

การใช้ระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบิน

FLIGHT OPERATIONS CONTROL USING AN EXPERT SYSTEM



วิทยานิพนธ์สำหรับปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2532

ห้องสมุดบัณฑิตวิทยาลัย สจล.

การใช้ระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบิน

Flight Operations Control Using An Expert System



ACCESSION NO. 00066

DATE 24 MAR 1990

CALL NUMBER.....

วิทยานิพนธ์สำหรับปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในห้องสมุดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2532

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มการให้คะแนนการสอบวิทยานิพนธ์

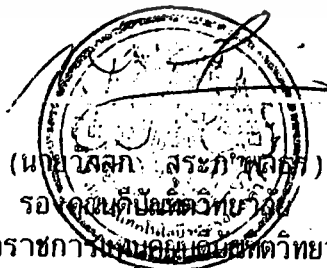
หัวข้อวิทยานิพนธ์ การใช้ระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบิน
(FLIGHT OPERATIONS CONTROL USING AN EXPERT SYSTEM)

ชื่อนักศึกษา นายศิลา ตังวารัตร์ เลขประจำตัว 30126-035
หลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้า วิชาเอก วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา ศ.ดร.ไพรัช อึ้งยงษ์

ชื่ออาจารย์ผู้ควบคุมการสอบ	ลายมือชื่อ	ผลการสอบ
ศ.ดร.ไพรัช อึ้งยงษ์		ดี
นายอารวิชัย กัญญา อุษยชา		ดีเยี่ยม
ดร.ศุภมิตร จิตตะยาศิธร		ดี
ดร.วราวัฒน์ ลิ้มโกคา		ดี
ผศ.ครรชิต ไมตรี		ผ่าน

วัน/เดือน/ปีที่สอบ 10 สิงหาคม 2532 เวลา 10.00 น. สถานที่ ห้อง A-305

บัณฑิตวิทยาลัยรับรองแล้ว



(นาย วัลลภ สระภักดิ์ศรี)
รองคณบดีบัณฑิตวิทยาลัย
รักษาราชการแทนคณบดีบัณฑิตวิทยาลัย
วันที่..16..เดือน..สิงหาคม..พ.ศ.2532..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	I
ABSTRACT	II
บทที่ 1 บทนำ	1
บทที่ 2 วัตถุประสงค์ ความเป็นมาและขอบเขตของงานวิจัย	3
2.1 วัตถุประสงค์	3
2.2 ความเป็นมาและขอบเขตของงานวิจัย	3
2.2.1 ความเป็นมา	3
2.2.2 ขอบเขตของงานวิจัย	4
บทที่ 3 ระบบผู้เชี่ยวชาญ	5
3.1 ความเป็นมาของระบบผู้เชี่ยวชาญ	5
3.1.1 วิวัฒนาการของระบบผู้เชี่ยวชาญ	5
3.1.2 การจัดการรูปแบบของความรู้	8
3.1.2 เครื่องมือที่ใช้พัฒนาระบบผู้เชี่ยวชาญ	9
3.2 ลักษณะ โดยทั่วไปของระบบผู้เชี่ยวชาญ	12
3.2.1 รูปแบบการทำงาน of ระบบผู้เชี่ยวชาญ	12
3.2.2 ส่วนประกอบของระบบผู้เชี่ยวชาญ	13
3.3 การแทนความรู้ของระบบผู้เชี่ยวชาญ	15
3.3.1 ประเภทของความรู้	15
3.3.2 วิธีการแทนความรู้ที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์	16
3.3.3 รูปแบบการแทนความรู้	17

3.4	เทคนิคสำหรับการสร้างระบบผู้เชี่ยวชาญ	19
บทที่ 4	ลักษณะของงานควบคุมปฏิบัติการบิน	21
4.1	ทางเลือกปฏิบัติการบินในเหตุการณ์ไม่ปกติ	21
4.1.1	การเลื่อนกำหนดเวลาของเที่ยวบิน	21
4.1.2	การเปลี่ยนเส้นทางการบิน	21
4.1.3	การยกเลิกเที่ยวบิน	21
4.1.4	การเปลี่ยนเครื่องบิน	23
4.2	ส่วนประกอบของการปฏิบัติการบิน	23
4.3	ผลกระทบของการเลือกปฏิบัติการบิน	26
บทที่ 5	โครงสร้างและการทำงานของโปรแกรม	28
5.1	ขั้นตอนการสร้างระบบ	28
5.2	การจัดรูปแบบความรู้ของการควบคุมปฏิบัติการบิน	29
5.2.1	ลำดับขั้นของกรอบ	29
5.2.2	ส่วนประกอบและผลกระทบของทางเลือกปฏิบัติการบิน	29
5.2.2.1	เงื่อนไขของส่วนประกอบทางเลือกปฏิบัติการบิน	29
5.2.2.2	เงื่อนไขของผลกระทบทางเลือกปฏิบัติการบิน	40
5.2.3	ฐานข้อมูลของระบบ	59
5.3	การทำงานของโปรแกรม	60
บทที่ 6	บทสรุปและข้อเสนอแนะ	64
	กิตติกรรมประกาศ	66
	เอกสารอ้างอิง	67
	ภาคผนวก	

ภาคผนวกที่ 2	ฐานข้อมูลของระบบ	70
ภาคผนวกที่ 3	โพล์ซาร์ทของเงื่อนไขทั้งหมด	76
ภาคผนวกที่ 4	รายละเอียดของโปรแกรม	109
ตารางที่ 3-1	ตัวอย่างระบบผู้เชี่ยวชาญคัตันๆ	7
ตารางที่ 3-2	ตัวอย่าง เบสิคระบบผู้เชี่ยวชาญ	11
ตารางที่ 4-1	ส่วนประกอบของการปฏิบัติการบิน	25
ตารางที่ 4-2	ผลกระทบของทางเลือกปฏิบัติการบิน	27



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ การใช้ระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบิน
 นักศึกษา นายศิลา ตั้งวาริธร
 อาจารย์ที่ปรึกษา ศาสตราจารย์ ดร. ไพรัช ธีชัยพงษ์
 ระดับการศึกษา วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
 ปีการศึกษา 2532

บทคัดย่อ

สิ่งสำคัญอันหนึ่งที่ขาดไม่ได้ในระบบผู้เชี่ยวชาญคือ การแทนความรู้ซึ่งมีได้หลายวิธีในแต่ละวิธีมีลักษณะที่แตกต่างกันออกไป วิทยานิพนธ์นี้ได้นำเอาวิธีการแทนความรู้ในรูปแบบของฐานกฎ (Rule-Base) และกรอบ (Frames) มาใช้กับระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบินและใช้ภาษา LISP เป็นภาษาสร้างระบบ เนื่องจากการควบคุมปฏิบัติการบินเป็นงานที่ต้องอาศัยความรวดเร็ว ถูกต้อง และความชำนาญในการปฏิบัติงาน เพื่อให้เกิดประสิทธิภาพมากที่สุดในการแก้ปัญหา เมื่อเกิดกรณีผิดปกติของเที่ยวบินต่างๆ เช่น เครื่องเสีย สภาพอากาศแปรปรวน หรือกรณีอื่นๆ เป็นต้น ฝ่ายผู้เชี่ยวชาญควบคุมปฏิบัติการบินจะเป็นผู้เลือกหนทางแก้ปัญหาเมื่อเกิดกรณีผิดปกติขึ้น หนทางในการแก้ปัญหาประกอบไปด้วย การยกเลิกเที่ยวบิน, การเลื่อนกำหนดเวลาของเที่ยวบิน, การเปลี่ยนเส้นทางการบิน, การเปลี่ยนแบบเครื่องบิน โดยระบบนี้จะให้ข้อมูลออกมาเพื่อสนับสนุนการตัดสินใจของผู้เชี่ยวชาญ

Thesis Title Flight Operations Control Using An Expert System
Name Sila Thangwaritorn
Thesis Advisor Pairah Thajchayapong Ph.D.
Level of study Master of Engineering in Electrical Engineering
Academic Year 1989



ABSTRACT

Knowledge representation, which is one of the most important aspect of expert systems development, is based on various methodologies. In this thesis, rule and frames are used in an expert system for flight operations control. The systems was developed in a LISP environment. The task of flight operations control is to solve such problems as aircraft on ground and weather change. This requires high-speed, accuracy and skill to efficiently solve the problems. When an aircraft is in an irregular condition, the operation control will either required to cancel the flight, delay the flight, reroute the flight orchange the aircraft. The system presented here performs such operation control. Still, the final decision is made by human experts using the systems information.

บทที่ 1

บทนำ

เทคนิคทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) ได้ถูกนำมาประยุกต์ใช้ให้เกิดประโยชน์กันอย่างกว้างขวาง ระบบผู้เชี่ยวชาญ (Expert Systems) ก็เป็นสาขาหนึ่งของงานวิจัยทางด้านปัญญาประดิษฐ์ ระบบผู้เชี่ยวชาญนั้นเป็นเทคโนโลยีทางด้านซอฟต์แวร์ที่นำมาประยุกต์ใช้งานด้านต่างๆ อย่างมากมาย เช่น ระบบผู้เชี่ยวชาญทางพยากรณ์อากาศ (WILLARD) ระบบผู้เชี่ยวชาญวิเคราะห์เคมี (DENDRAL) เป็นต้น

เนื่องจากงานควบคุมปฏิบัติการบินนั้นสิ่งสำคัญที่ต้องคำนึงถึงก็คือ ความถูกต้องแม่นยำและความรวดเร็วในการปฏิบัติงาน จึงเป็นเหตุให้มีการนำเอาเทคนิคของระบบผู้เชี่ยวชาญมาใช้งาน เพื่อเป็นการสนับสนุนงานทางด้านนี้ให้เกิดประสิทธิภาพมากขึ้นในกรณีที่เกิดเหตุขัดข้องต่างๆ เกี่ยวกับเที่ยวบิน เช่น เครื่องบินเสีย, นักบินไม่พร้อม ฯลฯ อันเป็นผลทำให้เที่ยวบินไม่สามารถบินได้ตามกำหนดเวลาในตารางบินที่ถูกระบุไว้ทุกๆ 6 เดือนอย่างแน่นอนแล้วจึงทำให้เจ้าหน้าที่ฝ่ายควบคุมปฏิบัติการบินจะต้องหาหนทางในการแก้ไขปัญหาให้เข้าสู่ภาวะปกติโดยเร็วที่สุด โดยจะต้องคำนึงถึง ความปลอดภัย ความรวดเร็วและผลกระทบต่อต่างๆ ที่เกี่ยวข้องอย่างมากมาย ซึ่งในบางครั้งการตัดสินใจภายใต้สภาวะเวลาที่จำกัด อาจทำให้ผลกระทบบางประการมิได้ถูกคำนึงถึง

วิทยานิพนธ์นี้ได้เสนอการนำเอาระบบผู้เชี่ยวชาญมาใช้ควบคุมปฏิบัติการบินโดยจะกล่าวถึงทฤษฎี โครงสร้างและการทำงานของระบบจึงได้แบ่ง เนื้อหาในวิทยานิพนธ์นี้ออกเป็น 5 บท ดังนี้

บทที่ 2 ได้กล่าวถึง วัตถุประสงค์และความเป็นมาของงานวิจัย

บทที่ 3 ทฤษฎีของระบบผู้เชี่ยวชาญ อธิบายถึงลักษณะของระบบผู้เชี่ยวชาญ วิธีการแทนความรู้ของระบบผู้เชี่ยวชาญ รวมไปถึง เทคนิคของการสร้างระบบผู้เชี่ยวชาญ

บทที่ 4 อธิบายถึงการทำงานของเจ้าหน้าที่ฝ่ายควบคุมปฏิบัติการบิน เป็นงานหลักบางส่วนเท่านั้นที่จะมาอธิบาย เพราะอันที่จริงแล้วงานของฝ่ายควบคุมปฏิบัติการบินเป็นงานที่ยุ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยากสลับซับซ้อนมากเกินไปที่จะอธิบายถึง ดังนั้นในที่นี้จะกล่าวถึง หนทางเลือกในการปฏิบัติ การบิน ข้อบังคับของการเลือกปฏิบัติการบิน และผลกระทบของการเลือกปฏิบัติการบิน

บทที่ 5 แสดงถึงโครงสร้างและการทำงานของโปรแกรมการจัดรูปแบบของความรู้ โครงสร้างของโปรแกรม และเงื่อนไขต่างๆที่นำมาใช้ในกระบวนการเก็บฐานข้อมูล และการแทน ความรู้

บทที่ 6 เป็นบทสุดท้ายจะสรุปผลการทำงานของระบบ พร้อมทั้งข้อเสนอแนะของการ พัฒนาในขั้นต่อไป



บทที่ 2

วัตถุประสงค์และความเป็นมาของระบบ

2.1 วัตถุประสงค์

ในงานวิจัยการใช้ระบบผู้เชี่ยวชาญควบคุมปฏิบัติการบินมีวัตถุประสงค์แยกออกเป็นข้อๆ ได้ดังนี้

- เพื่อที่จะนำเอาทฤษฎีและหลักการของระบบผู้เชี่ยวชาญมาประยุกต์ใช้ในงานควบคุมปฏิบัติการบิน
 - สนับสนุนการตัดสินใจของผู้เชี่ยวชาญ
 - เป็นการศึกษาและวิจัยถึงรูปแบบของความรู้เพื่อที่จะนำความรู้มารวบรวมนำไปใช้ต่อไป
 - เพื่อลดความสูญเสียในแง่ต่างๆ ลงได้อย่างมาก เช่น เวลา ค่าใช้จ่าย
- นอกจากนี้ ยังเป็นการศึกษาถึงโอกาส ความเป็นไปได้ของการนำเอาระบบผู้เชี่ยวชาญ ไปใช้กับหน่วยงานอื่นๆ เช่น ฝ่ายช่าง ลุกเรือ ฯลฯ หรือ กับหน่วยงานอื่นที่ไม่ใช่เกี่ยวกับการบินต่อไปในอนาคต รวมไปถึงการได้ทราบถึงปัญหาที่เกิดขึ้นในระหว่างการพัฒนาระบบด้วย

2.2 ความเป็นมาและขอบเขตของงานวิจัย

2.2.1 ความเป็นมา

เนื่องจากความก้าวหน้าทางเทคโนโลยีของคอมพิวเตอร์ได้เจริญไปอย่างรวดเร็ว การพัฒนาระบบผู้เชี่ยวชาญ เพื่อนำไปใช้งานในแง่ต่างๆก็เช่นเดียวกัน สำหรับระบบผู้เชี่ยวชาญของธุรกิจทางการบินนั้นได้นำมาใช้กันแพร่หลาย เช่นระบบ Gate Assignment ของสายการบิน United Airline เป็นระบบผู้เชี่ยวชาญที่คอยจัดการว่าเครื่องบินเครื่องใดจะจอดที่ทางเข้าออกของผู้โดยสารทางใด หรือระบบผู้เชี่ยวชาญเลือกอาหารสำหรับบริการผู้โดยสารบนเครื่องบินของสายการบินสิงคโปร์แอร์ไลน์ ส่วนระบบผู้เชี่ยวชาญเกี่ยวกับการควบคุมปฏิบัติการบินนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า มีสายการบิน SAS, สายการบิน AA (อเมริกันแอร์ไลน์) ที่กำลังพัฒนามาใช้กันอยู่ในปัจจุบัน

ด้วยเหตุนี้ทางการบินไทยจึง เกิดความคิดที่จะนำเอาระบบผู้เชี่ยวชาญมาใช้กับฝ่ายควบคุมปฏิบัติการบิน เพราะทางฝ่ายควบคุมปฏิบัติการบินมีความพร้อมในแง่ต่างๆ และอัตราการขยายตัวของจำนวนผู้โดยสาร จำนวนเที่ยวบินที่เพิ่มขึ้นในปัจจุบันและอนาคต ดังนั้นวิทยานิพนธ์นี้จึงได้ทำการรวบรวมความรู้ของฝ่ายควบคุมปฏิบัติการบินในบางส่วนขึ้นมาเพื่อพัฒนาระบบผู้เชี่ยวชาญขึ้น เพื่อนำมาช่วยเพิ่มประสิทธิภาพและสนับสนุนการตัดสินใจของฝ่ายควบคุมปฏิบัติการบิน

2.2.2 ขอบเขตของงานวิจัย

สำหรับวิทยานิพนธ์นี้เป็นระบบผู้เชี่ยวชาญที่ช่วยในการสนับสนุนการตัดสินใจของฝ่ายควบคุมปฏิบัติการบิน เมื่อผู้ควบคุมปฏิบัติการบินเลือกทาง เลือกต่างๆแล้วระบบจะพิจารณาถึงผลกระทบของทาง เลือกนั้น และจะแจ้งให้ทราบถึงข้อมูลของผลกระทบนั้นๆ เพื่อผู้ควบคุมปฏิบัติการบินจะได้นำข้อมูลเหล่านั้นมาทำการตัดสินใจในการ เลือก



บทที่ 3 ระบบผู้เชี่ยวชาญ

การศึกษาค้นคว้าทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) ได้มีมาตั้งแต่ปี ค.ศ.1956 สาขาที่ประสบผลสำเร็จอย่างมาก คือ สาขาระบบผู้เชี่ยวชาญซึ่งเริ่มมาตั้งแต่ราวปี ค.ศ.1965 จนปัจจุบันได้มีการสร้างโปรแกรมที่มีความสามารถสูง โปรแกรมเหล่านี้ได้ถูกออกแบบเพื่อแทนความรู้เฉพาะด้าน

3.1 ความเป็นมาของระบบผู้เชี่ยวชาญ

เนื้อหาวิชาระบบผู้เชี่ยวชาญเป็นการพิจารณาวิธีการและเทคนิคในการสร้างระบบผู้เชี่ยวชาญที่ใช้ในการแก้ปัญหาเฉพาะด้าน ความเชี่ยวชาญในที่นี้ประกอบด้วยความรู้ที่เกี่ยวกับเรื่องใดเรื่องหนึ่ง ความเข้าใจปัญหาเรื่องนั้นๆ และทักษะในการแก้ปัญหาบางอย่างในเรื่องนั้นๆ

3.1.1 วิวัฒนาการของระบบผู้เชี่ยวชาญ

วิวัฒนาการของระบบผู้เชี่ยวชาญใน 2 ทศวรรษที่ผ่านมา โครงการส่วนใหญ่จะใช้เวลาดำเนินการหลายปี ในที่นี้จะนำโครงการที่สำคัญและน่าสนใจที่ได้มีการพัฒนาและประสบความสำเร็จ มากล่าวพอเป็นที่สังเขปดังนี้ [2]

-DENDRAL ของมหาวิทยาลัยแสตนฟอร์ด ใช้เวลาพัฒนาจนถึงปัจจุบันนี้รวมทั้งสิ้น 16 ปีผลงานของโครงการนี้ก็คือ DENDRAL และ META-DENDRAL ระบบ DENDRAL ใช้วิเคราะห์ mass spectrographic, nuclear magnetic resonance และข้อมูลการทดลองเคมีอื่นๆ โดยสามารถวินิจฉัยหาโครงสร้างทางเคมีที่เป็นไปได้ของสารประกอบ ระบบ META-DENDRAL เป็นระบบที่พัฒนาต่อจาก DENDRAL โดยเพิ่มความรู้ในการเสนอและการเลือกกฎต่างๆที่แยกกันอยู่ สำหรับโครงสร้างทางอินทรีย์เคมีระบบสามารถที่จะให้กำเนิดกฎและทดสอบกฎเหล่านั้น โดยตรวจสอบกับข้อมูลที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MACSYMA เป็นระบบผู้เชี่ยวชาญทางด้านคณิตศาสตร์ พัฒนาขึ้นที่ MIT ระบบนี้มีความสามารถในการทำการคิดพีเอเรเนซีเอทและอินทิเกรท โดยใช้สัญลักษณ์และความสามารถลดรูปนิพจน์ทางคณิตศาสตร์ได้อย่างดีเยี่ยม MASYMA ประกอบด้วยกฎที่ได้รับจากผู้เชี่ยวชาญทางคณิตศาสตร์ประยุกต์หลายร้อยกฎ แต่ละกฎจะแสดงวิธีการเปลี่ยนรูปนิพจน์หนึ่งไปยังอีกนิพจน์หนึ่งที่สมมูลกัน สำหรับการแก้ปัญหาและเป็นการหาวิธีการเชื่อมโยงของกฎซึ่งทำการเปลี่ยนนิพจน์เริ่มต้นไปเป็นอีกนิพจน์ที่กะทัดรัดมากขึ้น ระบบพัฒนาด้วยภาษา LISP

- MYCIN เป็นระบบที่ใช้วินิจฉัยโรคจากัดเฉพาะโรคติดเชื้อทางเลือด ระบบประกอบด้วยกฎประมาณ 400 กฎจากการประเมินผลการทำงานของ MYCIN พบว่ามีความสามารถใกล้เคียงกับแพทย์ผู้เชี่ยวชาญ ระบบพัฒนาด้วยภาษา LISP

- PROSPECTOR เป็นระบบที่อาศัยรูปแบบการแทนความรู้ที่คล้ายคลึงสำหรับความสัมพันธ์ของการพบแหล่งแร่ ปัจจุบันระบบประกอบด้วยฐานความรู้ประมาณ 12 ฐาน-ความรู้ ที่เกี่ยวกับแต่ละประเภทของแหล่งแร่ ระบบพัฒนาด้วยภาษา LISP

- HEARSAY-II เป็นระบบเข้าใจคำพูด (speech understanding system) พัฒนาโดยมหาวิทยาลัยคาร์เนกีเมลลอน เป็นระบบแรกในสองระบบที่สามารถเข้าใจการสนทนาติดต่อกันรวมแล้วเป็นคำศัพท์ทั้งหมด 1,000 คำ แม้ว่าความสามารถของระบบนี้จะเทียบได้กับเด็กอายุ 10 ขวบก็ตาม แต่ระบบนี้ไม่ได้อาศัยวิธีการที่มีประสิทธิภาพของระบบผู้เชี่ยวชาญ งานสร้างระบบที่เข้าใจคำพูดนับได้ว่าเป็นงานที่ยากที่สุดเมื่อเทียบกับสาขาอื่น ๆ ในด้านปัญญาประดิษฐ์

สำหรับระบบอื่น ๆ ที่ไม่ได้กล่าวถึงจะแสดงให้ทราบในตารางที่ 3-1

SYSTEM	DATE	AUTHOR	SUBJECT
Dendral	1965	Stanford	Infers information about chemical structures
Macsyma	1965	MIT	Performs complex mathematical analysis
Hearsay	1973	Carnegie-Mellon	Natural-language interpreta- tion for subset language
Age	1973	Stanford	Expert-system-generation tool
Mycin	1972	Stanford	Diagnosis of blood disease
Teiresias	1972	Stanford	Knowledge transformation tool
Prospector	1972	Stanford Res.Inst	Mineral exploration and identification tool
Rosie	1978	Rand	Expert-system-building tool
OPS5	1974	Canegie-mellon	Expert-system-building tool
R1	1978	Canegie-mellon	Configuration for DEC computer equipment
Caduceus	1975	Univ.of.Pittsburgh	Diagnostic tool for internal medical

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ตารางที่ 3-1 แสดงระบบผู้เชี่ยวชาญในเขตนี้ [7]
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การจัดการรูปแบบของความรู้

ในช่วงปี 1950 ถึง 1960 ความรู้ที่ใช้ในโปรแกรมปัญญาประดิษฐ์ยุคแรก เป็นแบบแฮนด์คราฟต์(hand-craft) [1] ผู้เขียนโปรแกรมจะเปลี่ยนรูปแบบความรู้เป็นรหัสโดยไม่มี การแยกความรู้ออกจากกลไกเหตุผล โดยการทําคำความรู้แบบแฮนด์คราฟต์ที่ผู้เขียนโปรแกรมจะ ต้องมีการเรียนรู้ความชำนาญจากผู้เชี่ยวชาญด้วยถึงจะเขียนโปรแกรมได้คือ ผู้เขียนโปรแกรมจะต้องเป็นผู้เชี่ยวชาญก่อนจึงสามารถเขียนโปรแกรมขึ้นมาได้

แต่เมื่อไม่นานมานี้วิศวกรรมด้านความรู้กลายมาเป็นวิธีที่ใช้เพื่อได้มาซึ่งความรู้ โดยการให้ผู้เชี่ยวชาญมีการติดต่อโต้ตอบกับวิศวกรความรู้ หรือโปรแกรมที่ใช้สร้างระบบผู้เชี่ยวชาญ ดังนั้นวิศวกรรมความรู้จึงได้กลายเป็นสาขาวิชาหนึ่งที่ประกอบวิธีดำเนินการ (methodology) สาเหตุที่เกิดสาขาวิชานี้ขึ้นมาก็เนื่องมาจากความเชี่ยวชาญของผู้เชี่ยวชาญมักจะทำให้อยู่ในรูปของอัลกอริทึมไม่ได้ จึงต้องหาวิธีที่จะสามารถนำไปใช้อยู่ในรูปที่คอมพิวเตอร์สามารถประมวลผลได้ การนำเอาความเชี่ยวชาญ การประติดประต่อและการประมวลผลความรู้เป็นงานหลักของสาขาวิชานี้

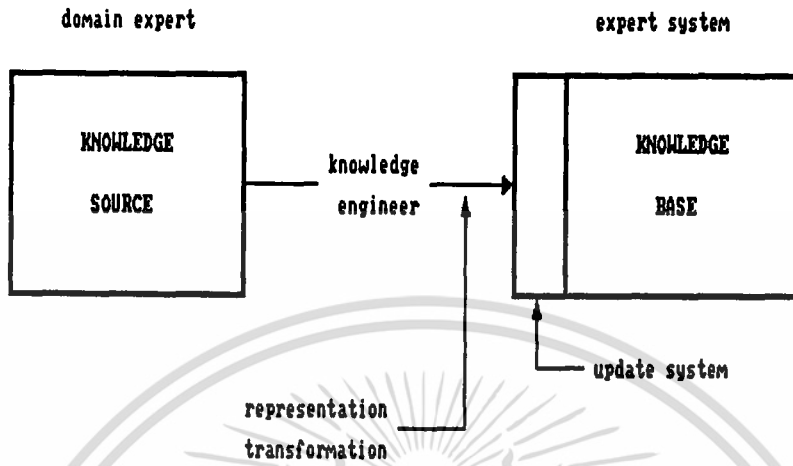
ศาสตราจารย์ Edward A. Feigenbaum ได้อธิบายความหมายของวิศวกรรมความรู้ (knowledge engineering) ไว้ดังนี้ [1,7]

"วิศวกรความรู้ (knowledge engineer) จะเป็นผู้ที่นำเอาหลักการและเครื่องมือของงานวิจัยปัญญาประดิษฐ์มาใช้กับปัญหาของงานที่ยาก ซึ่งต้องอาศัยความรู้ของผู้เชี่ยวชาญเพื่อแก้ปัญหานั้น หัวข้อทางเทคนิคของการได้มาซึ่งความรู้ การแทนความรู้และการใช้สิ่งเหล่านั้น ให้เหมาะสมกับการสร้างและอธิบายการอ้างอิง เหตุผล ซึ่งส่วนนี้เป็นปัญหาสำคัญในการออกแบบฐานความรู้ของระบบผู้เชี่ยวชาญ ศิลปะในการสร้างตัวแทนของความฉลาดนี้เป็นทั้งส่วนหนึ่งของการโปรแกรมและเป็นส่วนที่เพิ่มเติมเข้าไป ซึ่งมันเป็นศิลปะของการสร้างโปรแกรมคอมพิวเตอร์ที่มีความซับซ้อน เพื่อให้แทนและอ้างอิงถึงความรู้ที่มีอยู่"

ในเวลาอีกไม่นานนักวิศวกรความรู้จะเข้ามามีบทบาทสำคัญ ดังรูปที่ 3-1 เมื่อความรู้มีบทบาทในการแก้ไขปัญหที่สำคัญ ๆ ซึ่งสาขาวิศวกรรมความรู้จะเอื้ออำนวยให้เกิดผลดี 2 ด้านด้วยกัน [3] คือ

1. เทคโนโลยีทางด้านวิศวกรรมความรู้ จะนำไปสู่การแปรเปลี่ยนความรู้มาอยู่ในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ลักษณะของเอกสารนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1 หน้าที่ของวิศวกรความรู้ [7]

2. การค้นคว้าวิจัยทางด้านวิศวกรรมความรู้ จะนำไปสู่การรวบรวม การจัดระเบียบ การแลกเปลี่ยน การใช้ประโยชน์รูปแบบที่ใช้งานได้และ เป็นสิ่งสำคัญที่ช่วยเร่งการพัฒนา การขยายความรู้ ความเข้าใจในความรู้มากยิ่งขึ้น

3.1.3 เครื่องมือที่ใช้พัฒนาระบบผู้เชี่ยวชาญ

เครื่องมือที่ใช้พัฒนาระบบผู้เชี่ยวชาญส่วนมากจะ เป็นไปในลักษณะทางด้าน software สามารถแบ่งได้ 2 ประเภท คือ ภาษาโปรแกรมและระบบเปลือกผู้เชี่ยวชาญในการค้นคว้าและพัฒนาในด้านปัญญาประดิษฐ์ ต้องการภาษาคอมพิวเตอร์ที่มีความสามารถตีความ ภาษาที่ใช้ในด้านธุรกิจและวิทยาศาสตร์โดยทั่วไป ภาษาที่ใช้ในด้านนี้จะหนักไปในการประมวลผลในสิ่งที่ เป็นรูปสัญลักษณ์มากกว่าตัวเลขภาษาด้านปัญญาประดิษฐ์ในยุคแรกๆจะหนักไปในด้าน LISP-Processing และได้มีการพัฒนาให้มีความสามารถมากขึ้นเรื่อยๆในการ search, pattern matching เรื่อยไปจนถึง backtracking ปัจจุบันมีอยู่ 2 ภาษาที่นิยมใช้กันในด้านปัญญาประดิษฐ์ คือ ภาษา LISP และภาษา PROLOG

เอกสารนี้เป็นเอกสาร LISP[12] ซึ่ง เป็นภาษาที่ประมวลผลเกี่ยวกับข้อมูลประเภทใช้ linked list ภาษาแรก ผู้ที่กำเนิดคือ John McCarthy ในปี ค.ศ. 1958 คำว่า LISP ย่อมาจาก

LIST Processing เป็นภาษาที่จัดการเกี่ยวกับข้อมูลประเภท list เป็นหลัก McCarthy ได้กล่าวถึงแนวความคิดของ LISP มีดังนี้

1. การประมวลผลข้อมูลในรูปสัญลักษณ์มากกว่าตัวเลข
2. ประมวลผลข้อมูลที่เป็น List นั่นคือแสดงข้อมูลด้วยโครงสร้าง Link-list ในกลไกและ list หลาย ๆ ระดับชั้น
3. ความคมโครงสร้างขึ้นอยู่กับส่วนประกอบของ function มากกว่า function ที่ซับซ้อน
4. การเรียกตัวเอง เป็นแนวทางของการอธิบายขบวนการและปัญหา
5. การแสดงของโปรแกรม LISP ภายในเป็น link list และภายนอกเป็น lists หลาย ๆ ระดับ นั่นคือเหมือนกันในรูปของข้อมูลทั้งหมดที่แสดง

PROLOG[12] เป็นภาษาที่นิยมใช้ทางด้านปัญญาประดิษฐ์อีกภาษาหนึ่งแรกเริ่ม ได้ถูกพัฒนาขึ้นมาจาก U. of Marseilles, ฝรั่งเศส ในปี ค.ศ. 1972 คำว่า PROLOG ย่อมาจาก Programming in Logic โปรแกรมใน PROLOG จะมีการทำงานดังนี้

1. ระบุบางสิ่งเกี่ยวกับวัตถุและความสัมพันธ์
2. ระบุกฎเกี่ยวกับวัตถุและความสัมพันธ์กัน
3. ถามคำถามเกี่ยวกับวัตถุและความสัมพันธ์กัน

ดังนั้นเมื่อเรากำหนดความจริงตามนี้

likes (bob, mary).

likes (paul, mary).

likes (mary, bob).

และเมื่อถาม

? - likes (bob, mary).

PROLOG จะให้คำตอบ

yes

ที่กล่าวมาแล้วเป็นตัวอย่างของภาษาที่ส่วนใหญ่นำมาใช้ในการพัฒนาระบบผู้เชี่ยวชาญ ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า อาจมีภาษาอื่นได้อีก เช่น C, PASCAL, BASIC ฯลฯ

ถ้าเราสร้างโปรแกรมที่ทำหน้าที่เกี่ยวกับการอนุมานฐานความรู้และจัดการเกี่ยวกับการสร้างฐานความรู้ที่วิศวกรความรู้ป้อนให้ จะทำให้ประหยัดเวลาและช่วยให้สร้างระบบผู้เชี่ยวชาญได้มากขึ้นโปรแกรมที่ทำหน้าที่เหล่านี้เรียกว่า "ระบบเปลือกผู้เชี่ยวชาญ" (Expert system shell) ซึ่งเปรียบเสมือนเปลือกหรือโครงของระบบผู้เชี่ยวชาญที่ไม่มีฐานความรู้ของตนเอง ระบบเปลือกผู้เชี่ยวชาญ (Expert system shell) ที่พัฒนาขึ้นมามีตั้งแต่บนเครื่องไมโครคอมพิวเตอร์ไปจนถึงระดับเครื่องเมนเฟรม แต่ที่นิยมใช้กันส่วนมากมักจะอยู่บนเครื่องไมโครคอมพิวเตอร์ ในแต่ละระบบนั้นจะมีการจัดการแทนความรู้แตกต่างกันออกไป ดังตัวอย่างของระบบเปลือกผู้เชี่ยวชาญดังตารางที่ 3-2 ความสามารถของแต่ละระบบนั้นแตกต่างกันออกไป ในการเลือกใช้แต่ละระบบต้องพิจารณาถึงความสัมพันธ์ของการแทนความรู้และลักษณะของความรู้

Shell	Base system	Language	Reference	Model
Age	Hearsay II	Lisp	Nii, 1979	Rule, independent knowledge sources
Emycin	Mycin	Lisp	Buchanan, 1984	Rules, backward chaining diagnostic consultant
Expert	Casnet	Fortran	Weiss, 1984	Rules, classification diagnostic consultant
KAS	Prospector	Lisp	Duda, 1984	Rules, semantic networks forward

ฯลฯ

การพัฒนาระบบผู้เชี่ยวชาญบนภาษาโปรแกรม และ เลือกระบบผู้เชี่ยวชาญ นั้นมีข้อดี และ ข้อเสียแตกต่างกันออกไปขึ้นอยู่กับความต้องการของวิศวกรความรู้ในการเลือกใช้

3.2 ลักษณะโดยทั่วไปของระบบผู้เชี่ยวชาญ

3.2.1 รูปแบบการทำงานของระบบผู้เชี่ยวชาญ

การจัดรูปแบบการทำงานของระบบผู้เชี่ยวชาญ สามารถแยกออกได้เป็น 5 แบบ ซึ่งแยกตามพื้นฐานของความรู้ที่ได้มาและขึ้นอยู่กับความต้องการของวิศวกรความรู้ที่จัดการกับความรู้นั้นได้ดังนี้ [3]

ระบบฐานความรู้ (Knowledge-Base information systems) ลักษณะการทำงานของระบบเอนเอียงไปทางการจัดการฐานข้อมูลขนาดใหญ่ เป็นระบบที่ผู้ใช้ติดต่อถาม-ตอบระหว่างข้อมูล

ระบบสนับสนุนการตัดสินใจ (Decision support systems) เป็นระบบที่ช่วยแก้ปัญหาได้บ้าง เพื่อสนับสนุนการตัดสินใจของผู้เชี่ยวชาญของปัญหา ในระบบนี้จะประกอบไปด้วยฐานความรู้เป็นจำนวนมาก

ระบบให้คำปรึกษา (Consultation systems) เป็นระบบที่ให้คำปรึกษาวิเคราะห์ และ รายงานปัญหานั้นๆ สามารถสรุปผลในการแก้ปัญหาได้และผู้ใช้อย่างถึง เหตุผลของการแก้ปัญหาได้

ระบบแก้ปัญห (Problem solving systems) การแก้ปัญหาบางอย่างนั้น ในบางกรณีต้องอาศัยความรู้ที่ไม่สามารถอธิบายได้ ซึ่งเป็นหน้าที่ของวิศวกรความรู้จะต้องทำการจัดรูปแบบของความรู้เหล่านั้นรวมทั้งการตัดแปลงความรู้เหล่านั้นป้อนให้แก่ระบบ เพื่อให้แก้ไขปัญหานั้นได้อย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเกี่ยวกับการสอน (Coaching systems) เป็นระบบที่สอนนักเรียนหรือผู้ที่ยังไม่มีประสบการณ์ในการแก้ปัญหา เพื่อให้เขาเหล่านั้นสามารถแก้ไขปัญหาได้ ปัจจุบันบางระบบอาจไม่เหมาะสมกับบุคคลทุกคนไปแต่ในอนาคตระบบนี้จะมีการพัฒนาเป็นระบบที่เหมาะสมเฉพาะแต่ละบุคคล ฐานความรู้ของระบบนี้อาจมีมาใหญ่เกินไป การพัฒนาระบบนี้ค่อนข้างจะยุ่งยากพอสมควร

3.2.2 ส่วนประกอบของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญโดยทั่วไปจะประกอบด้วยส่วนสำคัญ 5 ส่วน ดังต่อไปนี้ [7]

1) ฐานความรู้ (Knowledge base) เป็นส่วนที่ใช้เก็บสะสมความรู้เบื้องต้นใน Domain Expert หรือขอบเขตของความรู้ในระบบผู้เชี่ยวชาญที่ต้องการสร้างขึ้น ในส่วนนี้เปรียบเสมือนกับข้อมูลในโปรแกรมธรรมดาหรือฐานข้อมูลต่างกันตรงที่ฐานความรู้ประกอบด้วย

ก) ข้อมูลความจริงต่าง ๆ (fact)

ข) สมมุติฐานและการเชื่อ (Assumption and beliefs)

ความรู้ที่เก็บสะสมอยู่นี้ คือ สิ่งที่เกิดขึ้นให้ระบบผู้เชี่ยวชาญสามารถทำงานได้ใกล้เคียงผู้เชี่ยวชาญ โดยปกติแล้วความรู้เหล่านี้มักจะเก็บอยู่ในรูปแบบความจริงและกฎ อย่างไรก็ตามกรรมวิธีการเก็บความรู้นี้ก็อาจจะแตกต่างกันจากระบบผู้เชี่ยวชาญประเภทหนึ่งกับอีกประเภทหนึ่ง

2) กลไกวินิจฉัย (Inference engine) หมายถึงส่วนของระบบผู้เชี่ยวชาญที่จะค้นหาความรู้จากฐานความรู้และช่วยสร้างความรู้ใหม่ๆ เพื่อนำไปสู่การสรุปผลเป็นคำตอบในระบบผู้เชี่ยวชาญมีความพยายามในการพัฒนาการวินิจฉัยขึ้นหลายรูปแบบ เพื่อให้ระบบผู้เชี่ยวชาญมีประสิทธิภาพสูง แต่การวินิจฉัยพื้นฐานที่เป็นที่รู้จักกันดี ก็คือการวินิจฉัยย้อนกลับ (Backward - Chaining), การวินิจฉัยไปข้างหน้า (Forward-chaining) และการวินิจฉัยแบบรวม (Hybrid) ซึ่งอาศัยวิธีการทั้งสองทำงานร่วมกันตามความเหมาะสม

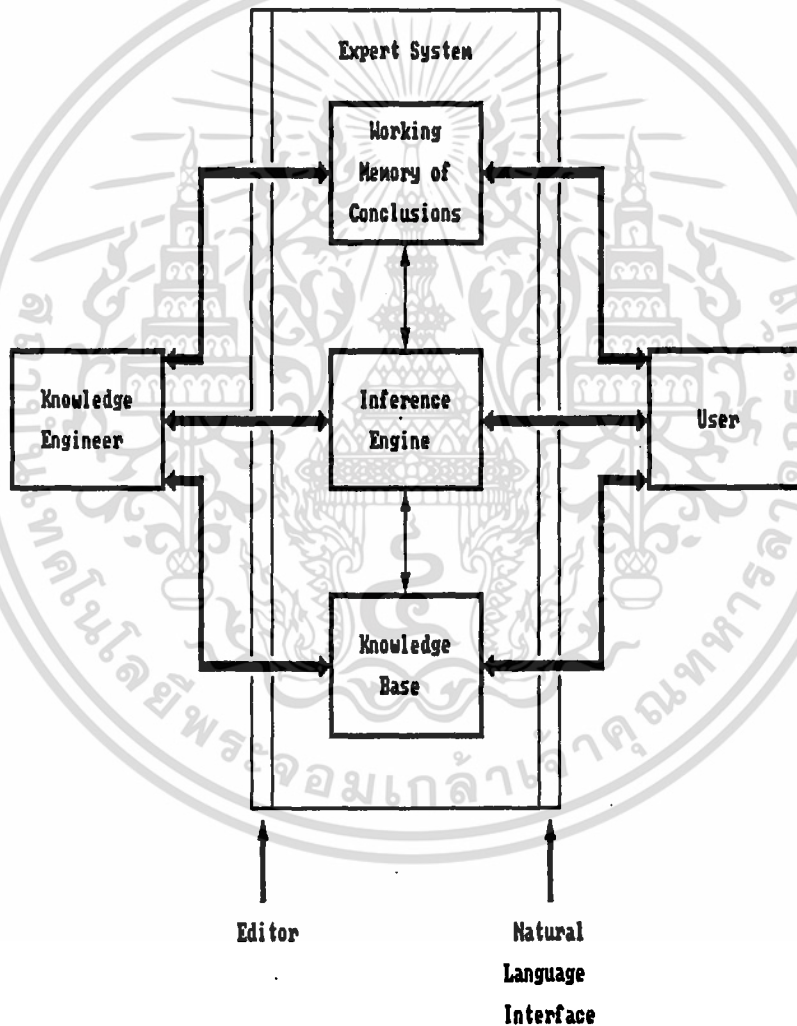
3) ส่วนดึงความรู้ (Knowledge Acquisition) เป็นส่วนดึงความรู้จาก

ผู้เชี่ยวชาญ เอกสาร หนังสือ หรือฐานข้อมูล

ผู้สร้างเป็นเอกสารที่ส่งวินโดวส์สำหรับกรูเซจ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทํางานของส่วนดังกล่าวนี้จะมีหน้าที่ใหญ่ ๆ อยู่ 2 ประการ คือ

- ก) เป็นหน่วยรับความรู้ เช่น กฎเกณฑ์ต่าง ๆ จากผู้เชี่ยวชาญ หรือจากวิศวกรความรู้ (Knowledge Engineer) แล้วนำความรู้ที่ได้เหล่านั้นส่งให้กับส่วนกลไกวินิจฉัยเพื่อนำไปใช้ในการวินิจฉัยต่อไป
- ข) ทำหน้าที่ติดต่อกับผู้ใช้ เพื่อรับข้อมูลที่ใช้ต้องการที่จะปรึกษามาทำการประมวลผลร่วมกับความรู้ที่มีอยู่ในฐานความรู้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ 3-2 โครงสร้างของระบบผู้เชี่ยวชาญ [8] เอกสารทุกครั้งที่มีการนำไปใช้

4) ส่วนอธิบาย (Explanation) เป็นส่วนที่ทบทวนที่ อธิบายและให้เหตุผลแก่ผู้ใช้งาน ในขณะที่ใช้งานนั้นอยู่ เช่น ให้เหตุผลแก่ผู้ใช้งานว่า ทำไมระบบผู้เชี่ยวชาญจึงได้ตั้งคำถามนั้นขึ้นมา และคำถามนั้นมีความเกี่ยวข้องกับกฎเกณฑ์ต่างๆในฐานความรู้อย่างไรบ้าง เป็นต้น การให้คำอธิบายนี้จะเป็นลักษณะเดียวกันกับผู้เชี่ยวชาญให้คำอธิบายแก่เรา ส่วนนี้ของโปรแกรมเป็นส่วนที่มีไม่อยู่ในโปรแกรมคอมพิวเตอร์มาตรฐานทั่วไป การอธิบายควรกระทำด้วยภาษาธรรมชาติที่เราเข้าใจได้ง่าย

5) ส่วนติดต่อกับผู้ใช้ (User Interface) เป็นตัวกลางระหว่างผู้ใช้กับระบบ เพื่อทำให้การสื่อสารระหว่างผู้ใช้กับระบบเป็นไปอย่างราบรื่น และทำให้ผู้ใช้ยอมรับระบบมากขึ้น จากที่กล่าวมาแล้ว ระบบผู้เชี่ยวชาญบางระบบอาจจะมีไม่ครบทั้ง 5 ส่วนก็ได้ แต่หัวใจของระบบผู้เชี่ยวชาญนั้นอยู่ที่ฐานความรู้ (Knowledge base) และกลไกวินิจฉัย (Inference Engine) ถึงแม้จะยังต้องพึ่งส่วนวินิจฉัยอยู่ก็ตาม ความรู้ที่แฝงเอาไว้วิชาชีพและทักษะประสบการณ์เข้าไปด้วยกันนี้ ทำให้ระบบผู้เชี่ยวชาญมีความสามารถในการแก้ปัญหาได้ดี

