October 2019

Fundamental IT Engineer Examination (Afternoon)

ให้ทำข้อสอบตามรายละเอียดต่อไปนี้

หมายเลขคำถาม	**Q1 – Q6**	**Q7 , Q8**
การเลือกคำถาม	**ทำทุกข้อ**	**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 Q7 และ Q8 ให้เลือก ⓢ ที่คุณเลือกที่จะตอบในช่อง "Selection Column" บนกระดาษคำตอบของคุณ

(4) **คำตอบ**
ให้ทำเครื่องหมายตรงคำตอบที่เลือกตามตัวอย่างที่แสดงอยู่ด้านล่าง

[คำถามตัวอย่าง]
ข้อสอบวัดระดับ Fundamental IT Engineer Examination รอบฤดูใบไม้ร่วงจัดในเดือนใด

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

เนื่องจากคำตอบที่ถูกเป็น "b) ตุลาคม" ดังนั้นให้ทำเครื่องหมายดังข้างล่างนี้

[ตัวอย่างคำตอบ]

Sample	ⓐ	●	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ

ห้ามเปิดดูข้อสอบก่อนได้รับอนุญาต
ข้อสงสัยที่เกี่ยวข้องกับคำถามในข้อสอบอาจจะไม่ถูกตอบ

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.

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
type: *var1*, …, *array1*[], …		Declares variables *var1*, … , and/or arrays *array1*[], … , by data *type* such as INT and CHAR.
FUNCTION: *function*(*type*: *arg1*, …)		Declares a *function* and its arguments *arg1*, … .
/* comment */		Describes a comment.
Process	*variable* ← *expression* ;	Assigns the value of the *expression* to the *variable*.
	function (*arg1*, …) ;	Calls the *function* by passing / receiving the arguments *arg1*, … .

	IF (*condition*) {
	process1
	}
	ELSE {
	process2
	}
		Indicates the selection process.
		If the *condition* is true, then *process1* is executed.
		If the *condition* is false, then *process2* is executed, when the optional ELSE clause is present.

	WHILE (*condition*) {
	process
	}
		Indicates the "WHILE" iteration process.
		While the *condition* is true, the *process* is executed repeatedly.

	DO { *process* } WHILE (*condition*);	Indicates the "DO - WHILE" iteration process. The *process* is executed once, and then while the *condition* is true, the *process* is executed repeatedly.

	FOR (*init* ; *condition* ; *incr*) {
	process
	}
		Indicates the "FOR" iteration process.
		While the *condition* is true, the *process* is executed repeatedly.
		At the start of the first iteration, the process *init* is executed before testing the *condition*.
		At the end of each iteration, the process *incr* is executed before testing the *condition*.

[Logical constants]

true, false

[Operators and their precedence]

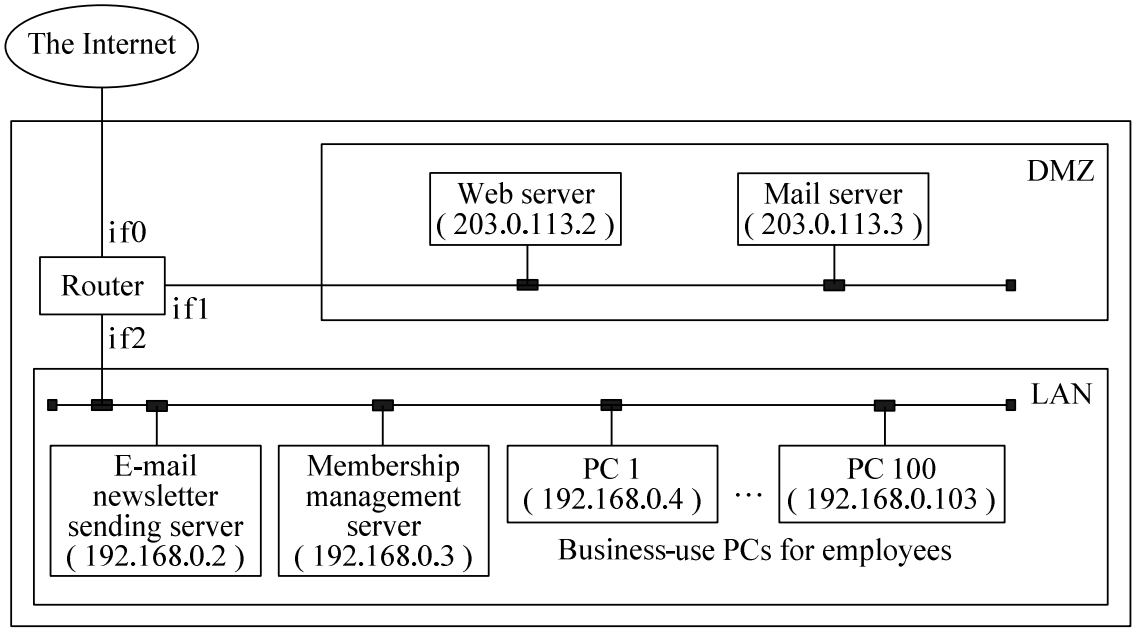
Type of operation	Unary	Arithmetic		Relational	Logical	
Operators	+, −, not	×, ÷, %	+, −	>, <, ≥, ≤, =, ≠	and	or
Precedence	High ←					→ Low

Note: With division of integers, an integer quotient is returned as a result.
The "%" operator indicates a remainder operation.

คำถาม **Q1** ถึง **Q6** เป็นคำถาม**บังคับ** ให้ทำทุกข้อ

Q1. อ่านคำอธิบายเกี่ยวกับความมั่นคงของเครือข่ายต่อไปนี้ แล้วตอบคำถามย่อย 1 ถึง 4

บริษัท A ได้สร้างและใช้งานเว็บไซต์แนะนำสินค้าโดยติดตั้งไว้บนเว็บเซอร์ฟเวอร์ภายในบริษัทและได้เปิดรับสมัครสมาชิกบนเว็บไซต์นี้ โดยจะสร้างและจัดส่งอีเมลจดหมายข่าวให้สมาชิกที่ลงทะเบียนไว้ โครงสร้างเครือข่ายของบริษัท A ถูกแสดงไว้ในรูปที่ 1



หมายเหตุ 1:หมายเลขไอพีของแต่ละอุปกรณ์ถูกแสดงอยู่ในวงเล็บ
หมายเหตุ 2:if0 if1 และ if2 คืออินเทอร์เฟสเครือข่ายของเราท์เตอร์

รูปที่ 1 โครงสร้างเครือข่ายของบริษัท A

การลงทะเบียนสมาชิกมีลำดับขั้นตอนดังต่อไปนี้:

(1) ผู้สนใจเป็นสมาชิกเข้าไปยังหน้าเว็บแอปพลิเคชันของระบบสมาชิกผ่านทางอินเทอร์เน็ตโดยใช้ HTTP บน TLS (ซึ่งต่อจากนี้จะเรียกว่า HTTPS) จากนั้นจึงป้อนที่อยู่อีเมล โดยหน้าเว็บในขั้นตอนนี้ถูกบริหารจัดการโดยเว็บเซอร์ฟเวอร์ (Web server)

(2) เว็บเซอร์ฟเวอร์สร้างหน้าเว็บขึ้นมาโดยเฉพาะสำหรับผู้สมัครแต่ละคน จากนั้นจึงส่งอีเมลที่มี URL ไปยังหน้าที่สร้างขึ้นนั้นให้กับผู้สมัครตามที่อยู่อีเมลที่ได้ป้อนไว้

(3) ผู้สมัครเข้าไปยังหน้าเว็บที่ถูกสร้างขึ้นสำหรับแต่ละบุคคลตามที่ได้ส่ง URL ให้ทางอีเมลในขั้นตอนที่ (2) โดยใช้ HTTPS จากนั้นจึงกรอกข้อมูลต่าง ๆ สำหรับการสมัครเป็นสมาชิก เช่น ชื่อและอาชีพ (นอกเหนือจากที่อยู่อีเมล)

(4) เว็บเซอร์ฟเวอร์บันทึกที่อยู่อีเมลจาก (1) และข้อมูลสมาชิกจาก (3) ลงในฐานข้อมูลที่รันอยู่บนเซอร์ฟเวอร์สำหรับจัดการข้อมูลสมาชิก (membership management server) ซึ่งจากนี้ไปจะเรียกว่าฐานข้อมูลสมาชิก

(5) เว็บเซอร์ฟเวอร์ส่งอีเมลไปยังที่อยู่อีเมลที่ระบุในขั้นตอนที่ (1) เพื่อแจ้งผู้สมัครว่าการลงทะเบียนเสร็จเรียบร้อยแล้ว

การสร้างและส่งอีเมลจดหมายข่าวมีลำดับขั้นตอนดังต่อไปนี้:

(1) หน้าเว็บสำหรับการกรอกข้อมูลจดหมายข่าวอยู่บนเซอร์ฟเวอร์สำหรับส่งอีเมลจดหมายข่าว (e-mail newsletter sending server)

(2) ผู้รับผิดชอบอีเมลจดหมายข่าวเปิดหน้าเว็บสำหรับกรอกข้อมูลจดหมายข่าวในเว็บบราวเซอร์ที่รันอยู่บนเครื่องพีซีของบริษัท (business-use PC) โดยใช้ HTTP จากนั้นจึงป้อนข้อความของอีเมลจดหมายข่าว

(3) เซอร์ฟเวอร์สำหรับส่งอีเมลจดหมายข่าวได้รับที่อยู่อีเมลของสมาชิกทั้งหมดมาจากฐานข้อมูลสมาชิก จากนั้นจึงส่งอีเมลจดหมายข่าวไปยังที่อยู่อีเมลเหล่านั้น

อีเมลถูกส่งผ่านบริการส่งเมล (mail transmission service) ที่รันอยู่บนเมลเซอร์ฟเวอร์ (mail server) ในการตอบคำถามนี้ ไม่ต้องคำนึงถึงกระบวนการแปลงชื่อต่าง ๆ เช่น URL หรือที่อยู่อีเมล

คำถามย่อย 1

บนหน้าเว็บของระบบแอปพลิเคชันสมาชิกที่ผู้สมัครเข้ามาเป็นหน้าแรกเพื่อลงทะเบียนเป็นสมาชิกนั้นได้ใช้วิธีการกำหนดให้ผู้สมัครเพียงป้อนที่อยู่อีเมลอย่างเดียวเท่านั้น แล้วจากนั้นจึงส่งช่องทางกรอกข้อมูลสมาชิกให้แยกต่างหาก จากกลุ่มคำตอบด้านล่าง ให้เลือกวัตถุประสงค์หลักของการใช้กระบวนการแบบสองขั้นตอนเช่นนี้

กลุ่มคำตอบ

a) เพื่อเข้ารหัสลับการสื่อสารและป้องกันการรั่วไหลของข้อมูลผู้สมัครสมาชิกไปยังบุคคลที่สาม
b) เพื่อป้องกันไม่ให้ผู้สมัครสมาชิกเข้าถึงฐานข้อมูลสมาชิกได้
c) เพื่อป้องกันการสมัครโดยใช้ที่อยู่อีเมลของบุคคลอื่นหรือการใช้ที่อยู่อีเมลที่ไม่ถูกต้อง
d) เพื่อป้องกันการบันทึกข้อมูลต่าง ๆ (นอกเหนือจากที่อยู่อีเมล) อย่างไม่ถูกต้อง

