากการจัดกลุ่มที่เหมาะสมจากกลุ่มคำตอบสำหรับ C

กลุ่มคำตอบสำหรับ A

- |              |           |              |
|--------------|-----------|--------------|
| a) 0         | b) $P[I]$ | c) $P[W[I]]$ |
| d) $T[P[I]]$ | e) $W[I]$ | f) $W[P[I]]$ |

กลุ่มคำตอบสำหรับ B

- |                     |                        |
|---------------------|------------------------|
| a) $T[I] + W[I]$    | b) $T[I] + W[P[I]]$    |
| c) $T[P[I]] + W[I]$ | d) $T[P[I]] + W[P[I]]$ |
| e) $W[I]$           | f) $W[P[I]]$           |

กลุ่มคำตอบสำหรับ C

	C1	C2
a)	and	And
b)	and	or
c)	or	And
d)	or	or

### คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องในคำอธิบายต่อไปนี้ ในที่นี้คำตอบที่จะถูกเติมลงในช่อง F1, F2 และ F3 จะต้องเลือกจากการจัดกลุ่มที่เหมาะสมจากกลุ่มคำตอบสำหรับ F

[คำอธิบายโปรแกรม 2]

โปรแกรมย่อย RepeatExchange เป็นการแลกเปลี่ยนค่ากันต่อไปครั้งแล้วครั้งเล่า ตามกฎที่กล่าวไว้ข้างต้น โดยจะเอาที่พูดลำดับการแลกเปลี่ยนและน้ำหนักรวมของเค้ของแต่ละคนหลังจากแลกเปลี่ยนตามลำดับจนเสร็จสิ้น ข้อกำหนดของอาร์กิวเมนต์สำหรับโปรแกรมย่อย RepeatExchange นั้นจะเหมือนกับโปรแกรมย่อย Exchange ในที่นี้ สมมติว่าคำตอบที่ถูกต้องถูกเติมลงในช่องว่าง  A จนถึง  C2 ในโปรแกรมที่ 2 แล้ว

[โปรแกรม 2]

```

FUNCTION: RepeatExchange(INT: N, INT: M, INT: W[], INT: P[]) {
    INT: T[], I, J, FLAG, TEMP

    FLAG ← 0;
    FOR (I ← 1; I ≤ N; I ← I + 1) {
        T[I] ← A;
    }
    FOR (I ← 1; I ≤ M; I ← I + 1) {
        T[P[I]] ← B;
    }
    WHILE (FLAG = 0) {
        FLAG ← 1;
        FOR (I ← 1; I ≤ M; I ← I + 1) {
            FOR (J ← 1; J ≤ M; J ← J + 1) {
                IF ((T[P[I]] > T[P[J]]) C1
                    (W[I] > W[J]) C2
                    ((T[P[I]] - T[P[J]]) > (W[I] - W[J]))) {
                    Print("Exchange between cake ", I, " and ", J);
                    T[P[I]] ← T[P[I]] - W[I] + W[J];
                    T[P[J]] ← T[P[J]] - W[J] + W[I];
                    TEMP ← P[I];
                    P[I] ← P[J];
                    P[J] ← TEMP;
                    FLAG ← 0;
                }
            }
        }
    }
    FOR (I ← 1; I ≤ N; I ← I + 1) {
        Print("Total weight of person ", I, " is ", T[I]);
    }
}

```

ตารางที่ 3 ตัวอย่างการจัดสรรของขั้วญอีกลักษณะหนึ่ง

ตารางที่ 3 ตัวอย่างการจัดสรรของขั้วญอีกลักษณะหนึ่ง

<i>i</i>	1	2	3	4	5	6	7	8	9
w[i]	15	12	9	8	6	1	1	1	1
P[i]	1	1	1	2	2	2	3	3	3

ข้อความเอาต์พุตต่อไปนี้เป็นผลลัพธ์ของการเรียกใช้ฟังก์ชัน RepeatExchange(3, 9, W[], P[]) โดยใช้พารามิเตอร์ W[] และ P[] จากตารางที่ 3

Exchange between cake 1 and 4  
 Exchange between cake 1 and 7  
 Exchange between cake   
 Exchange between cake 3 and 6  
 Exchange between cake   
 Total weight of person 1 is   
 Total weight of person 2 is   
 Total weight of person 3 is

