



TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

TEACH : A Microcomputer-Based Authoring System

นายนิพนธ์ ศุภศรี

Nipon Suppasri

วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

Master of Science Thesis in Computer Science

Prince of Songkla University

2537

๑

เลขหมู่	๐๙๗๖.๗๖.๕๔๒	๗๖๑	๒๕๓๗	๑.๒
Bib Key	๖๖๗๐๓			

(1)

ชื่อวิทยานิพนธ์ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์
ผู้เขียน นายนิพนธ์ ศุภศิริ
สาขาวิชา วิทยาการคอมพิวเตอร์

คณะกรรมการที่ปรึกษา

คณะกรรมการสอบ

..... ประธานกรรมการ
(อาจารย์วุฒิพงศ์ เตชะดำรงสิน)

..... ประธานกรรมการ
(อาจารย์วุฒิพงศ์ เตชะดำรงสิน)

..... กรรมการ
(รองศาสตราจารย์ไพโรจน์ ตีรณธนากุล)

..... กรรมการ
(รองศาสตราจารย์ไพโรจน์ ตีรณธนากุล)

..... กรรมการ
(อาจารย์อำนาจ เปาะทอง)

..... กรรมการ
(นายแพทย์อารักษ์ ทองปิยะภูมิ)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยรับนี้เป็น
ส่วนหนึ่งของการศึกษา ตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....

(ดร.ไพรัตน์ สงวนไทร)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์
ผู้เขียน นายนิพนธ์ ศุกศรี
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2536

บทคัดย่อ

TEACH เป็นระบบคอมพิวเตอร์ที่ผู้วิจัยออกแบบให้มีวัตถุประสงค์หลักเพื่อช่วยให้ผู้สอนสามารถสร้างบทเรียนบนเครื่องไมโครคอมพิวเตอร์ได้ด้วยตนเอง และผู้เรียนสามารถนำบทเรียนที่ผู้สอนสร้างไปเรียนรู้ด้วยตนเอง โดยระบบมีการบันทึกข้อมูลพฤติกรรมการเรียน และบันทึกผลการเรียนของผู้เรียน ซึ่งจะเป็นประโยชน์ต่อผู้สอนในการติดตามผลการเรียน การให้คำปรึกษา การให้คำแนะนำแก่ผู้เรียน ตลอดจนใช้ในการปรับปรุงบทเรียนได้ บทเรียนสำหรับระบบนี้ เป็นบทเรียนเชิงข้อความแบบกรอบการเรียนรู้ที่สร้างด้วยโปรแกรมประมวลคำทั่วไป ลักษณะจะประกอบด้วยเนื้อหาบทเรียนและภาษา TAL (TEACH Authoring Language) ซึ่งเป็นภาษาควบคุมการนำเสนอบทเรียนที่ผู้วิจัยได้พัฒนาขึ้น

TEACH ประกอบด้วย 2 โปรแกรมหลัก คือ TEACH.EXE และ ANALYSER.EXE โดย TEACH.EXE จะทำหน้าที่ควบคุมการนำเสนอบทเรียนตามคำสั่งของภาษา TAL บันทึกพฤติกรรมการเรียนและบันทึกสรุปผลการเรียน ส่วน ANALYSER.EXE จะใช้สำหรับเรียกดูสรุป ผลการเรียนและพฤติกรรมการเรียนของผู้เรียน

ในการวิจัยครั้งนี้ ผู้วิจัยได้ออกแบบและพัฒนาภาษา TAL โปรแกรม TEACH.EXE และ ANALYSER.EXE จัดทำคู่มือการใช้ TEACH จัดทำบทเรียน โปรแกรมการใช้ TEACH รวมทั้งได้มีการทดสอบการประยุกต์ใช้งานกับกลุ่มตัวอย่างด้วย และเพื่อให้มีความยืดหยุ่นในการใช้งาน จึงออกแบบระบบให้มีความต้องการอุปกรณ์ขั้นต่ำเป็นไมโครคอมพิวเตอร์ 16 บิต หน่วยความจำไม่น้อยกว่า 640 KB จอภาพแบบโมนิโครม ตลอดจนได้เลือกใช้ภาษา C ในการพัฒนา ด้วยเห็นว่าเป็นภาษาโครงสร้างและสามารถแยกส่วนการพัฒนาโปรแกรมออกเป็นผังชั้นย่อย ๆ ซึ่งจะทำให้สามารถปรับปรุงและขยายการพัฒนา TEACH ได้สะดวกยิ่งขึ้นในอนาคต

Thesis Title TEACH : A Microcomputer-Based Authoring System
Author Mr.Nipon Suppasri
Major Program Computer Science
Academic Year 1993

Abstract

TEACH is an authoring software designed to facilitate the teacher to develop courseware themselves, and the students can also study by themselves from the courseware. The system can record the students' learning behavior and performance. This enables the teachers to follow the students' performance, advise them, and also use the results to improve their courseware.

The courseware developed using TEACH is a text-based, frame-type courseware. The content of the courseware can be created by any typical wordprocessor. It consists of subject content and TAL (Teach Authoring System) which is the language designed to control the flow and presentation of the courseware. TEACH comprises 2 main programs, TEACH.EXE and ANALYSER.EXE. TEACH.EXE's function is to control the flow of the lesson, record the students' learning behavior and report their performance. ANALYSER.EXE is used by the teachers to review the students' learning behavior and their performance.

In this research, the author designed and developed TAL, TEACH.EXE, ANALYSER.EXE and create a handbook on how to use TEACH, a courseware to teach TEACH and also test the system with a group of 28 students. To make the system accessible to everyone

, the system was designed using minimum hardware. The computer needed is only a 16 bit PC, with 640 KB main memory and a monochrome monitor. C was chosen to develop the system because it is a structured-language, and the program can be decomposed into modules or functions. This will allow TEACH to be easily modified and expanded in the future.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จได้ด้วยความช่วยเหลืออย่างดียิ่งจากบุคคลที่เกี่ยวข้องหลายท่านดังนี้

อาจารย์วุฒิพงศ์ เตชะดำรงสิน ซึ่งกรุณาเป็นอาจารย์ที่ปรึกษา และรองศาสตราจารย์ไพโรจน์ ตีรณานุกุล ซึ่งกรุณาเป็นอาจารย์ที่ปรึกษาร่วม อาจารย์ทั้งสองท่านได้ให้ความช่วยเหลือให้คำปรึกษา เสนอแนะแนวทาง ตรวจสอบแก้ไขวิทยานิพนธ์ รวมทั้งให้กำลังใจกับผู้วิจัยมาตลอด

คุณวิภัทร ศุภติพรหม เจ้าหน้าที่ศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ ที่ให้คำแนะนำ และช่วยแก้ปัญหาการใช้ภาษา C ในการพัฒนาเพื่อการวิจัยครั้งนี้

คุณวราภรณ์ มั่นเศกวิทย์ และเพื่อนร่วมงาน ที่กรุณาทดสอบและปรับแก้ไขตัวอย่างบทเรียนโปรแกรม TEACH

นักศึกษาชั้นปีที่ 4 ภาควิชาครุศาสตร์และเทคโนโลยี คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี จำนวน 28 คน ที่ได้กรุณานำโปรแกรมไปทดสอบใช้งาน และช่วยประเมินผลแบบสอบถาม

อาจารย์ปวีรยา ศุภศิริ ภรรยาของผู้วิจัย ซึ่งนอกจากจะเป็นกำลังใจที่สำคัญแล้วยังเป็นผู้ช่วยในการจัดหาข้อมูลและจัดทำเอกสารวิทยานิพนธ์ อีกด้วย

นอกจากนี้ผู้วิจัยยังได้รับกำลังใจ ความร่วมมือและช่วยเหลืออย่างดียิ่งจาก มิตรสหายและครอบครัว

ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ โอกาสนี้

นิพนธ์ ศุภศิริ

สารบัญ

	หน้า
บทคัดย่อ.....	(3)
Abstract.....	(4)
กิตติกรรมประกาศ.....	(6)
สารบัญ.....	(7)
รายการตาราง.....	(9)
รายการภาพประกอบ.....	(10)
ตัวย่อและสัญลักษณ์.....	(11)
บทที่	
1. บทนำ.....	1
บทนำต้นเรื่อง.....	1
การตรวจเอกสาร.....	2
วัตถุประสงค์.....	12
ประโยชน์ที่คาดว่าจะได้รับการวิจัย.....	13
2. วิธีการวิจัย.....	14
วัสดุและเครื่องมือ.....	14
อุปกรณ์ที่ใช้ในการวิจัย.....	14
สถานที่ทำการวิจัย.....	14
วิธีดำเนินการวิจัย.....	15
3. ผล.....	52
4. บทวิจารณ์และสรุป.....	54
บรรณานุกรม.....	57
ภาคผนวก.....	61
ภาคผนวก ก. รายการโปรแกรม TEACH : ระบบสร้างบทเรียน ด้วยตนเองบนไมโครคอมพิวเตอร์.....	62

ภาคผนวก ข. ภาษา TAL (Teach Authoring Language).....	123
ภาคผนวก ค. แบบฟอร์มออกแบบกรอบการเรียน TEACH.....	130
ภาคผนวก ง. เอกสารคู่มือการใช้ TEACH : ระบบสร้างบทเรียน ด้วยตนเองบนไมโครคอมพิวเตอร์.....	132
ภาคผนวก จ. ตัวอย่างบทเรียนโปรแกรม เรื่อง การใช้ TEACH : ระบบ สร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์.....	145
ภาคผนวก ฉ. สรุปการประเมินผลการใช้ TEACH : ระบบสร้างบทเรียน ด้วยตนเองบนไมโครคอมพิวเตอร์.....	193
ประวัติผู้เขียน.....	196

รายการตาราง

ตาราง

หน้า

1. ตัวอย่าง โปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์.....8

รายการภาพประกอบ

ภาพประกอบ	หน้า
1. โครงสร้างของโปรแกรม เรื่อง ไฟไนต์อิลลิเมนต์.....	10
2. แบบแนะนำลำดับและเวลาที่ใช้ในการศึกษาเรื่องไฟไนต์อิลลิเมนต์.....	11
3. โครงสร้างกรอบการเรียนรู้.....	19
4. ผังงานระบบของระบบ TEACH AUTHORIZING SYSTEM ส่วนที่ 1.....	28
5. ผังงานระบบของระบบ TEACH AUTHORIZING SYSTEM ส่วนที่ 2.....	29
6. ผังงานมอดูลสำหรับโปรแกรม TEACH.EXE.....	30
7. ผังงานมอดูลสำหรับโปรแกรม ANALYSER.EXE.....	31

ตัวย่อและสัญลักษณ์

TAL = TEACH Authoring Language

CAI = Computer Assisted Instruction

บทที่ 1

บทนำ

1. บทนำต้นเรื่อง

ความก้าวหน้าทางวิทยาศาสตร์และเทคโนโลยีในรอบ 20 ปีที่ผ่านมา ทำให้เกิดองค์ความรู้ใหม่มากมาย ซึ่งมนุษย์จำเป็นจะต้องแสวงหาเพื่อให้ทันต่อการเปลี่ยนแปลงของวิทยาการ แต่ด้วยช่วงชีวิตที่จำกัดของมนุษย์ในการเรียนรู้ แนวทางหนึ่งที่จะช่วยให้เกิดการเรียนรู้ได้เร็ว นอกเหนือจากการเรียนในชั้นเรียน ก็คือ การเรียนรู้ด้วยตนเอง ดังอริสโตเติล (Aristotle) กล่าวไว้ว่า มนุษย์สามารถเรียนรู้ด้วยประสบการณ์ของตนเอง ซึ่งเป็นปรัชญาทางการศึกษาที่ยังเชื่อถือได้จนทุกวันนี้

การเรียนรู้ มีทั้งในรูปแบบของการแสวงหาความรู้ด้วยตนเอง และการเรียนในห้องเรียนที่มีครูเป็นผู้สอน การเรียนในห้องเรียน โดยมีครูเป็นผู้สอนส่วนมากใช้วิธีการถ่ายทอดความรู้แบบเผชิญหน้า ซึ่งเป็นวิธีที่มีข้อจำกัดบางประการ กล่าวคือ จำนวนผู้สอนที่มีความเชี่ยวชาญเฉพาะศาสตร์มีน้อย และการสอนผู้เรียนทั้งชั้นเรียน โดยใช้ผู้สอน 1 คน เป็นเรื่องยากที่จะสามารถ สอนผู้เรียนให้เข้าใจได้อย่างลึกซึ้งทุกคน เนื่องจากผู้สอนมีขีดจำกัดในการดูแลให้ได้อย่างทั่วถึง และผู้เรียนมีความสามารถในการรับรู้เร็วช้าไม่เท่ากัน ด้วยเหตุนี้ผู้เรียนบางคนที่มีฐานะทางการเงินดี จึงเสียค่าใช้จ่ายในการจ้างครูมาสอนพิเศษ ทำให้เกิดความเหลื่อมล้ำทางการศึกษาต่อผู้เรียนที่มีฐานะยากจน ดังนั้น การศึกษาในปัจจุบันจึงมีความพยายามที่จะนำความก้าวหน้าทางวิทยาศาสตร์และเทคโนโลยีเข้ามาช่วยให้ผู้เรียนทุกคนมีโอกาสเรียนรู้ได้เท่าเทียมกัน และเกิดการเรียนรู้ที่กว้างขวางขึ้น แนวทางหนึ่งที่ได้รับการสนใจอย่างมากก็คือ การใช้โปรแกรมช่วยการเรียนการสอนบนเครื่องไมโครคอมพิวเตอร์ เพราะสังเกตเห็นถึงคุณประโยชน์ของ โปรแกรมช่วยการเรียนการสอน เมื่อเปรียบเทียบกับการเรียนการสอนแบบเก่าจะมีข้อดีที่กว่า ในแง่ที่ว่า ผู้เรียนมีความสะดวกในการเรียนรู้ เพราะเมื่อใดที่มีความต้องการเรียนก็สามารถเรียนกับระบบโปรแกรมที่ผู้สอนจัดเตรียมไว้ โดยไม่จำเป็นต้องกำหนดนัดหมายเวลากับผู้สอน ผู้เรียนที่มีสติปัญญาต่างกันก็สามารถเรียนรู้ได้เท่าเทียมกัน เพราะ โปรแกรมช่วยการเรียนการสอนมีความสามารถ

ในการสอนซ้ำได้ มีการเก็บสะสมคะแนนจากแบบทดสอบ มีระบบการประเมินผล และบางระบบสามารถให้ข้อมูลย้อนกลับ (feedback) เพื่อพัฒนาระดับการเรียนรู้เฉพาะบุคคล ส่วนผู้สอนเองก็ได้ประโยชน์หลายอย่าง เช่น ไม่ต้องสอนผู้เรียนซ้ำซาก สามารถสอนในกลุ่มผู้เรียนที่กว้างขวางไม่เฉพาะเจาะจงในห้องเรียนเท่านั้น เพราะผู้เรียนสามารถนำไปศึกษาได้เองทั้งที่บ้านหรือที่โรงเรียน และผู้สอนยังสามารถติดตามประเมินผลผู้เรียนได้อย่างสะดวก ตลอดจนสามารถทำการปรับปรุงแก้ไขบทเรียนให้ดีขึ้นได้ง่าย

ด้วยคุณประโยชน์ของ โปรแกรมช่วยการเรียนการสอนที่มีต่อผู้สอน และผู้เรียนดังกล่าวมาแล้ว จึงมีการพัฒนาโปรแกรมช่วยการเรียนการสอนขึ้นใช้งาน โปรแกรมประเภทนี้มีมากกว่า 30 ปีแล้ว โดยมีขึ้นครั้งแรกในต่างประเทศก่อนแล้วแพร่หลายเข้าสู่ประเทศไทย แต่โปรแกรมช่วยการเรียนการสอนของต่างประเทศยังมีขีดความสามารถจำกัดในการประยุกต์ใช้ในการเรียนการสอนในประเทศไทย เพราะเป็นภาษาต่างประเทศ เนื้อหาหลักสูตรการเรียนการสอนต่างกัน เทคนิคการสอนต่างกัน นักคอมพิวเตอร์ในประเทศไทยจึงได้พัฒนาโปรแกรมช่วยการเรียนการสอนขึ้นมาเอง ซึ่งส่วนใหญ่จะมีลักษณะเฉพาะเจาะจงต่อเนื้อเรื่องหรือวิชาั้น ๆ และใช้เฉพาะกลุ่มผู้เรียน ยังไม่สามารถนำไปบรรจุเนื้อหาการเรียนได้ทุกวิชา เนื่องจากการพัฒนาโปรแกรมช่วยการเรียนสอน ต้องใช้ผู้ที่มีความชำนาญภาษาคอมพิวเตอร์ที่มีความสามารถเป็นผู้สอนด้วย อย่างไรก็ตาม ถือได้ว่า โปรแกรมช่วยการเรียนการสอนได้พัฒนาจนก้าวขึ้นสู่อีกระดับหนึ่งแล้ว และยังต้องการนักคอมพิวเตอร์มาพัฒนาต่อ เพื่อให้ได้โปรแกรมที่มีขีดความสามารถมากยิ่งขึ้น ด้วยเหตุผลและความจำเป็นดังกล่าว ผู้วิจัยจึงได้พัฒนาระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ ซึ่งให้ชื่อว่า TEACH เพื่อเป็นโปรแกรมสำเร็จรูปสำหรับผู้สอนใช้สร้างบทเรียน โปรแกรมวิชาต่าง ๆ โดยผู้สอนเป็นผู้พัฒนาบทเรียนบนไมโครคอมพิวเตอร์ด้วยตนเอง สามารถสร้างบทเรียนได้ง่าย และสะดวกโดยไม่จำเป็นต้องมีความรู้ความชำนาญในศาสตร์คอมพิวเตอร์ ซึ่งจะ เป็นระบบที่เหมาะสมกับสถานการณ์ปัจจุบันที่เครื่อง ไมโครคอมพิวเตอร์มีใช้อย่างแพร่หลาย ในวงการศึกษา บริษัท ห้างร้าน หน่วยงานต่าง ๆ และครอบครัว

2. การตรวจเอกสาร

ผู้วิจัยได้ศึกษาเอกสารต่าง ๆ และงานวิจัยที่เกี่ยวข้อง ซึ่งจะนำเสนอตามลำดับ ดังนี้

2.1 แนวคิดเกี่ยวกับคอมพิวเตอร์ช่วยสอน

2.2 แนวคิดเกี่ยวกับโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์

2.3 ตัวอย่างงานพัฒนาที่เกี่ยวข้องกับโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ทั้งในและต่างประเทศ

2.1 แนวคิดเกี่ยวกับคอมพิวเตอร์ช่วยสอน (CAI หรือ Computer Assisted Instruction)

2.1.1 ความเป็นมาของคอมพิวเตอร์ช่วยสอน

แนวคิดเกี่ยวกับการนำคอมพิวเตอร์มาช่วยสอน เริ่มครั้งแรกที่สหรัฐอเมริกาเมื่อปลาย ค.ศ. 1950 มหาวิทยาลัยที่เป็นผู้บุกเบิกเรื่องนี้ คือ มหาวิทยาลัยฟลอริดาและสแตนฟอร์ด ใช้นั้นใช้เครื่องคอมพิวเตอร์ขนาดใหญ่คือ IBM 1500 และใช้ในรูปแบบเทอร์มินัล ภาษาที่ใช้เป็นภาษาระดับสูงเรียกว่าภาษาซีเอไอ วิชาที่ทำตอนแรก ๆ คือ วิชาฟิสิกส์และสถิติ ต่อมาใช้ภาษาเบสิกแทน ทำให้นักศึกษาใช้เครื่องคอมพิวเตอร์ได้ง่ายขึ้น ในปัจจุบันมีภาษาสำหรับการเขียนโปรแกรมมากขึ้น และการใช้รูปแบบเทอร์มินัลต่อกับเครื่องเมนเฟรมไม่คล่องตัวเท่ากับการใช้เครื่องไมโครคอมพิวเตอร์ ซึ่งมีให้แพร่หลายมากขึ้นเรื่อย ๆ จึงทำให้ไมโครคอมพิวเตอร์เข้ามามีบทบาทแทน เพราะเป็นระบบที่ไม่ยุ่งยากและไม่ต้องลงทุนสูงนัก อีกทั้งมีการใช้งานอย่างแพร่หลายอยู่ในโรงเรียนและสถานศึกษาต่าง ๆ มากกว่าคอมพิวเตอร์ประเภทอื่น ๆ

2.1.2 ความหมายของคอมพิวเตอร์ช่วยสอน

คอมพิวเตอร์ช่วยสอนหมายถึงการนำคอมพิวเตอร์มาใช้ในการเรียนการสอน การทบทวน การทำแบบฝึกหัดหรือการประเมินผล(ทักษิณา สวนานนท์, 2529:59) ผู้เรียนนั่งอยู่หน้าเครื่องคอมพิวเตอร์หรือเทอร์มินัลที่ต่อกับเครื่องเมนเฟรม เรียกโปรแกรมสำเร็จรูปที่จัดเตรียมไว้เป็นพิเศษสำหรับการสอนวิชานั้น ๆ ขึ้นมาบนจอ โดยแสดงเรื่องราวเป็นคำอธิบายของบทเรียนหรือแสดงรูปภาพบนจอภาพ แต่ละคนใช้เวลาเรียนไม่เท่ากัน และสามารถสั่งเครื่องคอมพิวเตอร์ให้นำเสนอบทเรียนต่อไปได้ หรืออาจทดสอบความรู้ด้วยการป้อนคำถามซึ่งอาจเป็นแบบฝึกหัดหรือแบบทดสอบ แล้วเครื่องคอมพิวเตอร์จะตรวจให้ มีการชมเชย ให้กำลังใจเมื่อทำถูก ตำหนิหรือต่อว่าบ้างเมื่อทำผิด และอาจสั่งให้ผู้เรียนกลับไปศึกษาใหม่ หรือให้ไปศึกษาบทใหม่ต่อไป เป็นต้น

คอมพิวเตอร์ช่วยสอนเป็นระบบการเรียนการสอน ซึ่งเกิดจากปฏิสัมพันธ์ระหว่างผู้เรียนกับเครื่องคอมพิวเตอร์ โดยผู้เรียนจะศึกษาเนื้อหาจากบทเรียนที่ออกแบบไว้อย่างดีเก็บไว้ในแผ่นจานแม่เหล็ก และจะถูกเรียกแสดงผ่านจอภาพของเครื่องคอม-

พิวเตอร์ เนื้อหาอาจแสดงในรูปของตัวอักษร กราฟิก ภาพเคลื่อนไหว เสียงหรือร่วมกับสิ่งอื่น ๆ ในลักษณะของสื่อประสม เช่น สไลด์ แถบบันทึกเสียง โทรทัศน์ วิทยุทัศน์ ฯลฯ หลังจากแสดงเนื้อหาในเรื่องหนึ่ง ๆ คอมพิวเตอร์จะเสนอแบบฝึกหัดให้ผู้เรียนได้ทบทวนความเข้าใจ และสนองตอบต่อสิ่งนั้นผ่านทางแป้นพิมพ์ หลังจากนั้นเครื่องคอมพิวเตอร์จะพิจารณาตอบสนองว่า ผู้เรียนควรก้าวไปสู่เรื่องใหม่ หรือการซ่อมเสริมก่อน นั่นคือคอมพิวเตอร์ช่วยสอนเป็นระบบการเรียนการสอนแบบโปรแกรมชนิดหนึ่ง หรือตำราในรูปแบบที่เรียกว่าบทเรียนโปรแกรม (courseware) และเสนอบทเรียนได้อย่างซับซ้อนและนำพิศวงกว่าสื่อประเภทบทเรียนแบบอื่น ๆ

2.1.3 องค์ประกอบของคอมพิวเตอร์ช่วยสอน ประกอบด้วยส่วนต่าง ๆ ดังนี้ (พิสนธ์ จงตระกูล และมนตรีรัตน์ จรุงเดชากุล, 2531:705)

2.1.3.1 เครื่องคอมพิวเตอร์และอุปกรณ์ประกอบ

ก. อุปกรณ์สำหรับนำข้อมูลเข้า เช่น keyboard, mouse, light pen เป็นต้น

ข. หน่วยประมวลผลกลาง

ค. อุปกรณ์แสดงผล เช่น จอภาพ เครื่องพิมพ์ เครื่องฉายสไลด์ วิทยุทัศน์ เป็นต้น

2.1.3.2 โปรแกรมควบคุมการทำงานของเครื่องคอมพิวเตอร์

2.1.3.3 บทเรียนช่วยการเรียนการสอน ซึ่งผู้สอนเป็นผู้เตรียมเนื้อหา รูปภาพประกอบ ขั้นตอนของเรื่องราวที่จะสอน ซึ่งเป็นโครงเรื่อง (story board) ที่แสดงลำดับขั้นตอนของส่วนแสดงผลบนจอภาพประกอบตนเองต้องการ และให้ผู้เขียนโปรแกรมร่วมพิจารณาความเป็นไปได้ของโครงเรื่องนั้นๆ แล้วดำเนินการเขียนโปรแกรมให้ทำงานตามลำดับขั้นตอนของโครงเรื่องที่ตรงกับความต้องการของผู้สอน

2.1.4 ประเภทของบทเรียนช่วยการเรียนการสอน บทเรียนนี้มีหลายรูปแบบตามความเหมาะสมของบทเรียนแต่ละประเภทซึ่งสามารถแบ่งออกได้ ดังนี้(ทักษิณา ส่วนานนท์, 2529:63-64)

2.1.4.1 ประเภทฝึกทักษะ(drill and practice) เป็นบทเรียนในลักษณะแบบฝึกหัดให้ผู้เรียนได้ฝึกทักษะตามกรอบแนวคิดที่เรียนไปแล้ว ซึ่งอาจผ่านทางคอมพิวเตอร์ช่วยสอนประเภทอื่น หรือการเรียนในชั้นเรียนหรือการอ่านตำรามาก่อน ข้อมูลย้อนกลับ(feedback) สำหรับคำตอบของผู้เรียนว่าถูกหรือผิด อาจจะมีคำอธิบายหรือ

ไม่มีก็ได้ บทเรียนแบบนี้มักใช้สำหรับการเรียนรู้ระดับที่ใช้ความจำและความเข้าใจ

2.1.4.2 ประเภทสอนเนื้อหา(tutorial) บทเรียนประเภทนี้สามารถใช้ได้แทบทุกสาขาวิชา จะเสนอความรู้แก่ผู้เรียนในรูปการบรรยายและการตั้งคำถาม ข้อความที่เสนอนี้อาจเป็นความรู้ใหม่ ๆ หรือความรู้พื้นฐานที่จำเป็นต่อการเรียนรู้ในขั้นต่อไป

2.1.4.3 ประเภทสถานการณ์จำลอง(clinical simulation) เป็นบทเรียนจำลองสถานการณ์ที่เกิดขึ้นจริงหรืออาจเกิดขึ้นได้ นำมาเสนอแก่ผู้เรียนเพื่อให้เกิดการเรียนรู้หรือฝึกการตัดสินใจ บทเรียนแบบนี้ได้รับความนิยมมาก เพราะจะคล้ายคลึงกับการปฏิบัติจริง ทำให้ผู้เรียนเกิดความสนใจและรู้สึกท้าทาย ช่วยเพิ่มพูนความรู้และความสามารถในการแก้ปัญหาทางคลินิกแก่ผู้เรียน ช่วยให้ผู้เรียนได้นำความรู้พื้นฐานที่มีอยู่มาประยุกต์ วิเคราะห์และสังเคราะห์ ซึ่งเป็นระดับความรู้ที่สูงกว่าระดับความจำและความเข้าใจ แบ่งออกเป็น 4 รูปแบบใหญ่ ๆ ดังนี้

ก. การจำลองสถานการณ์ทางกายภาพ โดยมีการเสนอลักษณะทางกายภาพของวัตถุต่าง ๆ บนจอ ให้โอกาสผู้เรียนในการใช้หรือเรียนเกี่ยวกับสิ่งนั้น เช่น การใช้อุปกรณ์ในการทดลองวิทยาศาสตร์

ข. การจำลองสถานการณ์การปฏิบัติงาน จากการทำจำลองสถานการณ์ทางกายภาพ ผู้เรียนสามารถพัฒนาการมีส่วนร่วมให้มากกว่านั้นได้ เช่น การทดลองทางเคมี เป็นต้น

ค. การจำลองสถานการณ์ของเหตุการณ์ทั่วไป การจำลองสถานการณ์แบบนี้จะเน้นเกี่ยวกับทัศนคติและพฤติกรรมของคนเราในเหตุการณ์ต่าง ๆ มากกว่าการฝึกทักษะการปฏิบัติ เช่นสองประเภทแรก โดยผู้เรียนจะเป็นคนหนึ่งที่อยู่ในเหตุการณ์นั้น ๆ ด้วย

ง. การจำลองสถานการณ์กระบวนการ แบบนี้แตกต่างจากแบบอื่น ๆ หลายกรณี เช่น ผู้เรียนไม่ได้มีส่วนร่วมในสถานการณ์ หรือใช้เครื่องมือเช่นข้อ ก. และ ข. เพียงแต่ผู้เรียนเลือกค่าของพารามิเตอร์ต่าง ๆ ในตอนต้น และคอยสังเกต การเปลี่ยนแปลงที่เกิดขึ้นบนจอคอมพิวเตอร์

2.1.4.4 ประเภทเกมเพื่อการสอน (instructional games) เกมเป็นเครื่องมือสร้างแรงจูงใจให้กับผู้เรียนมาก แม้จุดมุ่งหมายใหญ่ของคอมพิวเตอร์ช่วยสอนประเภทนี้ จะเพื่อสื่อความหมายบทเรียนให้ผู้เรียนรู้อย่างมีประสิทธิภาพแล้ว เกมยัง

ให้อะไรหลาย ๆ อย่างแก่ผู้เรียน อาทิ เช่น

- การเรียนรู้หลักการและข้อเท็จจริง
- การพัฒนาทักษะ เช่น การแก้ปัญหา การตัดสินใจ หรือการเลือก

ยุทธวิธี

- ทักษะทางสังคมและการสื่อความหมาย
- การสร้างเจตคติ
- และทักษะอื่น ๆ อีกมาก เช่น การแข่งขัน การร่วมมือกัน บทบาท

ของโอกาส ความเป็นจริงของสังคม การลงโทษ การให้รางวัล เป็นต้น

การเขียนบทเรียนช่วยการเรียนการสอนแบบต่าง ๆ ขึ้นกับวัตถุประสงค์ของการนำไปใช้ เช่น ต้องการประเมินความก้าวหน้าในการเรียนรู้ ใช้บทเรียนประเภทฝึกทักษะ ต้องการให้ข้อมูลใหม่ ใช้บทเรียนประเภทสอนเนื้อหา ต้องการฝึกกระบวนการคิด ใช้บทเรียนสถานการณ์จำลอง หรือใช้ทั้ง 3 ประเภทรวมกัน เช่น สอนกรอนแนวคิดด้วยบทเรียนประเภทสอนเนื้อหา ให้ผู้เรียนประเมินตนเองโดยใช้บทเรียนประเภทฝึกทักษะ และทดสอบความสามารถในการประยุกต์ความรู้ วิเคราะห์และสังเคราะห์ด้วยบทเรียนประเภทสถานการณ์จำลอง

2.1.5 ประโยชน์ของการใช้ไมโครคอมพิวเตอร์ช่วยการเรียนการสอนมีดังนี้

2.1.5.1 การสอน

- สามารถสอนซ้ำกันหลายครั้ง ได้อย่างมีมาตรฐานเหมือนกันทุกครั้ง
- สามารถจัดลำดับขั้นตอนการโต้ตอบและแสดงผลได้ตามต้องการ
- มีความยืดหยุ่นในการปรับปรุงเปลี่ยนแปลงแก้ไขเพิ่มเติมเนื้อหา

วิชาได้สะดวก

2.1.5.2 การเรียน

- ผู้เรียนสามารถเลือกเวลาที่จะศึกษาได้ตามต้องการ
- ไม่เกิดความอายนเหมือนเมื่อเรียนกับอาจารย์โดยตรง ในชั้นเรียน
- ความเร็วในการรับรู้ปรับเข้ากับความสามารถของผู้เรียนได้ ซึ่งผู้

เรียนเรียนได้เรื่อย ๆ โดยไม่ต้องพะวงว่าจะตามเพื่อนไม่ทัน หรือต้องอายเพื่อน

2.2 แนวคิดเกี่ยวกับโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์

(authoring system)

คอมพิวเตอร์ช่วยการเรียนการสอนเข้ามามีบทบาทในสถาบันการศึกษาเพิ่มมากขึ้นทุกวัน การเขียนโปรแกรมเพื่อใช้งานคอมพิวเตอร์ช่วยการเรียนการสอนสามารถทำได้ 2 วิธี คือ

2.2.1 การเขียนบทเรียนโปรแกรมด้วยภาษาเครื่องหรือภาษาชั้นสูงภาษาใดภาษาหนึ่ง ซึ่งผู้เขียนบทเรียนจะต้องมีความรู้ในการเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างดี และเป็นผู้รู้เนื้อหาของบทเรียนเป็นอย่างดีด้วย จึงไม่ใช้งานที่ง่ายเพราะต้องอาศัยทักษะตลอดจนความรู้ความสามารถทางโปรแกรมคอมพิวเตอร์

2.2.2 การเขียนด้วยโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ โดยปกติ ผู้สอนจะเป็นผู้ที่รู้ถึงเนื้อหาวิชาและแนวทางในการถ่ายทอดเนื้อหาวิชาของตนได้ดีกว่าผู้อื่น ทั้งจากประสบการณ์การสอนและการฝึกฝนมาโดยตรง แต่ผู้สอนส่วนใหญ่ขาดทักษะการเขียนโปรแกรมคอมพิวเตอร์ ดังนั้นจึงเกิดความยุ่งยากในการใช้เครื่องคอมพิวเตอร์ช่วยสอน ทำให้ไม่สามารถพัฒนาบทเรียนคอมพิวเตอร์ช่วยสอนด้วยตนเองได้ จำเป็นต้องอาศัยโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ ซึ่งผู้เชี่ยวชาญทางด้านคอมพิวเตอร์จะทำหน้าที่พัฒนาขึ้นให้เหมาะสมกับบทเรียนโปรแกรม โดยโปรแกรมแบบนี้จะเสนอรูปแบบที่เหมาะสมต่อการเสนอกรอบของบทเรียนในขั้นตอนการสอนแต่ละขั้น และให้วิธีการที่ผู้สอนสามารถให้คำสั่งแก่คอมพิวเตอร์ในการนำเสนอ เนื้อหาบทเรียน รูปแบบในการตอบสนอง การยอมรับการตอบสนองหรือแม้แต่การซ่อมเสริม

โปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ที่ดี จะสามารถทำให้การเสนอเนื้อหาของบทเรียนมีคุณภาพมาก ไม่ว่าจะด้านข้อความ กราฟิกส์ ภาพเคลื่อนไหวหรือการใช้ประสมกับสื่ออื่น ๆ เช่น สไลด์ แถบบันทึกเสียง โทรทัศน์ วิทยุทัศน์ ฯลฯ รวมทั้งการออกแบบให้เป็นบทเรียนโปรแกรมเป็นแบบเส้นตรงหรือแบบแตกกิ่งก็ได้ อีกทั้งยังเปิดโอกาสให้ผู้เรียนมีส่วนร่วมในบทเรียนได้อย่างเต็มที่

2.3 งานพัฒนาที่เกี่ยวข้องกับโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ทั้งในและต่างประเทศ บางส่วนสามารถแสดงได้ดังตาราง 1 (นิสันต์ จงตระกูล และมนตรีรัตน์ จรุงเดชากุล, 2531:705)

ตาราง 1 ตัวอย่างโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์

ชื่อซอฟต์แวร์	ผู้พัฒนาซอฟต์แวร์	เครื่องที่ใช้
1. PC/PILOT	Washington Computer Services, USA	IBM
2. content write	Knowledge House, USA	Apple, Macintosh
3. microinstructor	C.V. Mosby Company, USA	IBM, Apple, Macintosh
4. formative evaluation programme (FEP)	จุฬาลงกรณ์มหาวิทยาลัย ประเทศไทย	IBM
5. System control for CAI (SCAI)	จุฬาลงกรณ์มหาวิทยาลัย ประเทศไทย	IBM
6. AIDS	Skill Corp Software, Inc.	Apple II
7. Block 82	California School for Deaf	Apple II
8. CAI WARE	Fireside Computing, Inc.	Radioshacks TRS 80
9. GHOST Writer	CAVRI Interactive Video	Apple II
10. Interactive Authoring System	McGraw-Hill Book Co.	IBM PC
11. PASS	Bell & Howell Visual Communicating Group P.	Apple II
12. The instructor	BCD Associates	Apple II
13. VITAL	Canadian Inc.	IBM PC

2.3.1 โปรแกรมคอมพิวเตอร์เพื่อการเรียนรู้แบบบูรณาการ "ทองจันทร์"

(พิสนธ์ จงตระกูลและคณะ, 2533:258-259)

พัฒนาโดย นพ. พิสนธ์ จงตระกูลและคณะ เป็น software ที่เขียนด้วยภาษาเบสิก บนเครื่อง IBM compatible และ/หรือ Sharp MZ 5600 เป็นโปรแกรมที่ประกอบด้วยโปรแกรมย่อย 6 ส่วน ทำงานแยกกันดังนี้

2.3.1.1 โปรแกรมย่อยที่ 1 เป็นโปรแกรมที่ใช้สร้างบทเรียนแบบความเรียง(text) ผู้เรียนสามารถเลือกศึกษาได้ตามหัวข้อ แต่ละหัวข้อมีการบรรยายเนื้อหาพร้อมทั้งรายละเอียด

2.3.1.2 โปรแกรมย่อยที่ 2 เป็นโปรแกรมที่ใช้สร้างบทเรียนแบบความเรียง ประกอบภาพสไลด์โดยพ่วงเครื่องคอมพิวเตอร์เข้ากับเครื่องฉายสไลด์ผ่านอุปกรณ์ที่คณะผู้วิจัยประดิษฐ์ขึ้นเอง

2.3.1.3 โปรแกรมย่อยที่ 3 เป็นโปรแกรมที่ใช้สร้างบทเรียนเพื่อประเมินความก้าวหน้าในการเรียนรู้ โดยใช้รูปแบบของข้อสอบแบบเลือกตอบ และชนิดมีคำตอบที่ถูกเพียงคำตอบเดียว บทเรียนจะให้คำอธิบายเพิ่มเติมในกรณีที่ผู้เรียนตอบถูก และอธิบายเหตุผลของคำตอบที่ผิด มีการให้คะแนน

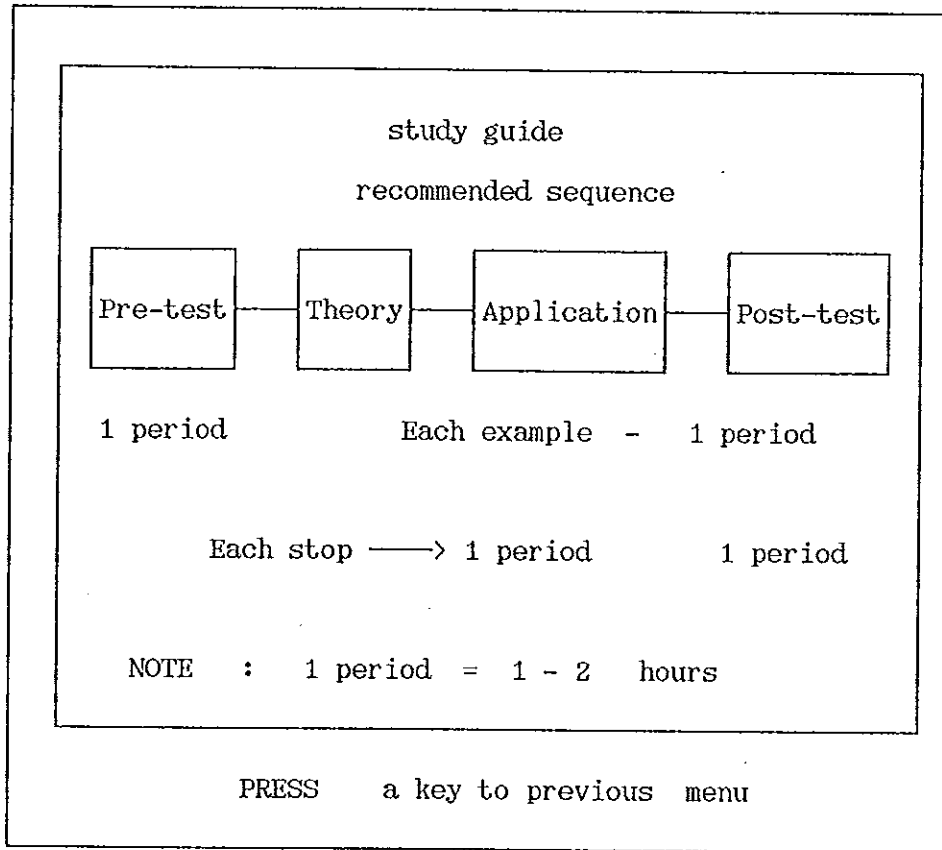
2.3.1.4 โปรแกรมย่อยที่ 4 เป็นโปรแกรมที่ใช้สร้างบทเรียนเพื่อประเมินความก้าวหน้าในการเรียนรู้ โดยใช้รูปแบบของข้อสอบแบบเลือกตอบ ชนิดมีคำตอบเดียวร่วมกับเครื่องฉายสไลด์

2.3.1.5 โปรแกรมย่อยที่ 5 เป็นโปรแกรมที่ใช้สร้างบทเรียนเพื่อประเมินความก้าวหน้าในการเรียนรู้ โดยใช้รูปแบบของข้อสอบอัตนัยชนิด Modified Essay Question ซึ่งนำเสนอด้วยสถานการณ์ และถามผู้เรียนเป็นระยะในลักษณะของสัมมุติฐาน การแปลผลเพื่อแก้ปัญหาของผู้ป่วยในเวลาที่กำหนด