3.3 การแทนความรู้ของระบบผู้เชี่ยวชาญ

การแทนความรู้ (Knowledge Representation) คือ ลักษณะการจัดรูปแบบของฐานความรู้เพื่อจัดเก็บบนเครื่องคอมพิวเตอร์ การเลือกหรือออกแบบลักษณะของการแทนความรู้ มีผลกระทบอย่างมากต่อการออกแบบชนิดของกลไกวินิจฉัย และประสิทธิภาพ โดยส่วนรวมของระบบผู้เชี่ยวชาญ

3.3.1. ประเภทของความรู้

ความรู้ที่มนุษย์เรามีอยู่นั้นมีหลายรูปแบบ ถึงแม้เราจะไม่รู้ว่าความรู้เหล่านั้นถูกเก็บอยู่ในสมองในรูปโครงสร้างแบบใด แต่การที่จะทำให้คอมพิวเตอร์สามารถใช้ความรู้ได้ เราจำเป็นต้องบันทึกความรู้ในรูปแบบโครงสร้างใดโครงสร้างหนึ่งเข้าไปในคอมพิวเตอร์ปัญหาที่เราเรียกว่า การแทนความรู้ (Knowledge Representation) ซึ่งเป็นหัวใจของการสร้างระบบความรู้ ความรู้แม้จะมีหลายรูปแบบแต่พอจะแยกออกเป็น 4 ประเภทใหญ่ๆ ได้ดังนี้ [7]

- ความรู้ที่จะบอกความจริง, ลักษณะหรือคุณสมบัติ
- ความรู้ที่บอกความสัมพันธ์, เหตุการณ์
- ความรู้ที่บอกขั้นตอนหรือวิธีการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ - ความรู้ที่บอกขั้นตอนหรือวิธีการ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ความรู้ที่เกี่ยวกับความรู้ (Meta Knowledge) เป็นความรู้เกี่ยวกับคุณลักษณะของความรู้อื่นหรือเกี่ยวกับวิธีการใช้ความรู้อื่นช่วยในการแก้ปัญหาให้มีประสิทธิภาพมากยิ่งขึ้น ความรู้นี้เปรียบเสมือนประสบการณ์ หรือsenseในการแก้ปัญหาที่มนุษย์เรามีอยู่ความรู้ชนิดนี้เป็นสิ่งที่ Expert Systems จะขาดไม่ได้

แต่ในโลกแห่งความจริงแล้วความรู้ไม่ได้ปรากฏอยู่ในรูปที่สามารถแบ่งออกได้ เป็นกรณีอย่างชัดเจน ความรู้หนึ่งเองเป็นสิ่งที่สำคัญที่ทำให้ผู้เชี่ยวชาญสามารถแก้ไขปัญหายากๆ ได้ โดยจะสร้างแบบจำลองเหตุผล (Causal Models) จากรูปแบบต่างๆ ของความรู้ ซึ่งผู้เชี่ยวชาญจะอาศัยการหาเหตุผลจาก แบบจำลองเหตุผล (Causal Models) เหล่านี้ เพื่อทำนายหาจุดบกพร่อง วางแผนและวิเคราะห์สถานการณ์ต่างๆ เป็นต้น

3.3.2 วิธีการแทนความรู้

วิธีการแทนความรู้ที่ดีจะต้องมีคุณสมบัติดังนี้ [7]

- 1) มีสมรรถภาพในการแทนความรู้ชนิดต่างๆ กล่าวคือ จะต้องสามารถบันทึกความรู้ทั้งที่โครงสร้าง ความรู้ที่ไม่แน่นอน เป็นต้น โดยการใช้โครงสร้างชนิดเดียวกัน และถ้าเป็นไปได้ โครงสร้างที่ใช้ในการแทนความรู้จะต้องเป็นโครงสร้างที่ง่าย แต่มีสมรรถภาพในการแสดงออกสูง
- 2) มี Modularity กล่าวคือ ความสามารถในการแยกออกเป็นส่วน ๆ (Module) ทั้งนี้เพื่อให้สามารถเพิ่มหรือแก้ไขฐานความรู้ได้ง่าย คุณสมบัติอันนี้จำเป็นเพราะทำให้เกิดความยืดหยุ่นในการใช้ฐานความรู้
- 3) ง่ายต่อการจัดการ กล่าวคือ เป็นคุณสมบัติที่ช่วยในการตรวจสอบฐานความรู้ เช่น ช่วยในการตรวจดูความขัดแย้งระหว่างความรู้ การซ้ำกัน หรือความผิดพลาดในความรู้
- 4) ง่ายต่อการเข้าใจของมนุษย์ การแทนความรู้ที่ดีนอกจากจะให้เข้ากับคอมพิวเตอร์แล้ว ยังต้องให้เข้ากับมนุษย์ได้ด้วย คุณสมบัติอันนี้ช่วยในการตรวจสอบหาที่ผิดพลาดในการพิมพ์ความรู้เข้าไปในฐานความรู้
- 5) เข้ากันได้กับการวินิจฉัย เนื่องจากการวินิจฉัยต้องใช้ความรู้ในฐานความรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าเป็นข้อมูล ดึงขึ้น เพื่อให้การวินิจฉัยมีประสิทธิภาพ การแทนความรู้จะต้องเข้าได้กับการวินิจฉัย

3.3.3 รูปแบบการแทนความรู้

รูปแบบการแทนความรู้แบ่ง เป็นหลายชนิดด้วยกันแต่ที่นิยมมากมีอยู่ 3 ชนิด ดังต่อไปนี้

1. การแทนความรู้โดยใช้กฎ [6]

กฎจะใช้แสดงความสัมพันธ์ระหว่างกัน ซึ่งจะอยู่ในรูปของ IF-THEN ส่วน IF เรียกว่าส่วนเงื่อนไขและส่วนของ THEN เรียกว่าส่วนข้อสรุปหรือส่วนการปฏิบัติ

การแทนความรู้โดยใช้กฎ จะประกอบด้วยส่วนย่อยหลัก 3 ส่วนด้วยกัน ดังนี้

- 1) ฐานกฎ (rule base)
- 2) ส่วนตีความ (interpreter)
- 3) working memory (WM)

ฐานกฎเป็นฐานความรู้ที่เก็บความรู้ที่อยู่ในรูปของกฎ working memory เป็นที่เก็บข้อมูลและสถานะของกฎ ข้อมูลและสถานะใน WM เป็น input ของส่วน IF ของกฎ และจะถูกอ้างอิงและเปลี่ยนแปลงโดยฐานกฎ ส่วนตีความจะตรวจดูเนื้อหาของกฎ และ WM แล้วก็เลือกกฎใดกฎหนึ่ง จากชุดของกฎที่มีเงื่อนไขครบขึ้นมาปฏิบัติการ

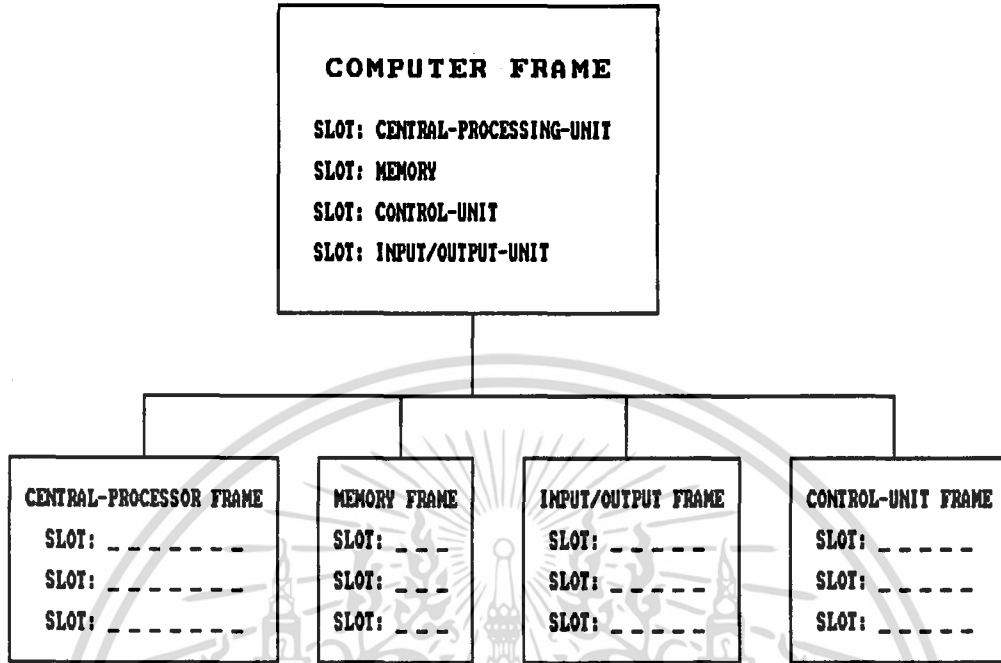
2) การแทนความรู้โดยใช้ข่ายความหมาย (semantics) [12]

จะแทนความสัมพันธ์ระหว่างสิ่งของสองสิ่ง เช่นการใช้กราฟซึ่งประกอบด้วย โหนด (node) และอาร์ค (arc) ใช้แทนความสัมพันธ์ระหว่างโหนด หากต้องการแทนความสัมพันธ์เท่านั้นจะใช้เส้นธรรมดา แต่ถ้าต้องการแทนทิศทางของความสัมพันธ์ จะต้องเพิ่มลูกศรให้ อาร์คด้วย โหนดจะแสดงวัตถุ (Object) สภาพ (state) หรือการยืนยัน (assertion) อาร์คที่แสดงความสัมพันธ์จะมีชื่อต่างๆ เพื่อบอกลักษณะความสัมพันธ์

3) การแทนความรู้โดยใช้กรอบ (frame) [5]

การแทนความรู้โดยใช้กรอบเป็นโครงสร้างแบบจำลองของความจำ และขบวนการเรียนรู้ของมนุษย์ โดยในกรอบจะมีการบันทึกข้อมูลเกี่ยวกับสภาพ เหตุการณ์ วัตถุหรือความคิด และบันทึกความสัมพันธ์ต่างระดับระหว่างสิ่งต่าง ๆ เหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 ตัวอย่างโครงสร้างของระบบกรอบ [5]

โครงสร้างข้อมูลพื้นฐานของกรอบจะประกอบไปด้วย กรอบและชุดของสลอตดัง

รูปที่ 3-3 สลอตนี้อาจเป็นค่าที่ชี้ไปยังกรอบหรือสลอตอื่น หรือเป็นชื่อขั้นตอนการคำนวณหรือโปรแกรมย่อยอื่น ๆ การแทนความรู้โดยผู้ใช้กรอบนี้ จะมีลักษณะ เด่นดังต่อไปนี้

- ก) เป็นการแทนความรู้แบบรวบยอด โดยใช้โครงสร้างข้อมูลชนิดเดียวกัน กรอบแทนความรู้ได้หลายประเภทเหมาะที่จะเป็น module สำหรับการแทนความรู้แบบรวบยอด
- ข) เป็นการแทนความรู้ที่ประหยัด การแทนความรู้ในลักษณะแบ่งเป็นระดับของกรอบนี้ทำให้สามารถเข้าถึงลักษณะการถ่ายทอดคุณสมบัติเพื่อประหยัดเนื้อหาในการเก็บความรู้ได้
- ค) ความยืดหยุ่นในการใช้ความรู้ ในระบบกรอบไม่มีการกำหนดการใช้ความรู้แบบตายตัว

ดังนั้นผู้ออกแบบจึงสามารถเลือกวิธีการใช้ความรู้ให้เหมาะสมกับลักษณะ และสภาพของเอกสารเป็นเอกสารที่ส่งวันวิชาหรือการเชิงในเพื่อการศึกษา ที่เน้น เมื่อนักผู้ใดเห็นไปใช้ประโยชน์ในการแก้ปัญหาได้ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 เทคนิคสำหรับการสร้างระบบผู้เชี่ยวชาญ

ในการพัฒนาระบบผู้เชี่ยวชาญ วิศวกรความรู้จะต้องพิจารณาหัวข้อ 3 ประการ เหล่านี้ก่อนคือ [12]

ก) การเลือกขอบเขตของงานหรือ Domain

ข) การค้นหาหัวข้อเรื่องต่าง ๆ ในขอบเขตงานนั้น

ค) จัดการแทนความรู้ในรูปแบบที่เหมาะสม

คำจำกัดความพื้นฐานอันหนึ่งของระบบผู้เชี่ยวชาญคือ การเข้าถึงจุดหมาย (goals) ได้ โดยผ่านทางข้อมูล, กฎ, การตัดสิ่งที่ไม่เกี่ยวข้องกับจุดหมายออก (Pruning) และกลไกการวินิจฉัย ดังนั้นถ้าเรากำหนดขอบเขตของงานที่สนใจได้แล้ว ก็จะต้องรวบรวมจุดหมายทั้งหมด และข้อมูลทั้งหมดที่เกี่ยวกับขอบเขตเล็กๆของเรานั้น แล้วจึงรวมความรู้ทั้งหมดที่ได้นี้เข้ากับกลไกการวินิจฉัย ซึ่งมีโครงสร้างการ Pruning เพื่อพัฒนาขึ้นเป็นระบบผู้เชี่ยวชาญ

ก) การเลือกขอบเขตงาน

ประการแรกแน่นอนที่งานนั้นจะต้องอยู่ในขอบเขตที่เราสนใจจะทำ ประการต่อไป เราต้องสามารถรวบรวมข้อมูล ในขอบเขตนั้นจำนวนหนึ่ง เข้าไว้ด้วยกันเป็นฐานความรู้ขนาดที่พอเหมาะ คือ ข้อมูลจะต้องมีไม่มาก และกระจัดกระจายจนเกินไป จนไม่สามารถรวบรวมเป็นกลุ่มได้

ข) การค้นหาหัวข้อเรื่องต่างๆในขอบเขตงาน

ในการหาขอบเขตงานจะทำให้การสร้างระบบผู้เชี่ยวชาญเป็นไปได้ง่ายขึ้น ในขั้นแรกอาจจะไปที่ห้องสมุด เพื่อหาหนังสือที่เกี่ยวกับเรื่องที่ต้องการในขณะเดียวกันก็หนีไปถึงผู้เชี่ยวชาญ เพื่อที่จะสามารถให้คำปรึกษากับเราได้ จากการอ่านหนังสือและพบปะกับผู้เชี่ยวชาญ เราจะได้ข้อมูลจำนวนหนึ่งแต่ไม่อยู่ในลักษณะที่รัดกุม เราต้องนำข้อมูลเหล่านั้นมาเลือกหัวข้อเรื่องที่เราสนใจ หรือต้องการสร้างเป็นระบบผู้เชี่ยวชาญ จากนั้นเราควรจะค้นหาว่าเรื่องนั้นให้ละเอียด เพื่อให้ได้ประโยชน์มากที่สุด

ค) จัดการแทนความรู้ในรูปแบบที่เหมาะสม

เมื่อเรามีจุดหมายที่แน่นอนแล้ว ขั้นตอนต่อไปคือจัดโครงสร้างของข้อมูลเหล่านั้นให้เป็นอิสระต่อกัน หนึ่งวิธีขึ้นเพื่อใช้แทนความรู้ตามลักษณะที่เหมาะสม ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ข้อมูลเรียบร้อยแล้วขั้นต่อไปก็ออกแบบ เริ่มด้วยการบรรยายลักษณะของระบบ
ว่าเราต้องการให้ระบบตอบสนองกับผู้ใช้อย่างไร จากนั้นก็เริ่มพัฒนาระบบต้นแบบโดยใช้เครื่องมือ
ช่วยพัฒนาซึ่งอาจจะ เป็นโปรแกรมหรือระบบเปลือกผู้เชี่ยวชาญ และการสร้างฐานความรู้
จากนั้นก็ทดสอบฐานความรู้นี้โดยการนำไป run บนระบบที่ได้สร้างขึ้น

เมื่อได้ระบบเรียบร้อยแล้วแต่ยังไม่สมบูรณ์ ดังนั้นเราต้องทำการ ทดสอบ ปรับปรุง
และขยายระบบจนกว่าจะได้ตามที่เรต้องการ

หลังจากระบบเสร็จสมบูรณ์แล้วก็สามารถนำไปใช้ได้ แต่ในขณะใช้งานวิศวกรรมความรู้
จะต้องเพิ่มเติมฐานความรู้อยู่ตลอดเวลา เพื่อให้ระบบเป็นปัจจุบันมากขึ้น



บทที่ 4

ลักษณะของงานควบคุมปฏิบัติการบิน

ภารกิจอันสำคัญของเจ้าหน้าที่ฝ่ายควบคุมปฏิบัติการบินนั้นคือ ทาหน้าที่ควบคุม ดูแล และประสานงานกับฝ่ายต่างๆ เช่นฝ่ายช่างเทคนิค ฝ่ายลูกเรือ (นักบินและพนักงานบริการบน เครื่อง) ฯลฯ เพื่อให้แต่ละเที่ยวบินสามารถบินได้ตามกำหนดเวลาที่ระบุไว้ในตารางการบินที่ซึ่งกำหนดขึ้นไว้ทุกๆ 6 เดือน และสามารถนำผู้โดยสารไปถึงจุดหมายปลายทางตามกำหนดเวลา ในตารางการบินโดยสะดวกและปลอดภัย ในกรณีที่เกิดปัญหาทำให้เที่ยวบินใดเที่ยวบินหนึ่งที่ไม่สามารถบินได้ตามกำหนดเวลาซึ่งอาจเป็นผลมาจากเครื่องเสีย สภาพอากาศแปรปรวน หรือปัญหาของลูกเรือ ฯลฯ ฝ่ายควบคุมปฏิบัติการบินจะพิจารณาถึงลักษณะของปัญหาที่เกิดขึ้นเพื่อเสนอหนทางในการแก้ปัญหาที่เป็นไปได้ออกมา โดยให้เกิดผลกระทบต่อเที่ยวบินนั้น หรือ เที่ยวบินถัดไปน้อยที่สุด โครงสร้างการทำงานของฝ่ายควบคุมปฏิบัติการบินแสดงดังรูป 4-1

4.1 ทางเลือกปฏิบัติการบินในเหตุการณ์ไม่ปกติ (Irregularity cases)

เมื่อเที่ยวบินใดเกิดปัญหาขึ้นมาจากทางเลือกของฝ่ายปฏิบัติการบินนั้น สามารถ
จำแนกออกเป็นบางส่วนได้ดังนี้

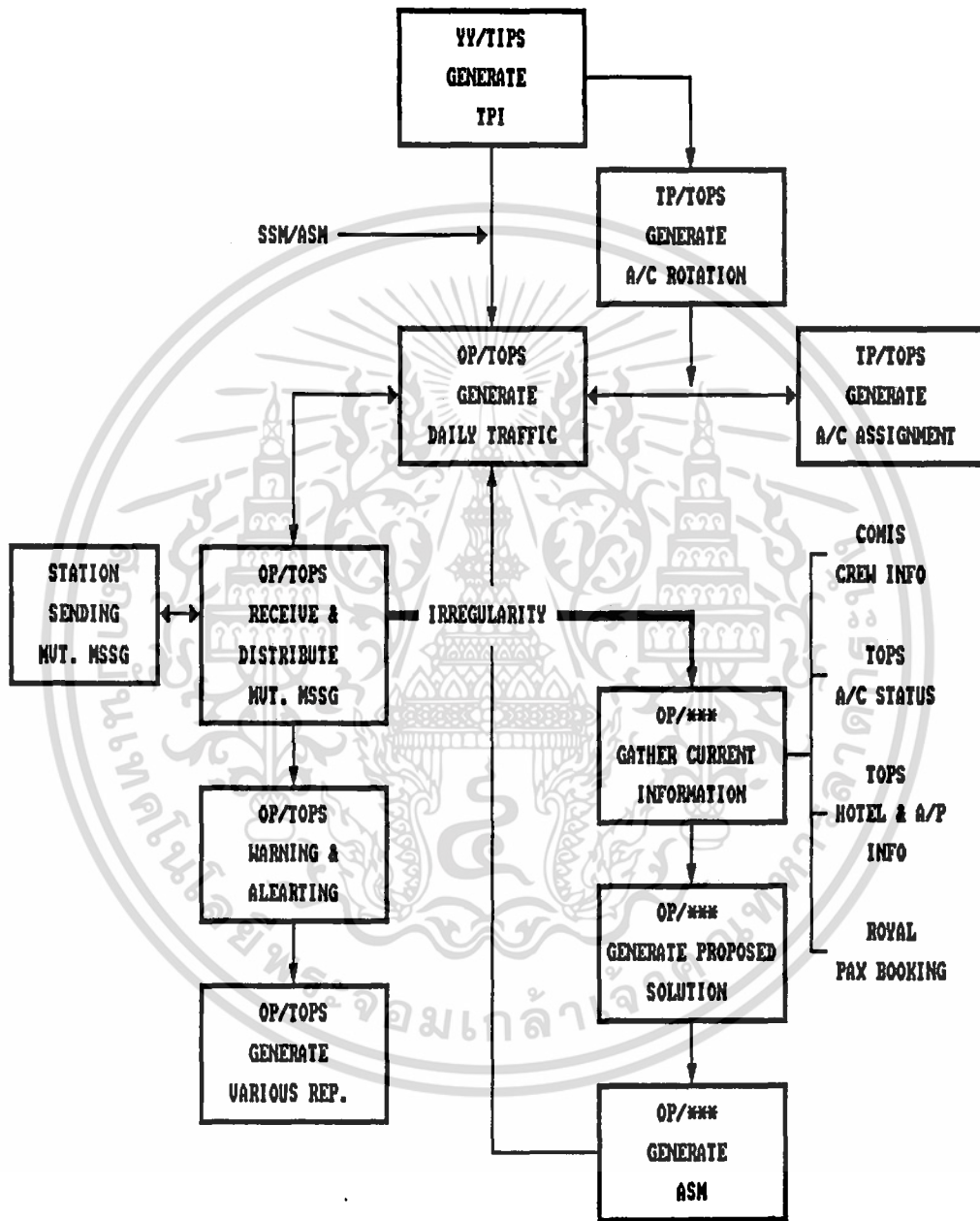
4.1.1 การเลื่อนกำหนดเวลาของเที่ยวบิน (DELAY) เป็นการเลื่อนกำหนด
เวลาของเที่ยวบินนั้นออกไปซึ่งอาจจะ เป็น นาที ชั่วโมง หรือวัน

4.1.2 การเปลี่ยนเส้นทางการบิน (REROUTE) ตามหมายกำหนดการเดิม
เครื่องบินอาจบินจากกรุงเทพฯ ไป อินโดนีเซีย แต่ต้องเปลี่ยนเส้นทางบินอาจต้องไปลงที่สิงคโปร์
หรือมาเลเซีย หรือประเทศอื่นแทน อันเนื่องมาจากเครื่องเสีย สภาพอากาศปลายทาง หรือ
ปัญหาอื่น ๆ

4.1.3 การยกเลิกเที่ยวบิน (CANCELLATION) ในแต่ละเที่ยวบิน (Flight
number) อาจจะแวะลงเพื่อรับส่งผู้โดยสารได้ถึง 2 หรือ 3 จุดหมาย (sectors) ได้แล้ว
แต่กรณี การยกเลิกเที่ยวบินจึงพอจะจำแนกออกเป็น 2 ลักษณะคือ

ยกเลิกบินเป็นเฮลิคอปเตอร์ลงวิ่งวิสาห์หรือบริการเชิงพาณิชย์เพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATIONS CONTROL



รูปที่ 4-1 ลักษณะการทำงานของฝ่ายควบคุมปฏิบัติการบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ก. การยกเลิกเที่ยวบินตลอดเส้นทาง (FLIGHT CANCELLATION) เป็น การยกเลิกทั้งเที่ยวบินไม่ว่าจะแหว่ที่จุดหมายใดบ้าง ตัวอย่างเช่น TG 913 บินจาก ลอนดอนแหว่ลงโคเปนเฮเก้น เพื่อรับส่งผู้โดยสารและ ท้าการบินต่อจากโคเปนเฮเก้น-กรุงเทพฯ (มี 2 SECTOR) การยกเลิก ตลอดเส้นทางก็คือ การไม่ท้าการบินไปในจุดหมายทั้ง 2 แห่งเลย
- ข. การยกเลิกบางส่วนของเที่ยวบิน (SECTOR CANCELLATION) มีลักษณะ คล้ายกับFLIGHT CANCELLATION ที่กล่าวในหัวข้อ ก.แต่จะยกเลิก เพียงบางส่วนเช่น TG913 อาจยกเลิกในส่วนของลอนดอน-โคเปนเฮเก้น หรือในส่วนของ โคเปนเฮเก้น-กรุงเทพฯ

4.1.4 การเปลี่ยนเครื่องบิน (CHANGE OF AIRCRAFT)ลักษณะของการเปลี่ยน เครื่องบินหมายถึงการหาเครื่องบินอื่นที่ว่างอยู่มาบินแทน โดยส่งผลกระทบต่อเที่ยวบินนั้นและ หรือ เที่ยวบินถัดไปให้น้อยที่สุด แบ่งออกเป็น 4 ประเภท คือ

- ก. เปลี่ยนเครื่องบินแบบเดียวกันตามกำหนดเวลา (CHANGE A/C SAME TYPE ONTIME) หมายถึงการหาเครื่องบินแบบเดียวกันที่ว่างอยู่มาบินแทนที่ตรง ตามกำหนดเวลาเดิม
- ข. เปลี่ยนเครื่องบินชนิดเดียวกันแต่เลื่อนกำหนดเวลาบินออกไป (CHANGE A/C SAME TYPE DELAY) หมายถึงการหาเครื่องบินแบบเดียวกันที่ว่างในเวลา ถัดออกไป ซึ่งจะต้องเลื่อนกำหนดเวลาบินออกไปด้วย (DELAY)
- ค. เปลี่ยนเครื่องบินต่างชนิดกันตามกำหนดเวลาเดิม (CHANGE A/C DIFF TYPE ONTIME)
- ง. เปลี่ยนเครื่องบินต่างชนิดกันเลื่อนกำหนดเวลาบินออกไป (CHANGE A/C DIFF TYPE DELAY)

4.2 ส่วนประกอบของการปฏิบัติการบิน (FLIGHT OPERATIONS ENVIRONMENT)

ในการเลือกปฏิบัติการบินของบริษัทการบินต่างๆ ส่วนมากจะต้องคำนึงถึง ความปลอดภัย ความสะดวกสบายของผู้โดยสารและการประหยัดเพื่อให้ได้กำไรเป็นหลัก แต่ปัจจัย อื่นๆเช่นเป็นเอกสารที่ส่งมอบไว้สำหรับกึ่งใช้งานเพื่อการศึกษาเท่านั้น เมื่อนูญัดเห็นว่าไปใช้ประโยชน์ด้านการค้า ต่างๆเหล่านี้ก็น้อยกับส่วนประกอบอื่นที่อีกมากมายที่สามารถเป็นผลของการปฏิบัติการบิน ซึ่ง

อาจจะ เป็นผลกระทบทำให้เที่ยวบินไม่สามารถออกได้ตามกำหนดการ หรืออาจถึงกับต้องยกเลิกเที่ยวบินได้ ปัจจัยเหล่านี้พอจะสรุปคร่าวๆดังนี้

- 4.2.1 เครื่องบิน (AIRCRAFT) สภาพความพร้อมของเครื่องบิน
- 4.2.2 ลูกเรือ (CREW) หมายถึง นักบิน พนักงานต้อนรับบนเครื่องบิน
- 4.2.3 น้ำมันเชื้อเพลิง (FUEL)
- 4.2.4 การอนุญาตในการบิน (PERMIT) หมายถึง การขออนุญาตบินผ่านน่านฟ้า หรือลงจอดรับส่งผู้โดยสารแต่ละประเทศ
- 4.2.5 แผนการบิน/ความสูง/เวลาเข้าออก (A/P AND ATC SLOT CLEARANCE)
- 4.2.6 เครื่องอำนวยความสะดวกของสนามบิน (A/P FACILITY) คือสนามบินนั้นต้องมีอุปกรณ์อำนวยความสะดวกให้แก่เครื่องบิน และผู้โดยสาร
- 4.2.7 ช่วงเวลาปิด/เปิดสนามบิน (A/P CURFEW) บางสนามบินอาจมีการปิด/เปิดสนามบินในเวลาากลางคืนหรือในเวลาที่อากาศแปรปรวน
- 4.2.8 รันเวย์ (RUNWAY) ลานขึ้นลงของเครื่องบินจะต้องสมบูรณ์เนื่องจากขนาดของเครื่องบินแต่ละเครื่องมีขนาดไม่เท่ากัน จึงต้องพิจารณาถึงความพร้อมของทางขึ้นลง และอุปกรณ์ในการขึ้นลง
- 4.2.9 อากาศ (WEATHER) สภาพอากาศตามเส้นทางหรือสนามบินพร้อมที่จะให้เครื่องบินผ่าน หรือขึ้นลงได้อย่างปลอดภัย
- 4.2.10 ปัญหาทางการเมือง (POLITICS) จะต้องพิจารณาด้วยว่าในขณะที่นั้นมีปัญหาทางการเมือง หรือความสัมพันธ์ทางการเมือง การทูต ฯลฯ
- 4.2.11 การไม่ปฏิบัติตามที่ของพนักงานที่เกี่ยวข้องกับการปฏิบัติการบิน (INDUSTRIAL DISPUTE) สามารถส่งผลกระทบต่อการบินได้ เช่นการนัดหยุดงานของพนักงานขับรถน้ำมัน หรือพนักงานคันสาธารณะ ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONSTRAINT	AIRCRAFT	CREW	FUEL	PERMIT	A/P SLOT	A/P FACILITY	A/P CURFEW	RUNWAY	WEATHER	POLITICS	INDUSTRY
ACTION											
1. CANCEL FLIGHT	-	-	-	-	-	-	-	-	-	-	-
2. CANCEL SECTOR	-	-	-	-	-	-	-	-	-	-	-
3. DELAY	AC1	CU1	FU1	-	SL1	FC1	CU1	RW1	WX1	PO1	ID1
4. REROUTE	AC2	CU1	FU1	PM1	SL1	FC1	CU1	RW1	WX1	PO1	ID1
5. CHANGE A/C SAME TYPE ONTIME	AC3	-	-	-	-	-	-	-	-	-	-
6. CHANGE A/C SAME TYPE DELAY	AC3	CU1	FU1	-	SL1	FC1	CU1	RW1	WX1	PO1	ID1
7. CHANGE A/C DIFF TYPE ONTIME	AC3	CU2	-	PM2	SL1	FC1	-	RW1	-	-	-
8. CHANGE A/C DIFF TYPE DELAY	AC3	CU2	FU1	PM2	SL1	FC1	CU1	RW1	WX1	PO1	ID1

ตารางที่ 4-1 ส่วนประกอบของการปฏิบัติการบิน

4.3 ผลกระทบของการเลือกปฏิบัติการบิน (FLIGHT OPERATION FACTOR)

เมื่อได้ทำการพิจารณาข้อบังคับของการเลือกปฏิบัติการบิน และสิ่งที่ต้องพิจารณาในระดับขั้นต่อมาก็คือ ผลกระทบต่อเครื่องบินนั้นโดยต้องทำให้เที่ยวบินนั้นเกิดผลกระทบให้น้อยที่สุด ซึ่งมิได้หลายกรณีพอสรุปได้เป็นหัวข้อสำคัญๆ ดังนี้

4.3.1 ค่าใช้จ่ายของเที่ยวบินนั้น (COST) เป็นค่าใช้จ่ายทั้งหมดซึ่งรวมถึงค่าใช้จ่ายที่ต้องเสียไปเพื่ออำนวยความสะดวกสบายให้กับผู้โดยสาร และค่าใช้จ่ายอื่นๆ ต่อเที่ยวบินนั้น

4.3.2 ความสะดวกสบายของผู้โดยสาร (PAX. CONVENIENT)

4.3.3 ผู้โดยสารคนสำคัญ (VIP)

4.3.4 ผลกระทบของเที่ยวบินถัดไป (NEXT FLIGHT or NEXT SECTOR)

4.3.5 ภาพพจน์ที่ติดต่อสายการบิน/การแข่งขันระหว่างสายการบิน (IMAGE/COMPETITION/SALES)

ที่กล่าวมาทั้งหมดนี้เป็นการปฏิบัติการบินของฝ่ายควบคุมปฏิบัติการบิน เมื่อเกิดเหตุการณ์ไม่ปกติต่อเที่ยวบินนั้น

IMPACT FACTOR ACTION	COST	PAX. CONV.	VIP	NXT. FLT. OR NXT. SECT.	COMPETITION
1. CANCEL FLIGHT	C1	P1	U1	F1	I1
2. CANCEL SECTOR	C1	P1	U1	F2	I1
3. DELAY	C2	P1	U1	F2	I1
4. REROUTE	C3	P2	U1	F3	I1
5. CHANGE A/C SAME TYPE ONTIME	C4	-	-	-	I1
6. CHANGE A/C SAME TYPE DELAY	C2	P1	U1	-	I1
7. CHANGE A/C DIFF TYPE ONTIME	C5	P1	U1	-	I1
8. CHANGE A/C DIFF TYPE DELAY	C6	P1	U1	-	I1

ตารางที่ 4-2 ผลกระทบของทางเลือกปฏิบัติการบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โครงสร้างและการทํางานของโปรแกรม

ในบทที่ 3 ได้กล่าวถึงทฤษฎีของระบบผู้เชี่ยวชาญ และบทที่ 4 ก็เป็นงานบางส่วน ของฝ่ายควบคุมปฏิบัติการบินที่นำมาประยุกต์ใช้กับระบบผู้เชี่ยวชาญ สำหรับบทนี้จะเป็นการรวม กันของทฤษฎีและการควบคุมปฏิบัติการบิน ซึ่งวิทยานิพนธ์นี้ได้ใช้ภาษา LISP (List-Processing) พัฒนาระบบขึ้นมา

5.1 ขั้นตอนการสร้างระบบ

ในการสร้างระบบผู้เชี่ยวชาญ เพื่อควบคุมปฏิบัติการบินนี้ได้มีการแบ่งการดำเนินงาน ออกเป็นขั้นตอนต่าง ๆ ดังนี้

ขั้นตอนที่ 1 ทําการศึกษารายละเอียดต่าง ๆ แบ่งออกเป็น

- ศึกษาทฤษฎีเกี่ยวกับระบบผู้เชี่ยวชาญ
- ศึกษารายละเอียดของงานควบคุมปฏิบัติการบิน
- กำหนดขอบเขตของงานที่เหมาะสม
- ทําการกำหนดวิธีการและ เลือก เครื่องมือที่ใช้ในการสร้างระบบฐานความรู้

ขั้นตอนที่ 2 พัฒนาระบบต้นแบบ

- กำหนดความรู้ที่จะใช้ในการสร้างระบบต้นแบบ
- ทําการจัดรูปแบบของความรู้ที่ได้ศึกษามา
- พัฒนาระบบฐานความรู้ที่ได้สร้าง เริ่มแรก
- ค้นหาปัญหาที่เกิดระหว่างการสร้างระบบต้นแบบ

ขั้นตอนที่ 3 การพัฒนาระบบฐานความรู้ให้สมบูรณ์

- ทําการปรับปรุง และ แก้ไขปัญหาที่เกิดขึ้นในระหว่างการพัฒนาระบบต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิได้อนุญาตให้เผยแพร่หรือใช้ใน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไข ปรับปรุง ส่วนติดต่อกับผู้ใช้ให้เหมาะสม

ขั้นตอนที่ 4 การประเมินผลระบบ โดยทำการทดสอบการทํางานของระบบให้กับ
ผู้ใช้เป็นครั้งสุดท้าย

5.2 การจัดรูปแบบความรู้ของการควบคุมปฏิบัติการบิน

เนื่องจากข้อมูลหรือความรู้เกี่ยวกับการควบคุมปฏิบัติการบินนั้น เป็นไปในลักษณะที่มีความสัมพันธ์กัน เพื่อเป็นการเพิ่มประสิทธิภาพและประหยัดเนื้อที่ในหน่วยความจำ และคุณสมบัติที่ดีของระบบรอบเกี่ยวกับความยืดหยุ่นของการใช้ความรู้ ในที่นี้จึงได้นำเอาวิธีการใช้ความรู้แบบกฎหรือเงื่อนไขร่วมกันด้วย

5.2.1 ลำดับชั้นของกรอบ (Hierarchies of frames)

ลักษณะของงานทางด้านปฏิบัติการบินที่กล่าวมาในบทที่ 4 สามารถจัดอยู่ในลำดับชั้นของกรอบ (frame) ได้ดังรูปที่ 5-1

ระดับบนสุดของรูบนี้แสดงถึงส่วนประกอบที่สำคัญของทางเลือกในการปฏิบัติการบิน เมื่อเกิดเหตุการณ์ไม่ปกติ ในกรอบบนสุด (operating action frame) ส่วนที่เป็น สล็อต (slot) เปรียบเสมือนกรอบย่อย (subframe) ส่วนกรอบครั้งที่ 2 และ 3 เป็นส่วนประกอบพื้นฐานของกรอบบนสุด (operating action frame) และส่วนประกอบของสล็อตในกรอบระดับ 2 และ 3 จะประกอบไปด้วยเงื่อนไขต่างๆที่อธิบายในหัวข้อถัดไป

5.2.2 ส่วนประกอบและผลกระทบของทางเลือกปฏิบัติการบิน

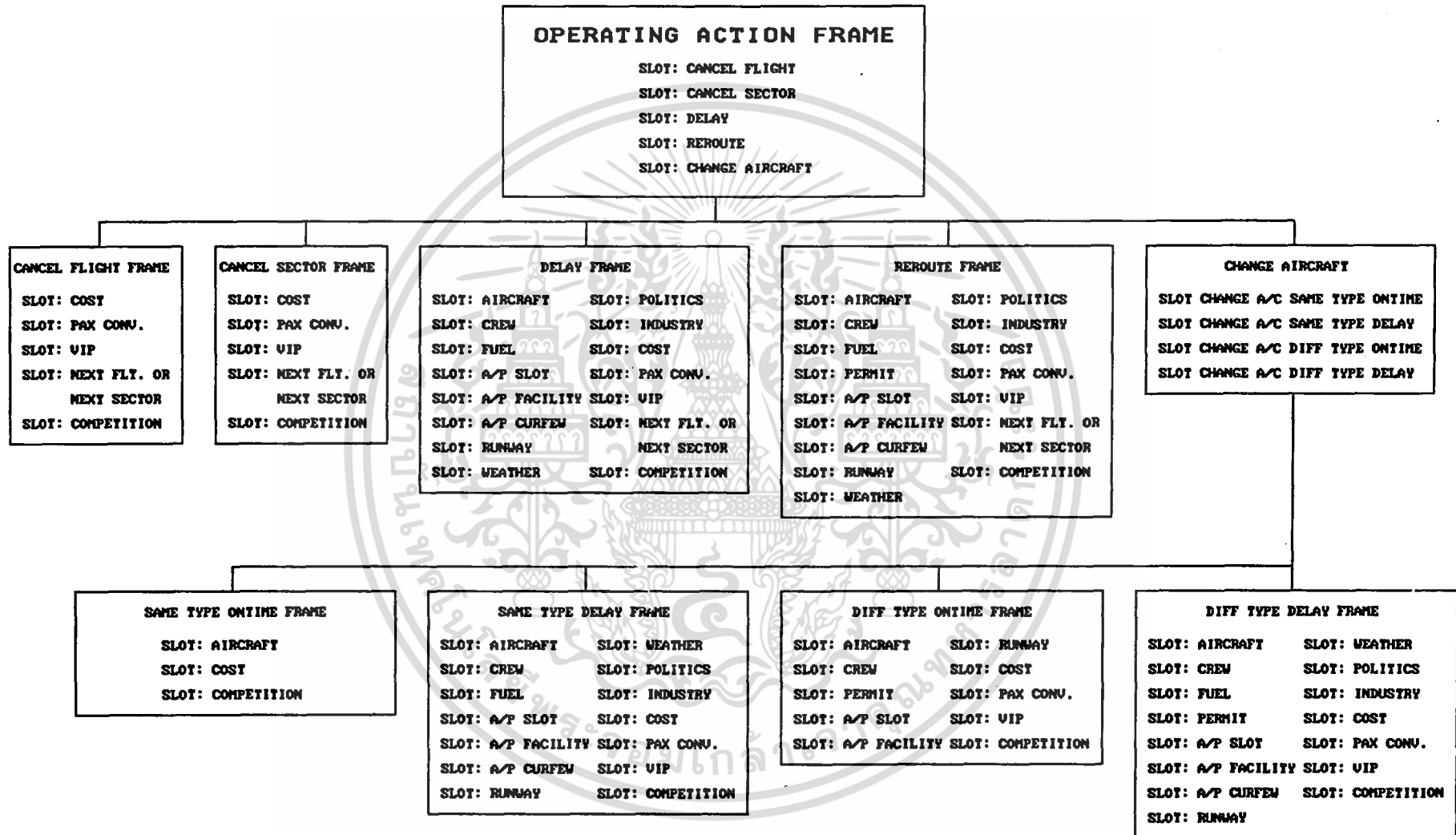
ในหัวข้อนี้จะกล่าวถึง เงื่อนไขต่างๆ ของส่วนประกอบและผลกระทบของการเลือกปฏิบัติการบินที่ระบบนำมาใช้ เพื่อแสดงข้อมูลสนับสนุนการตัดสินใจของผู้เชี่ยวชาญ

5.2.2.1 เงื่อนไขของส่วนประกอบทางเลือกปฏิบัติการบิน นั้นเป็นเงื่อนไขที่

สร้างให้ระบบเพื่อระบบนำค่าต่างๆ จากฐานข้อมูลมาเช็คเงื่อนไขเหล่านี้ และถ้าเงื่อนไขใดเป็นจริงก็จะแสดงข้อมูลออกมา

- เครื่องบิน (AIRCRAFT) แบ่งเงื่อนไขออกเป็น 3 ชุด คือ AC1; AC2; AC3;
AC1 เป็นเงื่อนไขที่ใช้ในทางเลือก การเลื่อนกำหนดเวลาของเที่ยวบินจะประกอบไปด้วย

เอกสารนี้ (1) ถ้า - เครื่องบินไม่มีปัญหาทางเทคนิคต่าง ๆ (ไม่ต้องตรวจสอบ)
ไม่ว่าการณีใดๆ แล้ว - อีกทีเครื่องบินพร้อมที่จะบิน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- (2) ถ้า - เครื่องบินมีปัญหาทางเทคนิคต่าง ๆ (ต้องตรวจซ่อม)
 - เวลาที่จะออกหลังจาก Delay แล้วไม่อยู่ในช่วง เวลาตรวจเช็คแล้ว - เครื่องพร้อมที่จะบิน
- (3) ถ้า - เครื่องบินมีปัญหาทางเทคนิคต่างๆ (ต้องตรวจซ่อม)
 - เวลาที่จะออกหลัง Delay อยู่ในช่วง เวลาตรวจเช็คแล้ว - เครื่องไม่พร้อมที่จะบิน

AC2 เป็นเงื่อนไขที่ต้องนำมาพิจารณา เมื่อต้องการเปลี่ยนเส้นทางบิน

- ทหารยะทางบินของ เส้นทางบินใหม่
- (1) ถ้า - ระยะทางบินหรือชม.บินใหม่ \leq ความสามารถทางการบินของ เครื่องลานั้น
- แล้ว - เครื่องพร้อมที่จะบินได้
- (2) ถ้า - ระยะทางบินหรือชม.บินใหม่ $>$ ความสามารถทางการบินของ เครื่องลานั้น
- แล้ว - เครื่องไม่พร้อมจะบิน

AC3 ในส่วนนี้จะนำมาพิจารณานกรณี เปลี่ยนเครื่องบิน

- (1) ถ้า - เครื่องบินมีปัญหาทางเทคนิค
- แล้ว - เครื่องบินมีปัญหาทางเทคนิค
- (2) ถ้า - เครื่องบินไม่มีปัญหาทางเทคนิค
- เวลาที่เครื่องออกไม่อยู่ในช่วง เวลาตรวจเช็ค
- แล้ว - เครื่องไม่มีปัญหาทางเทคนิค
- (3) ถ้า - เครื่องไม่มีปัญหาทางเทคนิค
- เวลาที่เครื่องออกอยู่ในช่วง เวลาตรวจเช็ค
- แล้ว - เครื่องมีปัญหาทางเทคนิค
- (4) ถ้า - เวลาที่มาถึงของ เครื่องที่จะนำมาเปลี่ยน + เวลาการเช็คเครื่อง (GST) $>$ เวลาออกของ เครื่องที่ต้องการเปลี่ยน
- แล้ว - ไม่มีเครื่องว่างที่จะบินแทน

หมายเหตุ *** สัญลักษณ์ AC1; FC1 ฯลฯ นั้นเป็นไฟล์ชาร์ทที่แสดงให้เห็นในภาคผนวก 3 ซึ่งเป็นการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีซีเอ็มให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ท้ายเล่ม