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่อง ในรูปที่ 2

เราท์เตอร์ตัวหนึ่งมีคุณสมบัติของไฟร์วอลล์แบบกรองแพคเก็ตที่ปรับเปลี่ยนได้ (dynamic packet filter) และอนุญาตให้ผ่านได้เฉพาะแพคเก็ตที่ตรงตามค่าที่กำหนดไว้ในไฟร์วอลล์เท่านั้น การกำหนดค่านี้เรียงลำดับโดยเริ่มจากต้นทาง (source) ปลายทาง (destination) และพอร์ตที่ใช้สื่อสาร (communication port) โดยคั่นด้วยจุลภาค (,) ค่าที่กำหนดไว้นี้อนุญาตให้แพคเก็ตจากต้นทางที่กำหนดสามารถผ่านไปยังพอร์ตที่ใช้สื่อสาร ณ ปลายทางได้ และการกำหนดค่านี้ยังอนุญาตให้แพคเก็ตที่ตอบสนองจากแพคเก็ตที่ได้รับอนุญาตให้ผ่านกลับมาได้ด้วย
ต้นทางและปลายทางถูกกำหนดได้โดยใช้ที่อยู่ไอพี (IP address) หรืออินเทอร์เฟสเครือข่าย (network interface) หากกำหนดด้วยที่อยู่ไอพี แพคเก็ตที่มาจากที่อยู่ไอพีต้นทาง (เมื่อกำหนดไว้เป็นต้นทาง) หรือที่จะไปยังที่อยู่ไอพี (เมื่อกำหนดไว้เป็นปลายทาง) จะได้รับอนุญาตให้ผ่านได้ แต่หากกำหนดด้วยอินเทอร์เฟสเครือข่าย แพคเก็ตที่มาจาก (เมื่อกำหนดไว้เป็นต้นทาง) หรือไปยัง (เมื่อกำหนดไว้เป็นปลายทาง) อินเทอร์เฟสเครือข่ายนั้นจะได้รับอนุญาตให้ผ่านได้
พอร์ตที่ใช้สื่อสารถูกกำหนดด้วยหมายเลขพอร์ตที่แต่ละบริการ (service) รอรับ (listen) แพคเก็ตอยู่ โปรโตคอล (protocol) และพอร์ตที่รอรับการเชื่อมต่อ (listening port number) ที่ถูกใช้กับบริการต่าง ๆ ที่รันอยู่บนเซอร์ฟเวอร์แต่ละเครื่องภายในบริษัท *A* ถูกแสดงอยู่ในตารางที่ 1

ตารางที่ 1 โปรโตคอลและพอร์ตที่รอรับการเชื่อมต่อจำแนกตามบริการ

บริการ (service)	โปรโตคอล (protocol)	พอร์ตที่รอรับการเชื่อมต่อ (listening port number)
Web	HTTP	80
	HTTPS	443
Member information database	Proprietary protocol	4194
E-mail (receiving)	POP3	110
E-mail (transmission)	SMTP	25

การกำหนดค่าของเราท์เตอร์ในปัจจุบันถูกแสดงในรูปที่ 2
การกำหนดค่าในบรรทัดที่ 1 อนุญาตให้เข้าถึงเว็บเซอร์ฟเวอร์จากอินเทอร์เน็ตโดยใช้ HTTP ได้ การกำหนดค่าในบรรทัดที่ 2 อนุญาตให้อีเมลที่มาจากอินเทอร์เน็ตส่งผ่าน (transmission) เข้ามายังเมลเซอร์ฟเวอร์ได้

```
if0,203.0.113.2,80
if0,203.0.113.3,25
203.0.113.3,if0,25
if2,203.0.113.2,80
if2,203.0.113.2,443
if2,203.0.113.3,25
if2,203.0.113.3,110
if0,[   A   ]
203.0.113.2,[   B   ]
```

รูปที่ 2 การกำหนดค่าเราท์เตอร์ปัจจุบัน

กลุ่มคำตอบสำหรับ A และ B

a) 192.168.0.2,443
b) 192.168.0.2,4194
c) 192.168.0.3,4194
d) 203.0.113.2,443
e) 203.0.113.2,4194
f) if0,443

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] ในคำอธิบายด้านล่าง

บริษัท *A* เอาท์ซอร์สการบำรุงรักษาเว็บเซอร์ฟเวอร์ไปยังผู้ให้บริการภายนอก โดยผู้ให้บริการได้รับอนุญาตให้ใช้งานเว็บเซอร์ฟเวอร์ของบริษัท *A* ผ่านอินเทอร์เน็ตมาจากเครื่องพีซีที่กำหนด ด้วยเหตุนี้บริษัท *A* จึงรันบริการ SSH ไว้บนพอร์ต 22 ของเว็บเซอร์ฟเวอร์ จากนั้นจึงได้เพิ่มบรรทัด "[C]" ไปยังการตั้งค่าของเราท์เตอร์
เมื่อแพคเก็ตที่ถูกส่งจากพีซีเครื่องนี้มาถึงเราท์เตอร์ หมายเลขไอพีของผู้ส่งแพคเก็ตมาคือ 198.51.100.2

กลุ่มคำตอบสำหรับ C

a) 198.51.100.2,203.0.113.2,22
b) 198.51.100.2,if0,22
c) 203.0.113.2,198.51.100.2,22
d) if0,198.51.100.2,22

คำถามย่อย 4

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] ในคำอธิบายด้านล่าง

บริการ SSH มีวิธีพิสูจน์ตัวจริงของไคลเอนต์ได้หลายวิธี เช่นวิธีการพิสูจน์ตัวจริงโดยใช้รหัสผ่าน และวิธีการพิสูจน์ตัวจริงโดยใช้กุญแจสาธารณะ (public key) ในที่นี้ บริษัท A เลือกใช้วิธีพิสูจน์ตัวจริงด้วยกุญแจสาธารณะ

วิธีการพิสูจน์ตัวจริงด้วยกุญแจสาธารณะตรวจสอบลายเซ็น (signature) ที่ถูกสร้างขึ้นด้วยกุญแจส่วนตัว (private key) โดยใช้ร่วมกันกับกุญแจสาธารณะที่เป็นคู่กัน และดำเนินการพิสูจน์ตัวจริงของไคลเอนต์ตามขั้นตอนต่อไปนี้:

(1) ไคลเอนต์ส่งลายเซ็นที่ถูกสร้างขึ้นด้วยกุญแจส่วนตัวไปพร้อมกันกับกุญแจสาธารณะที่เป็นคู่ของกุญแจส่วนตัวนั้นไปยังเซอร์ฟเวอร์

(2) เซอร์ฟเวอร์ยืนยันว่ากุญแจสาธารณะใน (1) นั้นเคยลงทะเบียนไว้กับเซอร์ฟเวอร์แล้ว จากนั้นจึงใช้กุญแจสาธารณะดังกล่าวตรวจสอบลายเซ็นจาก (1)

(3) การตรวจสอบที่ประสบความสำเร็จหมายความว่าไคลเอนต์มีกุญแจส่วนตัวที่เป็นคู่ของกุญแจสาธารณะที่เคยลงทะเบียนไว้กับเซอร์ฟเวอร์จริง ดังนั้น จึงพิสูจน์ได้ว่าไคลเอนต์ดังกล่าวเป็นตัวจริง

ด้วยวิธีนี้ เมื่อไคลเอนต์ใช้บริการ SSH บนเซอร์ฟเวอร์ด้วยการพิสูจน์ตัวจริงโดยกุญแจสาธารณะ จึงไม่มีความจำเป็นต้องส่งข้อมูลอย่างรหัสผ่านและ [D] ผ่านเครือข่าย

กลุ่มคำตอบสำหรับ D

a) กุญแจส่วนตัว

b) กุญแจส่วนตัวและกุญแจสาธารณะ

c) กุญแจสาธารณะ

Q2. อ่านคำอธิบายเกี่ยวกับนิพจน์ตรรกะ (logical expressions) จากนั้นให้ตอบคำถามย่อย

ในระบบที่รับค่าจำนวนเต็มขนาด 4 บิตที่มีค่าได้ตั้งแต่ 0000(0) ถึง 1111(15) จากสายอินพุต I3 I2 I1 และ I0 แล้วจึงตรวจสอบว่าจำนวนดังกล่าวตรงกับชนิดตัวเลขแบบใดในตารางที่ 1 จากนั้นระบบจะส่งผลลัพธ์ออกทางสายเอาต์พุต Sq Pr และ Mp ดังที่แสดงในตารางค่าความจริงในตารางที่ 2

ตารางที่ 1 รูปแบบตัวเลขสามชนิดที่ตรวจสอบได้จากระบบ

ชนิด (สายเอาต์พุต)	คำอธิบาย	ตัวเลขที่ตรงเงื่อนไข
ค่ายกกำลังสองของตัวเลข (Sq)	ตัวเลขจำนวนเต็มที่มีค่าเป็นกำลังสองของอีกตัวเลขหนึ่ง	0000(0), 0001(1), 0100(4), 1001(9)
จำนวนเฉพาะ (Pr)	จำนวนธรรมชาติ (natural number) หรือจำนวนนับที่มีค่ามากกว่า 1 ที่ไม่สามารถสร้างขึ้นได้จากการนำจำนวนธรรมชาติสองจำนวนที่มีค่าน้อยกว่ามาคูณกัน	0010(2), 0011(3), 0101(5), 0111(7), 1011(11), 1101(13)
ผลคูณของ 3 (Mp)	ตัวเลขจำนวนเต็มที่ได้จากการคูณด้วย 3	0000(0), 0011(3), 0110(6), 1001(9), 1100(12), 1111(15)

ตารางที่ 2 ตารางค่าความจริง

	Inputs				Outputs		
	I3	I2	I1	I0	Sq	Pr	Mp
0	0	0	0	0	1	0	1
1	0	0	0	1	1	0	0
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	1
4	0	1	0	0	1	0	0
5	0	1	0	1	0	1	0
6	0	1	1	0	0	0	1
7	0	1	1	1	0	1	0
8	1	0	0	0	0	0	0
9	1	0	0	1	1	0	1
10	1	0	1	0	0	0	0
11	1	0	1	1	0	1	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	1	0
14	1	1	1	0	0	0	0
15	1	1	1	1	0	0	1

ในนิพจน์ที่แสดงด้านล่าง สัญลักษณ์ • + และ $^{-}$ ถูกใช้เพื่อแสดงถึงกระบวนการทางตรรกะ AND OR และ NOT ตามลำดับ

อันดับแรกเป็นการพิจารณานิพจน์ตรรกะที่ตรวจสอบว่าตัวเลขนั้น ๆ มีค่าเป็นกำลังสองของอีกตัวเลขหนึ่งแล้วส่งผลลัพธ์ออกทางสายเอาต์พุต Sq นิพจน์ Sq-1 ที่แสดงอยู่ด้านล่างจะส่งค่า 1 ออกทางสายเอาต์พุต Sq เมื่อตัวเลขเป็น 0000(0) 0001(1) 0100(4) หรือ 1001(9) หากเป็นค่าอื่นให้ส่งค่า 0 แต่ละพจน์ (term) ในนิพจน์ Sq-1 สอดคล้องกับตัวเลขตัวหนึ่ง

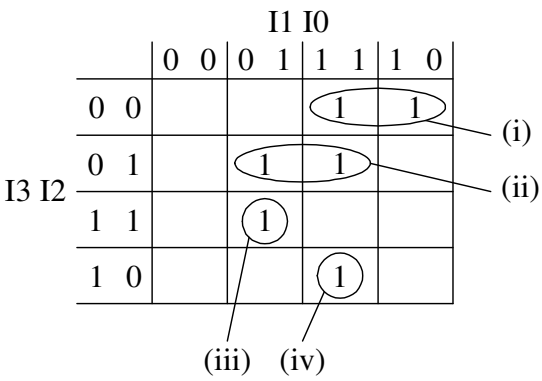
สองพจน์แรกในนิพจน์ Sq-1 ที่สอดคล้องกับตัวเลข 0000(0) และ 0001(1) สามารถรวมกันเป็นพจน์ที่เรียบง่ายกว่าได้ดังแสดงในนิพจน์ Sq-2

นิพจน์ Sq-1: $\overline{I3}\bullet\overline{I2}\bullet\overline{I1}\bullet\overline{I0} + \overline{I3}\bullet\overline{I2}\bullet\overline{I1}\bullet I0 + \overline{I3}\bullet I2\bullet\overline{I1}\bullet\overline{I0} + \boxed{A}$

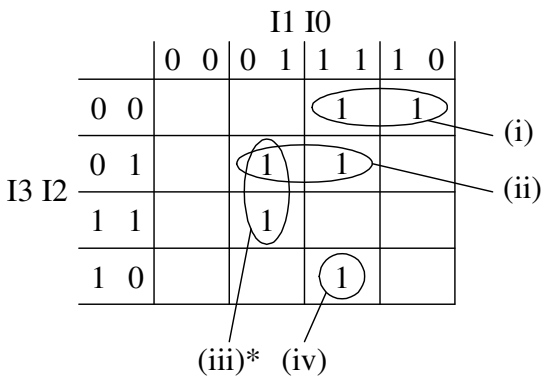
นิพจน์ Sq-2: $\boxed{B} + \overline{I3}\bullet I2\bullet\overline{I1}\bullet\overline{I0} + \boxed{A}$