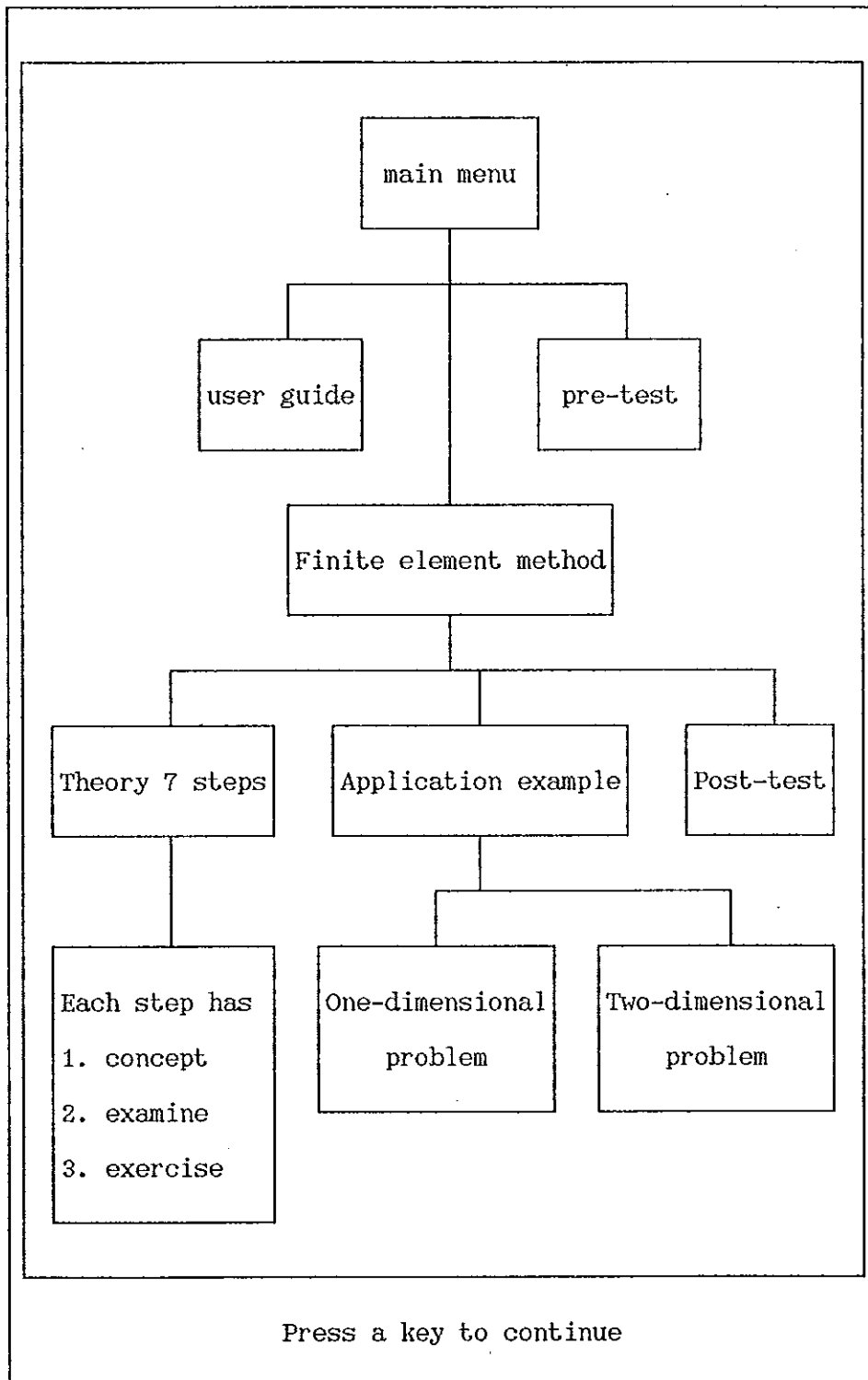
2.3.1.6 โปรแกรมย่อยที่ 6 เป็นโปรแกรมที่ใช้สร้างบทเรียนเพื่อฝึกการเลือกใช้อาย่างเหมาะสม โดยใช้สถานการณ์จำลอง

2.3.2 ไมโครคอมพิวเตอร์ช่วยสอนของภาควิชาวิศวกรรมโยธา มหาวิทยาลัยขอนแก่น ในเรื่องไฟไนต์อิลิเมนต์ที่ใช้กับไมโครคอมพิวเตอร์ IBM PC หรือ XT ที่มีหน่วยความจำอย่างน้อย 256 กิโลไบต์ และมี Hercules graphic card ในการแสดงกราฟิกส์ ดังแสดงในภาพประกอบ 1 และภาพประกอบ 2 (จุมพล มณีชัย, 2530:39)

ภาพประกอบ 1 โครงสร้างของโปรแกรม เรื่อง ไฟไนต์อิลิเมนต์



ภาพประกอบ 2 แบบแนะนำลำดับและเวลาที่ใช้ในการศึกษาเรื่องไฟไนต์เอลิเมนต์



3 วัตถุประสงค์

งานวิจัยนี้มีวัตถุประสงค์เพื่อ

3.1 ออกแบบและพัฒนา TEACH ซึ่งเป็นระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ เพื่อให้ผู้สอนใช้สร้างบทเรียนบนไมโครคอมพิวเตอร์สอนวิชาใด ๆ ที่ต้องการด้วยตนเอง โดยผู้สอนที่ใช้ระบบนี้ไม่จำเป็นต้องมีความสามารถในการเขียนโปรแกรม เพียงแต่จัดทำบทเรียนตามรูปแบบคำสั่งที่กำหนดให้ แล้วให้ผู้เรียนนำบทเรียนไปใช้ศึกษา โดยระบบ TEACH จะทำหน้าที่สอนแทนผู้สอน ทำหน้าที่ตรวจสอบตำแหน่งการเรียนของผู้เรียน แล้วจัดให้ผู้เรียนเข้าสู่ระบบการเรียนในตำแหน่งที่ต่อเนื่อง ขณะสอนจะนำเสนอเนื้อหาบทเรียนตามความสามารถของผู้เรียน ประเมินผลการเรียนของผู้เรียนตามเกณฑ์การประเมินที่ผู้สอนกำหนดไว้ในบทเรียน สามารถทดสอบผู้เรียนด้วยแบบทดสอบระบบสุ่มจากกรอบบทเรียนที่ผู้สอนกำหนด ในขณะที่เรียนจะมีข้อความโต้ตอบให้แรงจูงใจแก่ผู้เรียน ซึ่งสามารถเรียกมาจากแฟ้มข้อมูลโต้ตอบได้ และเมื่อยุติการเรียนจะแสดงผลการเรียนให้ผู้เรียนทราบ

ระบบ TEACH ที่ออกแบบและพัฒนาขึ้นจะประกอบด้วยส่วนสำคัญ 3 ส่วน คือ

3.1.1 โปรแกรมควบคุมระบบ ซึ่งประกอบด้วย 2 โปรแกรมหลัก คือ

TEACH.EXE และ ANALYSER.EXE

TEACH.EXE เป็นโปรแกรมที่ทำหน้าที่นำเสนอบทเรียน โปรแกรมที่ประกอบด้วยคำสั่งภาษา TAL (TEACH Authoring Language) โดยการสร้างตารางควบคุม ตรวจสอบให้ผู้เรียนเข้าสู่ระบบการเรียนในตำแหน่งที่ต่อเนื่องจากการเรียนที่ผ่านมา บันทึกสรุปผลการเรียน และบันทึกพฤติกรรมการเรียนของผู้เรียน

ANALYSER.EXE เป็นโปรแกรมที่ทำหน้าที่เรียกดูบันทึกสรุปผลการเรียน และบันทึกพฤติกรรมการเรียนของผู้เรียน

3.1.2 ภาษา TAL ซึ่งเป็นภาษาสำหรับใช้ในการเขียนบทเรียน โปรแกรมของระบบ TEACH โดยเฉพาะ ซึ่งจะช่วยจัดลำดับการนำเสนอบทเรียนและใส่เทคนิคการสอนในเนื้อหาบทเรียน

3.1.3 บทเรียน เป็นส่วนที่ผู้สอนเลือกใช้โปรแกรมประมวลคำ (word processor) ที่มีเผยแพร่ทั่วไปสร้างขึ้น เพื่อนำเสนอเนื้อหาของบทเรียนตามที่ต้องการ โดยบทเรียนโปรแกรมนี้ใช้คำสั่งตามรูปแบบภาษา TAL

3.2 จัดทำเอกสารคู่มือการใช้ TEACH เพื่อเป็นเอกสารประกอบการใช้งานระบบ TEACH

3.8 สร้างตัวอย่างบทเรียนโปรแกรมเรื่อง "การใช้ TEACH:ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์" ซึ่งเป็นบทเรียนโปรแกรมแนะนำการใช้ระบบ TEACH ที่ทำงานบนระบบ TEACH

4. ประโยชน์ที่คาดว่าจะได้รับการวิจัย

4.1 ได้ระบบโปรแกรมสร้างบทเรียน TEACH ที่สะดวกสำหรับผู้สอนนำไปใช้สร้างบทเรียนบนไมโครคอมพิวเตอร์ด้วยตนเอง

4.2 ผู้เรียนสามารถศึกษาบทเรียนบนไมโครคอมพิวเตอร์ด้วยตนเองภายใต้การควบคุมของโปรแกรมสร้างบทเรียน TEACH เสมือนมีผู้สอนคอยดูแลแบบหนึ่งต่อหนึ่ง ซึ่งเป็นการลดภาระงานสอนของผู้สอน

4.3 การเรียนของผู้เรียนทุกขั้นตอนจะถูกระบบ TEACH บันทึกไว้ ซึ่งจะเป็นประโยชน์ต่อผู้สอนในการตรวจสอบระดับความสามารถของผู้เรียน และใช้ในการประเมินผลบทเรียนของผู้สอนเอง

4.4 เป็นแนวทางสำหรับผู้สนใจในการพัฒนาระบบช่วยสร้างบทเรียน

4.5 เป็นส่วนหนึ่งที่จะสนับสนุนและผลักดันให้มีการใช้คอมพิวเตอร์ช่วยสอนในวงการศึกษาของประเทศมากขึ้น

บทที่ 2

วิธีการวิจัย

ในบทนี้กล่าวถึง ขั้นตอนและรายละเอียดของการทำวิจัยซึ่งเป็นงานวิจัยเพื่อออกแบบระบบและพัฒนาโปรแกรม โดยมีเป้าหมายเพื่อให้ได้โปรแกรมที่สามารถนำไปใช้งาน หัวข้อที่จะกล่าวถึงได้แบ่งออกดังนี้

1. วัสดุและเครื่องมือ
2. อุปกรณ์ที่ใช้ในการวิจัย
3. สถานที่ทำการวิจัย
4. วิธีดำเนินการวิจัย

1. วัสดุและเครื่องมือ

- 1.1 ผังโปรแกรมโครงสร้าง(structured flowchart)
- 1.2 ภาษาที่ใช้ในการพัฒนาโปรแกรมคือ Turbo C++ Version 1.0
- 1.3 โปรแกรมประมวลคำราชวิถี version 2.3

2. อุปกรณ์ที่ใช้ในการวิจัย

- 2.1 ไมโครคอมพิวเตอร์ 16 บิต ขนาดหน่วยความจำไม่น้อยกว่า 640 KB
จอภาพโมโนโครม
- 2.2 เครื่องพิมพ์รุ่น LX 800

3. สถานที่ทำการวิจัย

- 3.1 ศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์
- 3.2 ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์
- 3.3 สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี

4. วิธีดำเนินการวิจัย ได้ทำเป็นขั้นตอนดังนี้

- 4.1 ศึกษาระบบโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ที่มีใช้ใน
ประเทศและต่างประเทศ และวิเคราะห์สภาพการเรียนการสอน
- 4.2 ศึกษาเพื่อคัดเลือกซอฟต์แวร์ที่จะใช้เป็นเครื่องมือพัฒนาโปรแกรม
และสร้างแฟ้มข้อมูลต่าง ๆ
- 4.3 ออกแบบลักษณะบทเรียนโปรแกรม (courseware)
- 4.4 ออกแบบภาษา TAL
- 4.5 ออกแบบและพัฒนาระบบ TEACH
- 4.6 สร้างตัวอย่างบทเรียนโปรแกรมเพื่อสอนการใช้ระบบ TEACH
- 4.7 ประเมินผลการใช้งานโปรแกรม

4.1 ศึกษาระบบโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ที่มีใช้ใน
ประเทศและต่างประเทศ และวิเคราะห์สภาพการเรียนการสอน ข้อมูลที่ได้จากการศึกษาจะนำไป
ใช้ในการกำหนดวัตถุประสงค์สำหรับงานออกแบบลักษณะบทเรียนโปรแกรม งานออกแบบ
ภาษา TAL และงานออกแบบและพัฒนาระบบ TEACH การศึกษาส่วนนี้ได้ทำการศึกษาโปร-
แกรม เอกสารและหนังสือที่เกี่ยวข้อง ซึ่งได้กล่าวไว้ในบทที่ 1 หัวข้อ 1.2 เรื่องการ
ตรวจเอกสาร ในส่วนที่เพิ่มเติมคือ การวิเคราะห์สภาพการเรียนการสอนจริงและความ
ต้องการของผู้สอน ซึ่งสามารถวิเคราะห์ได้ดังนี้

รูปแบบของหลักสูตรวิชาต่าง ๆ ที่ใช้ในการเรียนการสอน ได้จัดแบ่งเป็นราย
วิชา บทหรือหัวข้อ จุดประสงค์การเรียนรู้หรือหัวข้อย่อยในบท โดยมีครูเป็นผู้จัดหาเนื้อหา
วิชา วางแผนการสอน ทำการถ่ายทอดความรู้ให้ผู้เรียนเกิดการเรียนรู้โดยใช้เทคนิคการ
สอนต่าง ๆ

นักเรียนหรือผู้เรียนในแต่ละชั้นปีจะเรียนหนึ่งหรือหลายวิชา แต่ละวิชามีหนึ่ง
หรือหลายบทเรียน แต่ละบทเรียนมีหนึ่งหรือหลายจุดประสงค์การเรียนรู้ แต่ละจุดประสงค์
การเรียนรู้สามารถจะอธิบายได้ด้วยหนึ่งชุดข้อความหรือหลายชุดข้อความ

การมอบหมายงานเพื่อสร้างเสริมการเรียนรู้ จะเป็นไปในลักษณะให้ผู้เรียน
ไปทำการศึกษาวินิจฉัย บทที่เท่าไร จุดประสงค์ใด ตัวอย่างเช่น ให้ผู้เรียนไปศึกษาวินิจฉัย
คณิตศาสตร์ บทที่ 1 จุดประสงค์ที่ 2 เป็นต้น

โดยผู้สอนจะสนใจผลการเรียนของผู้เรียน พฤติกรรมการเรียนเพื่อเป็นประโยชน์ต่อการให้ความรู้ ให้คำแนะนำ ประเมินความรู้ที่ผู้เรียนได้รับ ปรับปรุงบทเรียนและเทคนิคการสอน

4.2 ศึกษาเพื่อคัดเลือกซอฟต์แวร์ที่จะใช้เป็นเครื่องมือพัฒนาโปรแกรมและสร้างแฟ้มข้อมูลต่าง ๆ ได้ทำการศึกษาจากเอกสาร และทดสอบการใช้งานซอฟต์แวร์ และได้คัดเลือกซอฟต์แวร์ดังนี้

4.2.1 ซอฟต์แวร์ที่จะใช้เป็นเครื่องมือพัฒนาโปรแกรมได้เลือกใช้ Turbo C⁺⁺ Version 1.0 เนื่องจากมีคู่มือและโปรแกรมใช้งานแพร่หลายทั่วไป สามารถติดตั้งใช้งานกับเครื่องไมโครคอมพิวเตอร์ได้สะดวก ตลอดจนภาษา C เป็นภาษาที่รวมข้อดีเด่นซึ่งเหมาะกับงานพัฒนาดังนี้

- เป็นภาษามาตรฐานที่ไม่ขึ้นกับระบบปฏิบัติการหรือขึ้นกับฮาร์ดแวร์ เขียนแทนภาษาแอสเซมบลีได้ดี ค้นหาที่ผิดหรือแก้ไขตัวโปรแกรมได้ง่าย จึงเป็นภาษาระดับสูงที่ใช้งานได้เหมือนภาษาระดับต่ำ

- อาศัยหลักการที่เรียกว่า โปรแกรมโครงสร้าง มีโครงสร้างที่ตึงในแง่โครงสร้างภาษา และโครงสร้างของชนิดข้อมูล มีความสะดวกในการเขียน และถ้าใช้ตัวแปรชุดคำสั่ง (compiler) ที่มีคุณภาพสูง จะให้ภาษาเป้าหมาย (object code) สั้น ทำงานได้รวดเร็ว มีความคล่องตัวที่จะประยุกต์เข้ากับงานต่าง ๆ ได้เป็นอย่างดี เหมาะกับงานพัฒนาที่ต้องการความรวดเร็วเป็นสำคัญ

- เป็นภาษาที่มีอยู่เกือบทุกระบบปฏิบัติการบนเครื่องคอมพิวเตอร์ ตั้งแต่ไมโครคอมพิวเตอร์ ถึงเครื่องซูเปอร์คอมพิวเตอร์ เช่น Cray I

4.2.2 ซอฟต์แวร์ที่จะใช้เป็นเครื่องมือสร้างแฟ้มข้อมูลต่าง ๆ ได้ กำหนดให้สามารถใช้โปรแกรมประมวลคำที่มีทั่วไป เพื่อให้การพัฒนาบทเรียนสามารถ ทำได้แพร่หลายและกว้างขวาง

4.3 ออกแบบลักษณะบทเรียนโปรแกรม

จากการวิเคราะห์สภาพการเรียนการสอน จะเห็นว่ารายวิชาในแต่ละระดับ ได้จัดแบ่งออกเป็นบทหรือหัวข้อ จุดประสงค์การเรียนรู้หรือหัวข้อย่อยในบท แต่ละจุดประสงค์การเรียนรู้สามารถจะอธิบายได้ด้วยหนึ่งชุดข้อความหรือหลายชุดข้อความ ดังนั้น

ลักษณะบทเรียนโปรแกรมสำหรับ TEACH สามารถออกแบบให้ประกอบด้วยแฟ้มข้อมูล 2 กลุ่มที่มีความสัมพันธ์เชื่อมโยงกัน โดยชื่อแฟ้มและรูปแบบของข้อมูลในแฟ้ม คือ แฟ้มโครงสร้างบทเรียน และแฟ้มเนื้อหาวิชา ดังนี้

4.3.1 แฟ้มโครงสร้างบทเรียน เป็นแฟ้มสำหรับเก็บข้อมูลชื่อแฟ้มหัวข้อ และชื่อหัวข้อหรือชื่อวิชา เพื่อเป็นข้อมูลโยงความสัมพันธ์ไปยังแฟ้มเนื้อหาวิชาต่าง ๆ

การตั้งชื่อแฟ้มโครงสร้าง มีข้อกำหนดว่า ต้องประกอบด้วยอักขระ 3 ตัว และมีส่วนขยายของชื่อแฟ้มเป็น .STR

รูปแบบข้อมูลในแฟ้มโครงสร้าง มีข้อกำหนดว่าจะต้องประกอบด้วย 2 ส่วน ที่แยกจากกันด้วยช่องว่าง 1 ช่อง มีรูปแบบดังนี้

ชื่อแฟ้มหัวข้อ ชื่อหัวข้อ (โดยชื่อแฟ้มที่มีรหัส 000 จะเป็นชื่อวิชา)

ชื่อแฟ้มหัวข้อคือ ชื่อแฟ้มบทเรียนของหัวข้อนั้น ประกอบด้วยอักขระ 3 ตัวแรกของชื่อแฟ้มโครงสร้าง ตามด้วยตัวเลข 3 ตัว ซึ่งเป็นลำดับหัวข้อ โดยไม่ต้องมีส่วนขยายชื่อแฟ้ม และกำหนดเป็นการเฉพาะให้ชื่อแฟ้มหัวข้อที่มีรหัสตัวเลข 000 เป็นชื่อวิชา ดังตัวอย่างในภาคผนวก ง และภาคผนวก จ

4.3.2 แฟ้มเนื้อหาวิชา ลักษณะของข้อมูลในแฟ้มเนื้อหาวิชา คือ เนื้อหาของบทเรียนที่ภาษา TAL เป็นส่วนประกอบ โดยโครงสร้างของภาษา TAL จะมีลักษณะแบ่งเนื้อหาเป็นกรอบการเรียนหลาย ๆ กรอบการเรียน และช่วยในการนำเสนอบทเรียนให้ เป็นไปตามที่ผู้สอนต้องการ การตั้งชื่อแฟ้มจะใช้ชื่อเดียวกับแฟ้มหัวข้อที่อยู่ในแฟ้มโครงสร้าง ให้มีส่วนขยายของชื่อแฟ้มเป็น .TXT ดังนั้นในบทเรียนหนึ่งจะมีแฟ้มเนื้อหาวิชาได้หลายแฟ้ม ซึ่งจะสัมพันธ์กับจำนวนที่ปรากฏในแฟ้มโครงสร้างบทเรียน ดังตัวอย่างในภาคผนวก จ

การจัดสร้างแฟ้มทั้ง 2 กลุ่มกระทำโดยใช้โปรแกรมประมวลคำทั่วไป

4.4 ออกแบบภาษา TAL (TEACH Authoring Language)

ภาษา TAL เป็นคำสั่งสำหรับใช้กับ TEACH ซึ่งเป็นระบบคอมพิวเตอร์ช่วยสอนแบบอัตโนมัติ ภาษานี้จึงควรมีคุณสมบัติช่วยในการออกแบบและสร้าง บทเรียน โปรแกรม มีรูปแบบในลักษณะคำอธิบายเพราะเป็นข้อมูลไม่ใช่โปรแกรมทางคอมพิวเตอร์ ง่ายต่อการใช้เขียนบทเรียนหรือทำการปรับปรุงบทเรียน มีคำสั่งน้อย และมีรูปแบบเป็นลำดับขั้นตอนอย่างชัดเจน เนื้อหาเกี่ยวข้องการประเมินผลบทเรียนและประเมินผลการเรียนของผู้เรียน รับการโต้ตอบของผู้เรียน และบันทึกข้อมูลโดยอัตโนมัติ ดังนั้น จึงสามารถออกแบบภาษา

TAL ได้ดังนี้

- 4.4.1 เซตของอักขระในภาษา TAL ประกอบด้วย
 - 4.4.1.1 ตัวอักษร ซึ่งประกอบด้วย
 - ตัวอักษรภาษาอังกฤษทั้งหมด
 - ตัวอักษรภาษาไทยและสระทั้งหมด
 - 4.4.1.2 ตัวเลข มีทั้งหมด 10 ตัว ได้แก่ 0..9
 - 4.4.1.3 อักขระพิเศษทั้งหมด
- 4.4.2 แบบฟอร์มลงรหัส
 - การเขียนบทเรียนโปรแกรมด้วยภาษา TAL สามารถเขียนได้ดังนี้
 - 4.4.2.1 เริ่มเขียนคำสั่งที่คอลัมน์ที่ 1
 - 4.4.2.2 คำสั่งหรือข้อความในแต่ละบรรทัดจะมีความยาวไม่เกิน 100 ตัวอักษร
 - 4.4.2.3 บทเรียนโปรแกรมหนึ่งโปรแกรม จะแบ่งแยกเนื้อหาในบทเรียนเป็นส่วนย่อย ๆ เรียกว่า กรอบการเรียน (frame) โดยในแต่ละบทเรียนมีความยาวได้ไม่เกิน 500 บรรทัด หรือ 50 กรอบ
 - 4.4.2.4 กรอบการเรียนแต่ละกรอบจะต้องมีโครงสร้างตามข้อ 4.4.4
 - 4.4.2.5 กรอบการเรียนแต่ละกรอบ ควรกำหนดการจัดเรียงให้เป็นไปตามลำดับที่ต้องการควบคุม เพื่อสะดวกต่อการนำเสนอบทเรียนและบันทึกผลการเรียนของผู้เรียน
 - 4.4.3 คำสงวน
 - คำและเครื่องหมายต่อไปนี้ ถ้าไม่ใช่เป็นคำสั่ง สงวนไม่ให้พิมพ์ในคอลัมน์ที่ 1

```
FRAME SCREEN INPUT BRANCH EVALUATE $T @
```

```
, " ^
```

- 4.4.4 โครงสร้างของกรอบการเรียน มีลักษณะดังภาพประกอบ 3

ภาพประกอบ 3 โครงสร้างกรอบการเรียน

```

FRAME #no. [,frame_type] [,no_loop]
SCREEN ['label'] [,wait_time]
[string]
[["]string]
[']string]
[^]string]
[@([line] [,col])] ['string']
$C
$T#no
.
INPUT [@([line][,col])][message'][,wait_time]
BRANCH [full_score][,wait_time]
['value' [condition]] [:frame_no][,string'/message]
.
.

```

หมายเหตุ คำสั่งหรือข้อความภายในเครื่องหมายวงเล็บใหญ่ [] หมายถึงจะมีหรือไม่มีก็ได้

กรอบการเรียนแต่ละกรอบที่เป็นชนิดบรรยายและชนิดคำถาม จะเริ่มต้นด้วยคำสั่ง FRAME และตามด้วยคำสั่ง SCREEN บรรทัดต่อจาก SCREEN จะเป็นข้อความแสดงผลบนจอภาพ คำสั่งต่อจากนี้อาจเป็นคำสั่ง INPUT หรือ BRANCH หรือไม่มีทั้งคู่ก็ได้ ถ้ามีคำสั่ง BRANCH จะต้องมีคำสั่งย่อยของ BRANCH ตามอย่างน้อย 1 บรรทัด ต่อจากนั้นจะเป็นกรอบการเรียนถัดไป

แต่ถ้ากรอบการเรียนนั้นเป็นชนิดลุ่ม บรรทัดต่อจากคำสั่ง FRAME จะต้องเป็นชุดหมายเลขกรอบที่จะถูกลุ่มขึ้นใช้งาน มีรูปแบบดังนี้

```

FRAME #no.,R
([nos-noe][,no])

```

จะเห็นได้ว่า ภาษา TAL ประกอบด้วยคำสั่งหลัก 4 คำสั่ง คือ

FRAME สำหรับบอกหมายเลขกรอบ ชั้นดีของกรอบ จำนวนครั้งที่
ยอมให้เรียกใช้

SCREEN สำหรับกำหนดการเริ่มต้นแสดงข้อมูลออกทางจอภาพ

INPUT สำหรับการรอรับข้อมูลจากผู้ใช้บทเรียน

BRANCH สำหรับการแตกแขนงไปแสดงกรอบอื่น ๆ

และคำสั่งพิเศษอีก 1 คำสั่ง คือ

EVALUATE สำหรับใช้กำหนดเกณฑ์การประเมินผลการเรียน

4.4.5 รูปแบบและความหมายของคำสั่งภาษา TAL ดูได้ในภาคผนวก ข

4.5 ออกแบบระบบและพัฒนาโปรแกรม TEACH

ในส่วนการออกแบบระบบและพัฒนาโปรแกรมได้ใช้ผังโปรแกรมโครงสร้างในการดำเนินการ แบ่งเป็นขั้นตอนดังนี้

4.5.1 กำหนดวัตถุประสงค์ของระบบ

4.5.2 ออกแบบขั้นตอนการทำงานของโปรแกรม

4.5.3 ลงรหัสโปรแกรม

4.5.4 ทดสอบและแก้ไขโปรแกรม

4.5.5 จัดทำเอกสารประกอบโปรแกรม

4.5.1 กำหนดวัตถุประสงค์ของระบบ จะเป็นการพิจารณากำหนดปัญหาหรือความต้องการ ส่วนข้อมูลนำเข้า การประมวลผล และส่วนข้อมูลนำออก ซึ่งเป็นข้อมูลสำคัญในการออกแบบระบบและพัฒนาโปรแกรมเพื่อให้ได้โปรแกรมที่ถูกต้องตรงกับความต้องการ การพิจารณาแบ่งเป็น

4.5.1.1 การพิจารณากำหนดวัตถุประสงค์

4.5.1.2 การพิจารณากำหนดส่วนข้อมูลนำเข้า

4.5.1.3 การพิจารณากำหนดการประมวลผล

4.5.1.4 การพิจารณากำหนดส่วนข้อมูลนำออก

4.5.1.1 การพิจารณากำหนดวัตถุประสงค์เป็นการพิจารณาปัญหาหรือความต้องการให้ได้ความชัดเจน จากการศึกษาและวิเคราะห์ระบบโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์ที่มีใช้ในประเทศและต่างประเทศ และจากความต้องการของนักการศึกษา ทำให้สามารถกำหนดวัตถุประสงค์ได้ดังนี้

ต้องการพัฒนาระบบโปรแกรมช่วยการเรียนการสอนชนิดโปรแกรมสร้างบทเรียนด้วยตนเองบนคอมพิวเตอร์แบบกรอบ (frame type) ชื่อว่า TEACH ระบบนี้มีหน้าที่หลัก คือ ช่วยสอนบทเรียนโดยการนำเสนอเนื้อหา (บทเรียน) ตามรูปแบบ (เทคนิค การสอน) ซึ่งได้มีผู้สอน (เช่น นักการศึกษา นักวิชาการ ฯลฯ) กำหนดไว้ในลักษณะแฟ้มข้อมูล (แฟ้มบทเรียนสำเร็จรูป) ซึ่งเป็นบทเรียนที่รวมเทคนิคการสอนและการประเมินผลไว้ด้วยกันที่มีส่วนประกอบเป็นภาษา TAL เป็นตัวควบคุม ซึ่งความสามารถมีดังนี้

- ก. แฟ้มเนื้อหาวิชาแยกจากระบบควบคุม
- ข. นำเสนอบทเรียนแบบแตกสาขาได้ (branching)
- ค. นำเสนอข้อทดสอบแบบสุ่มเลือกจากที่กำหนดได้
- ง. กำหนดเกณฑ์การประเมินผลการเรียนได้
- จ. แสดงตำแหน่งการเรียนของผู้เรียน พร้อมผลการเรียนที่ผ่านมา
- ฉ. ผู้เรียนสามารถเลือกเข้าสู่บทเรียนในส่วนที่ต้องการ หรือให้ระบบจัดให้เข้าสู่บทเรียน ในตำแหน่งที่ต่อเนื่องจากการเรียนครั้งที่ผ่านมา
- ช. แสดงผลการเรียนให้ผู้เรียนทราบเมื่อจบการเรียน
- ซ. สามารถโต้ตอบกับผู้เรียน โดยใช้ข้อความจากแฟ้มการโต้ตอบ หรือใช้ข้อความที่กำหนดไว้ในแฟ้มบทเรียน
- ฅ. ประเมินผลผู้เรียน
- ญ. บันทึกข้อมูลการเรียนของผู้เรียน

4.5.1.2 การพิจารณากำหนดส่วนข้อมูลนำเข้า จากวัตถุประสงค์ที่กำหนดและลักษณะ โปรแกรมที่ต้องการ กำหนดให้ส่วนข้อมูลนำเข้าประกอบด้วยแฟ้มข้อมูล 4 แฟ้ม คือ แฟ้มข้อมูลโครงสร้างบทเรียน แฟ้มข้อมูลเนื้อหาวิชา (courseware) แฟ้มข้อมูลโต้ตอบ และแฟ้มข้อมูลสรุปผลการเรียนของผู้เรียน กำหนดให้แต่ละแฟ้มข้อมูลมีรายละเอียด ดังนี้

แฟ้มข้อมูล โครงสร้างบทเรียน

1) ชื่อ SSS.STR

เมื่อ SSS คือ รหัสวิชา

STR ย่อมาจาก STRUCTURE

2) จุดประสงค์ เพื่อเก็บข้อมูลสำหรับการกำหนดชื่อแฟ้มหัวข้อ และชื่อหัวข้อ ของแต่ละชุดการเรียน

3) การนำไปใช้ แฟ้มโครงสร้างบทเรียน SSS.STR ใช้เชื่อมกับแฟ้มข้อมูลสรุปผลการเรียนของผู้เรียน SSSIII.STS บนตารางข้อมูลประวัติการเรียน เพื่อรายงานสถานภาพการเรียนทั้งหมด พร้อมทั้งให้เลือกตำแหน่งที่จะเข้าเรียน

4) โครงสร้างแฟ้ม แฟ้มนี้เป็นชนิดแฟ้มข้อความโดยมีข้อมูลแต่ละบรรทัด ประกอบด้วย 2 ชุดข้อมูล ดังนี้

(1) ชื่อแฟ้มหัวข้อบทเรียน SSSLLL

เมื่อ SSS = รหัสวิชา

LLL = เลขที่หัวข้อ

กำหนดให้ SSS000 หมายถึง ชื่อวิชา

(2) ชื่อหัวข้อ ชื่อวิชา หรือชื่อหัวข้อ หรือชื่อจุดประสงค์

ตัวอย่าง

SMP000 บทเรียนทดสอบระบบผู้สอน

SMP001 คำแนะนำ

SMP002 จุดประสงค์ของระบบ

แฟ้มโครงสร้างหนึ่งแฟ้มกำหนดให้มีหัวข้อได้ไม่เกิน 9 หัวข้อ

แฟ้มข้อมูล เนื้อหาวิชา

1) ชื่อ SSSLLL.TXT

เมื่อ SSS คือ รหัสวิชา

LLL คือ ลำดับบทที่/ลำดับจุดประสงค์ที่

2) จุดประสงค์ เป็นแฟ้มข้อมูลให้ผู้เรียนใช้ศึกษา แฟ้มเนื้อหาวิชาหนึ่งแฟ้มสามารถกำหนดเนื้อหาเพื่อการเรียนรู้หนึ่งจุดประสงค์หรือมากกว่าก็ได้ ฉะนั้นบทเรียนหนึ่งๆ อาจประกอบด้วยแฟ้มเนื้อหาวิชาหลายแฟ้ม จำนวนแฟ้มเนื้อหาวิชาในแต่ละบทเรียนจะ

สัมพันธ์กับข้อมูลในแฟ้มโครงสร้างบทเรียน

- 3) การนำไปใช้ เป็นแฟ้มข้อมูลให้ผู้เรียนศึกษา
- 4) โครงสร้างแฟ้ม ประกอบขึ้นจากหนึ่งกรอบการเรียนหรือหลายกรอบการเรียน เป็นข้อความที่มีคำสั่งภาษา TAL ประกอบในลักษณะกรอบ(frame) 1 แฟ้มเนื้อหาบทบรรจุข้อมูลไม่เกิน 50 กรอบการเรียนหรือไม่เกิน 500 บรรทัด บรรทัดละ 100 ตัวอักษร

แฟ้มข้อมูลข้อความโต้ตอบ

- 1) ชื่อ CAICOM.MSG
- 2) จุดประสงค์ เป็นแฟ้มข้อมูลให้ผู้สอนสร้างขึ้นเพื่อเลือกข้อความโต้ตอบไปใช้งาน
- 3) การนำไปใช้ ผู้สอนสร้างแฟ้มข้อมูลข้อความโต้ตอบ โดยใช้โปรแกรมประมวล-
คำ และเรียกใช้ข้อความโต้ตอบโดยใช้คำสั่งในภาษา TAL
- 4) โครงสร้างแฟ้ม ประกอบด้วยรายการข้อมูลดังนี้
 - (1) หมายเลขรหัสเรียก
 - (2) ข้อความโต้ตอบ มีได้ไม่เกิน 50 ข้อความ ข้อความละ 100 อักขระ

แฟ้มข้อมูลสรุปผลการเรียนของผู้เรียน

- 1) ชื่อ SSSIII.STS
 - เมื่อ SSS คือ รหัสวิชา
 - III คือ รหัสผู้เรียน
 - STS คือ ส่วนขยายแฟ้ม ย่อมาจาก status
- 2) จุดประสงค์ เพื่อรวบรวมสรุปผลการเรียนของแต่ละแฟ้มเนื้อหาวิชาที่เป็นวิชาเดียวกันของผู้เรียนคนเดียวกันไว้ในแฟ้มข้อมูลนี้ โดยกำหนดให้สรุปผลการเรียนของผู้เรียนในหนึ่งแฟ้มเนื้อหาวิชา เก็บไว้เป็น 1 ระเบียบ
- 3) การนำไปใช้
 - การบันทึกข้อมูล เมื่อจบการเรียนในแต่ละแฟ้มเนื้อหาวิชาจะบันทึกสรุปผลการเรียนของผู้เรียน
 - การเรียกใช้งาน ข้อมูลในแฟ้มนี้จะนำไปรวมกับข้อมูลในแฟ้มข้อมูลโครงสร้างบทเรียนเพื่อแสดงสรุปผลการเรียนของผู้เรียนในวิชานั้นทั้งหมด ก่อนการเข้าเรียนเนื้อหาวิชาที่ต้องการ และจะแสดงข้อมูลนอกคั้งเมื่อผู้เรียนเลิกเรียน

4) โครงสร้างแฟ้ม ประกอบด้วยรายการข้อมูลดังนี้

- (1) ชื่อแฟ้มหัวข้อ (char fname[13])
- (2) ลำดับที่หัวข้อ (int itm_no)
- (3) ชื่อหัวข้อ (char obj_name[118])
- (4) วัน-เดือน-ปี ที่เข้าเรียน (char l_date[11])
- (5) เวลาที่เริ่มใช้บทเรียน (char s_time[9])
- (6) เวลาทั้งหมดที่ใช้ในการเรียน (int ut_time)
- (7) จำนวนข้อตอบถูก (int itm_right)
- (8) จำนวนข้อตอบผิด (int itm_wrong)
- (9) จำนวนข้อตอบไม่มีถูกผิด (int itm_non)
- (10) คะแนนเต็ม (int fullscore)
- (11) คะแนนทั้งหมดที่ได้ (int score)
- (12) เกณฑ์การประเมินผล (char e_law[100])
- (13) ผลการประเมิน (char evaluate)

กำหนดให้มีรหัสดังนี้ T = บทเรียนมีปัญหา หยุดการเรียน

S = ผู้เรียนหยุดเรียนกลางคัน

อื่น ๆ = ผลการประเมิน

5) หมายเหตุ

- การเรียนครั้งสุดท้าย ตรวจสอบได้จาก วันเดือนปี เวลาครั้งล่าสุด
- การสรุปผลการประเมิน (ผ่าน/ไม่ผ่าน/เกรด) จะกระทำเมื่อจบการเรียนในแต่ละแฟ้ม เนื้อหาเท่านั้น มิฉะนั้นจะรายงานเฉพาะ ถูก/ผิด ก็ข้อ
- การเลิกเรียนกลางคัน ให้ถือว่าผู้เรียนจะต้องเริ่มต้นเรียนแฟ้มนั้นใหม่

4.5.1.3 การพิจารณากำหนดการประมวลผล จากการพิจารณา

กำหนดให้การประมวลผลประกอบด้วย 2 โปรแกรมหลัก ดังนี้

ก. โปรแกรมผู้สอน (TEACH.EXE) เป็นโปรแกรมหลักของระบบ ทำหน้าที่นำแฟ้มข้อมูลเนื้อหาวิชามาเสนอให้ผู้เรียนศึกษา การนำเสนอเนื้อหาวิชาจะดำเนินการตามคำสั่งควบคุมในภาษา TAL ที่อยู่ในแฟ้มข้อมูลเนื้อหาวิชานั้น ๆ มีขั้นตอนการประมวลผล ดังนี้

รับข้อมูลของผู้เรียน (รหัสประจำตัว รหัสวิชา)

เปิดแฟ้ม เพิ่มข้อมูลสรุปผลการเรียนของผู้เรียน

เพิ่มข้อมูลโครงสร้างของบทเรียน

เชื่อมโยง 2 เพิ่มข้างต้น แสดงสถานะของผู้เรียน และให้ผู้เรียนเลือกจุด

ประสงค์หรือตำแหน่งที่ต้องการเข้าเรียน

เปิด เพิ่มข้อมูลเนื้อหาวิชาที่ผู้เรียนเลือก

เพิ่มข้อมูลบันทึกการเรียนของผู้เรียน

เพิ่มข้อมูลโต้ตอบ

สร้างตารางควบคุมบทเรียน

ดำเนินการประมวลผลเสนอเนื้อหาวิชาตามการควบคุมของภาษา TAL

บันทึกการเรียนของผู้เรียนขณะเรียน จนกระทั่งเลิกเรียน ซึ่งเป็นไปได้ใน

4 กรณี คือ TAL_Error, //END, 999, ครอบสุดท้าย

เมื่อจบการเรียนแสดงสรุปผลการเรียน

ข. โปรแกรมวิเคราะห์ผลการเรียน (ANALYSER.EXE)

เป็นโปรแกรมสำหรับใช้ดูข้อมูลผลการเรียนและข้อมูลพฤติกรรมการเรียนของผู้เรียนราย

บุคคล

4.5.1.4 การพิจารณากำหนดส่วนข้อมูลนำออก ซึ่งเป็นผลลัพธ์ที่ได้จากการใช้งานโปรแกรม TEACH ประกอบด้วย การแสดงผลทางจอภาพ ได้แก่ การนำเสนอเนื้อหาวิชาทางจอภาพ การนำเสนอข้อมูลสถานะการเรียนของผู้เรียนและเพิ่มข้อมูล 2 แฟ้มคือ แฟ้มข้อมูลบันทึกพฤติกรรมการเรียนของผู้เรียน และเพิ่มข้อมูลสรุปผลการเรียนของผู้เรียนซึ่งเป็นแฟ้มที่ใช้ในส่วนข้อมูลนำเข้าในตอนต้น

เพิ่มข้อมูลบันทึกพฤติกรรมการเรียน

1) ชื่อ SSS000.III

เมื่อ SSS คือ รหัสวิชา

000 คือ เลขที่จุดประสงค์

III คือ รหัสผู้เรียน

2) จุดประสงค์ เพื่อบันทึกพฤติกรรมการเรียนของผู้เรียนที่กระทำกับแฟ้มเนื้อหาวิชา โดยกำหนดให้การเรียนของผู้เรียนที่เรียนกับหนึ่งแฟ้มเนื้อหาวิชา บันทึกเก็บไว้ในหนึ่งแฟ้มข้อมูล ฉะนั้นในการเรียนของผู้เรียนแต่ละคนในแต่ละแฟ้มข้อมูลเนื้อหาวิชาจะเกิดแฟ้มข้อมูลบันทึกการเรียนขึ้นหนึ่งแฟ้มเสมอ กรณีที่ผู้เรียนเข้าเรียนบทเรียนที่เคยเรียนมาแล้ว ข้อมูลพฤติกรรมการเรียนจะถูกบันทึกต่อท้ายแฟ้มที่มีอยู่เดิม การบันทึกพฤติกรรมการเรียนจะบันทึกข้อมูล 1 ชุด เมื่อสิ้นสุดกรอบการเรียนแต่ละกรอบ

3) การนำไปใช้

- ประเมินผลการเรียน คิดคะแนน
- ตรวจสอบพฤติกรรมการเรียนของผู้เรียน เพื่อให้คำปรึกษาแนะนำ
- ประเมินคุณภาพของแฟ้มข้อมูลเนื้อหาวิชา