- (5) ถ้า - เวลาที่มาถึงของเครื่องที่จะนำมาเปลี่ยน + เวลาการเช็คเครื่อง (GST) \leq เวลาออกของเครื่องที่ต้องการเปลี่ยน
- เวลาที่กลับมาของเครื่องที่นำมาเปลี่ยน + GST \leq เวลาที่ออกใน sector หรือ เที่ยวบิน (flight) ถัดไป
แล้ว - เครื่องพร้อมที่จะบินแทน
- (6) ถ้า - เวลาที่มาถึงของเครื่องที่จะนำมาเปลี่ยน + เวลาการเช็คเครื่อง (GST) \leq เวลาออกของเครื่องที่ต้องการเปลี่ยน
- เวลาที่กลับมาของเครื่องที่นำมาเปลี่ยน + GST $>$ เวลาที่ออกใน sector หรือ flight ถัดไป
แล้ว - ไม่สามารถนำมาบินแทนได้
- (7) ถ้า - เป็นเครื่องชนิดเดียวกัน
แล้ว - แสดงผลในข้างต้น
- (8) ถ้า - เป็นเครื่องต่างชนิดกัน
- ระยะทางบินใหม่ $>$ ความสามารถของเครื่องบินที่จะเปลี่ยน
แล้ว - เครื่องไม่พร้อมที่จะบิน
- (9) ถ้า - เป็นเครื่องบินต่างชนิดกัน
- ระยะทางบินใหม่ \leq ความสามารถของเครื่องบินที่จะเปลี่ยน
แล้ว - เครื่องพร้อมที่จะบินได้

- ลูกเรือ (CREW) รวมทั้งนักบิน และพนักงานต้อนรับ แบ่งเงื่อนไขเป็น CW1 และ CW2 ดังนี้

CW1 เป็นเงื่อนไขในทางเลือก delay; reroute; เปลี่ยนเครื่องบินแล้วเลื่อนเวลาบิน

- (1) ถ้า - มีลูกเรือเตรียมแล้ว (assigned crew available)
- ค่า index ≤ 100

แล้ว - ลูกเรือพร้อมที่จะปฏิบัติหน้าที่

- (2) ถ้า - มีลูกเรือเตรียมแล้ว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งค่า index > 100 มีโทษ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีลูกเรือสำรอง (available)
- มีลูกเรือที่จะไปกับเครื่อง (qualified)
- แล้ว - ลูกเรือพร้อมที่จะปฏิบัติหน้าที่
- (3) ถ้า - มีลูกเรือเตรียมแล้ว
 - ค่า index > 100
 - มีลูกเรือสำรอง (standby)
 - ไม่มีลูกเรือที่จะไปกับเครื่อง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (4) ถ้า - มีลูกเรือเตรียมแล้ว
 - ค่า index > 100
 - ไม่มีลูกเรือสำรอง
 - เวลาที่ลูกเรือไม่พร้อมตรงกับเวลาเครื่องออกเดินทาง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (5) ถ้า - มีลูกเรือเตรียมแล้ว
 - ค่า index > 100
 - ไม่มีลูกเรือสำรอง
 - เวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาเครื่องออกเดินทาง
 - มีลูกเรือที่จะไปกับเครื่อง (qualified)
- แล้ว - ลูกเรือพร้อมที่จะปฏิบัติหน้าที่
- (6) ถ้า - มีลูกเรือเตรียมแล้ว
 - ค่า index ของลูกเรือ > 100
 - ไม่มีลูกเรือสำรอง
 - เวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาเครื่องออกเดินทาง
 - ไม่มีลูกเรือที่จะไปกับเครื่อง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (7) ถ้า - ไม่มีลูกเรือเตรียมล่วงหน้า (assigned not available)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีสิทธิเปลี่ยนแปลงโดยไม่ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีลูกเรือที่จะไปกับเครื่อง (qualified)
- แล้ว - ลูกเรือพร้อมที่จะปฏิบัติหน้าที่
- (8) ถ้า - ไม่มีลูกเรือเตรียมล่วงหน้า
- มีลูกเรือสำรอง
- ไม่มีลูกเรือที่จะไปกับเครื่อง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (9) ถ้า - ไม่มีลูกเรือเตรียมล่วงหน้า
- ไม่มีลูกเรือสำรอง
- เวลาที่ลูกเรือไม่พร้อมตรงกับเวลาที่เครื่องออกเดินทาง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (10) ถ้า - ไม่มีลูกเรือเตรียมล่วงหน้า
- ไม่มีลูกเรือสำรอง
- มีลูกเรือที่จะไปกับเครื่อง
- เวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาที่เครื่องออกเดินทาง
- แล้ว - ลูกเรือพร้อมที่จะปฏิบัติหน้าที่
- (11) ถ้า - ไม่มีลูกเรือเตรียมล่วงหน้า
- ไม่มีลูกเรือสำรอง
- ไม่มีลูกเรือที่จะไปกับเครื่อง
- เวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาที่เครื่องออกเดินทาง
- แล้ว - ลูกเรือไม่พร้อมที่จะปฏิบัติหน้าที่

CW2 จะใช้ในกรณีเปลี่ยนเครื่องบินต่างชนิดกันทั้งตามเวลาเดิมและเลื่อนเวลาออกบินมีเงื่อนไขดังนี้

- (1) ถ้า - มีลูกเรือสำรอง
- มีลูกเรือพร้อมที่จะบินกับเครื่อง
- แล้ว - มีลูกเรือพร้อมที่จะปฏิบัติหน้าที่

เอกสาร (2) ถ้าการส่งมีลูกเรือสำรองใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งผู้ดำเนินการต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่มีลูกเรือพร้อมที่จะบินกับเครื่อง

- แล้ว - ลุกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (3) ถ้า - ไม่มีลูกเรือสำรอง
- ช่วงเวลาที่ลูกเรือไม่พร้อมอยู่ในเวลาของเครื่องออกเดินทาง
- แล้ว - ลุกเรือไม่พร้อมที่จะปฏิบัติหน้าที่
- (4) ถ้า - ไม่มีลูกเรือสำรอง
- ช่วงเวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาเครื่องออกเดินทาง
 - มีลูกเรือพร้อมที่จะบินกับเครื่อง

- แล้ว - ลุกเรือพร้อมที่จะปฏิบัติหน้าที่
- (5) ถ้า - ไม่มีลูกเรือสำรอง
- ช่วงเวลาที่ลูกเรือไม่พร้อมไม่ตรงกับเวลาเครื่องออกเดินทาง
 - ไม่มีลูกเรือพร้อมที่จะบินกับเครื่อง
- แล้ว - ลุกเรือไม่พร้อมที่จะปฏิบัติหน้าที่

- ปัญหาเกี่ยวกับน้ำมันเชื้อเพลิง (FUEL) เงื่อนไขนี้ต้องใช้ในการเลือกปฏิบัติการเลื่อนเวลาการเดินทาง; เปลี่ยนเส้นทางบิน; เปลี่ยนเครื่องบินแบบเดียวกันแต่เลื่อนเวลาบิน; เปลี่ยนเครื่องต่างชนิดแต่เลื่อนเวลาบินออกไป มีเงื่อนไขคือ

FU1

- (1) ถ้า - ไม่มีปัญหาเกี่ยวกับสถานะการดำเนินงานที่สนามบินนั้น
- แล้ว - ไม่มีปัญหาทางน้ำมัน
- (2) ถ้า - มีปัญหาเกี่ยวกับน้ำมันที่สนามบินนั้น
- ช่วงเวลา TRANSIT หรือ TURN AROUND อยู่ในช่วงเวลาที่ปัญหาทางน้ำมัน
- แล้ว - มีปัญหาทางน้ำมัน
- (3) ถ้า - มีปัญหาเกี่ยวกับน้ำมันที่สนามบินนั้น
- ช่วงเวลา TRANSIT หรือ TURN AROUND ไม่อยู่ในช่วงเวลานั้น
- แล้ว - ไม่มีปัญหาทางน้ำมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การอนุญาตในการบิน (PERMIT) แบ่งออกเป็น 2 ส่วน PM1 นั้นใช้เฉพาะการเปลี่ยนเส้นทางบิน ส่วน PM2 นั้น ใช้เป็นเงื่อนไขเมื่อเลือกการเปลี่ยนเครื่องบินต่างชนิดที่เลื่อนเวลาบิน และบินตามเวลาเดิม

PM1 มีเงื่อนไข ดังนี้

- (1) ถ้า - มีการขออนุญาตไว้สำหรับสนามบินใหม่ที่ต้องการเปลี่ยนเส้นทางบิน
แล้ว - ไม่มีปัญหาในการอนุญาต
- (2) ถ้า - ไม่มีการขออนุญาตไว้สำหรับสนามบินใหม่ที่ต้องการเปลี่ยนเส้นทางบิน
แล้ว - มีปัญหาในการอนุญาต

PM2 มีเงื่อนไขต่อไปนี้

- (1) ถ้า - การอนุญาตของสนามบินที่จะนำเครื่องต่างชนิดไปลง
แล้ว - อนุญาตให้เครื่องลงได้
- (2) ถ้า - ไม่ได้รับอนุญาตสำหรับนำเครื่องต่างชนิดไปลง
แล้ว - ไม่อนุญาตให้เครื่องลง

- แผนการบิน/ความสูง/เวลาเข้าออก (AIRPORT SLOT) เงื่อนไขนี้จะถูกพิจารณาในกรณีเลื่อนเวลาบิน; เปลี่ยนเส้นทางบิน; เปลี่ยนเครื่องบินแบบเดียวกันแต่เลื่อนเวลา; เปลี่ยนเครื่องต่างชนิดกันทั้งตรงเวลาและเลื่อนเวลา

- SL1
- (1) ถ้า - แผนการบิน/ความสูง/เวลาเข้าออก ได้ถูกเตรียมแล้ว
แล้ว - ไม่มีปัญหาทางแผนการบิน/ความสูง/เวลาเข้าออก
 - (2) ถ้า - แผนการบิน/ความสูง/เวลาเข้าออกไม่ได้ถูกเตรียมไว้
- อยู่ในช่วงของการ TRANSIT หรือ TURN AROUND
แล้ว - มีปัญหาทางแผนการบิน/ความสูง/เวลาเข้าออก
 - (3) ถ้า - แผนการบิน/ความสูง/เวลาเข้าออกไม่ได้ถูกเตรียมไว้
- ไม่อยู่ในช่วงของการ TRANSIT หรือ TURN AROUND

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของกรมเจ้าท่า กระทรวงคมนาคม
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องอำนวยความสะดวกของสนามบิน (A/P FACILITY) จะพิจารณาทุก
ทางเลือก แต่ที่ไม่นับมาใช้คือ การเลิกเที่ยวบิน และการเปลี่ยนเครื่องบินเดียวกันตรง
ตามเวลา

- FC1 (1) ถ้า - มีเครื่องอำนวยความสะดวกของสนามบิน
แล้ว - ไม่มีปัญหาทางความสะดวก (facility)
- (2) ถ้า - ไม่มีเครื่องอำนวยความสะดวก
- ไม่อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - ไม่มีปัญหาทางความสะดวก (FACILITY)
- (3) ถ้า - ไม่มีเครื่องอำนวยความสะดวกของสนามบิน
- อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - มีปัญหาทางความสะดวก (FACILITY)

- เคอร์ฟิว (CURFEW) มี CU1 ใช้ในทางเลือกเดียวกับ FACILITY

- CU1 (1) ถ้า - มีเคอร์ฟิวที่สนามบิน
- ไม่อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - ไม่มีปัญหาทางเคอร์ฟิว
- (2) ถ้า - ไม่มีเคอร์ฟิวที่สนามบิน
แล้ว - ไม่มีปัญหาทางเคอร์ฟิว
- (3) ถ้า - มีเคอร์ฟิวที่สนามบิน
- อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - มีปัญหาทางเคอร์ฟิว

- รันเวย์ (RUNWAY) เป็นเงื่อนไขที่ต้องพิจารณาในทางเลือก เลื่อนเวลาการบิน; เปลี่ยน
เส้นทางบิน; เปลี่ยนเครื่องบินเดียวกันแต่เลื่อนเวลาบิน ; เปลี่ยนเครื่องต่างชนิดกันทั้งตามเวลา
เอ เดิม และ เลื่อนเวลาบินไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RW1 (1) ถ้า - รันเวย์พร้อมทุกอย่าง
แล้ว - ไม่มีปัญหาเกี่ยวกับรันเวย์
- (2) ถ้า - รันเวย์ไม่พร้อม
- เวลาปิดอยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - รันเวย์มีปัญหา
- (3) ถ้า - รันเวย์ไม่พร้อม
- เวลาปิดไม่อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - รันเวย์ไม่มีปัญหา

- สภาพอากาศ (WEATHER) ใช้กับกรณี เปลี่ยนเส้นทางบิน; เลื่อนเวลาบิน ; เปลี่ยน
เครื่องบินเดียวกันเลื่อนเวลาบิน; เปลี่ยนเครื่องต่างชนิดกันเลื่อนเวลาบิน ในส่วนนี้จะเป็นข้อมูล
ที่อยู่ในฐานข้อมูลของสนามบิน จะมีค่าตั้งแต่ 1-9 ดังนี้

WX1

ค่าของ WX	สถานะ
1	ปิด - ปิด
2	เมอร์จเจอร์ - ปิด
3	เปิด - ปิด
4	ปิด - เมอร์จเจอร์
5	ปิด - เปิด
6	เมอร์จเจอร์ - เมอร์จเจอร์
7	เปิด - เมอร์จเจอร์
8	เมอร์จเจอร์ - เปิด
9	เปิด - เปิด

เอกสารนี้ - ปัญหาทางการเมือง (POLITICAL) จะถูกพิจารณาทางเลือกเหมือนกับอากาศ
มีเงื่อนไขดังนี้ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PO1 (1) ถ้า - ไม่มีปัญหาเกี่ยวกับการเมืองที่สนามบินนั้น
แล้ว - ไม่มีปัญหาทางการเมือง
- (2) ถ้า - มีปัญหาทางการเมือง ที่สนามบินนั้น
- ไม่อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - ไม่มีปัญหาทางการเมือง
- (3) ถ้า - มีปัญหาทางการเมือง ของสนามบินนั้น
- อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - มีปัญหาทางการเมือง
- การไม่ปฏิบัติหน้าที่ของพนักงานที่เกี่ยวข้องกับการบิน (INDUSTRIAL DISPUTE) จะมีการพิจารณาเงื่อนไขเหมือนกับ ทางการเมืองและอากาศ มีเงื่อนไขดังต่อไปนี้

ID1

- (1) ถ้า - ไม่มีปัญหาเกี่ยวกับการปฏิบัติหน้าที่ของพนักงานที่เกี่ยวข้องกับการบิน
แล้ว - ไม่มีปัญหาทาง INDUSTRIAL DISPUTE
- (2) ถ้า - มีปัญหาเกี่ยวกับการปฏิบัติหน้าที่ของพนักงานที่เกี่ยวข้องกับการบิน
- ไม่อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - ไม่มีปัญหา INDUSTRIAL DISPUTE
- (3) ถ้า - มีปัญหาการปฏิบัติหน้าที่ของพนักงานที่เกี่ยวข้องกับการบิน
- อยู่ในช่วงเวลา TRANSIT หรือ TURN AROUND
แล้ว - มีปัญหา INDUSTRIAL DISPUTE

ที่กล่าวมาแล้วนั้นเป็นส่วนประกอบของทางเลือกในการปฏิบัติการบิน ซึ่งทั้งหมดเป็นเพียงหลักสำคัญ ๆ ที่ใช้ในระบบผู้เชี่ยวชาญ ส่วนในการปฏิบัติการบินอาจมีเพิ่มขึ้นอีกเล็กน้อย ในหัวข้อถัดไปจะกล่าวถึงผลกระทบของทางเลือกในการปฏิบัติการบิน

5.2.2.2 เงื่อนไขของผลกระทบทาง เลิกปฏิบัติการบิน มีหัวข้อใหญ่ที่

สามารถแบ่งออกได้ พอสังเขปดังนี้

- ค่าใช้จ่าย (COST) แยกออกได้เป็น 6 ส่วน มีเงื่อนไขที่ต่างกันออกไป ตามลำดับดังนี้
C1 เป็นค่าใช้จ่ายสำหรับกรณียกเลิกเที่ยวบิน (CANCEL) จะมีการคำนวณค่าใช้จ่ายโดยตรง
และระยะทางหรือชม.บิน เพื่อแสดงข้อมูลในทศ ๆ กรณี มีเงื่อนไขคือ

- (1) ถ้า - มีเครื่องบินที่จะบินไปทางเดียวกัน
- สามารถถ่ายจำนวนผู้โดยสารได้หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
แล้ว แสดงค่าใช้จ่ายโดยตรง ; แสดงระยะทางบิน
- (2) ถ้า - มีเครื่องบินที่จะบินไปทางเดียวกัน
- สามารถถ่ายจำนวนผู้โดยสารได้หมด
- ถ่ายผู้โดยสาร > 4 ชั่วโมง
แล้ว คำนวณค่าใช้จ่ายในการจัดหาที่พักและอาหารของผู้โดยสาร (2500 b/p/n)
- (3) ถ้า - มีเครื่องบินที่จะบินไปทางเดียวกัน
- ไม่สามารถถ่ายจำนวนผู้โดยสารได้หมด
แล้ว คำนวณระยะทางบิน; คำนวณจำนวนผู้โดยสารที่เหลือ : คำนวณรายได้
ที่ต้องสูญเสีย : คำนวณค่าจ้างเหมาเครื่องบิน
- (4) ถ้า - มีเครื่องบินที่จะบินไปทางเดียวกัน
- ไม่สามารถถ่ายจำนวนผู้โดยสารได้หมด
- ถ่ายผู้โดยสาร > 4 ชั่วโมง
แล้ว คำนวณระยะทางบิน; คำนวณจำนวนผู้โดยสารที่เหลือ : คำนวณรายได้
ที่ต้องสูญเสีย : คำนวณค่าจ้างเหมาเครื่องบิน และ คำนวณค่าใช้จ่ายใน
การจัดหาที่พักและอาหารของผู้โดยสาร (2500 b/p/n)
- (5) ถ้า - มีเครื่องบินที่จะบินไปทางเดียวกัน

เอกสารนี้เป็นเอกสารที่ส่ง ไม่สามารถถ่ายจำนวนผู้โดยสารได้หมดนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

- แล้ว คำนวนระยะทางบิน; คำนวนจำนวนผู้โดยสารที่เหลือ : คำนวนรายได้
ที่ต้องสูญเสีย : คำนวนค่าจ้างเหมาเครื่องบิน; คำนวนค่าใช้จ่ายโดยตรง
- (6) ถ้า - ไม่มีเครื่องบินบินไปในทางเดียวกัน
- แล้ว คำนวนระยะทางบิน; คำนวนรายได้ที่ต้องสูญเสีย : คำนวนค่าจ้างเหมา
เครื่องบิน

C2 เป็นค่าใช้จ่ายในการเลื่อนเที่ยวบินออกไป มีการคำนวนค่าใช้จ่ายโดยตรงและระยะทางบิน
หรือชม.บินเพื่อแสดงทุกครั้ง

- (1) ถ้า - เลื่อนเวลาการบินออกไป ≤ 1 ชั่วโมง
- แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน
- (2) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
- แล้ว คำนวนค่าอาหารต่อผู้โดยสารแต่ละคน (200 บาท/คน)
- (3) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมงแต่ ≤ 8 ชั่วโมง
 - ช่วงเวลาออกอยู่ระหว่าง 0100 - 1000 UTC
- แล้ว คำนวนค่าใช้จ่ายในการจัดหาที่พัก (1500 บาท/คน)
- (4) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมงแต่ ≤ 8 ชั่วโมง
 - ช่วงเวลาออกไม่อยู่ระหว่าง 0100 - 1000 UTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ แล้วนี้ คำนวนค่าใช้จ่ายในการจัดหาที่พักและอาหาร (2500 บาท/คน) ครั้งที่มีการนำไปใช้

- (5) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
- สามารถถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารมากกว่า 8 ชั่วโมง

แล้ว คำนวนค่าใช้จ่ายจัดหาที่พักและอาหาร (2500 บาท/คน)

- (6) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;

ค่าเช่าเครื่องบิน

- (7) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;

ค่าเช่าเครื่องบิน ; คำนวนค่าอาหาร (200 บาท/คน)

- (8) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง
- ถ่ายภายใน 8 ชั่วโมง
- ช่วงเวลาออกอยู่ระหว่าง 0100 - 1000 UTC

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;

ค่าเช่าเครื่องบิน ; คำนวนค่าที่พักและอาหาร (1500 บาท/คน)

- (9) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- มีเครื่องไปในทางเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น-อีกถ่ายผู้โดยสารไม่หมดคือหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง
- ถ่ายภายใน 8 ชั่วโมง
- ช่วงเวลาออกไม่อยู่ระหว่าง 0100-1000 UTC

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;
ค่าเช่าเครื่องบิน ; คำนวน ค่าที่พักและอาหาร (2500 บาท/คน)

(10) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง

- มีเครื่องไปในทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารมากกว่า 8 ชั่วโมง

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;
ค่าเช่าเครื่องบิน ; คำนวนค่าที่พัก และอาหาร (2500 บาท/คน)

(11) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง

- ไม่มีเครื่องบินไปในทางเดียวกัน

แล้ว คำนวนระยะทาง ; คำนวนผลการที่ต้องสูญเสีย ; ค่าเช่าเครื่องบิน

(12) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง

- ไม่มีเครื่องบินไปในทางเดียวกัน
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว คำนวนระยะทาง ; คำนวนผลการที่ต้องสูญเสีย ; ค่าเช่าเครื่องบิน ; คำนวนค่าอาหาร (200 บาท/คน)

(13) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง

- ไม่มีเครื่องบินไปในทางเดียวกัน
- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง
- ถ่ายภายใน 8 ชั่วโมง
- ช่วงเวลาออกอยู่ระหว่าง 0100 - 1000 UTC

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนระยะทางบิน ; ผลการที่ต้องสูญเสีย ;
ค่าเช่าเครื่องบิน ; จัดหาที่พักและอาหาร (1500 บาท/คน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (14) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- ไม่มีเครื่องบินไปในทางเดียวกัน
 - ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง
 - ถ่ายผู้โดยสารภายใน 8 ชั่วโมง
 - ช่วงเวลาออกไม่อยู่ระหว่าง 0100-1000 UTC

แล้ว คำนวณระยะทาง ; คำนวณผลกำไรที่ต้องสูญเสีย ; ค่าเช่าเครื่องบิน ; คำนวณค่าที่พัก และอาหาร (2500 บาท/คน)

- (15) ถ้า - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง

- ไม่มีเครื่องบินไปในทางเดียวกัน
- ถ่ายผู้โดยสารมากกว่า 8 ชั่วโมง

แล้ว คำนวณระยะทาง ; คำนวณผลกำไรที่ต้องสูญเสีย ; ค่าเช่าเครื่องบิน ; คำนวณค่าที่พัก และอาหาร (2500 บาท/คน)

C3 เป็นเงื่อนไขสำหรับผลกระทบค่าใช้จ่ายในการเปลี่ยนเส้นทางบินจะประกอบด้วย

- (1) ถ้า - ระยะทางสนามบินเดิม = ระยะทางของสนามบินใหม่

แล้ว การคำนวณค่าใช้จ่ายให้คิดในโพลีชาร์ทของการเลื่อนเวลาการบิน (delay cost)

- (2) ถ้า - ระยะทางของสนามบินเดิมไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป

แล้ว คำนวณระยะทางบิน และ ค่าใช้จ่ายที่มีผลโดยตรง

- (3) ถ้า - ระยะทางของสนามบินเดิมไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป

- มีเครื่องบินไปในทิศทางเดียวกัน
- สามารถถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว คำนวณระยะทางบิน และ ค่าใช้จ่ายที่มีผลโดยตรง

- (4) ถ้า - ระยะทางของสนามบินเดิมไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น - มีเครื่องบินไปในทิศทางเดียวกัน - ไม่อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ่ายผู้โดยสารหมด
- ถ่ายผู้โดยสาร > 4 ชั่วโมง

แล้ว คำนวณระยะทางบิน ; ค่าใช้จ่ายที่มีผลโดยตรง ; ค่าที่พักและอาหาร (2500 b/p/n)

- (5) ถ้า - ระยะทางของสนามบินเดิมไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป
- มีเครื่องบินไปในทิศทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด

แล้ว คำนวณผู้โดยสารที่เหลือ ; คำนวณระยะทางบิน ; ผลการที่ต้องเสียไป

- (6) ถ้า - ระยะทางของสนามบินเดิม ไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป
- มีเครื่องบินไปในทิศทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว คำนวณผู้โดยสารที่เหลือ ; คำนวณระยะทางบิน ; ผลการที่ต้องเสียไป

- (7) ถ้า - ระยะทางของสนามบินเดิม ไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป
- มีเครื่องบินไปในทิศทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง

แล้ว คำนวณระยะทางบิน และ ค่าใช้จ่ายที่มีผลโดยตรง ; คำนวณค่าที่พักและอาหาร (2500 บาท/คน)

- (8) ถ้า - ระยะทางของสนามบินเดิม ไม่เท่ากับ ระยะทางของสนามบินใหม่ที่จะไป
- ไม่มีเที่ยวบินที่ไปในทิศทางเดียวกัน

แล้ว คำนวณระยะทาง ; คำนวณผลการที่ต้องเสียไป ; ค่าใช้จ่ายที่มีผลโดยตรง

C4 ค่าใช้จ่ายของการเปลี่ยนเครื่องบินเดียวกัน ตรงตามเวลาเดิม

- คำนวณระยะทางบิน
- คำนวณค่าใช้จ่ายที่กระทบโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C5 ค่าใช้จ่ายของการเปลี่ยนเครื่องต่างชนิดกันตามเวลาเดิม จะต้องคำนวณระยะทางบิน และ ค่าใช้จ่ายที่กระทบโดยตรง เพื่อแสดงข้อมูลให้ผู้ใช้ทราบตลอด และ เงื่อนไขประกอบต่าง ๆ จะมีดังนี้

(1) ถ้า - ความจุผู้โดยสาร/สําทาระของเครื่องลาเดิม \leq ความจุผู้โดยสาร/สําทาระของลาใหม่

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง

(2) ถ้า - ความจุผู้โดยสาร/สําทาระของเครื่องลาเดิม $>$ ความจุผู้โดยสาร/สําทาระของลาใหม่

- มีเครื่อง ไปในทิศทาง เดียวกัน
- ถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง

(3) ถ้า - ความจุผู้โดยสาร/สําทาระของเครื่องลาเดิม $>$ ความจุผู้โดยสาร/สําทาระของลาใหม่

- มีเครื่อง ไปในทิศทาง เดียวกัน
- ถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผล
ค่าที่พักและอาหาร (2500 บาท/คน)

(4) ถ้า - ความจุผู้โดยสาร/สําทาระของเครื่องลาเดิม $>$ ความจุผู้โดยสาร/สําทาระของลาใหม่

- ถ่ายผู้โดยสารไม่หมด

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผู้
โดยสารที่เหลือ ; ผลการที่ต้องสูญเสีย ; ค่าเช่าเครื่องลาอื่น

(5) ถ้า - ความจุผู้โดยสาร/สําทาระของเครื่องลาเดิม $>$ ความจุผู้โดยสาร/สําทาระของลาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ถ่ายผู้โดยสารไม่หมด เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผู้โดยสารที่เหลือ ; ผลกาไรที่ต้องสูญเสีย ; ค่าเช่าเครื่องอื่น

(6) ถ้า - ความจุผู้โดยสาร/สําทาระของ เครื่องลาเดิม > ความจุผู้โดยสาร/สําทาระของลาใหม่

- ถ่ายผู้โดยสารไม่หมด

- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผู้โดยสารที่เหลือ ; ผลกาไรที่ต้องสูญเสีย ; ค่าเช่าเครื่องอื่น ; คำนวนค่าที่พักของผู้โดยสาร (2500 บาท/คน)

(7) ถ้า - ความจุผู้โดยสาร/สําทาระของ เครื่องลาเดิม > ความจุผู้โดยสาร/สําทาระของลาใหม่

- ไม่เครื่องหรือเที่ยวบินไปในทิศทางเดียวกัน

แล้ว แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผลกาไรที่ต้องสูญเสีย ; ค่าเช่าเครื่องบินแทน

C6 เป็นค่าใช้จ่ายสำหรับการเปลี่ยนเครื่องต่างชนิดกัน แล้วเลื่อนเวลาบินออกไป มีเงื่อนไขดังนี้

- คำนวนค่าใช้จ่ายโดยตรงที่ต้องเสีย - คำนวนระยะทางบิน (หรือชั่วโมงบิน)

(1) ถ้า - ความจุผู้โดยสาร/สําทาระ เครื่องเดิม = ความจุผู้โดยสาร/สําทาระของ เครื่องใหม่ที่จะ เปลี่ยน

- เลื่อนเวลาบิน ≤ 1 ชั่วโมง

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน

(2) ถ้า - ความจุผู้โดยสาร/สําทาระ เครื่องเดิม = ความจุผู้โดยสาร/สําทาระของ เครื่องใหม่ที่จะ เปลี่ยน

- เลื่อนเวลาบิน > 1 ชั่วโมง

- เลื่อนเวลาบิน ≤ 4 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหาร
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องดื่ม (200 บาท/คน)

(3) ถ้า - ความจุผู้โดยสาร/สัถาระ เครื่องเดิม = ความจุผู้โดยสาร/สัถาระของ
เครื่องใหม่ที่จะ เปลี่ยน

- เลื่อนเวลาบินออกไปมากกว่า 4 ชม. แต่ \leq 8 ชม.

- เวลาที่เครื่องออกอยู่ในช่วง เวลา 0100-1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหารและที่
พัก (1500บาท/p/n)

(4) ถ้า - ความจุผู้โดยสาร/สัถาระ เครื่องเดิม = ความจุผู้โดยสาร/สัถาระของ
เครื่องใหม่ที่จะ เปลี่ยน

- เลื่อนเวลาบินออกไปมากกว่า 4 ชม. แต่ \leq 8 ชม.

- เวลาที่เครื่องออกไม่อยู่ในช่วง เวลา 0100-1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหารและที่
พัก (2500บาท/p/n)

(5) ถ้า - ความจุผู้โดยสาร/สัถาระ เครื่องเดิม = ความจุผู้โดยสาร/สัถาระของ
เครื่องใหม่ที่จะ เปลี่ยน

- เลื่อนเวลาบินออกไปมากกว่า 8 ชม.

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหารและที่
พัก (2500บาท/p/n)

(6) ถ้า - ความจุผู้โดยสาร/สัถาระของเครื่องเดิม ไม่เท่ากับ ความจุผู้โดยสาร/
สัถาระของ เครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน

- สามารถถ่ายผู้โดยสารได้หมด

- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

- เลื่อนเวลาบินออกไป \leq 1 ชั่วโมง

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน

(7) ถ้า - ความจุผู้โดยสาร/สัถาระของเครื่องเดิม ไม่เท่ากับ ความจุผู้โดยสาร/
สัถาระของ เครื่องใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
- สามารถถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
- เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
- เลื่อนเวลาบิน ≤ 4 ชั่วโมง

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหาร
เครื่องดื่ม (200 บาท/คน)

- (8) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
 - เลื่อนเวลาบินออกไปมากกว่า 4 ชม.แต่ ≤ 8 ชม.
 - เวลาที่เครื่องออกอยู่ในช่วง เวลา 0100-1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าที่พักและ
อาหาร (1500 บาท/คน)

- (9) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
 - เลื่อนเวลาบินออกไปมากกว่า 4 ชม.แต่ ≤ 8 ชม.
 - เวลาที่เครื่องออกไม่อยู่ในช่วง เวลา 0100-1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหารและ
ที่พัก (2500 บาท/คน)

เอกสารนี้ (10) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกหนึ่งหน้าที่ให้ดำเนินการ นื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ภาระของเครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
- สามารถถ่ายผู้โดยสารได้หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
- เลื่อนเวลาบินออกไปมากกว่า 8 ชม.

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหารและที่พัก (2500 บาท/คน)

- (11) ถ้า - ความจุผู้โดยสาร/สัการะของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สัการะของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชม.

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหาร & ที่พัก (2500 บาท/คน)

- (12) ถ้า - ความจุผู้โดยสาร/สัการะของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สัการะของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชม.
 - เลื่อนเวลาบินออกไป ≤ 1 ชั่วโมง

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหาร & ที่พัก (2500 บาท/คน)

- (13) ถ้า - ความจุผู้โดยสาร/สัการะของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สัการะของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - สามารถถ่ายผู้โดยสารได้หมด
 - ถ่ายผู้โดยสารมากกว่า 4 ชม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น-อีก เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลื่อนเวลาบิน ≤ 4 ชั่วโมง

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหาร & ที่พัก (2500 บาท/คน) ; ค่าอาหาร& เครื่องดื่ม (200 บาท/คน)

- (14) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน

- สามารถถ่ายผู้โดยสารได้หมด

- ถ่ายผู้โดยสารมากกว่า 4 ชม.

- เลื่อนเวลาบินออกไปมากกว่า 4 ชม.แต่ ≤ 8 ชม.

- เวลาที่เครื่องออกอยู่ในช่วงเวลา 0100-1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหาร & ที่พัก ; (1500 บาท/คน)

- (15) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน

- สามารถถ่ายผู้โดยสารได้หมด

- ถ่ายผู้โดยสารมากกว่า 4 ชม.

- เลื่อนเวลาบินออกไปมากกว่า 4 ชม.แต่ ≤ 8 ชม.

- เวลาที่เครื่องออกไม่อยู่ในช่วงเวลา 0100 - 1000 UTC

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะเวลาบิน ; คำนวนค่าอาหาร & ที่พัก (2500 บาท/คน)

- (16) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน

- สามารถถ่ายผู้โดยสารได้หมด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของกรมการขนส่งทางอากาศ ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลื่อนเวลาบินออกไปมากกว่า 8 ชม.

แล้ว แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; จำนวนค่าอาหาร & ที่พัก (2500/คน)

- (17) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด

แล้ว จำนวนผู้โดยสารที่เหลือ ; จำนวนชั่วโมงบิน(ระยะทาง) ; จำนวนผลการที่เสียไป

- (18) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
 - เลื่อนเวลาบินออกไป ≤ 1 ชั่วโมง

แล้ว จำนวนผู้โดยสารที่เหลือ ; จำนวนชั่วโมงบิน(ระยะทาง) ; จำนวนผลการที่เสียไป; แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน

- (19) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับความจุผู้โดยสาร/สภาระของเครื่องใหม่
- มีเครื่องหรือเที่ยวบินไปในระยะทางเดียวกัน
 - ถ่ายผู้โดยสารไม่หมด
 - ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
 - เลื่อนเวลาบินออกไปมากกว่า 1 ชั่วโมง
 - เลื่อนเวลาบินออกไป ≤ 4 ชั่วโมง

แล้ว จำนวนผู้โดยสารที่เหลือ ; จำนวนชั่วโมงบิน(ระยะทาง) ; จำนวนผลการที่เสียไป; แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; จำนวน

ค่าเครื่องดื่ม (200 บาท/คน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในกิจการของมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(20) ถ้า - ความจุผู้โดยสาร/สําทาระของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สําทาระของ เครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปนระยะทาง เดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
- เลื่อนเวลาบินออกไปมากกว่า 4 ชม.แต่ <= 8 ชม.
- เวลาที่เครื่องออกอยู่นช่วง เวลา 0100-1000 UTC

แล้ว คำนวนผู้โดยสารที่เหลือ ; คำนวนชั่วโมงบิน(ระยะทาง) ; คำนวนผลกำไรที่เสียไป; แสดงค่าใช้จ่ายที่ต้องเสียโดยตรง ; ระยะทางบิน ; คำนวนค่าอาหาร & และที่พัก (1500 บาท/คน)

(21) ถ้า - ความจุผู้โดยสาร/สําทาระของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สําทาระของ เครื่องใหม่

- มีเครื่องหรือเที่ยวบินไปนทาง เดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง
- เลื่อนเวลามากกว่า 4 ชม.แต่ <= 8 ชั่วโมง
- เวลาที่ออกไม่อยู่นช่วง 0100 - 1000 UTC

แล้ว แสดงผลการคํานวนระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผู้โดยสารที่เหลือ ; คำนวนชั่วโมงบิน(ระยะทาง); ผลกำไรที่เสียไป คำนวนค่าเช่าเครื่องบิน ; คำนวนอาหารและที่พัก (2500 บาท/คน)

(22) ถ้า - ความจุผู้โดยสาร/สําทาระของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สําทาระของ เครื่องใหม่

- มีเครื่องบินไปนทาง เดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสารภายใน 4 ชั่วโมง

เลือกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เฉพาะผู้ถือครองเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
แล้ว แสดงผลการคํานวนระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวน

ผู้โดยสารที่เหลือ ; คำนวนชั่วโมงบิน(ระยะทาง) ; ผลการที่เสียไป คำนวน
ค่าเช่าเครื่องบิน ; คำนวนค่าที่พัก (2500 บาท/คน)

(23) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/ส
ภาระของ เครื่องใหม่

- มีเครื่องบินในทิศทางเดียวกัน

- ถ่ายผู้โดยสารไม่หมด

- ถ่ายผู้โดยสารมากกว่า 4 ชม.

แล้ว แสดงค่าใช้จ่ายที่ต้อง เสียโดยตรง ; ระยะทางบิน คำนวนค่าอาหารและที่พัก
(2500 บาท/คน)

(24) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/ส
ภาระของ เครื่องใหม่

- มีเครื่องบินในทิศทางเดียวกัน

- ถ่ายผู้โดยสารไม่หมด

- ถ่ายผู้โดยสารมากกว่า 4 ชั่วโมง

- เลื่อนเวลาบินออกไป < 1 ชม.

แล้ว แสดงค่าใช้จ่ายโดยตรง ; ระยะทางบิน คำนวนค่าอาหารและที่พัก (2500
บาท/คน)

(25) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/ส
ภาระของ เครื่องใหม่

- มีเครื่องบินบินไปทิศทางเดียวกัน

- ถ่ายผู้โดยสารไม่หมด

- ถ่ายผู้โดยสาร > 4 ชม.

- เลื่อนเวลาบิน > 1 ชม. แต่ <= 4 ชม.

แล้ว คำนวนค่าอาหารและ เครื่องดื่ม (200 บาท/คน) แสดงค่าใช้จ่ายโดยตรง ;
ระยะทางบิน

(26) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/ส

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ภายในเท่านั้น เมื่อผู้ใดเห็นแจ้งใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้นภาระของ เครื่องใหม่ ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีเครื่องบินไปนทิศทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสาร > 4 ชม.
- เลื่อนเวลาบิน > 4 ชม. แต่ <= 8 ชม.
- เวลาเครื่องออกอยู่ช่วง 0100 - 1000 UTC.

แล้ว คำนวนค่าอาหารและที่พัก (1500 บาท/คน)

(27) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับ ความจุผู้โดยสาร/สภาระของเครื่องใหม่

- มีเครื่องไปนทิศทางเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสาร > 4 ชม.
- เลื่อนเวลาบิน > 4 ชม. แต่ <= 8 ชม.
- เวลาที่ออกไม่อยู่ช่วง 0100 - 1000 UTC

แล้ว คำนวนค่าอาหารและที่พัก (2500 บาท/คน) แสดงผลการคำนวณระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนผลกำไรที่เสียไป ; คำนวนค่าเช่าเครื่องบิน

(28) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับ ความจุผู้โดยสาร/สภาระของเครื่องใหม่

- มีเครื่องไปทิศเดียวกัน
- ถ่ายผู้โดยสารไม่หมด
- ถ่ายผู้โดยสาร > 4 ชม.
- เลื่อนเวลาบิน > 8 ชม.

แล้ว คำนวนค่าอาหารและที่พัก (2500 บาท/คน)

(29) ถ้า - ความจุผู้โดยสาร/สภาระของเครื่องเดิม ไม่เท่ากับ ความจุผู้โดยสาร/สภาระของเครื่องใหม่

- ไม่มีเที่ยวบินที่บินไปนทิศทางเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น ยี่สิบห้าปีหลังจากที่ฉบับนี้จัดทำ และต้องขออนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(30) ถ้า - ความจุผู้โดยสาร/สํานาการะของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สํานาการะของ เครื่องใหม่

- ไม่มีเครื่องไปทิศทาง เดียวกัน

- เลื่อนเวลาบิน ≤ 1 ชม.

แล้ว คำนวนค่าเช่าเครื่องบิน ; ผลกาไรที่ต้องเสียไป

(31) ถ้า - ความจุผู้โดยสาร/สํานาการะของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สํานาการะของ เครื่องใหม่

- ไม่มีเครื่องไปทิศทาง เดียวกัน

- เลื่อนเวลาบิน > 1 ชม. แต่ ≤ 4 ชม.

แล้ว คำนวนอาหารและ เครื่องดื่ม (200 บาท/คน) ; ค่าเช่าเครื่องบิน

(32) ถ้า - ความจุผู้โดยสาร/สํานาการะของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สํานาการะของ เครื่องใหม่

- ไม่มีเครื่องไปทิศเดียวกัน

- เลื่อนเวลาบิน > 4 ชม. แต่ ≤ 8 ชม.

- เวลาที่เครื่องออกอยู่ช่วง 0100 - 1000 UTC

แล้ว คำนวนค่าใช้จ่ายเช่าเครื่องบิน ; คำนวนผลกาไรที่เสียไป ; แสดงผล การคํานวนระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนอาหาร และที่พัค (1500 บาท/คน)

(33) ถ้า - ความจุผู้โดยสาร/สํานาการะของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สํานาการะของ เครื่องใหม่

- ไม่มีเครื่องไปทิศเดียวกัน

- เลื่อนเวลาบิน > 4 ชม แต่ ≤ 8 ชม.

- เวลาที่ออกไม่อยู่ในช่วง 0100 - 1000 UTC

แล้ว คำนวนค่าใช้จ่ายเช่าเครื่องบิน ; คำนวนผลกาไรที่เสียไป ; แสดงผล การคํานวนระยะทางบิน ; ค่าใช้จ่ายที่กระทบโดยตรง ; คำนวนอาหาร

เอกสารนี้เป็นเอกสารลับไว้ใช้เฉพาะการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าหนงเอกสารหนาดังที่มิการนำไปใช้

(34) ถ้า - ความจุผู้โดยสาร/สํานาการะของ เครื่อง เดิม ไม่เท่ากับ ความจุผู้โดยสาร/สํานาการะของ เครื่องใหม่

ภาระของ เครื่องใหม่

- ไม่มีเครื่องไปนทิศทาง เดียวกัน
- เลื่อนเวลาบิน > 8 ชม.