ต่อมาเป็นการพิจารณานิพจน์ตรรกะที่ตรวจสอบว่าตัวเลขนั้น ๆ เป็นจำนวนเฉพาะ แล้วส่งผลลัพธ์ออกทางสายเอาต์พุต Pr

ตารางค่าความจริงสำหรับ Pr ในตารางที่ 2 ถูกแปลงเป็นผังคาร์โนห์ (Karnaugh map) ซึ่งในกรณีนี้ก็คือเมทริกซ์ขนาด 4 × 4 ดังแสดงในรูปที่ 1 ส่วนของแถวคืออินพุต I3 และ I2 และส่วนของคอลัมน์คืออินพุต I1 และ I0 ค่าที่แสดงในแต่ละเซลล์สอดคล้องกับค่าเอาต์พุต Pr ซึ่งมีค่าเป็น 1 หรือ 0 (เซลล์ที่มีค่าเป็น 0 ถูกละไว้)



รูปที่ 1 ผังคาร์โนห์สำหรับ Pr



รูปที่ 2 ผังคาร์โนห์สำหรับ Pr

ผังคาร์โนห์ถูกใช้เพื่อทำให้นิพจน์ตรรกะเรียบง่ายขึ้น เมื่อมีคู่ของ 1 ที่อยู่ติดกันในแนวตั้งหรือแนวนอน (คือคู่ที่มีวงรีล้อมรอบในรูปที่ 1 และ 2) ค่าเอาต์พุตก็จะสามารถแสดงได้ด้วยพจน์ที่เรียบง่ายขึ้น ดังเช่นการแปลงจากนิพจน์ Sq-1 ไปเป็น Sq-2

จากรูปที่ 1 นิพจน์ Pr-1 ทำให้ได้ผลลัพธ์ที่จะส่งออกไปยังสายเอาต์พุต Pr

นิพจน์ Pr-1: $\boxed{C} + \overline{I3}\bullet I2\bullet I0 + I3\bullet I2\bullet\overline{I1}\bullet I0 + I3\bullet\overline{I2}\bullet I1\bullet I0$

(i) (ii) (iii) (iv)

ในรูปที่ 1 ค่าเอาต์พุต 1 ของ (iii) ถูกประเมินในฐานะค่าเดี่ยวหรือเป็น 1 เพียงตัวเดียว แต่อันที่จริงแล้วค่านี้เป็นส่วนหนึ่งของคู่ของค่า 1 ที่อยู่ติดกันในแนวตั้งหรือ (iii)* ที่แสดงในรูปที่ 2 ดังนั้น พจน์ที่แทน (iii) ในนิพจน์ Pr-1 สามารถแสดงให้เป็นพจน์ที่เรียบง่ายขึ้นหรือ (iii)* ในนิพจน์ Pr-2 ที่แสดงอยู่ด้านล่าง ในกรณีนี้ ค่าเอาต์พุต 1 ที่สอดคล้องกับตัวเลข 0101(5) ถูกนำมาประเมินสองครั้ง แต่ไม่ส่งผลกับผลลัพธ์สุดท้ายแต่อย่างใด

นิพจน์ Pr-2: $\boxed{\quad C \quad} + \overline{I3} \bullet I2 \bullet I0 + \boxed{\quad D \quad} + I3 \bullet \overline{I2} \bullet I1 \bullet I0$

(i) (ii) (iii)* (iv)

ท้ายที่สุด พิจารณานิพจน์ตรรกะที่ตรวจสอบว่าตัวเลขนั้น ๆ เป็นผลคูณของ 3 แล้วส่งผลลัพธ์ออกไปยังสายเอาต์พุต Mp หลังจากสร้างผังคาร์โนห์ของ Mp ดังที่แสดงในรูปที่ 3 ให้เสร็จเรียบร้อยแล้วจะพบว่า $\boxed{\quad E \quad}$ คู่ของ 1 ที่อยู่ติดกัน

I3 I2 \ I1 I0	0 0	0 1	1 1	1 0
0 0	1		1	
0 1				
1 1				
1 0				

หมายเหตุ: ส่วนที่ถูกแรเงาคือส่วนที่ยังไม่แล้วเสร็จ

รูปที่ 3 ผังคาร์โนห์สำหรับ Mp

คำถามย่อย

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง $\boxed{\quad\quad}$ แต่ละช่องในคำอธิบายข้างต้น

กลุ่มคำตอบสำหรับ A

a) $I3 \bullet \overline{I2} \bullet I1 \bullet \overline{I0}$ b) $I3 \bullet \overline{I2} \bullet \overline{I1} \bullet I0$ c) $\overline{I3} \bullet I2 \bullet I1 \bullet \overline{I0}$ d) $\overline{I3} \bullet I2 \bullet \overline{I1} \bullet I0$

กลุ่มคำตอบสำหรับ B ถึง D

a) $I2 \bullet \overline{I1} \bullet I0$ b) $\overline{I2} \bullet I1 \bullet \overline{I0}$ c) $I3 \bullet I2 \bullet I1$ d) $I3 \bullet I2 \bullet \overline{I1}$

e) $I3 \bullet \overline{I2} \bullet I1$ f) $\overline{I3} \bullet I2 \bullet \overline{I1}$ g) $\overline{I3} \bullet \overline{I2} \bullet I1$ h) $\overline{I3} \bullet \overline{I2} \bullet \overline{I1}$

กลุ่มคำตอบสำหรับ E

a) ไม่มี b) มีหนึ่ง c) มีสอง d) มีสาม

Q3. อ่านคำอธิบายของโมเดลความสัมพันธ์เอนทิตี้ (entity relationship model) ต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

ร้านผลไม้ C ขายผลไม้และสินค้าจากผลไม้ เช่น ผลไม้อบแห้งและน้ำผลไม้ โดยที่ร้านนี้ได้ติดตั้งระบบการจัดการการเก็บเงิน (bill management system) ซึ่งเป็นระบบที่จัดการข้อมูลลูกค้าและข้อมูลการเรียกเก็บเงิน

รูปที่ 1 แสดงโครงสร้างตารางฐานข้อมูลที่ใช้ในระบบการจัดการการเก็บเงินและตัวอย่างข้อมูลที่จัดเก็บ โดยรายการต่าง ๆ ที่ขีดเส้นใต้ไว้คือคีย์หลัก (primary key)

Product table

ProductID	Category	ProductName	UnitPrice
FMG005	Fruit	Mango 5 kg pack	10.00
FWM010	Fruit	Watermelon 10 kg pack	25.00
JPA0100	Juice	Pineapple juice 1000 ml	2.00

Customer table

CustomerID	Type	Name	Phone	Address
CH005	Company	East Hotel	223-3333	58 East town, Central city
CR052	Company	West Restaurant	224-4444	20 West town, Central city
P0082	Person	Emily Flippo	225-6789	974 South hill, Central city

Bill table

BillID	Date	CustomerID	BillAmount
08170001	2019-08-17	P0082	4.00
08170002	2019-08-17	CH005	125.00

BillDetail table

BillID	ProductID	Quantity
08170001	JPA0100	2
08170002	FMG005	5
08170002	FWM010	3

รูปที่ 1 โครงสร้างตารางในฐานข้อมูลและตัวอย่างข้อมูลที่จัดเก็บ

ข้อมูลในแต่ละแถวของตารางใบเรียกเก็บเงินหรือบิล (Bill table) สอดคล้องกับใบเรียกเก็บเงินของลูกค้า รายละเอียดของแต่ละบิลจะถูกบันทึกไว้ในตาราง BillDetail และแต่ละแถวของตาราง BillDetail หมายถึงการซื้อหนึ่งครั้ง

คำถามย่อย 1

ทางร้านกำลังวางแผนที่จะนำเสนอเมนูผลไม้คละชนิดหั่นชิ้นเป็นผลิตภัณฑ์ใหม่ เพื่อเป็นการส่งเสริมการขายสินค้านี้กับลูกค้าที่มีแนวโน้มจะซื้อสินค้านี้ ทางร้านได้แยกชื่อ เบอร์โทรศัพท์ และที่อยู่ของลูกค้าที่ซื้อสินค้าประเภท "ผลไม้ (Fruit)" จำนวน 30 ครั้งหรือมากกว่าที่มียอดรวมการสั่งซื้อเป็นมูลค่า $300.00 หรือมากกว่า ในช่วงเวลาสามเดือน ตั้งแต่วันที่ 1 เมษายน 2019 ถึงวันที่ 30 มิถุนายน 2019

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องที่จะเติมลงในช่องว่าง [ ] ในข้อความสั่ง SQL ต่อไปนี้

```
SELECT Customer.Name, Customer.Phone, Customer.Address
   FROM Customer
   WHERE Customer.CustomerID IN
      (SELECT Bill.CustomerID
          FROM Bill, BillDetail, Product
          WHERE Bill.BillID = BillDetail.BillID
                AND Product.ProductID = BillDetail.ProductID
                [   A   ])
```

กลุ่มคำตอบสำหรับ A

a)
```
AND Bill.Date >= '2019-04-01' AND Bill.Date <= '2019-06-30'
AND Product.Category = 'Fruit'
AND Product.UnitPrice * BillDetail.Quantity >= 300.00
GROUP BY Bill.CustomerID
HAVING COUNT(*) >= 30
```

b)
```
AND Bill.Date >= '2019-04-01' AND Bill.Date <= '2019-06-30'
AND Product.Category = 'Fruit'
GROUP BY Bill.CustomerID
HAVING SUM(Product.UnitPrice * BillDetail.Quantity) >= 300.00
   AND COUNT(*) >= 30
```

c)
```
AND Bill.Date >= '2019-04-01' AND Bill.Date <= '2019-06-30'
GROUP BY Bill.CustomerID
HAVING Product.Category = 'Fruit'
   AND SUM(Product.UnitPrice * BillDetail.Quantity) >= 300.00
   AND COUNT(*) >= 30
```

d)
```
GROUP BY Bill.CustomerID
HAVING Bill.Date >= '2019-04-01' AND Bill.Date <= '2019-06-30'
   AND Product.Category = 'Fruit'
   AND SUM(Bill.BillAmount) >= 300.00
   AND COUNT(*) >= 30
```

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่เหมาะสมที่จะเติมในช่องว่าง [ ] แต่ละช่องในคำอธิบายและข้อความสั่ง SQL ต่อไปนี้

ทางร้านต้องการที่จะสามารถเปลี่ยนราคาสินค้าต่อหน่วยในในตาราง Product ได้หลายครั้ง ดังนั้น ทางร้านจึงตัดสินใจที่จะเพิ่มราคาต่อหน่วยที่ขาย (sold unit price) เป็นฟิลด์หนึ่งในตาราง BillDetail เพื่อให้สามารถระบุราคาขายต่อหน่วยในเวลาที่ทำการเรียกเก็บเงินได้ รูปที่ 2 แสดงโครงสร้างของตาราง BillDetail ที่ได้รับการแก้ไขแล้ว ราคาต่อหน่วยในตาราง Product จึงสามารถเปลี่ยนแปลงได้ตามต้องการก่อนเวลาทำการขายในวันนั้น ๆ

BillDetail table

BillID	ProductID	Quantity	SoldUnitPrice

รูปที่ 2 โครงสร้างตารางของตาราง BillDetail ที่ได้รับการแก้ไขแล้ว

ด้วยการเพิ่มราคาต่อหน่วยที่ขาย (sold unit price) เป็นรายการในตาราง BillDetail ทางร้านจึงสามารถหาข้อมูล เช่น [B] ได้