กลุ่มคำตอบสำหรับ D และ E

- |            |            |            |            |
|------------|------------|------------|------------|
| a) 1 and 8 | b) 2 and 4 | c) 2 and 5 | d) 2 and 6 |
| e) 2 and 7 | f) 3 and 4 | g) 3 and 5 | h) 4 and 6 |
| i) 4 and 7 | j) 7 and 8 |            |            |

กลุ่มคำตอบสำหรับ F

	F1	F2	F3
a)	16	18	20
b)	17	18	19
c)	17	19	18
d)	18	17	19
e)	18	18	18
f)	18	19	17
g)	19	17	18
h)	19	18	17

ให้เลือกตอบคำถาม Q7 หรือ Q8 เพียงหนึ่งข้อ

โดยการระบายทึบลงในวงกลม (S) ข้อที่ต้องการ และตอบคำถามข้อที่เลือก

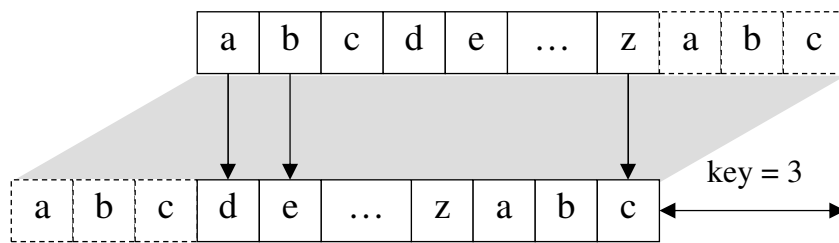
หากเลือกทั้งสองข้อ จะให้คะแนนเฉพาะข้อแรกเท่านั้น

**Q7.** อ่านคำอธิบายของโปรแกรมภาษา C และตัวโปรแกรมต่อไปนี้ จากนั้นให้ตอบคำถามย่อย 1 และ 2

การเข้ารหัสของซีซาร์ (Caesar's cipher) เป็นการเข้ารหัสอย่างง่าย ที่ใช้เทคนิคการทดแทน เมื่อมีข้อความ (ตัวหนังสือที่ถูกเข้ารหัส) และคีย์ (ตัวเลขจำนวนเต็ม  $k$ ) ตัวอักษรแต่ละตัวในข้อความจะถูกแทนที่ด้วยตัวอักษรที่ตำแหน่ง  $k$  ลำดับถัดไปตามลำดับตัวอักษร

ตัวอย่างเช่น เมื่อข้อความเป็น "abz" และ คีย์คือจำนวนเต็ม 3 ดังนั้นตัวอักษร "a" ในข้อความจึงถูกแทนที่ด้วยตัวอักษร "d" ซึ่งอยู่ถัดไป 3 ตำแหน่งในลำดับตัวอักษร ( $a \rightarrow b \rightarrow c \rightarrow d$ ) และในรูปแบบเดียวกัน "b" ก็จะถูกแทนที่ด้วย "e" ( $b \rightarrow c \rightarrow d \rightarrow e$ ) และ "z" ก็จะถูกแทนที่ด้วย "c" ( $z \rightarrow a \rightarrow b \rightarrow c$ )

ในการเข้ารหัสนี้ ตัวอักษรจะย้อนกลับไปยัง "a", "b", ... ตามหลัง "z" ดังนั้น ข้อความ "abz" ที่กำหนดให้ ก็จะถูกเข้ารหัสเป็น "dec" ดังรูปที่ 1 แสดงให้เห็นตัวอย่างการเข้ารหัสด้วยวิธีการเข้ารหัสของซีซาร์ (Caesar's cipher)



รูปที่ 1 ตัวอย่างการเข้ารหัสด้วยวิธีการเข้ารหัสของซีซาร์ (Caesar's cipher)

[คำอธิบายโปรแกรม]

(1) โปรแกรมอ่านอินพุตสองค่า คือ ข้อความ และคีย์

(i) ข้อความเป็นสายอักขระที่มีส่วนประกอบเป็นตัวอักษรภาษาอังกฤษเพียงอย่างเดียว (ตัวพิมพ์ใหญ่ 26 ตัวอักษร "A" ถึง "Z" และ ตัวพิมพ์เล็ก จำนวน 26 ตัวอักษร "a" ถึง "z") ความยาวของข้อความอยู่ระหว่าง 1 ถึง 100 ตัวอักษร