4) โครงสร้างแฟ้ม ประกอบด้วยรายการข้อมูลใน 1 ระเบียบ ดังนี้

- (1) หมายเลขแฟ้ม (int f_no)
- (2) ชนิดของแฟ้ม (char f_type)
- (3) จำนวนแฟ้มคำถามสะสม (int f_qno)
- (4) หมายเลขครั้งที่อนุญาตให้เรียกใช้แฟ้มซ้ำ (int loop_c)
- (5) หมายเลขครั้งที่เคยเรียกใช้แฟ้ม (int loop_u)
- (6) จำนวนเวลาสะสมที่ผ่านมาในบทเรียนก่อนเข้าใช้แฟ้มปัจจุบัน (int ti_lu)
- (7) เวลาเริ่มเข้าใช้แฟ้ม (int ti_f)
- (8) จำนวนเวลาทั้งหมดที่ใช้ในแฟ้ม (int ti_fu)
- (9) จำนวนเวลาที่กำหนดให้สำหรับแสดงข้อความในแฟ้ม (int ti_fs)
- (10) จำนวนเวลาที่ใช้จริงในการแสดงข้อความในแฟ้ม (int ti_fsu)
- (11) จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในแฟ้ม (int ti_fa)
- (12) จำนวนเวลาที่ใช้ในการตอบคำถามในแฟ้ม (int ti_fau)
- (13) จำนวนเวลาที่กำหนดให้ใช้แสดงข้อความโต้ตอบในแฟ้ม (int ti_fb)
- (14) จำนวนเวลาที่ใช้จริงในการแสดงข้อความโต้ตอบในแฟ้ม (int ti_fbu)
- (15) คำตอบของผู้เรียน (char ans_s[80])
- (16) ผลลัพธ์จากคำตอบ (char ans_r)

- (17) หมายเลขเฟรมต่อไป (int next_f)
 เมื่อ 1-100 = หมายเลขเฟรม
 998 = เลิกเรียนการค้น
 999 = จบการเรียน
- (18) จำนวนข้อสอบถูกสะสม (int no_r)
 (19) จำนวนข้อสอบผิดสะสม (int no_w)
 (20) จำนวนข้อสอบไม่มีถูกผิดสะสม (int no_n)
 (21) คะแนนที่ได้ในเฟรมนี้ (int sc_g)
 (22) คะแนนสะสมที่ได้ (int sc_u)
 (23) คะแนนรวมสะสม (int sc_t)
 (24) คะแนนเต็มของเฟรมนี้ (int sc_f)
- 5) หมายเหตุ การเข้าเรียน 1 เฟรม จะบันทึกรายการข้อมูล 24 รายการ การเก็บข้อมูลจะเกิดขึ้นทุกครั้งที่จบการเรียนในเฟรมหนึ่ง ๆ

4.3.2 ออกแบบขั้นตอนการทำงานของโปรแกรม ในส่วนนี้จะใช้ผังงานเป็นเครื่องมือในการออกแบบแสดงความหมายและขั้นตอนการทำงานของโปรแกรม ตลอดจนการเคลื่อนไหวของข้อมูลอย่างละเอียด แบ่งเป็น 3 ขั้นตอน คือ

- 4.3.2.1 ออกแบบผังงานระบบ(system flowchart)
 4.3.2.2 ออกแบบผังงานมอดูล(modular program flowchart)
 4.3.2.3 ออกแบบผังงานรายละเอียด(detailed program flowchart)

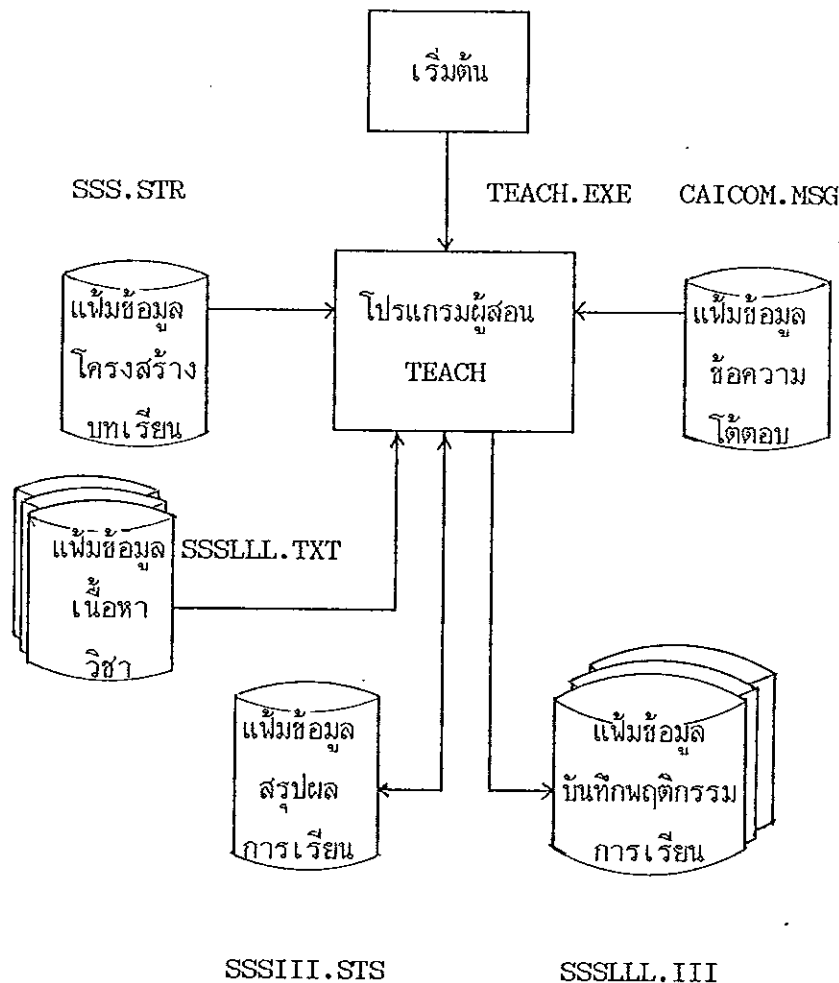
4.3.2.1 ออกแบบผังงานระบบ จุดมุ่งหมายของการออกแบบผังงานนี้ เพื่อแสดงการดำเนินงานของระบบว่ามีขอบเขตกว้าง ๆ อย่างไร ให้ทราบถึงความเกี่ยวพันของระบบงานตั้งแต่เริ่มต้นว่า มีการปฏิบัติงานและขั้นตอนอย่างไร ใช้วิธีการอย่างไร จากวัตถุประสงค์ที่ได้กำหนดไว้เมื่อนำมาพิจารณาพร้อมกับส่วนข้อมูลนำเข้า วิธีการประมวลผล และส่วนข้อมูลนำออก สามารถออกแบบผังงานระบบของระบบ TEACH แยกเป็น 2 ส่วน

ส่วนที่ 1 มีโปรแกรมหลัก TEACH เป็นโปรแกรมควบคุมจัดการการเรียนการสอน ดำเนินการสอน การโต้ตอบ บันทึกพฤติกรรมการณ์เรียนของผู้เรียน บันทึกสรุปผลการเรียน มีแฟ้มข้อมูลที่เกี่ยวข้องในการประมวลผล 5 แฟ้มข้อมูลคือ แฟ้มข้อมูลโครงสร้างบทเรียน แฟ้มข้อมูลเนื้อหาวิชา แฟ้มข้อมูลสรุปผลการเรียน แฟ้มข้อมูลบันทึกพฤติกรรม และแฟ้มข้อมูลข้อความโต้ตอบ

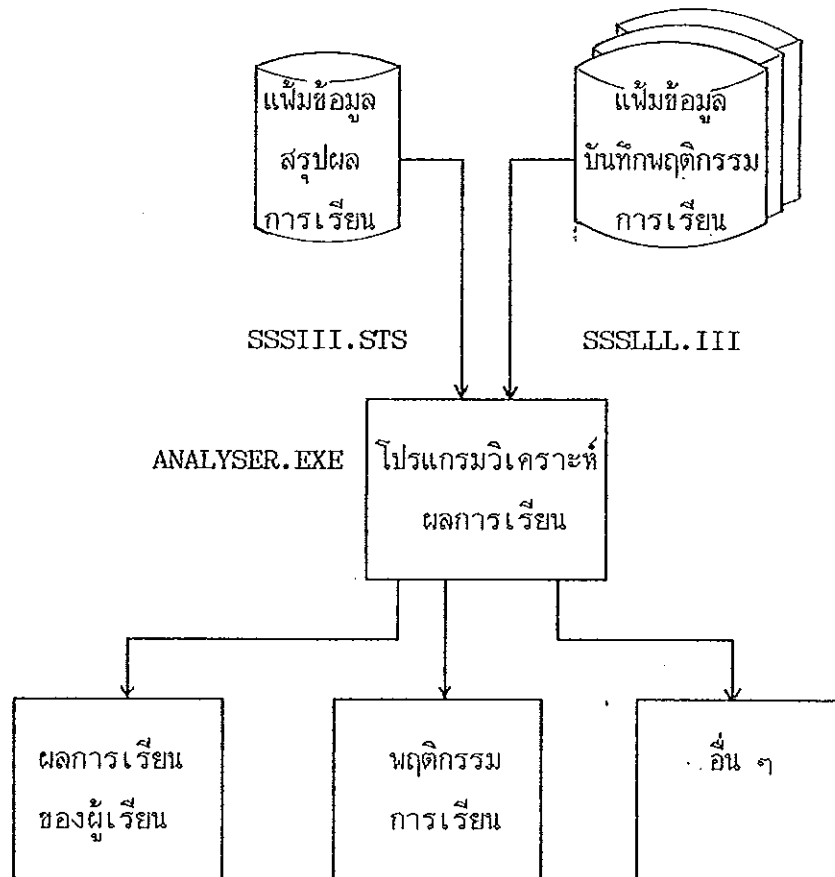
ส่วนที่ 2 มีโปรแกรมหลัก ANALYSER สำหรับให้ผู้สอนใช้ดูข้อมูลสรุป ผลการเรียนและดูข้อมูลพฤติกรรมการณ์เรียนของผู้เรียน ซึ่งเป็นประโยชน์ต่อการวิเคราะห์ประสิทธิภาพของบทเรียน ให้คำปรึกษาแก่ผู้เรียน ข้อมูลที่ใช้ในส่วนนี้คือข้อมูลจากแฟ้มข้อมูลสรุปผลการเรียน และแฟ้มข้อมูลบันทึกพฤติกรรม ซึ่งเป็ผลลัพท์จากส่วนที่ 1

ผังงานระบบของระบบ TEACH สามารถเขียนได้ดังภาพประกอบ 4 และ 5

ภาพประกอบ 4 ผังงานระบบของระบบ TEACH ส่วนที่ 1



ภาพประกอบ 5 ผังงานระบบของระบบ TEACH ส่วนที่ 2

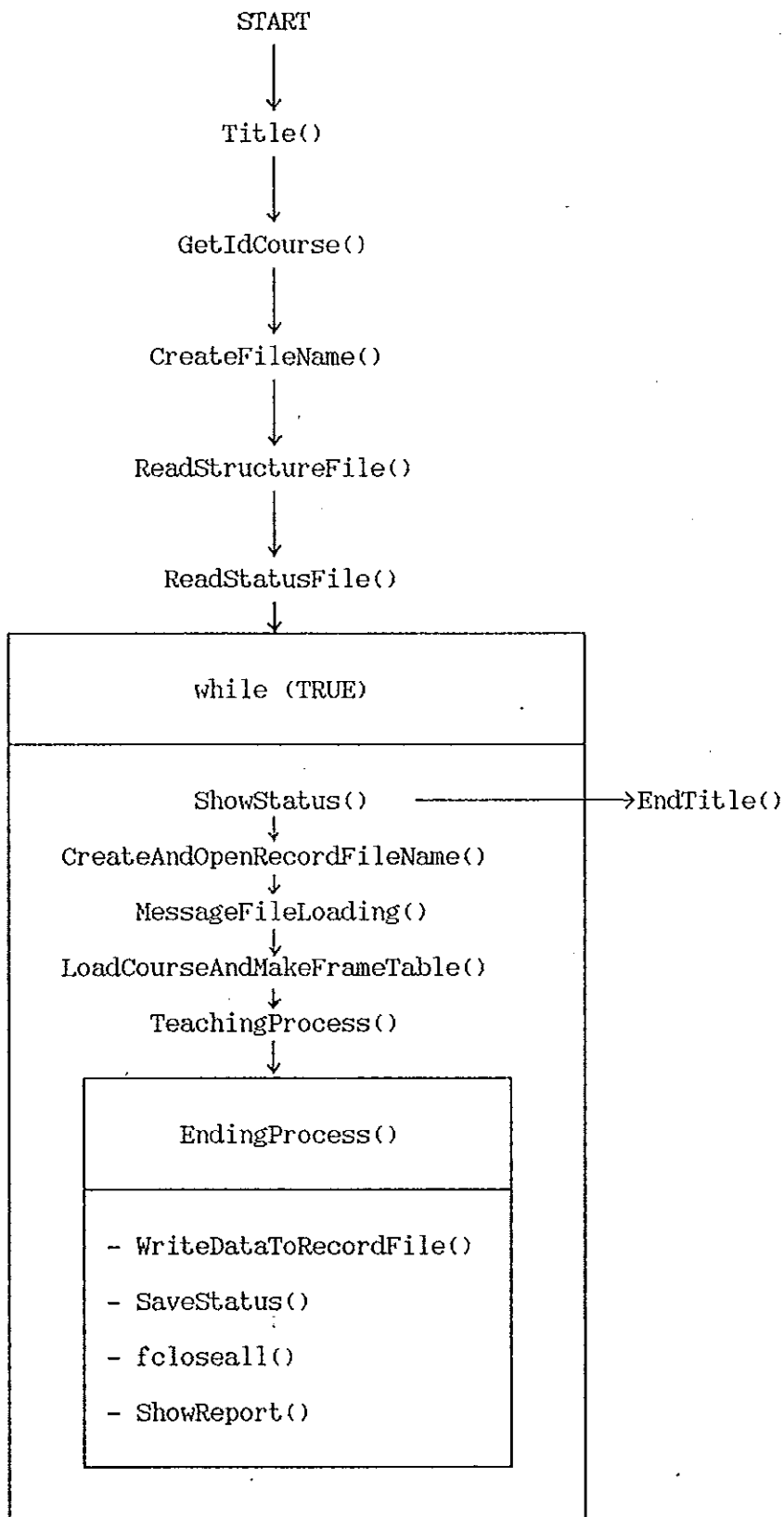


จากแผนผังแสดงความสัมพันธ์ระหว่างโปรแกรมกับแฟ้มข้อมูลต่าง ๆ ในระบบโปรแกรม TEACH จะประกอบด้วย 2 โปรแกรมหลัก และ 5 แฟ้มข้อมูล

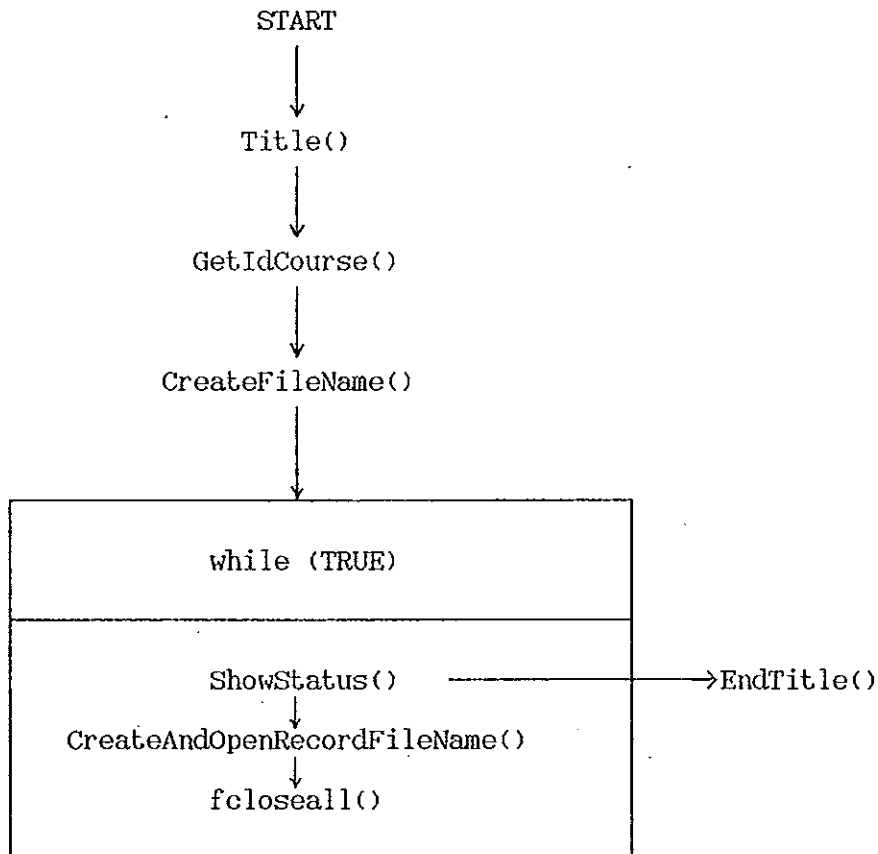
4.3.2.2 ออกแบบผังงานมอดูล เป็นการออกแบบผังงานเพื่อแบ่งโปรแกรมออกเป็นส่วน ๆ ให้ทำหน้าที่เฉพาะเรื่อง ในแต่ละส่วนมีลักษณะคล้ายโปรแกรมย่อย ผังงานนี้ใช้แสดงขั้นตอนการใช้งานมอดูลต่าง ๆ ในโปรแกรม โดยแสดงเฉพาะมอดูลหลักกว่ามีการเรียงกันเป็นอย่างไรในโปรแกรม ส่วนขั้นตอนการทำงานซึ่งเป็นรายละเอียดในแต่ละมอดูล จะแสดงในผังงานรายละเอียด

จากผังงานระบบและการกำหนดวัตถุประสงค์สามารถออกแบบผังงานมอดูลสำหรับโปรแกรม TEACH.EXE และ ANALYSER.EXE ได้ดังภาพประกอบ 6 และ 7 ตามลำดับ

ภาพประกอบ 6 ผังงานมอดูลสำหรับโปรแกรม TEACH.EXE



ภาพประกอบ 7 ผังงานมอดูลสำหรับโปรแกรม ANALYSER.EXE



4.3.2.3 ออกแบบผังงานรายละเอียด เป็นการออกแบบผังงานที่แสดงรายละเอียดของผังงานโปรแกรมมอดูลว่า แต่ละมอดูลมีขั้นตอนการทำงานอย่างไร ซึ่งจะไปลงรหัสด้วยภาษา C ต่อไป โดยอาจมีการเปลี่ยนแปลงรายละเอียดบางขั้นตอนตามรูปแบบของภาษา และเนื่องจากภาษา C ใช้คำว่า ฟังก์ชัน ในความหมายที่เทียบได้กับคำว่า มอดูล ดังนั้นเพื่อสะดวกกับการลงรหัส จึงขอใช้คำว่า ฟังก์ชัน แทน คำว่า มอดูล

การออกแบบผังงานรายละเอียดสำหรับการพัฒนาโปรแกรมครั้งนี้ จะใช้การให้ความหมาย รูปแบบ วิธีการทำงาน ข้อมูลเข้า ผลลัพธ์ที่ได้ของแต่ละฟังก์ชัน แทนการเขียนผังงานรายละเอียดในลักษณะภาพผังงาน และเมื่อลงรหัสด้วยภาษา C จะให้คำอธิบายการทำงานของโปรแกรมเพิ่มเติม ซึ่งจะช่วยให้สามารถทำความเข้าใจโปรแกรมได้ง่ายขึ้น

ฟังก์ชันที่ได้รับการออกแบบสำหรับใช้งานในส่วนของโปรแกรม TEACH.EXE มีดังนี้

- (1) void Pause(void)
- (2) int WaitTime(char StrA[])
- (3) int Wait(int Long)
- (4) int WaitBranch(int Long)
- (5) void Title(void)
- (6) void GetIdCourse(void)
- (7) void ReadStructureFile(void)
- (8) void ReadStatusFile(void)
- (9) void ShowStatus(void)
- (10) void CreateAndOpenRecordFileName(void)
- (11) void WriteDataToRecordFile(void)
- (12) void SaveStatus(void)
- (13) void LoadCourseAndMakeFrameTable(void)
- (14) int MessageFileLoading(void)
- (15) int FrameStatement(char StrA[])
- (16) void ScreenStatement(char StrA[])
- (17) int InputStatement(char StrA[])
- (18) void BranchStatement(char StrA[])
- (19) void EvaluateStatement(char StrA[])
- (20) char Grade(int x)
- (21) int DecodeCommands(int s_cmds, int e_cmds, char Str[])
- (22) int Get1Parameter(char Str[], int ptr)
- (23) void ShowReport(void)
- (24) void EndingProcess(void)
- (25) void EndTitle(void)
- (26) void InitialValue(void)
- (27) int CrossBlank(char Str[], int p)

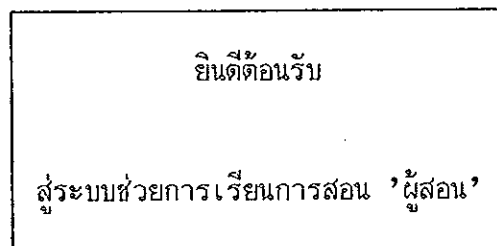
- (28) void TeachingProcess(void)
- (29) int RandomFrame(char StrA[])
- (30) void DisplayText(void)
- (31) int Display(char StrA[])
- (32) int SetCenter(char Lbl[])
- (33) int DecodeScore(char StrA[])
- (34) void Branching(void)
- (35) int colon(char StrA[])
- (36) int Message(char StrA[])
- (37) char *strmid(char s[], unsigned p, unsigned n)
- (38) char *strcut(char s[], unsigned p)
- (39) int strin(char s[], int p, char c)
- (40) char *strdate(void)
- (41) char *strtime(time_t t)
- (42) char *strctime(long t)
- (43) void TAL_Error(int codeno)

ความหมาย วิธีการทำงาน ข้อมูลเข้า ผลลัพธ์และรูปแบบฟังก์ชันของแต่ละฟังก์ชัน มีดังนี้

- (1) Pause ฟังก์ชันสำหรับหยุดรอรับการเคาะแป้นพิมพ์จากผู้ใช้ โปรแกรม
ถ้าเคาะแป้น ESC จะออกจากการใช้งาน โปรแกรม
ส่วนแป้นอื่น ๆ จะดำเนินโปรแกรมต่อไป
รูปแบบฟังก์ชัน void Pause(void)
- (2) WaitTime ฟังก์ชันสำหรับวิเคราะห์หาคำสั่ง \$T
input คือ ข้อความที่ต้องการวิเคราะห์หา \$T (char StrA[])
output คือ จำนวนเวลาที่จะให้รอคอย (int WaitTime)
รูปแบบฟังก์ชัน int WaitTime(char StrA[])

- (3) Wait ฟังก์ชันสำหรับหน่วงเวลา
 input คือ ระยะเวลาที่ต้องการหน่วง (int Long)
 output คือ ระยะเวลาที่ต้องคอย
 รูปแบบฟังก์ชัน int Wait(int Long)
- (4) WaitBranch ฟังก์ชันสำหรับหน่วงเวลา
 input คือ ระยะเวลาที่ต้องการหน่วง (int Long)
 output คือ ระยะเวลาที่ต้องคอย
 รูปแบบฟังก์ชัน int WaitBranch(int Long)
- (5) Title ฟังก์ชันสำหรับแสดงข้อความส่วนนำเข้าสู่ระบบ
 input คือ -
 output คือ ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบ
 รูปแบบฟังก์ชัน void Title(void)

ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบของฟังก์ชัน Title มีดังนี้



โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต

มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536

ที่ปรึกษาโครงการ : อาจารย์วุฒิพงศ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
 รศ. ไพโรจน์ ตีรณธนากุล กรรมการที่ปรึกษา

พัฒนาระบบโดย : นายนิพนธ์ ศุภศรี

- (6) GetIdCourse ฟังก์ชันสำหรับรับการป้อนรหัสผู้เรียนและชื่อบทเรียน
 input คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
 output คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
 รูปแบบฟังก์ชัน void GetIdCourse(void)

โดยมีหน้าจอการรับข้อมูล ดังนี้

ร ะ บ บ ใ ช้ ใ ช้ ก าร เ ร ย น ก าร ส ่อ น ' ผู้ ส ่อ น '

=====

โปรดป้อนชื่อบทเรียน[SSS###] :

โปรดป้อนรหัสผู้เรียน[XXXX] :

- (7) ReadStructureFile ฟังก์ชันสำหรับอ่านข้อมูลในแฟ้มโครงสร้างบทเรียนมาเก็บในตารางข้อมูลประวัติการเรียน (HistoryTBL)
 input คือ ข้อมูลในแฟ้มโครงสร้างบทเรียน
 output คือ ตารางข้อมูลประวัติการเรียน (HistoryTBL)
 รูปแบบฟังก์ชัน void ReadStructureFile(void)
- (8) ReadStatusFile ฟังก์ชันสำหรับอ่านข้อมูลในแฟ้มสรุปผลการเรียนมาเก็บในตารางข้อมูลประวัติการเรียน (HistoryTBL)
 input คือ ข้อมูลในแฟ้มสรุปผลการเรียน
 output คือ ตารางข้อมูล HistoryTBL ลำดับหัวข้อที่ต่อเนื่องจากการเรียนครั้งก่อน (ItmPtr)
 รูปแบบฟังก์ชัน void ReadStatusFile(void)
- (9) ShowStatus ฟังก์ชันสำหรับแสดงประวัติการเรียน (HistoryTBL) เพื่อให้เลือกหัวข้อที่จะเข้าเรียน
 input คือ ตารางข้อมูลประวัติการเรียน (HistoryTBL)
 output คือ ลำดับหัวข้อที่จะเข้าเรียน (ItmPtr)
 รูปแบบฟังก์ชัน void ShowStatus(void)

จอภาพแสดงประวัติการเรียน เพื่อให้เลือกหัวข้อที่จะเข้าเรียน เป็นต้น

คุณมีประวัติการศึกษาวิชา..... ตั้งขึ้น

หัวข้อ	เรียนเมื่อ	รวม	ผลการเรียน		
ที่	เรื่อง	วัน-เดือน-ปี(เวลา)	ใช้เวลา	คะแนน	เกรด
1
2
3

โปรด ใส่เลขหัวข้อที่คุณต้องการศึกษา
หรือ กด ENTER เข้าสู่การศึกษาหัวข้อที่...ต่อจากครั้งก่อน
หรือ กด 999 เลิกการใช้งาน >

(10) CreateAndOpenRecordFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มบันทึก
พฤติกรรมกรเรียน และเปิดแฟ้มใช้งาน แฟ้มนี้จะเกิดขึ้นได้
เมื่อมีการเลือกหัวข้อที่ต้องการเรียนแล้ว
input คือ ข้อมูลจากการเลือกหัวข้อที่ต้องการเรียน
output คือ ชื่อแฟ้มบันทึกพฤติกรรมกรเรียน (SSSIII.STS)
และการเปิดแฟ้มใช้งาน
รูปแบบฟังก์ชัน void CreatAndOpenRecordFileName(void)

(11) WriteDataToRecordFile ฟังก์ชันสำหรับเขียนข้อมูลลงแฟ้มบันทึก
พฤติกรรมกรเรียน
input คือ ข้อมูลพฤติกรรมกรเรียนในแต่ละกรอบกรเรียน
output คือ การเขียนข้อมูลลงแฟ้มบันทึกพฤติกรรมกรเรียน
รูปแบบฟังก์ชัน void WriteDataToRecordFile(void)

- (12) SaveStatus ฟังก์ชันสำหรับเขียนข้อมูลลงเพิ่มสรุปผลการเรียน
ฟังก์ชันนี้จะถูกใช้เมื่อยุติการเรียน
input คือ ข้อมูลสรุปผลการเรียน
output คือ การเขียนข้อมูลลงเพิ่มสรุปผลการเรียน
รูปแบบฟังก์ชัน void SaveStatus(void)
- (13) LoadCourseAndMakeFrameTable ฟังก์ชันสำหรับอ่านข้อมูลจากเพิ่ม
เนื้อหาวิชาบรรจุลงตารางข้อมูลบทเรียน (CDTBLE[500][100])
และตารางควบคุมกรอบการเรียน (FCTBLE[50][5])
input คือ ข้อมูลจากเพิ่มเนื้อหาวิชา
output คือ ตารางข้อมูลบทเรียน (CDTBLE[500][100])
ตารางควบคุมกรอบการเรียน (FCTBLE[50][5])
ลำดับที่สูงสุดของกรอบการเรียน (FCTBLE[0][1])
รูปแบบฟังก์ชัน void LoadCourseAndMakeFrameTable(void)
- (14) MessageFileLoading ฟังก์ชันสำหรับอ่านข้อมูลจากเพิ่มข้อความโต้ตอบ
(CAICOM.MSG) บรรจุลงตารางข้อมูลโต้ตอบ (MDTBLE[20][100])
input คือ ข้อมูลจากเพิ่มข้อความโต้ตอบ (CAICOM.MSG)
output คือ ตารางข้อมูลโต้ตอบ (MDTBLE), ลำดับที่สูงสุดของข้อความ
โต้ตอบ (MaxMGSNo)
รูปแบบฟังก์ชัน int MessageFileLoading(void)
- (15) FrameStatement ฟังก์ชันสำหรับดำเนินการเรียนการสอนตามคำสั่ง FRAME
input คือ ข้อความหลังคำสั่ง FRAME (char StrAC[])
output คือ บันทึกพฤติกรรมการเรียนของเฟรมที่แล้วลงเพิ่มบันทึกพฤติกรรม
การเรียน
รหัสผลการดำเนินการ (int FrameStatement)
รูปแบบฟังก์ชัน int FrameStatement(char StrAC[])

- (16) ScreenStatement ฟังก์ชันดำเนินการเรียนการสอนตามคำสั่ง Screen
input คือ ข้อความหลังคำสั่ง Screen (char StrA[])
output คือ แสดงข้อความหลังคำสั่ง SCREEN บนจอภาพ
และกำหนดเวลาในการแสดงข้อความ
รูปแบบฟังก์ชัน void ScreenStatement(char StrA[])
- (17) InputStatement ฟังก์ชันสำหรับดำเนินการเรียนการสอนตามคำสั่ง INPUT
input คือ ข้อความหลังคำสั่ง INPUT (char StrA[])
output คือ รหัสผลการดำเนินการ(int InputStatement)
รูปแบบฟังก์ชัน int InputStatement(char StrA[])
- (18) BranchStatement ฟังก์ชันสำหรับวิเคราะห์คำสั่ง BRANCH
input คือ ข้อความหลังคำสั่ง Branch (char StrA[])
output คือ คะแนนเต็มสำหรับเฟรมนี้ Vn
จำนวนเวลาที่รอคอยก่อนไปดำเนินงานในเฟรมถัดไป ST
รูปแบบฟังก์ชัน void BranchStatement(char StrA[])
- (19) EvaluateStatement ฟังก์ชันสำหรับวิเคราะห์คำสั่งการประเมินผลการเรียน
input คือ ข้อความที่ต้องการวิเคราะห์ (char StrA[])
output คือ ตารางระดับคะแนน
รูปแบบฟังก์ชัน void EvaluateStatement(char StrA[])
- (20) Grade ฟังก์ชันสำหรับให้เกรดผู้เรียน
input คือ คะแนนที่ผู้เรียนทำได้ (int x)
output คือ เกรดที่ได้ (char Grade)
รูปแบบฟังก์ชัน char Grade(int x)

(21) DecodeCommands ฟังก์ชันสำหรับจำแนกคำสั่งต่าง ๆ ของภาษา TAL

input คือ ลำดับที่เริ่มต้นของคำสั่งในตารางคำสั่ง (int s_cmds)
 ลำดับที่สุดท้ายของคำสั่งในตารางคำสั่ง (int e_cmds)
 ข้อความที่ต้องการให้จำแนกคำสั่ง (char Str[])

output คือ ลำดับที่ของคำสั่งในตารางคำสั่ง
 ถ้าไม่อยู่ในกลุ่มของคำสั่งที่กำหนดจะให้ค่าเท่ากับ 0

รูปแบบฟังก์ชัน

```
int DecodeCommands(int s_cmds, int e_cmds, char Str[])
```

(22) Get1Parameter ฟังก์ชันสำหรับจำแนกหาอักขระนำคำสั่งของภาษา TAL

input คือ ข้อความที่ต้องการจำแนกหาอักขระนำคำสั่ง (char Str[])
 ตำแหน่งของอักขระที่จะเริ่มต้นค้นหา (int ptr)

output คือ รหัสของอักขระที่พบ (int Get1Parameter)
 ข้อความหลังอักขระที่พบ (char StrOut[100])
 ตำแหน่งของอักขระที่จะเริ่มต้นหลังการค้นหา (int ActPtr)

รูปแบบฟังก์ชัน int Get1Parameter(char Str[], int ptr)

(23) ShowReport ฟังก์ชันสำหรับแสดงผลการเรียน

input คือ ข้อมูลจากแฟ้มสรุปผลการเรียน
 output คือ ข้อความแสดงผลการเรียนบนจอภาพ

รูปแบบฟังก์ชัน void ShowReport(void)

จอภาพแสดงผลการเรียน เป็นดังนี้

ผลการเรียนวิชา

หัวข้อที่ ... เรื่อง

เรียนเมื่อ วัน-เดือน-ปี เวลา ____ น. รวมใช้เวลา(ชม.:นาที:วินาที) ____

ผลการเรียน ตอบถูก ____ ข้อ ผิด ____ ข้อ สรุปได้ ____ / ____ คะแนน

คิดเป็นร้อยละ ____ ประเมินโดยใช้เกณฑ์ ____ ได้เกรด ____

(24) EndingProcess ฟังก์ชันสำหรับเก็บข้อมูลและปิดเพิ่มข้อมูลต่าง ๆ

ก่อนการจบการเรียน

input คือ -

output คือ การปิดเพิ่มข้อมูลต่าง ๆ

รูปแบบฟังก์ชัน void EndingProcess(void)

(25) EndTitle ฟังก์ชันสำหรับแสดงข้อความจบการใช้งานระบบ

input คือ -

output คือ ข้อความแสดงการจบการใช้งานบนจอภาพ

void EndTitle(void)

จอภาพแสดงข้อความจบการใช้งานระบบ

<p style="text-align: center;">ขอบคุณที่ใช้บริการ ระบบช่วยการเรียนการสอน 'ผู้สอน' สวัสดิ์</p>

(26) InitialValue ฟังก์ชันสำหรับกำหนดค่าเริ่มต้นก่อนการเรียน

input คือ -

output คือ ข้อมูลเริ่มต้นสำหรับเพิ่มต่าง ๆ และการทำงานของระบบ

รูปแบบฟังก์ชัน void InitialValue(void)

(27) CrossBlank ฟังก์ชันสำหรับล้างข้ามช่องว่าง

input คือ ข้อความที่ต้องการให้ข้ามช่องว่าง(char Str[])

ลำดับที่ของตัวอักขระที่ให้เริ่มต้นตรวจหาช่องว่าง(int p)

output คือ ลำดับที่ของตัวอักขระที่ไม่ใช่ช่องว่าง(int CrossBlank)

รูปแบบฟังก์ชัน int CrossBlank(char Str[], int p)

- (28) TeachingProcess ฟังก์ชันสำหรับควบคุมการเรียนการสอนตามบทเรียน
input คือ ข้อมูลเนื้อหาวิชาในตารางข้อมูลบทเรียน (CDTBLE[500][100])
ข้อมูลจากตารางควบคุมรอบการเรียน (FCFBLE[50][5])
output คือ การนำเสนอบทเรียน โดยการส่งการทำงานไปยังฟังก์ชัน
ต่อไปที่ FrameStatement() ScreenStatement()
DisplayText() InputStatement() BranchStatement()
Branching() EvaluateStatement()
รูปแบบฟังก์ชัน void TeachingProcess(void)
- (29) RandomFrame ฟังก์ชันสำหรับสุ่มหมายเลขเฟรม
input คือ ข้อความกลุ่มหมายเลขเฟรม (char StrA[])
output คือ หมายเลขเฟรมที่สุ่มได้ (int RandomFrame)
รูปแบบฟังก์ชัน int RandomFrame(char StrA[])
- (30) DisplayText ฟังก์ชันสำหรับการดำเนินการตามคำสั่งแสดงผลบนจอภาพ
input คือ ข้อความหรือคำสั่งแสดงผล (char StrA[])
output คือ ผลลัพธ์การแสดงผลบนจอภาพตามคำสั่งแสดงผล
รูปแบบฟังก์ชัน void DisplayText(void)
- (31) Display ฟังก์ชันสำหรับแสดงข้อความบนจอภาพ
input คือ ข้อความที่ต้องการนำเสนอบนจอภาพ (char StrA[])
output คือ รหัสผลการดำเนินการ (int Display)
รูปแบบฟังก์ชัน int Display(char StrA[])
- (32) SetCenter ฟังก์ชันสำหรับกำหนดตำแหน่งคอลัมน์ ที่จะทำให้แสดงข้อความ
อยู่กึ่งกลางจอภาพ
input คือ ข้อความที่ต้องการนำเสนอบนจอภาพ (char Lbl[])
output คือ ตำแหน่งคอลัมน์ที่ทำให้ข้อความอยู่กึ่งกลางจอ (int SetCenter)
รูปแบบฟังก์ชัน int SetCenter(char Lbl[])

- (38) `strcut` ฟังก์ชันสำหรับคัดลอกข้อความ(s) จากตำแหน่งที่กำหนด(p) ไปจนหมดข้อความ
input คือ ข้อความ(char s[]), ตำแหน่งที่กำหนด(unsigned p)
output คือ ข้อความ(char *strcut)
รูปแบบฟังก์ชัน `char *strcut(char s[], unsigned p)`
- (39) `strin` ฟังก์ชันสำหรับหาตำแหน่งของอักขระที่กำหนด(c)ตัวแรกที่พบบนข้อความ(s) โดยทำการค้นหาตั้งแต่ตำแหน่ง(p) ไปจนจบข้อความ
input คือ ข้อความ(char s[]), ตำแหน่งที่กำหนด(unsigned p), ตัวอักขระที่ต้องการค้นหา(char c)
output คือ ตำแหน่งของอักขระที่กำหนดตัวแรกที่พบบนข้อความ
(int strin) ถ้าไม่พบ strin = 0
รูปแบบฟังก์ชัน `int strin(char s[], int p, char c)`
- (40) `strdate` ฟังก์ชันสำหรับเรียกดูวัน เดือน ปี ของเครื่องในรูปแบบข้อความ
input คือ วัน เดือน ปี ของเครื่อง
output คือ วัน เดือน ปี ของเครื่องในรูปแบบข้อความ
(char *strdate) ดังนี้ dd-mm-yyyy
รูปแบบฟังก์ชัน `char *strdate(void)`
- (41) `strtime` ฟังก์ชันสำหรับเรียกดูเวลาของเครื่องในรูปแบบข้อความ
input คือ เวลาของเครื่องในขณะนั้น(time_t t)
output คือ เวลาของเครื่องในรูปแบบข้อความ
(char *strtime) ดังนี้ HH:MM:SS
รูปแบบฟังก์ชัน `char *strtime(time_t t)`

- (38) `strcut` ฟังก์ชันสำหรับตัดลอกข้อความ(s) จากตำแหน่งที่กำหนด(p) ไปจนหมดข้อความ
input คือ ข้อความ(char s[]), ตำแหน่งที่กำหนด(unsigned p)
output คือ ข้อความ(char *strcut)
รูปแบบฟังก์ชัน `char *strcut(char s[], unsigned p)`
- (39) `strin` ฟังก์ชันสำหรับหาตำแหน่งของอักขระที่กำหนด(c)ตัวแรกที่พบบนข้อความ(s) โดยทำการค้นหาตั้งแต่ตำแหน่ง(p) ไปจนจบข้อความ
input คือ ข้อความ(char s[]), ตำแหน่งที่กำหนด(unsigned p), ตัวอักขระที่ต้องการค้นหา(char c)
output คือ ตำแหน่งของอักขระที่กำหนดตัวแรกที่พบบนข้อความ
(int strin) ถ้าไม่พบ strin = 0
รูปแบบฟังก์ชัน `int strin(char s[], int p, char c)`
- (40) `strdate` ฟังก์ชันสำหรับเรียกดูวัน เดือน ปี ของเครื่องในรูปแบบข้อความ
input คือ วัน เดือน ปี ของเครื่อง
output คือ วัน เดือน ปี ของเครื่องในรูปแบบข้อความ
(char *strdate) ดังนี้ dd-mm-yyyy
รูปแบบฟังก์ชัน `char *strdate(void)`
- (41) `strtime` ฟังก์ชันสำหรับเรียกดูเวลาของเครื่องในรูปแบบข้อความ
input คือ เวลาของเครื่องในขณะนั้น(time_t t)
output คือ เวลาของเครื่องในรูปแบบข้อความ
(char *strtime) ดังนี้ HH:MM:SS
รูปแบบฟังก์ชัน `char *strtime(time_t t)`

(42) `strctime` ฟังก์ชันสำหรับเปลี่ยนวินาทีเป็นข้อความในรูปแบบ

ชั่วโมง:นาที:วินาที

input คือ เวลาเป็นวินาที(int t)

output คือ ข้อความในรูปแบบ ชั่วโมง:นาที:วินาที

(char *strtime) ดังนี้ HH:MM:SS

รูปแบบฟังก์ชัน char *strctime(long t)

(43) `TAL_Error` ฟังก์ชันสำหรับแสดงความผิดพลาดที่เกิดขึ้นในระบบ

input คือ รหัสความผิดพลาดที่เกิดขึ้น(int codeno)

output คือ ข้อความแสดงความผิดพลาด ประกอบด้วย
รหัสความผิดพลาด, บรรทัดที่เกิดความผิดพลาด,
หมายเลขกรอบที่เกิดความผิดพลาด

ฟังก์ชันที่ได้รับการออกแบบสำหรับใช้งานในส่วนของ โปรแกรม ANALYSER.EXE

มีดังนี้

- (1) void Pause(void)
- (2) void Title(void)
- (3) void GetIdCourse(void)
- (4) void CreateFileName(void)
- (5) void ShowStatus(void)
- (6) void CreateAndOpenRecordFileName(void)
- (7) void EndTitle(void)
- (8) char *strtime(time_t t)
- (9) char *strctime(long t)

ความหมาย วิธีการทำงาน ข้อมูลเข้า ผลลัพธ์และรูปแบบฟังก์ชัน ของแต่ละฟังก์ชัน

มีดังนี้

- (1) Pause ฟังก์ชันสำหรับหยุดรอรับการเคาะแป้นพิมพ์จากผู้ใช้โปรแกรม
ถ้าเคาะแป้น ESC จะออกจากการใช้งานโปรแกรม
ส่วนแป้นอื่น ๆ จะดำเนินโปรแกรมต่อไป
รูปแบบฟังก์ชัน void Pause(void)

- (2) Title ฟังก์ชันสำหรับแสดงข้อความส่วนนำเข้าสู่ระบบ
input คือ -
output คือ ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบ
รูปแบบฟังก์ชัน void Title(void)

ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบของฟังก์ชัน Title มีดังนี้

```

+-----+
| +-----+ |
| |              ยินดีต้อนรับ          | |
| |              | |
| |              | |
| |  ผู้ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน' | |
| +-----+ |
+-----+

```

โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต

มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536

ที่ปรึกษาโครงการ : อาจารย์วุฒิพงศ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา

พัฒนาระบบโดย : นายนิพนธ์ ศุภศรี

- (3) GetIdCourse ฟังก์ชันสำหรับรับการป้อนรหัสผู้เรียนและชื่อบทเรียน
input คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
output คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
รูปแบบฟังก์ชัน void GetIdCourse(void)

โดยมีหน้าจอการรับข้อมูล ดังนี้

ระบบวิเคราะห์ผลการเรียนการสอน 'ผู้สอน'

=====
 โปรดป้อนชื่อบทเรียน[SSSS###] :

โปรดป้อนรหัสผู้เรียน[XXXX] :

(4) CreateFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มสรุปผลการเรียน

อ่านข้อมูลในแฟ้มสรุปผลการเรียนมา

เก็บในตารางข้อมูลประวัติการเรียน(HistoryTBL)

input คือ รหัสผู้เรียนและรหัสบทเรียนจากฟังก์ชัน GetIdCourse

และข้อมูลในแฟ้มสรุปผลการเรียน

output คือ ชื่อแฟ้มสรุปผลการเรียน (StatusFile= SSSIII.STS)

ตารางข้อมูล HistoryTBL

รูปแบบฟังก์ชัน void CreateFileName(void)

(5) ShowStatus ฟังก์ชันสำหรับแสดงประวัติการเรียน(HistoryTBL)

เพื่อให้เลือกหัวข้อที่จะเข้าไปดูพฤติกรรมการเรียน

input คือ ตารางข้อมูลประวัติการเรียน(HistoryTBL)

output คือ ลำดับหัวข้อที่จะเข้าเรียน (ItmPtr)

รูปแบบฟังก์ชัน void ShowStatus(void)

จอกภาพแสดงประวัติการเรียนรู้เพื่อให้เลือกหัวข้อที่จะเข้าเรียน เป็นต้น

สรุปผลการเรียนตามลำดับการเข้าเรียนจำนวน ครั้ง

วิชา..... ของนักเรียนรหัส มีดังนี้

ครั้งที่ หัวข้อที่ ชื่อเพิ่ม เรื่อง

เรียนเมื่อ ..-..-..-(...:....) รวมใช้เวลา วินาที

ตอบถูก ... ข้อ ผิด ... ข้อ ไม่ถูกไม่ผิด ... ข้อ คะแนนเต็ม ... คะแนน

ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน ...