ในการติดตามสถานการณ์การขายหลังจากการแก้ไขโครงสร้างตารางแล้ว ทางร้านได้สร้างข้อความสั่ง SQL ดังต่อไปนี้เพื่อแสดงผลลัพธ์จำนวนเงินรวมในช่วง 3 เดือน ตั้งแต่ 1 กรกฎาคม 2019 ถึง 30 กันยายน 2019 โดยแบ่งตามหมวดหมู่ในตารางสินค้า (Product table) สำหรับสินค้าแต่ละชนิดที่แสดงอยู่ในตารางลูกค้า (Customer table)

```
SELECT Type, Category, [ C ] AS TotalBillAmount
   FROM (SELECT Customer.Type, Product.Category, [ D ]
            FROM Customer, Bill, BillDetail, Product
            WHERE Customer.CustomerID = Bill.CustomerID
                  AND Bill.BillID = BillDetail.BillID
                  AND BillDetail.ProductID = Product.ProductID
                  AND Bill.Date >= '2019-07-01'
                  AND Bill.Date <= '2019-09-30'
         ) BILL3Q
   GROUP BY [ E ]
ORDER BY Type ASC, TotalBillAmount DESC
```

กลุ่มคำตอบสำหรับ B

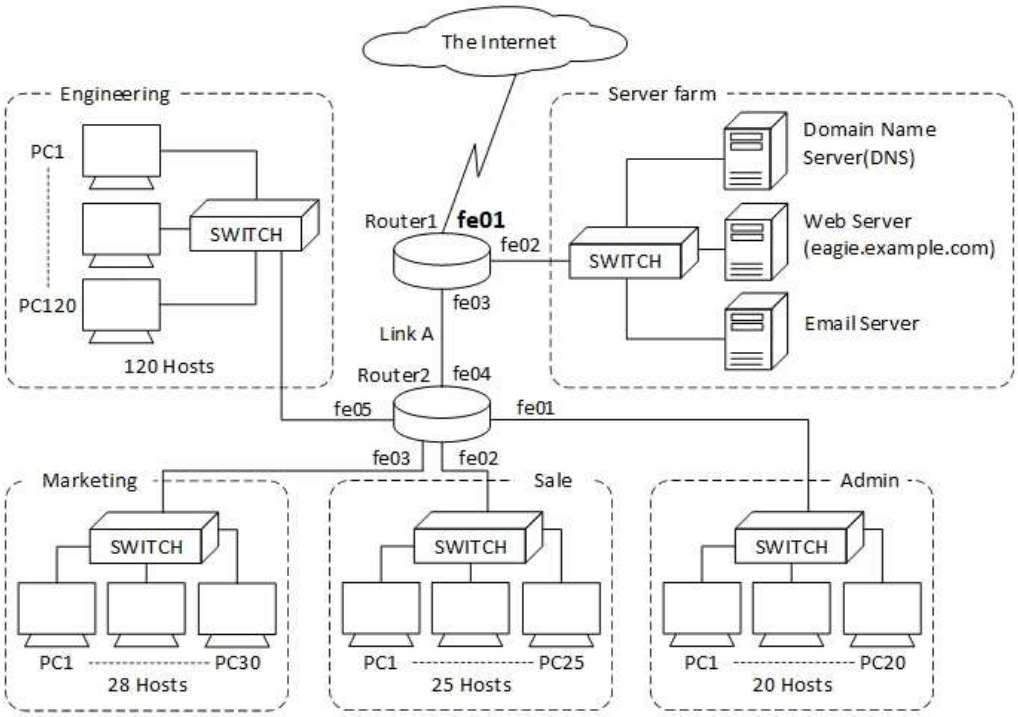
a) ความแตกต่างระหว่างราคาต่อหน่วยที่ลูกค้ารายหนึ่งซื้อสินค้าชิ้นหนึ่งในเวลาหนึ่ง ๆ กับราคาต่อหน่วยที่มีการเปลี่ยนแปลงทันทีหลังจากนั้น
b) การเปลี่ยนแปลงของราคาขายต่อหน่วยของสินค้าชิ้นหนึ่งที่ถูกซื้อไป
c) การเปลี่ยนแปลงของราคาต่อหน่วยของสินค้าทั้งหมด
d) ราคาต่อหน่วยของวันก่อนหน้าวันที่มีการเปลี่ยนแปลงราคาต่อหน่วยครั้งล่าสุดสำหรับสินค้าทั้งหมด

กลุ่มคำตอบสำหรับ C ถึง E

a) `Bill.BillAmount`
b) `BillDetail.SoldUnitPrice, BillDetail.Quantity`
c) `Product.UnitPrice, BillDetail.Quantity`
d) `SUM(BillAmount)`
e) `SUM(SoldUnitPrice * Quantity)`
f) `SUM(UnitPrice * Quantity)`
g) `Type, Category`
h) `Type, Category, TotalBillAmount`

Q4. อ่านคำอธิบายเกี่ยวกับการติดตั้งเครือข่ายสำหรับบริษัทแห่งหนึ่ง จากนั้นให้ตอบคำถามย่อย 1 ถึง 3

บริษัท D ออกแบบโครงสร้างพื้นฐานเครือข่ายโดยใช้ซับเน็ตมาสก์แบบยืดหยุ่น (variable length subnet masks: VLSM) การใช้ VLSM เป็นเทคนิคที่ใช้การแบ่งกลุ่มที่อยู่ไอพีที่มีออกเป็นลำดับชั้นของกลุ่มซับเน็ตที่มีขนาดแตกต่างกันเพื่อลดการสูญเปล่าของที่อยู่ไอพี
บริษัท D ใช้กลุ่มที่อยู่ไอพีส่วนตัว (private IP address) 192.168.1.0/24 สำหรับเครือข่ายภายในของบริษัท บริษัท D แบ่งเครือข่าย 192.168.1.0/24 ตามหลักการของ VLSM ออกเป็น 8 ซับเน็ตมาสก์ ได้แก่ 192.168.1.0/30 192.168.1.4/30 192.168.1.8/29 192.168.1.16/28 192.168.1.32/27 192.168.1.64/27 192.168.1.96/27 และ 192.168.1.128/25
รูปที่ 1 แสดงแผนผังเครือข่ายของบริษัท D ตารางที่ 1 แสดงผังรายการที่อยู่ไอพีของเราท์เตอร์และพีซีในแผนกผู้ดูแลระบบ (Admin section) ในที่นี้ทุกแผนกในบริษัท D ต้องสามารถเข้าถึงอินเทอร์เน็ตได้



รูปที่ 1 แผนผังเครือข่ายของบริษัท D

ตารางที่ 1 ผังรายการที่อยู่ไอพี (บางส่วน) ของเราท์เตอร์และพีซีในแผนกผู้ดูแลระบบ (Admin section)

Device	Interface	IP address/mask	Gateway	Section
Router 1	fe01	202.20.120.95/30	N/A	
	fe02	192.168.1.9/29	N/A	
Router 2	fe01	192.168.1.33/27	N/A	
	fe02	192.168.1.65/27	N/A	
	fe03	192.168.1.97/27	N/A	
PC1 … PC20		192.168.1.34/27 … 192.168.1.53/27	192.168.1.33	Admin

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ☐ แต่ละช่องในตารางที่ 2

จากการออกแบบบนพื้นฐานของ VLSM บริษัท D กำหนดซับเน็ตสำหรับแต่ละแผนกตามจำนวนโฮสต์ที่จำเป็นต้องใช้ ที่อยู่ไอพีที่สามารถใช้งานได้ลำดับต่ำที่สุดถูกกำหนดให้ส่วนของ Link A ซึ่งเชื่อมต่อระหว่างเราท์เตอร์ Router 1 กับ Router 2

ตารางที่ 2 แสดงข้อมูลบางส่วนของรายการจัดสรรที่อยู่ไอพี

ตารางที่ 2 รายการการจัดสรรที่อยู่ไอพี (บางส่วน)

Section	Number of hosts	Subnet
Server farm	6	192.168.1.8/29
Link A	2	A
Admin	20	192.168.1.32/27
Sale	25	192.168.1.64/27
Marketing	28	192.168.1.96/27
Engineering	120	B

กลุ่มคำตอบสำหรับ A และ B

a) 192.168.1.0/30 b) 192.168.1.8/29 c) 192.168.1.16/28
d) 192.168.1.32/27 e) 192.168.1.128/25

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่องในคำอธิบายต่อไปนี้

ทุก ๆ ส่วนสามารถเข้าถึงอินเทอร์เน็ต อีเมลภายในและบริการเว็บภายในได้โดยใช้เซอร์ฟเวอร์ DNS ภายในเครือข่าย ผู้ดูแลระบบต้องจัดสรรที่อยู่ไอพีให้กับเซอร์ฟเวอร์ต่าง ๆ ในเซอร์ฟเวอร์ฟาร์ม (server farm) ที่อยู่แรกที่สามารถใช้งานได้ของเครือข่าย 192.168.1.8/29 ถูกจัดสรรให้กับ fe02 ของเราท์เตอร์ Router 1 ขณะที่เซอร์ฟเวอร์ DNS ได้รับการจัดสรรที่อยู่สุดท้ายที่สามารถใช้งานได้ การทดสอบเครือข่ายถูกดำเนินการหลังจากได้กำหนดแผนผังเครือข่ายเสร็จเรียบร้อยแล้ว ผู้ทดสอบจากแต่ละแผนกพยายามเข้าถึงอินเทอร์เน็ต อีเมล และบริการเว็บ แต่ไม่มีผู้ทดสอบรายใดสามารถเข้าถึงบริการต่าง ๆ เหล่านี้ได้เลย โดยพบว่าการเข้าถึงทรัพยากรต่าง ๆ ที่ถูกใช้ร่วมกันสามารถทำได้หากเข้าถึงโดยใช้ที่อยู่ไอพีแทนการใช้ชื่อโดเมน ดังนั้น ผู้ดูแลระบบจึงตัดสินใจตรวจสอบการเชื่อมต่อด้วยคำสั่ง ping ตารางที่ 3 แสดงการตั้งค่าเครือข่ายในปัจจุบันของเซอร์ฟเวอร์ต่าง ๆ

ตารางที่ 3 การตั้งค่าเครือข่ายในปัจจุบันของเซอร์ฟเวอร์ต่าง ๆ

Server	IP address	Subnet mask	Gateway
Web	192.168.1. 10	255.255.255.248	192.168.1. 9
Email	192.168.1. 11	255.255.255.248	192.168.1. 9
DNS	192.168.1. 15	255.255.255.240	192.168.1. 9

จากการทดสอบพบว่านอกจากเซอร์ฟเวอร์ DNS แล้ว ผู้ดูแลระบบสามารถเชื่อมต่อไปยังเซอร์ฟเวอร์อื่น ๆ ทั้งหมดจากเครื่องพีซีของผู้ทดสอบแต่ละคนด้วยคำสั่ง ping ได้ ในเครื่องพีซีทั้งหมดนี้ ที่อยู่ไอพีของเซอร์ฟเวอร์ DNS ถูกกำหนดไว้ดังแสดงในตารางที่ 3

จากข้อเท็จจริงดังกล่าว สามารถสรุปได้ว่า [C] และ [D] ดังนั้น ปัญหานี้จึงสามารถแก้ไขได้ด้วยการป้อน [E] ลงในช่องคุณสมบัติไอพีของเซอร์ฟเวอร์ DNS

กลุ่มคำตอบสำหรับ C และ D

a) เกตเวย์ของเซอร์ฟเวอร์ DNS ไม่ถูกต้อง
b) ที่อยู่ไอพีของเซอร์ฟเวอร์ DNS ชนกับที่อยู่บรอดแคสต์ของซับเน็ต
c) ที่อยู่ไอพีของเซอร์ฟเวอร์ DNS ชนกับที่อยู่ไอพีของอีเมลเซอร์ฟเวอร์ (Email server)
d) ที่อยู่ไอพีของเซอร์ฟเวอร์ DNS ชนกับที่อยู่ไอพีของเว็บเซอร์ฟเวอร์ (Web server)
e) ซับเน็ตมาสก์ของเซอร์ฟเวอร์ DNS ไม่ถูกต้อง

กลุ่มคำตอบสำหรับ E

a) IP address: 192.168.1.8, subnet mask: 255.255.255. 240
b) IP address: 192.168.1.9, subnet mask: 255.255.255. 240
c) IP address: 192.168.1.9, subnet mask: 255.255.255. 248
d) IP address: 192.168.1.14, subnet mask: 255.255.255. 248