(ii) คีย์สำหรับการเข้ารหัส เป็นเลขจำนวนเต็ม ระหว่าง 1 ถึง 25

(2) โปรแกรมประกอบด้วยฟังก์ชันต่าง ๆ ดังนี้

(i) ฟังก์ชัน encrypt ทำหน้าที่เข้ารหัสข้อความที่กำหนดให้ ด้วยวิธีการเข้ารหัสของซีซาร์โดยใช้คีย์ที่กำหนด แล้วคืนค่าด้วยการอ้างอิง (reference) ไปยังข้อความที่ถูกเข้ารหัสแล้ว

(ii) ฟังก์ชัน is\_alpha จะตรวจสอบว่าข้อความที่กำหนดให้มีเพียงตัวอักษรภาษาอังกฤษหรือไม่ โดยคืนค่าเป็น 1 หากข้อความมีเพียงตัวอักษรภาษาอังกฤษ และคืนค่า 0 หากไม่เป็นเช่นนั้น

(iii) ฟังก์ชัน main อ่านอินพุตสองค่าจากอินพุตมาตรฐาน ทำการเรียกฟังก์ชัน encrypt และคืนค่าเอาต์พุตเป็นข้อความที่เข้ารหัสแล้วผ่านเอาต์พุตมาตรฐาน หากข้อความมีตัวอักษรอื่นปะปน ก็จะแจ้งให้กรอกข้อความใหม่ นอกจากนี้หาก ค่าคีย์มีค่าน้อยกว่า 1 หรือมากกว่า 25 ก็

จะแจ้งเตือนให้ใส่ค่าใหม่ด้วยเช่นกัน

(3) ผลลัพธ์ที่ได้ต่อไปนี้เป็นตัวอย่างเอาท์พุตที่ได้จากการทำงานของโปรแกรม

```
Enter a message to encrypt: abz89
Enter a message to encrypt: AbZ
Enter key: 0
Enter key: 3
Encrypted message: DeC
```

[โปรแกรม]

```
#include <stdio.h>

#define MAX_LENGTH (100)

void encrypt(char , int key) {
    char ch;
    int i;

    for (i = 0; message[i] != '\0' && message[i] != '\n' &&
        message[i] != '\r'; i++) {
        ch = message[i];
        if (ch >= 'a' && ch <= 'z') {
            ch = ;
        }
        else {
            ch = ;
        }
        message[i] = ch;
    }
}

int is_alpha(const char* message) {
    char ch;
    int i;

    for (i = 0; message[i] != '\0' && message[i] != '\n' &&
        message[i] != '\r'; i++) {
        ch = message[i];
        if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
            continue;
        }
        return 0;
    }
    return i != 0;
}
```



```

int main() {
    char message[MAX_LENGTH + 2];
    int key;

    do {
        printf("Enter a message to encrypt: ");
        fgets(message, MAX_LENGTH, stdin);
    } while (  );

    do {
        printf("Enter a key: ");
        if (scanf("%d", &key) == 0) {
            continue;
        }
    } while (  );    /* α */

    encrypt(message, key);
    printf("Encrypted message: %s\n", message);
    return 0;
}

```

### คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ A

- |             |              |
|-------------|--------------|
| a) &message | b) **message |
| c) *message | d) message   |

กลุ่มคำตอบสำหรับ B และ C

- |                          |                                |
|--------------------------|--------------------------------|
| a) 'A' + (ch + key) % 26 | b) 'A' + (ch - 'A' + key) % 26 |
| c) 'a' + (ch + key) % 26 | d) 'a' + (ch - 'a' + key) % 26 |
| e) (ch + key) % 26       | f) (ch - key) % 26             |
| g) ch + key              | h) ch - key                    |

กลุ่มคำตอบสำหรับ D และ E

- |                        |                          |
|------------------------|--------------------------|
| a) !is_alpha(*message) | b) !is_alpha(message)    |
| c) is_alpha(*message)  | d) is_alpha(message)     |
| e) key < 1    key > 25 | f) key <= 1    key >= 25 |
| g) key > 1 && key < 25 | h) key >= 1 && key <= 25 |

## คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง จงเลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง  แต่ละช่องในคำอธิบายต่อไปนี้

โปรแกรมทำการแปลงข้อความ "abz" ให้เป็น "dec" ด้วยค่าคีย์ 3 แล้วโปรแกรมต้องถูกปรับเปลี่ยนเพื่อให้ยอมรับค่าคีย์ที่เป็นลบ (negative key value) เพื่อให้ข้อความ (ที่ถูกเข้ารหัส) "dec" สามารถทำการแปลงกลับเป็นข้อความต้นฉบับคือ "abz" ด้วยค่าคีย์ -3