ครั้งที่ หัวข้อที่ ชื่อเพิ่ม เรื่อง

เรียนเมื่อ ..-..-..-(...:....) รวมใช้เวลา วินาที

ตอบถูก ... ข้อ ผิด ... ข้อ ไม่ถูกไม่ผิด ... ข้อ คะแนนเต็ม ... คะแนน

ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน ...

ครั้งที่ หัวข้อที่ ชื่อเพิ่ม เรื่อง

เรียนเมื่อ ..-..-..-(...:....) รวมใช้เวลา วินาที

ตอบถูก ... ข้อ ผิด ... ข้อ ไม่ถูกไม่ผิด ... ข้อ คะแนนเต็ม ... คะแนน

ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน ...

กตแก่นไต่ ๆ เพื่อดูลำดับการเข้าเรียนครั้งต่อไป

เมื่อจบการรายงานสรุปผลการเรียนตามลำดับการเข้าเรียนแล้ว จะปรากฏข้อความให้ผู้สอนป้อนหมายเลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการเรียน หรือหมายเลข 999 เพื่อเลิกการใช้โปรแกรม ดังนี้

โปรดใส่เลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการเรียน(1-X)

หรือ กด 999 เลิกการใช้งานโปรแกรม :

(6) CreateAndOpenRecordFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มบันทึก

พฤติกรรมกรรการเรียน และเรียกข้อมูลในแฟ้มมาแสดง

input คือ ข้อมูลจากการเลือกหัวข้อที่ต้องการเรียน

output คือ ชื่อแฟ้มบันทึกพฤติกรรมกรรการเรียน (SSSIII.STS)

และการแสดงข้อมูลในแฟ้ม

รูปแบบฟังก์ชัน void CreateAndOpenRecordFileName(void)

จอภาพแสดงข้อมูลในแฟ้มพฤติกรรมกรรการเรียนในแต่ละกรอบกรรการเรียน เป็นดังนี้

พฤติกรรมกรรการเรียนตามลำดับกรรการเข้าใช้กรรการเรียน

วิชา ของนักกรรการเรียนรหัส....

หัวข้อที่... เรื่อง..... มีดังนี้

=====

ลำดับเฟรมที่ ...

หมายเลขเฟรม:... ชนิด:... จำนวนเฟรมคำถามสะสม:...

จำนวนครั้ง ที่ยอมให้เข้าใช้:... ที่เคยเข้าใช้:...

เวลา

จำนวนเวลาสะสมที่ผ่านมาในบทเรียน:... วินาที

เวลาเริ่มเข้าใช้เฟรม :.....

จำนวนเวลาทั้งหมดที่ใช้ในเฟรม :.... วินาที

จำนวนเวลาที่กำหนดให้แสดงข้อความในเฟรม:...วินาที ใช้จริง:...วินาที

จำนวนเวลาที่กำหนดให้ตอบคำถามในเฟรม:...วินาที ใช้จริง:...วินาที

จำนวนเวลาที่กำหนดให้แสดงข้อความโต้ตอบในเฟรม:...วินาที ใช้จริง:...วินาที

คำตอบของผู้เรียน :.....

ผลลัพธ์จากคำตอบ :...

หมายเลขเฟรมต่อไป :...

จำนวนข้อตอบ ถูกสะสม:... ผิดสะสม:... ไม่มีถูกผิดสะสม :...

คะแนน

ทำได้ในเฟรมนี้:... ได้สะสม:... เต็มสะสม:... เต็มของเฟรมนี้:...

<<<< เคาะแบ้นไต ๆ เพื่อดูพฤติกรรมในเฟรมถัดไป >>>>

ซึ่งสามารถนำมาใช้เป็นส่วนหนึ่งของเอกสารประกอบได้ ดังเอกสารรายการโปรแกรมในภาคผนวก ก.

4.5.4 ทดสอบและแก้ไขโปรแกรม

ผู้วิจัยได้นำโปรแกรมที่ลงรหัสด้วยภาษา C แล้วเข้าคอมพิวเตอร์ ทำการตรวจสอบความถูกต้องของรูปแบบและกฎเกณฑ์ในการใช้ภาษา C โดยใช้ตัวแปลชุดคำสั่งของโปรแกรม Turbo C⁺⁺ จนกระทั่งไม่พบข้อผิดพลาด และตรวจสอบผลการประมวลผลของโปรแกรมโดยสมมุติข้อมูลตัวแทน (test data) ตรวจสอบความถูกต้องของผลลัพธ์ในฟังก์ชันต่าง ๆ และตรวจสอบการทำงานทั้งระบบโดยดำเนินการออกแบบบทเรียนโปรแกรมเรื่องการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ขึ้นมาทดสอบ นำข้อมูลที่ได้จากการทดสอบมาทำการปรับปรุงแก้ไขการประมวลผลของระบบโปรแกรม จนกระทั่งสามารถทำงานได้ตามวัตถุประสงค์ที่วางไว้

4.5.5 จัดทำเอกสารประกอบโปรแกรม

ในงานวิจัยนี้ ผู้วิจัยได้จัดทำเอกสารประกอบโปรแกรม ซึ่งเป็นงานสำคัญของการพัฒนาระบบ เอกสารนี้คือ คู่มือการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ เพื่อช่วยให้ผู้ใช้งาน TEACH เข้าใจวัตถุประสงค์ทราบถึงประเภท ชนิด และอุปกรณ์ของคอมพิวเตอร์ที่ใช้ แนวคิดเกี่ยวกับการออกแบบโปรแกรม ภาษา TAL และโครงสร้างของภาษา วิธีการเขียนบทเรียนโปรแกรม วิธีการใช้โปรแกรม รายละเอียดของระบบ ตัวอย่างการใช้งาน และผลลัพธ์ที่ได้ ดังเอกสารคู่มือการใช้ TEACH ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ในภาคผนวก ง.

4.6 สร้างตัวอย่างบทเรียนโปรแกรมเรื่อง การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ผู้วิจัยได้ใช้เอกสารคู่มือการใช้ TEACH เป็นเนื้อหาหลักในการออกแบบสร้างบทเรียนโปรแกรม โดยได้จัดแบ่งบทเรียนออกเป็น 6 หัวข้อ ซึ่งสามารถจัดเป็น 6 หน่วยเนื้อหาวิชา ดังนี้

4.6.1 คำนำ จุดมุ่งหมาย ลักษณะ ประโยชน์ และอุปกรณ์ที่จำเป็นของ TEACH

4.6.2 การใช้งานระบบ (การสร้างบทเรียน)

4.6.3 การใช้งานระบบ (การเรียนและการติดตามผล)

4.6.4 ภาษา TAL (TEACH Authoring Language)

4.6.5 ตัวอย่างการสร้างบทเรียนเพื่อใช้กับระบบ TEACH

4.6.6 สถิติการทำงานของบทเรียนตัวอย่างที่สร้างในหัวข้อที่ 5

จากนั้นจึงได้จัดทำเพิ่มโครงสร้างบทเรียนและเพิ่มเนื้อหาวิชาทั้ง 6 เพิ่มขึ้น และทำการทดสอบใช้งาน ปรับปรุงแก้ไขจนสามารถใช้งานได้ตามต้องการ รายละเอียดของตัวอย่างบทเรียนโปรแกรม การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ ดังเอกสารในภาคผนวก จ.

4.7 ประเมินผลการใช้งานโปรแกรม

ผู้วิจัยได้นำระบบ TEACH ไปทดสอบประยุกต์ใช้งานจริงกับ

กลุ่มตัวอย่าง ซึ่งเป็นนักศึกษาชั้นปีที่ 4 ภาควิชาครุศาสตร์และเทคโนโลยี คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้า ธนบุรี จำนวน 28 คน แบ่งเป็น 14 กลุ่ม กลุ่มละ 2 คน โดยกลุ่มตัวอย่างมีคุณสมบัติ ดังนี้

1. มีความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์ และภาษาคอมพิวเตอร์
2. สามารถใช้โปรแกรมประมวลคำได้
3. สามารถสร้าง courseware ได้
4. มีความต้องการและจำเป็นต้องที่จะพัฒนาบทเรียนโปรแกรมเพื่อใช้งานต่อไป

ขั้นตอนของการทดสอบ ผู้วิจัยได้ทำการทดสอบเมื่อวันที่ 2 กุมภาพันธ์ 2537 โดยเริ่มต้นผู้วิจัยได้แนะนำตนเองและแนะนำเกี่ยวกับการใช้ระบบ TEACH เป็นเวลา 1 ชั่วโมง หลังจากนั้นได้แจกคู่มือการใช้ TEACH : ระบบสร้างบทเรียนบนไมโครคอมพิวเตอร์ด้วยตนเอง ให้กลุ่มตัวอย่างไปศึกษาเพิ่มเติม พร้อมทั้งให้โปรแกรม TEACH.EXE และโปรแกรม ANALYSER.EXE ไปใช้สร้างบทเรียน แล้วให้กลุ่มตัวอย่างได้นำเครื่องมือไปทำการทดสอบสร้างบทเรียน เรื่อง เทคนิคการฝึก อบรม ซึ่งเป็นบทเรียนที่จัดทำไว้เป็นชุดการสอน (module) แล้ว

ภายหลังจากกลุ่มตัวอย่างได้ทดลองใช้โปรแกรม TEACH สร้างบทเรียนแล้ว ผู้วิจัยให้ตอบแบบประเมินผลการใช้โปรแกรม และสามารถสรุปผลความเห็นเกี่ยวกับการใช้โปรแกรม TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ ดังเอกสารในภาคผนวก ฉ.

บทที่ 3

ผล

งานวิจัยและพัฒนานี้ เป็นการพัฒนาระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ซึ่งให้ชื่อว่า TEACH โดยระบบมีความต้องการอุปกรณ์ขั้นต่ำเพียงใช้คอมพิวเตอร์ 16 บิต หน่วยความจำไม่น้อยกว่า 640 KB จอภาพโมโนโครม พัฒนาโดยใช้ภาษา C และเป็นระบบที่แยกเนื้อหาบทเรียนและโปรแกรมควบคุมออกจากกัน ผู้สอนสามารถใช้สร้างบทเรียนบนไมโครคอมพิวเตอร์ได้ด้วยตนเอง โดยระบบนี้ผู้สอนไม่จำเป็นต้องมีความสามารถในการเขียนโปรแกรม เพียงแต่จัดทำบทเรียนซึ่งพิมพ์โดยใช้โปรแกรมประมวลคำที่มีเผยแพร่ทั่วไป บทเรียนเรื่องหนึ่ง ๆ สำหรับระบบนี้จะประกอบด้วย หนึ่งแฟ้มโครงสร้างบทเรียนซึ่งใช้บอกว่บทเรียนชื่ออะไรประกอบด้วยหัวข้อใดชื่ออะไร และแฟ้มเนื้อหาวิชาจำนวนเท่ากับที่ระบุในแฟ้มโครงสร้างบทเรียน แฟ้มนี้เป็นข้อมูลเนื้อหาบทเรียนที่ประกอบด้วยคำสั่งภาษา TAL ซึ่งเป็นภาษาที่ผู้วิจัยได้พัฒนาขึ้นเพื่อใช้ในการเขียนบทเรียนโปรแกรมสำหรับระบบโปรแกรม TEACH โดยเฉพาะ ซึ่งจะช่วยจัดลำดับการนำเสนอบทเรียนและใส่เทคนิคการสอนในเนื้อหาบทเรียน

ในการเรียน ผู้เรียนสามารถเข้าเรียนโดยการเรียกโปรแกรม TEACH.EXE มาทำงาน โปรแกรม TEACH.EXE จะให้ผู้เรียนป้อนรหัสวิชาที่ต้องการเรียน และป้อนรหัสผู้เรียน จากนั้นโปรแกรมจะเสนอหัวข้อที่มีในบทเรียนและให้ผู้เรียนเลือก เมื่อผู้เรียนเลือกหัวข้อที่ต้องการ หรือถ้าไม่เลือกโปรแกรมจะจัดให้เข้าเรียนในหัวข้อที่ต่อเนื่องจากการเรียนครั้งที่ผ่านมา แล้วทำหน้าที่นำเสนอบทเรียนตามหัวข้อที่ระบุ โดยสร้างตารางควบคุมรอบการเรียนต่าง ๆ ควบคุมลำดับขั้นตอนการนำเสนอรอบการเรียน โดยการสุ่มรอบการเรียนมาใช้งาน หรือการกำหนดเงื่อนไขที่สัมพันธ์กับการตอบคำถามของผู้เรียน การแสดงผลบนจอภาพ การแสดงข้อความโต้ตอบให้แรงจูงใจแก่ผู้เรียนซึ่งสามารถเรียกมาจากแฟ้มข้อมูลโต้ตอบได้ บันทึกพฤติกรรมการเรียนของผู้เรียน และเมื่อยุติการเรียนหัวข้อใด ๆ โปรแกรมจะประเมินผลการเรียนของผู้เรียนตามเกณฑ์การประเมินที่ผู้สอนกำหนดไว้ในบทเรียน แสดงผลการเรียนให้ผู้เรียนทราบ และบันทึกสรุปผลการเรียน

การเรียกดูบันทึกสรุปผลการเรียน และบันทึกพฤติกรรมการเรียนของผู้เรียน ซึ่งเป็นข้อมูลที่เป็นประโยชน์ต่อผู้สอนในการให้คำปรึกษาแนะนำแก่ผู้เรียน และทำการประเมินผลบทเรียนเพื่อพิจารณาปรับปรุงบทเรียน ระบบ TEACH มีโปรแกรม ANALYSER.EXE ทำหน้าที่ช่วยอำนวยความสะดวกในส่วนนี้ โดยผู้สอนเรียกโปรแกรม ANALYSER.EXE ขึ้นมาทำงาน แล้วป้อนข้อมูลรหัสวิชาและข้อมูลรหัสผู้เรียนที่ต้องการเรียกดูข้อมูล โปรแกรมจะนำเสนอสรุปผลการเรียนตามลำดับการเข้าเรียนทุกครั้ง และให้ผู้สอนเลือกหัวข้อบทเรียนที่จะเรียกดูพฤติกรรมการเรียน จากนั้น โปรแกรมจะนำพฤติกรรมการเรียนในแต่ละกรอบการเรียนของหัวข้อที่เลือกมานำเสนอแจกแจงเพิ่มข้อมูล และ โปรแกรมจะกลับเข้ามาเสนอสรุปผลการเรียนให้ผู้สอนเลือกหัวข้ออื่นต่อไป ถ้าไม่ต้องการเรียกดูอีก ผู้สอนก็สามารถเลือกการเลิกใช้งานโปรแกรม ANALYSER.EXE ได้

นอกจากนี้ผู้วิจัยยังได้จัดทำเอกสารคู่มือการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ และจัดทำตัวอย่างบทเรียน โปรแกรมที่ใช้งานบนระบบ TEACH เรื่อง การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์ สำหรับเป็นคู่มือแนะนำการใช้งานระบบ TEACH และเป็นข้อมูลเผยแพร่ให้ผู้สนใจใช้ศึกษาเพื่อพัฒนาระบบ TEACH ต่อไป

ในการวิจัยครั้งนี้ ผู้วิจัยได้นำระบบ TEACH ไปทดสอบกับกลุ่มตัวอย่าง ซึ่งเป็นนักศึกษาชั้นปีที่ 4 ภาควิชาครุศาสตร์และเทคโนโลยี คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้า ธนบุรี จำนวน 28 คน จากการประเมินผลด้วยแบบสอบถามกลุ่มตัวอย่างส่วนใหญ่เห็นว่า ในภาพรวมแล้ว ความสามารถของระบบ TEACH อยู่ในระดับปานกลาง ซึ่งสามารถใช้งานได้ระดับหนึ่ง แต่อย่างไรก็ตามระบบ TEACH นี้ยังคงต้องพัฒนาต่อไปอีก เพื่อให้มีประสิทธิภาพและขีดความสามารถในการใช้งานเพิ่มขึ้นในอนาคต

บทที่ 4

วิจารณ์และสรุป

การวิจัยและพัฒนาครั้งนี้ ทำให้ได้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบน ไมโครคอมพิวเตอร์ ซึ่งสามารถใช้งานได้ตามวัตถุประสงค์ที่กำหนด และได้จัดทำเอกสารคู่มือ ตัวอย่างบทเรียน โปรแกรมเรื่อง การใช้ TEACH ให้ผู้สนใจสามารถนำระบบไปประยุกต์ใช้งานได้อย่างสะดวกขึ้น

ผลการวิจัยที่ได้ทำให้สามารถสรุปผลข้อดีและข้อด้อยของระบบ และข้อเสนอแนะในการพัฒนาระบบ TEACH ในอนาคต ซึ่งจะเป็นประโยชน์ต่อการพัฒนาระบบ และการนำระบบไปประยุกต์ใช้ในอนาคต ดังนี้

1. ข้อดีของระบบ TEACH คือ

1.1 ระบบซอฟต์แวร์กับบทเรียน โปรแกรมแยกออกจากกัน ทำให้การพัฒนา บทเรียน โปรแกรมเป็นอิสระ ผู้ผลิตบทเรียน โปรแกรมที่ใช้ระบบนี้ จะสามารถแลกเปลี่ยน บทเรียน โปรแกรมกันได้ หากมีการจัดตั้งกลุ่มบทเรียน โปรแกรม TEACH

1.2 การสร้างบทเรียน โปรแกรมสามารถทำได้โดยใช้โปรแกรมประมวลคำทั่วไป ไม่ยึดติดกับระบบภาษาไทย จึงสะดวกในการพัฒนาบทเรียน โปรแกรมขึ้นใช้งาน

1.3 ภาษา TAL เป็นภาษาที่มีรูปแบบโครงสร้างในลักษณะกรอบการเรียนรู้ ทำให้มองเห็นการจัดแยกกรอบการเรียนรู้อย่างเด่นชัด สะดวกแก่การจัดสร้างบทเรียน ประกอบด้วยคำสั่งไม่มาก ไม่ซับซ้อน ง่ายแก่การเรียนรู้ สามารถเรียนรู้เพื่อใช้งานได้โดยอัตโนมัติ และผู้อำนวยความสะดวกในการจัดแบ่งเนื้อหาวิชาออกเป็นส่วนย่อยๆ

1.4 TEACH เป็นระบบโปรแกรมสร้างบทเรียนด้วยตนเองบน ไมโครคอมพิวเตอร์ ชนิดกรอบการเรียนรู้ ทำให้สะดวกในการออกแบบบทเรียน โปรแกรม เพราะสามารถจัดแบ่งเนื้อหาออกเป็นส่วนย่อย ๆ ง่ายต่อการควบคุม และกำหนดทิศทางการสอน

1.5 เพิ่มบทเรียนได้จัดแบ่งเป็น 2 กลุ่ม ที่มีความสัมพันธ์เกี่ยวโยงกันคือ เพิ่มโครงสร้างบทเรียนและเพิ่มเนื้อหาวิชา จะทำให้การออกแบบบทเรียน โปรแกรมจัดทำเป็นมอดูล

ได้อย่างสะดวก เป็นผลให้สามารถแยกย่อยสร้างบทเรียนเป็นส่วน ๆ ได้มากเท่าที่ต้องการ

1.6 มีระบบการควบคุมและติดตามการเข้าใช้บทเรียนของผู้เรียน ที่ทำให้สามารถทราบถึงพฤติกรรมกรเรียนของผู้เรียนในขณะที่เรียนได้ แตกต่างจากระบบทั่วไปที่เมื่อเรียนแล้วจะได้แต่ผลการเรียน ส่วนนี้เป็นประโยชน์ต่อผู้สอนซึ่งแม้จะไม่ได้เห็นผู้เรียนในขณะที่เรียนก็ยังสามารถติดตามการเรียนของผู้เรียนได้ ทำให้สามารถให้คำแนะนำแก่ผู้เรียนได้ หรือสามารถตรวจสอบบทเรียนว่ามีส่วนใดควรทำการปรับปรุง

1.7 ระบบ TEACH พัฒนาโดยใช้ภาษา C ซึ่งเป็นภาษาที่มีคุณสมบัติที่ดีในการพัฒนาโปรแกรม การพัฒนาได้แบ่งแยกย่อยระบบงานออกเป็นส่วน ๆ ทำให้สะดวกและง่ายต่อการพัฒนาปรับปรุงให้มีขีดความสามารถในด้านสี กราฟิกหรือขยายขีดความสามารถในด้านอื่น ๆ ในอนาคตต่อไป

1.8 ในด้านเทคนิคการสอน ภาษา TAL ของระบบสามารถอำนวยความสะดวกให้แก่ผู้สร้างบทเรียน ไม่ว่าจะต้องการสอนแบบเส้นตรง แบบแตกแขนง หรือแบบสุ่มเลือกได้อย่างสะดวก

1.9 การประยุกต์สำหรับใช้ระบบในการทดสอบผู้เรียน หรือใช้ในการทดสอบเพื่อเก็บคะแนนก็ยังสามารถกำหนดเทคนิคการสุ่มเลือกข้อสอบ (กรอบคำถาม) ได้อย่างสะดวก ทำให้บทเรียนหรือบททดสอบไม่ซ้ำซาก สามารถพลิกแฉ่งไปตามการใช้งานของผู้ใช้ และแบบทดสอบสามารถประยุกต์เขียนได้ในรูปแบบเลือกถูก ผิด ตัวเลือก หรือเติมคำก็ได้

2. ข้อดีของระบบ TEACH คือ

2.1 TEACH เป็นระบบชนิดข้อความ ไม่สามารถแสดงภาพนิ่ง ภาพเคลื่อนไหว การจำลองสถานการณ์ จึงทำให้การใช้งานมีขอบเขตจำกัด

2.2 TEACH ไม่มีระบบภาษาไทยเอง ในการใช้งานบางครั้งก่อให้เกิดปัญหาในเครื่องไมโครคอมพิวเตอร์ต่างชนิดกัน

2.3 เป็นระบบใช้กับผู้เรียนคนเดียว

3. ข้อเสนอแนะในการพัฒนาระบบ TEACH ต่อไปในอนาคต

3.1 พัฒนาเป็นระบบ LAN หรือระบบห้องเรียน

3.2 พัฒนาระบบ สี กราฟิก รูปภาพ animation simulation เสียงประกอบ โดยควรสร้างระบบดึงข้อมูลจากระบบอื่น ๆ เข้ามาผสมผสาน (โปรแกรมอื่นเข้ามาทำงาน

ในระบบ) โดยให้ถือว่าเป็นเพียงส่วนหนึ่งในคำสั่ง SCREEN เท่านั้น

3.3 พัฒนา editor เฉพาะสำหรับระบบ TEACH ในลักษณะของระบบสร้างบทเรียนอัตโนมัติช่วยสร้างบทเรียน (code generator)

3.4 พัฒนาระบบวิเคราะห์การเรียนรู้

บรรณานุกรม

- กัลยา แม่นมิตร. 2534. "ระบบช่วยสร้างบทเรียน พี-เอส-เอ็ม", วิทยานิพนธ์
วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย
มหาวิทยาลัยสงขลานครินทร์.
- กิตติศักดิ์ เกิดประสง. 2533. "การพัฒนาระบบช่วยสร้างบทเรียนแคลคูลัสเบื้องต้น",
วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์.
- ครรชิต มาลัยวงศ์ และ วิชิต ภู่วัตร์. 2532. เทคนิคการออกแบบโปรแกรม.
กรุงเทพฯ : ซีเอ็ดดูเคชั่น จำกัด.
- จุมพล มณีไย. 2530. "การใช้ไมโครคอมพิวเตอร์ช่วยงานสอน", ข่าวสาร กฟผ.
17(กันยายน 2530), 36-39.
- ชนะ โสภารักษ์. 2531. ศัพท์ไมโครคอมพิวเตอร์. พิมพ์ครั้งที่ 5. กรุงเทพฯ
: อมรินทร์การพิมพ์.
- ทักษิณา สวานานนท์. 2529. "คอมพิวเตอร์ช่วยสอน", คอมพิวเตอร์รีวิว.
3(กันยายน 2529), 56-57.
- บุญเลิศ เอี่ยมทัศนาศิน ภู่วรรณ และสมนึก ศิริโต. 2531. โปรแกรมคอมพิวเตอร์ภาษาซี. พิมพ์ครั้งที่ 4. กรุงเทพฯ : ซีเอ็ดดูเคชั่น จำกัด.
- ประกาศิต ชำดีบุรุษ และ อาทิตย์ จิตต์จุฬานนท์. 2533. โครงสร้างข้อมูลและ
อัลกอริทึม. กรุงเทพฯ : ซีเอ็ดดูเคชั่น จำกัด.

- พิสนธ์ จงตระกูล. 2533. "กลยุทธ์ในการพัฒนาคอมพิวเตอร์ช่วยสอนของสถาบัน
การศึกษา", จุฬาลงกรณ์เวชสาร. 34(เมษายน 2533), 247-259.
- พิสนธ์ จงตระกูลและมณีรัตน์ จรุงเดชากุล. 2531. "บทเรียนคอมพิวเตอร์ช่วยสอน(ซีเอ
ไอ)ที่เหมาะสมสำหรับการศึกษาศาสตร์ ควรเป็นเช่นไร", จุฬาลงกรณ์เวช
สาร. 32(สิงหาคม 2531), 703-711.
- ไพรัตน์ สังข์สกุล และ เอลิม ดวงยี่หว่า. 2529. การโปรแกรมภาษาซี.
กรุงเทพฯ : โรงพิมพ์ควอวิทีพริ้นท์.
- ไพโรจน์ ตีรณานกุล. 2528. ไมโครคอมพิวเตอร์ประยุกต์ทางการศึกษา. กรุงเทพฯ
: ศูนย์สื่อเสริมกรุงเทพ
- ไพโรจน์ ตีรณานกุล. 2536. เอกสารประกอบการสอนวิชา ETD 405 Instruction
Packages. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี.
- ยีน ภู่วรรณ ชัยยงค์ วงศ์ชัยวัฒน์ และ ไพศาล สงวนหมู่. 2530. เทคโนโลยี
ไมโครคอมพิวเตอร์ 16 บิต. พิมพ์ครั้งที่ 2. กรุงเทพฯ : ซีเอ็ดดูเคชั่น จำกัด.
- ริชาร์ด เอส. วีเนอร์ และ ลูอิส เจ. ฟินแลน. ม.ป.ป. หลักการโปรแกรม
Object-Oriented และภาษา C⁺⁺. เรียบเรียงจาก An Introduction
to Object-Oriented Programming and C⁺⁺. โดย นพดล ตั้งควรร
รณวานิช และ สรรค์ เสกขุนทด. กรุงเทพฯ : ซีเอ็ดดูเคชั่น จำกัด.
- วสันต์ อติศัพท์. 2530. "คอมพิวเตอร์ช่วยสอน", วารสารศึกษาศาสตร์.
3(กุมภาพันธ์-พฤษภาคม 2530), 17-27.
- วสันต์ อติศัพท์. 2530. "คอมพิวเตอร์ช่วยสอน", วารสารศึกษาศาสตร์.
3(มิถุนายน-กันยายน 2530), 75-90.

ส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี, สถาบัน. กระทรวงศึกษาธิการ.

2533. หนังสือเรียนหลักการเขียนโปรแกรม คพ.ทป 014 ระดับมัธยมศึกษาตอนต้นและระดับมัธยมศึกษาตอนปลาย. กรุงเทพฯ : องค์การค้ำ
ครูสภาลาดพร้าว.

อลล, สตีเวนส์. ม.ป.ป. การพัฒนาระบบฐานข้อมูลด้วยภาษาซี. เรียบเรียง
จาก C Data Base Development. โดย อนุชิต สุเมธวิทย์ และ วีร
ตั้งมั่นภักดีพงศ์. กรุงเทพฯ : ซีเอ็ดยูเคชั่น จำกัด.

เฮอร์เบิร์ต ซิลด์. ม.ป.ป. การประยุกต์ใช้งานภาษาซี. เรียบเรียงจาก C
Power User's Guide. โดย ศิววัฒน์ ศีวะบวร พรชัย จักรธำรงค์
และ จีรศักดิ์ ชัยวิริยะกุล. กรุงเทพฯ : ซีเอ็ดยูเคชั่น จำกัด.

Ghezzi, Carlo and Jazayeri, Mehdi. 1982. Programming
Language Concepts. New York : John Wiley & Sons, Inc.

Harrington, Steven. 1988. Computer Graphics : A Programming
Approach. 2nd ed. New York : McGraw-Hill Book Co.

Horspool, R. N. 1986. C Programming in the Berkeley Unix
Environment. Ontario : Prentice-Hall Canada, Inc.

Johnsonbauch, Richard and Kalin, Martin. 1990. Applications
Programming in C. Quen zon City : Forefront Book Co, Inc.

Korites, B. J. 1983. Graphic Software for Microcomputers.
Duxbury : Kern Publication.

Miller, H. L. & Quilici, A. E. 1989. The Official Borland Turbo C Survival Guide. New York : John Wiley & sons.

Schildt, Herbert. 1989. Born to Code in C.
New York : McGraw Hill.

Stevens, Al. 1991. Teach Yourself...C⁺⁺. 2nd ed. Manila
: Global Publishing.

Turbo C 2.0 : Reference and User's Guide. 1988. USA :
Borland International.

ภาคผนวก

ส่วนนี้เป็นผลงานที่ได้จากการวิจัยพัฒนาระบบ TEACH ประกอบด้วยส่วนต่าง ๆ คือ

ภาคผนวก ก. รายการโปรแกรม TEACH : ระบบสร้างบทเรียนด้วยตนเองบน
ไมโครคอมพิวเตอร์

ภาคผนวก ข. ภาษา TAL (TEACH Authoring Language)

ภาคผนวก ค. แบบฟอร์มออกแบบกรอบการเรียน TEACH

ภาคผนวก ง. เอกสารคู่มือการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบน
ไมโครคอมพิวเตอร์

ภาคผนวก จ. ตัวอย่างบทเรียนโปรแกรม เรื่อง การใช้ TEACH : ระบบสร้าง
บทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ภาคผนวก ฉ. สรุปการประเมินผลการใช้ TEACH : ระบบสร้างบทเรียน
ด้วยตนเองบนไมโครคอมพิวเตอร์

ภาคผนวก ก.

รายการโปรแกรม

TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ส่วนนี้ประกอบด้วยรายการโปรแกรมต้นฉบับ (source program) ของ TEACH.EXE และ ANALYSER.EXE ซึ่งจัดเป็นสารบัญได้ดังนี้

สารบัญ

เรื่อง	หน้า
รายการโปรแกรมต้นฉบับของ TEACH.EXE	
TEACH.H.....	63
TEACH1.C.....	67
TEACH2.C.....	77
TEACH3.C.....	84
TEACH4.C.....	90
รายการโปรแกรมต้นฉบับของ ANALYSER.EXE	
ANALYSER.C	110

```

/*=====
06/Fab/94 18:30 TEAA.H
Authoring system for microcomputer
=====
#define ONLY 0 /*- ชนิดของบทเรียน แบบไม่มีโครงสร้าง -*/
#define MANY 1 /*- ชนิดของบทเรียน แบบมีโครงสร้าง ---*/
#define ESC 0x1b /*- แป้น escape -----*/
#define COLON 0x3a /*- เครื่องหมาย : -----*/
#define APOSTROPHE 0x27 /*- เครื่องหมาย ' -----*/
#define Quotation 0x22 /*- เครื่องหมาย " -----*/
#define BACKSPACE 0x08 /*- แป้นลบย้อนกลับ -----*/
#define RETURN 0x0d /*- แป้น ENTER -----*/
#define TRUE 1 /*- ค่าเมื่อเป็น จริง -----*/
#define FALSE 0 /*- ค่าเมื่อเป็น เท็จ -----*/
#define LEN 20 /*- ความยาวของชื่อแฟ้ม -----*/
#define ELAW "(A:80,B:70,C:60,D:50,E:0)" /*- เกณฑ์ทั่วไปในการประเมินผล --*/
/*-----
ส่วนการประกาศชื่อแฟ้มข้อมูล
-----*/
extern char FrameFile[LEN]; /* R-> SSSL00.TXT | */
extern char StrFile[LEN]; /* R-> SSS.STR | SSS = Subject code */
extern char RecordFile[LEN]; /* RW> SSSL00.III (R)! LL = Lesson no. */
extern char StatusFile[LEN]; /* W-> SSSIII.STS (R) | OO = Object no. */
/*extern char MessageFile[LEN]; R-> CAICOM.MSG | III = Student Id. */
extern FILE *fp2, *fp4;
/*-----
ส่วนการกำหนดโครงสร้างข้อมูลของแฟ้มข้อมูลและตารางข้อมูลต่าง ๆ
-----*/
extern struct StatusFile { char fname[13]; /* ชื่อแฟ้มหัวข้อ */
int itm_no; /* ลำดับที่หัวข้อ */
char obj_name[100]; /* ชื่อหัวข้อ */
char l_date[11]; /* วัน-เดือน-ปี ที่เข้าเรียน */
char s_time[9]; /* เวลาที่เริ่มใช้บทเรียน */
long ut_time; /* เวลาทั้งหมดที่ใช้ในการเรียน */
int itm_right; /* จำนวนข้อตอบถูก */
int itm_wrong; /* จำนวนข้อตอบผิด */
int itm_non; /* จำนวนข้อตอบไม่มีถูกผิด */
int fullscore; /* คะแนนเต็ม */
int score; /* คะแนนทั้งหมดที่ได้ */
char e_law[100]; /* เกณฑ์การประเมินผล */
};

```

```

char evaluate;      /* ผลการประเมิน */
                   /* T = บทเรียนมีปัญหา หยุดการเรียน */
                   /* S = ผู้เรียนหยุดเรียนกลางคัน */
                   /* ๑ล = ผลการประเมิน */
} StatusData,      /*- เพิ่มสรุปผลการเรียน ---*/
  HistoryTBL[10];  /*- ข้อมูลประวัติการเรียน --*/
extern struct recdata
{
int f_no;           /* หมายเลขเฟรม */
char f_type;       /* ชนิดเฟรม */
int f_qno;         /* จำนวนเฟรมคำถามสะสม */
int loop_c;        /* จำนวนครั้งที่ยอมให้เข้าใช้ */
int loop_u;        /* จำนวนครั้งที่เคยเข้าใช้ */
long ti_lu;        /* จำนวนเวลาสะสมที่ผ่านมาในบทเรียน */
long ti_f;         /* เวลาเริ่มเข้าใช้เฟรม */
long ti_fu;        /* จำนวนเวลาทั้งหมดที่ใช้ในเฟรม */
long ti_fs;        /* จำนวนเวลาที่กำหนดให้สำหรับแสดงข้อความในเฟรม */
long ti_fsu;       /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความในเฟรม */
long ti_fa;        /* จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในเฟรม */
long ti_fau;       /* จำนวนเวลาที่ใช้ในการตอบคำถามในเฟรม */
long ti_fb;        /* จำนวนเวลาที่กำหนดให้ใช้แสดงข้อความโต้ตอบในเฟรม */
long ti_fbu;       /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความโต้ตอบในเฟรม */
char ans_s[80];    /* คำตอบของผู้เรียน */
char ans_r;        /* ผลลัพธ์จากคำตอบ */
int next_f;        /* 1-100 = หมายเลขเฟรมต่อไป */
                   /* 999 = จบการเรียน */
                   /* 1000 = เลิกเรียนการค้น */

int no_r;          /* จำนวนข้อสอบถูกสะสม */
int no_w;          /* จำนวนข้อสอบผิดสะสม */
int no_n;          /* จำนวนข้อสอบไม่มีถูกผิดสะสม */
int sc_g;          /* คะแนนที่ได้ในเฟรมนี้ */
int sc_u;          /* คะแนนสะสมที่ได้ */
int sc_t;          /* คะแนนรวมสะสม */
int sc_f;          /* คะแนนเต็มของเฟรมนี้ */
} record,          /*- เพิ่มบันทึกผลการเรียน -*/
  act;

extern struct date today; /*int da_mon, da_day, da_year */
extern struct time now;  /*int ti_hour, ti_min, ti_sec, ti_hund */
extern int FCTBL[50][5]; /* -->SSSLL00.TXT Frame Control Table */
extern char CDTBL[500][80]; /* -->SSSLL00.TXT Course Data Table */
extern char MDTBL[20][80]; /* -->CAICOM.MSG MessageData */

```

```

/*-----
ส่วนการประกาศตัวแปรทั่วไป
-----*/

extern char *Commands[];
extern int  CmdsLen[8];
extern char FrameType[3];
extern char EvaG[10];
extern int  EvaS[10];
extern int  LessonType, StudentType, Max;
extern int  LLptr;      /* ตัวชี้เลขบรรทัดบทเรียน -----*/
extern int  ActPtr;    /* ตัวชี้อักษรบนบรรทัดบทเรียน ----*/
extern char StrOut[100]; /* ผลลัพธ์ที่ได้จากการวิเคราะห์คำสั่ง -*/
extern int  LFptr;    /* ตัวชี้เฟรมปัจจุบัน -----*/
extern int  ItmPtr;   /* ตัวชี้ลำดับที่ของแฟ้มบทเรียนปัจจุบัน */
extern int  FLAGE;
extern char cmd[10], *tcmd;
extern time_t StartLearnTime, EndLearnTime ;
extern time_t StartFrameTime, EndFrameTime ;
/* current variable */
extern char uelaw[100]; /* เกณฑ์การประเมินผลที่ใช้จริงผล*/
extern char uevaluate; /* ผลการประเมินที่ผู้เรียนได้รับ S=หยุดเรียนกลางคัน
                        T=บทเรียนผิดพลาด ฯลฯ=ผลการเรียน */

/*-----
ส่วนการประกาศฟังก์ชันต้นแบบ
-----*/

extern void Pause(void);
extern int  WaitTime(char StrA[]);
extern long Wait(long Long);
extern long WaitBranch(long Long);
extern void Title(void);
extern void GetIdCourse(void);
extern void ReadStructureFile(void);
extern void ReadStatusFile(void);
extern void ShowStatus(void);
extern void CreateAndOpenRecordFileName(void);
extern void WriteDataToRecordFile(void);
extern void SaveStatus(void);
extern void LoadCourseAndMakeFrameTable(void);
extern void MessageFileLoading(void);

```

```
extern void FrameStatement(char StrA[]);
extern void ScreenStatement(char StrA[]);
extern int  InputStatement(char StrA[]);
extern void BranchStatement(char StrA[]);
extern void EvaluateStatement(char StrA[]);
extern char Grade(long x);
extern int  DecodeCommands(int s_cmds, int e_cmds, char Str[]);
extern int  Get1Parameter(char Str[], int ptr);
extern void ShowReport(void);
extern void EndingProcess(void);
extern void EndTitle(void);
extern void InitialValue(void);
extern int  CrossBlank(char Str[], int p);
extern void TeachingProcess(void);
extern int  RandomFrame(char StrA[]);
extern void DisplayText(void);
extern int  Display(char StrA[]);
extern int  SetCenter(char Lbl[]);
extern int  DecodeScore(char StrA[]);
extern void Branching(void);
extern int  colon(char StrA[]);
extern int  Message(char StrA[]);
extern char *strmid(char s[], unsigned p, unsigned n);
extern char *strcut(char s[], unsigned p);
extern int  strin(char s[], int p, char c);
extern char *strdate(void);
extern char *strtime(time_t t);
extern char *strctime(long t);
extern void TAL_Error(int codeno);
```

```

/*=====
16/Fab/94 18:00 TEACH1.C
Authoring system for microcomputer
=====

```

Programme by Nipon Suppasri. Semtember 4, 1990.

Master of Computer Science Thesis. Prince of Songkla University.

NOTE: This program is the tool for run computer lesson that write
in TEACH form.