คำถามย่อย 3

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] ในคำอธิบายต่อไปนี้

เพื่อเพิ่มความเข้มแข็งของความมั่นคง (security) ในส่วนของเซอร์ฟเวอร์ฟาร์ม บริษัท D ตัดสินใจกำหนดที่อยู่ MAC แบบคงที่ให้กับเซอร์ฟเวอร์ต่าง ๆ ไว้บนสวิทช์ โดยสวิทช์นี้จะอนุญาตให้ส่งผ่านเฟรมไปยังที่อยู่ MAC ของเซอร์ฟเวอร์ที่ลงทะเบียนไว้ได้ หากพบว่าที่อยู่ MAC ไม่ตรงกับที่อยู่ที่ลงทะเบียนไว้ เซอร์ฟเวอร์นั้นก็จะไม่สามารถเชื่อมต่อกับพอร์ตของสวิทช์ได้ หลังจากการกำหนดความมั่นคงในระดับพอร์ตเรียบร้อยแล้ว พบว่าพอร์ตที่ต่ออยู่กับอีเมลเซอร์ฟเวอร์หยุดกระพริบ จากนั้นผู้ดูแลระบบจึงตรวจสอบการกำหนดค่าของสวิทช์และที่อยู่ MAC ของเซอร์ฟเวอร์ที่สัมพันธ์กัน ตารางที่ 4 แสดงข้อมูลที่อยู่ MAC ที่ตรวจสอบได้โดยผู้ดูแลระบบ จากตารางที่ 4 ผู้ดูแลระบบพบสาเหตุของปัญหาและแก้ปัญหาดังกล่าวโดย [F] หลังจากตรวจสอบว่าอีเมลเซอร์ฟเวอร์ที่เชื่อมต่ออยู่นั้นถูกต้องหรือไม่

ตารางที่ 4 ข้อมูลที่อยู่ MAC

Port of switch	Registered MAC	Connected server	Detected MAC
fe02	00-40-0B-20-D9-43	DNS	00-40-0B-20-D9-43
fe03	00-0D-BD-48-33-3E	Web	00-0D-BD-48-33-3E
fe04	00-D0-BA-73-38-E5	Email	00-D0-BA-73-8B-E5

กลุ่มคำตอบสำหรับ F

a) เปลี่ยนเกตเวย์ตั้งต้น (default gateway) ของอีเมลเซอร์ฟเวอร์เป็น 192.168.1.9
b) เปลี่ยนที่อยู่ไอพีของอีเมลเซอร์ฟเวอร์เป็น 192.168.1.13
c) เปลี่ยนที่อยู่ MAC ของอีเมลเซอร์ฟเวอร์เป็น to 00-D0-BA-73-38-E5
d) เปลี่ยนที่อยู่ MAC ที่ลงทะเบียนไว้ที่พอร์ต fe04 เป็น 00-D0-BA-73-8B-E5

Q5. อ่านคำอธิบายของระบบการให้รางวัลของโรงเรียนต่อไปนี้ แล้วตอบคำถามย่อยที่ 1 ถึง 3

โรงเรียน E เป็นโรงเรียนในสังกัดกระทรวงการศึกษาที่เปิดการเรียนการสอนสำหรับนักเรียนระดับประถมศึกษาและมัธยมศึกษาปีที่ 1 ถึง 6 เนื่องจากอายุของนักเรียนมีตั้งแต่ 6 ถึง 18 ปี โรงเรียน E จึงตั้งเป้าที่จะรักษาสมดุลการเติบโตของนักเรียนใจแต่ละช่วงอายุโดยมุ่งเน้นที่การพัฒนาดังต่อไปนี้

(1) ความประพฤติ

- กล่าวถึงพฤติกรรมและทัศนคติของนักเรียนที่มีต่อผู้อื่น และรวมถึงการยินยอมปฏิบัติตามนโยบายและแนวปฏิบัติที่ดีตามที่ได้ตกลงกันไว้แล้ว
- การให้คะแนนความประพฤติ (เรียงจากดีที่สุดไปหาน้อยที่สุด):
  ดีเยี่ยม > ดี > พอใช้ > ควรปรับปรุง

(2) กิจกรรมเสริมนอกหลักสูตร

- กิจกรรมต่าง ๆ ที่ไม่เกี่ยวข้องกับวิชาการซึ่งโดยทั่วไปจะจัดขึ้นนอกชั้นเรียน เช่น กีฬา วารสารศาสตร์ การแข่งขันต่าง ๆ ทัศนศึกษา หรือ การอบรมความเป็นผู้นำ เป็นต้น
- เมื่อนักเรียนเข้าร่วมกิจกรรมนอกห้องเรียนในแต่ละครั้ง ทางโรงเรียนจะให้คะแนนแก่นักเรียนแต่ละคน ผลรวมของคะแนนที่ได้รับจะคำนวณเมื่อสิ้นสุดปีการศึกษา โดยคะแนนดังกล่าวถือเป็นคะแนนกิจกรรมนอกหลักสูตร (Extra-curricular points: ECP) ของนักเรียน
  ตัวอย่างของคะแนนที่ให้กับแต่ละกิจกรรมแสดงดังด้านล่างนี้:
  1. ให้ 4 คะแนนสำหรับนักกีฬาในทีมว่ายน้ำของโรงเรียน E
  2. ให้ 10 คะแนนสำหรับนักเรียนที่ชนะเลิศในการแข่งขันคณิตศาสตร์ระหว่างโรงเรียน

(3) ด้านวิชาการ

- กิจกรรมการเรียนการสอนจะจัดภายในห้องเรียนหรือห้องแลบของโรงเรียน โดยต้องมีส่วนเกี่ยวข้องกับวิชาเรียน เช่นวิชาวิทยาศาสตร์ คณิตศาสตร์ ภาษาอังกฤษ หรือ ประวัติศาสตร์

[เงื่อนไขการผ่านเกณฑ์]

เพื่อจูงใจนักเรียนในการประพฤติดี เมื่อสิ้นสุดปีการศึกษาจะมีรางวัลพิเศษมอบให้แก่นักเรียนที่ผ่านเกณฑ์ต่อไปนี้

(1) รางวัลความประพฤติดี

- มีผลคะแนนความประพฤติรวมอยู่ในเกณฑ์ "ดี" เป็นอย่างน้อย

(2) รางวัลนักกิจกรรมนอกหลักสูตรยอดเยี่ยม Best in Extra-Curricular Activities Award

- มีผลคะแนนกิจกรรมนอกหลักสูตร (ECP) สูงสุด ของนักเรียนในระดับชั้นเดียวกัน

(3) รางวัลความเป็นเลิศทางวิชาการ

- มีผลคะแนนความประพฤติรวมอยู่ในเกณฑ์ "ดี" เป็นอย่างน้อย
- นักเรียนควรมีคะแนนในแต่ละรายวิชาไม่น้อยกว่า 85 คะแนน
- นักเรียนควรมีคะแนนด้านวิชาการโดยเฉลี่ย 95 คะแนนหรือสูงกว่า

(4) รางวัลบัณฑิตโดดเด่น

- มีคะแนนความประพฤติอยู่ในเกณฑ์ "ดี" เป็นอย่างน้อย
- นักเรียนควรมีคะแนนในแต่ละรายวิชาไม่น้อยกว่า 85 คะแนน
- ควรมีคะแนนประสิทธิภาพโดยรวม (Overall performance score: OPS) 95 คะแนนหรือสูงกว่า
- ต้องเป็นนักเรียนที่สามารถสำเร็จการศึกษาได้ (อยู่ในชั้นมัธยมศึกษาปีที่ 6)

[ขั้นตอนการพิจารณาผู้ได้รับรางวัลบัณฑิตโดดเด่น]

(1) รวบรวมรายชื่อนักเรียนที่กำลังศึกษาอยู่ในชั้นมัธยมศึกษาปีที่ 6

(2) นำรายชื่อนักเรียนที่ไม่ผ่านเกณฑ์ออก โดยพิจารณาจากเงื่อนไขดังต่อไปนี้

(i) นักเรียนที่มีคะแนนความประพฤติที่อยู่ในเกณฑ์ต่ำกว่า "ดี"

(ii) นักเรียนที่มีคะแนนในวิชาใด ๆ ต่ำกว่า 85 คะแนน

(3) คำนวณและบันทึกคะแนนกิจกรรมนอกหลักสูตร (ECP) ของนักเรียนแต่ละคน เพื่อหาค่าคะแนนกิจกรรมนอกหลักสูตรต่ำสุด (ECPMin) ซึ่งพิจารณาจาก ECP ที่ต่ำที่สุดจากนักเรียนทั้งหมด และคะแนนกิจกรรมนอกหลักสูตรสูงที่สุด (ECPMax) พิจารณาจากคะแนน ECP สูงที่สุดจากนักเรียนทั้งหมด

(4) คำนวณคะแนน ECScore สำหรับนักเรียนแต่ละคนโดยใช้เงื่อนไขและสูตรดังต่อไปนี้:

$$\text{ECScore} = 100 \times ((\text{ECP} - \text{ECPMin}) \div (\text{ECPMax} - \text{ECPMin}))$$

ในที่นี้ ให้ถือว่า ECPMax > ECPMin

(5) คำนวณคะแนน OPS ของนักเรียนแต่ละคนโดยใช้สูตรดังต่อไปนี้:

$$\text{OPS} = 0.85 \times \text{ค่าเฉลี่ยคะแนนวิชาการ} + 0.15 \times \text{ECScore}$$

(6) นักเรียนทั้งหมดที่มีคะแนน OPS ตั้งแต่ 95 ขึ้นไปจะได้รับรางวัลบัณฑิตโดดเด่น

คำถามย่อย 1

จากคำอธิบายใน [เงื่อนไขการผ่านเกณฑ์] ให้เลือกข้อความที่เป็นจริงสำหรับนักเรียนที่มีคุณสมบัติในการได้รับรางวัลบัณฑิตโดดเด่นจากกลุ่มคำตอบด้านล่าง

กลุ่มคำตอบ

a) นักเรียนที่ได้รับคะแนนเฉลี่ยของแต่ละวิชาไม่น้อยกว่า 95 คะแนน

b) นักเรียนจะต้องผ่านเกณฑ์รางวัลนักกิจกรรมนอกหลักสูตรยอดเยี่ยมด้วย

c) นักเรียนจะต้องผ่านเกณฑ์รางวัลความเป็นเลิศทางวิชาการด้วย

d) นักเรียนจะต้องผ่านเกณฑ์รางวัลความประพฤติดีด้วย

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง ☐ แต่ละช่องในตารางที่ 1

เพื่อเพิ่มประสิทธิภาพในการพิจารณารางวัลแก่นักเรียน โรงเรียน E จึงพัฒนาระบบการพิจารณารางวัลขึ้นมา โดยใช้ตารางการตัดสินใจเพื่อกำหนดเงื่อนไขคุณสมบัติของแต่ละรางวัล

ตารางที่ 1 ตารางการตัดสินใจเพื่อพิจารณารางวัลให้แก่นักเรียน

การพิจารณารางวัลที่จะมอบให้กับนักเรียนแต่ละคน																	
ส่วนเงื่อนไข	คะแนนความประพฤติอยู่ในเกณฑ์ "ดี" เป็นอย่างน้อย	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N
	ไม่มีวิชาที่ได้รับคะแนนต่ำกว่า 85 คะแนน	-	-	Y	Y	Y	Y	Y	Y	Y	A		-	N	N	-	-
	มีคะแนน OPS 95 คะแนนหรือสูงกว่า	N	N	Y	Y	Y	Y	N	N	-			-	-	-	-	-
	เรียนอยู่ชั้นมัธยมศึกษาปีที่ 6	Y	Y	Y	Y	Y	Y	-	-	N			N	-	-	-	-
	มีคะแนนด้านวิชาการโดยเฉลี่ย 95 คะแนนหรือสูงกว่า	N	N	N	N	Y	Y	Y	Y	Y			N	-	-	-	-
	ได้รับคะแนน ECP สูงที่สุดในระดับชั้นเดียวกัน	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
ส่วนการกระทำ	มอบรางวัลความประพฤติดี	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	-
	มอบรางวัลนักกิจกรรมนอกหลักสูตรยอดเยี่ยม	X	-	X	-	X	B				-	X	-	X	-	X	-
	มอบรางวัลความเป็นเลิศทางวิชาการ	-	-	-	-	X					X	-	-	-	-	-	-
	มอบรางวัลบัณฑิตโดดเด่น	-	-	X	X	X					-	-	-	-	-	-	-