เพื่อให้ยอมรับค่าคีย์ที่เป็นลบและสามารถทำงานได้ตามฟังก์ชันที่อธิบายข้างต้นโปรแกรมจึงถูกปรับเปลี่ยนดังนี้

- (1) ขยายขอบเขตของค่าคีย์ออกเป็น  $-25 \leq \text{key value} \leq 25$  รวมทั้ง 0 ซึ่งข้อความที่ถูกเข้ารหัสจะเหมือนกับต้นฉบับเมื่อค่าคีย์ถูกกำหนดเป็น 0
- (2) แทนที่บรรทัดที่ถูกทำเครื่องหมาย `/* a */` ในฟังก์ชัน main:

```
} while (  E );          /* a */
```

ด้วยสามบรรทัดต่อไปนี้:

```
} while (  F );  
if (key < 0)  
     G ;
```

ในที่นี้ ผลลัพธ์ของ  $k \% d (d > 0)$  มีสัญลักษณ์เดียวกันกับ  $k$  ตัวอย่างเช่น  $-12 \% 5$  คือ -2

กลุ่มคำตอบสำหรับ F

- |                                   |                                   |
|-----------------------------------|-----------------------------------|
| a) <code>key != (key % 25)</code> | b) <code>key != (key % 26)</code> |
| c) <code>key == (key % 25)</code> | d) <code>key == (key % 26)</code> |

กลุ่มคำตอบสำหรับ G

- |                                |                                |
|--------------------------------|--------------------------------|
| a) <code>key = key % 25</code> | b) <code>key = key % 26</code> |
| c) <code>key = 25 - key</code> | d) <code>key = 26 - key</code> |
| e) <code>key += 25</code>      | f) <code>key += 26</code>      |

**Q8. อ่านคำอธิบายโปรแกรมภาษาจาวาและตัวโปรแกรม จากนั้นให้ตอบคำถามย่อย 1 ถึง 3**

[คำอธิบายโปรแกรม]

โปรแกรมต่อไปนี้ใช้ดำเนินการทางคณิตศาสตร์กับตัวเลขทศนิยมแบบดับเบิล (double numbers) และจำนวนตรรกยะ (rational numbers) ซึ่งประกอบด้วยหนึ่งอินเทอร์เฟซ และคลาสจำนวน 3 คลาส

โปรแกรม 1 คืออินเทอร์เฟซ ArithmeticOperation ซึ่งประกอบด้วยตัวดำเนินการทางคณิตศาสตร์สองกระบวนการคือการบวกและการลบของตัวเลข 2 จำนวน ทุกคลาสที่เกิดจากการใช้อินเทอร์เฟซนี้ จะไม่สามารถเปลี่ยนแปลงได้ (immutable) ซึ่งจะช่วยให้ทำงานบางด้านได้อย่างเหมาะสม

- (1) พารามิเตอร์ชนิด T คือ ชนิดของอาร์กิวเมนต์และค่าที่ส่งคืนของเมธอด
- (2) เมธอด add จะคืนค่าผลลัพธ์ของการบวก: ค่านี้ถูกแสดงด้วย this + ค่าที่ถูกกำหนดโดยอาร์กิวเมนต์ b
- (3) เมธอด sub จะคืนค่าผลลัพธ์ของการลบ: ค่านี้ถูกแสดงด้วย this - ค่าที่ถูกกำหนดโดยอาร์กิวเมนต์ b

โปรแกรม 2 คือคลาส abstract ชื่อ ComparableNumber ซึ่งมีส่วนช่วยในการ implement สองอินเทอร์เฟซ โดยอินเทอร์เฟซแรกคือ java.lang.Comparable สำหรับการจัดเรียงอินสแตนซ์ทั้งหมดของคลาสนี้ และอีกอินเทอร์เฟซคือ ArithmeticOperation สำหรับดำเนินการทางคณิตศาสตร์ คลาส ComparableNumber ประกอบด้วยเมธอดต่อไปนี้

- (1) เมธอด Abstract ชื่อ doubleValue จะคืนค่าตัวเลขทศนิยมแบบดับเบิล (double value) ของคลาส ComparableNumber นี้ ตัวอย่างเช่น ค่าตัวเลขทศนิยมของ  $1/5$  คือ 0.2
- (2) เมธอด compareTo จะเปรียบเทียบสองค่าระหว่าง ComparableNumbers a (this) และ b ที่ระบุในอาร์กิวเมนต์ ถ้า a มีค่ามากกว่า b เมธอดนี้จะคืนค่าที่มากกว่า 0, ถ้า a มีค่าน้อยกว่า b จะคืนค่าที่น้อยกว่า 0 มิฉะนั้นจะคืนค่า 0