ผู้สอน: ระบบโปรแกรมช่วยการเรียนการสอนแบบอัตโนมัติ พัฒนาระบบโดย นายนิพนธ์ ศุภศรี
โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2535
ที่ปรึกษาโครงการ: อาจารย์วุฒิพงษ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
รศ.ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา */

```

/*-----
ส่วนการ INCLUDE FILE
-----*/

```

```

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

```

```

/*-----
ส่วนการนิยามค่าคงที่ต่าง ๆ
-----*/

```

```

#define ONLY 0 /*- ชนิดของบทเรียน แบบไม่มีโครงสร้าง*/
#define MANY 1 /*- ชนิดของบทเรียน แบบมีโครงสร้าง -*/
#define ESC 0x1b /*- แป้น escape -----*/
#define COLON 0x3a /*- เครื่องหมาย : -----*/
#define APOSTROPHE 0x27 /*- เครื่องหมาย ' -----*/
#define Quotation 0x22 /*- เครื่องหมาย " -----*/
#define BACKSPACE 0x08 /*- แป้นลบย้อนกลับ -----*/
#define RETURN 0x0d /*- แป้น ENTER -----*/
#define TRUE 1 /*- ค่าเมื่อเป็น จริง -----*/
#define FALSE 0 /*- ค่าเมื่อเป็น เท็จ -----*/
#define LEN 20 /*- ความยาวของชื่อแฟ้ม -----*/
#define ELAW "(A:80,B:70,C:60,D:50,E:0)" /*- เกณฑ์ทั่วไปในการประเมินผล */

```

```

/*-----
ส่วนการประกาศชื่อแฟ้มข้อมูล
-----*/
char FrameFile[LEN];      /* R-> SSSL00.TXT      |
char StrFile[LEN];       /* R->      SSS.STR      | SSS = Subject code */
char RecordFile[LEN];    /* RW> SSSL00.III (R) | LL = Lesson no.  */
char StatusFile[LEN];    /* W-> SSSIII.STS (R) | OO = Object no.  */
/* char MessageFile[LEN];    R-> CAICOM.MSG      | III = Student Id. */
FILE *fp2, *fp4;
/*-----
ส่วนการกำหนดโครงสร้างข้อมูลของแฟ้มข้อมูลและตารางข้อมูลต่าง ๆ
-----*/
struct StatusFile { char fname[13];      /* ชื่อแฟ้มหัวข้อ */
  int itm_no;      /* ลำดับที่หัวข้อ */
  char obj_name[100]; /* ชื่อหัวข้อ */
  char l_date[11]; /* วัน-เดือน-ปี ที่เข้าเรียน */
  char s_time[9]; /* เวลาที่เริ่มใช้บทเรียน */
  long ut_time; /* เวลาทั้งหมดที่ใช้ในการเรียน */
  int itm_right; /* จำนวนข้อตอบถูก */
  int itm_wrong; /* จำนวนข้อตอบผิด */
  int itm_non; /* จำนวนข้อตอบไม่มีถูกผิด */
  int fullscore; /* คะแนนเต็ม */
  int score; /* คะแนนทั้งหมดที่ได้ */
  char e_law[100]; /* เกณฑ์การประเมินผล */
  char evaluate; /* ผลการประเมิน
                /* T = บทเรียนมีปัญหา หยุดการเรียน */
                /* S = ผู้เรียนหยุดเรียนกลางคัน */
                /* ฯลฯ= ผลการประเมิน */
} StatusData; /*- แฟ้มสรุปผลการเรียน ---*/
struct StatusFile HistoryTBL[10]; /*- ข้อมูลประวัติการเรียน --*/
struct recdata
{ int f_no; /* หมายเลขแฟรม */
  char f_type; /* ชนิดแฟรม */
  int f_qno; /* จำนวนแฟรมคำถามสะสม */
  int loop_c; /* จำนวนครั้งที่ยอมให้เข้าใช้ */
  int loop_u; /* จำนวนครั้งที่เคยเข้าใช้ */
  long ti_lu; /* จำนวนเวลาสะสมที่ผ่านมาในบทเรียน */
  long ti_f; /* เวลาเริ่มเข้าใช้แฟรม */
  long ti_fu; /* จำนวนเวลาทั้งหมดที่ใช้ในแฟรม */
  long ti_fs; /* จำนวนเวลาที่กำหนดให้สำหรับแสดงข้อความในแฟรม */
}

```

```

long ti_fsu;      /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความในเฟรม */
long ti_fa;      /* จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในเฟรม */
long ti_fau;     /* จำนวนเวลาที่ใช้ในการตอบคำถามในเฟรม */
long ti_fb;      /* จำนวนเวลาที่กำหนดให้ใช้แสดงข้อความโต้ตอบในเฟรม */
long ti_fbu;     /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความโต้ตอบในเฟรม */
char ans_s[80];  /* คำตอบของผู้เรียน */
char ans_r;      /* ผลลัพธ์จากคำตอบ */
int next_f;     /* 1-100 = หมายเลขเฟรมต่อไป */
                /* 999 = จบการเรียน */
                /* 1000 = เลิกเรียนการค้น */

int no_r;       /* จำนวนข้อสอบถูกสะสม */
int no_w;       /* จำนวนข้อสอบผิดสะสม */
int no_n;       /* จำนวนข้อสอบไม่มีถูกผิดสะสม */
int sc_g;       /* คะแนนที่ได้ในเฟรมนี้ */
int sc_u;       /* คะแนนสะสมที่ได้ */
int sc_t;       /* คะแนนรวมสะสม */
int sc_f;       /* คะแนนเต็มของเฟรมนี้ */
} record, /*- แฟ้มบันทึกผลการเรียน -*/
act;

struct date today; /*int da_mon, da_day, da_year */
struct time now; /*int ti_hour, ti_min, ti_sec, ti_hund */
int FCTBL[50][5]; /* -->SSSLOO.TXT Frame Control Table */
char CDTBL[500][80]; /* -->SSSLOO.TXT Course Data Table */
char MDTBL[20][80]; /* -->CAICOM.MSG MessageData */

/*-----
ส่วนการประกาศตัวแปรทั่วไป
-----*/

char *Commands[] = {"NON", "FRAME", "SCREEN", "INPUT", "BRANCH",
                   "EVALUATE", "$C", "$T" };
int CmdsLen[8] = {3, 5, 6, 5, 6, 8, 2, 2};
char FrameType[3]= {'E', 'Q', 'R'};
char EvaGE[10] = { ' ', 'A', 'B', 'C', 'D', 'E'};
int EvaS[10] = { 100, 80, 70, 60, 50, 0};
int LessonType, StudentType, Max;
int LLptr; /* ตัวชี้เลขบรรทัดบทเรียน -----*/
int ActPtr; /* ตัวชี้อักษรบนบรรทัดบทเรียน ----*/
char StrOut[80]; /* ผลลัพธ์ที่ได้จากการวิเคราะห์คำสั่ง -*/
int LFptr; /* ตัวชี้เฟรมปัจจุบัน -----*/
int ItmPtr; /* ตัวชี้ลำดับที่ของแฟ้มบทเรียนปัจจุบัน */

```



```

char cmd[9], *tcmd;

time_t StartLearnTime, EndLearnTime ;
time_t StartFrameTime, EndFrameTime ;
/* current variable */
char uelaw[100]; /* เลขที่การประเมินผลที่ใช้จริง*/
char uevaluate; /* ผลการประเมินที่ผู้เรียนได้รับ S=หยุดเรียนกลางคัน
                 T=บทเรียนผิดพลาด ฯลฯ=ผลการเรียน */

/*-----
ส่วนการประกาศฟังก์ชันต้นแบบ
-----*/

void Pause(void);
int WaitTime(char Str[]);
long Wait(long Long);
long WaitBranch(long Long);
void Title(void);
void GetIdCourse(void);
void ReadStructureFile(void);
void ReadStatusFile(void);
void ShowStatus(void);
void CreateAndOpenRecordFileName(void);
void WriteDataToRecordFile(void);
void SaveStatus(void);
void LoadCourseAndMakeFrameTable(void);
void MessageFileLoading(void);
void FrameStatement(char Str[]);
void ScreenStatement(char Str[]);
int InputStatement(char Str[]);
void BranchStatement(char Str[]);
void EvaluateStatement(char Str[]);
char Grade(int x);
int DecodeCommands(int s_cmds, int e_cmds, char Str[]);
int Get1Parameter(char Str[], int ptr);
void ShowReport(void);
void EndingProcess(void);
void EndTitle(void);
void InitialValue(void);
int CrossBlank(char Str[], int p);
void TeachingProcess(void);
int RandomFrame(char Str[]);

```

```

void DisplayText(void);
int  Display(char StrA[]);
int  SetCenter(char Lbl[]);
int  DecodeScore(char StrA[]);
void Branching(void);
int  colon(char StrA[]);
int  Message(char StrA[]);
char *strmid(char s[], unsigned p, unsigned n);
char *strcut(char s[], unsigned p);
int  strin(char s[], int p, char c);
char *strdate(void);
char *strtime(time_t t);
char *strotime(long t);
void TAL_Error(int codeno);
/*-----
ฟังก์ชันหลัก (main function)
-----*/

main ()
{ Title();
  GetIdCourse();
  while (TRUE)
  { ReadStructureFile();
    ReadStatusFile();
    ShowStatus(); /*EndTitle()*/
    CreateAndOpenRecordFileName();
    MessageFileLoading();      /*Load CAICOM.MSG */
    LoadCourseAndMakeFrameTable();
    TeachingProcess();
    EndingProcess(); /*{ WriteDataToRecordFile(); SaveStatus();
    fcloseall(); ShowReport();}*/
  } /* while */
}

/*-----
Title  ฟังก์ชันสำหรับแสดงข้อความส่วนนำเข้าสู่ระบบ
input  คือ -
output คือ ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบ
-----*/

void Title(void)
{ clrscr();
  gotoxy(23, 9); printf("+-----+\n");
}

```



```

{ gotoxy(1,13);clreol();
  gotoxy(1,14);clreol();
  gotoxy(15,13);printf(" โปรดป้อนรหัสผู้เรียน[XXX] :"); gets(StudentNo);
  if ((strlen(StudentNo) < 1) || (strlen(StudentNo) > 3))
    { gotoxy(30,14); printf("\007 \007 \007 รหัสผู้เรียนไม่ถูกต้อง");
      Wait(2);      count++;
      if (count > 2) TAL_Error(101);
    }
  else flag = 0 ;
}

/* สร้างชื่อแฟ้มโครงสร้างบทเรียน(text file) = SSS.STR */
strcat(StrFile,CourseName); strcat(StrFile, ".STR");
/* แฟ้มสรุปผลการเรียน(record file) = SSSIII.STS */
strcat(StatusFile,CourseName);
strcat(StatusFile,StudentNo);
strcat(StatusFile, ".STS");
}

/*-----
ReadStructureFile ฟังก์ชันสำหรับอ่านข้อมูลในแฟ้มโครงสร้างบทเรียนมาเก็บ
ในตารางข้อมูลประวัติการเรียน(HistoryTBL)
input คือ ข้อมูลในแฟ้มโครงสร้างบทเรียน
output คือ ตารางข้อมูลประวัติการเรียน(HistoryTBL)
-----*/

void ReadStructureFile(void)
{ char str[128], stra[12], fnme[12], strb[118];
  int p_stra, p_strb, p_str;
  int t_ndx, i, k ;
  FILE *fp1;
  if ( (fp1 = fopen(StrFile,"r")) == NULL ) LessonType = ONLY;
  else
    {LessonType = MANY; HistoryTBL[0].itm_no = 0;
      while (!feof(fp1))
        { fgets(str,128,fp1);
        if (ferror(fp1)) TAL_Error(210);/*การอ่านแฟ้มโครงสร้างบทเรียนมีปัญหา*/
        else
          { p_stra = p_strb = p_str = k = 0 ;
            for (;str[p_str] == ' '); p_str++;
            for (;!isdigit(str[p_str]);p_str++) fnme[k++]=str[p_str];
          }
        }
    }
}

```

```

    for (;str[p_str] != ' ');
{  stra[p_stra++] = str[p_str];  fnme[k++]=str[p_str++];}
    stra[p_stra] = fnme[k] = '\0';
    for (;str[p_str] == ' '); p_str++;
    for (;str[p_str] != '\n'); strb[p_strb++] = str[p_str++];
    strb[p_strb] = '\0';

    t_ndx = atoi(stra);
    i      = fmod(t_ndx,10);
    strcpy(HistoryTBL[i].fname      , fnme);
    strcat(HistoryTBL[i].fname      , ".txt");
    HistoryTBL[i].itm_no            = t_ndx;
    strcpy(HistoryTBL[i].obj_name   , strb);
    strcpy(HistoryTBL[i].l_date,"-----");
    strcpy(HistoryTBL[i].s_time,"--:--:--");
    HistoryTBL[i].ut_time = 0;
    HistoryTBL[i].itm_right = 0;
    HistoryTBL[i].itm_wrong = 0;
    HistoryTBL[i].itm_non = 0;
    HistoryTBL[i].fullscore = 0;
    HistoryTBL[i].score = 0;
    strcpy(HistoryTBL[i].e_law," ");
    HistoryTBL[i].evaluate = '-';
    if (i > HistoryTBL[0].itm_no) HistoryTBL[0].itm_no = i;
}
}
}
fclose(fp1);
}
/*-----
ReadStatusFile ฟังก์ชันสำหรับอ่านข้อมูลในแฟ้มสรุปผลการเรียนมาเก็บใน
ตารางข้อมูล HistoryTBL
input คือ ข้อมูลในแฟ้มสรุปผลการเรียน
output คือ ตารางข้อมูล HistoryTBL
ลำดับหัวข้อต่อเนื่องจากการเรียนครั้งก่อน (itmPtr)
-----*/

void ReadStatusFile(void)
{int i, StopItmNo, loop;
  int itm_no;
  /* อ่านข้อมูลในแฟ้ม */

```

```

if ((fp2 = fopen(StatusFile,"rb")) == NULL) /* ไม่เคยใช้บทเรียนมาก่อน */
{ if (LessonType == ONLY) /* NEW / ONLY */
  { ItmPtr = 1;
    HistoryTBL[0].itm_no = 0;    HistoryTBL[1].itm_no = 1;
    strcpy(HistoryTBL[0].fname, strmid(StatusFile,1,3));
    strcat(HistoryTBL[0].fname, ".txt");
    strcpy(HistoryTBL[1].fname, HistoryTBL[0].fname);
    strcpy(HistoryTBL[0].obj_name, strmid(StatusFile,1,3));
    strcpy(HistoryTBL[1].obj_name, strmid(StatusFile,1,3));
    printf("ONLY = %s\n", HistoryTBL[1].fname);
  }
  else ItmPtr = 1; /* NEW / MANY */
}
else /* เคยใช้บทเรียนมาแล้ว */
{ while(fread(&StatusData, sizeof StatusData, 1, fp2) == 1)
  /* อ่าน sts ไล่ His_TBL */
  { if (ferror(fp2)) TAL_Error(220);
    itm_no = StatusData.itm_no;
    strcpy(HistoryTBL[itm_no].fname , StatusData.fname) ;
    HistoryTBL[itm_no].itm_no = StatusData.itm_no ;
    strcpy(HistoryTBL[itm_no].obj_name , StatusData.obj_name) ;
    strcpy(HistoryTBL[itm_no].l_date , StatusData.l_date) ;
    strcpy(HistoryTBL[itm_no].s_time , StatusData.s_time ) ;
    HistoryTBL[itm_no].ut_time = StatusData.ut_time ;
    HistoryTBL[itm_no].itm_right = StatusData.itm_right ;
    HistoryTBL[itm_no].itm_wrong = StatusData.itm_wrong ;
    HistoryTBL[itm_no].itm_non = StatusData.itm_non ;
    HistoryTBL[itm_no].fullscore = StatusData.fullscore ;
    HistoryTBL[itm_no].score = StatusData.score ;
    HistoryTBL[itm_no].evaluate = StatusData.evaluate ;
  }
  /* set NextLearn */
  ItmPtr = itm_no;
  if ((HistoryTBL[itm_no].evaluate != '-')
    && (HistoryTBL[itm_no].evaluate != 'T')
    && (HistoryTBL[itm_no].evaluate != 'S'))
  { StopItmNo = HistoryTBL[0].itm_no + 1;
    if (itm_no == HistoryTBL[0].itm_no) {loop = 1; i = 1;}
    else { loop = 0; i = itm_no + 1;}
    for (; ((i < StopItmNo) && (loop < 2)); i = i+1)

```

```
    { if ((HistoryTBL[i].evaluate == '-') ||
      (HistoryTBL[i].evaluate == 'T') ||
      (HistoryTBL[i].evaluate == 'S'))
      {ItmPtr = i; i = StopItmNo + 1;}
    if (i == HistoryTBL[0].itm_no) {i = 0; loop = loop+1;}
    if (i == itm_no ) i = StopItmNo;
      }
      }
      }
    fclose (fp2);
  }
```

```

/*=====
06/Fab/94 18:00 TEACH2.C
Authoring system for microcomputer
=====

```

ส่วนการ INCLUDE FILE

```

-----*/
#include <dos.h>
#include <math.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "teaa.h"

```

```

/*-----
PAUSE ฟังก์ชันสำหรับหยุดรอรับการเคาะแป้นพิมพ์จากผู้ใช้โปรแกรม
ถ้าเคาะแป้น ESC จะออกจากการใช้งานโปรแกรม ส่วนแป้น
อื่น ๆ จะดำเนินโปรแกรมต่อไป
-----*/

```

```

void Pause(void)
{static char msg[] = "กดแป้น Esc เลิกใช้งาน หรือ กดแป้นใด ๆ ทำต่อไป ...";
int c;
gotoxy(20,23); printf("%s\n",msg);/* Put msg at bottom of screen*/
c = getch(); /* Read a character from kbd*/
if( ESC == c ) exit( 1 ); /* If user wish to return to OS */
if( 0 == c ) c = getch(); /* non-ASCII key scan for keyboard*/
}

```

```

/*-----
strmid ฟังก์ชันสำหรับคัดลอกข้อความ(s) จากตำแหน่งที่กำหนด(p)
ไปเท่ากับจำนวนตัวอักษร(n) ที่กำหนด
input คือ ข้อความ(char s[])
ตำแหน่งที่กำหนด(unsigned p)
จำนวนตัวอักษร(unsigned n)
output คือ ข้อความ(char *strmid)
-----*/

```

```

char *strmid(char s[],unsigned p,unsigned n)
{ unsigned i,j;

```



```

char s1[100];
  i = p-1;  j = 0;  s1[0] = '\0';
  if (p > strlen(s)) s1[0] = '\0';
  else { while ((i < p-1+n) && (s[i] != '\0')) s1[j++] = s[i++];
        s1[j] = '\0';}
  return(s1);
}

```

```

/*-----
strcut ฟังก์ชันสำหรับคัดลอกข้อความ(s) จากตำแหน่งที่กำหนด(p)
       ไปจนหมดข้อความ
input  คือ ข้อความ(char s[])
       ตำแหน่งที่กำหนด(unsigned p)
output คือ ข้อความ(char *strcut)
-----*/

```

```

char *strcut(char s[], unsigned p)
{ unsigned i, j;
  char s1[100];
  s1[0] = '\0';
  if (p > strlen(s)) s1[0] = '\0';
  else { i = p-1;  j = 0;
        while (s[i] != '\0') s1[j++] = s[i++];
        s1[j] = '\0';}
  return(s1);
}

```

```

/*-----
strin ฟังก์ชันสำหรับหาตำแหน่งของอักขระที่กำหนด(c) ตัวแรกที่พบบนข้อความ(s)
      โดยทำการค้นหาตั้งแต่ตำแหน่ง(p) ไปจนจบข้อความ
input  คือ ข้อความ(char s[])
       ตำแหน่งที่กำหนด(unsigned p)
       ตัวอักขระที่ต้องการค้นหา(char c)
output คือ ตำแหน่งของอักขระที่กำหนดตัวแรกที่พบบนข้อความ(int strin)
       ถ้าไม่พบ strin = 0
-----*/

```

```

int strin(char s[], int p, char c)
{ int i, n;
  n = p;
  if (p > strlen(s)) n = 0;
  else { i = p - 1;
        while ((s[i] != c) && (s[i] != '\0')) {n = n+1; i = i+1;}
        if (s[i] == '\0') n = 0;}
}

```

```

    return(n);
}
/*-----
strdate ฟังก์ชันสำหรับเรียกดูวัน เดือน ปี ของเครื่องในรูปแบบข้อความ
input   คือ วัน เดือน ปี ของเครื่อง
output  คือ วัน เดือน ปี ของเครื่องในรูปแบบข้อความ(char *strdate)
        dd-mm-yyyy
-----*/

char *strdate(void)
{struct date today;
char *s = "00-00-0000";
char month_[3], day_[3], year_[5];
  getdate(&today);
  itoa(today.da_mon,month_,10);    itoa(today.da_day,day_,10);
  itoa(today.da_year,year_,10);
  if (strlen(month_) < 2) { s[3] = '0';    s[4] = month_[0];}
  else                    { s[3] = month_[0]; s[4] = month_[1];}
  if (strlen(day_) < 2)   { s[0] = '0';    s[1] = day_[0];}
  else                    { s[0] = day_[0]; s[1] = day_[1];}
                          s[5] = '-';    s[2] = '-';
                          s[6] = year_[0]; s[7] = year_[1];
                          s[8] = year_[2]; s[9] = year_[3];
                          s[10] = '\0';

  return(s);
}
/*-----
strtime ฟังก์ชันสำหรับเรียกดูเวลาของเครื่องในรูปแบบข้อความ
input   คือ เวลาของเครื่องในขณะนี้(time_t t)
output  คือ เวลาของเครื่องในรูปแบบข้อความ(char *strtime)
        HH:MM:SS
-----*/

char *strtime(time_t t)
{ struct tm *sys_time;
char *s = "00:00:00" ;
char hour_[3], min_[3], sec_[3];
  sys_time = localtime(&t);
  itoa(sys_time->tm_hour, hour_,10);
  itoa(sys_time->tm_min , min_,10);
  itoa(sys_time->tm_sec , sec_,10);
  if (strlen(hour_) < 2) {s[0] = '0' ; s[1] = hour_[0];}

```

```

else                                     {s[0] = hour_[0]; s[1] = hour_[1];}
s[2] = ':';
if (strlen(min_) < 2) {s[3] = '0'      ; s[4] = min_[0];}
else                                     {s[3] = min_[0] ; s[4] = min_[1];}
s[5] = ':';
if (strlen(sec_) < 2) {s[6] = '0'      ; s[7] = sec_[0];}
else                                     {s[6] = sec_[0] ; s[7] = sec_[1];}
s[8] = '\0';
return(s);
}
/*-----
strctime ฟังก์ชันสำหรับเปลี่ยนวินาทีเป็นข้อความในรูปแบบ ชั่วโมง:นาที:วินาที
input   คือ เวลาเป็นวินาที(int t)
output  คือ ข้อความในรูปแบบ ชั่วโมง:นาที:วินาที(char *strtime)
        HH:MM:SS
-----*/
char *strctime(long t)
{ long h, m, s1;
  char *s = "00:00:00";
  char hour_[3], min_[3], sec_[3];
  { s1 = t-(t/60*60);
    m = (t/60)-(t/60/60*60);
    h = t/3600;
    ltoa(h, hour_,10);
    ltoa(m, min_,10);
    ltoa(s1, sec_,10);
    if (strlen(hour_) < 2) {s[0] = '0'      ; s[1] = hour_[0];}
    else                   {s[0] = hour_[0]; s[1] = hour_[1];}
    s[2] = ':';
    if (strlen(min_) < 2) {s[3] = '0'      ; s[4] = min_[0];}
    else                   {s[3] = min_[0] ; s[4] = min_[1];}
    s[5] = ':';
    if (strlen(sec_) < 2) {s[6] = '0'      ; s[7] = sec_[0];}
    else                   {s[6] = sec_[0] ; s[7] = sec_[1];}
    s[8] = '\0';
  }
  return(s);
}
/*-----
TAL_Error ฟังก์ชันสำหรับแสดงความผิดพลาดที่เกิดขึ้นในระบบ

```

input คือ รหัสความผิดพลาดที่เกิดขึ้น(int codeno)

output คือ ข้อความแสดงความผิดพลาด ประกอบด้วย
รหัสความผิดพลาด, บรรทัดที่เกิดความผิดพลาด,
เลขเฟรมที่เกิดความผิดพลาด

```
-----*/
void TAL_Error(int codeno)
{ /* ตำแหน่งแสดง ERROR */
  clrscr();
  gotoxy(15,16);
  printf("กรุณาติดต่อผู้สอนและแจ้งข้อความในบรรทัดข้างล่างให้ผู้สอนทราบ\n");
  /* ความผิดพลาดที่เกิดจากผู้เรียน กำหนดให้ช่วงรหัสความผิดพลาดอยู่ระหว่าง
  100 - 199 ความผิดพลาดนี้เกิดขึ้นก่อนการเข้าสู่การเรียน จึงไม่มีการบันทึก
  ข้อมูลใด ๆ ของผู้เรียน ให้ออกจากระบบได้ทันที */
  printf("\n");
  if (codeno < 200)
  { /* error id error lesson code */
    printf("E%d ",codeno);
    switch(codeno)
    { case 100 : printf("ใส่รหัสบทเรียนไม่ถูกต้อง 3 ครั้ง ไม่อนุญาตให้เข้าสู่ระบบ\n");
      break;
      case 101 : printf("ใส่รหัสผู้เรียนไม่ถูกต้อง 3 ครั้ง ไม่อนุญาตให้เข้าสู่ระบบ\n");
      break;
      case 102 : printf("ใส่เลขหัวข้อไม่ถูกต้อง 3 ครั้ง ไม่อนุญาตให้เข้าสู่ระบบ\n");
      break;
    }
  }
  else
  { printf("E%d F%d L%d ***",codeno,LFptr,LLptr);
    /* ความผิดพลาดที่เกิดจากการเขียนอ่านแฟ้มข้อมูล กำหนดให้ช่วงรหัสความ
    ผิดพลาดอยู่ระหว่าง 200 - 299 ความผิดพลาดนี้เกิดเกี่ยวกับแฟ้มข้อมูล
    จึงไม่สามารถบันทึกข้อมูลใด ๆ ของผู้เรียนได้ ให้ออกจากระบบได้ทันที */
    if (codeno < 300)
    { switch(codeno)
    { case 200 : break;
      case 201 : printf("เพิ่มเนื้อหาบทเรียนมีขนาดมากกว่า 500 บรรทัด\n");
      break;
      case 210 : printf("มีปัญหาขณะอ่านแฟ้มโครงสร้างบทเรียน\n");
      break;
      case 220 : printf("มีปัญหาขณะอ่านแฟ้มสรุปผลการเรียน(%s)\n",StatusFile);
      break;
    }
  }
  }
}
```

```

case 221 : printf("มีปัญหาลงเขียนข้อมูลลงแฟ้มสรุปผลการเรียน(%s)\n", StatusFile);
          break;
case 222 : printf("มีปัญหาลงเปิดแฟ้มสรุปผลการเรียน(%s) เพื่อบันทึกข้อมูล\n",
          StatusFile);
          break;
case 230 : printf("ไม่พบแฟ้มเนื้อหาวิชา(%s)\n", HistoryTBL[LFptr].fname);
          break;
case 231 : printf("มีปัญหาลงอ่านแฟ้มเนื้อหาวิชา(%s)\n", HistoryTBL[LFptr].fname);
          break;
case 240 : printf("มีปัญหาในการเปิดแฟ้มบันทึกการเรียน(%s)\n", StatusFile);
          break;
case 241 : printf("มีปัญหาลงเขียนแฟ้มบันทึกการเรียน(%s)\n", StatusFile);
          break;
case 250 : printf("มีปัญหาลงอ่านแฟ้มข้อความโต้ตอบ(CAICOM.MSG)\n");
          break;
}
}
else
{ /* ความผิดพลาดที่เกิดจากบทเรียน กำหนดให้ช่วงรหัสความผิดพลาดอยู่
ระหว่าง 300 - 399 ความผิดพลาดนี้เกิดเนื่องจากไวยากรณ์ของคำสั่ง
ในแฟ้มบทเรียนไม่ถูกต้อง และมักจะเกิดเมื่อมีการเรียนไปบ้างแล้ว
จึงให้บันทึกข้อมูลการเรียนของผู้เรียนก่อนออกจากระบบ */
uevaluate = 'T';
WriteDataToRecordFile();
SaveStatus();
fcloseall();
printf("\n");
switch(codeno)
{ case 300 : printf("\n");
          break;
case 301 : printf("หมายเลขเฟรมในการสุมไม่ปรากฏในบทเรียน\n");
          break;
case 310 : printf("หลังคำสั่ง SCREEN กำหนดตำแหน่งแสดงผลไม่ได้\n");
          break;
case 321 : printf("ไม่ระบุจำนวนเวลาในการคอยคำตอบ\n");
          break;
case 322 : printf("มีเครื่องหมาย , แต่ไม่มีคำสั่งตาม\n");
          break;
case 330 : printf("หมายเลขเฟรมไม่ถูกต้อง\n");
          break;

```

```
case 340 : printf("หมายเลขเฟรมไม่อยู่ในช่วงที่มี\n");
          break;
case 341 : printf("คำสั่งย่อยใน BRANCH ไม่ถูกต้อง\n");
          break;
case 342 : printf("คำสั่งย่อยใน BRANCH ไม่มีข้อความโต้ตอบที่กำหนด\n");
          break;
case 350 : printf("ไม่ได้กำหนดคะแนนเต็มหลัง V\n");
          break;
case 351 : printf("บรรทัดตามหลังคำสั่ง Branch ผิดไวยกรณ์\n");
          break;
case 352 : printf("มี , แต่ไม่มีข้อความโต้ตอบ ใน Branch ผิดไวยกรณ์\n");
          break;
case 510 : printf("คำสั่ง Evaluate ผิดไวยกรณ์\n");
          break;
}
}

}
exit(1);
}
```

```

/*=====
   06/Fab/94 18:00  TEACH3.C
   Authoring system for microcomputer
=====
/*-----
ส่วนการ INCLUDE FILE
-----*/

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include <process.h>
#include <float.h>
#include <sys\types.h>
#include <stddef.h>
#include "teaa.h"

/*-----
   ShowStatus ฟังก์ชันสำหรับแสดงประวัติการเรียน(HistoryTBL) เพื่อให้เลือก
   หัวข้อที่จะเข้าเรียน
   input คือตารางข้อมูลประวัติการเรียน(HistoryTBL)
   output คือ ลำดับหัวข้อที่จะเข้าเรียน (ItmPtr)
-----*/

void ShowStatus(void)
{int i, j, x;
  int flag;
  char InputItm[6];
  x = 0;
  clrscr();
  printf("คุณมีประวัติการศึกษา ดังนี้\n ");
  printf("วิชา %s\n ",HistoryTBL[0].obj_name);
  printf("|หัวข้อ|-----|เรียนเมื่อ-----|");
  printf(" ---รวม---|ผลการเรียน-|\n");
  printf(" |ที่-|-----เรื่อง-----|วัน-เดือน-ปี(เวลา)---|");
}

```

```

printf("-ใช้เวลา-|-คะแนนเกรด-|\n");

for(i = 1;i<=HistoryTBL[0].itm_no;i++)
{ printf(" %4d %-60s \n", HistoryTBL[i].itm_no,HistoryTBL[i].obj_name);
  printf("%40s(%8s) ",HistoryTBL[i].l_date, HistoryTBL[i].s_time);
  printf(" %8s %3d/%-3d %c \n", strctime(HistoryTBL[i].ut_time),
    HistoryTBL[i].score, HistoryTBL[i].fullscore,
    HistoryTBL[i].evaluate);
}
flag = 1;
while (flag == 1)
{gotoxy(1,18);
  printf("   โปรด ใส่เลขหัวข้อที่ต้องการศึกษาแล้วกด ENTER\n");
  printf("   หรือ กด ENTER เพื่อศึกษาหัวข้อที่ %d ต่อจากครั้งก่อน\n",ItmPtr);
  printf("   \n");
  gotoxy(1,20); printf("   หรือ พิมพ์ 999 แล้วกด ENTER เพื่อเลิกใช้งาน   \n'
  if (x >= 3) TAL_Error(102);
  x = x+1;
  gets(InputItm);
  if ((strlen(InputItm) == 0)) flag = 0;
  else if (strcmp(InputItm,"999") == 0)
    { printf("\n   โปรดกด Y เพื่อยืนยันว่าต้องการเลิกใช้งาน\n");
      if (toupper(getch()) == 'Y') EndTitle();
    }
  else
    if (strlen(InputItm) == 1)
      { j = InputItm[0];
        if (j - 48 <= HistoryTBL[0].itm_no)
          { ItmPtr = j-48; flag = 0;}
        else
          {gotoxy(30,22); printf("เลขที่ใส่ไม่ถูกต้อง\n"); delay(2000);
            gotoxy(1,22); clreol();}
        }
      }
    else
      { gotoxy(30,22); printf("เลขที่ใส่ไม่ถูกต้อง\n"); delay(2000);
        gotoxy(1,22); clreol();}
      }
}
}
/*-----
CreateAndOpenRecordFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มบันทึกการเรียน

```


และเปิดใช้งาน แฟ้มนี้จะสร้างเมื่อได้เลือก
แฟ้มบทเรียนแล้ว

input คือ ข้อมูลจากการเลือกหัวข้อที่ต้องการเรียน

output คือ ชื่อแฟ้มบันทึกพฤติกรรมการณ์การเรียน(RecordFile) และการเปิดแฟ้มใช้งาน

```
-----*/
void CreateAndOpenRecordFileName(void)
```

```
{int len;
```

```
strcpy(RecordFile,HistoryTBL[ItmPtr].fname);
```

```
len = strlen(RecordFile);
```

```
RecordFile[len-1] = StatusFile[5];
```

```
RecordFile[len-2] = StatusFile[4];
```

```
RecordFile[len-3] = StatusFile[3];
```

```
/* เปิดแฟ้ม */
```

```
if ((fp4 = fopen(RecordFile,"ab"))==NULL) /* เพิ่มข้อมูลต่อท้ายแฟ้ม */
```

```
if ((fp4 = fopen(RecordFile,"wb"))==NULL) /* เปิดแฟ้มใหม่ */
```

```
TAL_Error(240);
```

```
}
```

```
-----*/
```

MessageFileLoading ฟังก์ชันสำหรับอ่านข้อมูลจากแฟ้มข้อความโต้ตอบ(CAICOM.MSG)

บรรจุลงตารางข้อมูลโต้ตอบ(MDTBL[20][100])

input คือ ข้อมูลจากแฟ้มข้อความโต้ตอบ

output คือ ตารางข้อมูลโต้ตอบ(MDTBL)

ลำดับที่สูงสุดของข้อความโต้ตอบ (FCTBL[0][3])

```
-----*/
```

```
void MessageFileLoading(void)
```

```
{char ms[100];
```

```
int ms_no;
```

```
FILE *fp5;
```

```
if ( (fp5 = fopen("CAICOM.MSG","r")) != NULL )
```

```
{ FCTBL[0][3] = 0;
```

```
while (!feof(fp5))
```

```
{ if (fscanf(fp5,"%d %s",&ms_no,ms) == 0) TAL_Error(250);
```

```
strcpy(MDTBL[ms_no],ms);
```

```
if (ms_no > FCTBL[0][3]) FCTBL[0][3] = ms_no;
```

```
if (ferror(fp5)) TAL_Error(250);
```

```
}
```

```
fclose(fp5);
```

```
}
```

```
}
```

```
-----*/
```

LoadCourseAndMakeFrameTable ฟังก์ชันสำหรับอ่านข้อมูลจากแฟ้มบทเรียน
 บรรจูลงตารางข้อมูลบทเรียน (CDTBLL500][100])
 และตารางควบคุมกรอบการเรียน (FCTBLE50][5])

input คือ ข้อมูลจากแฟ้มเนื้อหาวิชา

output คือ ตารางข้อมูลบทเรียน (CDTBLL500][100])
 ตารางควบคุมกรอบการเรียน (FCTBLE50][5])
 ลำดับที่สูงสุดของกรอบการเรียน (FCTBLE0][1])

-----*/

/* Frame #no. [,frame][,type][,no.loop] */

void LoadCourseAndMakeFrameTable(void)

{char str[80];

int CallFrame, LineNo, TypeNo, CanUseLoop ;

int fcptr, i, p1, p2 , maxf;

FILE *fp3 ;

FCTBLE[0][0] = 0; /* frame.no start */

FCTBLE[0][2] = maxf = 0; /* max frame.no */

FCTBLE[0][1] = 0; /* max line.no */

LineNo = CallFrame = 0;

if ((fp3=fopen(HistoryTBL[itmPtr].fname,"r"))==NULL) TAL_Error(230);

else

{ while (!feof(fp3))

{ LineNo++;

if (LineNo > 500) TAL_Error(201);

fgets(CDTBLL[LineNo],80,fp3);

strcpy(str,CDTBLL[LineNo]);

if (ferror(fp3)) TAL_Error(231);

if (strcmp(strupr(strmid(str,1,5)),Commands[1]) == 0)

{ TypeNo = 9;

fcptr = atoi(strcut(str,6));

if (fcptr == 0) TAL_Error(330);

if (fcptr > maxf) maxf = fcptr; /* Max Frame No*/

p1 = strin(str,6,',,');

if (p1 == 0) {TypeNo = 0; CanUseLoop = 0;}

else

{ p1 = p1 + 1;

for(;str[p1-1] == ' ';p1++);

for (i = 0;(i < 3) && (FrameType[i] != str[p1-1]);i++);

if (i < 3)

{ TypeNo = i;

p2 = strin(str,p1,',,');

```

    if (p2 != 0) CanUseLoop = atoi(strcut(str,p2+1));
    else CanUseLoop = 0;
}
else {CanUseLoop = atoi(strcut(str,p1)); TypeNo = 0;}
}
FCTBL[fcptr][0] = 0;           /*Next Frame */
FCTBL[fcptr][1] = LineNo;
FCTBL[fcptr][2] = TypeNo;
FCTBL[fcptr][3] = CanUseLoop;
FCTBL[fcptr][4] = 0;           /*Used Loop */
FCTBL[CallFrame][0] = fcptr;
CallFrame = fcptr;
}
}
FCTBL[0][2] = maxf;
FCTBL[0][1] = LineNo + 1 ;
fclose(fp3);
}
}
/*-----
ShowReport ฟังก์ชันสำหรับแสดงผลการเรียน
input คือ ข้อมูลจากแฟ้มสรุปผลการเรียน
output คือ ข้อความแสดงผลการเรียนบนจอภาพ
-----*/
void ShowReport(void)
{
float per;
clrscr();gotoxy(1,10);
printf("ผลการเรียนวิชา %s\n\n",HistoryTBL[ItmPtr].fname);
printf("หัวข้อที่ %d เรื่อง %s\n\n",HistoryTBL[ItmPtr].itm_no,
HistoryTBL[ItmPtr].obj_name);
printf(" เรียนเมื่อ %s เวลา %s น. รวมใช้เวลา(ชม.:นาที:วินาที) %s \n",
HistoryTBL[ItmPtr].l_date, HistoryTBL[ItmPtr].s_time,
strctime(HistoryTBL[ItmPtr].ut_time));
if ((HistoryTBL[ItmPtr].evaluate != 'T')
&& (HistoryTBL[ItmPtr].evaluate != 'S'))
{ printf(" ผลการเรียน ตอบถูก %d ข้อ ผิด %d ข้อ สรุปได้ %d/%-3d คะแนน\n",
HistoryTBL[ItmPtr].itm_right,HistoryTBL[ItmPtr].itm_wrong,
HistoryTBL[ItmPtr].score,HistoryTBL[ItmPtr].fullscore);
if (HistoryTBL[ItmPtr].score > 0)

```

```
per=(float)HistoryTBL[ItemPtr].score
    *100/(float)HistoryTBL[ItemPtr].fullscore;
else per = 0;
printf(" คิดเป็นร้อยละ %5.2f ประเมินโดยใช้เกณฑ์ %s ได้เกรด %c \n\n"
    ,per,HistoryTBL[ItemPtr].e_low,HistoryTBL[ItemPtr].evaluate);
}
else
{ /*หยุดกลางคัน/บทเรียน*/
    printf(" การเรียนไม่เสร็จสมบูรณ์ ไม่สามารถรายงานผลการเรียนให้ได้\n\n");
}
printf("                                [ กด enter เพื่อดูสรุปทั้งวิชา ]\n");
getch();
}
```

```

/*=====
 16/Fab/94 18:00 TEACH4.C
 Authoring system for microcomputer
=====*/

/*-----
ส่วนการ INCLUDE FILE
-----*/

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "teaa.h"

/*-----
DecodeCommands ฟังก์ชันสำหรับจำแนกคำสั่งต่าง ๆ ของภาษา TAL
input คือ ลำดับที่เริ่มต้นของคำสั่งในตารางคำสั่งที่จะให้จำแนก(int s_cmds)
        ลำดับที่สุดท้ายของคำสั่งในตารางคำสั่งที่จะให้จำแนก(int e_cmds)
        ข้อความที่ต้องการให้จำแนกคำสั่ง(char Str[])
output คือ ลำดับที่ของคำสั่งในตารางคำสั่ง
        ถ้าไม่อยู่ในกลุ่มของคำสั่งที่กำหนดจะให้ค่าเท่ากับ 0
-----*/

int DecodeCommands(int s_cmds, int e_cmds, char Str[])
{int idx;
  for(idx=s_cmds;idx<=e_cmds;idx++)
    {if (strcmp(Commands[idx],strupr(strmid(Str,1,CmdsLen[idx])))==0)
      return(idx); /*พบว่าอยู่ในกลุ่มคำสั่งที่กำหนดให้คำรหัสคำสั่ง*/
    }
  return(0); /*ไม่อยู่ในกลุ่มคำสั่งที่กำหนดให้ค่า 0*/
}

/*-----
CrossBlank ฟังก์ชันสำหรับสิ่งข้ามช่องว่าง
input คือ ข้อความที่ต้องการให้ข้ามช่องว่าง(char Str[])
        ลำดับที่ของตัวอักษรที่ให้เริ่มต้นตรวจหาช่องว่าง(int p)
output คือ ลำดับที่ของตัวอักษรที่ไม่ใช่ช่องว่าง(int CrossBlank)
-----*/

```

```

int CrossBlank(char Str[], int p) .
{ while ((Str[p-1] == ' ') && (p < strlen(Str) ) ) p++; return(p);}
/*-----
Get1Parameter ฟังก์ชันสำหรับจำแนกหาอักขระนำคำสั่ง ของภาษา TAL
input คือ ข้อความที่ต้องการจำแนกหาอักขระนำคำสั่ง(char Str[])
ตำแหน่งของอักขระที่จะเริ่มต้นค้นหา(int ptr)
output คือ รหัสของอักขระที่พบ(int Get1Parameter)
ข้อความหลังอักขระที่พบ(char StrOut[100])
ตำแหน่งของอักขระที่จะเริ่มต้นหลังการค้นหา(int ActPtr)
-----*/

int Get1Parameter(char Str[],int ptr)
{int g, g1, code;
  StrOut[0] = '\0';
  ActPtr = ptr;
  code = 0;
  while (Str[ActPtr-1] == ' ') {ActPtr++;}
  /* กรณีที่ 1 ข้อความที่ส่ง เท่ากับ null string หรือ ตัวชี้มากกว่าความยาวของข้อความ*/
  if ((ActPtr == 0) || (ActPtr > strlen(Str)))
    { StrOut[0] = '\0'; code = 1; }
  else /* กรณีที่ 2 ตัดข้อความระหว่างเครื่องหมาย @ (---) */
    if (strcmp(strmid(Str,ActPtr,2),"@(" ) == 0) /* ES = @(___)^ */
      { g = strin(Str,ActPtr+2,')');
        strcpy(StrOut,strmid(Str,ActPtr+2,g-ActPtr-2));
        ActPtr = g+1; code = 2;
      }
    else /* กรณีที่ 3 ตัดข้อความระหว่างเครื่องหมาย quate '___' */
      if (Str[ActPtr-1] == '\')
        { g = strin(Str,ActPtr+1,'\');
          strcpy(StrOut,strmid(Str,ActPtr+1,g-ActPtr-1));
          ActPtr = g + 1 ; code = 3;
        }
      else /* กรณีที่ 4 เครื่องหมาย comma , */
        { g = strin(Str,ActPtr,',');
          if (g != 0)
            { g1 = strin(Str,g+1,',');
              if (g1 != 0) /*ตัดช่วง ,___, */
                { strcpy(StrOut,strmid(Str,g+1,g1-g-1));
                  ActPtr = g1; code = 4;
                }
            }
          else /*ตัดช่วง ,___ จบประโยค */