หมายเหตุ: ในส่วนเงื่อนไข สัญลักษณ์ "Y" หมายถึงเงื่อนไขเป็นจริง ส่วนสัญลักษณ์ "N" หมายถึงเงื่อนไขเป็นเท็จ และสัญลักษณ์ "-" หมายความว่าเงื่อนไขดังกล่าวไม่เกี่ยวข้องว่าจะเป็นจริงหรือเท็จ

ในส่วนการกระทำ สัญลักษณ์ "X" หมายถึงการกระทำดังกล่าวจะถูกดำเนินการเมื่อเป็นไปตามเงื่อนไขทั้งหมด และสัญลักษณ์ "-" หมายถึงการกระทำดังกล่าวไม่ถูกดำเนินการ

กลุ่มคำตอบสำหรับ A

a)

N	-
-	-
N	N
Y	N

b)

N	-
-	-
N	N
Y	Y

c)

Y	-
-	-
N	N
N	N

d)

Y	-
-	-
N	N
N	Y

e)

Y	-
-	-
N	N
Y	N

f)

Y	-
-	-
N	N
Y	Y

กลุ่มคำตอบสำหรับ B

a)

-	X	-	X
X	X	X	X
-	X	-	X

b)

-	X	-	X
X	X	X	X
X	-	-	-

c)

-	X	-	X
X	X	X	X
X	-	-	X

d)

X	-	X	-
X	X	X	X
-	X	-	X

e)

X	-	X	-
X	X	X	X
X	-	-	-

f)

X	-	X	-
X	X	X	X
X	-	-	X

คำถามย่อย 3

จากคำอธิบายในส่วน [ขั้นตอนการพิจารณาผู้ได้รับรางวัลบัณฑิตโดดเด่น] ให้เลือกข้อความที่เป็นจริงสำหรับนักเรียนชั้นมัธยมศึกษาปีที่ 6 ทุกคนจากกลุ่มคำตอบด้านล่างนี้

กลุ่มคำตอบ

a) นักเรียนที่ได้รับคะแนนความประพฤติในเกณฑ์ "ดีเยี่ยม" มีโอกาสที่จะได้รับรางวัลบัณฑิตโดดเด่นมากกว่า เมื่อเทียบกับนักเรียนที่ได้รับคะแนนความประพฤติในเกณฑ์ "ดี" ภายใต้เงื่อนไขว่านักเรียนทั้งสองจะมีคะแนนด้านวิชาการโดยเฉลี่ยเท่ากันและไม่มีวิชาที่มีคะแนนเฉลี่ยต่ำกว่า 85 คะแนน

b) นักเรียนที่มีคะแนน ECP สูง จะมีโอกาสที่จะได้รับรางวัลบัณฑิตโดดเด่นกว่านักเรียนที่ได้รับคะแนน ECP ต่ำกว่า ภายใต้เงื่อนไขว่านักเรียนทั้งสองมีคะแนนด้านวิชาการโดยเฉลี่ยเท่ากันและมีคะแนนความประพฤติอยู่ในเกณฑ์ "ดี"

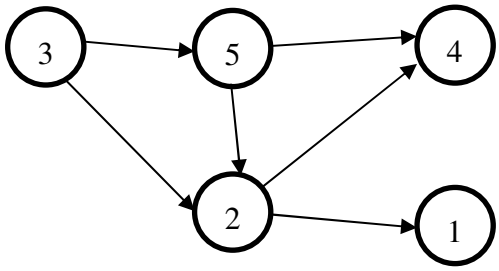
c) นักเรียนคนใดก็ตามที่มีคะแนน ECP เท่ากับคะแนน ECPMax มีคะแนนความประพฤติอยู่ในเกณฑ์ "ดี" และมีคะแนนเฉลี่ยของแต่ละวิชาเท่ากับ 85 คะแนนหรือสูงกว่า มีคุณสมบัติที่จะได้รับรางวัลบัณฑิตโดดเด่น

d) นักเรียนคนใดก็ตามที่มีคะแนน ECP เท่ากับคะแนน ECPMin อาจยังมีคุณสมบัติที่จะได้รับรางวัลบัณฑิตโดดเด่นได้

Q6. อ่านคำอธิบายของโปรแกรมและตัวโปรแกรมต่อไปนี้ จากนั้นให้ตอบคำถามย่อย 1 และ 2 (สามารถดูสัญลักษณ์ที่ใช้ในภาษาเทียม (pseudo-language) ในด้านหน้าของสมุดคำถามนี้)

โปรแกรมนี้จะแก้ไขปัญหาการจัดตารางเวลางานจำนวน N งาน (Task) โดยงานหนึ่งงานอาจจะมีความขึ้นอยู่กับ (dependent) งานอื่น ๆ โดยการขึ้นต่อกันนั้นสามารถนำเสนอได้โดยกราฟแบบมีทิศทาง (directed graph) ที่มีจำนวน *N* โหนด สมมติถ้างาน *t* มีความขึ้นอยู่กับงาน s แล้วจะมีเส้นลูกศรเชื่อมจากโหนด *s* ไปถึงโหนด *t* โปรแกรมนี้จะแสดงผลลัพธ์เป็นลำดับของงานจำนวน N งานที่สามารถจัดลำดับได้ด้วยการรักษาความขึ้นต่อกันระหว่างแต่ละงาน

รูปที่ 1 แสดงถึงกราฟที่มีจำนวนโหนด 5 โหนด ซึ่งแทนงานที่ 1 ถึงงานที่ 5 และมี 6 เส้นลูกศรเชื่อมโยง ยกตัวอย่างเช่นงานที่ 2 ขึ้นอยู่กับงานที่ 3 และงานที่ 5 ดังนั้นจึงมีเส้นลูกศรชี้จากโหนดที่ 3 และโหนดที่ 5 ไปยังโหนดที่ 2



รูปที่ 1 ตัวอย่างกราฟแบบมีทิศทางที่มีโหนด 5 โหนด

[คำอธิบายโปรแกรม]

ตัวแปรและฟังก์ชันทั้งหมดที่ถูกใช้ในโปรแกรมได้แสดงคำอธิบายไว้ในตารางที่ 1 และตารางที่ 2 สำหรับดัชนี (index) ของอาร์เรย์ทั้งหมดจะเริ่มต้นด้วย 1 โดยค่าของตัวแปร *N* และค่าในอาร์เรย์ adj[N][N] ได้มีการตั้งค่าไว้ก่อนล่วงหน้าเพื่อทำให้เหมือนกับกราฟในรูปที่ 1 โดยค่าของ adj[s][t] ถูกตั้งค่าเป็น 1 หากมีเส้นลูกศรจากโหนด *s* ไปยังโหนด *t* มิฉะนั้น adj[s][t] ก็จะถูกตั้งค่าเป็น 0 ดังนั้น adj[2][1], adj[2][4], adj[3][2], adj[3][5], adj[5][2] และ adj[5][4] จะถูกตั้งค่าเป็น 1 และส่วนที่เหลือจะมีค่าเป็น 0

ตารางที่ 1 อธิบายตัวแปรที่ถูกใช้ในโปรแกรม

ตัวแปร	คำอธิบาย
adj[N][N]	เมทริกซ์ (Adjacency matrix) adj[s][t] จะถูกกำหนดค่าเป็น 1 เมื่อมีเส้นลูกศรจากโหนด s ชี้มายังโหนด t และจะถูกกำหนดค่าเป็น 0 ถ้าไม่มีเส้นลูกศรเข้ามา
indeg[N]	indeg[j] ระบุจำนวนเส้นลูกศรที่เข้ามายังโหนด j
queue_array[N]	อาร์เรย์สำหรับจัดการคิว (queue) โดยมีสองฟังก์ชัน insert_queue() และ remove_queue()เพื่อแทรกโหนด (enqueue) และลบโหนดจากคิว (dequeue) ตามลำดับ
schedule[]	ลำดับของงาน

ตารางที่ 2 อธิบายฟังก์ชันที่ถูกใช้ในโปรแกรม

ฟังก์ชัน	คำอธิบาย
Scheduling()	ค้นหาลำดับของงานและใส่ลงไปในอาร์เรย์ schedule[]
indegree(INT: vertex)	คืนค่าจำนวนเส้นลูกศรที่เข้ามายังโหนดหนึ่ง ๆ
insert_queue(INT: vertex)	เพื่อแทรกโหนดลงในคิว
remove_queue()	เพื่อลบโหนดจากคิว

อัลกอริทึมสามารถอธิบายได้ดังนี้

ขั้นตอนที่ 1: แทรกเฉพาะโหนดต่าง ๆ ที่มีค่า indeg[] เป็น 0 ลงในคิว

ขั้นตอนที่ 2: นำ (pop) โหนดต่าง ๆ ออกจากคิว โดยเมื่อโหนด s (s = 1, 2, ... , N) ที่มีค่า indeg[s] เป็น 0 ถูกนำออกมาจากคิว โหนด s นั้นจะถูกใส่ไว้ในอาเรย์ schedule[]

ขั้นตอนที่ 3: ถ้าหากโหนด s ถูกนำออกในขั้นตอนที่ 2 แล้ว ให้ลบเส้นลูกศรทั้งหมดที่มาจากโหนด s ด้วยการลดค่าของตัวแปร indeg[t] ลง 1 โดยทำเช่นนี้กับทุก ๆ ตัวแปร t (t = 1, 2, ... , N) ที่มีเส้นลูกศรจากโหนด s ไปยังโหนด t จากนั้นจึงแทรกโหนดทั้งหมดที่มีค่า indeg[t] เป็น 0 ลงในคิว

ขั้นตอนที่ 4: ทำซ้ำขั้นตอนที่ 2 และ 3 จนกว่าคิวจะว่างเปล่า (empty)

ขั้นตอนที่ 5: หากจำนวนของโหนดที่แทรกลงในอาร์เรย์ schedule[] มีค่าเท่ากับ N เราจะได้ตารางงานของทั้ง N โหนด พร้อมทั้งยังรักษาความขึ้นต่อกันไว้ได้ มิฉะนั้น หมายความว่าไม่มีตารางงานที่รักษาความขึ้นต่อกันได้

ในที่นี้ กำหนดให้ไม่มีเส้นลูกศรใดที่ประกอบกันเป็นลูปปิด

[Program]

```
GLOBAL: INT: N ← 5  /* Assign 5 to N to represent Fig.1 */
GLOBAL: INT: adj[N][N] ← {{0,0,0,0,0}, /* Assign values to array */
                          {1,0,0,1,0}, /* adj to represent Fig.1 */
                          {0,1,0,0,1},
                          {0,0,0,0,0},
                          {0,1,0,1,0}}
GLOBAL: INT: front ← 1, rear ← 1
GLOBAL: INT: indeg[N]
GLOBAL: INT: queue_array[N]
GLOBAL: INT: schedule[N]

FUNCTION: insert_queue(INT: vertex) {
    queue_array[rear] ← vertex;
    rear ← rear + 1;
}

INT: FUNCTION: remove_queue() {
   INT: item
   item ← queue_array[   A   ];
   [   B   ]
   return item;
}

INT: FUNCTION: indegree(INT: vertex) {
   INT: i, in_deg ← 0
   FOR (i ← 1; i ≤ N; i ← i + 1) {
      IF (adj[i][vertex] = 1) {
         in_deg ← in_deg + 1;
      }
   }
   return in_deg;
}

FUNCTION: Scheduling() {
   INT:  i, j ← 1, vertex
   FOR (i ← 1; i ≤ N; i ← i + 1) {
      indeg[i] ← indegree(i);
      IF (indeg[i] = 0) {
         insert_queue(i);              /* α */
      }
   }
```

```
    WHILE (front < rear) {
       vertex ← remove_queue();
       schedule[j] ← vertex;
       j ← j + 1;
       FOR (i ← 1; i ≤ N; i ← i + 1) {  /* γ */
          IF (adj[vertex][i] = 1) {
             adj[vertex][i] ← 0;         /* β */
             indeg[i] ← indeg[i] - 1;
             IF (indeg[i] = 0) {
                insert_queue(i);
             }
          }
       }
    }

    Print("The order of tasks is:");
    FOR (i ← 1; i ≤ N; i ← i + 1) {
       Print(" ", schedule[i]);
    }
}
```