แล้ว ค่าอาหารและที่พัก (2500 บาท/คน) ; ค่าเช่าเครื่องบิน

- ความสะดวกสบายของผู้โดยสาร (PAX. Convenient) จะประกอบไปด้วยสถานที่พัก สนามบิน ฯลฯ ในที่นี้จะกล่าวถึง สถานที่พักและสนามบิน ซึ่งแบ่งออกได้เป็น P1 และ P2 มีเงื่อนไขที่นำมาคิดได้ดังนี้

P1 เป็นผลกระทบเมื่อเลือกการปฏิบัติในการยกเลิกเที่ยวบิน ; เลื่อนเวลาบิน ; เปลี่ยนเส้นการบิน ; เปลี่ยนเครื่องบินทั้งหมด ยกเว้นการเปลี่ยนเครื่องบินตรงตามเวลาเดิม ระบบจะแสดงข้อมูลเกี่ยวกับ

- ความสะดวกสบายของสนามบิน - ดี
- ไม่ดี
- ความสะดวกสบายของสถานที่พัก - ดี
- ไม่ดี

P2 เป็นความสะดวกของผู้โดยสาร เมื่อ ผู้ปฏิบัติการบินเลือก การเปลี่ยนเส้นทางบิน คือ

- เปรียบเทียบเส้นทาง เดิม และ เส้นทางใหม่แสดงให้เห็นถึงผลกระทบต่างๆ
- แสดงข้อมูลถึงความสะดวกของผู้โดยสารทั้งหมด ซึ่งมีลักษณะคล้าย P1

- ผู้โดยสารคนสำคัญ และภาพจน์ที่ดีต่อสายการบิน (VIP and IMAGE) ข้อมูลเหล่านี้ จะอยู่ในฐานข้อมูลของเที่ยวบิน โดยระบบจะแสดงข้อมูลออกมาเกี่ยวกับ

- ถ้า ฐานข้อมูล มีสัญลักษณ์ A แสดงว่า เที่ยวบินนั้นมี VIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาพพจน์ที่ติดต่อสายการบิน จะบอกมาเป็น - A = มีภาพพจน์ต่อสายการบินมาก
- B = ปานกลาง
- C = ไม่ค่อยมีภาพพจน์

- ผลกระทบของเที่ยวบินถัดไป (NEXT FLIGHT or NEXT SECTOR) แบ่งออกเป็น 3

ส่วนคือ

- F1 เป็นผลกระทบเมื่อผู้ปฏิบัติการเลือกการยกเลิกเที่ยวบิน ระบบจะต้อง
- เช็คว่าจะผลกระทบของเครื่องบินหรือเที่ยวบินนั้นรึไม่ แสดงออกมาให้ผู้ชี้เห็น
- F2 เป็นเงื่อนไขที่กระทบต่อการยกเลิกเที่ยวบินบางส่วน (SECTOR) และการเลื่อนเวลาบินออกไปมีเงื่อนไขดังนี้
- คำนวนเวลาที่ใช้ในการเลื่อนเที่ยวบิน
 - (1) ถ้า - เวลาที่ออกของเที่ยวถัดไป (ETD) > STD ของเที่ยวถัดไปอีกแล้ว - มีผลกระทบต่อเที่ยวบินถัดไป
 - (2) ถ้า - เวลาที่ออกของเที่ยวถัดไป (ETD) <= STD ของเที่ยวถัดไปอีกแล้ว - ไม่มีผลกระทบต่อเที่ยวถัดไป
- F3 เป็นเงื่อนไขสำหรับการเปลี่ยนเส้นทางการบิน
- (1) ถ้า - ระยะทางบินเดิม = ระยะทางบินใหม่
 - เวลาถึงเดิมของสนามบินสุดท้าย >= เวลาถึงของสนามบินสุดท้ายแล้ว - ไม่มีผลกระทบต่อเที่ยวถัดไป
 - (2) ถ้า - ระยะทางบินเดิม = ระยะทางบินใหม่
 - เวลาถึงเดิมของสนามบินสุดท้าย < เวลาถึงของสนามบินสุดท้ายแล้ว คำนวนเวลาของเที่ยวถัดไป + GST
 - (3) ถ้า - ระยะทางบินเดิม = ระยะทางบินใหม่
 - เวลาถึงเดิมของสนามบินสุดท้าย < เวลาถึงของสนามบินสุดท้าย
 - เวลาที่ออก (ETD) <= เวลาออกของเที่ยวถัดไป (STD)

เอกสารนี้เป็นเอกสารที่แล้วไม่มีผลกระทบต่อเที่ยวถัดไป ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ (4) ถ้า - ระยะทางบินเดิม = ระยะทางบินใหม่ ถ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เวลาถึง เดิมของสนามบินสุดท้าย < เวลาถึงของสนามบินสุดท้าย
- เวลาที่ออก (ETD) > เวลาออกของเที่ยวถัดไป (STD)

แล้ว คำนวน เวลาในการ delay ของเที่ยวถัดไป (ETD - STD)

(5) ถ้า - ระยะทางเดิม ไม่เท่ากับ ระยะทางใหม่

แล้ว - หาเที่ยวบินถัดไป ; แสดงผลกระทบต่อเที่ยวบินถัดไป

เงื่อนไขของส่วนประกอบผลกระทบของทางเลือกปฏิบัติการบินที่กล่าวมาแล้วนั้น ได้
แสดงให้เห็นเป็นโพชาร์ทไว้ทั้งหมดในส่วนของภาคผนวกที่ 3 ท้ายเล่ม

5.2.3 ฐานข้อมูลของระบบ

ในฐานข้อมูลจะมีค่าของ Y ; N ; และ เวลา ; ค่าเหล่านี้จะแทนความหมาย
คือ ถ้าไม่มีปัญหา = Y มีปัญหา = N แล้วจะตามด้วยช่วง เวลาที่มีปัญหา ตัวอย่างของฐานข้อมูล
แสดงในภาคผนวกที่ 2

1) ฐานข้อมูลเกี่ยวกับสนามบิน จะประกอบไปด้วยสนามบินทั้งหมดประมาณ 50
สนามบิน เป็นสนามบินที่ทางบริษัท การบินไทยได้รับอนุญาตเปิดทำการธุรกิจเกี่ยวกับการบิน ใน
แต่ละสนามบินจะมีข้อมูลเกี่ยวกับ น้ำมันเชื้อเพลิง ; แผนการบิน/ความสูง/เวลาเข้าออก ;
เครื่องอำนวยความสะดวกของสนามบิน ; ช่วงเวลาเปิด/ปิดสนามบิน ; รันเวย์ ; อากาศ ;
ปัญหาการเมือง ; การไม่ปฏิบัติตามหน้าที่ของเจ้าหน้าที่ที่เกี่ยวข้อง ; การอนุญาตในการบิน ; ภาพ-
พจน์ต่อสายการบิน

2) ฐานข้อมูลเกี่ยวกับเครื่องบิน

จำนวนเครื่องบินของทางการบินไทยที่ใส่ไว้ในฐานข้อมูลจะมีทั้งหมด 31 ลำ ในแต่ละ
ลำจะมีข้อมูลเกี่ยวกับ เลขทะเบียนเครื่อง ; แบบของเครื่องบิน ; ความพร้อมของเครื่องบิน
; ความพร้อมของลูกเรือระยะทางบินสูงสุด ; ค่าของ GST (ช่วงเวลาในการเช็คเครื่อง) ;
จำนวนที่นั่งสูงสุด

ในฐานข้อมูลของเครื่องบิน 31 ลำ สามารถแยกชนิดของเครื่องบินออกเป็น

เอกสารนี้ เครื่อง DC 10 จะมี 3 ลำ มีเลขทะเบียนเครื่อง TMA - TMC
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังได้ขออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เครื่อง AB 6 (AIR BUS 600) มี 7 ลำ เลขทะเบียนตั้งแต่ TAA-TAG

เครื่องบิน AB 4 (AIR BUS 400) มี 11 ลำ เลขทะเบียนตั้งแต่ TKG-TGP และ
TGR ;TGT;TGW;TGX และ TAX

เครื่องบิน AB 3 (AIR BUS 300) มี 2 ลำ เลขทะเบียน TAZ ; TAY.

เครื่องบิน 742 (โบอิง 747-200) มี 6 ลำ เลขทะเบียน TGA;TGB;TGC;TGF;
TGS;TGG

เครื่องบิน 743 (โบอิง 747-300) มี 2 ลำ เลขทะเบียน TGD;TGE

3) ฐานข้อมูลของเที่ยวบิน ในส่วนของฐานข้อมูลนี้จะแยกออกเป็น 6 files ดังนี้

ก) ฐานข้อมูลของ TG 900 - TG 1000

ข) ฐานข้อมูลของ TG 700 - TG 900

ค) ฐานข้อมูลของ TG 600 - TG 700

ง) ฐานข้อมูลของ TG 500 - TG 600

จ) ฐานข้อมูลของ TG 400 - TG 500

ฉ) ฐานข้อมูลของ TG 300 - TG 400

ข้อมูลของแต่ละเที่ยวบินจะประกอบด้วยข้อมูลของเลขเที่ยวบิน (flight-number)
; วันที่บิน ; ส่วนที่จะเป็น (SECTOR) ; เลขทะเบียนเครื่องบินที่ใช้บิน ; เที่ยวบินออกจาก ;
เวลาที่ออก ; ไปถึงสนามบินใด ; เวลาที่ถึง ; จำนวนผู้โดยสารที่ไป ; มีแขกผู้มีเกียรติระ
เบลา (VIP) ; ระยะทางที่บิน

5.3 การทำงานของโปรแกรม

เมื่อเข้าสู่ระบบผู้เชี่ยวชาญ ผู้ใช้จะต้องใส่ข้อมูลเกี่ยวกับเที่ยวบินที่มีปัญหา ซึ่งเป็น
เที่ยวบินที่อยู่ในตารางบินปกติ ข้อมูลที่ผู้ใช้ต้องใส่ให้ระบบเมื่อระบบถาม คือ เลขของเที่ยวบิน
(flight-no) วันเดือนปีที่บิน (flight -Date) และส่วนที่บิน (sector) ระบบจะดึงข้อมูล
เกี่ยวกับเที่ยวบินนั้น ข้อมูลของเครื่องบินนั้น และข้อมูลของสนามบินที่เกี่ยวข้อง (ออกเดินทางและ
ไปถึง) แล้วระบบจะเข้าสู่เมนู เลือกปฏิบัติการบิน ในส่วนของเมนูทางเลือกปฏิบัติการบินจะ
ประกอบไปด้วยทางเลือกต่าง ๆ คือ แสดงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1 ยกเลิกเที่ยวบินบางส่วน (SECTOR CANCELLATION) ในส่วนนี้จะนำข้อมูลที่ได้ มาไปตรวจเช็คกับเงื่อนไขต่าง ๆ ของ ค่าใช้จ่าย (C1); ความสะดวกสบาย ของผู้โดยสาร (P1); แยกคนสำคัญ (V1) ; ผลกระทบต่อเที่ยวบินนั้น (F2); ภาพพจน์ต่อสายการบิน (I1)

5.3.2 ยกเลิกเที่ยวบินทั้งหมด (FLIGHT CANCELLATION) ระบบจะแจ้งข้อมูลออกมา เช่นเดียวกับการยกเลิกเที่ยวบินบางส่วน แต่ต่างกันตรงที่ผลกระทบต่อเที่ยวบิน ถัดไปใช้เงื่อนไขของโพลีชาร์ท F1

5.3.3 การเลื่อนเวลาของเที่ยวบิน เงื่อนไขต่าง ๆ ที่ระบบต้องตรวจสอบเพื่อแสดงข้อมูลออกมาคือ เงื่อนไขของเครื่องบิน (AC1); ลูกเรือ (CW1) ; น้ำมันเชื้อเพลิง (FU1) ; แผนการบิน/ความสูง/เวลาเข้าออก (SL1) ; เครื่องอำนวยความสะดวกของสนามบิน (FC1) ; ช่วงเวลาปิด/เปิดสนามบิน (CU1) ; รันเวย์ (RW1) ; อากาศ (WX1) ; การเมือง (PO1) การไม่ปฏิบัติหน้าที่ของพนักงานที่เกี่ยวข้อง (ID1) ข้อมูลของผลกระทบนั้นจะเหมือนกับยกเลิกเที่ยวบินบางส่วน แต่ต่างกันตรง (C2) เท่านั้น

5.3.4 การเปลี่ยนเส้นทางบิน ระบบจะถามผู้ซ้ถึงสนามบินใหม่ที่ต้องการเปลี่ยนไป ชั่วโมงบินที่ต้องใช้เวลาที่ต้องเดินทางไปถึง แล้วเงื่อนไขที่ระบบนำข้อมูลไปเช็คจะเหมือนกับ การเลื่อนเวลาบิน ต่างกันที่ เครื่องบิน (AC2) และมีการตรวจเงื่อนไขของ การอนุญาตการบิน (PM1) ; ค่าใช้จ่าย (C3) ; ความสะดวกสบายผู้โดยสาร (P2) ; แยกคนสำคัญ (V1) ; ผลกระทบเที่ยวบินถัดไป (F3) ภาพพจน์ต่าง ๆ (I1)

5.3.5 การเปลี่ยนเครื่องบิน เมื่อเลือกการกระทำนี้ระบบจะถามถึง เครื่องบินลำใหม่ที่จะนำมาแทน โดยฝ่ายผู้ใช้จะต้องตอบให้แก่ระบบโดยระบบจะแจ้งข้อมูลแยกออกเป็น

- เปลี่ยนเครื่องบินแบบเดียวกัน ตรงตามเวลาเดิม จะพิจารณา เครื่องบิน (AC3) และผลกระทบคือ ค่าใช้จ่าย (C4) กับ ภาพพจน์ต่อสายการบิน (I1)

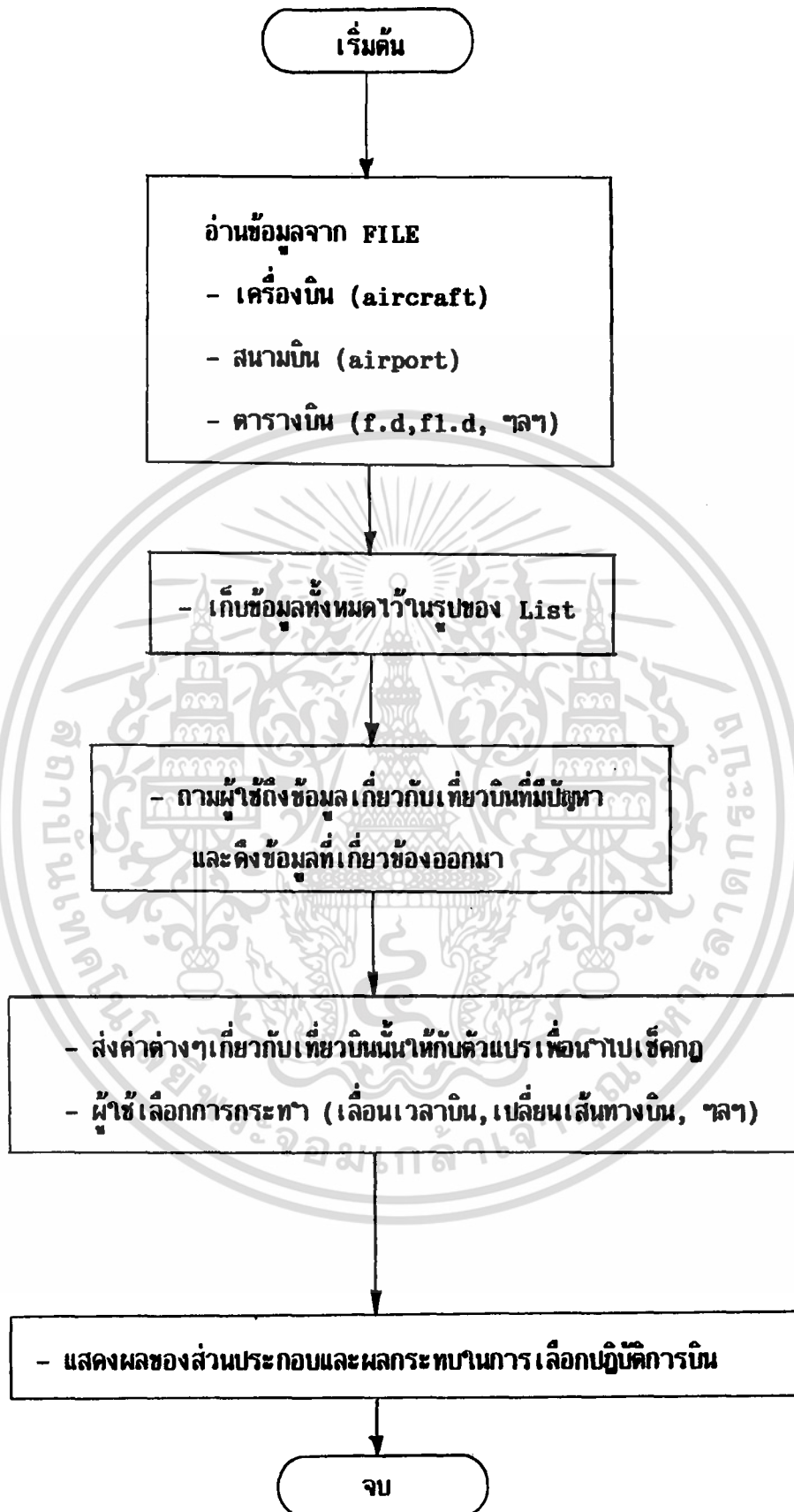
- เปลี่ยนเครื่องบินแบบเดียวกันเลื่อนเวลานับออกไป จะพิจารณาเครื่องบิน (AC3)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ใดเห็นใบเขียวระบุเงื่อนไขในการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ก็กัและในส่วนอื่นก็เหมือนกับการเลื่อนเวลาบิน ถ้า ส่วนผลกระทบก็เช่นเดียวกับไปใช้

เลื่อนเวลาบิน แต่ไม่คิดผลกระทบต่อเที่ยวบินถัดไป

- เปลี่ยนเครื่องต่างชนิด เวลาบินตามเดิม จะพิจารณาเครื่องบิน (AC 3) ; ลุกเรือ (CW2) ; การอนุญาตในการบิน (PM2) ; แผนการบิน/ความสูง/ เวลาเข้าออก (SL1) ; สิ่งอำนวยความสะดวก (FC1); รันเวย์ (RW1) ส่วนผลกระทบจะคิด ค่าใช้จ่าย (CS); ความสะดวกสบายผู้โดยสาร (P1); แยกคนสำคัญ (V1); ภาพพจน์ต่อสายการบิน (I1)
- เปลี่ยนเครื่องต่างชนิดและเลื่อนเวลาบิน จะเช็คเงื่อนไขเหมือนกับ การเปลี่ยนเครื่องต่างชนิดเวลาบินเหมือนเดิม แต่จะเพิ่มเชื้อเพลิง (FU1) ; ช่วงเวลาเปิด/ปิดสนามบิน (CU1) ; อากาศ (WX1) ทางการเมือง (PO1) ; การไม่ปฏิบัติตามหน้าที่ของพนักงานที่เกี่ยวข้อง (ID1) สำหรับผลกระทบจะต่างกับเปลี่ยนเครื่องต่างชนิดเวลาบิน เหมือนเดิม คือ ค่าใช้จ่าย (C6) เหมือนกันหมด

การทำงานทั้งหมดของระบบนี้ แสดงได้ดังรูป 5-2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีรูปที่ 5-2 ไฟล์ซาร์ที่แสดงการทำงานผ่านของระบบ อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและข้อเสนอแนะ

สำหรับขั้นแรกที่เริ่มพัฒนาระบบนี้ ได้ทำการพัฒนาบนเปลือกกระบวนผู้เชี่ยวชาญ (EXPERT SYSTEM SHELL) ชื่อว่า Personal Consultant Plus ของบริษัท Texas Instruments เกิดปัญหาอย่างหนึ่งก็คือ การดึงข้อมูลของตารางบิน สนามบิน เครื่องบิน จากฐานข้อมูล ซึ่ง PC-PLUS ทำการดึงข้อมูลจาก DBASE III PLUS ทำให้เสียเวลาในการดึงข้อมูลและสรุปผลมาก เกินกว่าที่ทางการบินไทยยอมรับจึงได้ทำการพัฒนาระบบขึ้นมาใหม่โดยใช้ภาษา LISP (Scheme LISP) เพื่อแก้ไขข้อบกพร่องนี้และทำให้ทางการบินไทยยอมรับ

ขั้นตอนการพัฒนาระบบที่สำคัญที่สุดคือ การนำเอาความรู้และประสบการณ์ของผู้เชี่ยวชาญ (Knowledge Acquisition) มาแทนให้อยู่ในรูปของทฤษฎี เพราะจะต้องทำการค้นคว้ารวบรวมจากประสบการณ์ความรู้ของผู้เชี่ยวชาญที่เขาเหล่านั้นใช้ในการแก้ปัญหา โดยการสัมภาษณ์ สังเกตจากสถานการณ์จริง

ประเมินผลการใช้งานของระบบ

แม้ว่าระบบนี้ได้พัฒนาอยู่บนเครื่องระดับไมโครคอมพิวเตอร์ก็ตาม แต่ก็สามารถนำไปใช้งานดังต่อไปนี้

- 1) สำหรับฝึกอบรมพนักงานใหม่ของฝ่ายควบคุมปฏิบัติการบิน ให้ทราบถึงหลักการคิดเมื่อเกิดเหตุการณ์ผิดปกติเกี่ยวกับเที่ยวบิน
- 2) การคำนวณหาค่า INDEX ของนักบิน
- 3) การเปลี่ยนแปลงข้อมูลสามารถทำได้ตลอดเวลาโดยผู้ใช้สามารถแก้ไข เพิ่มเติม หรือเปลี่ยนแปลงข้อมูลเกี่ยวกับ ตารางบิน สนามบิน เครื่องบิน เพราะข้อมูลเหล่านี้มีการเปลี่ยนแปลงตลอดเวลา
- 4) การแสดงถึงความเป็นไปได้ที่จะนำเอากระบวนผู้เชี่ยวชาญ (EXPERT SYSTEMS) มาใช้

งานทางด้าน การควบคุมปฏิบัติการบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาเหตุที่ทำให้พัฒนาระบบนั้นทางกลับกัน

ในการพัฒนาระบบจากข้อสรุปของการกระทำ (ACTIONS ต่างๆคือ CANCEL , DELAY ,REROUTE, CHANGE A/C) ไปหาส่วนประกอบและผลกระทบของทางเลือกปฏิบัติการบิน โดยมิได้พัฒนาขึ้นเมื่อเกิดเหตุการณ์ผิดปกติ เช่น เครื่องเสีย โดยให้ผู้ใช้ถามถึงทางปฏิบัติของการแก้ปัญหา โดยให้ระบบสรุปผลการแก้ปัญหาออกมา (CANCEL, DELAY, REROUTE, CHANGE A/C) เนื่องจาก

- 1) ในการตัดสินใจของผู้เชี่ยวชาญในบางครั้งก็ยังมีข้อผิดพลาดเกิดขึ้นได้เพราะปัญหาแต่ละกรณีนั้นผู้เชี่ยวชาญต้องใช้กฎเกณฑ์ต่างๆเป็นจำนวนมาก เข้าช่วยในการตัดสินใจ ซึ่งกฎเกณฑ์เหล่านี้ขึ้นอยู่กับประสบการณ์ของแต่ละบุคคลด้วย เช่น ปัญหาในการเปลี่ยนเครื่องบิน ผลกระทบในการเปลี่ยนเครื่องบินมีมากมายดังนั้นการที่จะพัฒนาระบบไปสู่ขั้นตอนการตัดสินใจแทนผู้เชี่ยวชาญนั้นเป็นสิ่งที่ยุ่งยากอยู่พอสมควร
- 2) ข้อมูลที่นำมาช่วยในการตัดสินใจนั้นเป็นความลับของทาง บริษัท การบินไทย จำกัด ที่ได้สงวนเอาไว้
- 3) ทำให้เกิดความง่าย สะดวกรวดเร็ว แก่ผู้ใช้

ข้อเสนอแนะที่จะศึกษาและพัฒนาต่อไป

สิ่งที่ควรจะศึกษาและพัฒนาต่อไปคือ

- 1) การพัฒนาให้ระบบนี้สามารถนำเอาฐานข้อมูลเกี่ยวกับ ตารางบิน เครื่องบิน ฯลฯ จากเครื่องเมนเฟรมของ บริษัท การบินไทย จำกัด ได้ทันทีโดยมิต้องให้ผู้ใช้ทำการป้อนข้อมูลด้วยตนเอง เพราะการป้อนข้อมูลแต่ละครั้งอาจทำให้เกิดความผิดพลาดได้
- 2) ในระบบนี้ใช้ภาษา LISP ที่เป็น interpreter ยังมิได้พัฒนาเป็น LISP ที่เป็น compiler ถ้ามีการพัฒนาเป็น LISP compiler แล้วจะทำให้การทำงานของระบบเพิ่มความเร็วและประสิทธิภาพขึ้นอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยคำแนะนำจาก ศ.ดร.ไพรัช ธัชยพงษ์ และขอขอบคุณ คุณบรรลือ เรืองเล็ก คุณธีรทัต พรพิบูลย์ และ คุณคมสรพร รัชดิธรรกุล ที่ให้คำแนะนำรายละเอียดวิธีการความรู้ปัญหาต่างๆเกี่ยวกับการควบคุมปฏิบัติการบิน และคุณรวาวิช กาญจน อัยยหา ที่ให้คำแนะนำและทดสอบระบบเพื่อหาข้อผิดพลาด รวมทั้งพี่พนักงานบริษัท การบินไทย จำกัดทุกท่าน ตลอดจนเพื่อนและน้องนักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่างๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] BUCHANAN, B.G., AND SHORTLIFFE, E.H.; "RULE-BASE EXPERT SYSTEMS: The MYCIN experiment of the Standford Heuristic Programming Project" ; ADDISON-WESLEY, 1984
- [2] HAYES-ROTH, F. LENAT, D.B. AND WATERMAN, D.A.; "BUILDING EXPERT SYSTEMS"; ADDISON-WESLEY ; 1983 .
- [3] GREENWELL, MIKE; "KNOWLEDGE ENGINEERING FOR EXPERT SYSTEMS."; ELLIS HORWOOD LIMITED; 1988
- [4] JACKSON, PETER; "INTRODUCTION TO EXPERT SYSTEMS."; ADDISON-WESLEY; 1986
- [5] RAUCH-HINDIN, W.B. "A GUIDE TO COMMERCIAL ARTIFICIAL INTELLIGENCE." ; PRENTICE HALL; 1988
- [6] KELLER, R. "EXPERT SYSTEM TECHNOLOGY development & application" ; PRENTICE-HALL ; 1987
- [7] ROLSTON D, W. "PRINCIPLES OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS DEVELOPMENT"; MCGRAW-HILL; 1988
- [8] SILVERMAN, BARRY G. "EXPERT SYSTEMS FOR BUSINESS" ADDISON-WESLEY; 1987
- [9] LEVINE, ROBERT I. "A COMPREHENSIVE GUIDE TO AI AND EXPERT SYSTEMS USING TURBO PASCAL." MCGRAW-HILL ; 1988
- [10] SMITH, JERRY D. "AN INTRODUCTION TO SCHEME" ; PRENTICE-HALL; 1988
- [11] ABELSON, HAROLD "STRUCTURE AND INTERPRETATION OF COMPUTER PROGRAMS." ; MIT PRESS; 1985
- [12] HARMON, PAUL "EXPERT SYSTEMS artificial intelligence in business"; JOHN WILEY & SONS ; 1985

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวกที่ 1

อธิบายศัพท์

ในภาคผนวกนี้จะกล่าวถึง ศัพท์ที่เกี่ยวข้องกับการบิน ซึ่งจะอธิบายถึงความหมายพอสังเขป โดยศัพท์ที่นำมาอธิบายนี้เป็นเพียงบางส่วนเท่านั้น

SECTOR ช่วงการบินจากสถานีหนึ่ง (DEPARTURE) ไปอีกสถานีหนึ่ง (DESTINATION) ในแต่ละเที่ยวบินหนึ่งอาจจะมี 1 หรือ 2 SECTOR เช่น TG 913 บินจากลอนดอน มากรุงเทพฯ จะมี 2 sector คือ จากลอนดอนมา โคเปนเฮเก้น และ จากโคเปนเฮเก้น มากรุงเทพฯ เป็นวัน

GST (minimum ground stop) เป็นเวลาที่จอดพักเพื่อนำเครื่องบินขึ้นบินต่อไปยัง sector ถัดไป หรือในเที่ยวบินถัดไป โดยเครื่องแต่ละแบบจะมีช่วงเวลานี้แตกต่างกันออกไป GST นี้แบ่งออกเป็น TRANSIT และ TURN AROUND

TRANSIT เป็นเวลาที่เครื่องบินถึงจุดหมายปลายทางแล้วพักเช็ครีกร์, ทาความสะอาด, เติมน้ำมันเพื่อจะบินใน sector ถัดไป ใช้กับเที่ยวบินที่มี 2 sector

TURN AROUND คือเวลาที่ต้องการเช็ครีกร์, ทาความสะอาด, เติมน้ำมันคล้ายกับ TRANSIT แต่อาจเป็นเวลาเท่ากับหรือมากกว่า TRANSIT เช่น TG 403 บิน กรุงเทพฯ - สิงคโปร์ TG404 บิน สิงคโปร์ - กรุงเทพฯ สำหรับเที่ยวบิน TG 404 ยังใช้เครื่องบินแบบเดิมจาก TG403 จึงต้องมีช่วงเวลา TURN AROUND เพื่อเช็คสภาพความพร้อมของเครื่องบิน

ETD (Estimate time departure) เป็นเวลาโดยประมาณที่เที่ยวบินนั้นจะต้องเดินทางออกจากสนามบินนั้น มักใช้กับเครื่องที่มีการ DELAY

ETA (Estimate time Arrival) เหมือนกับ ETD คือเป็นเวลาโดยประมาณที่เที่ยวบินนั้นจะต้องเดินทางมาถึงสนามบินตามตารางบิน

INDEX คือ มาตรฐานวัดเวลาการทำงานของนักบินและลูกเรือ ตั้งแต่ช่วงเวลาที่นักบิน

หรือลูกเรือเดินทางออกจากสนามบินสู่ที่พัก (30 นาที หลังจากเครื่องถึง) มีเอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยกรมการขนส่งทางอากาศเพื่อใช้ในการคำนวณค่าไม่ต่ำกว่าเกณฑ์ใดๆ ทั้งสิ้น การกำหนดค่ามาตรฐานสูงสุดเอาไว้ 100 จะเปรียบเทียบกับชั่วโมง คือ นำไปใช้

1 index = 10 นาที

1 landing = 6 index

TAXING	คือช่วงเวลาที่เครื่องบินจะบินขึ้นเพื่อไป และหลังจากเครื่องบินถึงสนามบินจุดหมายปลายทาง (เครื่องบินเคลื่อนที่อยู่บน RUNWAY)
CREW	ลูกเรือที่ทำงานอยู่บนเครื่องบิน หมายถึง นักบิน (COCKPIT CREW) และพนักงานต้อนรับบนเครื่องบิน (CABIN CREW)
SLOT	เวลาน้อยที่สุดเพื่อจัดระยะห่างระหว่างเครื่องบิน 2 ลำ ในเส้นทางเดียวกัน
FACILITY	สิ่งอำนวยความสะดวก ของสนามบินหรือเมือง ๆ นั้น เช่น สนามบินมีระบบนำร่องหรือไม่ มีอุปกรณ์ที่ใช้กับเครื่องบิน ฯลฯ
REVENUE LOSS	รายได้ที่ทางการบินไทยต้องสูญเสียไป เนื่องจากเที่ยวบินนั้น ต้องยกเลิกเที่ยวบิน เปลี่ยนเส้นทางการบิน ฯลฯ
AIRBORN HOUR	ชั่วโมงบินของเครื่องบินที่บินอยู่ในอากาศ (นับจาก TAKE-OFF ถึง LANDING)
CHARTER COST	ค่าใช้จ่ายสำหรับการเช่าเครื่องบิน มาบินแทนเที่ยวบินที่เกิดปัญหาขึ้น
REST PAX	จำนวนผู้โดยสารที่ตกค้างอยู่ เนื่องจากการยกเลิกเที่ยวบิน หรือ เปลี่ยนเส้นทางบิน ฯลฯ
UTC	= UNIVERSAL COORDINATED TIME เป็นเวลามาตรฐานโลก (เดิมเรียก GMT) กรุงเทพฯจะมีเวลาดังจาก UTC = +7 ชั่วโมง
PAX CONVENIENCE	ความสะดวกสบายของผู้โดยสาร เช่น มีโรงแรมเตรียมให้ไว้เปล่า เมื่อเกิดกรณีติดขัดขึ้นกับเที่ยวบินนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
509	890718	1	hstna	bkk	1100	ruh	1835	195	b	3589	()
510	890718	1	hstna	ruh	1935	bkk	0240	150	a	3589	a
503	890417	1	hstac	bkk	1145	dha	1845	210	b	3378	()
509	890417	1	hstaa	bkk	1100	ruh	1835	225	b	3589	a
307	890717	1	hstab	bkk	0700	cmb	1815	189	a	1489	()
308	890717	1	hstab	cmb	1115	bkk	1430	177	b	1489	()
313	890718	1	hstab	bkk	0445	ccu	0710	210	b	1005	()
314	890718	1	hstab	ccu	0825	bkk	1040	200	b	1005	()
305	890417	1	hstgl	bkk	0750	rgn	0900	180	c	363	a
306	890417	1	hstgl	rgn	1000	bkk	1110	100	c	363	a
913	890417	1	hstgb	lhr	1000	cph	1200	310	a	611	()
913	890417	2	hstgb	cph	1300	bkk	2400	310	a	5376	()

ความหมายของคอลัมน์ A1-A12 (ตารางบิน)

A1 = เลขของเที่ยวบิน (Flight Number)

A2 = วันที่บิน (Flight date)

A3 = SECTOR (ความหมายอยู่ในภาคผนวก ก)

A4 = เลขทะเบียนของเครื่องบิน

A5 = สถานที่เดินทาง (DEPARTURE)

A6 = เวลาที่เดินทาง

A7 = จุดหมายที่จะไป

A8 = เวลาที่ไปถึงจุดหมายปลายทาง

A9 = ผู้โดยสารทั้งหมด

A10 = ภาพพจน์ที่ติดต่อสายการบิน/การแข่งขันระหว่างสายการบิน

A11 = ระยะทางทั้งหมด

A12 = ผู้โดยสารคนสำคัญ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
hstaa	dic	y	()	()	y	y	()	()	y	6000	130	233
hstab	dic	n	8907181000	8907191200	y	n	8907160900	8907170100	n	6000	130	233
hstac	dic	y	()	()	y	y	()	()	n	6000	130	233
hstaa	ab6	n	8907161600	8907170900	n	n	8907171000	8907171300	y	3500	100	247
hstab	ab6	y	()	()	y	y	()	()	n	3500	100	247
hstac	ab6	y	()	()	n	n	8907191200	8907210030	n	3500	100	247
hstgk	ab4	y	()	()	y	y	()	()	y	2300	100	223
hstgl	ab4	y	()	()	y	y	()	()	y	2300	100	223
hstgn	ab4	n	8907192100	8907210900	y	y	()	()	n	2300	100	223
hstaz	ab3	y	()	()	y	y	()	()	y	1600	100	195

ฐานข้อมูลเกี่ยวกับเครื่องบิน

ความหมายของคอลัมน์ B1-B13 (ฐานข้อมูลเครื่องบิน)

- B1 = เลขทะเบียนของเครื่องบิน
- B2 = แบบของเครื่องบิน
- B3 = มีเครื่องบินพร้อมหรือไม่
- B4 = เวลาเริ่มต้นที่เครื่องบินพร้อม
- B5 = เวลาสิ้นสุดที่เครื่องบินพร้อม
- B6 = มีลูกเรือพร้อมหรือไม่ (available)
- B7 = มีลูกเรือสำรองหรือไม่ (standby)
- B8 = เวลาเริ่มต้นที่ลูกเรือ standby
- B9 = เวลาสิ้นสุดที่ลูกเรือ standby
- B10 = มีลูกเรือที่จะบินกับเครื่องลำนั้นหรือไม่ (qualified)
- B11 = ระยะทางบินที่เครื่องบินลำนั้นสามารถบินได้
- B12 = เวลาที่ต้องใช้สำหรับตรวจเช็คเครื่อง
- B13 = บรรทุกผู้โดยสารมากที่สุด

c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24
bkk	y	()	()	y	()	()	y	()	()	y	()	()	y	()	()	1	n	8907181200	8907191945	n	8907161300	8907161500	y
fra	y	()	()	y	()	()	n	8907121000	8907122000	y	()	()	n	8907181600	8907182000	4	y	()	()	n	8907181700	8907182300	n
ml	y	()	()	n	8907151200	8907181000	y	()	()	n	8907181000	8907191000	y	()	()	7	y	()	()	n	8907171200	8907181000	y
osa	n	8907161700	8907181000	n	8907170900	8907171900	y	()	()	y	()	()	n	8907192300	8907201000	2	y	()	()	y	()	()	n
ath	y	()	()	y	()	()	n	8907181900	8907182300	y	()	()	y	()	()	3	y	()	()	n	8907181700	8907191300	y
vie	y	()	()	y	()	()	y	()	()	n	8907171000	8907191000	y	()	()	6	y	()	()	y	()	()	n
akl	y	()	()	y	()	()	y	()	()	y	()	()	y	()	()	2	y	()	()	y	()	()	y
sin	n	8907201900	8907202100	y	()	()	y	()	()	y	()	()	y	()	()	2	y	()	()	y	()	()	y
ruh	y	()	()	y	()	()	y	()	()	n	8907211800	8907221300	y	()	()	1	y	()	()	y	()	()	y
ccu	n	8907191800	8907211500	n	8907161700	8907211000	y	()	()	n	8907181600	8907182100	y	()	()	3	n	8907161000	8907181800	y	()	()	y
nrt	y	()	()	y	()	()	y	()	()	y	()	()	y	()	()	9	y	()	()	y	()	()	n

ฐานข้อมูลของสนามบิน

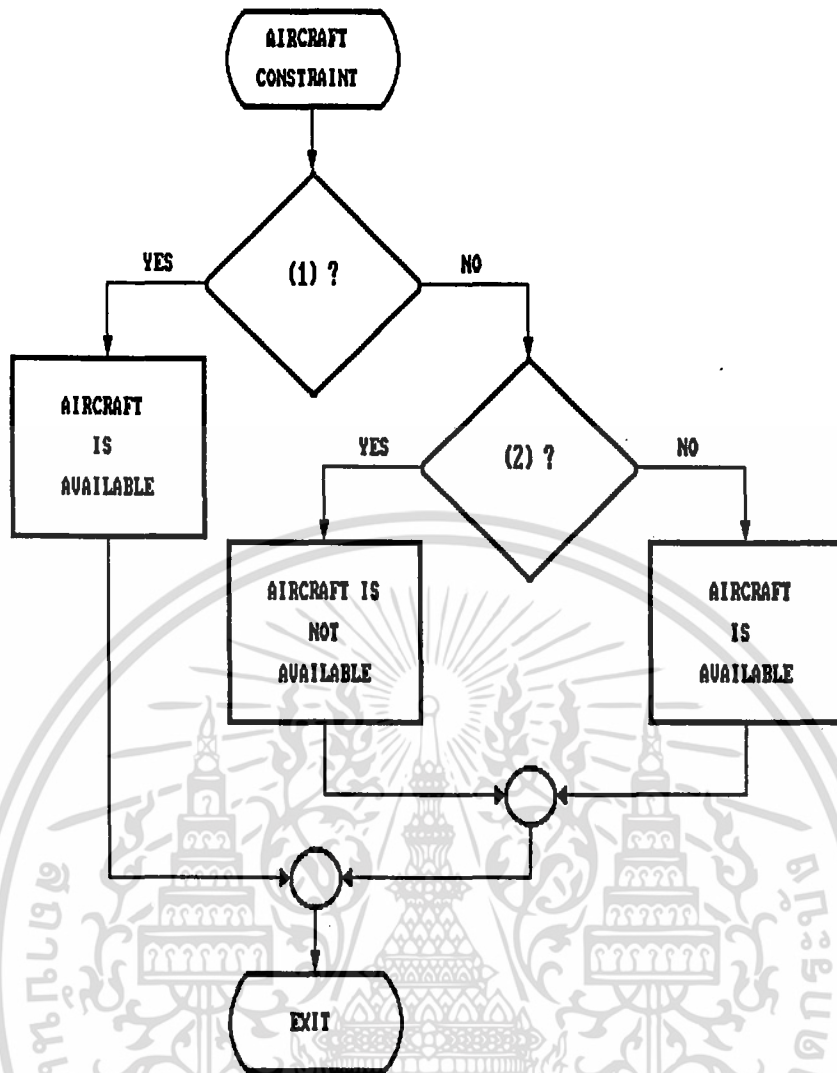
ความหมายของคอลัมน์ C1-C13 (สนามบิน)

- C1 = ชื่อของสนามบิน
- C2 = ปัญหาเกี่ยวกับน้ำมันเชื้อเพลิงที่สนามบินนั้น
- C3 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับน้ำมันเชื้อเพลิง
- C4 = เวลาสิ้นสุดที่มีปัญหาเกี่ยวกับน้ำมันเชื้อเพลิง
- C5 = ปัญหาเกี่ยวกับ slot ที่สนามบินนั้น
- C6 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับ slot
- C7 = เวลาสิ้นสุดที่มีปัญหาเกี่ยวกับ slot
- C8 = ปัญหาเกี่ยวกับสิ่งอำนวยความสะดวกที่สนามบินนั้น
- C9 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับความสะดวกที่สนามบินนั้น
- C10 = เวลาสิ้นสุดที่มีปัญหาเกี่ยวกับความสะดวกที่สนามบินนั้น
- C11 = ปัญหาเกี่ยวกับเคอร์ฟิวที่สนามบินนั้น
- C12 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับเคอร์ฟิว
- C13 = เวลาสิ้นสุดที่มีปัญหาเกี่ยวกับเคอร์ฟิว
- C14 = ปัญหาเกี่ยวกับ runway ที่สนามบินนั้น
- C15 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับ runway
- C16 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับ runway
- C17 = สภาพอากาศที่สนามบินนั้น
- C18 = ปัญหาเกี่ยวกับทางการ เมืองที่สนามบินนั้น
- C19 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับการเมือง
- C20 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับการเมือง
- C21 = ปัญหาเกี่ยวกับการบินที่สนามบินนั้น
- C22 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับการบิน
- C23 = เวลาเริ่มต้นที่มีปัญหาเกี่ยวกับการบิน

C24 = เป็นการขออนุญาตทางการบินการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

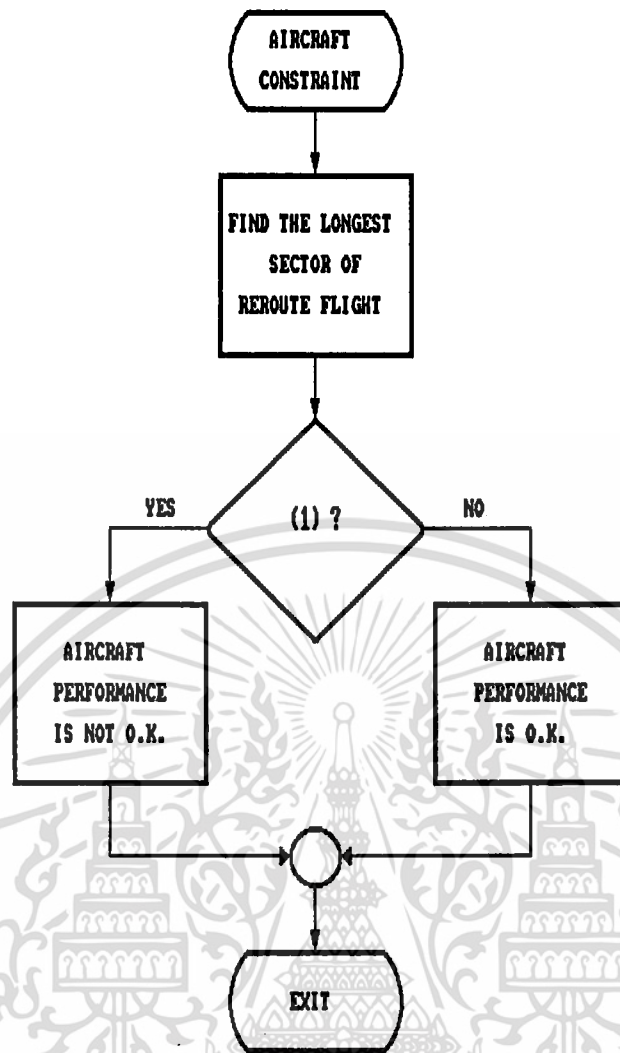


(1) = AIRCRAFT IS AVAILABLE (DOES NOT HAVE TECH. PROBLEM)

(2) = ETD DATE IS IN NON-AVAILABLE PERIOD

AC1 - AIRCRAFT CONSTRAINT

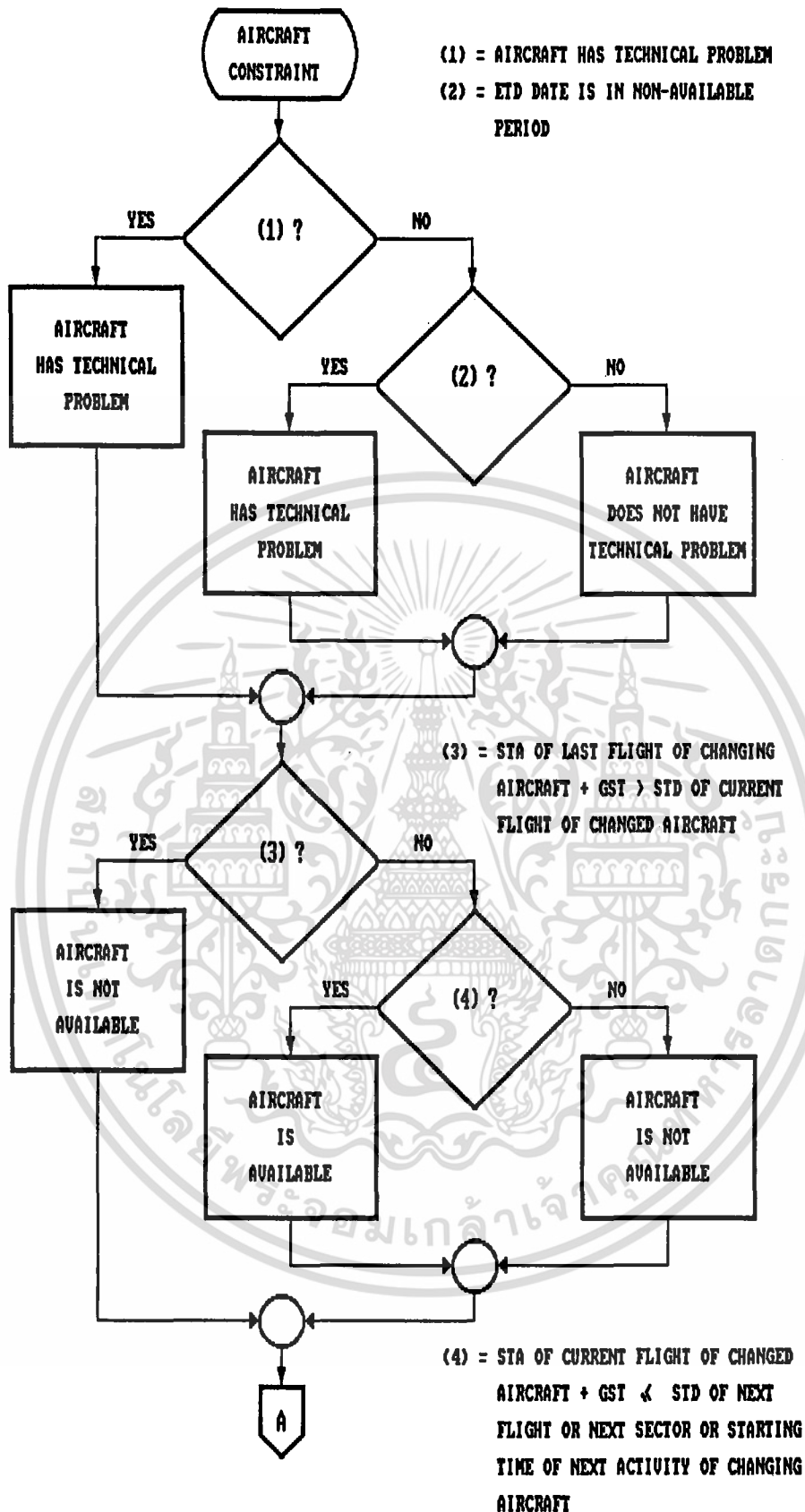
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1) = THE LONGEST FLIGHT HOUR > MAXIMUM FLIGHT HOURS