```

```

    { strcpy(StrOut, strcut(Str, g+1));
      ActPtr = strlen(Str)+1;      /*ActPtr ออกจากบรรทัด*/
      code = 5;
    } } }
return(code);
}

/*-----
InitialValue ฟังก์ชันสำหรับกำหนดค่าเริ่มต้นก่อนการเรียน
input คือ -
output คือ ข้อมูลเริ่มต้นสำหรับแฟ้มต่าง และการทำงานของระบบ
-----*/

void InitialValue(void)
{ /*- ข้อมูลเริ่มต้นสำหรับแฟ้มบันทึกพฤติกรรมการเรียน -*/
  act.f_no = 0 ;
  act.f_type= '\0' ;
  act.f_qno = act.loop_c = act.loop_u =0;
  act.ti_lu = 0;
  act.ti_f = StartFrameTime = StartLearnTime = time('\0');
  act.ti_fu = act.ti_fs = act.ti_fsu = act.ti_fa;
  act.ti_fu = act.ti_fau = act.ti_fb = act.ti_fbu = 0 ;
  strcpy(act.ans_s , strdate());
  strcat(act.ans_s , " ");
  strcat(act.ans_s , strtime(StartLearnTime));
  strcat(act.ans_s , " จุดเริ่มต้นบันทึกพฤติกรรมการเรียน");
  act.ans_r = ' ' ;
  act.next_f= LFptr = FCTBL[0][0] ;      /* เฟรมเริ่มต้น */
  LLptr = FCTBL[LFptr][1];      /* เลขบรรทัดบทเรียนเริ่มต้น */
  act.no_r = act.no_w = act.no_n = 0;
  act.sc_g = act.sc_u = act.sc_t = act.sc_f = 0;
  /*- ข้อมูลเริ่มต้นสำหรับแฟ้มสรุปผลการเรียน -*/
  strcpy(StatusData.fname , HistoryTBL[ItmPtr].fname) ;
  StatusData.itm_no = ItmPtr ;
  strcpy(StatusData.obj_name , HistoryTBL[ItmPtr].obj_name);
  strcpy(HistoryTBL[ItmPtr].l_date, strdate());
  strcpy(StatusData.l_date , HistoryTBL[ItmPtr].l_date);
  strcpy(HistoryTBL[ItmPtr].s_time , strtime(StartLearnTime));
  strcpy(StatusData.s_time , HistoryTBL[ItmPtr].s_time);
  StatusData.ut_time = 0;
  StatusData.itm_right = StatusData.itm_wrong = StatusData.itm_non = 0;
  StatusData.fullscore = StatusData.score = 0;
}

```

```

EvaluateStatement(ELAW); /*เกณฑ์การประเมินผลทั่วไป (elaw)*/
uevaluate          = 'S'      ;
StatusData.evaluate = uevaluate; /* ผลประเมิน S=หยุดกลางคัน
                                   T=บทเรียนผิด ฯลฯ=ผลการเรียน */
}
/*-----
TeachingProcess ฟังก์ชันสำหรับควบคุมการเรียนการสอนตามบทเรียน
input   คือ ข้อมูลเนื้อหาวิชาในตารางข้อมูลบทเรียน(CDTBL[500][100])
         ข้อมูลกรอบการเรียนจากตารางควบคุมกรอบการเรียน(FC_TBL[50][5])
output  คือ การนำเสนอบทเรียนโดยการส่งการทำงานไปยังฟังก์ชันต่อไป
         FrameStatement(), ScreenStatement(), DisplayText(),
         InputStatement(), BranchStatement(), Branching(),
         EvaluateStatement()
-----*/

void TeachingProcess(void)
{int commands ;
char StrB[100];
InitialValue(); /* -->LLptr, LFptr */
while (LLptr < FC_TBL[0][1]) /* Max Line Number */
{ commands = DecodeCommands(1,6,CDTBL[LLptr]);
if (commands != 0)
{ strcpy(StrB, strcut(CDTBL[LLptr], CndsLen[commands]+1));
switch(commands)
{ case 1: FrameStatement(StrB);          break;
case 2: ScreenStatement(StrB);  DisplayText(); break;
case 3: InputStatement(StrB);    break;
case 4: BranchStatement(StrB);  Branching();  break;
case 5: EvaluateStatement(StrB); break;
}
}
LLptr++;
} /*เมื่อตัวชี้เลขบรรทัดมีค่ามากกว่าค่าเลขบรรทัดสูงสุดแสดงว่าจบการเรียน */
}
/*-----
FrameStatement ฟังก์ชันสำหรับดำเนินการเรียนการสอนตามคำสั่ง FRAME
input   คือ ข้อความหลังคำสั่ง FRAME (char StrA[])
output  คือ บันทึกพฤติกรรมกรการเรียนของเฟรมที่แล้วลงเพิ่มบันทึกพฤติกรรมกรเรียน
         รหัสผลการดำเนินการ
-----*/

void FrameStatement(char StrA[100])

```



```

{int p;
  EndFrameTime = time('\0');
  act.ti_fu = difftime(EndFrameTime, StartFrameTime); /* เวลาทั้งหมดที่ใช้ในเฟรม */
  p = 1; for (; StrA[p-1] == ' '; p = p+1);
  clrscr();
  act.next_f = atoi(strcut(StrA, p));
  WriteDataToRecordFile(); /* บันทึกพฤติกรรมการเรียนของเฟรมที่แล้ว */
  act.f_no = LFptr = act.next_f;
  act.ti_f = StartFrameTime = EndFrameTime;
  act.ti_lu = difftime(StartFrameTime, StartLearnTime);
  act.next_f = FCTBL[LFptr][0];
  act.f_type = FrameType[FCTBL[LFptr][2]];
  act.loop_c = FCTBL[LFptr][3];
  FCTBL[LFptr][4] = FCTBL[LFptr][4] + 1;
  act.loop_u = FCTBL[LFptr][4];
  act.ti_fu = act.ti_fs = act.ti_fsu = act.ti_fa = 0;
  act.ti_fau = act.ti_fb = act.ti_fbu = 0;
  act.ans_s[0] = '\0';
  act.ans_r = ' ';
  act.sc_g = act.sc_f = 0;
  if ((act.loop_c != 0) && (act.loop_u > act.loop_c)) /* ครบจำนวนครั้งที่กำหนด */
    { LFptr = act.next_f = FCTBL[LFptr][0];
      LLptr = FCTBL[LFptr][1] - 1; }
  else
  if (toupper(act.f_type) == 'R') /* เฟรมชนิดสุ่ม */
  { LLptr = LLptr + 1;
    LFptr = act.next_f = RandomFrame(CDTBL[LLptr]);
    LLptr = FCTBL[LFptr][1] - 1;
  }
  else if (act.f_type == 'Q') act.f_qno++ ;
}
/*-----
RandomFrame ฟังก์ชันสำหรับสุ่มหมายเลขเฟรม
input คือ ข้อความกลุ่มหมายเลขเฟรม (char StrA[])
output คือ หมายเลขเฟรมที่สุ่มได้ (int RandomFrame)
-----*/
int RandomFrame(char StrA[])
{int RdTBL[30];
  int p, L1, L2, PtrL, indx, nof, nos, noe, W1;
  int Comma, LenLS;

```

```

char LSC80J, WSC12J;
indx = 0; p = 1;
for(;StrAcp-1] == ' ';p = p +1); /* กำจัดช่องว่าง */
L1 = strin(StrA,p,'(');
if (L1 < 1) TAL_Error(310); /* ไม่มีหมายเลขเฟรมให้ลุ่ม */
L2 = strin(StrA,L1,')');
strcpy(LS, strmid(StrA,L1+1,L2-L1-1));
LenLS = strlen(LS);
if (LenLS < 1) TAL_Error(310) /* ไม่มีหมายเลขเฟรมให้ลุ่ม */;
/* สร้างตารางลุ่ม */
PtrL = 1;
indx = Comma = 0; /* กำหนดค่าเริ่มต้น */
while (PtrL <= LenLS)
{ Comma = strin(LS,PtrL,','); /* ค้นหา , */
  nof = nos = noe = 0;
  if (Comma == 0)
  { Comma = LenLS +1; /* หหมด , = ชุดสุดท้าย */
    strcpy(WS, strmid(LS,PtrL,Comma-PtrL));
  }
  else strcpy(WS, strmid(LS,PtrL,Comma-PtrL));
  PtrL = Comma+1; /* กำหนดค่าตัวชี้ใหม่ */
  W1 = strin(WS,1,'-');
  p = 1;
  if (W1 == 0)
  { for(;WSCp-1] == ' ';p++); /* กำจัดช่องว่าง */
    nof = atoi(strcut(WS,p));
    if ((nof > FCTBL[0][2]) || (nof == 0)) TAL_Error(301);
    /* หมายเลขเฟรมในการลุ่มไม่ปรากฏในทเรียน */
    if ((FCTBL[nof][3] == 0) || (FCTBL[nof][3] > FCTBL[nof][4]))
      RdTBL[++indx] = nof;
  }
  else
  { for(;WSCp-1] == ' ';p++); /* กำจัดช่องว่าง */
    nos = atoi(strcut(WS,p));
    for(;WSCw1] == ' ';w1++); /* กำจัดช่องว่าง */
    noe = atoi(strcut(WS,w1+1));
    for (nof = nos; nof <= noe; nof++)
    {if ((nof > FCTBL[0][2]) || (nof == 0)) TAL_Error(301);
      if ((FCTBL[nof][3] == 0) || (FCTBL[nof][3] > FCTBL[nof][4]))
        RdTBL[++indx] = nof;
    }
  }
}

```

```

    }
}
}
RdTBL[0] = indx; /* จำนวนเฟรมสูงสุดในตารางลุ่ม */
if (indx == 0) return(FCTBLE[LFptr][0]);
randomize();
L1 = random(RdTBL[0])+1; /* หมายเลขเฟรมที่ลุ่มได้ */
return(RdTBL[L1]);
}
}
/*-----*/
ScreenStatement ฟังก์ชันสำหรับดำเนินการเรียนการสอนตามคำสั่ง Screen
input คือ ข้อความหลังคำสั่ง Screen (char StrA[])
output คือ แสดงข้อความหลังคำสั่ง Screen บนจอภาพ
และกำหนดเวลาในการแสดงข้อความ
/*-----*/
void ScreenStatement(char StrA[]) /* SCREEN [Label][,wait.time]*/
{int pcode,i;
  clrscr();
  ActPtr = 1;
  pcode = 2;
  i = 0;
  while (pcode > 1)
  { pcode = Get1Parameter(StrA,ActPtr);
    i++;
    if (pcode == 2) /* @ ( ) */
      TAL_Error(310); /* หลังคำสั่ง SCREEN กำหนดตำแหน่งแสดงผลไม่ได้ */
    if (pcode == 3) /* ' ' */
      { highvideo();
        gotoxy(SetCenter(StrOut),1);printf("%s\n",StrOut); normvideo();}
    if (pcode > 3) /* , */
      act.ti_fs = WaitTime(StrOut); /* เวลาที่กำหนดให้ในการแสดงผล */
  }
}
}
/*-----*/
Wait ฟังก์ชันสำหรับหน่วงเวลา
input คือ ระยะเวลาที่ต้องการหน่วง(int Long)
output คือ ระยะเวลาที่ต้องคอย
/*-----*/
long Wait(long Long)
{ time_t timestart, timeptr;

```

```

long TimeOut, TimeUse;
    timestart = timeptr = time('\0');
    TimeOut = 1;
    TimeUse = 0;
    while ( !kbhit() && (TimeOut == 1))
        { timeptr = time('\0');
          TimeUse = difftime(timeptr,timestart); /*จำนวนเวลาที่ใช้ในการอ่าน*/
          if (TimeUse > Long) TimeOut = 0;      /* Timeout */
        }
    if (TimeOut != 0) getch();
    return(TimeUse);
}

```

```

/*-----
DisplayText ฟังก์ชันสำหรับจำแนกการดำเนินการตามคำสั่งแสดงผลบนจอภาพ
input คือ ข้อความหรือคำสั่งแสดงผล (char StrAC[])
output คือ ผลลัพธ์การแสดงผลบนจอภาพตามคำสั่งแสดงผล
-----*/

```

```

void DisplayText(void)
{int flag, i;
  flag = TRUE;
  while (flag == TRUE)
  { LLptr++;
    i = (DecodeCommands(1,7,CDTBL[LLptr]));
    if ((i > 0) && (i < 6))
    { if (act.ti_fs != 0) act.ti_fsu = Wait(act.ti_fs);
      LLptr--; flag = FALSE;}
    else
    { if (i == 0) Display(CDTBL[LLptr]); /* str[]'[]"[]^[] @(), */
      else
      switch(i)
      { case 6:/* $C */clrscr(); break;
        case 7:/* $T */delay(1000*atoi(strcut(CDTBL[LLptr],3)));break;
      } } }
}

```

```

/*-----
Display ฟังก์ชันสำหรับแสดงข้อความบนจอภาพ
input คือ ข้อความที่ต้องการนำแสดงบนจอภาพ (char StrAC[])
output คือ รหัสผลการดำเนินการ (int Display)
-----*/

```

```

int Display(char StrAC[]) /* ^, ", @(), str */

```

```

{int G1, G2, OldP1, p;
  int row, col;
  char ES[100], BS[100];
  row = wherey();
  col = wherex();
  OldP1 = p = 1;
  if (p > strlen(StrA)) { ES[0] = '\0'; return(0); }
  if (strcmp(strmid(StrA,p,1),"^" ) == 0)          /* prefix กึ่งกลาง */
    { p++; strcpy(ES, strcut(StrA,p)); col = SetCenter(ES); }
  else
  if (strcmp(strmid(StrA,p,1),"'" ) == 0)          /* prefix ชัดขวา */
    { p++; strcpy(ES, strcut(StrA,p)); col = SetCenter(ES)*2; }
  else
  if (StrA[0] == '"')          /* prefix ชัดซ้าย */
    { p++; strcpy(ES, strcut(StrA,p)); col = 1; }
  else
  if (strcmp(strmid(StrA,p,2),"@" ) == 0)          /* ES = @(____)^ */
    { G1 = strin(StrA,p+2,')');
      strcpy(BS, strmid(StrA,p+2,G1-p-2));
      row = atoi(BS);
      G2 = strin(BS,1,',');
      col = atoi(strcut(BS,G2+1));

      G1 = strin(StrA,G1+1,'\');
      G2 = strin(StrA,G1+1,'\');
      strcpy(ES, strmid(StrA,G1+1,G2-G1-1));
    }
  else { strcpy(ES, strcut(StrA,OldP1)); col = 1; }
  gotoxy(col, row); printf("%s", ES);
  return(1);
}

```

```

/*-----
SetCenter ฟังก์ชันสำหรับกำหนดตำแหน่งคอลัมน์ ที่จะทำให้แสดงข้อความอยู่กึ่งกลาง
      จอภาพ
input คือ ข้อความที่ต้องการนำแสดงบนจอภาพ (char Lbl[])
output คือ ตำแหน่งคอลัมน์ที่จะทำให้อข้อความอยู่กึ่งกลางจอภาพ (int SetCenter)
-----*/

```

```

int SetCenter(char Lbl[])
{int i, colptr, L_len;
  L_len = strlen(Lbl);

```

```

i = 0 ;
colptr = 0;
for (i = 0; i <= L_len; i++) if (toascii(Lbl[i]) < 215) colptr++;
if (colptr <= 80) colptr = 40 - colptr/2;
else colptr = 0;
return(colptr);
}
/*-----
InputStatement ฟังก์ชันสำหรับดำเนินการเรียนการสอนตามคำสั่ง INPUT
           [INPUT [str][,waittime]]
           StrA = [' " ^ @,(, str '\b']
input คือ ข้อความหลังคำสั่ง INPUT (char StrA[])
output คือ รหัสผลการดำเนินการ(int InputStatement)
-----*/

int InputStatement(char StrA[])
{int row, col;
  int i, G1;
  int pcode,TimeOut;
  time_t timestart, timeptr;
  act.ti_fa = act.ti_fau = 0;
  ActPtr = 1;
  pcode = 2;
  while (pcode > 1)
  { pcode = Get1Parameter(StrA,ActPtr);
    if (pcode < 4) /* @(___) */
    { if (pcode == 2)
      { row = atoi(StrOut);
        G1 = strin(StrOut,1,',');
        col = atoi(streut(StrOut,G1+1));
        gotoxy(col,row);
        pcode = Get1Parameter(StrA,ActPtr);
      }
    }
    if (pcode == 3) /* '_____' */
    printf("%s",StrOut);
  }
  if (pcode > 3) /* , */
  { act.ti_fa = WaitTime(StrOut); } /* จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในเฟรม */
}
/* wait input answer */
i = 0;

```

```

col = wherex(); row = wherey();
strcpy(act.ans_s, " ");
timestart = timeptr = time('\0');
TimeOut = 1;
while (TRUE)
{ while ( (!kbhit()) && (TimeOut == 1) )
  { timeptr = time('\0');
    act.ti_fau = difftime(timeptr,timestart);
    /* จำนวนเวลาที่ใช้ในการตอบคำถามในเฟรม */
    if ((act.ti_fa > 0) && (act.ti_fau > act.ti_fa)) /* Timeout */
    { TimeOut = 0; act.ans_s[i] = '\0';
      if (act.f_type == 'Q') printf(" \n หมดเวลา \007 \007 \007 \n");
      else printf(" \n \007 \007 \007 \n");
      Wait(2);
      return(0);
    }
  }
  act.ans_s[i] = getch();
/* if (iscntrl(act.ans_s[i]) == 0)
  {if (j == 0) i = i + 1; else j = 0;}
  else */
  switch (act.ans_s[i])
  { case RETURN : act.ans_s[i] = '\0';
    if (strcmp(strupr(strmid(act.ans_s,1,7)),
      "//FRAME") == 0)
    { LFptr = act.next_f = atoi(strcut(act.ans_s,8));
      LLptr = FCTBL[LFptr][i]-1;}
    else
    if (strcmp(strupr(strmid(act.ans_s,1,5)),
      "//END") == 0)
    { act.next_f = 998; LLptr = FCTBL[0][i];}
    printf("\n"); /* จบประโยคคำตอบ */
    return(0);
  }
  case '\b' : if (i <= 0) i = 0;
    else i = i - 1;
    break;
/* case ESC : break; */
  default : i = i +1; break;
}
act.ans_s[i] = '\0';

```

```

gotoxy(col,row); clreol();
gotoxy(col,row); printf("%s",act.ans_s);
}
}
/*-----
WaitTime ฟังก์ชันสำหรับวิเคราะห์หาคำสั่ง $T
input คือ ข้อความที่ต้องการวิเคราะห์หา $T (char StrA[])
output คือ จำนวนเวลาที่จะให้รอคอย (int WaitTime)
-----*/
int WaitTime(char StrA[])
{int g;
int no; /* จำนวนเวลาที่จะให้รอคอย */
g = strin(StrA,1,'$');
if ((g != 0) && (toupper(StrA[g]) == 'T'))
{ no = atoi(strcut(StrA,g+2)); /* จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในเฟรม */
if (no == 0) TAL_Error(321); /* ไม่ระบุจำนวนเวลา */
else return(no);
}
else
TAL_Error(322); /* มีเครื่องหมาย , แต่คำสั่งตามไม่ใช่ $T */
return(0);
}
/*-----
BranchStatement ฟังก์ชันสำหรับวิเคราะห์คำสั่ง BRANCH
Branch Vn [, $T]
input คือ ข้อความหลังคำสั่ง Branch (char StrA[])
output คือ คะแนนเต็มสำหรับเฟรม Vn
จำนวนเวลาที่รอคอยก่อนไปดำเนินงานในเฟรมถัดไป $T
-----*/
void BranchStatement(char StrA[])
{int pcode;
act.sc_f = DecodeScore(StrA); /* fullscore = Vn */
pcode = Get1Parameter(StrA,ActPtr);
if (pcode > 3) act.ti_fb = WaitTime(StrOut); /* เวลาของข้อความได้ตอบ */
}
/*-----
DecodeScore ฟังก์ชันสำหรับวิเคราะห์หาคะแนน Vn
input คือ ข้อความหลังคำสั่ง Branch (char StrA[])
output คือ คะแนนเต็มสำหรับเฟรม (int DecodeScore)
ลำดับที่ของอักขระที่จะวิเคราะห์ต่อไป (int ActPtr)
-----*/

```



```

-----*/
int DecodeScore(char StrA[])
{int V, G1;
  G1 = strin(StrA,1,'V');
  V = 0 ; /* initail V */
  if (G1 != 0)
  { V = atoi(strcut(StrA,G1+1));
    if (V == 0) TAL_Error(350);/*ไม่กำหนดคะแนนเต็มหลัง V*/
  }
  else if (act.f_type == 'Q') V = 1;
  ActPtr = G1+1;
  return(V);
}
/*-----*/
Branching ฟังก์ชันสำหรับตรวจสอบคำตอบและแตกแขนงไปยังเฟรมต่าง ๆ
      [ 'c'[Vn C W N][C:fno][E,str MSG#]
      C = FULL, W = 0, NIDef = No count score Vn:fno,MSG#
input  คือ ข้อมูลเนื้อหาวิชาในตารางข้อมูลบทเรียน(CDTBL[500][100])
output คือ การส่งการทำงานไปยังกรอบการเรียนอื่น
-----*/
void Branching(void)
{int flag, flag1, flag2, i,G1, fno;
  char ASC[100], BSC[100];
  flag = 1;
  flag2 = 0;
  while (flag == 1)
  { LLptr++;
    strcpy(AS,CDTBL[LLptr]);
    i = DecodeCommands(1,8,AS);
    if (i == 1) { LLptr--; flag = 0;} /* FRAME */
    else
    { if ((i > 1) && (i < 14)) TAL_Error(351);
      /*บรรทัดหลังคำสั่ง Branch ผิดไวยกรณ์*/
    else /* 'n'[Vn[CWN]:fno, str||MSG# */
    { ActPtr = 1;
      fno = 0;
      flag1 = 0;
      while ((ActPtr < strlen(AS)) && (flag1 == 0))
      { for(; ASC[ActPtr-1]==' '; ActPtr++);
        switch(ASC[ActPtr-1])

```

```

{case '\': G1 = strin(AS,ActPtr+1,'\');
        strcpy(BS, strmid(AS,ActPtr+1,G1-ActPtr-1));
        if (strcmp(BS,act.ans_s) == 0) /*ตอบถูก*/
        { ActPtr = G1+1;
          act.ans_r = AS[ActPtr-1];
        }
        else flag1 = 1;
        break;
case 'V': ActPtr++;
        act.sc_g = atoi(strcut(AS,ActPtr));
        act.no_r = act.no_r + 1;
        act.sc_u = act.sc_u + act.sc_g;
        act.sc_t = act.sc_t + act.sc_f;
        for(; isdigit(AS[ActPtr-1]) != 0; ActPtr++);
        /*เลื่อนเมื่อพบตัวเลข*/
        break;
case 'C': act.sc_g = act.sc_f;
        act.no_r = act.no_r + 1;
        act.sc_u = act.sc_u + act.sc_g;
        act.sc_t = act.sc_t + act.sc_f;
        ActPtr++;
        break;
case 'W': act.sc_g = 0;
        act.no_w = act.no_w + 1;
        act.sc_t = act.sc_t + act.sc_f;
        ActPtr++;
        break;
case 'N': act.sc_g = 0;
        act.no_n = act.no_n + 1;
        ActPtr++;
        break;
case ':': flag2 = 1;
        ActPtr++;
        for(;AS[ActPtr-1] == ' '; ActPtr++);
        fno = atoi(strcut(AS,ActPtr));
        if (((fno < 0) || (fno > FCTBLIOJ[2]))
            && (fno != 999))
            TAL_Error(340); /* หมายเลขเฟรมไม่มีในบทเรียน */
        for(;isdigit(AS[ActPtr-1])!=0; ActPtr++);
        /*เลื่อนเมื่อพบตัวเลข*/

```

```

        break;
    case ',': Message(strcut(AS,ActPtr));
        act.ti_fbu = WaitBranch(act.ti_fb);
        flag = 0;
        flag1 = 1;
        break;
    default : TAL_Error(341);/*คำสั่งย่อยใน BRANCH ไม่ถูกต้อง*/
}
}
if (flag2 == 1)
{ LFptr = fno;
  if (LFptr == 999) LLptr = FCTBL[LFptr][1];
  else LLptr = FCTBL[LFptr][1] - 1;
  flag = 0;
  flag1 = 1;
  act.next_f = LFptr;
}
}
}
} /* while */
}
/*-----
Message ฟังก์ชันสำหรับแสดงข้อความโต้ตอบ
input คือ ข้อความที่ต้องการวิเคราะห์ (char StrA[])
output คือ ข้อความโต้ตอบปรากฏบนจอภาพ 0=ไม่มี string, 1=Message, 2=string
-----*/
int Message(char StrA[])
{int G1, p, g, Mno, code;
char BS[100];
Mno = 0;
G1 = strin(StrA,1,',');
if (G1 == 0) code = 0; /*ไม่มี , */
else /*มี , แสดงว่าข้อความ */
{ p = strin(StrA,G1+1,'$');
  if ((StrA[p-1]=='$') && (StrA[p]=='M'))
  { Mno = atoi(strcut(StrA,p+2));
    if ((Mno > 0) && (Mno < FCTBL[0][3]))
    { strcpy(BS,MDTBL[Mno]); code = 1;}
    else TAL_Error(342); /*ไม่มีหมายเลขข้อความ*/
  }
}
}

```

```

else
{ p = strin(StrA,G1+1,'\');
  g = strin(StrA,p+1,'\');
  if ((p==0)||(g == 0)) TAL_Error(352);
      /*มี , แต่ไม่มีข้อความโต้ตอบใน Branch*/
  strcpy(BS,strmid(StrA,p+1,g-p-1));
  code = 2;
}
gotoxy(SetCenter(BS),wherey()+1); printf("%s\n",BS);
}
return(code);
}
/*-----*/
WaitBranch ฟังก์ชันสำหรับหน่วงเวลา
input คือ ระยะเวลาที่ต้องการหน่วง(int Long)
output คือ ระยะเวลาที่ต้องคอย
-----*/
long WaitBranch(long Long)
{ time_t timestart, timeptr;
  long TimeOut, TimeUse;
  timestart = timeptr = time('\0');
  TimeOut = 1;
  while ( !kbhit() && (TimeOut == 1) )
  { timeptr = time('\0');
    TimeUse = difftime(timeptr,timestart); /* จำนวนเวลาที่ใช้ในการอ่าน */
    if ((Long != 0)&&(TimeUse > Long)) TimeOut = 0; /* Timeout */
  }
  if (TimeOut != 0) getch();
  return(TimeUse);
}
/*-----*/
SaveStatus ฟังก์ชันสำหรับเขียนข้อมูลลงแฟ้มสรุปผลการเรียน
ฟังก์ชันนี้จะถูกใช้เมื่อยุติการเรียน
input คือ ข้อมูลสรุปผลการเรียน
output คือ เขียนข้อมูลลงแฟ้มสรุปผลการเรียน
-----*/
void SaveStatus(void)
{ if (( fp2 = fopen(StatusFile,"ab")) == NULL) TAL_Error(222);
  /*มีปัญหาคณะเปิดแฟ้มสรุปผลการเรียนเพื่อบันทึกข้อมูล*/
else

```

```

{EndLearnTime          = time('\0') ;
  if ((FCTBL[LFptr][4] == 0) || (LFptr == 999))
  { if (act.sc_u > 0) uevaluate = Grade(act.sc_u*100/act.sc_t);
    else uevaluate = Grade(0);
  }
  StatusData.ut_time      = difftime(EndLearnTime,StartLearnTime);
  HistoryTBL[ItmPtr].ut_time = StatusData.ut_time;
  StatusData.itm_right = HistoryTBL[ItmPtr].itm_right = act.no_r;
  StatusData.itm_wrong = HistoryTBL[ItmPtr].itm_wrong = act.no_w;
  StatusData.itm_non   = HistoryTBL[ItmPtr].itm_non   = act.no_n;
  StatusData.fullscore = HistoryTBL[ItmPtr].fullscore = act.sc_t;
  StatusData.score     = HistoryTBL[ItmPtr].score     = act.sc_u;
  if (act.sc_t == 0)
  { strcpy(uelaw," - ");
    uevaluate = 'P';
  }
  strcpy(HistoryTBL[ItmPtr].e_law , uelaw);
  strcpy(StatusData.e_law , uelaw) ; /* เก็บค่าการประเมินผลทั่วไป (elaw)*/
  StatusData.evaluate = HistoryTBL[ItmPtr].evaluate = uevaluate ;
  fwrite(&StatusData,sizeof StatusData,1,fp2);
  if (ferror(fp2)) TAL_Error(221); /*มีปัญหาคณะเขียนข้อมูลลงแฟ้มสรุปผลการเรียน*/
}
}
/*-----
EvaluateStatement ฟังก์ชันสำหรับวิเคราะห์คำสั่งการประเมินผลการเรียน
input   คือ ข้อความที่ต้องการวิเคราะห์ (char StrA[])
output  คือ ตารางระดับคะแนน
-----*/

void EvaluateStatement(char StrA[])
{int  p, L1, L2, PtrL, indx, W1;
  int  Comma, LenLS;
  char LS[80], WS[12];
  p = 1;
  for(;StrA[p-1] == ' ';p++); /* กำจัดช่องว่าง */
  L1 = strin(StrA,p,'(');
  if (L1 < 1) TAL_Error(510); /* คำสั่ง EVALUATE ไม่ถูกต้อง */
  L2 = strin(StrA,L1,')');
  if (L2 < L1) TAL_Error(510); /* คำสั่ง EVALUATE ไม่ถูกต้อง */
  strcpy(LS,StrA,L1+1,L2-L1-1);
  strcpy(uelaw,LS);
}

```

```

LenLS = strlen(LS);
if (LenLS < 1) TAL_Error(510); /* คำสั่ง EVALUATE ไม่ถูกต้อง */
/* สร้างตารางระดับเกรด */
PtrL = 1;
indx = Comma = 0; /* กำหนดค่าเริ่มต้น */
while (PtrL <= LenLS)
{
  indx = indx + 1;
  Comma = strin(LS,PtrL,','); /* ค้นหา , */
  if (Comma == 0)
  {
    Comma = LenLS + 1; /* หหมด , = ชุดสุดท้าย */
    strcpy(WS,strcmp(LS,PtrL,Comma-PtrL));
  }
  else strcpy(WS,strcmp(LS,PtrL,Comma-PtrL));
  PtrL = Comma+1; /* กำหนดค่าตัวชี้ใหม่ */
  p = 1;
  W1 = strin(WS,p,':');
  if (W1 == 0) TAL_Error(510); /* คำสั่ง EVALUATE ไม่ถูกต้อง */
  else
  {
    for(;WS[p-1] == ' ';p++); /* กำจัดช่องว่าง */
    EvaG[indx] = WS[p-1];
    W1 = W1 + 1;
    for(;WS[W1-1] == ' ';W1++); /* กำจัดช่องว่าง */
    EvaS[indx] = atoi(strcut(WS,W1));
  }
}
EvaS[0] = indx; /* จำนวนเกรดที่กำหนด */
}

```

```

/*-----
Grade ฟังก์ชันสำหรับให้เกรดผู้เรียน
input คือ คะแนนที่ผู้เรียนทำได้ (int x)
output คือ เกรดที่ได้ (char Grade)
-----*/

```

```

char Grade(long x)
{
  int i;
  char Grd;
  Grd = 'T';
  for (i = 1 ; i <= EvaS[0]; i++)
    if (x >= EvaS[i]) {Grd = EvaG[i]; i = EvaS[0]+1;}
  return(Grd);
}

```

```

}
/*-----
WriteDataToRecordFile ฟังก์ชันสำหรับเขียนข้อมูลลงแฟ้มบันทึกผลการเรียน
input คือ ข้อมูลพฤติกรรมการเรียนในแต่ละกรอบการเรียน
output คือ การเขียนข้อมูลลงแฟ้มบันทึกพฤติกรรมการเรียน
-----*/

void WriteDataToRecordFile(void)
{ record.f_no      = act.f_no;   record.f_type = act.f_type;
  record.f_qno     = act.f_qno;  record.loop_c = act.loop_c;
  record.loop_u    = act.loop_u; record.ti_lu  = act.ti_lu;
  record.ti_f      = act.ti_f;   record.ti_fu  = act.ti_fu;
  record.ti_fs     = act.ti_fs;  record.ti_fsu = act.ti_fsu;
  record.ti_fa     = act.ti_fa;  record.ti_fau = act.ti_fau;
  record.ti_fb     = act.ti_fb;  record.ti_fbu = act.ti_fbu;
  strcpy(record.ans_s , act.ans_s); record.ans_r = act.ans_r;
  record.next_f    = act.next_f; record.no_r    = act.no_r;
  record.no_w      = act.no_w;   record.no_n  = act.no_n;
  record.sc_g      = act.sc_g;   record.sc_u  = act.sc_u;
  record.sc_t      = act.sc_t;   record.sc_f  = act.sc_f;
  fwrite(&record, sizeof record, 1, fp4);
  if (ferror(fp4)) TAL_Error(241); /*มีปัญหาขณะเขียนข้อมูลลงแฟ้มบันทึกการเรียน*/
}
/*-----
EndingProcess ฟังก์ชันสำหรับเก็บข้อมูลและปิดแฟ้มข้อมูลต่าง ๆ
ก่อนการจบการเรียน
input คือ -
output คือ การปิดแฟ้มข้อมูลต่าง ๆ
-----*/

void EndingProcess(void)
{ EndFrameTime = time('\0');
  act.ti_fu = difftime(EndFrameTime, StartFrameTime);
  /* เวลาทั้งหมดที่ใช้ในเฟรม */
  act.next_f = LFptr;
  WriteDataToRecordFile(); /*บันทึกพฤติกรรมการเรียนของเฟรมที่แล้ว*/
  SaveStatus(); fcloseall(); ShowReport();
}
/*-----
EndTitle ฟังก์ชันสำหรับแสดงข้อความจบการใช้งานระบบ
input คือ -
output คือ ข้อความแสดงการจบการใช้งานบนจอภาพ
-----*/

```

```
-----*/  
void EndTitle(void)  
{ clrscr();  
  gotoxy(25,10); printf("+-----+\n");  
  gotoxy(25,11); printf("|      ขอขอบคุณที่ใช้บริการ      |\n");  
  gotoxy(25,12); printf("| ระบบช่วยการเรียนการสอน ผู้สอน |\n");  
  gotoxy(25,13); printf("|      สวัสดี      |\n");  
  gotoxy(25,14); printf("+-----+\n");  
  exit(1);  
}
```



```

/*=====
02/05/94 22:00 ANALYSER.C
=====

Programme by Nipon Suppasri. January 9, 1994.
Master of Computer Science Thesis. Prince of Songkla University.
NOTE: This program is the tool for analy student output.
โปรแกรมเรียกดูบันทึกการเรียนของผู้เรียน พัฒนาระบบโดย นายนิพนธ์ ศุภศรี
โครงการวิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2535
ที่ปรึกษาโครงการ: อาจารย์วุฒิพงศ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา */

/*-----
ส่วนการ INCLUDE FILE
-----*/

#include <dos.h>
#include <math.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

/*-----
ส่วนการนิยามค่าคงที่ต่าง ๆ
-----*/

#define ESC 0x1b /*- เป็น escape -----*/
#define TRUE 1 /*- ค่าเมื่อเป็น จริง -----*/
#define FALSE 0 /*- ค่าเมื่อเป็น เท็จ -----*/
#define LEN 20 /*- ความยาวของชื่อแฟ้ม -----*/
#define ELAW "(A:80,B:70,C:60,D:50,E:0)" /*แผนที่ทั่วไปในการประเมินผล*/

/*-----
ส่วนการประกาศชื่อแฟ้มข้อมูล
-----*/

char RecordFile[LEN]; /* RW> SSSL00.III (R) | LL = Lesson no. */
char StatusFile[LEN]; /* W-> SSSIII.STS (R) | OO = Object no. */
FILE *fp1, *fp2;

/*-----
ส่วนการกำหนดโครงสร้างข้อมูลของแฟ้มข้อมูลและตารางข้อมูลต่าง ๆ
-----*/

```

```

struct StatusFile { char fname[13];      /* ชื่อแฟ้มหัวข้อ */
    int itm_no;          /* ลำดับที่หัวข้อ */
    char obj_name[100]; /* ชื่อหัวข้อ */
    char l_date[11];    /* วัน-เดือน-ปี ที่เข้าเรียน */
    char s_time[9];     /* เวลาที่เริ่มใช้บทเรียน */
    long ut_time;       /* เวลาทั้งหมดที่ใช้ในการเรียน */
    int itm_right;      /* จำนวนข้อตอบถูก */
    int itm_wrong;      /* จำนวนข้อตอบผิด */
    int itm_non;        /* จำนวนข้อตอบไม่มีผู้ผิด */
    int fullscore;      /* คะแนนเต็ม */
    int score;          /* คะแนนทั้งหมดที่ได้ */
    char e_law[100];    /* เกณฑ์การประเมินผล */
    char evaluate;      /* ผลการประเมิน */
                        /* T = บทเรียนมีปัญหา หยุดการเรียน */
                        /* S = ผู้เรียนหยุดเรียนกลางคัน */
                        /* ฯลฯ = ผลการประเมิน */
} StatusData;          /*- แฟ้มสรุปผลการเรียน ---*/

struct ItmFile { char fname[13];      /* ชื่อแฟ้มหัวข้อ */
    char obj_name[100]; /* ชื่อหัวข้อ */
} ItmData[10]; /*- แฟ้มหัวข้อ ---*/

struct StatusFile HistoryTBL[10]; /*- ข้อมูลประวัติการเรียน ---*/

struct recdata
{ int f_no;          /* หมายเลขแฟ้ม */
    char f_type;     /* ชนิดแฟ้ม */
    int f_qno;       /* จำนวนแฟ้มคำถามสะสม */
    int loop_c;      /* จำนวนครั้งที่ยอมให้เข้าใช้ */
    int loop_u;      /* จำนวนครั้งที่เคยเข้าใช้ */
    long ti_lu;      /* จำนวนเวลาสะสมที่ผ่านมาในบทเรียน */
    long ti_f;       /* เวลาเริ่มเข้าใช้แฟ้ม */
    long ti_fu;      /* จำนวนเวลาทั้งหมดที่ใช้ในแฟ้ม */
    long ti_fs;      /* จำนวนเวลาที่กำหนดให้สำหรับแสดงข้อความในแฟ้ม */
    long ti_fsu;     /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความในแฟ้ม */
    long ti_fa;      /* จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในแฟ้ม */
    long ti_fau;     /* จำนวนเวลาที่ใช้ในการตอบคำถามในแฟ้ม */
    long ti_fb;      /* จำนวนเวลาที่กำหนดให้ใช้แสดงข้อความโต้ตอบในแฟ้ม */
    long ti_fbu;     /* จำนวนเวลาที่ใช้จริงในการแสดงข้อความโต้ตอบในแฟ้ม */
    char ans_s[80];  /* คำตอบของผู้เรียน */
    char ans_r;      /* ผลลัพธ์จากคำตอบ */
    int next_f;      /* 1-100 = หมายเลขแฟ้มต่อไป */
                    /* 999 = จบการเรียน */
}

```

```

/* 1000 = เลิกเรียนการค้น */
int no_r; /* จำนวนข้อสอบถูกสะสม */
int no_w; /* จำนวนข้อสอบผิดสะสม */
int no_n; /* จำนวนข้อสอบไม่มีถูกผิดสะสม */
int sc_g; /* คะแนนที่ได้ในเฟรมนี้ */
int sc_u; /* คะแนนสะสมที่ได้ */
int sc_t; /* คะแนนรวมสะสม */
int sc_f; /* คะแนนเต็มของเฟรมนี้ */
} record; /*- แฟ้มบันทึกผลการเรียน --*/
struct date today; /*int da_mon, da_day, da_year */
struct time now; /*int ti_hour, ti_min, ti_sec, ti_hund */
/*-----
ส่วนการประกาศตัวแปรทั่วไป
-----*/

int LessonType, StudentType, Max;
int ItmPtr; /* ตัวชี้ลำดับที่ของแฟ้มบทเรียนปัจจุบัน */
int His_Time; /* จำนวนครั้งที่เข้าเรียน */
int Max_Itm; /* จำนวนหัวข้อ */
int FLAGE;
char StudentNo[LEN], CourseName[LEN];
char ueLaw[100]; /* เกณฑ์การประเมินผลที่ใช้จริงผล */
char uevaluate; /* ผลการประเมินที่ผู้เรียนได้รับ S = หยุดเรียนกลางคัน */
/* T = บทเรียนผิดพลาด */
/* ฯลฯ = ผลการเรียน */

/*-----
ส่วนการประกาศฟังก์ชันต้นแบบ
-----*/

void Pause(void);
long Wait(long);
void Title(void);
void GetIdCourse(void);
void CreateFileName(void);
void ShowStatus(void);
void CreateAndOpenRecordFileName(void);
void EndTitle(void);
char *strtime(time_t t);
char *strctime(long t);
/*-----
ฟังก์ชันหลัก (main function)

```

```

-----*/
main ()
{ Title();
  GetIdCourse();
  CreateFileName();
  while (TRUE)
  { ShowStatus();/*EndTitle()*/
    CreateAndOpenRecordFileName();
    fcloseall();
  }/* while */
}
/*-----
   PAUSE ฟังก์ชันสำหรับหยุดรอรับการเคาะแป้นพิมพ์จากผู้ใช้โปรแกรม
   ถ้าเคาะแป้น ESC จะออกจากการใช้งานโปรแกรม ส่วนแป้นอื่น ๆ จะดำเนินโปรแกรมต่อไป
-----*/
void Pause(void)
{static char msg[] = "กดแป้น Esc เลิกใช้งาน หรือ กดแป้นใด ๆ ทำต่อไป ...";
  int c;
  gotoxy(20,23); printf("%s\n",msg);/* Put msg at bottom of screen */
  c = getch();          /* Read a character from kbd */
  if( ESC == c ) exit( 1 ); /* If user wish to return to OS */
  if( 0 == c ) c = getch(); /* non-ASCII key scan for keyboard */
}
/*-----
   Wait ฟังก์ชันสำหรับหน่วงเวลา
   input คือ ระยะเวลาที่ต้องการหน่วง(int Long)
   output คือ ระยะเวลาที่ต้องคอย
-----*/
long Wait(long Long)
{ time_t timestart, timeptr;
  long TimeOut, TimeUse;
  timestart = timeptr = time('\0');
  TimeOut = 1;
  TimeUse = 0;
  while ( !kbhit() && (TimeOut == 1))
  { timeptr = time('\0');
    TimeUse = difftime(timeptr,timestart); /* จำนวนเวลาที่ใช้ในการอ่าน */
    if (TimeUse > Long) TimeOut = 0; /* Timeout */
  }
  if (TimeOut != 0) getch();
}