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในแต่ละช่องว่าง [ ] แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ A

a) `1` b) `N` c) `front`
d) `front - 1` e) `front + 1`

กลุ่มคำตอบสำหรับ B

a) `front ← front - 1` b) `front ← front + 1` c) `front ← rear - 1`
d) `front ← rear` e) `front ← rear + 1`

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่องในคำอธิบายต่อไปนี้

ฟังก์ชัน Scheduling จะถูกเรียกใช้ด้วยรายการงานที่แสดงในรูปที่ 1 โดย ณ บรรทัดที่มีสัญลักษณ์ /* α */ จะถูกประมวลผลทั้งสิ้น [C] ครั้ง และบรรทัดที่มีสัญลักษณ์ /* β */ จะถูกประมวลผลทั้งสิ้น [D] ครั้ง นอกจากนี้ ลำดับของงานที่แสดงออกมาโดยโปรแกรมจะเป็นดังนี้:

```
The order of tasks is: [ E ]
```

และถ้าบรรทัดที่มีสัญลักษณ์ / * γ * / ถูกแทนที่ด้วยประโยคคำสั่ง:

```
FOR (i ← N; i ≥ 1; i ← i - 1) {
```

ลำดับของงานที่แสดงออกมาจากโปรแกรมภายหลังก็จะเปลี่ยนแปลงเป็น:

```
The order of tasks is: [ F ]
```

กลุ่มคำตอบสำหรับ C และ D

a) 1	b) 2	c) 3
d) 4	e) 5	f) 6
g) 7	h) 8	

กลุ่มคำตอบสำหรับ E และ F

a) 1 2 4 5 3	b) 1 4 2 5 3	c) 3 2 1 4 5
d) 3 5 2 1 4	e) 3 5 2 4 1	f) 3 5 4 2 1
g) 4 1 2 5 3	h) 5 4 1 2 3	

สำหรับข้อสอบข้อที่ **Q7** และ **Q8 ให้เลือกทำเพียงหนึ่ง**ในสองข้อ
จากนั้น ให้ระบายทับในวงกลม Ⓢ ในกระดาษคำตอบสำหรับข้อที่เลือกทำ
หากเลือกไว้ทั้งสองข้อ จะตรวจให้คะแนนเฉพาะข้อแรกเท่านั้น

Q7. อ่านคำอธิบายโปรแกรมภาษา C และตัวโปรแกรมต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

เกมคอมพิวเตอร์เกมหนึ่งได้รับการออกแบบมาเพื่อจำลองการแข่งขันในหมู่ผู้เล่น N ($2 \le N \le 10$) คน โดยกำหนดหมายเลขผู้เล่นเป็น 0, 1, …, N − 1 เกมนี้เล่นบนกระดานที่มีจำนวน 50 ช่องเรียงต่อกันเป็นแถว โดยกำหนดหมายเลขเป็น 1, 2, …, 50

กติกาของเกมมีดังนี้:

(1) ผู้เล่นทั้งหมดอยู่ที่ช่องตารางที่ 1 เมื่อเริ่มเกม ผู้เล่นที่สามารถไปถึงช่องที่ 50 ก่อนจะเป็นผู้ชนะ

(2) ผู้เล่นทุกคนจะได้รับ "หมายเลขนำโชค (lucky number)" ที่แตกต่างกันจำนวนสามหมายเลขจาก 1 ถึง 6 ก่อนเริ่มเกม หนึ่งในตัวเลขเรียกว่า "หมายเลขนำโชคหลัก (primary lucky number)" และอีกสองหมายเลขที่เหลือเรียกว่า "หมายเลขนำโชครอง (secondary lucky number)" ตัวเลขเหล่านี้จะถูกใช้ในกฎถัดไป

(3) ผู้เล่นแต่ละคนทอยลูกเต๋าสองครั้งตามลำดับตามหมายเลขของตนจากน้อยไปหามาก เพื่อได้สิทธิในการเคลื่อนที่ไปตามจำนวนช่องที่ทอยได้

(i) การทอยลูกเต๋าครั้งแรก คือจำนวนช่องที่ผู้ทอยสามารถเคลื่อนที่ไปได้ตามจำนวนที่ทอยได้ ตัวอย่างเช่น หากผู้เล่นทอยลูกเต๋าได้ 3 ผู้เล่นสามารถเคลื่อนที่ไปได้ 3 ช่องในรอบนี้

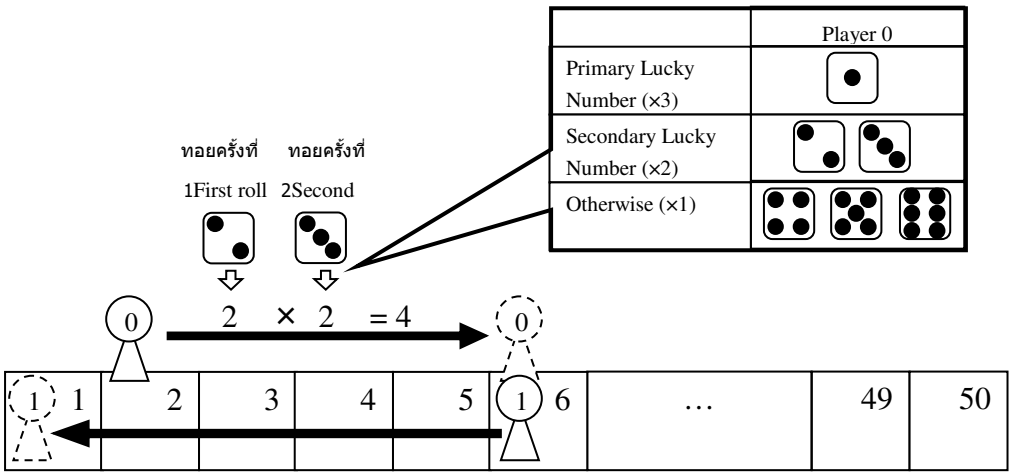
(ii) การทอยลูกเต๋าครั้งที่สอง เป็นการกำหนดตัวคูณ ที่จะคูณกับจำนวนช่องที่ผู้เล่นสามารถเคลื่อนที่ไปได้ ตัวคูณถูกกำหนดไว้ตามที่แสดงในตารางที่ 1

ตารางที่ 1 ตารางแสดงตัวคูณ

การทอยลูกเต๋าครั้งที่สอง	ตัวคูณ
หมายเลขนำโชคหลัก	×3
หมายเลขนำโชครอง	×2
อื่น ๆ	×1

(4) หากผู้เล่นเคลื่อนย้ายไปยังที่ช่องที่ถูกจับจองโดยผู้เล่นอื่นอยู่ก่อนแล้ว ผู้เล่นที่เข้ามาใหม่จะได้ครอบครองช่องนั้น และผู้เล่นที่อยู่ก่อนหน้าต้องถอยกลับไปยังช่องที่ 1

รูปที่ 1 แสดงตัวอย่างการเคลื่อนที่ของผู้เล่นในระหว่างเกมดำเนินอยู่ จากตัวอย่างนี้มีผู้เล่นสองคน: ผู้เล่น 0 หยุดอยู่ที่ช่องที่ 2 และผู้เล่น 1 ได้หยุดที่ช่องที่ 6 สมมติว่าผู้เล่น 0 ได้รับ 1 เป็นหมายเลขนำโชคหลัก โดยมี 2 และ 3 เป็นหมายเลขนำโชครอง ขณะนี้ผู้เล่น 0 ทอยลูกเต๋าสองครั้งและได้รับ 2 ตามด้วย 3 ซึ่งจะช่วยให้ผู้เล่น 0 ย้ายไปช่องที่ 6 และผู้เล่น 1 ซึ่งครอบครองช่องที่ 6 อยู่ ต้องถูกย้ายกลับไปช่องที่ 1



รูปที่ 1 ตัวอย่างการเคลื่อนที่ของผู้เล่นในเกม

คำอธิบายโครงสร้างข้อมูลและฟังก์ชั่นต่าง ๆ ที่ใช้ในการขับเคลื่อนเกมเป็นดังต่อไปนี้ :

(1) ฟังก์ชัน rand1 คืนค่าตัวเลขสุ่มจาก 1 (รวมทั้ง 1) ถึงตัวเลขที่กำหนดโดยอาร์กิวเมนต์ (รวมทั้งตัวเลขนั้น)

(2) ฟังก์ชัน init จัดวางผู้เล่นทุกคนไว้ยังช่องที่ 1 และเลือกหมายเลขนำโชคของผู้เล่นแต่ละคน

(3) ฟังก์ชัน move ทำหน้าที่เคลื่อนย้ายผู้เล่นที่กำหนดด้วยอาร์กิวเมนต์เพียงตัวเดียว และกำหนดให้ผู้เล่นอื่นกลับไปช่องที่ 1 หากจำเป็น

(4) ฟังก์ชัน print พิมพ์สถานะปัจจุบันของเกมไปยังเอาต์พุตมาตรฐาน ฟังก์ชันนี้จะแสดง '|' สำหรับทุก ๆ 10 ช่อง และ '-' สำหรับช่องอื่น ๆ ยกเว้นเงื่อนไขต่อไปนี้:

(i) ในช่องที่ 1 และ 50 ฟังกชันจะแสดง 'S' และ 'G ' ออกมาแทนตามลำดับ

(ii) ในช่องอื่น ๆ ที่ผู้เล่นเข้าครอบครองนอกจาก (i) ฟังก์ชันจะแสดงหมายเลขของผู้เล่น '0' ถึง '9'

(5) อาร์เรย์ position ทำหน้าที่เก็บจำนวนช่องที่ผู้เล่นแต่ละคนเคลื่อนย้ายไปโดยเริ่มนับจากช่องที่ 1 ตัวอย่างเช่นหาก position[0] มีค่าเป็น 1 หมายความว่าผู้เล่น 0 ไปหยุดที่ช่องที่ 2

(6) อาร์เรย์ p_choice เก็บค่าตัวคูณ (3, 2 หรือ 1) ที่กำหนดไว้ในตารางที่ 1 ตัวอย่างเช่นเมื่อผู้เล่น N ทอยเต๋าครั้งที่สองได้ D แล้ว p_choice [N] [D - 1] ก็จะกำหนดตัวคูณสำหรับคูณจำนวนช่องเดินที่ทอดเต๋าได้ในตานั้น

[Program]

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int rand1(int);
void init_lucky_numbers(int);
void init(void);
int move(int);
void print(void);

#define N 4

int p_choice[N][6];
int position[N];

int rand1(int n) {
    return 1 + rand() / (((unsigned int)RAND_MAX + 1) / n);
}

void init_lucky_numbers(int playerNO) {
    int i, j, k;
    const int multipliers[] = { 1, 1, 1, 2, 2, 3 };
    for (i = 0; i < 6; i++) {
        p_choice[playerNO][i] = [   A   ];
    }
    for (j = 6; j > 0; j--) {
        k = rand1([   B   ]);
        for (i = 0; i < 6; i++) {
            if (p_choice[playerNO][i] == 0) {
                k--;
                if (k == 0) {
                    p_choice[playerNO][i] = multipliers[j - 1];
                    break;
                }
            }
        }
    }
}
```

```
void init(void) {
    int i;
    /* srand resets the random number sequence */
    srand(time(NULL));
    for (i = 0; i < N; i++) {
        position[i] = 1;
        init_lucky_numbers(i);
    }
}

int move(int playerNO) {
    int base_steps = rand1(6);
    int multiplier = p_choice[playerNO][rand1(6) - 1];
    int steps = base_steps * multiplier;
    int i;
    position[playerNO] += steps;
    if (position[playerNO] >= 50) {
        position[playerNO] = 50;
        return 1;
    }
    for (i = 0; i < N; i++) {
        if (i != playerNO && position[i] == position[playerNO]) {
            position[i] = 1;
            break;
        }
    }
    return 0;
}

void print(void) {
    char status[]
            = "S--------|---------|---------|---------|---------G";
    int playerNO;
    int squareNO;

    for (playerNO = N-1; playerNO >= 0; playerNO--) {
        squareNO = position[playerNO] - 1;
        if (status[squareNO] == '-' || status[squareNO] == '|') {
            status[squareNO] = [    C    ];
        }
    }
    printf("%s\n", [    D    ]);
}
```