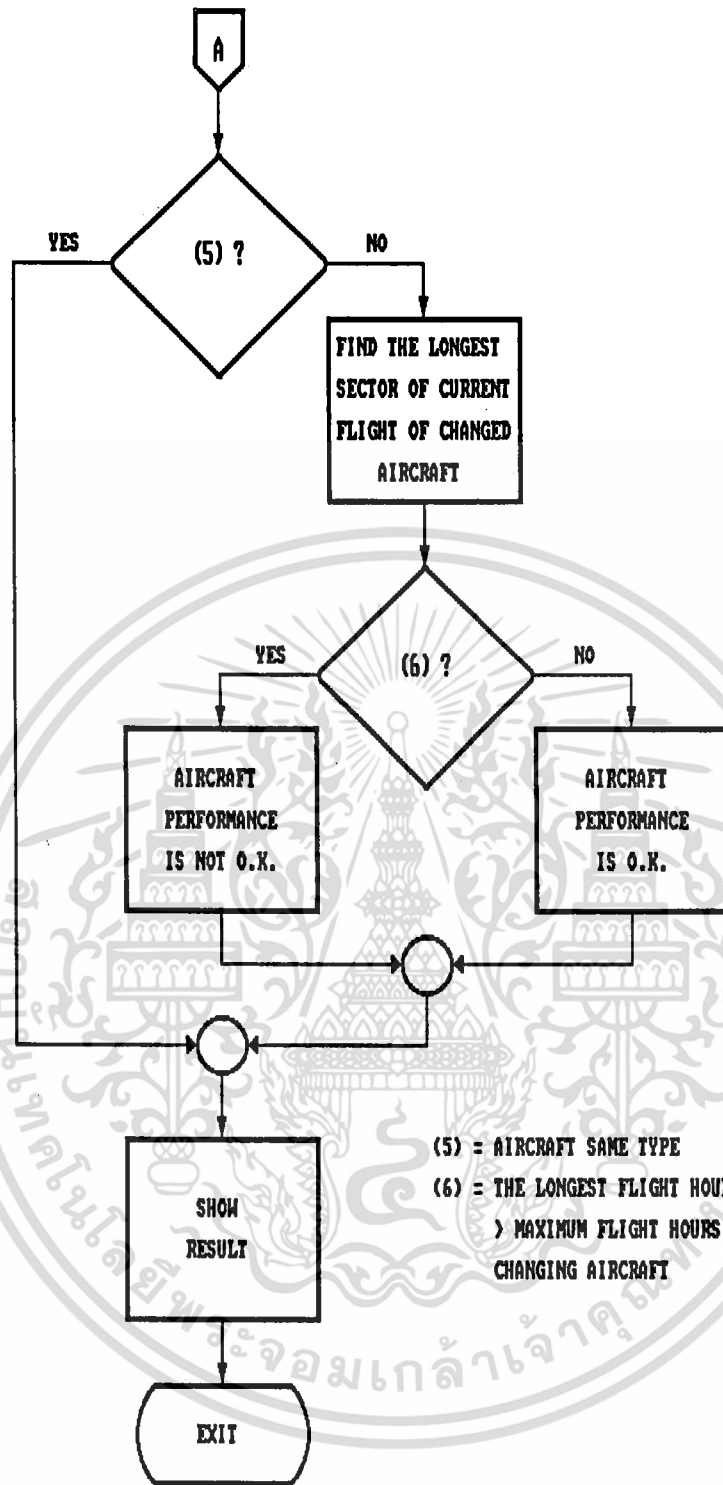
AC2 - AIRCRAFT CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



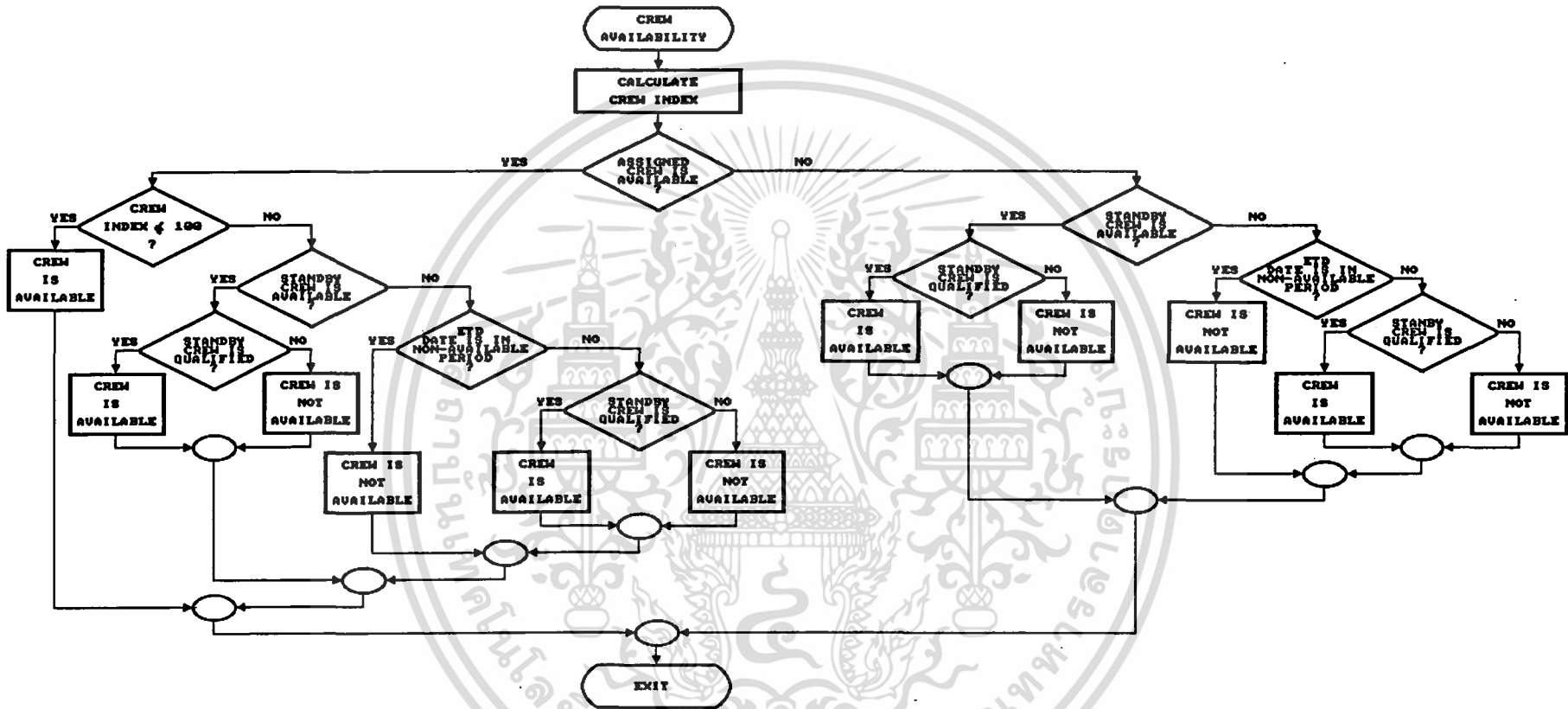
AC3 - AIRCRAFT CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

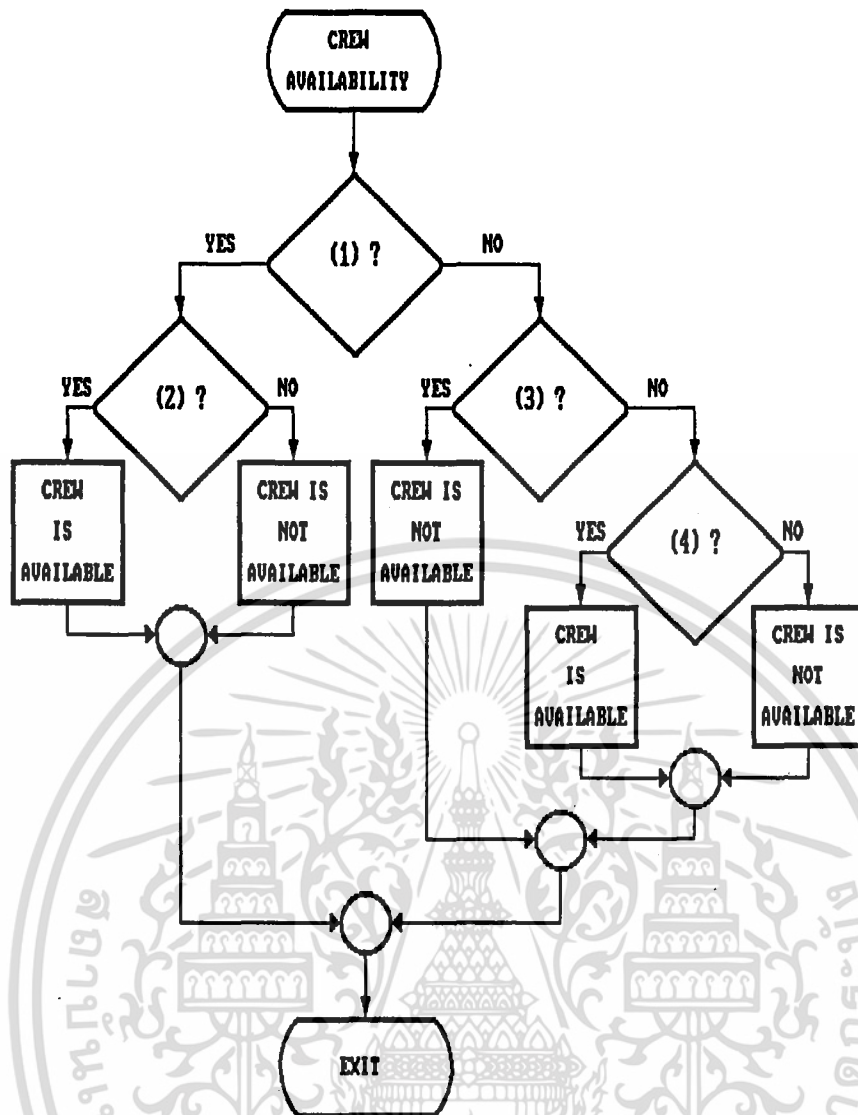


AC3 - AIRCRAFT CONSTRAINT (CONTINUED)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



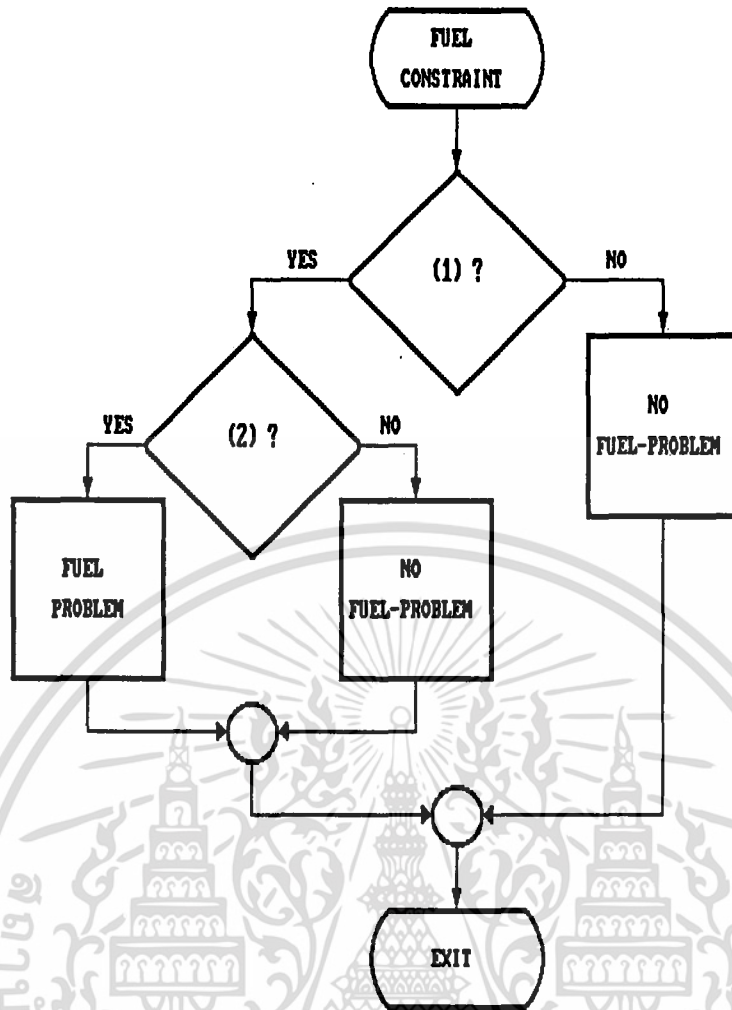
CW1 - CREM AVAILABILITY



- (1) = STANDBY CREW IS AVAILABILITY
- (2) = STANDBY CREW IS QUALIFIED
- (3) = ETD DATE IS IN NON-AVAILABLE PERIOD
- (4) = STANDBY CREW IS QUALIFIED

CW2 - CREW AVAILABILITY

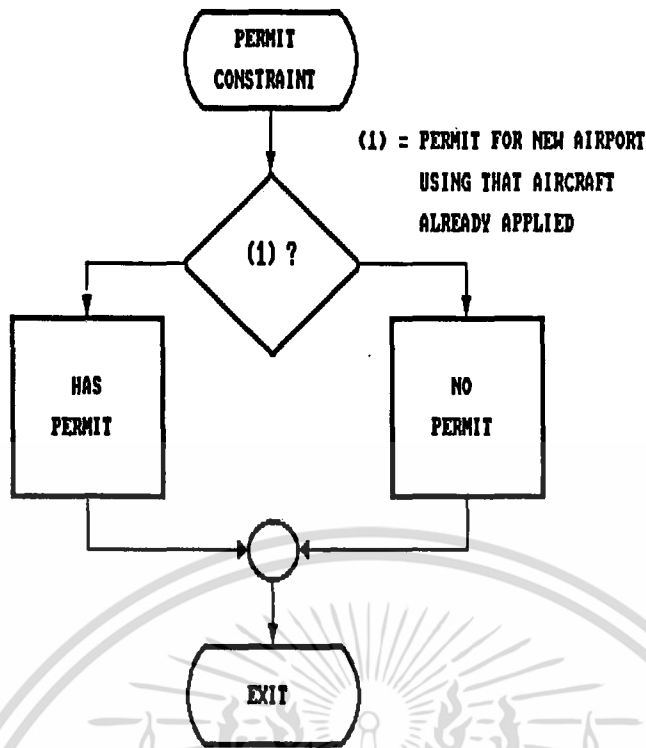
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



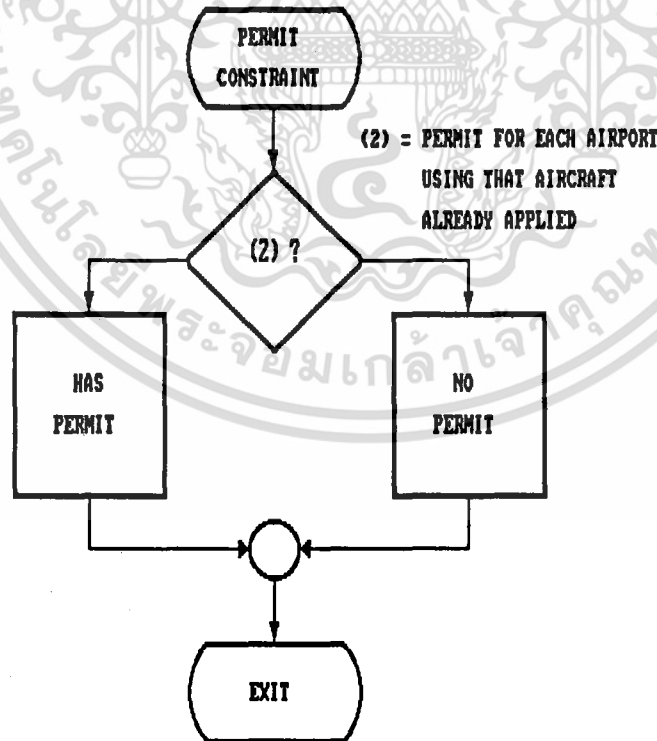
(1) = THERE IS FUEL PROBLEM AT AIRPORT
(2) = TRANSIT OR TURNAROUND PERIOD IS IN FUEL PROBLEM PERIOD

FU1 - FUEL CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

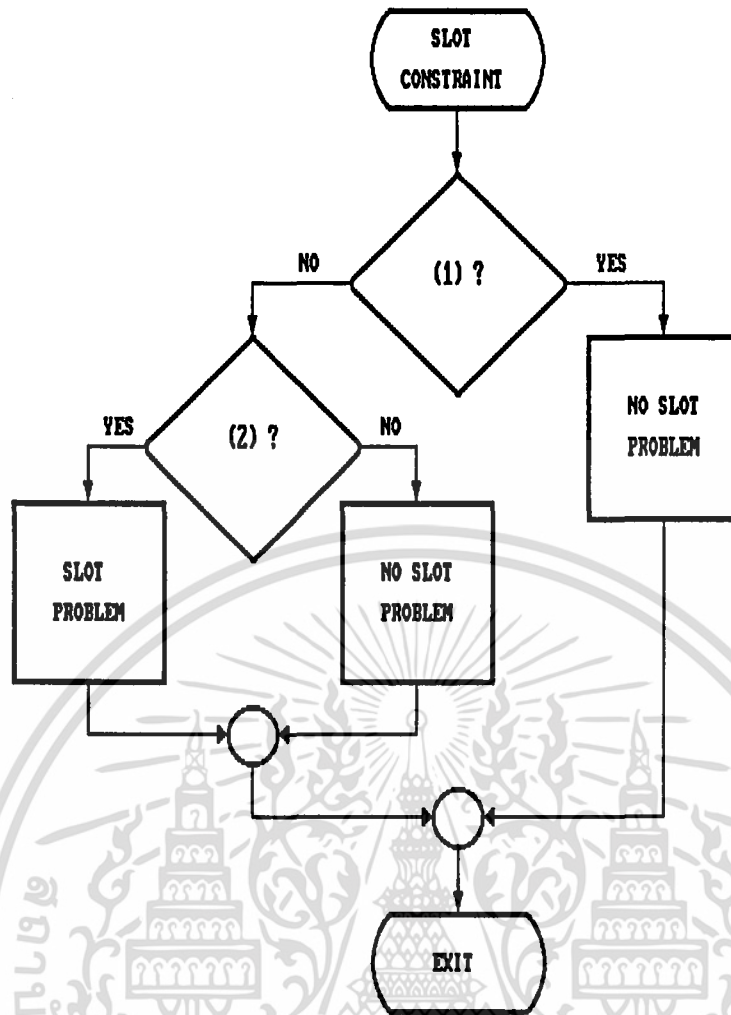


PM1 - PERMIT CONSTRAINT



PM2 - PERMIT CONSTRAINT

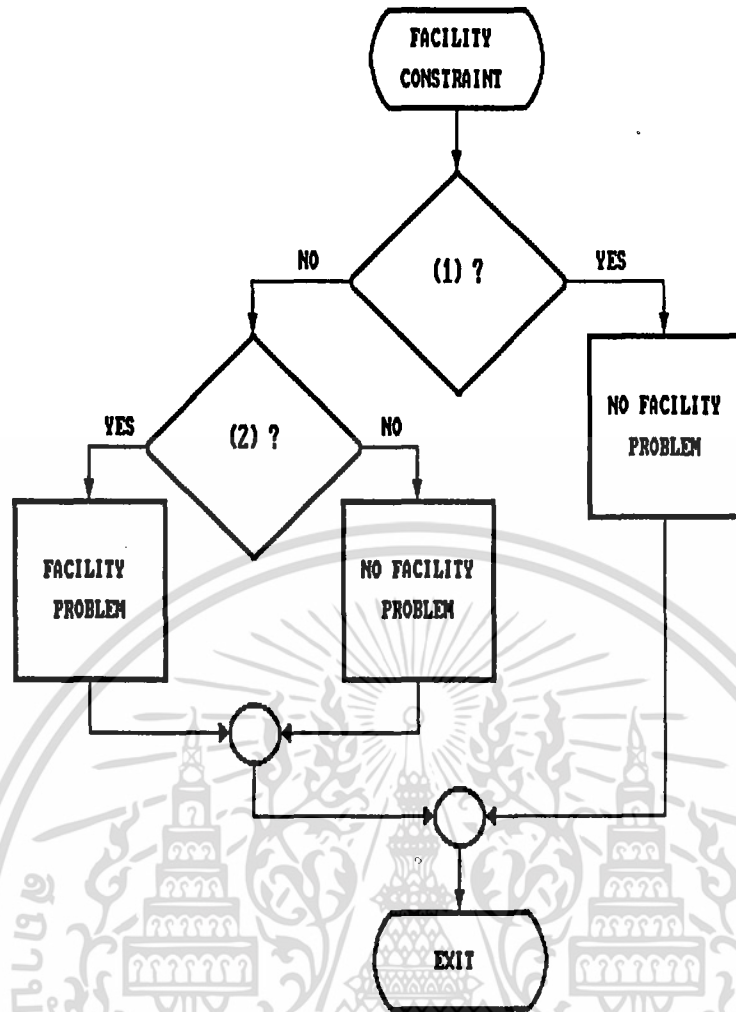
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1) = SLOT FOR AIRCRAFT TYPE IS AVAILABLE AT AIRPORT
(2) = TRANSIT OR TURNAROUND PERIOD IS IN SLOT PROBLEM PERIOD

SL1 - SLOT CONSTRAINT

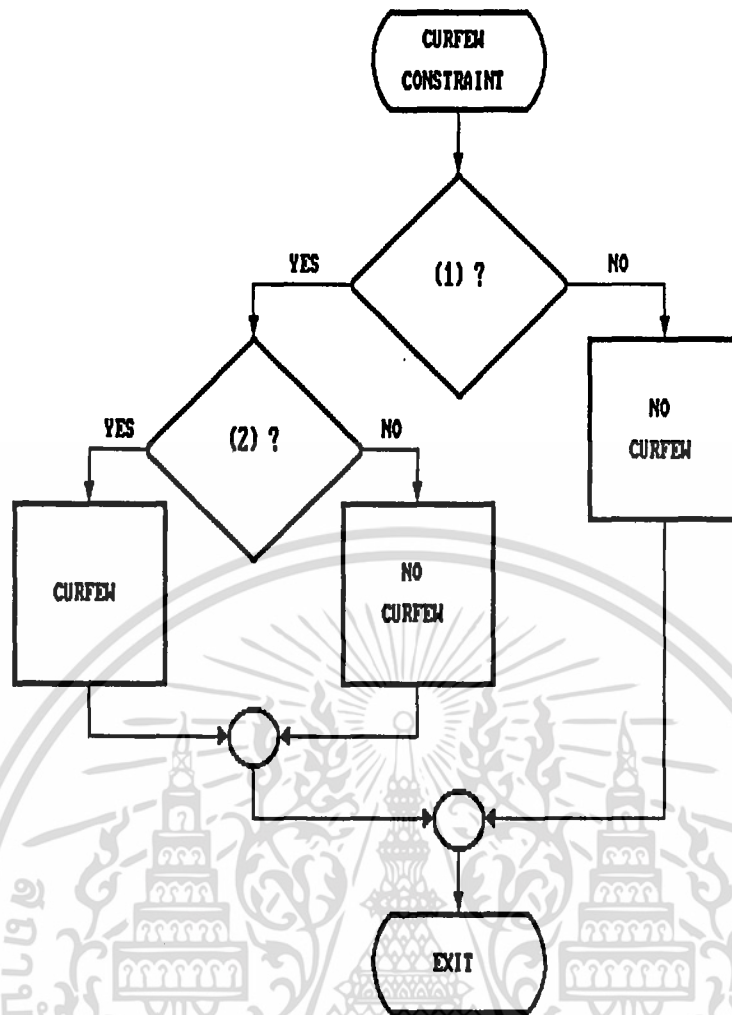
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1) = FACILITY FOR AIRCRAFT TYPE IS AVAILABLE AT AIRPORT
(2) = TRANSIT OR TURNAROUND PERIOD IS IN FACILITY PROBLEM PERIOD

FC1 - FACILITY CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

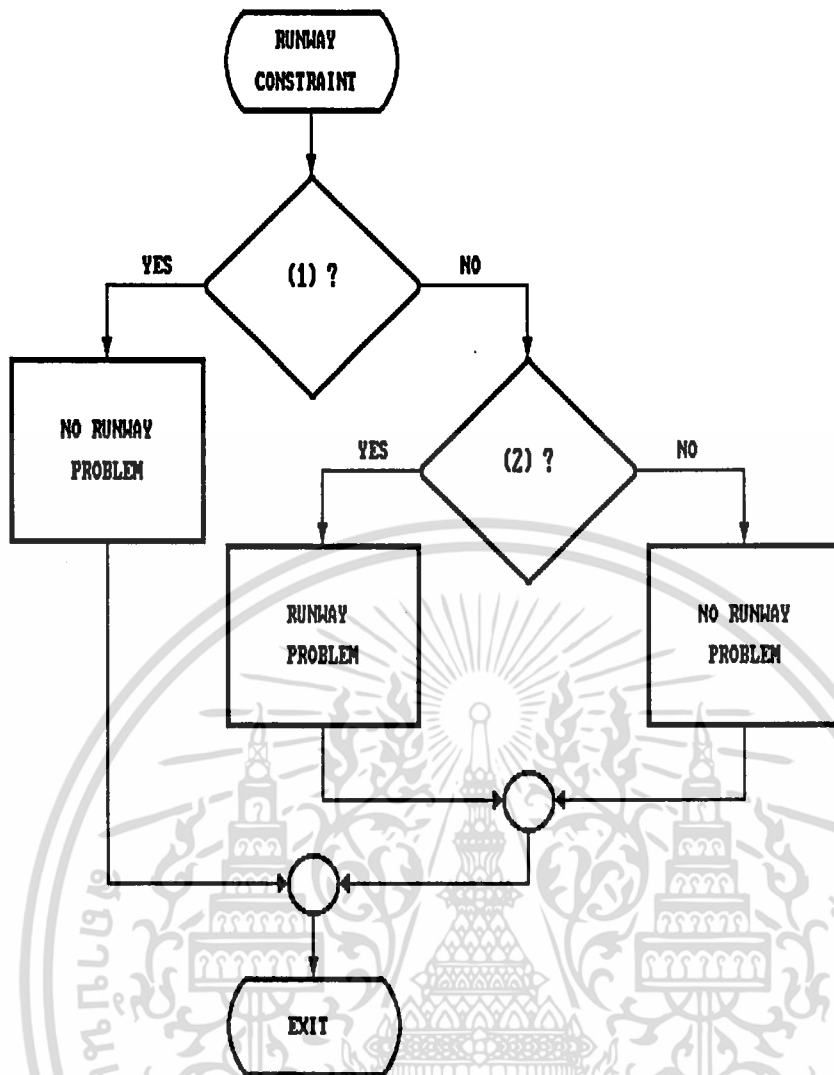


(1) = THERE IS CURFEW AT AIRPORT

(2) = TRANSIT OR TURNAROUND PERIOD IS IN CURFEW PERIOD

CU1 - CURFEW CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1) = RUNWAY IS AVAILABLE

(2) = TRANSIT OR TURNAROUND IS IN RUNWAY CLOSED PERIOD

RW1 - RUNWAY CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

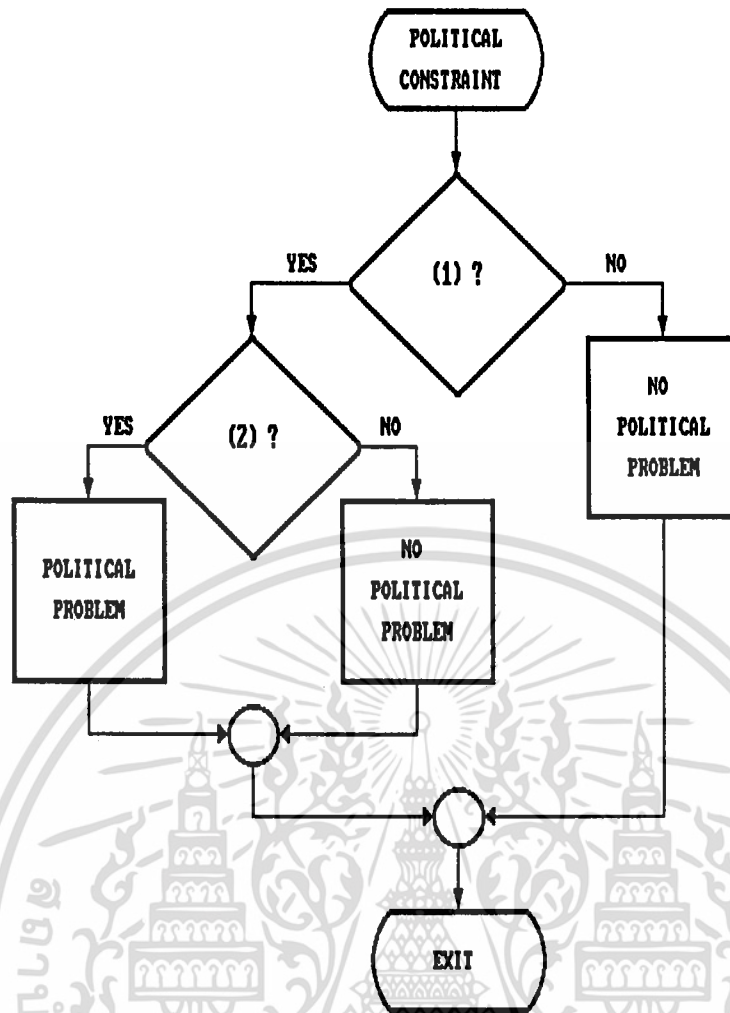
WX-INDEX

STATUS

- | | |
|---|----------------------|
| 1 | CLOSE-TO CLOSE |
| 2 | MARGINAL-TO-CLOSE |
| 3 | OPEN-TO-CLOSE |
| 4 | CLOSE-TO-MARGINAL |
| 5 | CLOSE-TO-OPEN |
| 6 | MARGINAL-TO-MARGINAL |
| 7 | OPEN-TO-MARGINAL |
| 8 | MARGINAL-TO-OPEN |
| 9 | OPEN-TO-OPEN |

WX1 - WEATHER CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

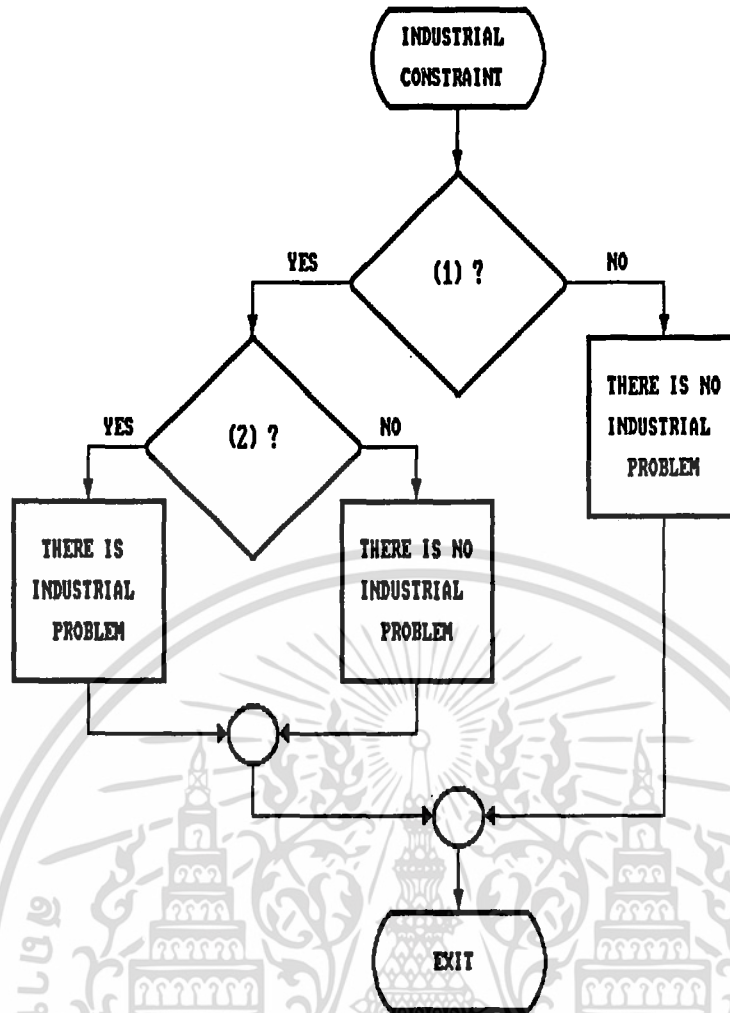


(1) = THERE IS POLITICAL PROBLEM AT AIRPORT

(2) = TRANSIT OR TURNAROUND PERIOD IS IN POLITICAL PROBLEM PERIOD

P01 - POLITICAL CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

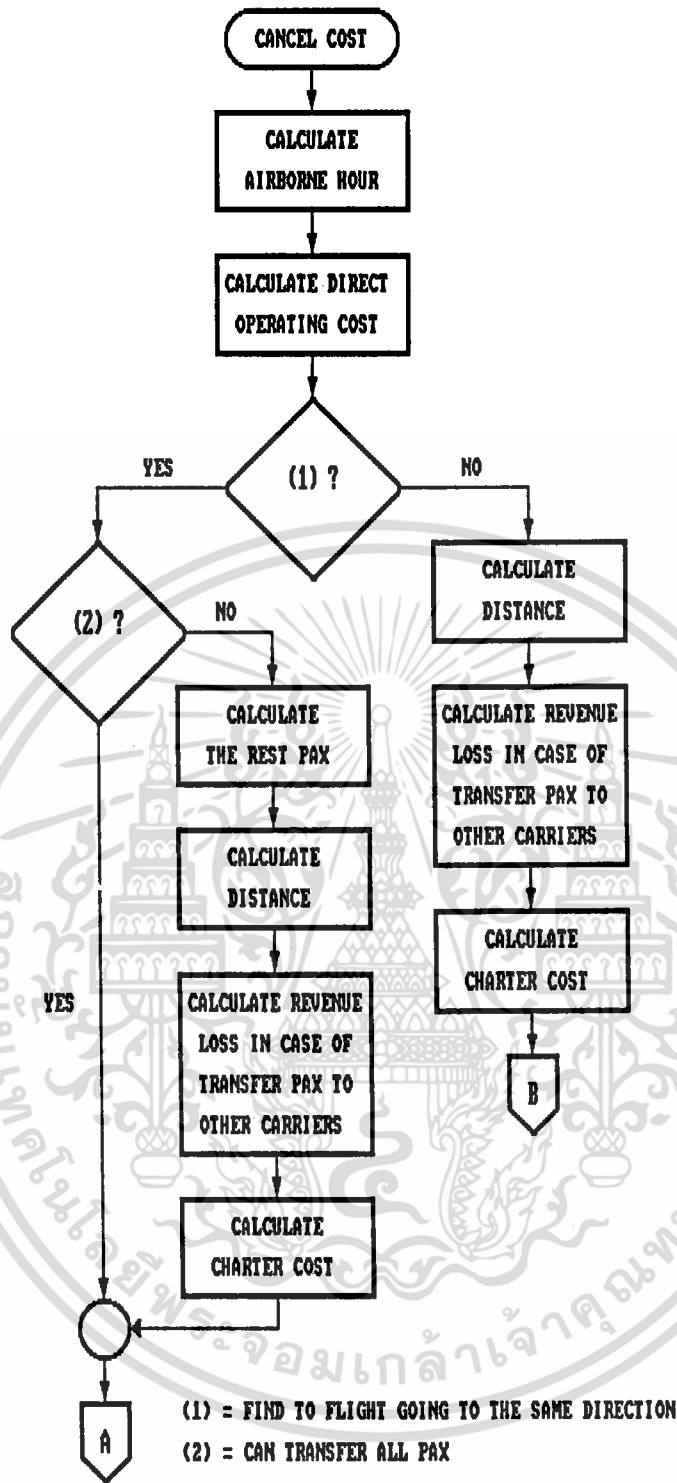


(1) = AIRPORT HAS INDUSTRIAL PROBLEM

(2) = TRANSIT OR TURNAROUND PERIOD IS IN INDUSTRIAL PROBLEM PERIOD

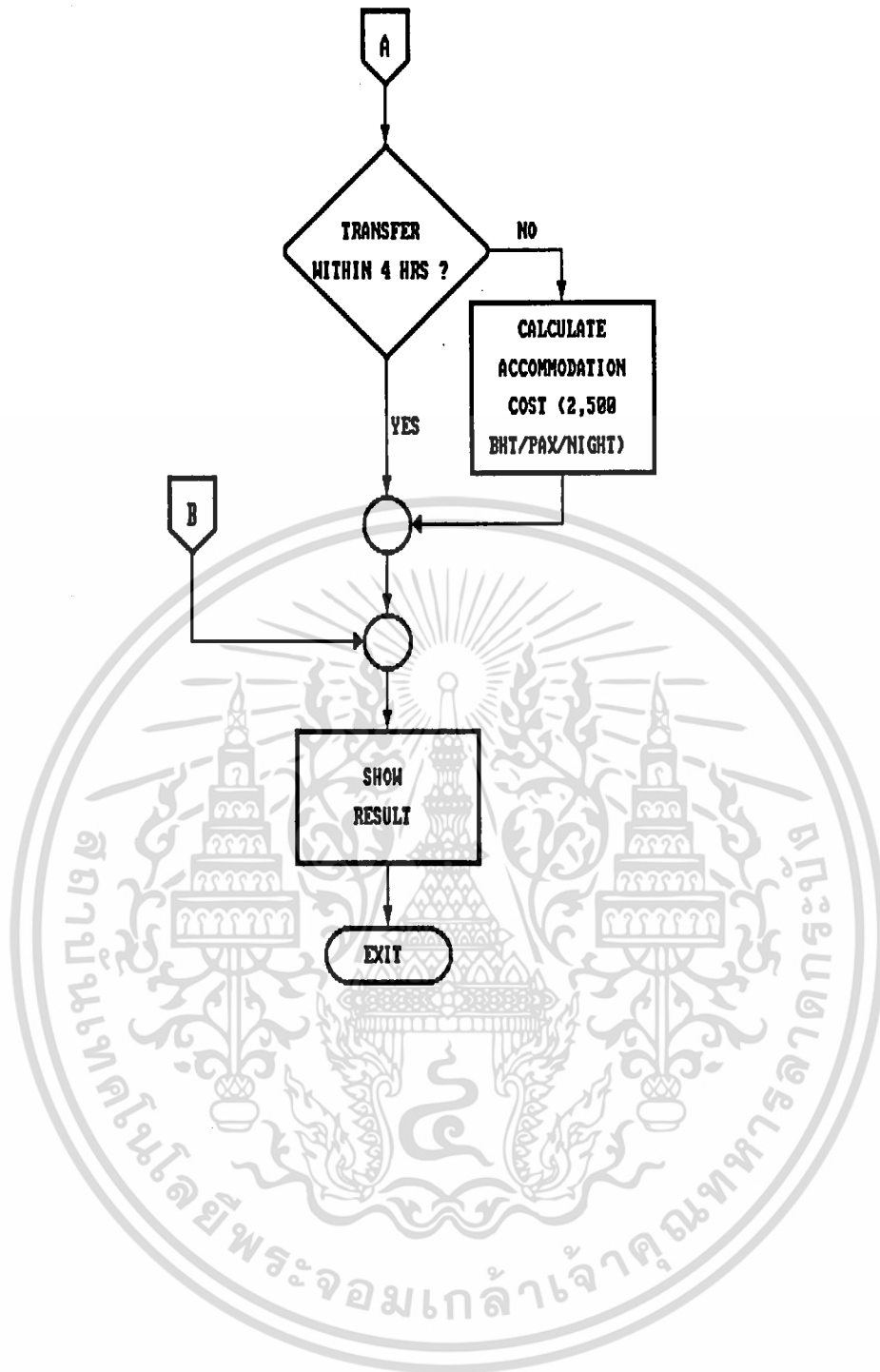
ID1 - INDUSTRIAL CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



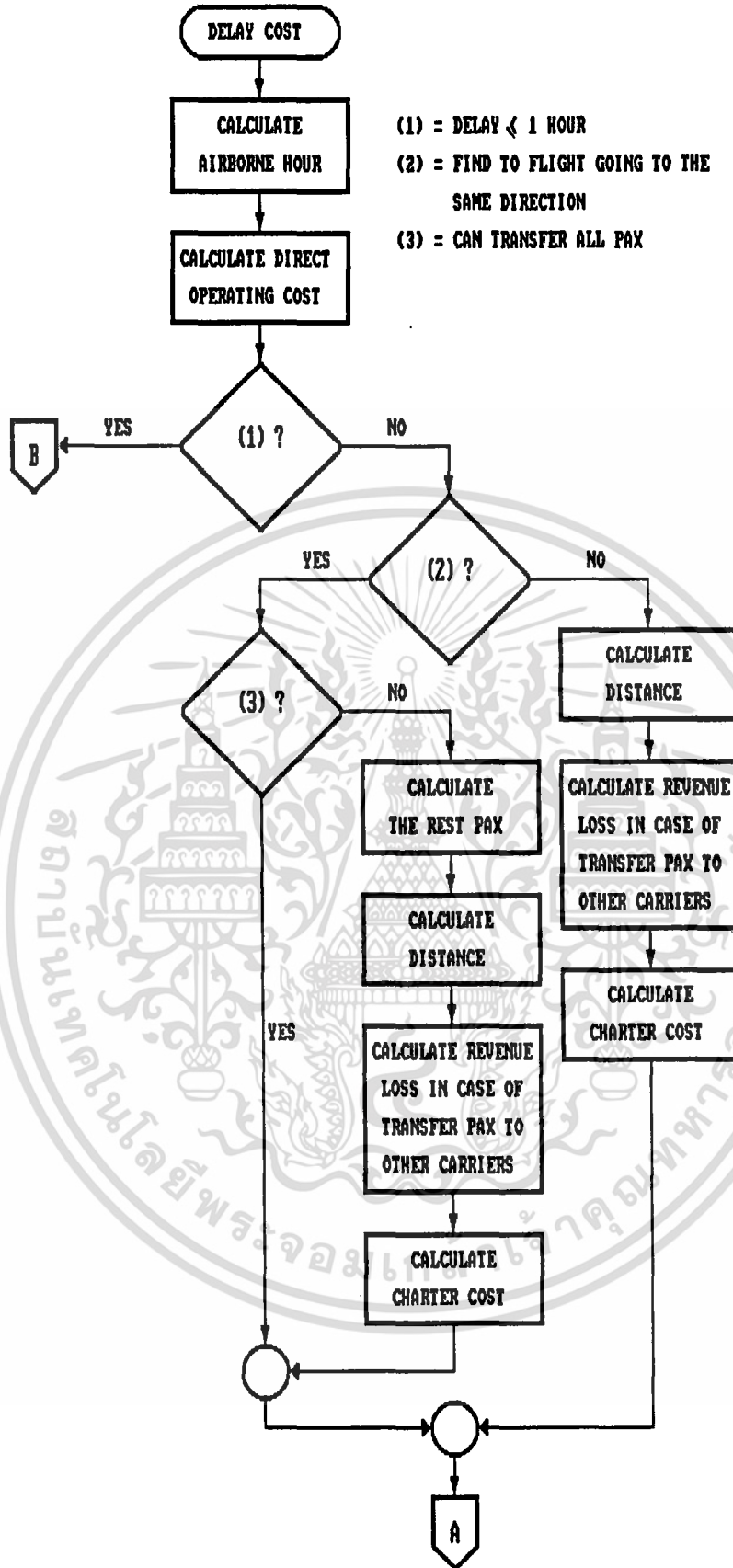
C1 - CANCEL COST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