โปรแกรม 3 คือคลาส DoubleNumber ซึ่งแทนตัวเลขที่มีจุดทศนิยมชนิด double (ชนิดของข้อมูลพื้นฐานชนิดหนึ่งในภาษาจาวา) ซึ่งคลาสนี้ extends มาจากคลาส ComparableNumber

โปรแกรม 4 คือคลาส RationalNumber ซึ่งแทนตัวเลขจำนวนตรรกยะ คลาสนี้ extends มาจากคลาส ComparableNumber โดยที่จำนวนตรรกยะ คือจำนวนที่สามารถเขียนในรูปเศษส่วนได้ ( $p | q$ ) โดยทั้งเศษ (p) และส่วน (q) เป็นจำนวนเต็ม และส่วน (q) ต้องไม่เป็นศูนย์ ( $q \neq 0$ ) และคลาส RationalNumber ประกอบด้วยคอนสตรัคเตอร์และเมธอด ดังนี้

- (1) คอนสตรัคเตอร์ RationalNumber จะสร้างและกำหนดค่าตั้งต้นของจำนวนตรรกยะด้วยตัวเลขที่เป็นเศษ (numerator) และ ตัวเลขที่เป็นส่วน (denominator) ตามที่ระบุ ถ้าตัวเลขที่เป็นเศษ มีค่าเท่ากับศูนย์แล้ว ArithmeticException จะถูกโยน (thrown)
- (2) เมธอด greatestCommonDivisor จะคืนค่าตัวหารร่วมมากของจำนวนเต็มสองจำนวนที่คำนวณด้วยขั้นตอนวิธีแบบยุคลิด (Euclidean algorithm)

- (3) เมธอด toString จะคืนค่าสายอักขร (string) ของจำนวนตรรกยะนี้
- (4) เมธอด add จะคืนค่าผลลัพธ์ของกระบวนการทางคณิตศาสตร์: (this + b)
- (5) เมธอด sub จะคืนค่าผลลัพธ์ของกระบวนการทางคณิตศาสตร์: (this + (-b))

[โปรแกรม 1]

```
public interface ArithmeticOperation<T> {  
    public T add(T b);  
    public T sub(T b);  
}
```

[โปรแกรม 2]

```
public abstract class ComparableNumber<T> implements A {  
    public abstract double doublevalue();  
  
    @Override  
    public int compareTo(ComparableNumber<?> o) {  
        return Double.compare(doublevalue(), o.doublevalue());  
    }  
}
```

[โปรแกรม 3]

```
public class DoubleNumber extends ComparableNumber<DoubleNumber> {  
    private final double value;  
  
    public DoubleNumber(double value) {  
        this.value = value;  
    }  
  
    @Override  
    public double doublevalue() {  
        return value;  
    }  
  
    @Override  
    public DoubleNumber add(DoubleNumber b) {  
        return new DoubleNumber(value + b.value);  
    }  
  
    @Override  
    public DoubleNumber sub(DoubleNumber b) {  
        return new DoubleNumber(value - b.value);  
    }  
}
```

```

@Override
public String toString() {
    return String.valueOf(value);
}
}

```

[โปรแกรม 4]

```

public class RationalNumber extends B {
    private final int numerator;
    private final int denominator;

    public RationalNumber(int numerator, int denominator) {
        if (denominator == 0) {
            throw new ArithmeticException("denominator is zero");
        }
        int gcd = greatestCommonDivisor(Math.abs((long)numerator),
                                         Math.abs((long)denominator));

        denominator /= gcd;
        numerator /= gcd;
        if (denominator < 0) {
            denominator = -denominator;
            numerator = -numerator;
        }
        this.numerator = numerator;
        this.denominator = denominator;
    }

    private int greatestCommonDivisor(long a, long b) {
        long m = a;
        long n = b;

        while (n != 0) {
            long temp = C;
            m = n;
            n = temp;
        }
        return (int) m;
    }
}

```

```

@Override
public String toString() {
    StringBuilder sb = new StringBuilder().append(numerator);
    if (denominator != 1) {
        sb.append("/").append(denominator);
    }
    return sb.toString();
}

@Override
public RationalNumber add(RationalNumber b) {
    // optimization for special cases
    if (numerator == 0) {
        return b;
    }
    if (b.numerator == 0) {
        return this;
    }
    // add cross-product terms for numerator
    return new RationalNumber((numerator * b.denominator)
                               + (b.numerator * denominator),
                               denominator * b.denominator);
}

@Override
public RationalNumber sub(RationalNumber b) {
    return add(new RationalNumber());
}

@Override
public double doubleValue() {
    return (double) ;
}
}