```
int main() {
    int playerNO = 0;
    init();                                  α
    while (!move(playerNO)) {
        print();
        playerNO = [    E    ];
    }
    /* print the last status of the game */
    print();
    printf("Player NO. %d win!\n", playerNO);
    return 0;
}
```

β

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ A

a) `-1` b) `-N` c) `0`
d) `1` e) `N`

กลุ่มคำตอบสำหรับ B

a) `j - 1` b) `j` c) `j + 1`
d) `N - 1` e) `N` f) `N + 1`

กลุ่มคำตอบสำหรับ C

a) `playerNO - '0'` b) `playerNO + '0'`
c) `position[playerNO]` d) `position[playerNO] - 1`

กลุ่มคำตอบสำหรับ D

a) `*status` b) `status` c) `status[0]`
d) `status[50]`

กลุ่มคำตอบสำหรับ E

a) `(playerNO % N) + 1` b) `(playerNO + 1) % N`
c) `playerNO % N` d) `playerNO + 1`

คำถามย่อย 2

เมื่อโปรแกรมถูกแก้ไขดังที่แสดงด้านล่างนี้เพื่อพิมพ์ผู้เล่นทุกคนที่ไปถึงช่องที่ 50 ด้วยจำนวนตาที่เดินเท่ากันกับผู้ชนะ

(1) แทรกบรรทัดต่อไปนี้ไว้ก่อนตำแหน่งที่กำหนดโดย α

```
unsigned int finished = 0;
```

(2) แทนที่ 7 บรรทัดที่กำหนดโดย β ด้วย 9 บรรทัดต่อไปนี้

```
while (!finished) {
    for (playerNO = 0; playerNO < N; playerNO++) {
        [   F   ];
        print();
        if (position[playerNO] == 50) {
            printf("Player NO. %d win!\n", playerNO);
        }
    }
}
```

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [F] แต่ละช่องในโปรแกรมข้างต้น

กลุ่มคำตอบสำหรับ F

a) `finished &= move(playerNO)`
b) `finished = move(playerNO)`
c) `finished = ~move(playerNO)`
d) `finished ^= move(playerNO)`
e) `finished |= move(playerNO)`

Q8. อ่านคำอธิบายโปรแกรมภาษาจาวาและตัวโปรแกรมต่อไปนี้ แล้วตอบคำถามย่อย 1 และ 2

[คำอธิบายโปรแกรม]

โปรแกรมนี้ใช้ติดตามคะแนนสอบของนักเรียน โดยที่นักเรียนทุกคนจะมีรหัสประจำตัว (ID) ที่ไม่ซ้ำกัน นักเรียนบางคนอาจจะไม่มีคะแนนและคะแนนอาจจะมีการเปลี่ยนแปลงในภายหลัง นอกจากนี้โปรแกรมยังสนับสนุนการจัดกลุ่มของนักศึกษาโดยนักเรียนในแต่ละกลุ่มนั้นสามารถถูกจัดเรียงข้อมูลด้วยแอททริบิวต์ที่แตกต่างกันได้

(1) คลาส Student สื่อถึงข้อมูลนักเรียนที่มีแอททริบิวต์ 3 อย่าง ดังนี้
   (i) ID: รหัสนักเรียน
   (ii) name: ชื่อนักเรียน
   (iii) score: ค่าคะแนน หากไม่ปรากฏจะกำหนดค่าให้เป็น -1.0

   คลาสนี้จะมีคอนสตรัคเตอร์ 2 ตัว โดยตัวหนึ่งจะเก็บค่าแอททริบิวต์ทั้งหมดในพารามิเตอร์ ในขณะที่อีกตัวหนึ่งนั้น จะรับค่าเฉพาะ ID และ name ในพารามิเตอร์ และค่า -1.0 จะถูกใช้เป็นค่าของ score เพื่อสื่อว่านักเรียนคนนี้ยังไม่ได้ทำข้อสอบ หรือยังไม่ปรากฏค่าคะแนน
   เมธอด withScore จะคืนค่าอินสแตนซ์ Student ที่ถูกสร้างขึ้นด้วย ID และ name ของ Student นี้ พร้อมระบุค่า score

(2) คลาส StudentGroup สื่อถึงกลุ่มของนักเรียน โดยที่คอนสตรัคเตอร์จะสร้างอินสแตนซ์ที่ระบุความจุของจำนวนนักเรียนที่มากที่สุดที่อินสแตนซ์สามารถจะเก็บได้ และสามารถเพิ่มคลาส Students ลงในคลาส StudentGroup ได้ด้วยการเรียกเมธอด addStudents สำหรับค่า score ของนักเรียนแต่ละรายนั้น สามารถจะปรับแก้ไขด้วยการเรียกเมธอด updateScore นอกจากนี้ เมธอด sort สามารถจัดเรียงข้อมูลนักเรียนได้ตามลำดับที่กำหนด

(3) enum SortOrder นั้นจะใช้กำหนดค่าคงที่เพื่อระบุว่าแอทริบิวต์ของคลาส Student จะถูกจัดเรียงอย่างไรใน StudentGroup โดยที่ enum จะอิมพลีเมนต์ อินเตอร์เฟส Comparator เพื่อให้เมธอด sort ของ StudentGroup สามารถเรียกใช้เมธอด compare เพื่อเปรียบเทียบคลาส Student จำนวน 2 ชุด ในที่นี้ ค่าคงที่ประกอบด้วย
   (i) BY_ID: กำหนดให้ใช้แอททริบิวต์ ID ในการจัดเรียงลำดับจากน้อยไปมาก
   (ii) BY_NAME: กำหนดให้ใช้แอทริบิวต์ name ในการจัดเรียงลำดับจากน้อยไปมาก โดยไม่สนใจตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก (case-insensitive)
   (iii) BY_SCORE: กำหนดให้ใช้แอททริบิวต์ score ในการจัดเรียงลำดับจากมากไปน้อย

(4) คลาส StudentGroupTester ใช้ทดสอบคลาสด้านบน และ enum ทั้งนี้ ผลลัพธ์ต่อไปนี้เกิดจากการรันเมธอด main ของ StudentGroupTester

```
Sorted by SCORE:
(111, Abco, 100.0)
(444, Cdef, 70.0)
(222, Ghio, 0.0)
(333, Jokl, no score)

Sorted by NAME:
(111, Abco, 100.0)
(444, Cdef, 70.0)
(222, Ghio, 0.0)
(333, Jokl, no score)

Sorted by ID:
(111, Abco, 100.0)
(222, Ghio, 0.0)
(333, Jokl, no score)
(444, Cdef, 70.0)
```

[Program 1]

```
class Student {
   private final int id;
   private final String name;
   private final double score;

   Student(int id, String name) {
      [   A   ](id, name, -1);
   }

   Student(int id, String name, double score) {
      this.id = id;
      this.name = name;
      this.score = score;
   }

   int getId() { return id; }

   String getName() { return name; }

   double getScore() { return score; }
```

```
    Student withScore(double newScore) {
        return [   B   ](id, name, newScore);
    }

    @Override
    public String toString() {
        return String.format("(%d, %s, %s)", id, name,
                                score [   C   ] 0 ? "no score" : score);
    }
}
```

[Program 2]

```
class StudentGroup {
    private final Student[] students;
    private int size;

    StudentGroup(int capacity) {
        students = new Student[capacity];
    }

    void addStudents(Student... studentsToAdd) {
        for (Student student : studentsToAdd) {
            if (size == students.length) {
                throw new IllegalStateException("No more space");
            }
            students[   D   ] = student;
        }
    }

    void updateScore(int id, double score) {
        for (int i = 0; i < size; i++) {
            if (students[i].getId() == id) {
                students[i] = students[i].withScore(score);
                return;
            }
        }
        throw new IllegalArgumentException("Student not found");
    }

    void sort(SortOrder order) {
        for (int i = 0; i < [   E   ] - 1; i++) {
            for (int j = i + 1; j < [   E   ]; j++) {
                if (order.compare(students[i], students[j]) [   F   ] 0) {
                    swap(i, j);
```

```
            }
         }
      }
   }

   private void swap(int i, int j) {
      Student temp = students[i];
      students[i] = students[j];
      students[j] = temp;
   }

   @Override
   public String toString() {
      StringBuilder records = new StringBuilder();
      for (int i = 0; i < size; i++) {
         records.append(students[i]).append(System.lineSeparator());
      }
      return records.toString();
   }
}
```

[Program 3]

```
import java.util.Comparator;

enum SortOrder implements Comparator<Student> {
   BY_ID((s1, s2) -> Integer.compare(s1.getId(), s2.getId())),
   BY_NAME((s1, s2) -> s1.getName().compareToIgnoreCase(s2.getName())),
   BY_SCORE((s1, s2) -> -Double.compare(s1.getScore(), s2.getScore()));

   private final Comparator<Student> comparator;

   SortOrder(Comparator<Student> comparator) {
      this.comparator = comparator;
   }

   String getAttributeName() {
      return name().substring(3);
   }

   // Compares its two arguments for order. Returns a negative integer, zero, or a positive
   // integer as the first argument is less than, equal to, or greater than the second.
   @Override
   public int compare(Student s1, Student s2) {
      return comparator.compare(s1, s2);
   }
}
```

[Program 4]

```
public class StudentGroupTester {
   public static void main(String[] args) {
      StudentGroup studentGroup = new StudentGroup(4);
      studentGroup.addStudents(new Student(111, "Abco", 100),
                               new Student(222, "Ghio", 0),
                               new Student(333, "Jokl"),
                               new Student(444, "Cdef", 70));

      sortAndPrint(studentGroup, SortOrder.BY_SCORE);
      sortAndPrint(studentGroup, SortOrder.BY_NAME);
      sortAndPrint(studentGroup, SortOrder.BY_ID);
      /* α */
   }

   private static void sortAndPrint(StudentGroup studentGroup,
                                    SortOrder order) {
      studentGroup.sort(order);
      System.out.printf("Sorted by %s:%n%s%n",
                        order.getAttributeName(), studentGroup);
   }
}
```

คำถามย่อย 1

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่องใน Program 1 และ Program 2

กลุ่มคำตอบสำหรับ A และ B

a) new Student b) Student c) super d) this

กลุ่มคำตอบสำหรับ C

a) != b) < c) <= d) ==
e) > f) >=

กลุ่มคำตอบสำหรับ D

a) ++size b) --size c) size d) size++
e) size--

กลุ่มคำตอบสำหรับ E

a) size
b) size + 1
c) size - 1
d) student.length
e) student.length + 1
f) student.length - 1

กลุ่มคำตอบสำหรับ F

a) !=
b) <
c) ==
d) >

คำถามย่อย 2

จากกลุ่มคำตอบด้านล่าง ให้เลือกคำตอบที่ถูกต้องเพื่อเติมลงในช่องว่าง [ ] แต่ละช่องในผลลัพธ์ของโปรแกรม

หากแทนที่คอมเมนต์ **/* a */** ในเมธอด main ของ StudentGroupTest ด้วยคำสั่งสองบรรทัดต่อไปนี้ จะทำให้ได้ผลลัพธ์เพิ่มเติมดังแสดงในกล่องด้านล่างขึ้นมาจากผลลัพธ์ที่แสดงอยู่ก่อนหน้านี้

```
studentGroup.updateScore(333, 70);
sortAndPrint(studentGroup, SortOrder.BY_SCORE);
```

```
Sorted by SCORE:
    (111, Abco, 100.0)
    ( [   G   ], 70.0)
    ( [   H   ], 70.0)
    (222, Ghio, 0.0)
```

กลุ่มคำตอบสำหรับ G และ H

a) 333, Cdef
b) 333, Jokl
c) 444, Cdef
d) 444, Jokl