```

```

return(TimeUse);
}

```

```

/*-----
strtime ฟังก์ชันสำหรับเรียกดูเวลาของเครื่องในรูปแบบข้อความ
input   คือ เวลาของเครื่องในขณะนั้น(time_t t)
output  คือ เวลาของเครื่องในรูปแบบข้อความ(char *strtime)
        HH:MM:SS
-----*/

```

```

char *strtime(time_t t)
{ struct tm *sys_time;
  char *s = "00:00:00" ;
  char hour_[3], min_[3], sec_[3];
  sys_time = localtime(&t);
  itoa(sys_time->tm_hour, hour_,10);
  itoa(sys_time->tm_min , min_,10);
  itoa(sys_time->tm_sec , sec_,10);
  if (strlen(hour_) < 2) {s[0] = '0'      ; s[1] = hour_[0];}
  else                  {s[0] = hour_[0]; s[1] = hour_[1];}
  s[2] = ':';
  if (strlen(min_) < 2) {s[3] = '0'      ; s[4] = min_[0];}
  else                  {s[3] = min_[0] ; s[4] = min_[1];}
  s[5] = ':';
  if (strlen(sec_) < 2) {s[6] = '0'      ; s[7] = sec_[0];}
  else                  {s[6] = sec_[0] ; s[7] = sec_[1];}
  s[8] = '\0';
  return(s);
}

```

```

/*-----
strctime ฟังก์ชันสำหรับเปลี่ยนวินาทีเป็นข้อความในรูปแบบ ชั่วโมง:นาที:วินาที
input   คือ เวลาเป็นวินาที(int t)
output  คือ ข้อความในรูปแบบ ชั่วโมง:นาที:วินาที(char *strtime)
        HH:MM:SS
-----*/

```

```

char *strctime(long t)
{ long h, m, s1;
  char *s = "00:00:00";
  char hour_[3], min_[3], sec_[3];
  { s1 = fmod(t,60);
    m = fmod((t - s1)/60,60);

```

```

h = fmod((t-s1-m*60)/3600-5,24);
ltoa(h, hour_,10);
ltoa(m, min_,10);
ltoa(s1, sec_,10);
if (strlen(hour_) < 2) {s[0] = '0'      ; s[1] = hour_[0];}
else                    {s[0] = hour_[0]; s[1] = hour_[1];}
s[2] = ':';
if (strlen(min_) < 2)  {s[3] = '0'      ; s[4] = min_[0];}
else                    {s[3] = min_[0] ; s[4] = min_[1];}
s[5] = ':';
if (strlen(sec_) < 2) {s[6] = '0'      ; s[7] = sec_[0];}
else                    {s[6] = sec_[0] ; s[7] = sec_[1];}
s[8] = '\\0';
}
return(s);
}

/*-----
Title   ฟังก์ชันสำหรับแสดงข้อความส่วนนำเข้าสู่ระบบ
input   คือ -
output  คือ  ข้อความบนจอภาพแสดงส่วนนำเข้าสู่ระบบ
-----*/

void Title(void)
{int i;
  clrscr();
  i = 20;
  gotoxy(i, 8); printf("+-----+\n");
  gotoxy(i, 9); printf("| +-----+ |\n");
  gotoxy(i,10); printf("| |                | |\n");
  gotoxy(i,11); printf("| |                | |\n");
  gotoxy(i,12); printf("| |  ผู้ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน' | |\n");
  gotoxy(i,13); printf("| +-----+ |\n");
  gotoxy(i,14); printf("+-----+\n");
  gotoxy(1,16);
  printf("                โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต\n");
  printf("                มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536\n");
  printf("    ที่ปรึกษาโครงการ : อาจารย์วุฒิมงคล เตชะดำรงสิน ประธานกรรมการที่ปรึกษา\n");
  printf("                รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา\n");
  printf("    พัฒนาระบบโดย   : นายนิพนธ์ ศุภศรี\n");
  Wait(4);
}

```

}

```

/*-----*/
GetIdCourse ฟังก์ชันสำหรับรับการป้อนรหัสผู้เรียนและชื่อบทเรียน
input คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
output คือ ชื่อบทเรียน (CourseName) และ StudentNo (รหัสผู้เรียน)
-----*/

void GetIdCourse(void)
{int flag, count;
  clrscr();
  gotoxy(15,9);
  printf(" ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน'");
  gotoxy(15,10);
  printf(" =====");
  flag = 1; count = 0;
  while (flag ) /* ตรวจสอบการป้อนรหัสที่ */
  { gotoxy(1,12);clreol();
    gotoxy(1,13);clreol();
    gotoxy(15,12);
    printf(" โปรดป้อนรหัสบทเรียน[SSSS] :"); gets(CourseName);
    if (strlen(CourseName) != 3)
    { gotoxy(30,13); printf("รหัสบทเรียนไม่ถูกต้อง"); Wait(2);
      count++;
      if (count > 2)
      { clrscr(); gotoxy(15,13);
        printf("\007 \007 \007 รหัสบทเรียนไม่ถูกต้อง 3 ครั้ง ไม่มีสิทธิ์เข้าใช้โปรแกรม");
        Wait(2); exit(1);
      }
    }
  }
  else flag = 0 ;
}

flag = 1; count = 0;
while (flag ) /* ตรวจสอบการป้อนรหัสที่ */
{ gotoxy(1,13);clreol();
  gotoxy(1,14);clreol();
  gotoxy(15,13);printf(" โปรดป้อนรหัสของผู้เรียน[XXX] :"); gets(StudentNo);
  if ((strlen(StudentNo) < 1) || (strlen(StudentNo) > 3))
  { gotoxy(30,14); printf("รหัสผู้เรียนไม่ถูกต้อง"); Wait(2);
    count++;
    if (count > 2)

```

```

    { clrscr(); gotoxy(15,13);
      printf("\007 \007 \007 รหัสผู้เรียนไม่ถูกต้อง 3 ครั้ง ไม่มีสิทธิ์เข้าใช้โปรแกรม");
      Wait(2);
      Wait(2); exit(1);
    }
  }
else flag = 0 ;
}
}
/*-----
CreatFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มสรุปผลการเรียน
              อ่านข้อมูลในแฟ้มสรุปผลการเรียนมาเก็บในตารางข้อมูลประวัติการเรียน
input คือ รหัสผู้เรียนและรหัสสัทเรียน และข้อมูลจากแฟ้มสรุปผลการเรียน
output คือ ชื่อแฟ้มสรุปผลการเรียน (StatusFile= SSSI.II.STS)
              ตารางข้อมูล HistoryTBL
-----*/
void CreatFileName(void)
{int i;
  /* สร้างชื่อแฟ้มต่าง ๆ ขึ้น */
  /* แฟ้มสรุปผลการเรียน(record file) = SSSI.II.STS */
  strcat(StatusFile,CourseName);
  strcat(StatusFile,StudentNo);
  strcat(StatusFile, ".STS");
  i = 0;
  Max_Itm = 0;
  if ((fp1 = fopen(StatusFile,"rb")) == NULL)
  { gotoxy(25,15); printf("ไม่มีข้อมูลการเข้าเรียนของนักเรียนที่ระบุ\n");
    getch(); exit(1);
  }
  while(fread(&StatusData,sizeof StatusData,1,fp1)==1)
  { if (ferror(fp1))
    { gotoxy(15,16);printf("การอ่านแฟ้มบันทึกการเรียนของนักเรียนมีปัญหา\n");
      getch(); exit(1);
    }
    strcpy(HistoryTBL[i].fname,StatusData.fname);
    HistoryTBL[i].itm_no = StatusData.itm_no;
    strcpy(HistoryTBL[i].obj_name ,StatusData.obj_name);
    strcpy(HistoryTBL[i].l_date ,StatusData.l_date);
    strcpy(HistoryTBL[i].s_time ,StatusData.s_time);
    HistoryTBL[i].ut_time = StatusData.ut_time;
  }
}

```



```

HistoryTBL[i].itm_right = StatusData.itm_right;
HistoryTBL[i].itm_wrong = StatusData.itm_wrong;
HistoryTBL[i].itm_non = StatusData.itm_non;
HistoryTBL[i].fullscore = StatusData.fullscore;
HistoryTBL[i].score = StatusData.score;
strcpy(HistoryTBL[i].e_law ,StatusData.e_law);
HistoryTBL[i].evaluate = StatusData.evaluate;
if (Max_Itm < HistoryTBL[i].itm_no) Max_Itm = HistoryTBL[i].itm_no;

/*- เพิ่มหัวข้อ ----*/
strcpy(ItmData[StatusData.itm_no].fname,StatusData.fname);
/* ชื่อเพิ่มหัวข้อ */
strcpy(ItmData[StatusData.itm_no].obj_name,StatusData.obj_name);
/* ชื่อหัวข้อ */

i = i+1;
}
His_Time = i-1;
fclose(fp1);
}
/*-----
ShowStatus ฟังก์ชันสำหรับแสดงประวัติการเรียน(HistoryTBL) เพื่อให้เลือก
หัวข้อที่จะเข้าเรียน
input คือ ตารางข้อมูลประวัติการเรียน(HistoryTBL)
output คือ ลำดับหัวข้อที่จะเข้าเรียน (ItmPtr)
-----*/
void ShowStatus(void)
{int i, j, k, m, x;
int count, flag;
char InputItm[6];
clrscr();
printf("สรุปผลการเรียนตามลำดับการเข้าเรียนจำนวน %d ครั้ง\n",His_Time+1);
printf("วิชา %s ของนักเรียนรหัส %s มีดังนี้\n",CourseName,StudentNo);
printf("=====");
for(i = 0;i <= His_Time;i++)
{ gotoxy(1,4);
for(k = 0;(k < 3)&&(i<= His_Time);k++)
{printf("\nครั้งที่ %d หัวข้อที่ %d ชื่อเพิ่ม %s\n",
i+1, HistoryTBL[i].itm_no, HistoryTBL[i].fname);
printf("เรื่อง %s \n", HistoryTBL[i].obj_name);
}
}
}

```

```

printf("เรียนเมื่อ %s(%s) รวมใช้เวลา %ld วินาที\n",
      HistoryTBL[i].l_date, HistoryTBL[i].s_time, HistoryTBL[i].ut_time);
printf(" ตอบถูก %d ข้อ ผิด %d ข้อ ไม่ถูกไม่ผิด %d ข้อ คะแนนเต็ม %d คะแนน\n",
      HistoryTBL[i].itm_right, HistoryTBL[i].itm_wrong,
      HistoryTBL[i].itm_non, HistoryTBL[i].fullscore);
printf(" ทำได้ %d คะแนน ใช้เกณฑ์ %s ได้ระดับคะแนน %c\n",
      HistoryTBL[i].score, HistoryTBL[i].e_law,
      HistoryTBL[i].evaluate);
i++;
}
if (i <= His_Time)
{ gotoxy(1,21);
  printf("=====\n");
  printf("          กดแป้นใด ๆ เพื่อลบล้างการเข้าเรียนครั้งต่อไป");
  getch();
  for (m =4;m<25;m++){ gotoxy(1,m); clrhol();}
}
i = i-1;
}
flag = 1; x = 0;
while (flag == 1)
{gotoxy(1,21);
  printf("=====\n");
  printf("โปรดใส่เลขหัวข้อที่ต้องการเรียกดูเหตุการณ์การเรียน(1-%d)\n",Max_Itm);
  printf(" หรือ กด 999 เลิกการใช้งานโปรแกรม      :");
  if (x >= 3)
    { clrscr();
      printf("\n          ใส่เลขหัวข้อไม่ถูกต้อง ยุติการใช้งาน\n");
      exit(1);}
  x = x+1;
  gets(InputItm);
  if (strcmp(InputItm,"999") == 0)
    { printf("\n          โปรดกด Y เพื่อยืนยันว่าต้องการเลิกใช้งาน\n");
      if (toupper(getch()) == 'Y') EndTitle();
    }
  else
    if (strlen(InputItm) == 1)
      { j = InputItm[0];
        if ((j - 48 <= Max_Itm) && (j - 48 > 0))
          { ItmPtr = j - 48; flag = 0;}

```

```

else
    { gotoxy(30,24); printf("เลขที่ใส่ไม่ถูกต้อง"); delay(2000);
      gotoxy(1,24); clrscr();}
}
else
    { gotoxy(30,24); printf("เลขที่ใส่ไม่ถูกต้อง"); delay(2000);
      gotoxy(1,24); clrscr();}
}
}
}

/*-----*/
CreatAndOpenRecordFileName ฟังก์ชันสำหรับสร้างชื่อแฟ้มบันทึกพฤติกรรมการเรียน
และเปิดใช้งาน แฟ้มนี้จะสร้างเมื่อได้เลือกแฟ้มบทเรียนแล้ว
= (HistoryTBL[ItmPtr].fname - ".txt" + ".StudentNo
input คือ ข้อมูลจากการเลือกหัวข้อที่ต้องการเรียน
output คือ ชื่อแฟ้มบันทึกการเรียน(RecordFile)
-----*/

void CreatAndOpenRecordFileName(void)
{int len, i,x;
FILE *fp;
strcpy(RecordFile,ItmData[ItmPtr].fname);
len = strlen(RecordFile);
RecordFile[len-1] = StudentNo[2];
RecordFile[len-2] = StudentNo[1];
RecordFile[len-3] = StudentNo[0];
/* เปิดแฟ้ม อ่านข้อมูลในแฟ้ม */
if ((fp = fopen(RecordFile,"rb")) == NULL)
    { clrscr();
      gotoxy(15,15);
      printf("ไม่มีแฟ้มข้อมูลพฤติกรรมการเรียนของนักเรียนที่ระบุ\n");
      getch(); exit(1);
    }
clrscr();
x = 0;
printf("พฤติกรรมการเรียนตามลำดับการเข้าใช้กรอบการเรียน\n");
printf("วิชา %s ของนักเรียนรหัส %s\n",CourseName,StudentNo);
printf("หัวข้อที่ %d เรื่อง %s มีดังนี้\n",ItmPtr,ItmData[ItmPtr].obj_name);
printf("=====");
while(fread(&record,sizeof record,1,fp)==1)
    { if (ferror(fp))
      { gotoxy(15,16);printf("การอ่านแฟ้มบันทึกพฤติกรรมการเรียนของนักเรียนมีปัญหา\n");
        getch(); exit(1);

```

```

}
gotoxy(1,5);
if (record.f_no == 0) {printf("\007 \007 \007"); x = 0;}
clreol();printf("ลำดับเฟรมที่ %d\n",x);x = x+1;
clreol();printf("หมายเลขเฟรม: %d ชนิด: %c จำนวนเฟรมคำถามสะสม: %d\n"
,record.f_no, record.f_type, record.f_qno);
clreol();printf("จำนวนครั้ง ที่ยอมให้เข้าใช้: %d ที่เคยเข้าใช้: %d\n"
,record.loop_c ,record.loop_u);
clreol();printf("เวลา\n");
clreol();printf("จำนวนเวลาสะสมที่ผ่านมาในบทเรียน: %ld วินาที\n",record.ti_lu);
clreol();printf("เวลาเริ่มเข้าใช้เฟรม : %s\n",strctime(record.ti_ff));
clreol();printf("จำนวนเวลาทั้งหมดที่ใช้ในเฟรม : %ld วินาที\n",record.ti_fu);
clreol();printf(" - กำหนดแสดงข้อความ: %ld วินาที ใช้จริง: %ld วินาที\n"
,record.ti_fs, record.ti_fsu);
clreol();printf(" - กำหนดให้ใช้ตอบคำถาม: %ld วินาที ใช้จริง: %ld วินาที\n"
,record.ti_fa, record.ti_fau);
clreol();printf(" - กำหนดแสดงข้อความโต้ตอบ: %ld วินาที ใช้จริง: %ld วินาที \n\n"
,record.ti_fb, record.ti_fbu);
clreol();printf("คำตอบของผู้เรียน : %s\n",record.ans_s);
clreol();printf("ผลลัพธ์จากคำตอบ : %c\n",record.ans_r);
clreol();printf("หมายเลขเฟรมต่อไป : %d\n",record.next_f);
clreol();printf("จำนวนข้อสอบ ถูกสะสม: %d ผิดสะสม: %d ไม่มีถูกผิดสะสม : %d\n"
,record.no_r, record.no_w, record.no_n);
clreol();printf("คะแนน\n");
clreol();printf("ทำได้ในเฟรมนี้: %d ได้สะสม: %d เต็มสะสม: %d เต็มของเฟรมนี้: %d\n"
,record.sc_g,record.sc_u, record.sc_t, record.sc_f);
printf("\n <<<<< เคาะเป็นไต ๆ เพื่อดูพฤติกรรมในเฟรมถัดไป >>>>>");
getch();
}
printf("\007");
fclose(fp);
}
/*-----*/
EndTitle ฟังก์ชันสำหรับแสดงข้อความจบการใช้งานระบบ
input คือ -
output คือ ข้อความแสดงการจบการใช้งานบนจอภาพ
-----*/
void EndTitle(void)
{ clrscr();
gotoxy(25,10); printf("+-----+\n");

```

```
gotoxy(25,11); printf("      · ขอขอบคุณที่ใช้บริการ      |\n");
gotoxy(25,12); printf("      | ระบบวิเคราะห์ผลการเรียนการสอน ผู้สอน |\n");
gotoxy(25,13); printf("      |                  สวัสดิ์                  |\n");
gotoxy(25,14); printf("      +-----+|\n");
exit(1);
}
```

ภาคผนวก ข.

ภาษา TAL (TEACH Authoring Language)

เพื่อความสะดวกในการศึกษาภาษา TAL จึงได้จัดทำสารบัญชดังนี้

สารบัญ

	เรื่อง	หน้า
1	แบบฟอร์มลวงรหัส.....	124
2	คำสั่งวน.....	124
3	โครงสร้างของกรอบการเรียน.....	124
4	คำสั่ง.....	125
	4.1 คำสั่ง FRAME.....	125
	4.2 คำสั่ง SCREEN.....	126
	4.3 คำสั่ง INPUT.....	127
	4.4 คำสั่ง BRANCH.....	127
	4.5 คำสั่งพิเศษ EVALUATE.....	128

ภาษา TAL (Teach Authoring Language)

1. แบบฟอร์มลวงรหัส

การเขียนบทเรียน โปรแกรมด้วยภาษา TAL สามารถเขียนได้ดังนี้

- 1.1 เริ่มเขียนคำสั่งที่คอลัมน์ที่ 1
- 1.2 คำสั่งหรือข้อความในแต่ละบรรทัดจะมีความยาวไม่เกิน 100 ตัวอักษร
- 1.3 บทเรียนโปรแกรมหนึ่งบท จะแบ่งแยกเนื้อหาในบทเรียนเป็นส่วนย่อย ๆ เรียกว่า กรอบการเรียน (FRAME) โดยในแต่ละบทเรียนมีความยาวได้ไม่เกิน 500 บรรทัดหรือ 50 กรอบ
- 1.4 กรอบการเรียนแต่ละกรอบจะต้องมีโครงสร้างตามข้อ 3
- 1.5 กรอบการเรียนแต่ละกรอบ ควรกำหนดการจัดเรียงให้เป็นไปตามลำดับที่ต้องการควบคุม เพื่อสะดวกต่อการนำเสนอบทเรียนและบันทึกผลการเรียนของผู้เรียน

2. คำสงวน

คำและเครื่องหมายต่อไปนี้ ถ้าไม่ใช่เป็นคำสั่ง สงวนไม่ให้พิมพ์ในคอลัมน์ที่ 1

FRAME SCREEN INPUT BRANCH EVALUATE \$T @ ' " ^

3. โครงสร้างของกรอบการเรียน

โครงสร้างกรอบการเรียน มีลักษณะดังนี้

```
FRAME #no. [,frame_type] [,no_loop]
SCREEN ['label'] [,wait_time]
[string]
[["]string]
[[']string]
[[^]string]
[@([line] [,col])] ['string']
$C
$T#no

INPUT [@([line][,col])[message'][,wait_time]
BRANCH [full_score][,wait_time]
['value' [condition]] [:frame_no][, 'string'/message]
```

หมายเหตุ คำสั่งหรือข้อความภายในเครื่องหมายวงเล็บใหญ่ หมายถึงจะมีหรือไม่มีก็ได้

กรอบการเรียนแต่ละกรอบที่เป็นชนิดบรรยายและชนิดคำถาม จะเริ่มต้นด้วยคำสั่ง FRAME และตามด้วยคำสั่ง SCREEN บรรทัดต่อจาก SCREEN จะเป็นข้อความแสดงผลบนจอภาพ คำสั่งต่อจากนี้อาจเป็นคำสั่ง INPUT หรือ BRANCH หรือไม่มีทั้งคู่ก็ได้ ถ้ามีคำสั่ง BRANCH จะต้องมีคำสั่งย่อยของ BRANCH ตามอย่างน้อย 1 บรรทัด ต่อจากนั้นจะเป็นกรอบการเรียนถัดไป

แต่ถ้ากรอบการเรียนนั้นเป็นชนิดสุ่ม บรรทัดต่อจากคำสั่ง FRAME จะต้องเป็นชุดหมายเลขกรอบที่จะถูกสุ่มขึ้นใช้งาน มีรูปแบบดังนี้

FRAME #no.,R
([nos-noe][,no])

จะเห็นได้ว่า ภาษา TAL ประกอบด้วยคำสั่งหลัก 4 คำสั่ง คือ

FRAME สำหรับบอกหมายเลขกรอบ ชนิดของกรอบ จำนวนครั้งที่ยอมให้เรียกใช้
SCREEN สำหรับกำหนดการเริ่มต้นแสดงข้อมูลออกทางจอภาพ
INPUT สำหรับการรอรับข้อมูลจากผู้ใช้บทเรียน
BRANCH สำหรับการแตกแขนงไปแสดงกรอบอื่น ๆ
และคำสั่งพิเศษอีก 1 คำสั่ง คือ
EVALUATE สำหรับใช้กำหนดเกณฑ์การประเมินผลการเรียน

4. คำสั่ง

คำสั่งที่ใช้ในภาษา TAL มีรูปแบบและความหมาย ดังต่อไปนี้

4.1 คำสั่ง FRAME

เป็นคำสั่งใช้บอกหมายเลขกรอบ ชนิดของกรอบ จำนวนครั้งที่ยอมให้ใช้งาน รูปแบบของคำสั่ง

FRAME #no [,frame_type] [,no_loop]

ตัวอย่าง

FRAME 3,Q,2 หมายถึงกรอบหมายเลข 3 เป็นชนิดคำถาม
ยอมให้เรียกใช้งานได้ไม่เกิน 2 ครั้ง

#no - หมายเลขกรอบ มีได้ตั้งแต่ 1-99 สำหรับ 1 แฟ้มเนื้อหา
(1 แฟ้มเนื้อหาไม่ควรเรียกใช้กรอบเกิน 50 กรอบหรือ 1 แฟ้มเนื้อหาควรใช้เวลาเรียนประมาณ 20 นาที)

frame type - ชนิดของกรอบแบ่งได้เป็น 3 ชนิด คือ

E(Explanation) กรอบชนิดคำบรรยาย สำหรับใช้อธิบายเนื้อหา

Q(Question) กรอบชนิดคำถาม

R(Random) กรอบชนิดสุ่มเลือกกรอบอื่นมานำเสนอ โดยจะสุ่มตัวเลขจากกลุ่มตัวเลขที่อยู่ในบรรทัดต่อจากคำสั่ง FRAME กลุ่มตัวเลขนี้กำหนดให้มีได้ไม่เกิน 20 จำนวนเขียนในลักษณะพิสัยหรือเลขโดดก็ได้ ตัวอย่างเช่น

FRAME 2,R,1

(1-4, 8, 10)

หมายถึง กรอบหมายเลข 2 เป็นชนิดสุ่ม ถูกเรียกใช้งานได้หนึ่งครั้ง โดยเอากรอบหมายเลข 1,2,3,4,8,10 กรอบหนึ่งกรอบใดนำมาเสนอ

*ข้อควรระวัง ในการใช้กรอบชนิดสุ่ม

- กรอบชนิดนี้จะไม่คำสั่งหลักอื่นตามด้วย เพราะเป็นการสุ่มกรอบอื่นมาแทนที่กรอบปัจจุบัน
- กรอบที่ถูกสุ่มควรกำหนดการแตกแขนง (มีคำสั่ง BRANCH) ให้เหมาะสม เพื่อไม่ให้ลำดับ

ขั้นตอนการสอนสลับส่น และในกรอบที่ถูกส่มควรกำหนดการเข้าใช้ เป็น 1 ถ้าไม่ต้องการให้ถูกส่มซ้ำ
 - กรณีไม่มีการกำหนดทิศทางของกรอบ โปรแกรมจะเรียกกรอบที่อยู่ถัดจากกรอบปัจจุบันมาทำงาน

- ห้ามส่มเลือกตัวเอง

- ถ้ากรอบที่ถูกส่มใช้หมดแล้ว โปรแกรมจะข้ามไปทำงานกรอบถัดไปที่

กรณีผู้สร้างบทเรียน ไม่ระบุชนิดของกรอบ และจำนวนครั้งที่ยอมให้เรียกใช้ ถือว่าเป็นกรอบชนิดบรรยาย ยอมให้เรียกใช้ได้ ไม่จำกัดจำนวนครั้ง

no_loop - จำนวนครั้งที่ยอมให้เรียกใช้ ถ้าใช้ครบจำนวนครั้งแล้วจะผ่านไป
 กรอบถัดไปที่

4.2 คำสั่ง SCREEN

เป็นคำสั่งกำหนดการเริ่มต้นแสดงข้อมูลทางจอภาพ

รูปแบบของคำสั่ง

```
SCREEN ['label'] [,wait_time]
```

```
[string]
```

```
[["]string]
```

```
[[']string]
```

```
[[^]string]
```

```
@([line] [,col]) ['string']
```

```
$C
```

```
$T#no
```

เมื่อ label เป็นข้อความที่จะนำเสนอในบรรทัดแรกกลางจอภาพ

wait_time จำนวนเวลาที่กำหนดให้แสดงข้อมูลบนจอภาพ โดยเริ่มจับเวลาเมื่อข้อความภายใต้คำสั่ง SCREEN ถูกนำเสนอหมด และจะผ่านไปดำเนินการสอนต่อไปเมื่อหมดเวลาที่กำหนด หรือผู้เรียนกด Enter

ข้อความในบรรทัดต่อจากคำสั่ง SCREEN จะเป็นข้อความ[string]ที่ต้องการให้ถูกนำเสนอออกทางจอภาพ ตามลักษณะที่เห็นในบทเรียน ถ้าไม่มีการกำหนดตำแหน่งแสดงข้อความโดยคำสั่งย่อย @(line,col) ข้อความนั้นไม่ต้องอยู่ระหว่างเครื่องหมาย '.....' และสามารถใช้อัตวนำหน้า " หรือ ' หรือ ^ เพื่อจัดข้อความชิดซ้ายหรือชิดขวาหรือกึ่งกลางบรรทัดได้ตามลำดับ

หากมีการใช้คำสั่งย่อยกำหนดตำแหน่งแสดงข้อความ ข้อความจะแสดงบนจอภาพในตำแหน่งที่กำหนด และหากมีข้อความในบรรทัดตามมา ข้อความนั้นจะแสดงในบรรทัดต่อบรรทัดที่กำหนดไว้ การจัดข้อความเพื่อแสดงออกทางจอภาพ มีคำสั่งย่อยสำหรับอำนวยความสะดวกดังนี้ @ line, col 'string' กำหนดตำแหน่งการแสดงผลข้อความ(string)บนจอภาพ

\$C ลบข้อความทั้งหมดบนจอภาพ กำหนดตำแหน่งแสดงข้อความใหม่ ณ บรรทัดที่ 1

\$T#no ถ่วงเวลาตามจำนวนหมายเลขที่ให้ไว้ (วินาที)

4.3 คำสั่ง INPUT

เป็นคำสั่งสำหรับรอรับการโต้ตอบจากผู้ใช้งาน

รูปแบบของคำสั่ง

```
INPUT @([line][,col])[,'message'][,wait_time]
```

เมื่อ @([line][,col]) กำหนดตำแหน่งแสดงข้อความรอรับการโต้ตอบ

['message'] ข้อความ

[,wait_time] กำหนดเวลาในการรอการโต้ตอบ(\$T) ถ้าหมดเวลาหรือผู้เรียนกด Enter จะผ่านไปยังกรอบถัดไป

ตัวอย่าง

```
INPUT @ (8, 3) 'โปรดเลือก' , $T10
```

4.4 คำสั่ง BRANCH

เป็นคำสั่งสำหรับตรวจสอบคำตอบ และให้ข้อความโต้ตอบ แล้วแตกแขนงไปกรอบอื่น โดยข้อความโต้ตอบจากบทเรียนไปยังผู้เรียนในส่วนนี้จำกัดให้มีได้ไม่เกิน 80 ตัวอักษร

รูปแบบของคำสั่ง

```
BRANCH [full_score][,wait_time]
```

```
['value'[condition]] [:frame_no] [, 'string'/message]
```

เมื่อ full_score กำหนดคะแนนเต็มเมื่อเงื่อนไขเป็น C

wait_time กำหนดเวลาในการรอผู้เรียนอ่านข้อความโต้ตอบ(\$T) ถ้าหมดเวลา หรือผู้เรียนกด Enter จะผ่านไปยังการสอนอื่นถัดไป

['value'[condition]] value คือ คำตอบของผู้เรียนหรือข้อมูลที่ผู้เรียนป้อนให้เครื่อง

condition คือ การกำหนดค่าคะแนนที่ให้ ความถูก ผิด หรือไม่ถูก

ไม่ผิดสำหรับ value ที่ผู้เรียนป้อน สามารถกำหนดได้ดังนี้

V#no - ให้คะแนน #no คะแนน

C - ถูก จะได้คะแนนเต็ม

W - ผิด จะได้ 0 คะแนน

N - ไม่ถูกไม่ผิด

frame_no หมายเลขกรอบที่จะถูกเรียกมาทำงานต่อไป ถ้าเว้นไว้จะเรียกกรอบที่อยู่ ถัดจากกรอบนี้มาทำงาน

'string'/message ข้อความโต้ตอบกลับ จะใช้ข้อความ(string) หรือ \$M#no ซึ่ง

เป็นข้อความจากแฟ้มโต้ตอบก็ได้ ข้อความนี้จะแสดงกึ่งกลาง

บรรทัด และมีความยาวไม่เกิน 80 ตัวอักษร

ถ้าไม่มีการกำหนดหมายเลข FRAME ที่จะไป รวมทั้งไม่มีคำสั่ง BRANCH โปรแกรม จะกำหนดให้ไปยังกรอบที่อยู่ถัดจากกรอบปัจจุบันเสมอ

แฟ้มข้อมูลข้อความโต้ตอบ แฟ้มนี้มีชื่อเฉพาะว่า CAICOM.MSG จัดประสงค์เพื่อเป็นแฟ้ม ข้อมูลสำหรับผู้สอนสร้างข้อความโต้ตอบที่ใช้อยู่เก็บไว้เลือกไปใช้งาน ฉะนั้นจะใช้ข้อความโต้ตอบจากแฟ้มนี้ได้เมื่อมีการสร้างไว้ก่อนหน้าแล้ว โดยใช้โปรแกรมประมวลคำมีรูปแบบดังนี้

หมายเลขรหัสเรียก ข้อความโต้ตอบ

โดยมีข้อกำหนดให้มีข้อความได้ไม่เกิน 50 ข้อความ ข้อความละ 100 อักขระ
 การเรียกใช้ข้อความเรียกโดยใช้คำสั่งใน \$M#no (เมื่อ #no = หมายเลขรหัสเรียก)
 ตัวอย่าง

- 1 เก่งมาก
- 2 ถูกต้อง
- 4 ดมมาก

4.5 คำสั่งพิเศษ EVALUATE เกณฑ์การประเมินผลการเรียนว่าผ่านการเรียนหรือไม่
 ภายใต้ TEACH AUTHORIZING SYSTEM ทั่วไปจะกำหนดให้ผ่านบทเรียนได้ด้วย
 คะแนนไม่น้อยกว่า 60% ของคะแนนเต็ม และได้รับเกรดการเรียนดังนี้

- ได้คะแนน 80%-100% ให้เกรด A
- ได้คะแนน 70%-79% ให้เกรด B
- ได้คะแนน 60%-69% ให้เกรด C
- ได้คะแนน 50%-59% ให้เกรด D
- ได้คะแนน 0%-49% ให้เกรด E

ในการเปลี่ยนแปลงเกณฑ์การประเมินผล สามารถเปลี่ยนแปลงได้โดยใช้คำสั่ง EVALUATE
 โดยมีรูปแบบคำสั่งดังนี้

EVALUATE (เกรดที่1:คะแนนต่ำสุดของเกรดที่1, เกรดที่2:คะแนนต่ำสุดของ
 เกรดที่2,, เกรดที่ก:คะแนนต่ำสุดของเกรดที่ก,)

ตัวอย่าง

EVALUATE(A:80,B:70,C:60,D:50,E:0)

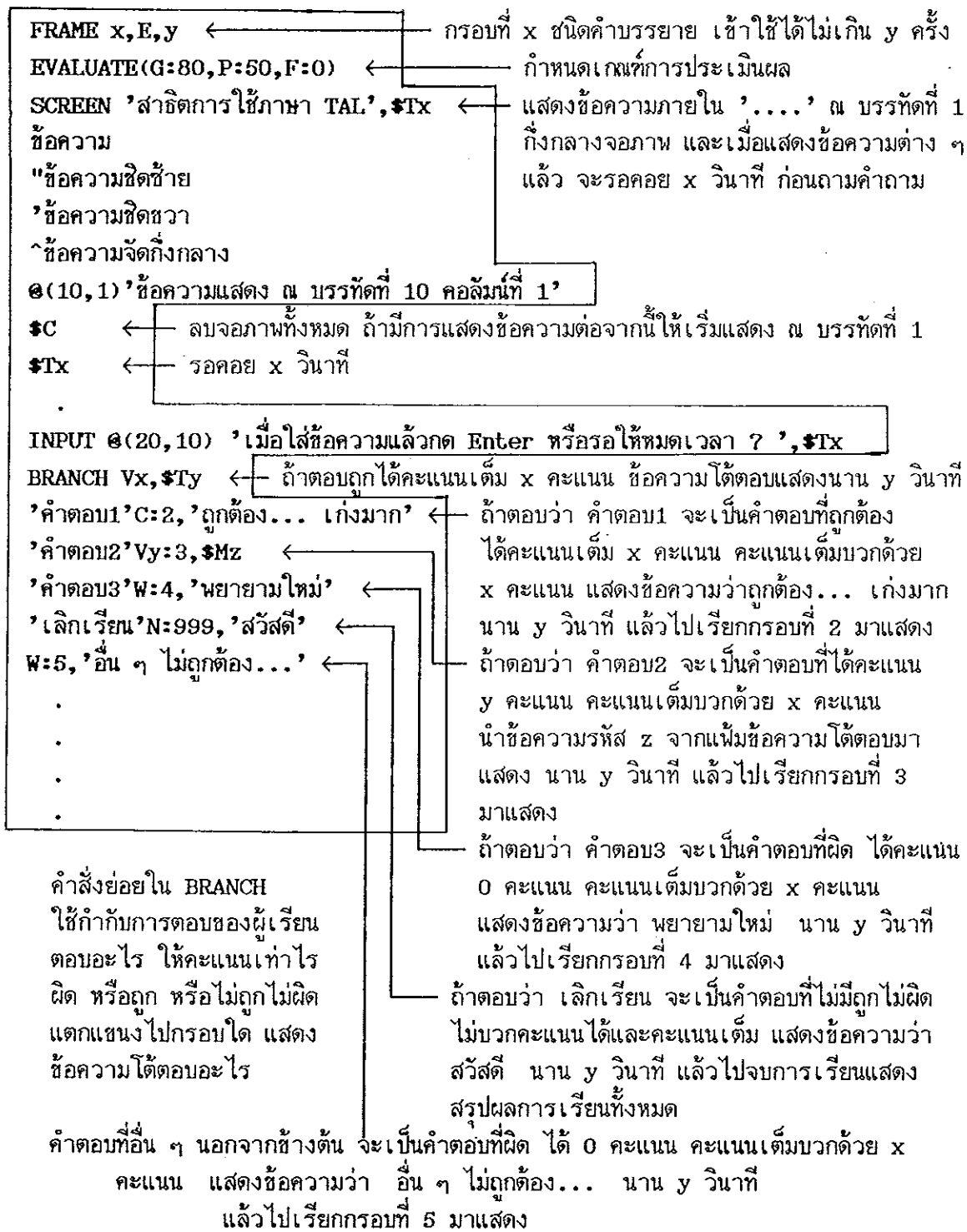
EVALUATE(A:90,B:80,C:70,D:60,E:0)

EVALUATE(G:80,P:50,F:0)

คำสั่ง EVALUATE จะควบคุมการประเมินผลเฉพาะหัวข้อการเรียนนั้น ๆ
 การกำหนดจะต้องกำหนดให้อยู่หลังคำสั่ง FRAME ซึ่งเป็นกรอบที่ผู้เรียนจะต้องใช้ เพื่อไม่ให้
 เกิดความผิดพลาดควรใส่ไว้ตั้งแต่เริ่มต้นบทเรียน

หมายเหตุ

1. ในการเรียนหรือทดสอบสามารถเลือกกรอบการเรียนหมายเลขใดก็ได้ โดยใส่คำ
 ตอบเป็น //FRAME#no เมื่อ #no คือ หมายเลขกรอบ
2. ถ้าต้องการหยุดการทดสอบที่กรอบใด ๆ ให้ใส่คำตอบเป็น //END
3. การควบคุมบทเรียนให้เป็นแบบเส้นตรง หรือแบบแตกแขนง หรือแบบสุ่มเลือก
 สามารถกระทำได้โดยการควบคุมการแตกแขนงในคำสั่ง BRANCH หรือเลือกใช้กรอบชนิด
 สุ่มเลือก
4. ตัวอย่างกรอบการเรียนเต็มรูปแบบ 1 กรอบพร้อมคำอธิบายผลลัพธ์ เป็นดังนี้



ภาคผนวก ค.

แบบฟอร์มออกแบบกรอบการเรียนรู้ TEACH

แบบฟอร์มออกแบบกรอบการเรียน TEACH

* ใช้ช่วยออกแบบบทเรียน โปรแกรมสำหรับระบบ TEACH 1 กรอบการเรียน

FRAME __, [JE []Q []RC _____], _____
 SCREEN ' _____ ', \$T _____

[' = ซิดซ้าย, " = ซิดขวา, ^ = กึ่งกลาง, @ (L,C) ' _____ ' = กำหนดตำแหน่ง
 \$C = ลบจอ, \$T _____ = หน่วงเวลา]

บรรทัดที่ _____ (ส่วนข้อความที่ปรากฏบนจอภาพ)

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	

INPUT @ (__, __) ' _____ ', \$T _____

BRANCH V __, \$T _____

['value' [condition]] [:frame.no][, 'string'/message]

' _____ ' C W N V __ : __, ' _____ '
 ' _____ ' C W N V __ : __, ' _____ '
 ' _____ ' C W N V __ : __, ' _____ '
 ' _____ ' C W N V __ : __, ' _____ '
 ' _____ ' C W N V __ : __, ' _____ '

ภาคผนวก ง.