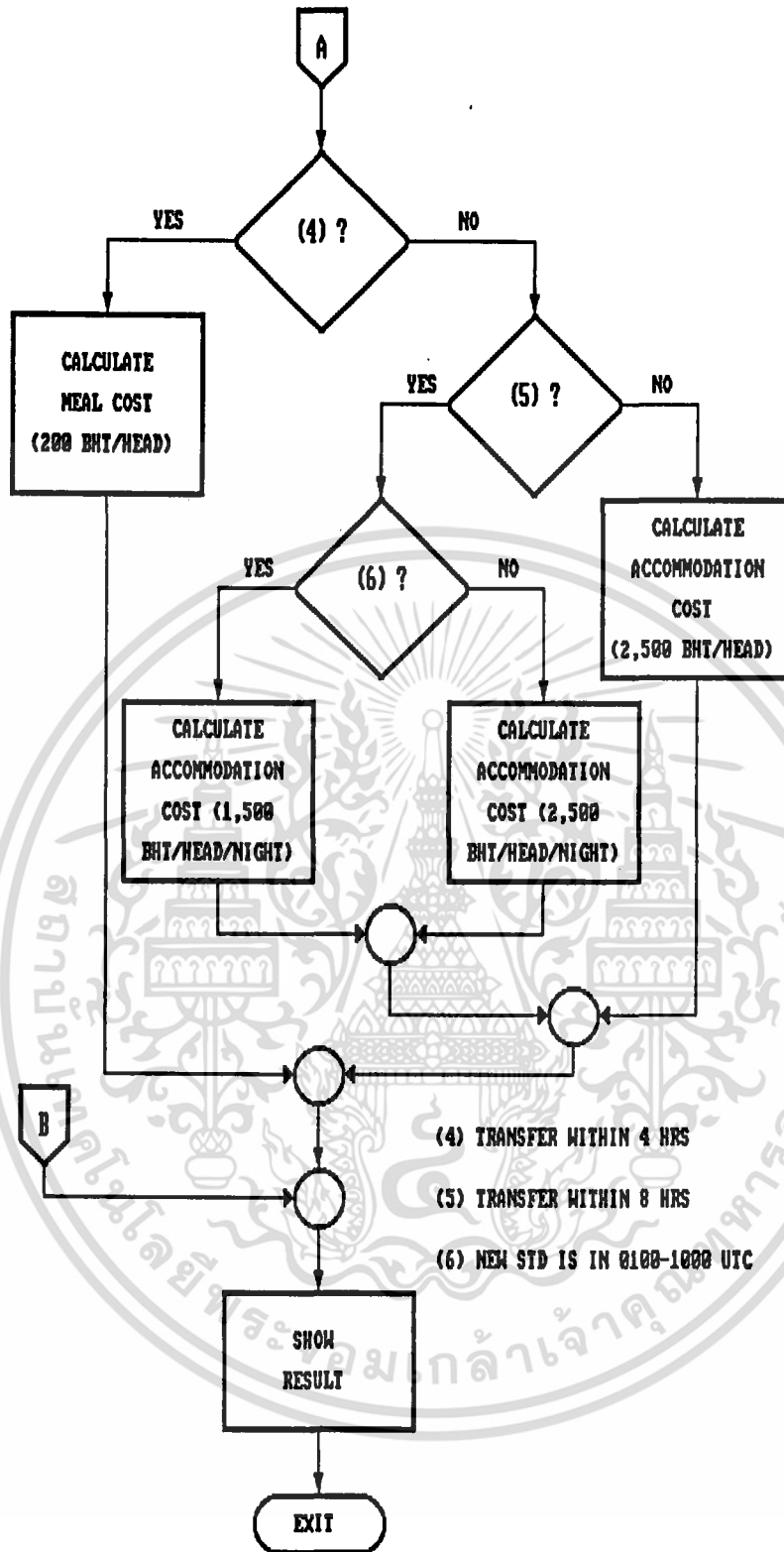
C1 - CANCEL COST (CONTINUED)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



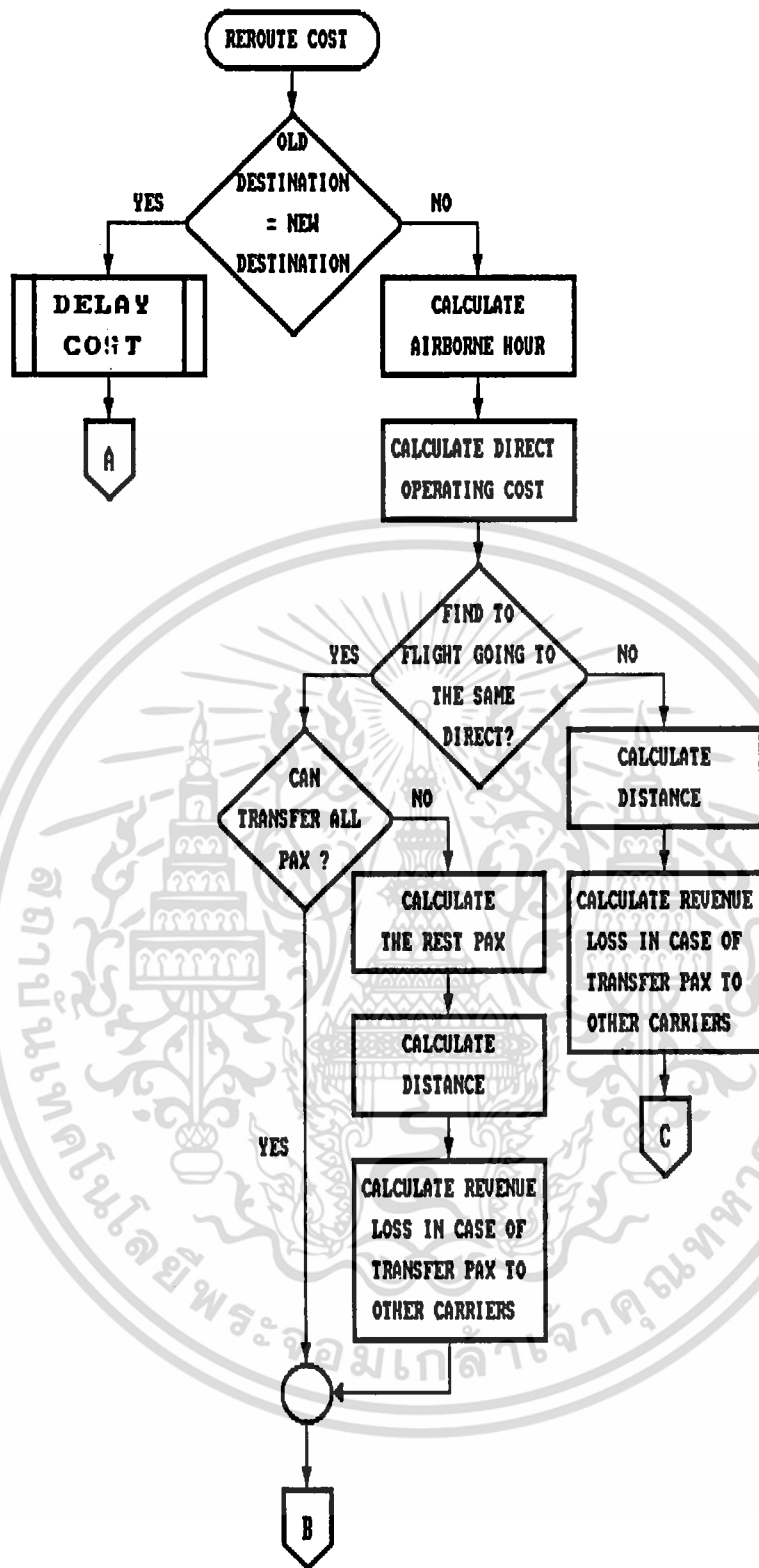
C2 - DELAY COST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



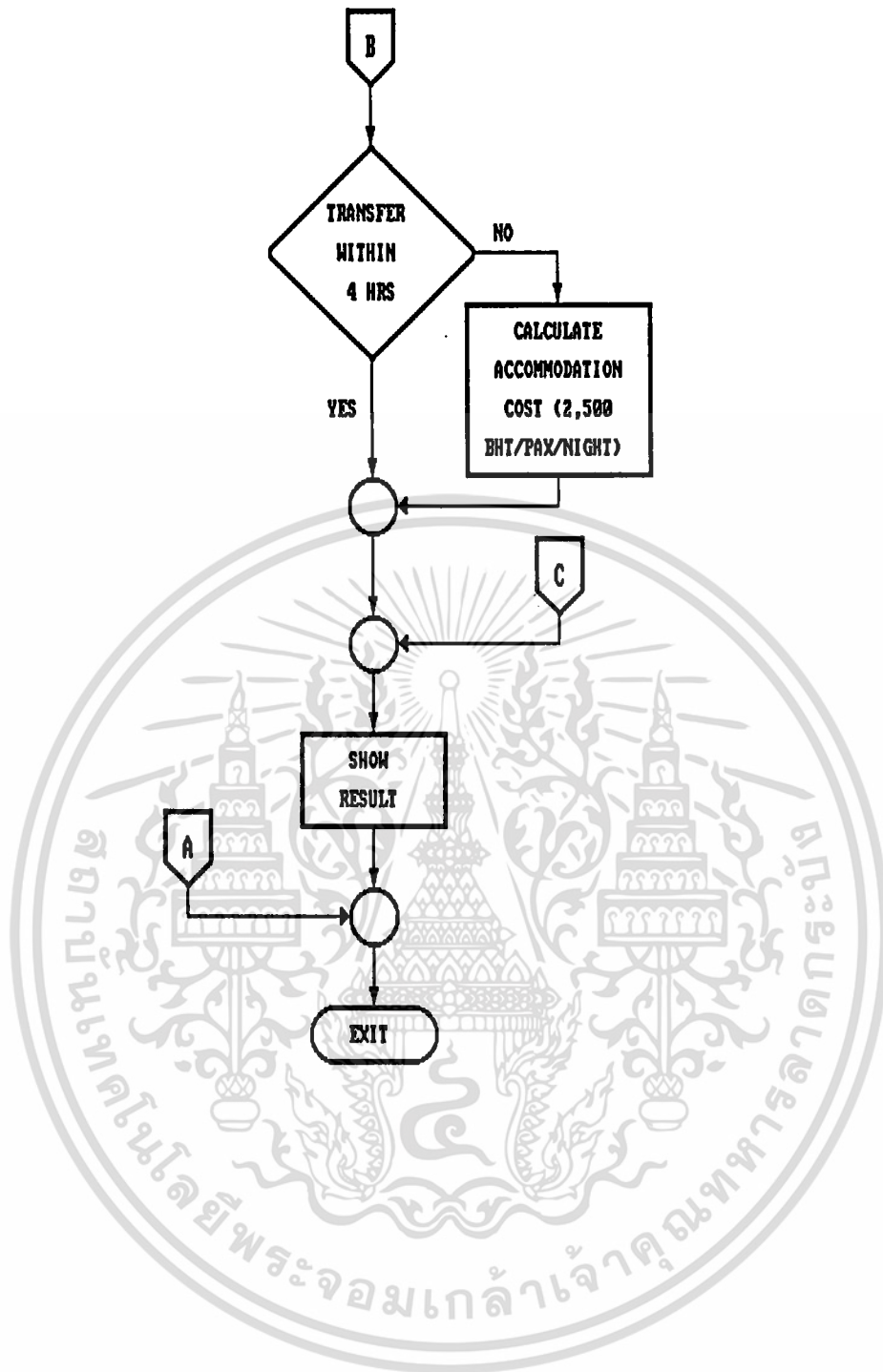
C2 - DELAY COST (CONTINUED)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



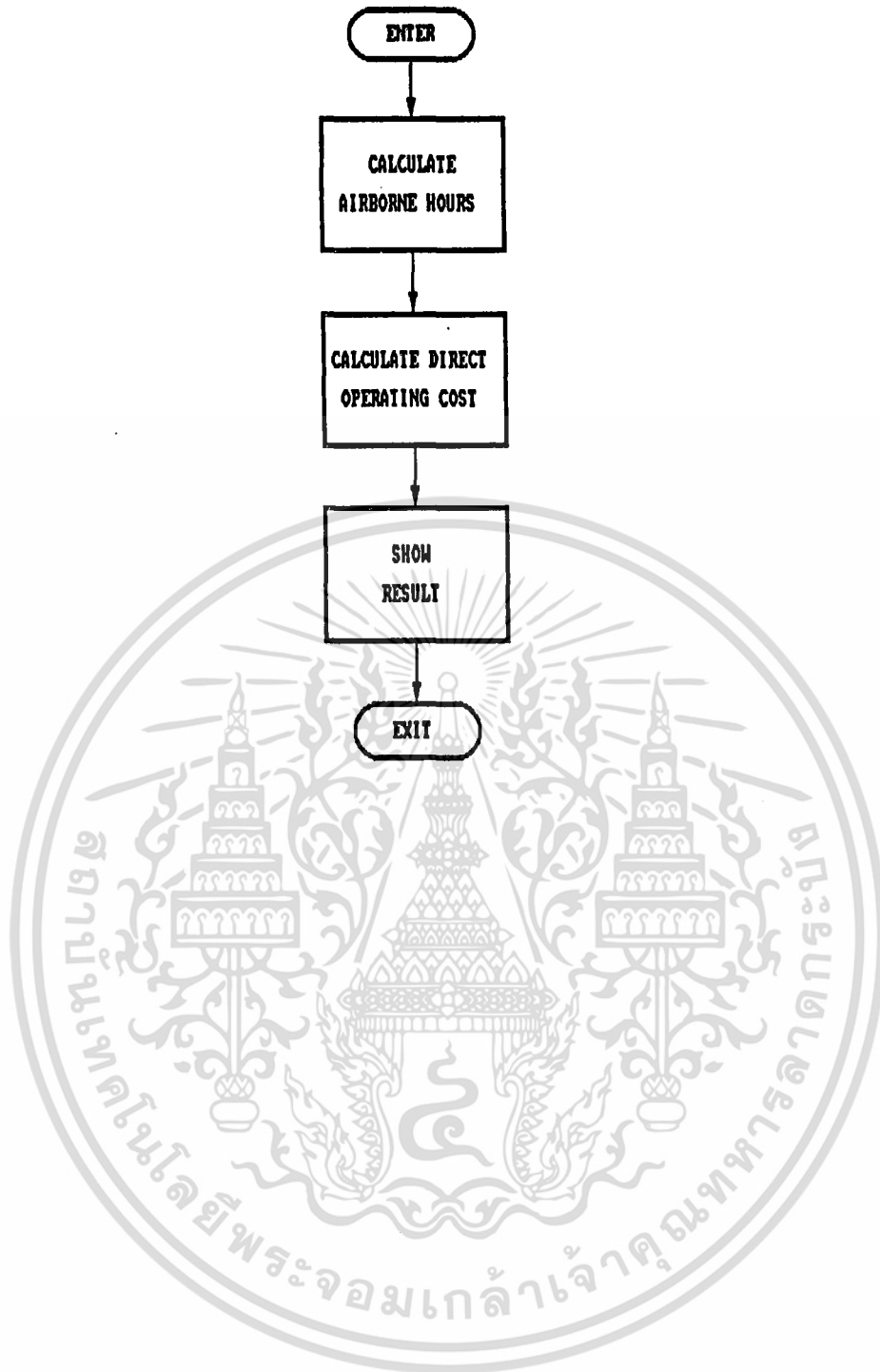
C3 - REROUTE COST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



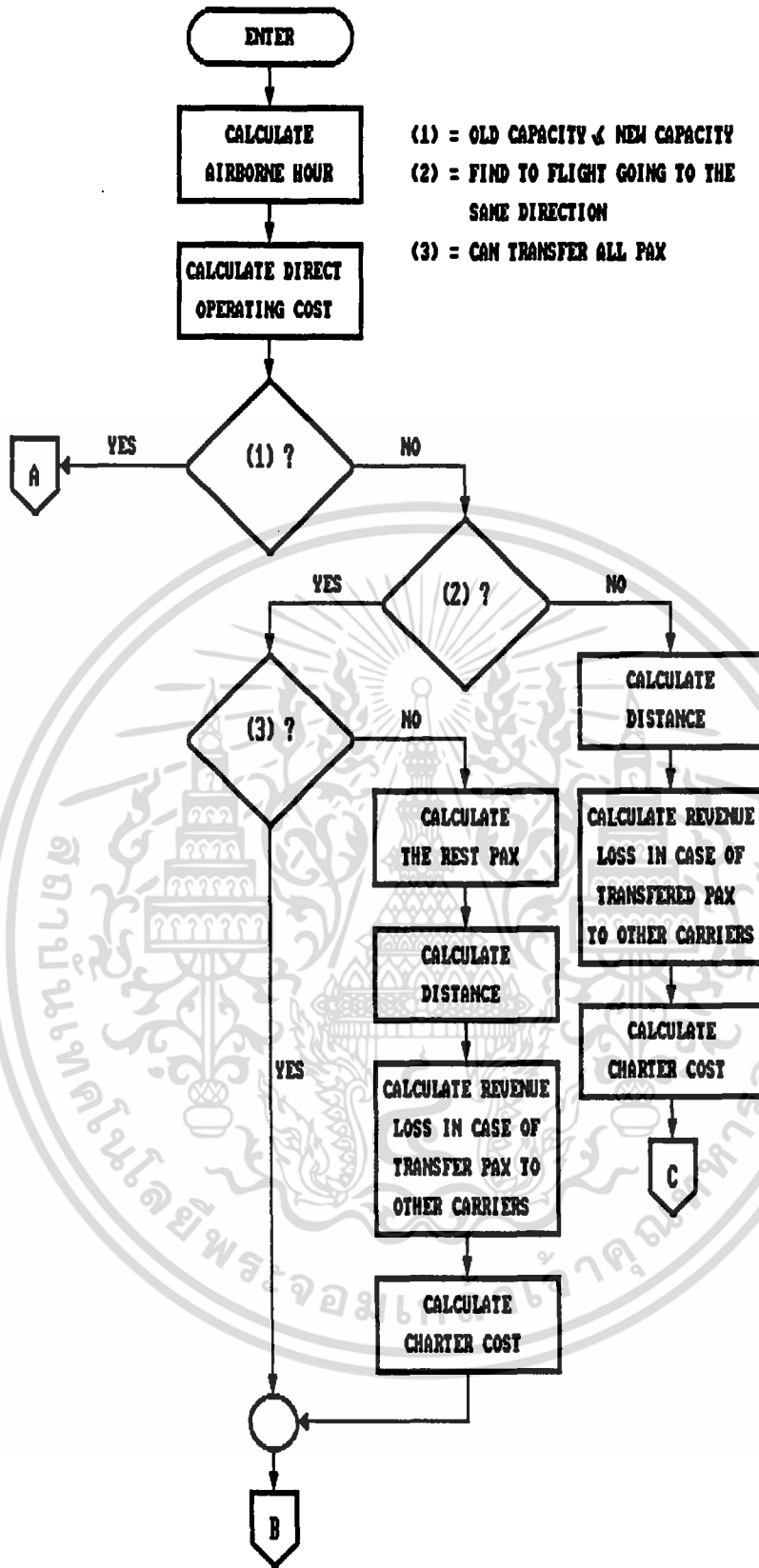
C3 - REROUTE COST (CONTINUED)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



C4 - COST OF CHANGE AIRPORT SAME TYPE-ON TIME

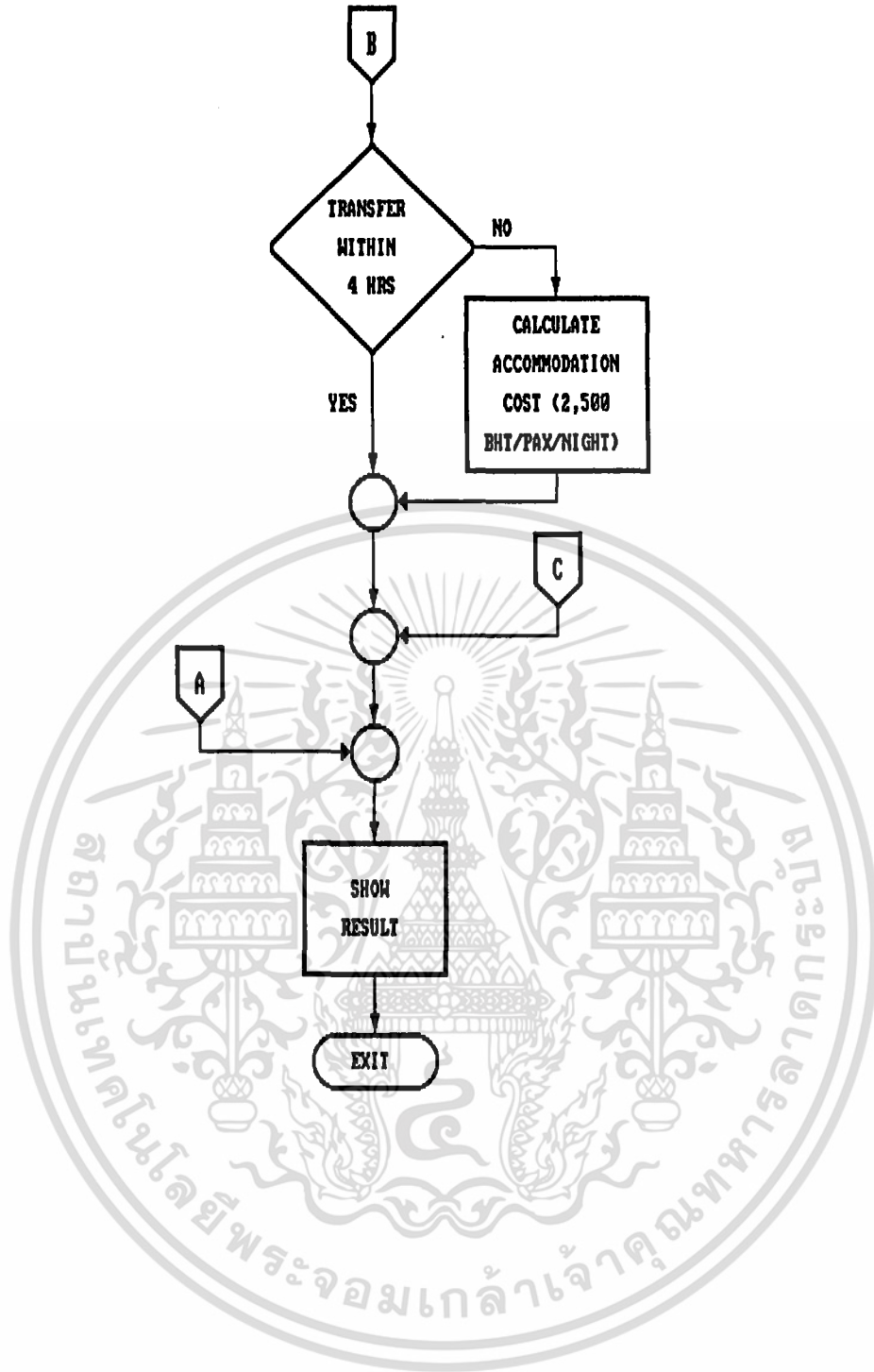
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(1) = OLD CAPACITY & NEW CAPACITY
(2) = FIND TO FLIGHT GOING TO THE SAME DIRECTION
(3) = CAN TRANSFER ALL PAX

C5 - COST OF CHANGE AIRPORT DIFFERENT TYPE-ONTIME

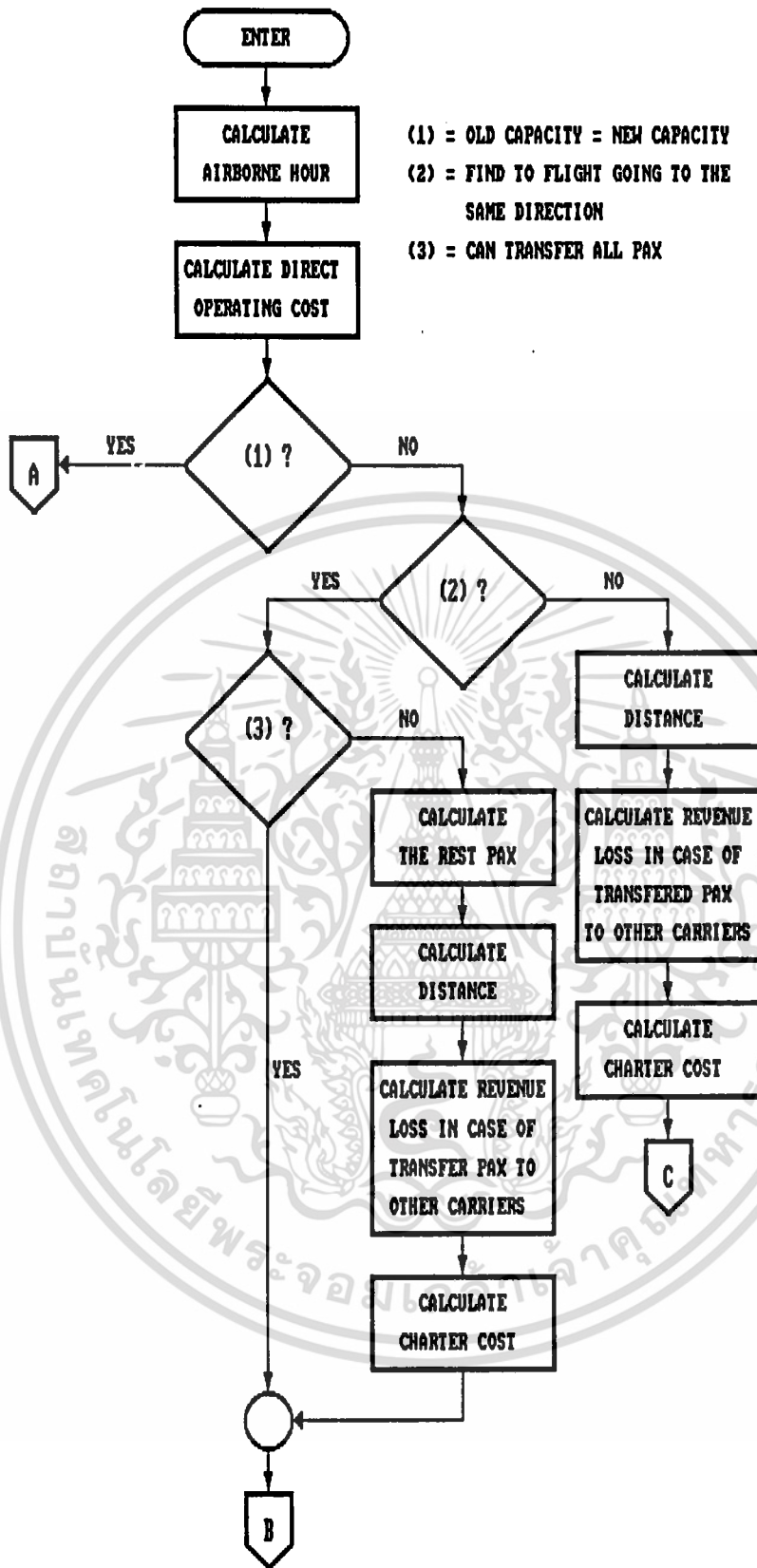
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



C5 - COST OF CHANGE AIRCRAFT

DIFFERENT TYPE-ON TIME (CONTINUED)

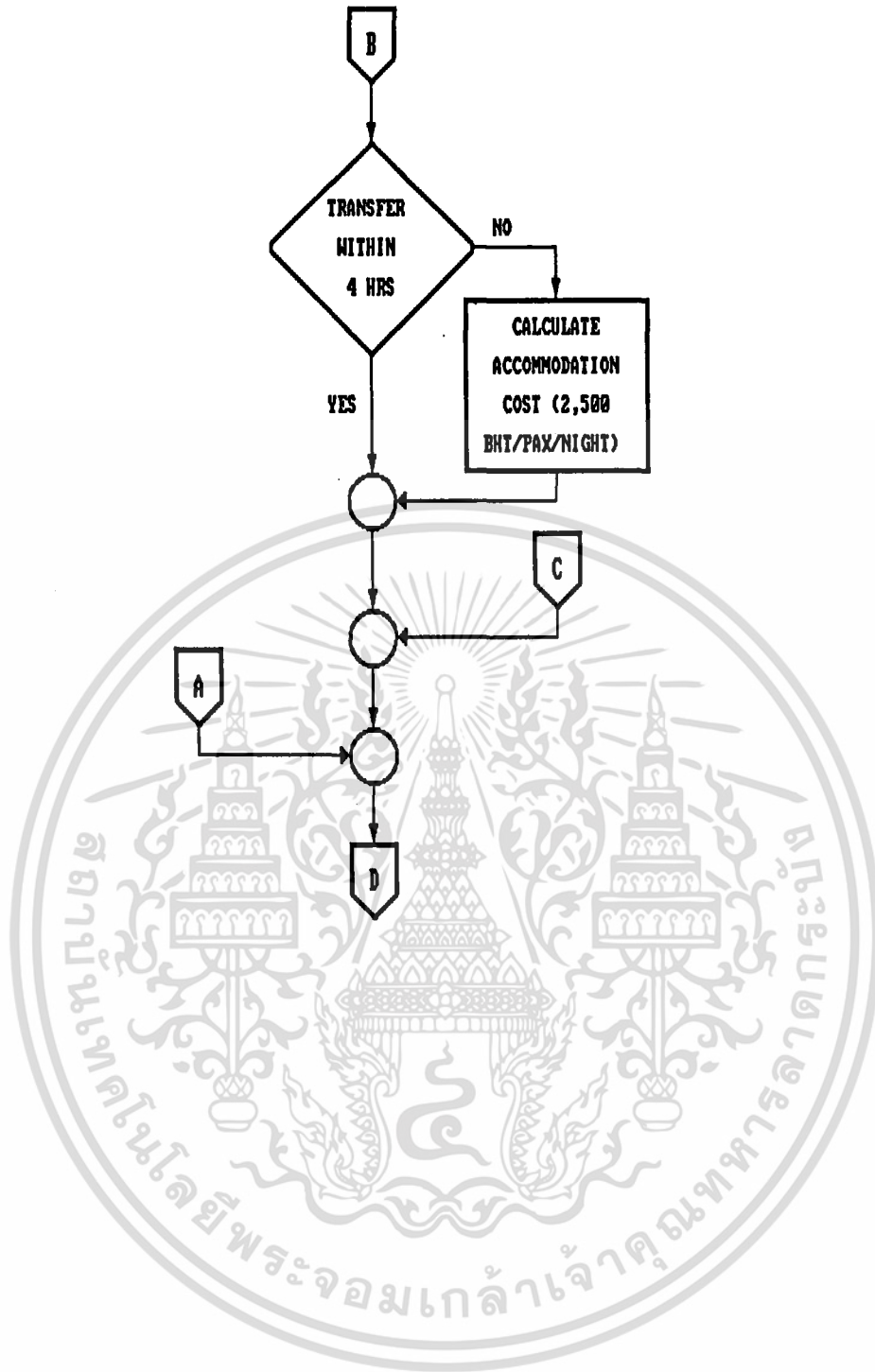
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



C6 - COST OF CHANGE AIRPORT

DIFFERENT TYPE-DELAY

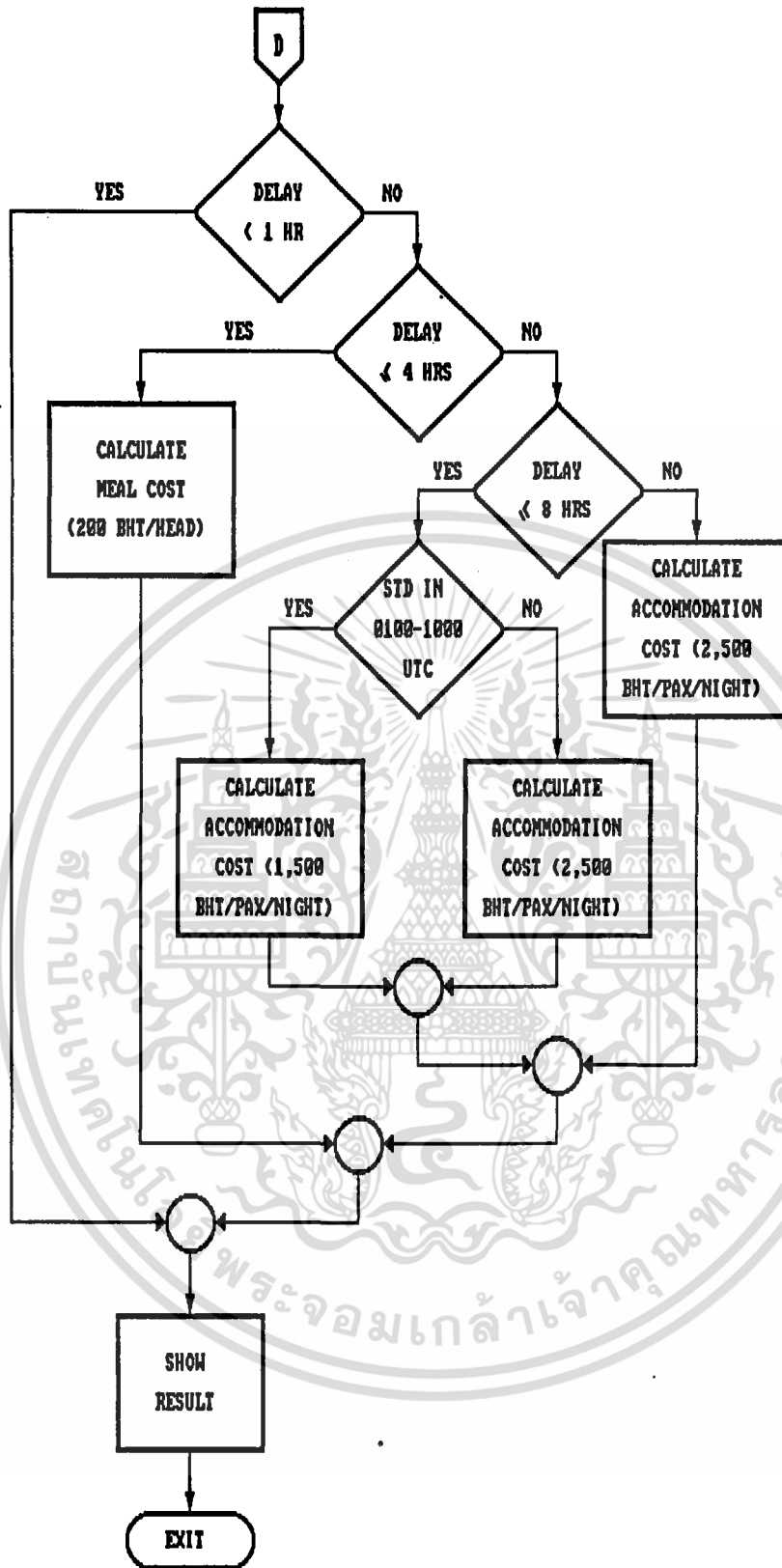
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



C6 - COST OF CHANGE AIRCRAFT

DIFFERENT TYPE-DELAY (CONTINUED-1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



C6 - COST OF CHANGE AIRPORT

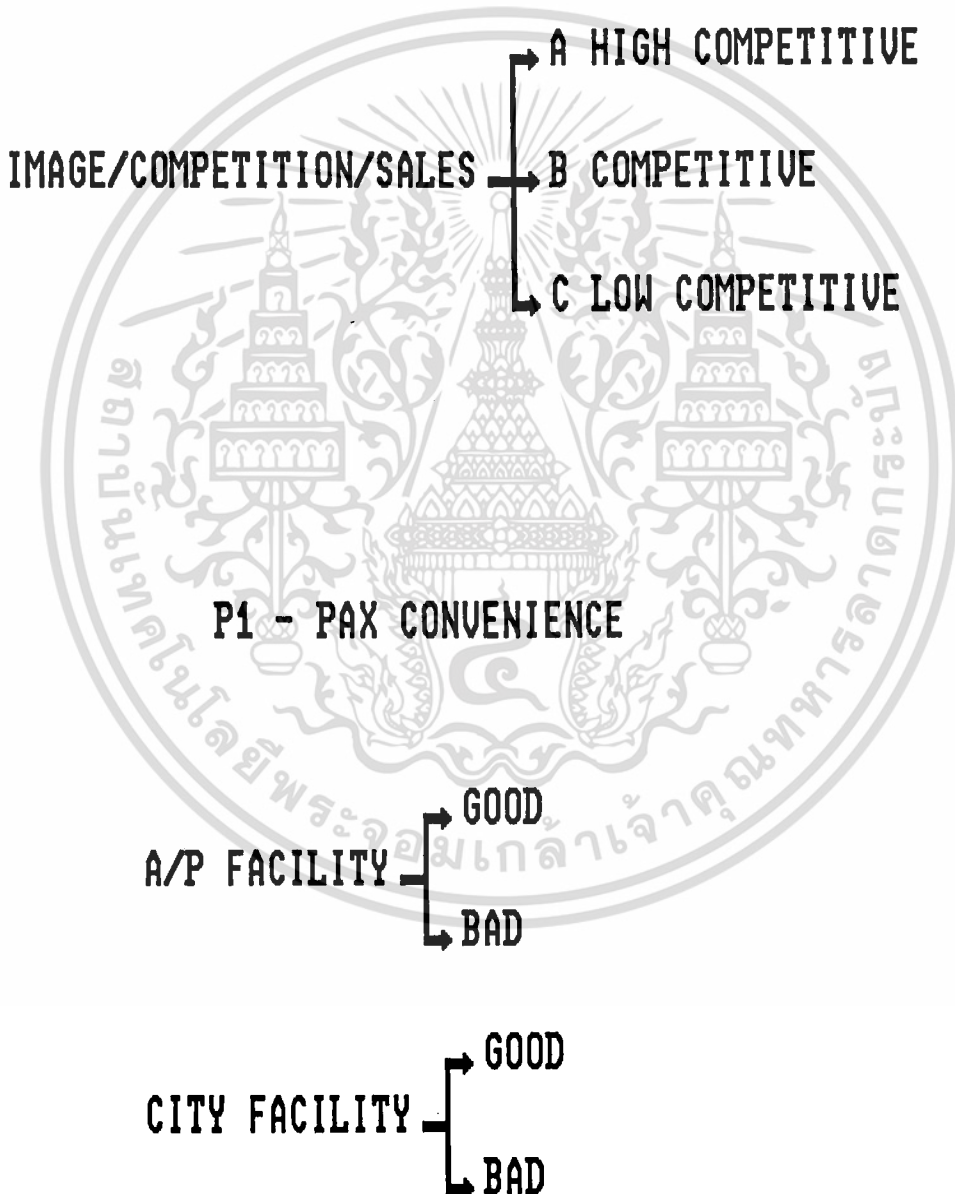
DIFFERENT TYPE-DELAY (CONTINUED-2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

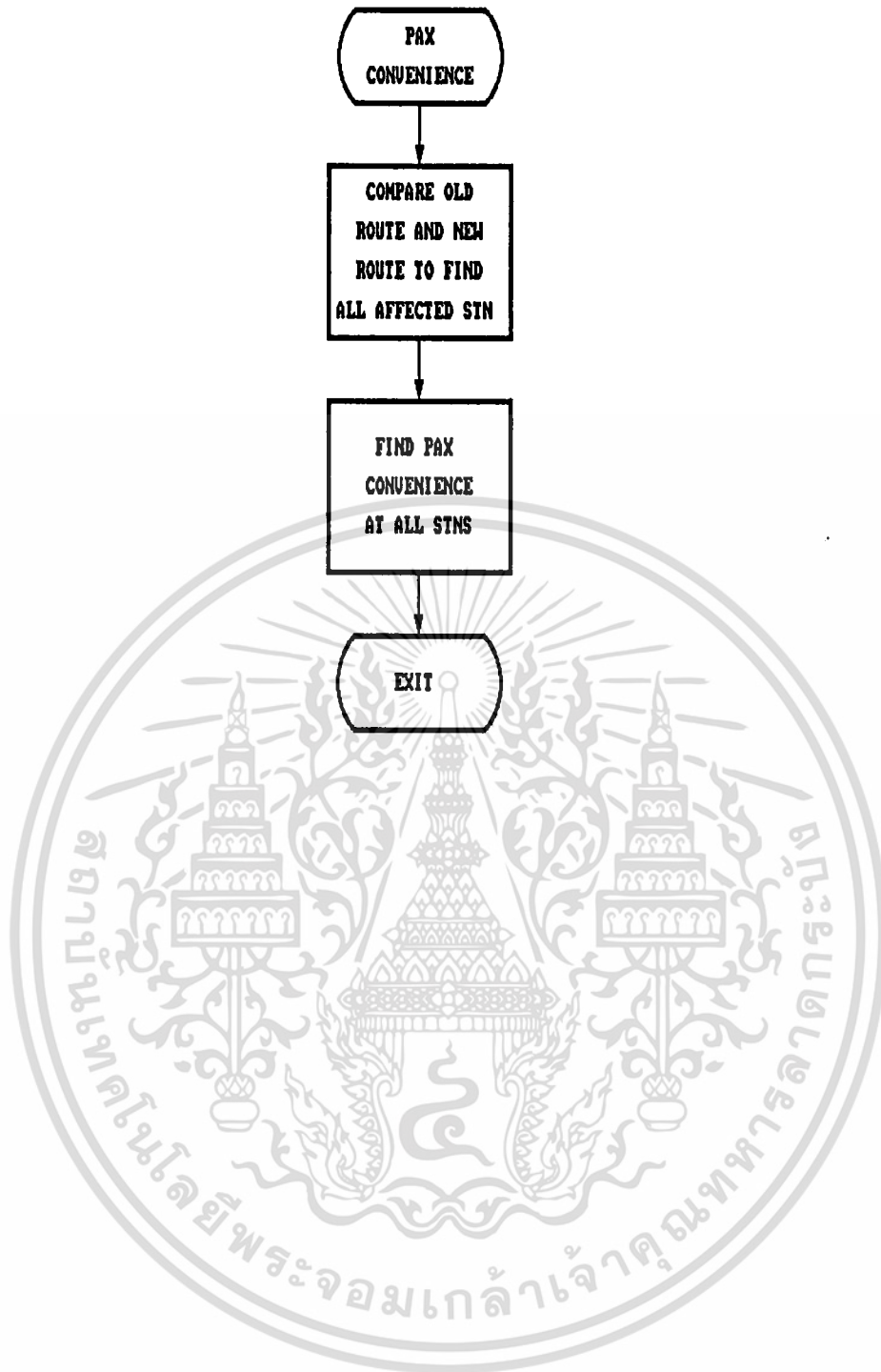
V1 - VIP

VIP GRADE — A = HAS VIP

I1 - (IMAGE/COMPETITION/SALES)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



P2 - PAX CONVENIENCE

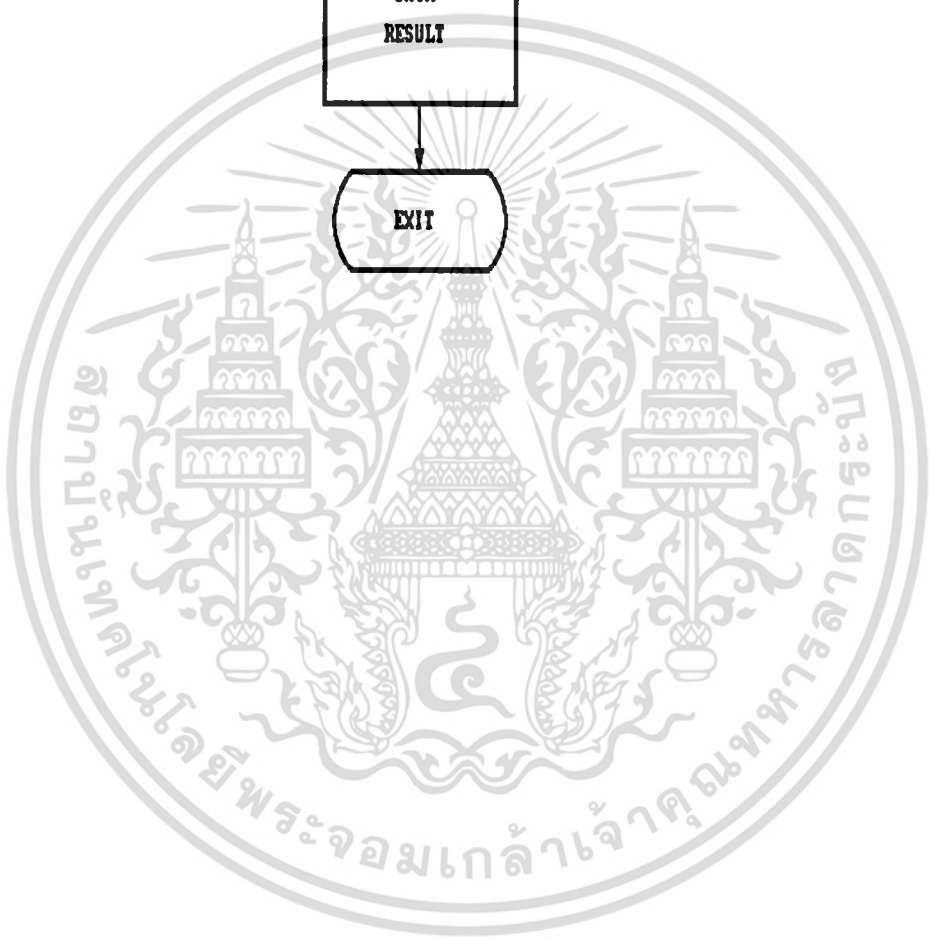
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NEXT FLIGHT
OR NEXT SECTOR