```

### คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องที่จะเติมลงในช่องว่าง  แต่ละช่องในโปรแกรมด้านบน

กลุ่มคำตอบสำหรับ A และ B

- a) ArithmeticOperation
- b) ArithmeticOperation, Comparable
- c) ArithmeticOperation, Comparable<ComparableNumber<?>>
- d) ArithmeticOperation<T>
- e) ArithmeticOperation<T>, Comparable
- f) ArithmeticOperation<T>, Comparable<ComparableNumber<?>>
- g) Comparable
- h) Comparable<ComparableNumber<?>>
- i) ComparableNumber
- j) ComparableNumber<RationalNumber>

กลุ่มคำตอบสำหรับ C

- a)  $(m + n - 1) / m$
- b)  $(m + n - 1) / n$
- c)  $m \% n$
- d)  $m / n$
- e)  $n \% m$
- f)  $n / m$

กลุ่มคำตอบสำหรับ D

- a) -b.denominator, b.numerator
- b) -b.numerator, b.denominator
- c) -denominator, numerator
- d) -numerator, denominator
- e) b.denominator, b.numerator
- f) b.numerator, b.denominator
- g) denominator, numerator
- h) numerator, denominator

กลุ่มคำตอบสำหรับ E

- a) (denominator \* numerator)
- b) (numerator / denominator)
- c) denominator \* numerator
- d) denominator / numerator
- e) numerator \* denominator
- f) numerator / denominator

## คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องที่จะเติมลงในช่องว่าง  แต่ละช่องในโปรแกรม 5

โปรแกรม 5 คือคลาส TestNumbers สำหรับทดสอบอินเทอร์เฟซและคลาสต่าง ๆ ข้างต้น เมื่อเรียกใช้งานเมธอด main ของคลาสนี้ ก็จะแสดงรายการของ number ทั้งหมดโดยเรียงลำดับจากน้อยไปมาก (ascending order)

[โปรแกรม 5]

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class TestNumbers {
    public static void main(String[] args) {
        List<ComparableNumber<?>> numbers = new ArrayList<>();
        RationalNumber a = new RationalNumber(1, 2);
        RationalNumber b = new RationalNumber(3, 8);
        numbers.add(a);
        numbers.add(b);
        numbers.add(a.add(b));
        numbers.add(a.sub(b));
        DoubleNumber c = new DoubleNumber(0.1);
        numbers.add(c);
        Collections.sort(numbers);
        for (  F  ) {
            System.out.println(number);
        }
        /* a */
    }
}
```

ผลลัพธ์ต่อไปนี้จะเกิดขึ้นเมื่อเมธอด main ของคลาส TestNumbers ถูกดำเนินการ

```
0.1
1/8
3/8
1/2
7/8
```



กลุ่มคำตอบสำหรับ F

- a) ComparableNumber<?> number : numbers
- b) ComparableNumber<DoubleNumber> number : numbers
- c) ComparableNumber<RationalNumber> number : numbers
- d) DoubleNumber number : numbers
- e) int i = 0; i < numbers.size(); i++
- f) int i = numbers.size() - 1; i >= 0; i--
- g) RationalNumber number : numbers

### คำถามย่อย 3

การแทนที่บรรทัด /\*  $\alpha$  \*/ ในโปรแกรม 5 ด้วยบรรทัดที่ปรากฏนี้จะทำให้เกิดข้อผิดพลาด

```
System.out.println(a.add(c));
```

ให้เลือกคำอธิบายที่ถูกต้องสำหรับข้อผิดพลาดที่เกิดขึ้นจากการแทนที่บรรทัดดังกล่าวจากกลุ่มคำตอบด้านล่างนี้

กลุ่มคำตอบ

- a) ClassCastException จะถูกโยนเมื่อรันโปรแกรม
- b) จะเกิดความผิดพลาดในการคอมไพล์ขึ้นขณะคอมไพล์โปรแกรม
- c) NullPointerException จะถูกโยนเมื่อรันโปรแกรม
- d) ArithmeticException จะถูกโยนเมื่อรันโปรแกรม
- e) IllegalArgumentException จะถูกโยนเมื่อรันโปรแกรม