CHECK EFFECTS
TO NEXT ACTIVITY
OF THAT ACREG

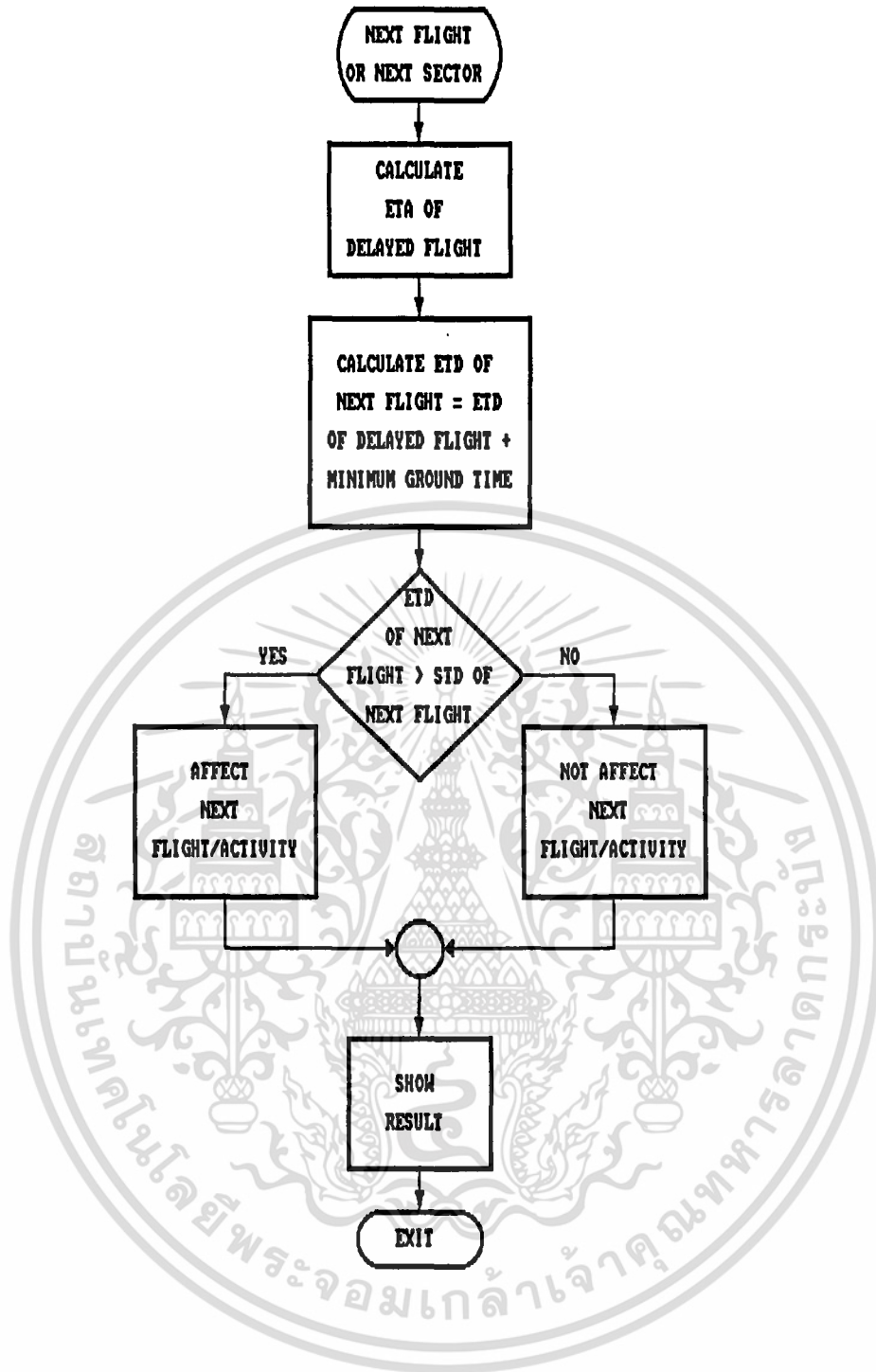
SHOW
RESULT

EXIT



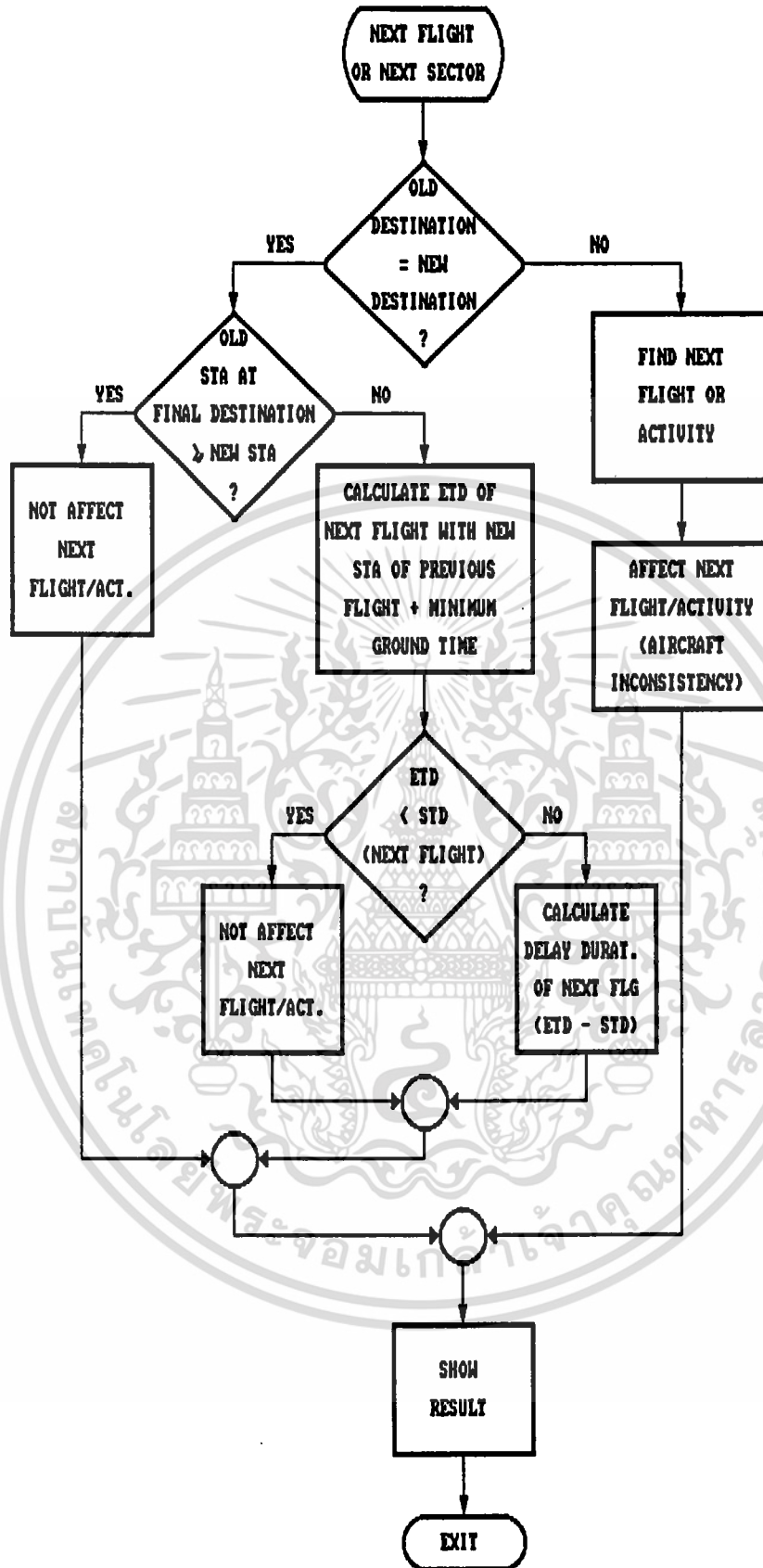
F1 - NEXT ACTIVITY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



F2 - NEXT ACTIVITY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



F3 - NEXT ACTIVITY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวกที่ 4 รายละเอียดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;; define retrieve data;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;
(define same)(define tr1)(define bkng1)(define sq)
(define dela) (define tdem1) (define delm1) (define tep)
(define tdeh1) (define delh1) (define del) (define dip)
(define fli1)(define flt1) (define secl)(define fltno1)
(define secno1)(define acreg1)(define vip1) (define acs)
(define tdep1)(define a1) (define q1) (define tdel)
(define flhr1)(define dst1)(define s)(define dstt)
(define a) (define q) (define samea) (define samed)
(define tfl)(define on) (define tw) (define app) (define ap)
(define ct 115)(define tde2) (define tdeh2) (define gos) (define cw)
(define cwi) (define md) (define px) (define cf)
(define seq) (define sec2) (define appl) (define fac1)
(define dst2) (define arr2) (define dep2) (define fac2)
(define bkng2) (define mseat1) (define tiar1)
(define tidep1) (define fltd1) (define ic)
(define fuel) (define fuelf) (define fuelt) (define slot)
(define slotf) (define slott) (define fac) (define facf) (define fact)
(define cur) (define curf) (define curt) (define rw) (define rwf)
(define rwt) (define rwlf) (define rwlt) (define wx) (define pol)
(define polf) (define polt) (define ind) (define indf) (define indt)
(define cwst1) (define cwstfl) (define cwstt1) (define cwq1)
(define maxfl) (define gst1) (define acty1)
(define ac) (define acpf) (define acpt) (define cwa)
(define cwst) (define cwstf) (define cwstt) (define cwq)
(define maxf) (define gst) (define mseat)(define acty)
(define flt) (define fltd) (define fltno) (define secno)
(define sec) (define acreg) (define tidep) (define arr)
(define dep) (define bkn) (define tiar) (define bkng)
(define vip) (define dst) (define flhr)
(define pt (lambda (x y)
  (let* ((xm (- x (* (truncate (/ x 100)) 100)))
        (ym (- y (* (truncate (/ y 100)) 100))))
    (cond
     ((< (+ xm ym) 60) (+ x y))
     (t (+ x y 40))))))

(define wind
  (make-window "" #\true))
  (clear-graphics)
  (window-set-attribute! wind 'border-attributes 21)
  (window-set-position! wind 6 16)
  (window-set-size! wind 6 42)
  (window-clear wind)
  (display " one moment please Loading .... " wind)
  (display (integer->char 174) wind)
  (define map (lambda (a b) (apply append (mapcar a b))))
  (define get (lambda (a b)
    (map (lambda (x)
      (if (equal? a (car x)) (list x)) b)))
  (define get2 (lambda (a b)
    (map (lambda (x)
      (if (eq? a (caddr x)) (append x))) b)))
  (define get3 (lambda (a b)

```

```

      (map (lambda (x)
            (if (eq? a (car x)) (append x))) b)))
(define get4 (lambda (a b)
              (map (lambda (x)
                    (if (eq? a (caddr x)) (list x))) b)))
(define get5 (lambda (a b)
              (map (lambda (x)
                    (if (eq? a (list-ref x 4)) (list x))) b)))
(define get51 (lambda (a b)
               (map (lambda (x)
                     (if (eq? a (list-ref x 4)) (append x))) b)))
(define get6 (lambda (a b)
              (map (lambda (x)
                    (if (eq? a (list-ref x 6)) (list x))) b)))
(define dget (lambda (x)
              (min (car (mapcar caddr (mapcar caddr x)))
                  (cadr (mapcar caddr (mapcar caddr x))))))

(define flight-db) (define airport) (define aircraft)
;; FILE RETRIVE DATABASE (AIRPORT) ;;

(set! AIRPORT '())
(let ((db-file "airport") (strg-stream 'AIRPORT))
  (set! db-file (open-input-file "airport")) ; open the file stream
  (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
      ;;test for EOF
      ((eof-object? sentence)
       (set! AIRPORT (reverse AIRPORT)))
      (set! strg-stream (open-input-string sentence))
      (DO ((R (READ strg-stream) (READ strg-stream))
          ((eof-object? R) (close-input-port strg-stream)
           (set! AIRPORT (cons (REVERSE LINE) AIRPORT)))
          (set! line (cons R LINE))))))

;; FILE RETRIVE DATABASE (AIRCRAFT) ;;

(set! AIRCRAFT '())
(let ((db-file "aircraft") (strg-stream 'AIRCRAFT))
  (set! db-file (open-input-file "aircraft")) ; open the file stream
  (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
      ;;test for EOF
      ((eof-object? sentence)
       (set! AIRCRAFT (reverse AIRCRAFT)))
      (set! strg-stream (open-input-string sentence))
      (DO ((R (READ strg-stream) (READ strg-stream))
          ((eof-object? R) (close-input-port strg-stream)
           (set! AIRCRAFT (cons (REVERSE LINE) AIRCRAFT)))
          (set! line (cons R LINE))))))

;; FILE RETRIVE DATABASE (flight-no) ;;
(define tg3
  (lambda ()
    (set! FLIGHT-DB '())
    (let ((db-file "f5.d") (strg-stream 'flight-db))

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(set! db-file (open-input-file "f5.d")) ; open the file stream
(do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
;;test for EOF
  (eof-object? sentence)
  (set! FLIGHT-DB (reverse FLIGHT-DB))
  (set! strg-stream (open-input-string sentence))
  (DO ((R (READ strg-stream) (READ strg-stream)))
      ((eof-object? R) (close-input-port strg-stream)
       (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
      (set! line (cons R LINE))))))

(define tg4
  (lambda ()
    (set! FLIGHT-DB '())
    (let ((db-file "f4.d") (strg-stream 'flight-db))
      (set! db-file (open-input-file "f4.d")) ; open the file stream
      (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
;;test for EOF
        ((eof-object? sentence)
         (set! FLIGHT-DB (reverse FLIGHT-DB)))
        (set! strg-stream (open-input-string sentence))
        (DO ((R (READ strg-stream) (READ strg-stream)))
            ((eof-object? R) (close-input-port strg-stream)
             (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
            (set! line (cons R LINE))))))

(define tg5
  (lambda ()
    (set! FLIGHT-DB '())
    (let ((db-file "f3.d") (strg-stream 'flight-db))
      (set! db-file (open-input-file "f3.d")) ; open the file stream
      (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
;;test for EOF
        ((eof-object? sentence)
         (set! FLIGHT-DB (reverse FLIGHT-DB)))
        (set! strg-stream (open-input-string sentence))
        (DO ((R (READ strg-stream) (READ strg-stream)))
            ((eof-object? R) (close-input-port strg-stream)
             (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
            (set! line (cons R LINE))))))

(define tg6
  (lambda ()
    (set! FLIGHT-DB '())
    (let ((db-file "f2.d") (strg-stream 'flight-db))
      (set! db-file (open-input-file "f2.d")) ; open the file stream
      (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
;;test for EOF
        ((eof-object? sentence)
         (set! FLIGHT-DB (reverse FLIGHT-DB)))
        (set! strg-stream (open-input-string sentence))
        (DO ((R (READ strg-stream) (READ strg-stream)))
            ((eof-object? R) (close-input-port strg-stream)
             (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
            (set! line (cons R LINE))))))

(define tg7
  (lambda ()

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(set! FLIGHT-DB '())
(let ((db-file "fl.d") (strg-stream 'flight-db))
  (set! db-file (open-input-file "fl.d")) ; open the file stream
  (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
      ;;test for EOF
      ((eof-object? sentence)
       (set! FLIGHT-DB (reverse FLIGHT-DB)))
      (set! strg-stream (open-input-string sentence))
      (DO ((R (READ strg-stream) (READ strg-stream))
          ((eof-object? R) (close-input-port strg-stream)
           (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
          (set! line (cons R LINE))))))
(define tg9
  (lambda ()
    (set! FLIGHT-DB '())
    (let ((db-file "f.d") (strg-stream 'flight-db))
      (set! db-file (open-input-file "f.d")) ; open the file stream
      (do ((sentence (read-line db-file) (read-line db-file)) (line nil nil))
          ;;test for EOF
          ((eof-object? sentence)
           (set! FLIGHT-DB (reverse FLIGHT-DB)))
          (set! strg-stream (open-input-string sentence))
          (DO ((R (READ strg-stream) (READ strg-stream))
              ((eof-object? R) (close-input-port strg-stream)
               (set! FLIGHT-DB (cons (REVERSE LINE) FLIGHT-DB)))
              (set! line (cons R LINE))))))
    (fast-load "aircr.fsl")
    (aircr1)(mal)
    (fast-load "action.fsl") (fast-load "const.fsl")(fast-load "cost.fsl")
    (fast-load "pr.fsl")
    (window-set-position! wind 1 1)
    (window-set-size! wind 23 78)
    (begin
      (window-clear 'console)
      (window-clear wind)
      (window-set-cursor! wind 5 80)
      (ai 15)
      (display "          FLIGHT OPERATIONS CONTROL USING AN EXPERT SYSTEM

```

THIS PROGRAM

BY

SILA THANGWARITORN

```

"wind)
(window-set-cursor! wind 20 15)
(ai 13)
(display " KMITL & THAI AIRWAY INTERNATIONAL LIMITED " wind)
(read-char)
(m)
(set! flhr (qw tidep tiar)) ;;;; CALCULATE FLIGHT HOUR
(aircr1) ;;;;;;;;; data of aircraft (old)

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(m1) ;;;;;;;;;; data from AIRPORT (DEPARTURE)
(submanu))



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;      GET DATA FROM AIRCRAFT      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;      OLD AIRCRAFT                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define aircr1
  (lambda ()
    (let* ((acr (get3 acreg aircraft))
           (acty (cadr acr)))           ;;aircraft type
      (set! ac (list-ref acr 2))
      (set! acpf (list-ref acr 3))
      (set! acpt (list-ref acr 4))
      (set! cwa (list-ref acr 5))
      (set! cwst (list-ref acr 6))
      (set! cwstf (list-ref acr 7))
      (set! cwstt (list-ref acr 8))
      (set! cwq (list-ref acr 9))     ;; crew qualified
      (set! maxf (list-ref acr 10))  ;; maximum flight distance
      (set! gst (list-ref acr 11))   ;; minimum ground time
      (set! mseat (list-ref acr 12)))) ;; maximum seat
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;      GET DATA FROM AIRCRAFT      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;      NEW AIRCRAFT                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (define aircr2
      (lambda ()
        (let* ((acr1 (get3 acreg1 aircraft))
               (acty1 (cadr acr1)))
          (set! accl (list-ref acr1 2))
          (set! acpf1 (list-ref acr1 3))
          (set! acpt1 (list-ref acr1 4))
          (set! cwa1 (list-ref acr1 5))
          (set! cwst1 (list-ref acr1 6))
          (set! cwstf1 (list-ref acr1 7))
          (set! cwstt1 (list-ref acr1 8))
          (set! cwq1 (list-ref acr1 9))
          (set! maxf1 (list-ref acr1 10))
          (set! gst1 (list-ref acr1 11))
          (set! mseat1 (list-ref acr1 12))))
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;;;;      THIS IS MAIN MANU      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        (define m
          (lambda ()
            (begin
              (window-clear 'console)
              (window-clear wind)
              (ai 15)
              (display " WHICH FLIGHT DO YOU WANT TO TEST ? TG "
                wind))
              (let ((fl #\?))
                (set! fl (read wind)))
              (cond ((and (>= fl 900)
                          (<= fl 1000))
                     (tg9)))
              (cond ((and (>= fl 700) (<= fl 890))
                     (tg8))))))

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    (tg7)))
(cond ((and (>= fl 600) (<= fl 699))
      (tg6)))
(cond ((and (>= fl 500) (<= fl 599))
      (tg5)))
(cond ((and (>= fl 400) (<= fl 499))
      (tg4)))
(cond ((and (>= fl 300) (<= fl 399))
      (tg3)))
(set! fl (get fl flight-db))
(begin
  (window-clear 'console)
  (window-clear wind)
  (display " WHAT FLIGHT DATE YOU WANT TO TEST (yymmdd) ? "
    wind))
  (let ((fltd #\?))
    (set! fltd (read wind))
    (set! flt (get1 fltd fl))))
(begin
  (window-clear 'console)
  (window-clear wind)
  (display " WHAT SECTOR YOU WANT TO TEST (1,2 or 3 [all flight].) ? "
    wind))
  (let ((secn #\?))
    (set! secn (read wind))
    (set! secno secn)
    (cond ((and (eq? secno 3) (>= (length flt) 2))
      (let* ((one (get2 1 flt))
              (two (get2 2 flt)))
        (set! sec (append one two))
        (set! fltno (car sec))
        (set! fltd (cadr sec))
        (set! acreg (list-ref sec 3))
        (set! dep (list-ref sec 4))
        (set! tidep (list-ref sec 5))
        (set! gos (list-ref sec 7))
        (set! gos1 (list-ref sec 17))
        (set! arr (list-ref sec 18))
        (set! tiar (list-ref sec 19))
        (set! ic (list-ref sec 9))
        (set! vip (list-ref sec 11))
        (set! bkng (+ (list-ref sec 8) (list-ref sec 20)))
        (set! dst (+ (list-ref sec 10) (list-ref sec 22))))
      (else (set! sec (get2 secno flt))
        (set! bkn nil)
        (set! fltno (car sec))
        (set! fltd (cadr sec))
        (set! secno (list-ref sec 2))
        (set! acreg (list-ref sec 3))
        (set! dep (list-ref sec 4))
        (set! tidep (list-ref sec 5))
        (set! arr (list-ref sec 6))
        (set! tiar (list-ref sec 7))
        (set! bkng (list-ref sec 8))
        (set! ic (list-ref sec 9)))

```

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(set! dst (list-ref sec 10))
(set! vip (list-ref sec 11))))
(if (null? sec)
  (and (window-set-cursor! wind 6 9)
    (display " THIS FLIGHT-NO IS NOT MY DATA " wind)
    (window-set-cursor! wind 10 9)
    (display " PLEASE... TYPE AGAIN (PRESS ANY KEY TO CONTINUE) THANK YO
U " wind)
    (window-set-cursor! wind 12 25)
    (ai 13)
    (display (integer->char 3) wind)
    (display (integer->char 3) wind)
    (read-char)
    (ai 15)
    (m))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; GET DATA FROM AIRPORT FILE ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define mal
  (lambda ()
    (set! app (get3 dep airport))
    (set! ap (car app))
    (set! fuel (cadr app))
    (set! fuelf (list-ref app 2))
    (set! fuelt (list-ref app 3))
    (set! slot (list-ref app 4))
    (set! slotf (list-ref app 5))
    (set! slott (list-ref app 6))
    (set! fac (list-ref app 7))
    (set! facf (list-ref app 8))
    (set! fact (list-ref app 9))
    (set! cur (list-ref app 10))
    (set! curf (list-ref app 11))
    (set! curt (list-ref app 12))
    (set! rw (list-ref app 13))
    (set! rwf (list-ref app 14))
    (set! rwt (list-ref app 15))
    (set! wx (list-ref app 16))
    (set! pol (list-ref app 17))
    (set! polf (list-ref app 18))
    (set! polt (list-ref app 19))
    (set! ind (list-ref app 20))
    (set! indf (list-ref app 21))
    (set! indt (list-ref app 22))
    (set! per (list-ref app 23))
    (aircrl))) ;;; LOAD data of aircraft ;;;;

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; GET DATA FROM AIRPORT FILE ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define ma2
  (lambda ()
    (set! appl (get3 arr airport))
    (set! ap (car appl))

```

ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(set! fuel (cadr appl))
(set! fuelf (list-ref appl 2))
(set! fuelt (list-ref appl 3))
(set! slot (list-ref appl 4))
(set! slotf (list-ref appl 5))
(set! slott (list-ref appl 6))
(set! fac (list-ref appl 7))
(set! facf (list-ref appl 8))
(set! fact (list-ref appl 9))
(set! cur (list-ref appl 10))
(set! curf (list-ref appl 11))
(set! curt (list-ref appl 12))
(set! rw (list-ref appl 13))
(set! rwf (list-ref appl 14))
(set! rwt (list-ref appl 15))
(set! wx (list-ref appl 16))
(set! pol (list-ref appl 17))
(set! polf (list-ref appl 18))
(set! polt (list-ref appl 19))
(set! ind (list-ref appl 20))
(set! indf (list-ref appl 21))
(set! indt (list-ref appl 22))
(set! per (list-ref appl 23)))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;          MANU TO SELECT ACTION          ;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define submanu
  (lambda ()
    (begin
      (window-clear 'console)
      (window-clear wind)
      (ai 15)
      (display "FLIGHT OPERATIONS CONTROL

1 CANCEL FLIGHT
2 CANCEL SECTOR
3 DELAY
4 REROUTE
5 CHANGE AIRCRAFT " wind)
      (window-set-cursor! wind 16 15)
      (ai 13)
      (display " PRESS 'M' FOR TEST OTHER FLIGHT " wind)
      (window-set-cursor! wind 18 15)
      (display " PRESS 'Q' FOR QUIT " wind)
      (window-set-cursor! wind 20 15)
      (display " PRESS 'U' FOR UPDATE DATA " wind)
      (window-set-cursor! wind 10 15)
      (ai 11)
      (display " WHAT ACTION TO SELECT (1-5) ? "
        wind))
      (ai 15)
      (let* ((ch #\?))
        (set! ch (read (ai 14)wind))
        (ai 15)
        (cond ((or (cond ((eq? ch 1) (cancel)))

```

เมื่อได้รับเอกสารที่ส่งมา โปรดแจ้งการซึ่งงานที่เกี่ยวข้องหากท่านนั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(cond ((eq? ch 2) (cancelse)))  
(cond ((eq? ch 3) (dlay)))  
(cond ((eq? ch 4) (reroute)))  
(cond ((eq? ch 5) (chac)))  
(cond ((eq? ch 'q) (exit)))  
(cond ((eq? ch 'l) (scheme-reset)))  
(cond ((eq? ch 'u) (and (edwin) (submanu))))  
(cond ((eq? ch 'm) (m))))))  
(submanu)))
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(define cancel
  (lambda ()
    (window-clear 'console)
    (window-clear wind)
    (cond ((and (>= (length flt) 2) (or (eq? secno 2) (eq? secno 1)))
      (let ((one (get2 1 flt))
            (two (get2 2 flt)))
        (set! sq (append one two))
        (set! dst2 (+ (list-ref sq 10) (list-ref sq 22)))
        (set! arr2 (list-ref sq 18))
        (set! ic (list-ref sq 9))
        (set! vip (list-ref sq 11))
        (set! dep2 (list-ref sq 4))
        (set! bkng2 (+ (list-ref sq 8) (list-ref sq 20))))
      (else (set! sq sec)
            (set! bkng2 bkng)
            (set! ic (list-ref sq 9))
            (set! vip (list-ref sq 11))
            (set! dst2 dst)
            (set! arr2 arr)
            (set! dep2 dep)))
    (cond ((eq? secno 3) (set! sq sec)
          (set! bkng2 bkng)
          (set! dst2 dst)))
    (let* ((samea (get6 arr2 (delete! sec ;; [check same arrival
      (get1 fltd flight-db))) ;; direction(new flight))
          (samed (get5 dep2 samea)) ;; check same departure (new flight)
          (set! a (list-ref samed 4))
          (set! q (list-ref samed 6))
          (set! bkng1 (list-ref samed 8)) ;; sum of booking (new flight)
          (set! mseat1 (list-ref (get3 (list-ref samed 3) aircraft) 13))
          (set! tiar1 (list-ref samed 7))
          (set! tidepl1 (list-ref samed 5))
          (set! fltd1 (list-ref samed 1)))
      (cond ((null? fltd1)
        (c1)
        (read-char)
        (p1)(f1)
        (read-char)
        (window-clear 'console)
        (window-clear wind)
        (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
        (let ((ask #\?))
          (set! ask (read wind))
          (cond ((eq? ask 'y)
            (c01))
            (else (submanu))))))
      (else
        (c1)
        (read-char)
        (p1) (f1)
        (read-char)
        (window-clear 'console)
        (window-clear wind)
        (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)

```

เส้เอกสารที่นำเข้าไปใช้บนเครื่องคอมพิวเตอร์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

(read-char)
  (window-clear 'console)
  (window-clear wind)
  (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind
)
  (let ((ask #\?))
    (set! ask (read wind))
    (cond ((eq? ask 'y)
           (c11))
          (else (submanu))))))

```

```

  (else
(c100)
(read-char)
(pl) (f2)
(read-char)
  (window-clear 'console)
  (window-clear wind)
  (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
    (let ((ask #\?))
      (set! ask (read wind))
      (cond ((eq? ask 'y)
             (c11))
            (else (submanu))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;      DELAY ACTION      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define dlay
  (lambda ()
(begin
  (window-clear 'console)
  (window-clear wind)
  (display " What time you want to delay (YYMMDDhhmm)? "
wind)
    (let ((tdepp #\?))
      (set! tdepp (read wind))
      (set! tdep tdepp))
    ;;;;;;;;;;; CALCULATE DELAY HOUR ;;;;;;;;;;;
    (set! dip (truncate (/ tdep 10000)))
    (set! tep (- tdep (* 10000 (truncate (/ tdep 10000)))))
    (cond ((eq? fltd dip)
           (set! dela (qw tidep tep)))
          (else (and (set! del (* 2400 (- dip fltd)))
                     (set! dela (pt del (qw tidep tep))))))
    (set! dela (/ dela 100))

```

```

(mal)
(ac1)
(window-clear 'console)
(window-clear wind)
(window-set-cursor! wind 0 12) (ai 49)
(display " AT AIRPORT " wind)
(display " (DEPARTURE):" wind)
(display dep wind)
(window-set-cursor! wind 0 50) (ai 95)

```

ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(display " DELAY TIME = " wind)
(display tdep wind)(ai 15)
(window-set-cursor! wind 2 0) (ac1)
(window-set-cursor! wind 4 0) (cw1)
(window-set-cursor! wind 6 0) (fu1)
(window-set-cursor! wind 8 0) (sl1)
(window-set-cursor! wind 10 0) (fc1)
(window-set-cursor! wind 12 0) (cu1)
(window-set-cursor! wind 14 0) (rw1)
(window-set-cursor! wind 16 0) (wx1)
(window-set-cursor! wind 18 0) (po1)
(window-set-cursor! wind 20 0)(id1)
```

```
(begin
  (display (integer->char 2))
  (window-set-cursor! wind 25 10)
  (read-char) wind)
(delay1)))
```

```
(define delay1
  (lambda ()
    (set! arr (list-ref sec 6))
    (ma2)
    (window-clear 'console)
    (window-clear wind)
    (window-set-cursor! wind 0 12) (ai 49)
    (display " AT AIRPORT " wind)
    (display " (ARRIVAL) : " wind)
    (display arr wind)
    (window-set-cursor! wind 0 50) (ai 95)
    (display " DELAY TIME = " wind)
    (display tdep wind) (ai 15)
    (window-set-cursor! wind 2 0) (ac1)
    (window-set-cursor! wind 4 0) (cw1)
    (window-set-cursor! wind 6 0) (fu1)
    (window-set-cursor! wind 8 0) (sl1)
    (window-set-cursor! wind 10 0) (fc1)
    (window-set-cursor! wind 12 0) (cu1)
    (window-set-cursor! wind 14 0) (rw1)
    (window-set-cursor! wind 16 0) (wx1)
    (window-set-cursor! wind 18 0) (po1)
    (window-set-cursor! wind 20 0) (id1)
  (begin
    (display (integer->char 2))
    (window-set-cursor! wind 25 10)
    (read-char) wind)
  (delcost)
  (read-char)))
```

```
(define delcost
  (lambda ()
    (cond ((>= (length flt) 2)
      (set! samea (get6 arr (delete! tw (delete! on (get1 fltd flight-db))))))
      (else (set! samea (get6 arr (delete! sec ; [check same arrival
        (get1 fltd flight-db))))) ;; direction(new flight))
      (cond ((> (length (get5 dep samea))) 1)
        (else 0))))
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      (set! sam (dget (get5 dep samea)))
      (set! samed (get51 sam samea)))
      (else (set! samed (get51 dep samea))))
(set! a (list-ref samed 4))
(set! q (list-ref samed 6))
(set! bkng1 (list-ref samed 8))           ;; sum of booking (new flight)
(set! mseat1 (list-ref (get3 (list-ref samed 3) aircraft) 12))
(set! tiar1 (list-ref samed 7))
(set! tidepl (list-ref samed 5))
(set! fltd1 (list-ref samed 1))
(cond ((and (eq? a dep) (eq? q arr))
      (set! same 'y))
      (else (set! same 'n)))
(cond ((null? fltd1)
      (c2)
      (read-char)
      (p1) (f2)
      (read-char)
      (window-clear 'console)
      (window-clear wind)
      (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind
)
      (let ((ask #\?))
      (set! ask (read wind))
      (cond ((eq? ask 'y)
            (delly))
            (else (submanu))))))
      (else
      (set! flhr1 (qw tidepl tiar1))      ;;;; CALCULATE FLIGHT HOUR
      (c2)
      (READ-CHAR)
      (p1) (f2)
      (read-char)
      (window-clear 'console)
      (window-clear wind)
      (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
      (let ((ask #\?))
      (set! ask (read wind))
      (cond ((eq? ask 'y)
            (delly))
            (else (submanu)))))))))
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;:::::::::          REROUTE OF FLIGHT          ;::::::::::::::::::::::::::::::::::::::::::
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
(define reroute
  (lambda ()
    (begin
      (window-clear 'console)
      (window-clear wind)
      (ai 11)
      (display " WHAT IS THE NEW DESTINATION ? "
        wind))
      (let ((narr #\?))
      (set! narr (read (ai 14) wind))
      (set! arr narr))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(newline)
(ai 11)
(display " WHAT TIME DOSE THE FLIGHT ARRIVAL AT NEW DESTINATION (YMMDDhhmm)? " wind)
      (let ((tdepp #\?))
        (set! tdepp (read (ai 14) wind))
        (set! tdep tdepp))
```

```
(ma2)
(ai 11)
(display " WHAT HOUR TO NEW DESTINATION ? " wind)
      (let ((a #\?))
        (set! a (read (ai 14) wind))
        (set! flhrl a))
```

```
(ai 11)
(display " HAS A PERMIT FOR NEW AIRPORT ? " wind)
      (let ((p #\?))
        (set! p (read (ai 14) wind))
        (set! per p))
```

```
(newline)
(ai 15)
(window-clear 'console)
(window-clear wind)
(window-set-cursor! wind 0 12) (ai 49)
(display " AT AIRPORT " wind)
(display " (ARRIVAL) : " wind)
(display arr wind) (ai 15)
(window-set-cursor! wind 2 0) (ac2)
(window-set-cursor! wind 4 0) (cw1)
(window-set-cursor! wind 6 0) (ful)
(window-set-cursor! wind 8 0) (sll)
(window-set-cursor! wind 10 0) (fcl)
(window-set-cursor! wind 12 0) (cul)
(window-set-cursor! wind 14 0) (rw1)
(window-set-cursor! wind 16 0) (wx1)
(window-set-cursor! wind 18 0) (pol)
(window-set-cursor! wind 20 0) (id1)
(window-set-cursor! wind 22 0) (pm1)
(begin
  (display (integer->char 2))
  (window-set-cursor! wind 25 10)
  (read-char) wind)
(rrtcost)
(read-char)
(submanu)))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; FIND TG FLIGHT GOING TO SAME DIRECTION ;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
(define rrtcost
  (lambda ()
    (let* ((samea (get6 arr (delete! sec ;; [check same arrival
      (get1 fltd flight-db)))) ;; direction(new flight)]
          (samed (get5 dep samea))) ;; check same depature (new flight)
      (set! a (list-ref samed 4))
      (set! q (list-ref sec 6))
```

บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(set! bkng1 (list-ref samed 8))      ;; sum of booking (new flight)
(set! mseat1 (list-ref (get3 (list-ref samed 3) aircraft) 13))
(set! tiar1 (list-ref samed 7))
(set! tidepl (list-ref samed 5))
(set! fltd1 (list-ref samed 1))
(set! dst1 (list-ref samed 10))
(cond ((null? fltd1)
      (cond ((and (eq? q arr)
                  (set! same 'y))
              (c2)                ;; LOAD c2
              (read-char)
              (p2)(f2)
              (submanu))
            (else (set! same 'n)
                  (c3)
                  (read-char)
                  (P2)(f2)
                  (read-char)
                  (window-clear 'console)
                  (window-clear wind)
                  (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
                  )
            (let ((ask #\?))
                (set! ask (read wind))
                (cond ((eq? ask 'y)
                      (rer1))
                      (else (submanu))))))
      (else
       (set! flhr1 (QW tidepl tiar1))  ;;; CALCULATE NEW FLIGHT HOUR
       (cond ((and (eq? q arr)
                   (set! same 'y))
               (c2)                ;; LOAD c2
               (read-char)
               (p2) (f2)
               (READ-CHAR)
               (submanu))          ;; LOAD submanu
             (else (set! same 'n)
                   (c3)
                   (read-char)
                   (p2) (f2)
                   (read-char)
                   (window-clear 'console)
                   (window-clear wind)
                   (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
                   (let ((ask #\?))
                       (set! ask (read wind))
                       (cond ((eq? ask 'y)
                             (rer1))
                             (else (submanu))))))))))

(define chac
  (lambda ()
    (cond ((>= (length flt) 2)
           (set! samea (get6 arr (delete! tw (delete! on (get1 fltd flight-db))))))
          (else (set! samea (get6 arr (delete! sec) ;; [check same arrival
  
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    (get1 fltd flight-db))))))    ;; direction(new flight)]
(cond ((> (length (get5 dep samea)) 1)
      (set! sam (dget (get5 dep samea)))
      (set! samed (get51 sam samea)))
      (else (set! samed (get51 dep samea))))
(set! a (list-ref samed 4))
(set! q (list-ref samed 6))

(begin
  (window-clear 'console)
  (window-clear wind)
  (newline)
  (display " WHICH FLIGHT DO YOU WANT TO TEST ? TG "
    wind))
  (let ((fl #\?))
    (set! fl (read wind)))
  (cond ((and (>= fl 900)
             (<= fl 1000))
        (tg9)))
  (cond ((and (>= fl 700) (<= fl 890))
        (tg7)))
  (cond ((and (>= fl 600) (<= fl 699))
        (tg6)))
  (cond ((and (>= fl 500) (<= fl 599))
        (tg5)))
  (cond ((and (>= fl 400) (<= fl 499))
        (tg4)))
  (cond ((and (>= fl 300) (<= fl 399))
        (tg3)))
  (set! fl (get fl flight-db))
  (begin
    (window-clear 'console)
    (window-clear wind)
    (display " WHAT FLIGHT DATE (yymmdd) ? "
      wind))
      (let ((fltd #\?))
        (set! fltd (read wind))
        (set! flt1 (get1 fltd fl)))

  (begin
    (window-clear 'console)
    (window-clear wind)
    (display " WHAT SECTOR YOU WANT TO TEST (1 or 2) ? "
      wind))
      (let ((secn #\?))
        (set! secn (read wind))
        (set! secl (get2 secn flt1))))

  (if (null? secl)
      (and (window-set-cursor! wind 6 9)
           (display " THIS FLIGHT-NO IS NOT MY DATA " wind)
           (window-set-cursor! wind 10 9)
           (display " PLEASE... TYPE AGAIN (PRESS ANY KEY TO CONTINUE) THANK YOU "
             wind)
           (window-set-cursor! wind 12 25))
      (ai 13))

```

เอกสาร (ai 13) การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      (display (integer->char 3) wind)
      (display (integer->char 3) wind)
      (read-char)
      (ai 15)
      (chac)))

;;; get data FROM FLIGHT.D FILE ;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      (set! fltnol (car secl))
      (set! fltdl (cadr secl))
      (set! secnol (caddr secl))
      (set! acregl (list-ref secl 3))
      (set! tidepl (list-ref secl 5))
      (set! tiarl (list-ref secl 7))
      (set! bkngl (list-ref secl 8))
      (set! vip1 (list-ref secl 9))
      (set! dst1 (list-ref secl 10))
      (set! flhr1 (qw tidepl tiarl)) ;;;; CALCULATE NEW FLIGHT HOUR
      (aircr2)
      (window-clear 'console)
      (chm)
      (read-char)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; CHANG AIRCRAFT MANU ;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      (define chm
        (lambda ()
          (window-clear wind)
          (cond ((eq? acregl acregl)
                 (set! acs 'y))
                (else (set! acs 'n)))
          (ai 11)(display " CHANGE A/C (SELECT 1-4)

                1) SAME TYPE ONTIME
                2) SAME TYPE DELAY
                3) DIFF TYPE ONTIME
                4) DIFF TYPE DELAY " wind)
          (window-set-cursor! wind 13 15)
          (ai 14)
          (display " PRESS Q FOR EXIT " wind)
          (window-set-cursor! wind 15 15)
          (display " PRESS S TO SUBMANU (SELECT OTHER ACTION)" wind)
          (window-set-cursor! wind 10 7)
          (display " WHAT ACTION TO SELECT? " wind)
          (ai 15)
          (let* ((ch #\?))
            (set! ch (read (ai 14) wind))
            (cond ((or (cond ((eq? ch 1) (sameon)))
                      (cond ((eq? ch 2) (samede)))
                      (cond ((eq? ch 3) (diffon)))
                      (cond ((eq? ch 4) (diffde)))
                      (cond ((eq? ch 'q) (exit)))
                      (cond ((eq? ch 's) (submanu))))))
              (chm))))

```

```
(DEFINE SAMEON
  (lambda ()
    (window-clear wind)
    (cond ((eq? acs 'y) (ai 95)
      (display " CHANGE A/C SAME TYPE ONTIME " wind)
      (ai 15) (ac3) (read-char)(c4)(read-char)
      (window-clear 'console)
      (window-clear wind)
      (display " DO YOU WANT DISPLAY OUTPUT TO PRINTER (Y or N) ? " wind)
    )

```

```
      (let ((ask #\?))
        (set! ask (read wind))
        (cond ((eq? ask 'y)
          (sson)
          (else (chm))))))
    (else (and
      (ai 95)
      (window-set-cursor! wind 8 18)
      (display " CANNOT CHANGE AIRCRAFT SAME TYPE " wind)
      (window-set-cursor! wind 9 17)
      (display " PRESS ANYKEY TO CONTINUE" wind)
      (read-char) (ai 15)
      (chm))))))

```

```
(DEFINE SAMEDE
  (lambda ()
    (window-clear wind)
    (display " WHAT TIME TO DEPARTURE (hhmm) ? " wind)
    (set! tdep (read wind))
    ;;;;;;;;;; CALCULATE DELAY HOUR ;;;;;;;;;;
    (set! dela (/ (qw tidep tdep) 100))
    (window-clear wind)
    (window-set-cursor! wind 0 12)
    (cond ((eq? acreg1)
      (set! acs 'y)
      (ai 95)
      (display " CHANGE A/C SAME TYPE DELAY " wind)

```

```
(ai 15)
(ac3)
(window-set-cursor! wind 6 0)(cw1)
(window-set-cursor! wind 8 0)(ful)
(window-set-cursor! wind 10 0)(sl1)
(window-set-cursor! wind 12 0)(fc1)
(window-set-cursor! wind 14 0)(cul)
(window-set-cursor! wind 16 0)(rw1)
(window-set-cursor! wind 18 0)(wx1)
(window-set-cursor! wind 20 0)(po1)
(window-set-cursor! wind 22 0)(id1)
(read-char)(c2)
(read-char)
(pl)
(read-char)(chm))

```

```
(else (and (set! acs 'n)

```

เอกสารนี้เป็นเอกสาร (ai 95) วัสดุสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(window-set-cursor! wind 8 18)
(display " CANNOT CHANGE AIRCRAFT SAME TYPE " wind)
(window-set-cursor! wind 9 17)
(display " PRESS ANYKEY TO CONTINUE" wind)
(read-char)
(ai 15)
(chm))))))
```

```
(DEFINE DIFFON
  (lambda ()
    (window-clear wind)
    (window-set-cursor! wind 0 12)
    (cond ((eq? acregl acreg)
      (and (window-set-cursor! wind 8 18)
        (ai 95)
        (display " CANNOT CHANGE AIRCRAFT DIFF TYPE ONTIME " wind)
        (window-set-cursor! wind 9 17)
        (display " PRESS ANYKEY TO CONTINUE " wind)
        (read-char)
        (ai 15)
        (chm)))
      (else (ai 95)(display " CHANGE A/C DIFF TYPE ONTIME " wind)
        (ai 15)
        (ac3)
        (window-set-cursor! wind 8 0)
        (cw2)
        (window-set-cursor! wind 10 0)
        (pm2)
        (window-set-cursor! wind 12 0)
        (sl1)
        (window-set-cursor! wind 14 0)
        (fc1)
        (window-set-cursor! wind 16 0)
        (rw1)
        (read-char)
        (c5)
        (read-char)
        (p1)
        (read-char)
        (window-clear wind)
        (ai 15)
        (chm))))))
```

```
(DEFINE DIFFDE
  (lambda ()
    (window-clear wind)
    (window-set-cursor! wind 0 12)
    (cond ((eq? acregl acreg)
      (and (window-set-cursor! wind 8 18)
        (ai 95)
        (display " CANNOT CHANGE AIRCRAFT DIFF TYPE ONTIME " wind)
        (window-set-cursor! wind 9 17)
        (display " PRESS ANYKEY TO CONTINUE " wind)
        (read-char)
        (ai 15)
        (chm))))))
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(chm)))

(else (window-clear wind)
      (window-set-cursor! wind 2 4)
      (display " WHAT TIME TO DEPARTURE (hhmm) ? " wind)
      (set! tdep (read wind))
      ;;;;;;;;;; CALCULATE DELAY HOUR ;;;;;;;;;;
      (set! dela (/ (qw tidep tdep) 100))
      (window-clear wind)
      (ai 95)
      (display " CHANGE A/C DIFF TYPE DELAY TG " wind)
      (AI 14)
      (display fltno wind)
      (ai 95)
      (display " TO TG " wind)
      (ai 14)
      (display fltno1 wind)
      (ai 15)
      (ac3)
      (window-set-cursor! wind 8 0)
      (cw2)
      (window-set-cursor! wind 10 0)
      (ful)
      (window-set-cursor! wind 12 0)
      (pm2)
      (window-set-cursor! wind 14 0)
      (sl1)
      (window-set-cursor! wind 16 0)
      (fc1)
      (read-char)
      (window-clear wind)
      (ai 95)
      (display " CHANGE A/C DIFF TYPE DELAY TG " wind)
      (AI 14)
      (display fltno wind)
      (ai 95)
      (display " TO TG " wind)
      (ai 14)
      (display fltno1 wind)
      (ai 15)
      (window-set-cursor! wind 2 0)
      (cul)
      (window-set-cursor! wind 4 0)
      (rw1)
      (window-set-cursor! wind 6 0)
      (wx1)
      (window-set-cursor! wind 8 0)
      (po1)
      (window-set-cursor! wind 10 0)
      (idl)
      (read-char)
      (ai 15)
      (c6)
      (read-char)
      (pl)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; condition of DELAY (AIRCRAFT) ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define acl
  (lambda ()
    (cond ((eq? ac 'y) (display " AIRCRAFT IS AVAILABLE" wind)))
    (if (and (eq? ac 'n)
              (>= tdep acpf)
              (<= tdep acpt))
        (begin (ai 13)
                 (display " AIRCRAFT IS NOT AVAILABLE" wind)
                 (ai 12)
                 (display "(" wind)
                 (display acpf wind)
                 (display " TO " wind)
                 (display acpt wind)
                 (display ")" wind)))
        (ai 15)
        (if (and (eq? ac 'n)
                  (or (< tdep acpf) (> tdep acpt)))
            (display " AIRCRAFT IS AVAILABLE" wind))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; condition of REROUTE (AC2 AIRCRAFT PER) ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define ac2
  (lambda ()
    (cond ((> dst maxf)
           (ai 12)
           (display " AIRCRAFT PERFORMANCE IS NOT OK. " wind)
           (ai 15)))
          ((< dst maxf)
           (display " AIRCRAFT PERFORMANCE IS OK. " wind))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; condition aircraft (AC3 TECHNICAL) ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define ac3
  (lambda ()
    (set! tdep tiar)
    (set! tdepl tiarl)
    (window-set-cursor! wind 2 0)
    (cond ((eq? ac 'y) (display " AIRCRAFT HAS TECHNICAL PROBLEM " wind)))
    (if (and (eq? ac 'n)
              (>= tiar acpf)
              (<= tiarl acpt))
        (begin (ai 13)(display " AIRCRAFT HAS TECHNICAL PROBLEM " wind)
                 (ai 12)
                 (display "(" wind)
                 (display acpf wind)
                 (display " TO " wind)
                 (display acpt wind)
                 (display ")" wind)(ai 15)))
        (if (and (eq? ac 'n)
                  (or (< tiar acpf) (> tiarl acpt)))
            (begin (display " AIRCRAFT DOSE NOT HAVE TECHNICAL PROBLEM " wind)
                    (display " AIRCRAFT DOSE NOT HAVE TECHNICAL PROBLEM " wind))))

```

เอกสารนี้เป็นเอกสารของกองทัพอากาศไทย ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      (ai 12)
      (display "(" wind)
      (display acpf wind)
      (display " TO " wind)
      (display acpt wind)
      (display ")" wind)(ai 15)))
(set! al (pt tiar1 gst1))      ;; al = tiar1 + gst1
(set! ql (pt tiar gst))      ;; ql = tiar + gst
(window-set-cursor! wind 4 0)
(cond ((> al tiar)           ;; gst = ground...
      (ai 13)
      (display " AIRCRAFT IS NOT AVAILABLE " wind)(ai 15)))
(if (and (< al tiar)        ;; tiar1 = time arrival (new a/c)
        (<= ql tiar1))
    (display " AIRCRAFT IS AVAILABLE " wind))
(if (and (< al tiar)
        (>= ql tiar1))
    (begin (ai 13)
           (display " AIRCRAFT IS NOT AVAILABLE " wind)(ai 15)))
(window-set-cursor! wind 6 0)
(if (and (eq? acs 'n)      ;; acs = aircraft same type
        (> dst1 maxfl))   ;; flhr1 = flight-hour (new destination)
    (begin (ai 13)
           (display " AIRCRAFT PERFORMANCE IS NOT OK. " wind)
           (ai 15)))
(if (and (eq? acs 'n)
        (< dst1 maxfl))   ;; maxfl = maximum flight-hour (new flight)
    (display " AIRCRAFT PERFORMANCE IS OK. " wind))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; CONDITION OF CREW ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define cw) (define cw22) (define cwii) (define cw0)
(define md1) (define md2)
(define cw11) (define ald) (define cwi3) (define cwi33)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; CALCULATE CREW INDEX ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define index

```

```

  (lambda ()
    (cond ((= (length flt) 2)
          (set! tw (get2 2 flt))
          (set! sq (append on tw))
          (set! tdel (list-ref sq 5))
          (set! tde2 (list-ref sq 7))
          (set! tdeh2 (list-ref sq 17))
          (set! tdeh1 (list-ref sq 19)))
          (else (set! sq sec)
                (set! tdel (list-ref sq 5))
                (set! tdeh1 (list-ref sq 7))))
    (let* ((dm (- tdel (* 100 (truncate (/ tdel 100)))))
           (am (- tdeh1 (* 100 (truncate (/ tdeh1 100)))))
           (dh (/ (- tdel dm) 100))
           (ah (/ (- tdeh1 am) 100)))

```

(cond ((> tdel tdeh1) สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
(if (and (eq? cwa 'y) (> cwi 100)
        (eq? cwst 'y) (eq? cwq 'n))
    (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
           (ai 12)
           (display "(" wind) (display cwstf wind)
           (display "TO" wind) (display cwstt wind)
           (display ")" wind) (ai 11)
           (display "("wind) (display "CREW INDEX= " wind)
           (display cwi wind) (display ";MAXIMUM HOUR = " wind)
           (display md2 wind) (display ":" wind)
           (display md wind) (ai 14)
           (display ald wind) (display ")" wind))))
```

```
(ai 15)
(if (and (eq? cwa 'y) (> cwi 100)
        (eq? cwst 'n) (eq? cwq 'n)
        (or (< tdep cwstf) (> tdep cwstt))))
    (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
           (ai 12)
           (display "(" wind) (display cwstf wind)
           (display "TO" wind) (display cwstt wind)
           (display ")" wind) (ai 11)
           (display "("wind) (display "CREW INDEX= " wind)
           (display cwi wind) (display ";MAXIMUM HOUR = " wind)
           (display md2 wind) (display ":" wind)
           (display md wind) (ai 14)
           (display ald wind) (display ")" wind))))
```

```
(ai 15)
(if (and (eq? cwa 'y) (> cwi 100)
        (eq? cwst 'n) (>= tdep cwstf)
        (<= tdep cwstt))
    (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
           (ai 12)
           (display "(" wind) (display cwstf wind)
           (display "TO" wind) (display cwstt wind)
           (display ")" wind) (ai 11)
           (display "("wind) (display "CREW INDEX= " wind)
           (display cwi wind) (display ";MAXIMUM HOUR = " wind)
           (display md2 wind) (display ":" wind)
           (display md wind) (ai 14)
           (display ald wind) (display ")" wind))))
```

```
(ai 15)
(if (and (eq? cwa 'y) (<= cwi 100))
    (begin (display " CREW IS AVAILABLE" wind) (ai 11)
           (display "("wind) (display "CREW INDEX= " wind)
           (display cwi wind) (display ";MAXIMUM HOUR = " wind)
           (display md2 wind) (display ":" wind)
           (display md wind) (ai 14)
           (display ald wind) (display ")" wind))))
```

```
(ai 15)
(if (and (eq? cwa 'n) (eq? cwst 'y)
        (eq? cwq 'y))
    (begin (display " CREW IS AVAILABLE" wind) (ai 11)
           (display "("wind) (display "CREW INDEX= " wind)
           (display cwi wind) (display ";MAXIMUM HOUR = " wind)
           (display md2 wind) (display ":" wind)
           (display md wind) (ai 14)
           (display ald wind) (display ")" wind))))
```

เอกสาร (display md2 wind) (display ":" wind) เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      (display md wind) (ai 14)
      (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwa 'n) (eq? cwst 'n)
         (eq? cwq 'n) (or (< tdep cwstf) (> tdep cwstt))))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
         (ai 12) (display "(" wind)
         (display cwstf wind) (display "TO" wind)
         (display cwstt wind) (display ")" wind)
         (ai 11) (display "("wind)
         (display "CREW INDEX= " wind) (display cwi wind)
         (display ";MAXIMUM HOUR = " wind) (display md2 wind)
         (display ":" wind) (display md wind) (ai 14)
         (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwa 'n) (eq? cwst 'n)
         (eq? cwq 'y) (or (< tdep cwstf) (> tdep cwstt))))
  (begin (display " CREW IS AVAILABLE" wind) (ai 11)
         (display "("wind) (display "CREW INDEX= " wind)
         (display cwi wind) (display ";MAXIMUM HOUR = " wind)
         (display md2 wind) (display ":" wind)
         (display md wind) (ai 14)
         (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwa 'n) (eq? cwst 'n)
         (>= tdep cwstf) (<= tdep cwstt)))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
         (ai 12) (display "(" wind)
         (display cwstf wind) (display "TO" wind)
         (display cwstt wind) (display ")" wind)
         (ai 11) (display "("wind) (display "CREW INDEX= " wind)
         (display cwi wind) (display ";MAXIMUM HOUR = " wind)
         (display md2 wind) (display ":" wind)
         (display md wind) (ai 14)
         (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwa 'n) (eq? cwst 'y)
         (eq? cwq 'n))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
         (ai 12) (display "(" wind)
         (display cwstf wind) (display "TO" wind)
         (display cwstt wind) (display ")" wind)
         (ai 11) (display "("wind) (display "CREW INDEX= " wind)
         (display cwi wind) (display ";MAXIMUM HOUR = " wind)
         (display md2 wind) (display ":" wind)
         (display md wind) (ai 14)
         (display ald wind) (display ")" wind)))
(ai 15)))

```

```

;;;;;;;;;;;;;
;; condition of DELAY (CREW 2) ;;
;;;;;;;;;;;;;
(define cw2

```

```

  (lambda ()

```

```

(index) รมเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(if (and (eq? cwst 'y) (eq? cwq 'y))
  (begin (display " CREW IS AVAILABLE" wind)
    (ai 11) (display "(" wind)
    (display "CREW INDEX= " wind) (display cwi wind)
    (display ";MAXIMUM HOUR = " wind) (display md2 wind)
    (display ":" wind) (display md wind)
    (ai 14) (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwst 'y) (eq? cwq 'n))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
    (ai 11) (display "(" wind)
    (display "CREW INDEX= " wind)
    (display cwi wind) (display ";MAXIMUM HOUR = " wind)
    (display md2 wind) (display ":" wind)
    (display md wind) (ai 14)
    (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwst 'n) (>= tdep cwstf))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE" wind)
    (ai 12) (display "(" wind) (display cwstf wind)
    (display "TO" wind) (display cwstt wind)
    (display ")" wind) (ai 11)
    (display "(" wind) (display "CREW INDEX= " wind)
    (display cwi wind) (display ";MAXIMUM HOUR = " wind)
    (display md2 wind) (display ":" wind)
    (display md wind) (ai 14)
    (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwst 'n)
  (or (> tdep cwstf) (< tdep cwstt))
  (eq? cwq 'n))
  (begin (ai 13) (display " CREW IS NOT AVAILABLE " wind)
    (ai 12) (display "(" wind) (display cwstf wind)
    (display "TO" wind) (display cwstt wind)
    (display ")" wind) (ai 11)
    (display "(" wind) (display "CREW INDEX= " wind)
    (display cwi wind) (display ";MAXIMUM HOUR = " wind)
    (display md2 wind) (display ":" wind)
    (display md wind) (ai 14)
    (display ald wind) (display ")" wind)))
(ai 15)
(if (and (eq? cwst 'n)
  (or (> tdep cwstf) (< tdep cwstt))
  (eq? cwq 'y))
  (begin (display " CREW IS AVAILABLE " wind)
    (display "(" wind) (display cwstf wind)
    (display "TO" wind) (display cwstt wind)
    (display ")" wind) (ai 11)
    (display "(" wind) (display "CREW INDEX= " wind)
    (display cwi wind) (display ";MAXIMUM HOUR = " wind)
    (display md2 wind) (display ":" wind)
    (display md wind) (ai 14)
    (display ald wind) (display ")" wind)))
(ai 15)
(newline)))

```

นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; condition DELAY (FUEL) ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define ful
  (lambda ()
    (cond ((eq? fuel 'y) (display " NO FUEL-PROBLEM" wind)))
    (if (and (eq? fuel 'n)
              (>= tdep fuelf)
              (<= tdep fuelt))
        (begin (ai 13)
                (display " FUEL-PROBLEM " wind)
                (ai 12)
                (display "(" wind)
                (display fuelf wind)
                (display " TO " wind)
                (display fuelt wind)
                (display ")" wind)))
        (if (and (eq? fuel 'n)
                  (or (< tdep fuelf) (> tdep fuelt)))
            (display " NO FUEL-PROBLEM " wind)
            (ai 15)))
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;; condition of REROUTE (PERMIT) ;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (define pml
      (lambda ()
        (cond ((eq? per 'y)
              (display " PERMIT FOR NEW AIRPORT USING THAT A/C ALREADY APPLIED" wi
nd)))
        (cond ((eq? per 'n)
              (ai 13)
              (display " NO PERMIT FOR NEW AIRPORT USING THAT A/C ALREADY APPLIED"
wind)))
        (ai 15)))
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;; condition of REROUTE (PERMIT) ;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (define pm2
      (lambda ()
        (cond ((eq? per 'y)
              (display " PERMIT FOR EACH AIRPORT USING THAT A/C ALREADY APPLIED" w
ind)))
        (cond ((eq? per 'n)
              (ai 13)
              (display " NO PERMIT FOR EACH AIRPORT USING THAT A/C ALREADY APPLIED
" wind)))
        (ai 15)))
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;; condition DELAY (AIRPORT SLOT) ;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (define sl1
      (lambda ()

```

เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(cond ((eq? slot 'y) (display " SLOT IS AVIALABLE" wind)))
(if (and (eq? slot 'n)
        (>= tdep slotf)
        (<= tdep slott))
    (begin (ai 13)
           (display " SLOT IS NOT AVAILABLE " wind)
           (ai 12)
           (display "(" wind)
           (display slotf wind)
           (display " TO " wind)
           (display slott wind)
           (display ")" wind)))
(ai 15)
(if (and (eq? slot 'n)
        (or (< tdep slotf) (> tdep slott)))
    (begin (display " SLOT IS AVAILABLE " wind)
           (ai 12)
           (display "(" wind)
           (display slotf wind)
           (display " TO " wind)
           (display slott wind)
           (display ")" wind)
           (ai 15))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; condition DELAY (A/P FACILITY) ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define fcl
  (lambda ()
    (cond ((eq? fac 'y) (display " FACILITY IS AVAILABLE" wind)))
    (if (and (eq? fac 'n)
            (>= tdep facf)
            (<= tdep fact))
        (begin (ai 13)
               (display " FACILITY IS NOT AVAILABLE" wind)
               (ai 12)
               (display "(" wind)
               (display facf wind)
               (display " TO " wind)
               (display fact wind)
               (display ")" wind)))
        (if (and (eq? fac 'n)
                (or (< tdep facf) (> tdep fact)))
            (begin
              (display " FACILITY IS AVAILABLE " wind)
              (ai 12)
              (display "(" wind)
              (display facf wind)
              (display " TO " wind)
              (display fact wind)
              (display ")" wind)))
            (ai 15))))))

```

;;;
 ;;; condition DELAY (A\P CURFEW) ;;;
 ;;;
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define cul
  (lambda ()
    (cond ((eq? cur 'y) (display " NO CURFEW " wind)))
    (if (and (eq? cur 'n)
              (>= tdep curf)
              (<= tdep curt))
        (begin (ai 13)
                (display " HAS CURFEW " wind)
                (ai 12)
                (display "(" wind)
                (display curf wind)
                (display " TO " wind)
                (display curt wind)
                (display ")" wind)))
        (if (and (eq? cur 'n)
                  (or (< tdep curf) (> tdep curt)))
            (begin (display " NO CURFEW " wind)
                    (ai 12)
                    (display "(" wind)
                    (display curf wind)
                    (display " TO " wind)
                    (display curt wind)
                    (display ")" wind)))
            (ai 15)))
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;; condition DELAY (A\ P RUNWAY) ;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (define rwl
      (lambda ()
        (cond ((eq? rw 'y) (display " RUNWAY NOT PROBLEM " wind)))
        (if (and (eq? rw 'n)
                  (>= tdep rwf)
                  (<= tdep rwt))
            (begin (ai 13)
                    (display " RUNWAY HAS PROBLEM " wind)
                    (ai 12)
                    (display "(" wind)
                    (display rwf wind)
                    (display " TO " wind)
                    (display rwt wind)
                    (display ")" wind)))
            (if (and (eq? rw 'n)
                      (or (< tdep rwf) (> tdep rwt)))
                (begin (display " RUNWAY HAS NOT PROBLEM" wind)
                        (ai 12)
                        (display " (" wind)
                        (display rwf wind)
                        (display " TO " wind)
                        (display rwt wind)
                        (display ") " wind)))
                (ai 15)))
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;;; condition DELAY (WEATHER) ;;;
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define wx1
  (lambda ()
    (cond ((eq? wx '1) (display " WEATHER IS CLOSE-CLOSE" wind)))
    (cond ((eq? wx '2) (display " WEATHER IS MARGINAL-CLOSE" wind)))
    (cond ((eq? wx '3) (display " WEATHER IS OPEN-CLOSE" wind)))
    (cond ((eq? wx '4) (display " WEATHER IS CLOSE-MAGINAL" wind)))
    (cond ((eq? wx '5) (display " WEATHER IS CLOSE-OPEN" wind)))
    (cond ((eq? wx '6) (display " WEATHER IS MAGINAL-MAGINAL" wind)))
    (cond ((eq? wx '7) (display " WEATHER IS OPEN-MAGINAL" wind)))
    (cond ((eq? wx '8) (display " WEATHER IS MAGINAL-OPEN" wind)))
    (cond ((eq? wx '9) (display " WEATHER IS OPEN-OPEN" wind))))
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;; condition DELAY (POLITICS) ;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  (define pol
    (lambda ()
      (cond ((eq? pol 'y) (display " NO POLITICS PROBLEM " wind)))
      (if (and (eq? pol 'n)
              (>= tdep polf)
              (<= tdep polt))
          (begin (ai 13)
                 (display " POLITICS PROBLEM" wind)
                 (ai 12)
                 (display "(" wind)
                 (display polf wind)
                 (display " TO " wind)
                 (display polt wind)
                 (display ")" wind)))
          (ai 15)
          (if (and (eq? pol 'n)
                  (or (< tdep polf) (> tdep polt)))
              (begin (display " NO POLITICS PROBLEM " wind)
                     (ai 12)
                     (display "(" wind)
                     (display polf wind)
                     (display " TO " wind)
                     (display polt wind)
                     (display ")" wind)))
              (ai 15))))
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;; CONDITION DELAY (INDUSTRY) ;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  (define id1
    (lambda ()
      (cond ((eq? ind 'y) (display " NO INDUSTRY PROBLEM" wind)))
      (if (and (eq? ind 'n)
              (>= tdep indf)
              (<= tdep indt))
          (begin (ai 13)
                 (display " INDUSTRIAL PROBLEM " wind)
                 (ai 12)
                 (display "(" wind)
                 (display indf wind)
                 (display " TO " wind)
                 (display indt wind))
          (ai 15))))

```

ไม่ว่าการนี้ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(display "" wind)))
(ai 15)
(if (and (eq? ind 'n)
         (or (< tdep indf) (> tdep indt)))
    (begin (display " NO INDUSTRIAL PROBLEM " wind)))
(ai 15)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; PAX CONVENIENCE OF DEPARTURE ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define pl
  (lambda ()
    (window-clear 'console)
    (window-clear wind)
    (window-set-cursor! wind 0 15)
    (ma1)
    (ma2)
    (set! fac (list-ref app 7))
    (ai 11)
    (display " PAX CONVENIENCE " wind)
    (cond ((eq? fac 'y)
           (set! px 'good)
           (set! cf 'GOOD))
          (else (and (set! px 'BAD)
                     (set! cf 'BAD))))
    (window-set-cursor! wind 2 3)
    (ai 14)
    (display " AIRPORT FACILITY IS " wind)
    (display px wind)
    (window-set-cursor! wind 4 3)
    (display " CITY FACILITY (HOTEL) IS " wind)
    (display cf wind)
    (ai 15)
    (cond ((eq? vip 'a)
           (window-set-cursor! wind 8 3)
           (ai 10)
           (display " THIS FLIGHT HAS VIP " wind)
           (ai 15))
          (else (window-set-cursor! wind 10 3)
                (ai 10)
                (display " THIS FLIGHT HAS NOT VIP " wind)))
    (AI 15)
    (cond ((eq? ic 'a)
           (window-set-cursor! wind 12 3)
           (display " THIS FLIGHT IS HIGH COMPETITIVE " wind)))
    (cond ((eq? ic 'b)
           (window-set-cursor! wind 12 3)
           (display " THIS FLIGHT IS COMPETITIVE " wind)))
    (cond ((eq? ic 'c)
           (window-set-cursor! wind 12 3)
           (display " THIS FLIGHT IS LOW COMPETITIVE " wind))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; PAX CONVENIENCE (P2) ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(define p2
  (lambda ()
    (ma1) (ma2)
    (window-clear 'console)
    (window-clear wind)
    (window-set-cursor! wind 0 15)
    (AI 11)
    (display " PAX CONVENIENCE " wind)
    (set! fac (list-ref app 7))
      (set! cf 'GOOD))
      (else (and (set! px 'BAD)
                (set! cf ' BAD))))))
(window-set-cursor! wind 4 3)
(AI 14)
(display " AIRPORT FACILITY IS " wind)
(window-set-cursor! wind 6 3)
(display " CITY FACILITY (HOTEL) IS " wind)
(display cf wind)
(AI 15)
  (cond ((eq? vip 'a)
        (window-set-cursor! wind 10 3)
        (ai 10)
        (display " THIS FLIGHT HAS VIP " wind)
        (ai 15))
        (else (window-set-cursor! wind 10 3)
              (ai 10)
              (display " THIS FLIGHT HAS NOT VIP " wind))))
(AI 15)
(cond ((eq? ic 'a)
      (window-set-cursor! wind 12 3)
      (display " THIS FLIGHT IS HIGH COMPETITIVE " wind)))
(cond ((eq? ic 'b)
      (window-set-cursor! wind 12 3)
      (display " THIS FLIGHT IS COMPETITIVE " wind)))
(cond ((eq? ic 'c)
      (window-set-cursor! wind 12 3)
      (display " THIS FLIGHT IS LOW COMPETITIVE " wind))))
;;;;;;;;;;;;; NEXT FLIGHT/ ACTIVITY ;;;;;;;;;;;;;;
;;;;;;;;;;;;;
(define f1
  (lambda ()
    (window-set-cursor! wind 14 3)
    (display " NOT EFFECTS TO NEXT FLIGHT/ACTIVITY " wind)))
(define f2
  (lambda ()
    (window-set-cursor! wind 14 3)
    (display " HAS EFFECTS TO NEXT FLIGHT/ACTIVITY " wind)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; CONDITION OF CANCEL COST ;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define cl
  (lambda ()
    (cond ((and (eq? (length flt) 2)(and (eq? a dep2) (eq? q arr2)))
      (set! same 'y)
      (else (set! same 'n)))
    (cond ((and (eq? (length flt) 1) (and (eq? a dep) (eq? q arr)))
      (set! same 'y)
      (else (set! same 'n)))
    (window-set-cursor! wind 4 6)
    (ai 13)
    (display " CANCEL COST OF FLIGHT = TG " wind)
    (display fltno wind)
    (ai 15)
    (window-set-cursor! wind 6 3)
    (display " DIRECT OPERATING COST " wind)
    (if (and (eq? same 'y)
      (<= (- bkng2 (- mseatl bkng1)) 0)
      (< (/ (- tidepl tr1) 100) 4.00))
      (display " THIS FLIGHT IS ONLY DIRECT OPERATING COST "))
    (if (and (eq? same 'y)
      (> (/ (- tidepl tr1) 100) 4.00))
      (begin (window-set-cursor! wind 8 3)
        (display " ACCOMMODATION COST (PER NIGHT) = " wind)
        (display (* 2500 bkng1) wind)))
    (if (and (eq? same 'y)
      (> (- bkng2 (- mseatl bkng1)) 0)
      (<= (/ (- tidepl tr1) 100) 4.00))
      (begin (window-set-cursor! wind 8 3)
        (display " REST PAX = " wind)
        (display (- bkng2 (- mseatl bkng1)) wind)
        (window-set-cursor! wind 10 3)
        (display " CHARTER COST " wind)
        (window-set-cursor! wind 12 3)
        (display " REVENUE LOSS " wind)))
    (if (and (eq? same 'y)
      (> (- bkng2 (- mseatl bkng1)) 0)
      (> (/ (- tidepl tr1) 100)
        4.00))
      (begin (window-set-cursor! wind 8 3)
        (display " ACCOMMODATION COST (PER NIGHT) = " wind)
        (display (* 2500 bkng1) wind)
        (window-set-cursor! wind 10 3)
        (display " REST PAX = " wind)
        (display (- bkng2 (- mseatl bkng1)) wind)
        (window-set-cursor! wind 12 3)
        (display " CHARTER COST " wind)
        (window-set-cursor! wind 14 3)
        (display " REVENUE LOSS " wind)))
    (cond ((eq? same 'n)
      (begin (window-set-cursor! wind 8 3)
        (display " DISTANCE = " wind)
        (display dst2 wind)

```

เอกสารนี้เป็นลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(window-set-cursor! wind 10 3)
(display " REVENUE LOSS IN CASE OF TRANSFER " wind)
(window-set-cursor! wind 12 3)
(display " CALCULATE CHARTER COST " wind))))))

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::: CONDITION OF CANCEL SECTOR COST ::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
(define c100
  (lambda ()
    (cond ((and (eq? a dep2) (eq? q arr2))
      (set! same 'y)
      (else (set! same 'n)))
    (window-set-cursor! wind 4 6)
    (ai 13)
    (display " CANCEL COST OF FLIGHT = TG " wind)
    (display fltno wind)
    (ai 15)
    (window-set-cursor! wind 6 3)
    (display " DIRECT OPERATING COST " wind)
    (if (and (eq? same 'y)
      (<= (- bkng (- mseatl bkng1)) 0)
      (< (/ (- tidepl tr1) 100) 4.00))
      (display " THIS FLIGHT IS ONLY DIRECT OPERATING COST "))
    (if (and (eq? same 'y)
      (<= (- bkng (- mseatl bkng1)) 0)
      (> (/ (- tidepl tr1) 100) 4.00))
      (begin (window-set-cursor! wind 8 3)
        (display " ACCOMMODATION COST (PER NIGHT) = " wind)
        (display (* 2500 bkng) wind)))
      (if (and (eq? same 'y)
        (> (- bkng (- mseatl bkng1)) 0)
        (<= (/ (- tidepl tr1) 100) 4.00))
          (begin (window-set-cursor! wind 8 3)
            (display " REST PAX = " wind)
            (display (- bkng (- mseatl bkng1)) wind)
            (window-set-cursor! wind 10 3)
            (display " CHARTER COST " wind)
            (window-set-cursor! wind 12 3)
            (display " REVENUE LOSS " wind)))
          (if (and (eq? same 'y)
            (> (- bkng (- mseatl bkng1)) 0)
            (> (/ (- tidepl tr1) 100)
              4.00))
              (begin (window-set-cursor! wind 8 3)
                (display " ACCOMMODATION COST (PER NIGHT) = " wind)
                (display (* 2500 bkng) wind)
                (window-set-cursor! wind 10 3)
                (display " REST PAX = " wind)
                (display (- bkng (- mseatl bkng1)) wind)
                (window-set-cursor! wind 12 3)
                (display " CHARTER COST " wind)
                (window-set-cursor! wind 14 3)
                (display " REVENUE LOSS " wind)))
              (if (and (eq? same 'n)
                (window-set-cursor! wind 10 3)
                (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
                (window-set-cursor! wind 12 3)
                (display " CALCULATE CHARTER COST " wind))))))

```

ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(begin (window-set-cursor! wind 8 3)
  (display " DISTANCE = " wind)
  (display dstit wind)
  (window-set-cursor! wind 10 3)
  (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
  (window-set-cursor! wind 12 3)
  (display " CALCULATE CHARTER COST " wind))))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;          CONDITION OF DELAY COST          ;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
(define c2
  (lambda ()
    (window-clear 'console)
    (window-clear wind)
    (window-set-cursor! wind 0 15)
    ;;;(ai 95)
    (display " DELAY COST OF TG " wind)
    (window-set-cursor! wind 1 3)
    (ai 10)
    (display " DIRECT OPERATING COST " wind)
    (ai 15)
    (cond ((< dela 1)
      (display " THIS FLIGHT IS ONLY DIRECT OPERATION COST " wind))
      (else
       (if (and (eq? same 'y) ;;2
                (<= (- bkng (- mseatl bkng1)) 0)
                (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
           (begin (window-set-cursor! wind 2 3)
             (display " MEAL COST = " wind)
             (display (* 200 bkng) wind)))
          (if (and (eq? same 'y) ;;3
                  (<= (- bkng (- mseatl bkng1)) 0)
                  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
                  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 8.00)
                  (and (> tdep 0100) (< tdep 1000)))
              (and (window-set-cursor! wind 4 3)
                (display " ACCOMMODATION COST (PER NIGHT) = " wind)
                (display (* 1500 bkng) wind)))
             (if (and (eq? same 'y) ;;4
                     (<= (- bkng (- mseatl bkng1)) 0)
                     (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
                     (or (< tdep 0100) (> tdep 1000)))
                 (begin (window-set-cursor! wind 6 3)
                   (display " ACCOMMODATION COST (PER NIGHT) = " wind)
                   (display (* 2500 bkng) wind)))
                  (if (and (eq? same 'y) ;;5
                          (<= (- bkng (- mseatl bkng1)) 0)
                          (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
                          (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 8.00))
                      (begin (window-set-cursor! wind 8 3)
                        (display " ACCOMMODATION COST = " wind)
                        (display (* 2500 bkng) wind)))
                     (if (and (eq? same 'y) ;;6
                               (> (- bkng (- mseatl bkng1)) 0)
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยสุโขทัยฯ ห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
(begin (window-set-cursor! wind 10 3)
  (display " DISTANCE = " wind)
  (window-set-cursor! wind 12 3)
  (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
  (window-set-cursor! wind 14 3)
  (display " CALCULATE CHARTER COST " wind)
  (window-set-cursor! wind 16 3)
  (display " REST PAX = " wind)
  (display (- bkng (- mseat1 bkng1)) wind)
  (window-set-cursor! wind 18 3)
  (display " MEAL COST = " wind)
  (display (* 200 bkng) wind)))
(if (and (eq? same 'y) ;;7
  (> (- bkng (- mseat1 bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 8.00)
  (or (< tdep 0100) (> tdep 1000)))
(begin (window-set-cursor! wind 10 3)
  (display " DISTANCE = " wind)
  (display dst wind)
  (window-set-cursor! wind 12 3)
  (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
  (window-set-cursor! wind 14 3)
  (display " CALCULATE CHARTER COST " wind)
  (window-set-cursor! wind 16 3)
  (display " REST PAX = " wind)
  (display (- bkng (- mseat1 bkng1)) wind)
  (window-set-cursor! wind 18 3)
  (display " ACCOMMODATION COST = " wind)
  (display (* 1500 bkng) wind)))
(if (and (eq? same 'y) ;;8
  (> (- bkng (- mseat1 bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 8.00)
  (and (> tdep 0100) (< tdep 1000)))
(begin (window-set-cursor! wind 10 3)
  (display " DISTANCE = " wind)
  (display dst wind)
  (window-set-cursor! wind 12 3)
  (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
  (window-set-cursor! wind 14 3)
  (display " CALCULATE CHARTER COST " wind)
  (window-set-cursor! wind 16 3)
  (display " REST PAX = " wind)
  (display (- bkng (- mseat1 bkng1)) wind)
  (window-set-cursor! wind 18 3)
  (display " ACCOMMODATION COST = " wind)
  (display (* 2500 bkng) wind)))
(if (and (eq? same 'y) ;;9
  (> (- bkng (- mseat1 bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 8.00))
(begin (window-set-cursor! wind 10 3)
  (display " DISTANCE = " wind)
  (display dst wind)

```

```

(window-set-cursor! wind 12 3)
(display " REVENUE LOSS IN CASE OF TRANSFER " wind)
(window-set-cursor! wind 14 3)
(display " CALCULATE CHARTER COST " wind)
(window-set-cursor! wind 16 3)
(display " REST PAX = " wind)
(display (- bkng (- mseat1 bkng1)) wind)
(window-set-cursor! wind 18 3)
(display " ACCOMMODATION COST = " wind)
(display (* 2500 bkng) wind)))
(if (eq? same 'n) ;;10
    (begin (window-set-cursor! wind 10 3)
            (display " DISTANCE = " wind)
            (display dst wind)
            (window-set-cursor! wind 12 3)
            (display " REVENUE LOSS IN CASE OF TRANSFER " wind)
            (window-set-cursor! wind 14 3)
            (display " CALCULATE CHARTER COST " wind))))
(if (and (eq? same 'n) ;;11
        (<= dela 4.00))
    (begin (window-set-cursor! wind 16 3)
            (display " CALCULATE MEAL = " wind)
            (display (* 200 bkng) wind)))
(if (and (eq? same 'n) ;;12
        (> dela 4.00)
        (<= dela 8.00)
        (and (> tdep 0100) (< tdep 1000)))
    (begin (window-set-cursor! wind 16 3)
            (display " ACCOMMODATION COST = " WIND)
            (display (* 1500 bkng) wind)))
(if (and (eq? same 'n) ;;13
        (> dela 4.00)
        (<= dela 8.00)
        (or (< tdep 0100) (> tdep 1000)))
    (begin (window-set-cursor! wind 16 3)
            (display " ACCOMMODATION COST = " wind)
            (display (* 2500 bkng) wind)))
(if (and (eq? same 'n) ;;14
        (> dela 4.00)
        (> dela 8.00))
    (begin
        (window-set-cursor! wind 16 3)
        (display " ACCOMMODATION COST = " WIND)
        (display (* 2500 bkng) wind))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; CONDITION OF REROUTE COST ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(define c3
  (lambda ()
    (window-clear 'console)
    (window-clear wind)
    (window-set-cursor! wind 0 15)

```

```

(ai 95)นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

```

(display " REROUTE COST OF TG " wind)

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(display fltno wind)
(ai 15)
(if (eq? same 'n)
    (begin (window-set-cursor! wind 2 3)
            (display " DIRECT OPERATING COST " wind)
            (window-set-cursor! wind 4 3)
            (display " AIRBORN HOUR = " wind)
            (display flhr1 wind)
            (window-set-cursor! wind 8 3)
            (display " REVENUE LOSS " wind)))
(if (and (eq? same 'y)
         (<= (- bkng (- mseat1 bkng1)) 0)
         (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
    (begin (display " AIRBORN HOUR = " wind)
            (display flhr1 wind)
            (window-set-cursor! wind 2 3)
            (display " DIRECT OPERATING COST " wind)))
(if (and (eq? same 'y)
         (<= (- bkng (- mseat1 bkng1)) 0)
         (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
    (begin (window-set-cursor! wind 2 3)
            (display " DIRECT OPERATING COST " wind)
            (window-set-cursor! wind 4 3)
            (display " AIRBORN HOUR = " wind)
            (display flhr1 wind)
            (window-set-cursor! wind 10 3)
            (display " ACCOMMODATION COST (PER NIGHT) = " wind)))
(if (and (eq? same 'y)
         (> (- bkng (- mseat1 bkng1)) 0))
    (begin (window-set-cursor! wind 12 3)
            (display " REST PAX = " wind)
            (display (- bkng (- mseat1 bkng1)) wind)
            (window-set-cursor! wind 14 3)
            (display " DISTANCE = " wind)
            (display dst wind)
            (window-set-cursor! wind 8 3)
            (display " REVENUE LOSS TRANSFER " wind)))
(if (and (eq? same 'y)
         (> (- bkng (- mseat1 bkng1)) 0)
         (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
    (begin (window-set-cursor! wind 6 3)
            (display " DIRECT OPERATING COST " wind)))
(if (and (eq? same 'y)
         (> (- bkng (- mseat1 bkng1)) 0)
         (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
    (begin (window-set-cursor! wind 10 3)
            (display " ACCOMMODATION COST (PER NIGHT) = " wind)
            (display (* 2500 bkng) wind))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; COST OF CHANGE AIRCRAFT SAME TYPE-ONTIME ;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define c4

```

```

  (lambda ()
    (window-clear 'console)

```

ไม่วารณิใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (<= dela 4.00))
  (begin (display " MEAL COST = " wind)
    (display (* 200 bkng) wind)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (<= dela 8.00)
  (and (> tdep 0100) (< tdep 1000)))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 1500 bkng) wind)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (<= dela 8.00)
  (or (< tdep 0100) (> tdep 1000)))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (> dela 8.00))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (< dela 1))
  (begin (window-set-cursor! wind 2 3)
    (display " AIRBORNE HOUR = " wind)
    (display (/ flhr 100) wind)
    (window-set-cursor! wind 4 3)
    (display " DIRECT OPERATING COST " wind)
    (window-set-cursor! wind 6 3)))
(if (and (eq? same 'y)
  (<= (- bkng (- mseatl bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)

```

```

(> dela 1)
(<= dela 4.00)
(begin (display " MEAL COST = " wind)
      (display (* 200 bkng) wind)))
(if (and (eq? same 'y)
        (<= (- bkng (- mseat1 bkng1)) 0)
        (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
        (> dela 1)
        (> dela 4.00)
        (<= dela 8.00)
        (and (> tdep 0100) (< tdep 1000))))
(begin (window-set-cursor! wind 6 3)
      (display " ACCOMMODATION COST (PER NIGHT) = " wind)
      (display (* 1500 bkng) wind)))
(if (and (eq? same 'y)
        (<= (- bkng (- mseat1 bkng1)) 0)
        (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
        (> dela 1)
        (> dela 4.00)
        (<= dela 8.00)
        (or (< tdep 0100) (> tdep 1000))))
(begin (window-set-cursor! wind 6 3)
      (display " ACCOMMODATION COST (PER NIGHT) = " wind)
      (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
        (<= (- bkng (- mseat1 bkng1)) 0)
        (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
        (> dela 1)
        (> dela 4.00)
        (> dela 8.00))
    (begin (window-set-cursor! wind 6 3)
          (display " ACCOMMODATION COST (PER NIGHT) = " wind)
          (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
        (> (- bkng (- mseat1 bkng1)) 0)) ;; CAN NOT TRANSFER ALL PAX
    (begin (window-set-cursor! wind 8 3)
          (display " REST PAX = " wind)
          (display (- bkng (- mseat1 bkng1)) wind)
          (window-set-cursor! wind 10 3)
          (display " DISTANCE = " wind)
          (display dst wind)
          (window-set-cursor! wind 12 3)
          (display " REVENUE LOSS " wind)
          (window-set-cursor! wind 14 3)
          (display " CHARTER COST " wind)))
    (if (and (eq? same 'y)
            (> (- bkng (- mseat1 bkng1)) 0)
            (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
            (< dela 1))
        (begin (window-set-cursor! wind 2 3)
              (display " AIRBORNE HOUR = " wind)
              (display (/ flhr 100) wind)
              (window-set-cursor! wind 4 3)
              (display " DIRECT OPERATING COST " wind)
              (window-set-cursor! wind 6 3)))

```

เอกสารที่แสดงนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (<= dela 4.00))
  (begin
    (window-set-cursor! wind 6 3)
    (display " MEAL COST = " wind)
    (display (* 200 bkng) wind)))
(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (<= dela 8.00)
  (and (> tdep 0100) (< tdep 1000)))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 1500 bkng) wind)))
(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (<= dela 8.00)
  (or (< tdep 0100) (> tdep 1000)))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (<= (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 1)
  (> dela 4.00)
  (> dela 8.00))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (> dela 8.00))
  (begin (window-set-cursor! wind 6 3)
    (display " ACCOMMODATION COST (PER NIGHT) = " wind)
    (display (* 2500 bkng) wind)))
(if (and (eq? same 'y)
  (> (- bkng (- mseatl bkng1)) 0)
  (> (/ (- (list-ref samed 5) (list-ref sec 5)) 100) 4.00)
  (< dela 1))
  (begin (window-set-cursor! wind 2 3)
    (display " AIRBORNE HOUR = " wind)
    (display (/ flhr 100) wind)
    (window-set-cursor! wind 4 3)
    (display " DIRECT OPERATING COST " wind)
    (window-set-cursor! wind 6 3)))

```

เอกสารที่แนบมาเป็นเอกสารที่จัดทำขึ้นโดยกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

(if (and (eq? same 'n)
        (> dela 1)
        (<= dela 4.00))
    (begin (window-set-cursor! wind 6 3)
           (display " MEAL COST = " wind)
           (display (* 200 bkng) wind)))
(if (and (eq? same 'n)
        (> dela 1)
        (> dela 4.00)
        (<= dela 8.00)
        (and (> tdep 0100) (< tdep 1000)))
    (begin (window-set-cursor! wind 6 3)
           (display " ACCOMMODATION COST (PER NIGHT) = " wind)
           (display (* 1500 bkng) wind)))
(if (and (eq? same 'n)
        (> dela 1)
        (> dela 4.00)
        (<= dela 8.00)
        (or (< tdep 0100) (> tdep 1000)))
    (begin (window-set-cursor! wind 6 3)
           (display " ACCOMMODATION COST (PER NIGHT) = " wind)
           (display (* 2500 bkng) wind)))
(if (and (eq? same 'n)
        (> dela 1)
        (> dela 4.00)
        (> dela 8.00))
    (begin (window-set-cursor! wind 6 3)
           (display " ACCOMMODATION COST (PER NIGHT) = " wind)
           (display (* 2500 bkng) wind))))))

```

```

(define stt
  (lambda ()
    (set! s ai)
    (set! k window-set-cursor!)
    (set! w wind)
    (set! wind ())
    (set! window-set-cursor! (lambda (x y z) (newline)))
    (set! ai (lambda (x) nil))))
  (define sendd
    (lambda ()
      (set! wind w)
      (set! ai s)
      (set! window-set-cursor! k)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;; DELAY OUTPUT TO PRINTER ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define w) (define k)
(define s)
(define delly
  (lambda ()
    (stt)
    (with-output-to-file "sii.tmp"
      (lambda ()
        (display " ACTION DELAY ")
        (newline)(newline)
        (display " FLIGHT-NO = ")
        (display fltno)
        (display " SECTOR OF FLIGHT = ")
        (display secno)
        (newline)
        (display " TIME TO DELAY (YYMMDDhhmm) = ")
        (display tdep)
        (newline)
        (display " AIRPORT (DEPARTURE) : ")
        (display dep)
        (newline)
        (newline) (ac1) (newline) (cw1)
        (newline) (ful) (newline) (sll)
        (newline) (fc1) (newline) (cul)
        (newline) (rw1) (newline) (wx1)
        (newline) (pol) (newline) (idl)
        (ma2)
        (newline)
        (set! arr (list-ref sec 6))
        (newline)
        (newline)
        (display " AIRPORT (ARRIVAL) : ")
        (display arr )
        (newline) (ac1) (newline) (cw1)
        (newline) (ful) (newline) (sll)
        (newline) (fc1) (newline) (cul)
        (newline) (rw1) (newline) (wx1)

```

ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม้วารณใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(newline) (po1) (newline) (id1)
(newline) (newline)
(c2)
(dos-call "" "p"))
(sendd)
(submanu))
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::: CANCEL FLIGHT OUTPUT TO PRINTER ::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
(define c01
  (lambda ()
    (stt)
      (with-output-to-file "sii.tmp"
        (lambda ()
          (c1)
            (dos-call "" "p"))
          (sendd)
          (submanu)))
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    :::::::::::: CANCEL SECTOR OUTPUT TO PRINTER ::::::::::
    ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
    (define c11
      (lambda ()
        (stt)
          (with-output-to-file "sii.tmp"
            (lambda ()
              (window-set-cursor! wind 0 3)
              (ai 11)
              (display " CANCEL SECTOR : " wind)
              (display seq wind)
              (window-set-cursor! wind 0 23)
              (ai 14)
              (display " DEPARTURE : " wind)
              (display (list-ref sec2 4) wind)
              (display " ARRIVAL : " wind)
              (display (list-ref sec2 6) wind)
              (c100)
              (dos-call "" "p"))
              (sendd)
              (submanu)))
            ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            :::::::::::: REROUTE OUTPUT TO PRINTER ::::::::::
            ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
            (define rer1
              (lambda ()
                (stt)
                  (with-output-to-file "sii.tmp"
                    (lambda ()
                      (window-set-cursor! wind 0 12)
                      (display " AT AIRPORT " wind)
                      (display " (ARRIVAL) : " wind)
                      (display arr wind)
                      (window-set-cursor! wind 2 0) (ac2)
                      (window-set-cursor! wind 4 0) (cw1)

```



```

(window-set-cursor! wind 6 0) (ful)
(window-set-cursor! wind 8 0) (sl1)
(window-set-cursor! wind 10 0) (fc1)
(window-set-cursor! wind 12 0) (cu1)
(window-set-cursor! wind 14 0) (rw1)
(window-set-cursor! wind 16 0) (wx1)
(window-set-cursor! wind 18 0) (po1)
(window-set-cursor! wind 20 0) (id1)
(window-set-cursor! wind 22 0) (pm1)
(begin
  (display (integer->char 2))
  (window-set-cursor! wind 25 10)
  (read-char) wind)
(c3)
(dos-call "" "p"))
(sendd)
(submanu)))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; CHANGE AIRCRAFT OUTPUT TO PRINTER ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(define sson
  (lambda ()
    (stt)
    (with-output-to-file "sii.tmp"
      (lambda ()
        (display " CHANGE A/C SAME TYPE ONTIME " wind)
        (ai 15) (ac3) (read-char)(c4)(read-char)
        (dos-call "" "p"))))
    (sendd)
    (submanu)))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้