เอกสารคู่มือ

การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

เพื่อความสะดวกในการศึกษาเอกสารคู่มือนี้ จึงได้จัดทำสารบัญขึ้นดังนี้

สารบัญ

เรื่อง	หน้า
1 คำนำ.....	133
2 จุดมุ่งหมายของ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์.....	133
3 ลักษณะของระบบ TEACH และประโยชน์ในการนำไปใช้งาน.....	133
4 อุปกรณ์ที่จำเป็นสำหรับระบบ TEACH	135
4.1 Hardware.....	135
4.2 Software.....	135
5 การใช้งานระบบ.....	136
5.1 การสร้างบทเรียน.....	136
5.2 การเรียน.....	138
5.3 การติดตามการเรียน.....	141

คู่มือการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

1. คำนำ

TEACH เป็นระบบโปรแกรมช่วยการเรียนการสอนแบบอัตโนมัติ (โปรแกรมที่ผู้สอนสามารถสร้างบทเรียนบนไมโครคอมพิวเตอร์ได้ด้วยตนเอง) พัฒาระบบโดย นายนิพนธ์ ศุกศรี เป็นโครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536 โดยมี อาจารย์วุฒินงค์ เตชะดำรงสิน เป็นประธานกรรมการที่ปรึกษา และรศ.ไพโรจน์ ตรีธนากุล เป็นกรรมการที่ปรึกษา

2. จุดมุ่งหมายของ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ระบบ TEACH หรือระบบโปรแกรมผู้สอน พัฒนาขึ้นโดยมีจุดมุ่งหมายเพื่อเป็นเครื่องมืออำนวยความสะดวกแก่บุคคลที่ต้องการสร้างบทเรียนบนเครื่องไมโครคอมพิวเตอร์ (บทเรียนสำหรับให้ผู้เรียนใช้เรียนด้วยตนเองบนเครื่องไมโครคอมพิวเตอร์) โดยผู้สร้างบทเรียนไม่จำเป็นต้องเป็นนักเขียนโปรแกรมมาก่อนก็สามารถสร้างบทเรียนได้ เพียงแต่ศึกษารูปแบบเพิ่มข้อมูลที่ให้กับระบบ วิธีการใช้งานโปรแกรม และคำสั่งซึ่งเป็นภาษาที่ได้ออกแบบไว้ชื่อ TEACH Authoring Language หรือ TAL

ภาษา TAL นี้ มีคำสั่งง่าย ๆ เพียงไม่กี่คำสั่ง เช่น FRAME SCREEN INPUT BRANCH เป็นต้น สามารถใช้เวลาสั้น ๆ ในการศึกษาใช้งาน ภาษานี้ทำหน้าที่ควบคุมการนำเสนอบทเรียน ซึ่งจะทำให้ผู้สอนสามารถใส่เทคนิคการสอนตามที่ต้องการได้ ดังนั้น บทเรียนสำหรับใช้กับระบบนี้จึงประกอบด้วยภาษา TAL ซึ่งผู้สอนสามารถสร้างได้ง่าย ๆ โดยจัดนิมพ์บทเรียนเป็นแฟ้มข้อมูลด้วยโปรแกรมประมวลคำ (word processor) และเมื่อจะใช้งานบทเรียนก็สั่งให้บทเรียนทำงานโดยผ่านโปรแกรม TEACH

บทเรียนที่สร้างสำเร็จแล้วจะประกอบด้วย แฟ้มโครงสร้างบทเรียน ซึ่งเป็นแฟ้มที่ไว้บอกว่าเป็นบทเรียนชุดนี้มีกี่หัวข้อ แต่ละหัวข้อชื่ออะไร และแฟ้มเนื้อหาวิชา ซึ่งเป็นแฟ้มบรรจุเนื้อหาของแต่ละหัวข้อ โดยเนื้อหาของบทเรียนจะถูกกำกับด้วยคำสั่งภาษา TAL ซึ่งจะทำหน้าที่ควบคุมการนำเสนอบทเรียนเมื่อนำไปใช้กับระบบ TEACH

นอกจากการนำเสนอบทเรียนแล้ว ระบบ TEACH ยังทำหน้าที่จัดการการเรียน โดยการบันทึกสรุปผลการเรียน บันทึกพฤติกรรมการเรียนของผู้เรียน และจัดให้ผู้เรียนเข้าศึกษาในตำแหน่งที่ต่อเนื่องจากครั้งก่อน หรือตามความประสงค์ของผู้เรียน

ประโยชน์ที่ได้รับจากระบบนี้คือ ผู้เรียนได้เรียนบทเรียนที่มีปฏิริยากับผู้เรียนแต่ละคน โดยเฉพาะ ผู้สอนสามารถติดตามประเมินผลบทเรียน ศึกษาพฤติกรรมการเรียนและผลการเรียนของผู้เรียน ได้จากข้อมูลที่ระบบบันทึกไว้

3. ลักษณะของ TEACH และประโยชน์ในการนำไปใช้งาน

ระบบนี้มีลักษณะเป็นระบบโปรแกรมช่วยการเรียนการสอนแบบอัตโนมัติ (โปรแกรมที่อำนวยความสะดวกให้ผู้สอนสามารถสร้างบทเรียนบนไมโครคอมพิวเตอร์ได้ด้วยตนเอง) ชนิดกรอบการเรียน (FRAME) กล่าวคือบทเรียนที่จะนำมาใช้กับระบบจะถูกจัดแบ่งออกเป็นส่วนย่อย ๆ ที่เรียกว่ากรอบ (FRAME) แต่ละกรอบจะมีขนาดข้อมูลประมาณ 1 หน้าจอ (24 บรรทัด) หรือ

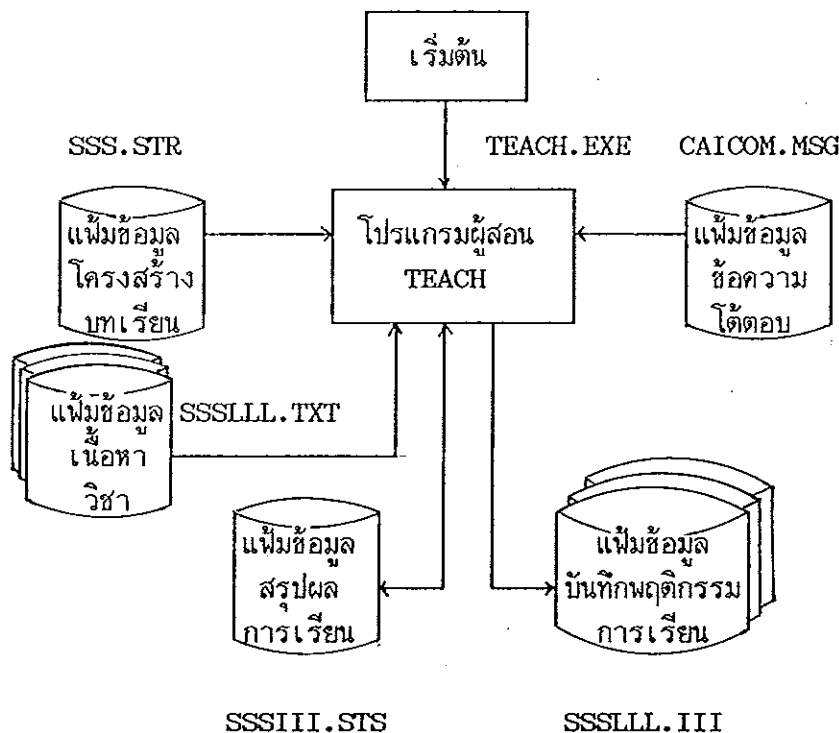
น้อยกว่าเพื่อสะดวกแก่การศึกษาของผู้เรียน โดยมีผังงานระบบ TEACH แยกได้เป็น 2 ส่วน คือ

ส่วนที่ 1 มีโปรแกรมหลัก TEACH เป็นโปรแกรมควบคุมจัดการการเรียนการสอน ดำเนินการสอน การโต้ตอบ บันทึกพฤติกรรมการเรียนของผู้เรียน บันทึกสรุปผลการเรียน มีเพิ่มข้อมูลที่เกี่ยวข้องในการประมวลผล 5 เพิ่มข้อมูลคือ เพิ่มข้อมูลโครงสร้างบทเรียน เพิ่มข้อมูลเนื้อหาวิชา เพิ่มข้อมูลสรุปผลการเรียน เพิ่มข้อมูลบันทึกพฤติกรรม และ เพิ่มข้อมูลข้อความโต้ตอบ

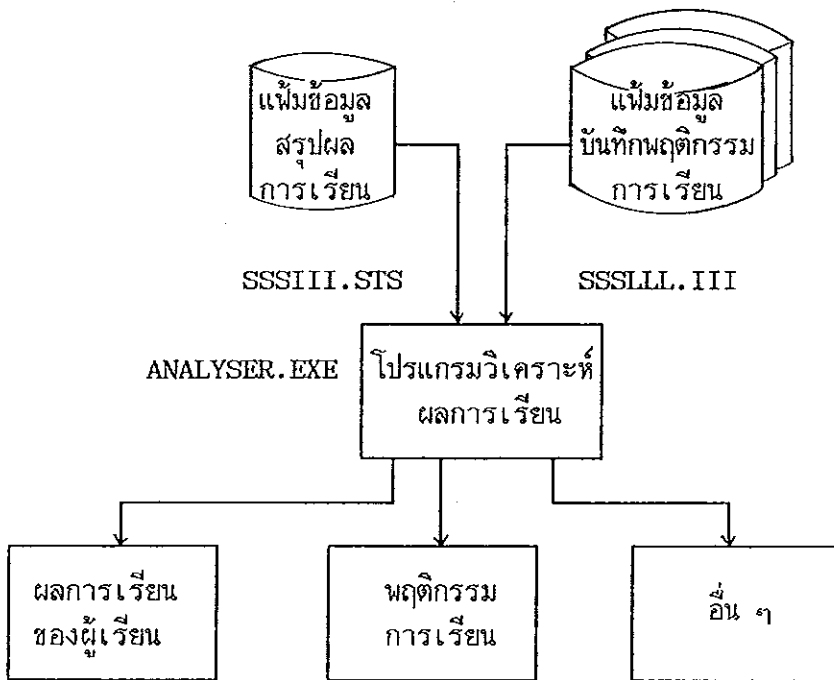
ส่วนที่ 2 มีโปรแกรมหลัก ANALYSER สำหรับให้ครูสอนใช้ดูข้อมูลสรุปผลการเรียน และดูข้อมูลพฤติกรรมการเรียนของผู้เรียน ซึ่งเป็นประโยชน์ต่อการวิเคราะห์ประสิทธิภาพของบทเรียน ให้คำปรึกษาแก่ผู้เรียน ข้อมูลที่ใช้ในส่วนนี้คือข้อมูลจากเพิ่มข้อมูลสรุปผลการเรียน และเพิ่มข้อมูลบันทึกพฤติกรรม ซึ่งเป็นผลลัพธ์จากส่วนที่ 1

ผังงานของระบบแสดงได้ดังภาพประกอบ 1 และ 2

ภาพประกอบ 1 ผังงานระบบของระบบ TEACH ส่วนที่ 1



ภาพประกอบ 2 ผังงานระบบของระบบ TEACH ส่วนที่ 2



4. อุปกรณ์ที่จำเป็นสำหรับระบบ TEACH

ในการใช้งานระบบ TEACH จำเป็นต้องมีอุปกรณ์ที่เป็น Hardware และ Software ดังนี้

4.1 Hardware

- 4.1.1 เครื่องไมโครคอมพิวเตอร์ 16 บิต (IBM PC, XT, AT หรือ Compatible) ที่มีหน่วยความจำไม่ต่ำกว่า 640 KB
- 4.1.2 Floppy disk drive 1 ตัว
- 4.1.3 PC-DOS รุ่น 3.0 ขึ้นไป หรือ MS-DOS รุ่น 3.0 ขึ้นไป

4.2 Software ระบบ TEACH มีโปรแกรมที่ใช้งานดังต่อไปนี้

- 4.2.1 TEACH.EXE
- 4.2.2 ANALYSER.EXE
- 4.2.3 THAI.COM
- 4.2.4 แฟ้มบทเรียนที่มีส่วนขยายเป็น .TXT
- 4.2.5 แฟ้มโครงสร้างบทเรียนที่มีส่วนขยายเป็น .STR
- 4.2.6 CAICOM.MSG แฟ้มข้อความโต้ตอบ
เมื่อมีการเรียนจะมีแฟ้มบันทึกข้อมูลการเรียนรู้เกิดขึ้นคือ
- 4.2.7 แฟ้มข้อมูลสรุปผลการเรียนรู้ที่มีส่วนขยายเป็น .STS
- 4.2.8 แฟ้มข้อมูลบันทึกพฤติกรรมการเรียนรู้ที่มีส่วนขยายเป็น เลขรหัสผู้เรียน

5. การใช้งานระบบ

การใช้งานระบบ จะประกอบด้วยบุคคล 2 ฝ่าย คือ ผู้สร้างบทเรียนหรือผู้สอน กับผู้เรียน ซึ่งบุคคลทั้ง 2 ฝ่ายมีงานการเรียนการสอนที่สัมพันธ์ต่อกันคือ การสร้างบทเรียน การเรียน และการติดตามการเรียน

5.1 การสร้างบทเรียน

การสร้างบทเรียนกระทำโดยผู้สร้างบทเรียนหรือผู้สอน ในที่นี้หมายถึง ครู อาจารย์ ผู้สอน หรือใครก็ตามที่มีความประสงค์จะสร้างบทเรียนสำหรับใช้สอนบนเครื่องไมโครคอมพิวเตอร์ การสร้างบทเรียนจะเริ่มต้นที่ ผู้สร้างจะต้องมีบทเรียนที่ตนเองต้องการจะสอนไว้แล้ว และมีการจัดแบ่งบทเรียนที่จะให้สอนเป็นหัวข้อ ในแต่ละหัวข้อควรแบ่งเป็นกรอบการเรียน ฉะนั้นบทเรียนหนึ่ง ๆ อาจประกอบด้วย หัวข้อเดียว หรือหลายหัวข้อก็ได้

หัวข้อในที่นี้อาจหมายถึง จุดประสงค์การเรียนหนึ่งจุดประสงค์ หรือบทเรียนหนึ่งบท หรือส่วนการเรียนหนึ่งส่วน แล้วแต่ผู้สร้างบทเรียนจะกำหนด โดยมีเงื่อนไขว่า บทเรียนหนึ่ง ๆ มี หัวข้อได้ไม่เกิน 10 หัวข้อ และแต่ละหัวข้อจะมีกรอบการเรียนได้ไม่เกิน 50 กรอบการเรียน (หรือไม่เกิน 500 บรรทัด)

กรอบการเรียน คือ ขอบเขตของเนื้อหาวิชาที่ผู้สอนต้องการจะให้ผู้เรียนเรียน เนื้อหาที่เหมาะสมสำหรับ 1 กรอบการเรียนควรนำเสนอเนื้อหา 1 แนวคิด (concept) โดย ปริมาณข้อความควรมีความยาวไม่เกิน 1 หน้าจอภาพ (ไม่เกิน 24 บรรทัด) จะทำให้ง่ายต่อการควบคุมลำดับขั้นตอนการนำเสนอ และง่ายต่อการทำความเข้าใจบทเรียนของผู้เรียน จุดประสงค์ของแต่ละกรอบขึ้นกับความต้องการของผู้สอนว่าจะกำหนดให้เป็นกรอบสำหรับสอนเนื้อหา (ชนิด E) กรอบสำหรับถามคำถาม (ชนิด Q) หรือกรอบสำหรับส่งเสริมเลือกกรอบการเรียนอื่น ๆ มานำเสนอ (ชนิด R)

การสร้างบทเรียนไปใช้กับระบบ TEACH มีขั้นตอน ดังนี้

- (1) กำหนดหัวข้อต่าง ๆ ในบทเรียน แล้วนำไปจัดทำเป็นแฟ้มโครงสร้าง
- (2) จัดทำแฟ้มเนื้อหาวิชาแต่ละหัวข้อ
- (3) ทดสอบการใช้งาน
- (4) แก้ไขและปรับปรุง
- (5) นำไปใช้งาน

ตัวอย่างการจัดทำบทเรียนเรื่องการบวกเลขเพื่อนำไปใช้กับระบบ TEACH
ขั้นตอนการจัดทำมีดังนี้

- (1) กำหนดหัวข้อต่าง ๆ ในบทเรียนแล้วนำไปจัดทำเป็นแฟ้มโครงสร้าง

หัวข้อที่กำหนดไว้	พิมพ์ด้วย WP เป็นแฟ้มโครงสร้างชื่อ MAT.STR
วิชาคณิตศาสตร์ประถม 1 1 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด 2 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด 3 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด 4 การบวกเลขสองหลักสองจำนวนที่มีตัวทด	MAT000 วิชาคณิตศาสตร์ประถม 1 MAT001 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด MAT002 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด MAT003 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด MAT004 การบวกเลขสองหลักสองจำนวนที่มีตัวทด

การตั้งชื่อแฟ้มโครงสร้างและรูปแบบข้อมูลในแฟ้มโครงสร้าง การตั้งชื่อแฟ้มโครงสร้างมีข้อกำหนดว่า ต้องประกอบด้วยอักขระ 3 ตัว และมีส่วนขยายของชื่อแฟ้มเป็น .STR

รูปแบบข้อมูลในแฟ้มโครงสร้าง มีข้อกำหนดว่า จะต้องประกอบด้วย 2 ส่วนที่แยกจากกันด้วยช่องว่าง 1 ช่อง มีรูปแบบดังนี้

ชื่อแฟ้มหัวข้อ ชื่อหัวข้อ (โดยชื่อแฟ้มที่มีรหัส 000 จะเป็นชื่อวิชา)

ชื่อแฟ้มหัวข้อคือ ชื่อแฟ้มบทเรียนของหัวข้อนั้น ชื่อจะประกอบด้วยอักขระ 3 ตัวแรกของชื่อแฟ้มโครงสร้าง ตามด้วยตัวเลข 3 ตัว ซึ่งเป็นลำดับหัวข้อ โดยไม่ต้องมีส่วนขยายชื่อแฟ้ม และกำหนดเป็นพิเศษให้ชื่อแฟ้มหัวข้อที่มีรหัสตัวเลข 000 เป็นชื่อวิชา

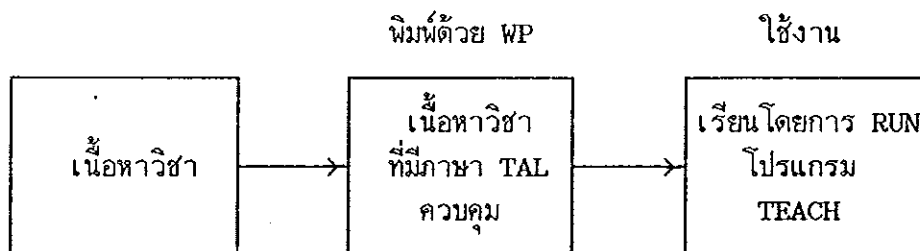
จากตัวอย่างข้างต้นสามารถตั้งชื่อแฟ้มวิชาเป็น MAT.STR

ชื่อหัวข้อ คือ ชื่อหัวข้อนั้น ๆ ในตัวอย่างได้แก่ การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด เป็นต้น

(2) จัดทำแฟ้มเนื้อหาวิชาแต่ละหัวข้อ

ลักษณะของข้อมูลในแฟ้มเนื้อหาวิชา คือ เนื้อหาของบทเรียนที่มีภาษา TAL เป็นส่วนประกอบ โดยโครงสร้างของภาษา TAL จะมีลักษณะแบ่งเนื้อหาเป็นกรอบการเรียน และช่วยในการนำเสนอบทเรียนให้เป็นไปตามที่ผู้สอนต้องการ การสร้างเนื้อหาวิชาทำได้ง่าย ๆ โดยการพิมพ์เนื้อหาวิชาที่มีภาษา TAL เป็นส่วนประกอบด้วยโปรแกรมประมวลคำทั่วไป และตั้งชื่อแฟ้มเป็นชื่อเดียวกับแฟ้มหัวข้อที่อยู่ในแฟ้มโครงสร้างให้มีส่วนขยายของชื่อแฟ้มเป็น .TXT ในภาคผนวก ค แบบฟอร์มออกแบบกรอบการเรียน TEACH จะช่วยให้สามารถออกแบบกรอบการเรียน ได้สะดวกขึ้น

จำนวนแฟ้มเนื้อหาวิชาที่จะต้องสร้างจะเท่ากับจำนวนหัวข้อที่มีในแฟ้มโครงสร้าง จากตัวอย่างแฟ้มโครงสร้าง MAT.STR จะมีแฟ้มเนื้อหาวิชาจำนวน 4 แฟ้ม ได้แก่ MAT001.TXT MAT002.TXT MAT003.TXT และ MAT004.TXT



จากการจัดแบ่งหัวข้อดังกล่าว นำหัวข้อที่ 1 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด จัดทำเป็นบทเรียนสำหรับใช้บนไมโครคอมพิวเตอร์ โดยพิมพ์เป็นแฟ้มข้อความด้วยโปรแกรมประมวลคำ และตั้งชื่อแฟ้มเป็น MAT001.TXT ดังต่อไปนี้

ตัวอย่าง การจัดทำบทเรียนเรื่องการบวกเลข
หัวข้อที่ 1 การบวกเลขหลักเดียวสองจำนวนที่
ไม่มีตัวทด เพื่อนำไปใช้กับระบบ TEACH

เนื้อหาบทเรียนที่มีภาษา TAL ควบคุม พิมพ์ด้วย WP

ผลลัพธ์เมื่อเรียนโดยการ RUN โปรแกรม TEACH

เนื้อหาบทเรียน

กรอบที่ 1 คำอธิบายเกี่ยวกับการบวก การบวกเลขสองจำนวน คือ การนำเลขทั้งสองจำนวนมารวมกัน
กรอบที่ 2 ตัวอย่างที่ 1 ตัวอย่างเช่น 1+1 มีค่าเท่ากับ 2
กรอบที่ 3 ตัวอย่างที่ 2 ตัวอย่างเช่น 2+3 มีค่าเท่ากับ 5
กรอบที่ 4 โจทย์ทดสอบที่ 1 ผลบวกของ 2 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8
กรอบที่ 5 โจทย์ทดสอบที่ 2 ผลบวกของ 4 กับ 2 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8
กรอบที่ 6 โจทย์ทดสอบที่ 3 ผลบวกของ 5 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8

FRAME 1,2 SCREEN 'กรอบที่ 1 คำอธิบายเกี่ยวกับการบวก' การบวกเลขสองจำนวน คือ การนำเลขทั้งสองจำนวนมารวมกัน INPUT #120,251 'OK Enter ไปยังหน้าจอต่อไป'
FRAME 2,3 SCREEN 'กรอบที่ 2 ตัวอย่างที่ 1' ตัวอย่างเช่น 1+1 มีค่าเท่ากับ 2 INPUT #120,251 'OK Enter ไปยังหน้าจอต่อไป'
FRAME 3,4 SCREEN 'กรอบที่ 3 ตัวอย่างที่ 2' ตัวอย่างเช่น 2+3 มีค่าเท่ากับ 5 INPUT #120,251 'OK Enter ไปยังหน้าจอต่อไป'
FRAME 4,5,6 SCREEN 'กรอบที่ 4 โจทย์ทดสอบที่ 1' ผลบวกของ 2 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 INPUT #130,101 'เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7' BRANCH VE,47310 '*C18,'*C19... *C20...' V18,'*C20...'
FRAME 5,6,7 SCREEN 'กรอบที่ 5 โจทย์ทดสอบที่ 2' ผลบวกของ 4 กับ 2 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 INPUT #130,101 'เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7' BRANCH VE,47310 '*C18,'*C19... *C20...' V18,'*C20...'
FRAME 6,7,8 SCREEN 'กรอบที่ 6 โจทย์ทดสอบที่ 3' ผลบวกของ 5 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 INPUT #130,101 'เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7' BRANCH VE,47310 '*C18,'*C19... *C20...' V18,'*C20...'

หน้าจอที่ 1 คำอธิบายเกี่ยวกับการบวก การบวกเลขสองจำนวน คือ การนำเลขทั้งสองจำนวนมารวมกัน OK Enter ไปยังหน้าจอต่อไป	เมื่อผู้เรียนกด Enter จะไปหน้าจอที่ 2
หน้าจอที่ 2 ตัวอย่างที่ 1 ตัวอย่างเช่น 1+1 มีค่าเท่ากับ 2 OK Enter ไปยังหน้าจอต่อไป	เมื่อผู้เรียนกด Enter จะไปหน้าจอที่ 3
หน้าจอที่ 3 ตัวอย่างที่ 2 ตัวอย่างเช่น 2+3 มีค่าเท่ากับ 5 OK Enter ไปยังหน้าจอต่อไป	เมื่อผู้เรียนกด Enter จะไปหน้าจอที่ 4
หน้าจอที่ 4 โจทย์ทดสอบที่ 1 ผลบวกของ 2 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7	จอภาพที่ 4 นี้ จะให้ผู้เรียนได้เลือกคำตอบที่เรียนคอมพิวเตอร์จะ มีข้อความโต้ตอบว่า *C18... และให้ผู้เรียนกดเข้ามา ซึ่งจอภาพที่ 4 อีกครั้งหนึ่ง โดยให้โอกาสในการตอบไม่เกิน 2 ครั้ง ถ้าผู้เรียนตอบถูกต้องจะมีความโต้ตอบว่า *C19... เก่ง มาก แล้วผ่านไปยังจอภาพที่ 5 กรอบที่ 5 มีข้อที่ 2 ครั้งก็จะ ผ่านไปยังจอภาพที่ 6
หน้าจอที่ 5 โจทย์ทดสอบที่ 2 ผลบวกของ 4 กับ 2 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7	จอภาพที่ 5 นี้ จะให้ผู้เรียนได้เลือกคำตอบที่เรียนคอมพิวเตอร์จะ มีข้อความโต้ตอบว่า *C18... และให้ผู้เรียนกดเข้ามาซึ่ง จอภาพที่ 5 ในอีกครั้งหนึ่ง โดยให้โอกาสในการตอบไม่เกิน 2 ครั้ง ถ้าผู้เรียนตอบถูกต้องจะมีความโต้ตอบว่า เก่งมาก แล้ว ผ่านไปยังจอภาพที่ 6 กรอบที่ 6 มีข้อที่ 2 ครั้งก็จะผ่านไปยัง จอภาพที่ 6
หน้าจอที่ 6 โจทย์ทดสอบที่ 3 ผลบวกของ 5 กับ 3 มีค่าเท่าไร ก. 4 ข. 5 ค. 6 ง. 7 จ. 8 เลือกคำตอบที่ถูกต้อง กด ก ข ค ง หรือ จ แล้วกด Enter 7	จอภาพที่ 6 นี้ จะให้ผู้เรียนได้เลือกคำตอบที่เรียนคอมพิวเตอร์จะ มีข้อความโต้ตอบว่า *C18... และให้ผู้เรียนกดเข้ามาซึ่ง จอภาพที่ 6 ในอีกครั้งหนึ่ง โดยให้โอกาสในการตอบไม่เกิน 2 ครั้ง ถ้าผู้เรียนตอบถูกต้องจะมีความโต้ตอบว่า ยอด เยี่ยม... แล้วเลิกการเรียน กรณีที่ตอบไม่ถูกต้อง 2 ครั้งก็จะ ผ่านไปเลิกการเรียน

จะเห็นว่าภาษา TAL ช่วยเพิ่มเทคนิคการสอนและจัด
ลำดับขั้นตอนการนำเสนอบทเรียน ทำให้สามารถสร้างความสนใจ
และดึงดูดใจผู้เรียนที่จะเรียนรู้มากขึ้น TAL เป็นภาษาที่สามารถ
ทำความเข้าใจเพื่อใช้งานได้ในเวลาไม่นาน มีรูปแบบ
และโครงสร้างง่าย ๆ ดังรายละเอียดในภาคผนวกท้ายเล่ม

(3) ทดสอบการใช้งาน เมื่อจัดทำแฟ้ม โครงสร้างบทเรียน และแฟ้มเนื้อหาวิชาแล้วเสร็จ ให้นำแฟ้มข้อมูลทั้งหมดไปทำการทดสอบใช้งาน โดยการเรียกการทำงานของโปรแกรม TEACH.EXE ป้อนข้อมูลชื่อวิชาและรหัสผู้สอน ตรวจสอบการทำงานของแฟ้มเนื้อหาวิชาว่าเป็นไปตามความต้องการของผู้สอนหรือไม่ หากไม่เป็นไปตามความต้องการก็แก้ไขปรับปรุงแฟ้มเนื้อหาวิชาต่อไป

(4) แก้ไขและปรับปรุง จะเป็นการแก้ไขปรับปรุงแฟ้ม โครงสร้างบทเรียนและแฟ้มเนื้อหาวิชาให้เป็นไปตามประสงค์ของผู้สอน หลังจากการตรวจสอบการใช้งานแล้ว การแก้ไขนี้จะใช้โปรแกรมประมวลคำเช่นเดียวกับการสร้าง

(5) นำไปใช้งาน เมื่อได้บทเรียนโปรแกรมตามประสงค์ ก็สามารถนำไปให้ผู้เรียนใช้งาน โดยผู้สอนแจ้งรหัสบทเรียนที่ต้องการให้ผู้เรียนทราบ และเพื่อให้สามารถติดตามการเรียนของผู้เรียนแต่ละคนได้ ผู้สอนควรกำหนดรหัสให้ผู้เรียนแต่ละคนไว้ ในการเรียนผู้สอนชี้แนะให้ผู้เรียนทราบว่า การใช้บทเรียน จะต้องเรียกโปรแกรม TEACH มาทำงาน และผู้เรียนจะต้องป้อนรหัสวิชา และรหัสผู้เรียน แล้วเลือกหัวข้อที่ต้องการจะเรียน เมื่อจบการเรียน โปรแกรมจะสรุปผลการเรียนของผู้เรียน

5.2 การเรียน

เมื่อผู้สอนจัดเตรียมบทเรียนเสร็จแล้ว (มีแฟ้ม โครงสร้างบทเรียนและแฟ้มเนื้อหาบทเรียนแล้ว) ผู้สอนก็สามารถกำหนดให้ผู้เรียนเข้าไปศึกษาบทเรียน โดยผู้สอนจะเป็นผู้กำหนดว่า ผู้เรียนจะเรียนวิชาอะไร และมีรหัสประจำตัวหมายเลขใด แต่ในกรณีที่เป็นบทเรียนเปิด (บทเรียนที่ให้ผู้เรียนคนใดเข้าใช้ก็ได้ตามความสนใจของผู้เรียนเอง) ในที่นี้ถ้าไม่สนใจว่าผู้เรียนเป็นใครก็อนุญาตให้ผู้เรียนกำหนดรหัสตัวเลข 3 ตัวชั้นใช้ได้อย่างอิสระ

ผู้เรียนสามารถเข้าสู่ระบบ TEACH ได้ดังนี้

5.2.1 พิมพ์ A:TEACH แล้วกด Enter ผลจะได้ข้อความนำเข้าสู่ระบบ ดังนี้

ยินดีต้อนรับ
สู่ระบบช่วยการเรียนการสอน 'ผู้สอน'

โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536
ที่ปรึกษาโครงการ : อาจารย์วุฒินงค์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
รศ. ไพโรจน์ ตีรณานุกุล กรรมการที่ปรึกษา
พัฒนาระบบโดย : นายนิพนธ์ ศุภศิริ

ข้อความนี้จะปรากฏบนจอานประมาณ 40 วินาทีหรือถ้าผู้เรียนกดแป้นใด ๆ ก็จะไปยังจอภาพต่อไป

5.2.2 หลังจากนั้นจอภาพจะปรากฏข้อความให้ผู้เรียนป้อนข้อมูลรหัสบทเรียนและรหัสผู้เรียน ดังนี้

ระบบช่วยการเรียนการสอน 'ผู้สอน'

โปรดป้อนรหัสบทเรียน[SSS###] :

โปรดป้อนรหัสผู้เรียน[XXXX] :

ระบบจะตรวจสอบรหัสบทเรียนและรหัสผู้เรียนว่ามีรูปแบบที่ถูกต้องหรือไม่ ถ้าผู้เรียนป้อนผิดเกิน 3 ครั้ง โปรแกรมจะถุกยกเลิกการทำงาน

5.2.3 เมื่อรหัสบทเรียนและรหัสผู้เรียนได้รับการยอมรับจากระบบแล้ว ระบบจะแสดงข้อมูลประวัติการเรียนของผู้เรียน เพื่อให้ผู้เรียนทราบผลการเรียนที่ผ่านมาและให้ผู้เรียนเลือกหัวข้อที่ต้องการจะศึกษาต่อไป ข้อมูลประวัติการเรียนของผู้เรียนมีรูปแบบ ดังนี้

คุณมีประวัติการศึกษาวិชา.....		ดังนี้		
หัวข้อที่	เรื่อง	เรียนเมื่อ วัน-เดือน-ปี(เวลา)	รวม ใช้เวลา	ผลการเรียน คะแนน เกรด
1
2
3

โปรด ใส่เลขหัวข้อที่คุณต้องการศึกษา
หรือ กด ENTER เข้าสู่การศึกษาหัวข้อที่...ต่อจากครั้งก่อน
หรือ กด 999 เลิกการใช้งาน >

5.2.4 เมื่อผู้เรียนเลือกหัวข้อที่ต้องการจะศึกษาแล้ว ระบบจะนำเสนอบทเรียนในหัวข้อที่เลือกจนกระทั่งจบบทเรียน เมื่อจบบทเรียน ระบบจะรายงานผลการเรียนในรูปแบบดังนี้

ผลการเรียนวิชา
หัวข้อที่ ... เรื่อง
เรียนเมื่อ วัน-เดือน-ปี เวลา _____ น. รวมใช้เวลา(ชม. : นาที : วินาที) _____
ผลการเรียน ตอบถูก _____ ข้อ ผิด _____ ข้อ สรุปได้ _____ / _____ คะแนน
คิดเป็นร้อยละ _____ ประเมินโดยใช้เกณฑ์ _____ ได้เกรด _____
[กด Enter เพื่อดูสรุปทั้งวิชา]

5.2.5 เมื่อผู้เรียนกด Enter ภายหลังจากผลรายงานแล้ว ระบบจะแสดงประวัติการเรียนของผู้เรียนตามหัวข้อที่ 3 อีกครั้งหนึ่ง เพื่อให้ผู้เรียนเลือกศึกษาหัวข้ออื่นต่อไป หรือจะเลิกการเรียนโดยการพิมพ์ 999 ก็ได้ ซึ่งระบบจะให้ผู้เรียนยืนยันความต้องการเลิกใช้งานก่อน เมื่อผู้เรียนต้องการเลิกเรียนจริง ระบบจะแสดงจอภาพการเลิกงาน ดังนี้

ขอขอบคุณที่ใช้บริการ
ระบบช่วยการเรียนการสอน 'ผู้สอน'
สวัสดี

5.3 การติดตามการเรียน

การติดตามการเรียนในที่นี้หมายถึง การติดตามดูสรุปผลการเข้าเรียนบทเรียนของผู้เรียน และติดตามดูพฤติกรรมการเรียนของผู้เรียนที่เกิดขึ้นกับบทเรียน ทั้งนี้เพื่อผู้สอนจะได้นำข้อมูลที่ได้ไปใช้ในการให้คำแนะนำแก่ผู้เรียน และใช้ในการปรับปรุงโปรแกรมบทเรียนให้มีประสิทธิภาพยิ่งขึ้น เมื่อผู้สอนต้องการติดตามดูพฤติกรรมการเรียนของผู้เรียน ผู้สอนจะต้องทราบรหัสประจำตัวของผู้เรียน และรู้ว่าผู้เรียนเรียนวิชาอะไร จากนั้นเข้าสู่โปรแกรม ANALYSER ได้ดังนี้

5.3.1 ผู้สอนพิมพ์ A:ANALYSER แล้วกด Enter ผลจะได้ข้อความนำเข้าสู่ระบบ ดังนี้

```

+-----+
| +-----+ |
| |              ยินดีต้อนรับ              | |
| |              | |
| |  สู่ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน' | |
| +-----+ |
+-----+

```

โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536
ที่ปรึกษาโครงการ : อาจารย์วุฒิพงศ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา
พัฒนาระบบโดย : นายนิพนธ์ ศุภศรี

ข้อความนี้จะปรากฏบนจอานประมาณ 40 วินาทีหรือถ้าผู้เรียนกดแป้นใด ๆ ก็จะไปยังจอภาพต่อไป

5.3.2 หลังจากนั้นจอภาพจะปรากฏข้อความให้ผู้สอนป้อนข้อมูลรหัสบทเรียนและรหัสผู้เรียนดังนี้

ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน'

โปรต็อนรหัสบทเรียน[SSSS] :
 โปรต็อนรหัสของผู้เรียน[XXXX] :

โปรแกรมจะตรวจสอบรหัสบทเรียนและรหัสผู้เรียนว่ามีรูปแบบที่ถูกต้องหรือไม่ ถ้าผู้สอนป้อนผิดเกิน 3 ครั้ง โปรแกรมจะยุติการทำงาน

5.3.3 เมื่อรหัสบทเรียนและรหัสผู้เรียนได้รับการยอมรับจากโปรแกรมแล้ว โปรแกรมจะค้นหาแฟ้มสรุปผลการเรียนของผู้เรียนที่ระบุ

ถ้าไม่พบจอภาพจะแสดงข้อความว่า ไม่มีข้อมูลการเข้าเรียนของผู้เรียนที่ระบุหรือถ้าเปิดแฟ้มแล้วมีปัญหาจะแสดงข้อความว่า การอ่านแฟ้มบันทึกการเรียนของผู้เรียนมีปัญหา จากนั้นจะยุติการทำงานของ โปรแกรม

ถ้าพบแฟ้มสรุปผลการเรียน แสดงว่ามีการเข้าใช้บทเรียน โปรแกรมจะแสดงข้อมูลสรุปผลการเรียนตามลำดับการเข้าเรียนของทุกครั้งที่เข้าเรียนของผู้เรียน หน้าจอละ 3 ครั้งแล้วรอให้ผู้สอนกดแป้น Enter เพื่อดูข้อมูลต่อไป ดังนี้

สรุปผลการเรียนตามลำดับการเข้าเรียนจำนวน ครั้ง
 วิชา..... ของนักเรียนรหัส มีดังนี้

ครั้งที่ หัวข้อที่ ชื่อแฟ้ม เรื่อง
 เรียนเมื่อ ..-..-..(.....) รวมใช้เวลา วินาที
 ตอบถูก ข้อ ผิด ข้อ ไม่ถูกไม่ผิด ข้อ คะแนนเต็ม คะแนน
 ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน

ครั้งที่ หัวข้อที่ ชื่อแฟ้ม เรื่อง
 เรียนเมื่อ ..-..-..(.....) รวมใช้เวลา วินาที
 ตอบถูก ข้อ ผิด ข้อ ไม่ถูกไม่ผิด ข้อ คะแนนเต็ม คะแนน
 ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน

ครั้งที่ หัวข้อที่ ชื่อแฟ้ม เรื่อง
 เรียนเมื่อ ..-..-..(.....) รวมใช้เวลา วินาที
 ตอบถูก ข้อ ผิด ข้อ ไม่ถูกไม่ผิด ข้อ คะแนนเต็ม คะแนน
 ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน

กดแป้นใด ๆ เพื่อดูลำดับการเข้าเรียนครั้งต่อไป

เมื่อจบการรายงานสรุปผลการเรียนตามลำดับการเข้าเรียนแล้ว จะปรากฏข้อความให้ผู้
 สอนบอกรายงานเลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการเรียน หรือหมายเลข 999 เพื่อเลิกการใช้
 โปรแกรม ดังนี้

โปรดใส่ เลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการเรียน (1-X)
 หรือ กต 999 เลิกการใช้งานโปรแกรม :

5.3.4 เมื่อผู้สอนเลือกหัวข้อที่ต้องการจะติดตามแล้ว ข้อมูลในแฟ้มพฤติกรรมการเรียน
 ในหัวข้อที่เลือกจะถูกนำมาแสดงจนกระทั่งจบแฟ้ม รายงานพฤติกรรมการเรียนแต่ละกรอบการ
 เรียนเมื่อรูปแบบดังนี้

พฤติกรรมการเรียนตามลำดับการเข้าใช้กรอบการเรียน

วิชา ของนักเรียนรหัส

หัวข้อที่ ... เรื่อง มีดังนี้

ลำดับแฟ้มที่

หมายเลขกรอบ: ชนิด: จำนวนกรอบคำถามสะสม:

จำนวนครั้งที่ยอมให้เข้าใช้: ที่เคยเข้าใช้:

เวลา

จำนวนเวลาสะสมที่ผ่านมาในบทเรียน: วินาที

เวลาเริ่มเข้าใช้กรอบ :

จำนวนเวลาทั้งหมดที่ใช้ในกรอบ : วินาที

จำนวนเวลาที่กำหนดให้สำหรับแสดงข้อความในกรอบ: วินาที ใช้จริง: วินาที

จำนวนเวลาที่กำหนดให้ใช้ตอบคำถามในกรอบ: วินาที ใช้จริง: วินาที

จำนวนเวลาที่กำหนดให้ใช้แสดงข้อความโต้ตอบในกรอบ: วินาที ใช้จริง: วินาที

คำตอบของผู้เรียน :

ผลลัพธ์จากคำตอบ : ...

หมายเลขกรอบต่อไป : ...

จำนวนข้อสอบ ถูกสะสม: ผิดสะสม: ไม่มีถูกผิดสะสม :

คะแนน

ทำได้ในกรอบนี้: ได้สะสม: เต็มสะสม: เต็มของกรอบนี้:

<<<< เคาะแบ้นไต ๆ เพื่อดูพฤติกรรมในกรอบถัดไป >>>>

ภาคผนวก จ.

ตัวอย่างบทเรียน โปรแกรม

เรื่อง การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ตัวอย่างบทเรียนนี้ให้ชื่อว่า TEA ประกอบด้วยแฟ้มโครงสร้างบทเรียนและแฟ้มเนื้อหาวิชา 6 แฟ้ม ข้อมูลในแต่ละแฟ้มที่นำแสดงไว้ในส่วนนี้ สามารถจัดเป็นสารบัญได้ดังนี้

สารบัญ

เรื่อง	หน้า
แฟ้มโครงสร้างบทเรียน	
TEA.STR.....	146
แฟ้มเนื้อหาวิชา	
TEA001.TXT.....	147
TEA002.TXT.....	156
TEA003.TXT.....	163
TEA004.TXT.....	172
TEA005.TXT.....	184
TEA006.TXT.....	191

- TEA000 การใช้ TEACH ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์
- TEA001 คำนำ จุดมุ่งหมาย ลักษณะ ประโยชน์ และอุปกรณ์ที่จำเป็นของ TEACH
- TEA002 การใช้งานระบบ(การสร้างบทเรียน)
- TEA003 การใช้งานระบบ(การเรียนรู้และการติดตามผล)
- TEA004 ภาษา TAL (TEACH Authoring Language)
- TEA005 ตัวอย่างการสร้างบทเรียนเพื่อใช้กับระบบ TEACH
- TEA006 สถิติการทำงานของบทเรียนตัวอย่างที่สร้างในหัวข้อที่ 5

FRAME 1,E

SCREEN

คู่มือการใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ในหัวข้อที่ 1 นี้ จะกล่าวถึง.....

- คำนำ
- จุดประสงค์ของ TEACH ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์
- ลักษณะของระบบ TEACH และประโยชน์ในการนำไปใช้งาน
- อุปกรณ์ที่จำเป็นสำหรับระบบ TEACH

เพื่อให้เข้าใจถึงความเห็นมาของระบบ TEACH จุดประสงค์ ลักษณะ ประโยชน์ และอุปกรณ์ที่จำเป็นสำหรับระบบ TEACH ก่อนไปสู่การใช้งานในหัวข้อต่อไป

INPUT @ (24,18) '> โปรดกดแป้น ENTER เพื่อเรียนต่อไป '

FRAME 2,E

SCREEN 'คำนำ'

ผู้สอน: เป็นระบบโปรแกรมช่วยการเรียนการสอนแบบอัตโนมัติ
(โปรแกรมที่ผู้สอนสามารถสร้างบทเรียนบนไมโครคอมพิวเตอร์ได้ด้วยตนเอง)

§T4

พัฒนาระบบโดย: นายนิพนธ์ ศุภศรี
เป็นโครงการวิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต
มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536

§T4

โดยมี: อาจารย์วชิรพงศ์ เตชะดำรงสิน เป็นประธานกรรมการที่ปรึกษา
และ รศ. ไพโรจน์ ตีรธนากุล เป็นกรรมการที่ปรึกษา

@(23,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(24,20)'> หรือกดแป้น ENTER เพื่อจุดมุ่งหมายของ TEACH '
 BRANCH
 '1'N:1

FRAME 3,E
 SCREEN

จุดมุ่งหมายของ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

ระบบ TEACH หรือระบบโปรแกรมผู้สอน

จุดมุ่งหมาย เพื่อเป็นเครื่องมืออำนวยความสะดวกแก่บุคคลที่ต้องการสร้าง
 บทเรียนบนเครื่อง ไมโครคอมพิวเตอร์ (บทเรียนสำหรับให้
 ผู้เรียนใช้เรียนด้วยตนเองบนเครื่อง ไมโครคอมพิวเตอร์)
 โดยผู้สร้างบทเรียนไม่จำเป็นต้องเป็นนักเขียนโปรแกรมมาก่อน
 ก็สามารถสร้างบทเรียนได้
 เพียงแต่ศึกษา รูปแบบแฟ้มข้อมูลที่ใช้กับระบบ
 วิธีการใช้งาน โปรแกรม
 และคำสั่งซึ่งเป็นภาษาที่ได้ออกแบบไว้ชื่อ
 TEACH Authoring Language
 หรือ TAL

@(22,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(23,20)'> หรือกดแป้น ENTER เพื่อดูความหมายของภาษา TAL '
 BRANCH
 '1'N:2

FRAME 4,E
 SCREEN

TEACH Authoring Language คือ ภาษาที่มีคำสั่งง่าย ๆ เพียงไม่กี่คำสั่ง เช่น FRAME SCREEN INPUT BRANCH เป็นต้น สามารถใช้เวลาสั้น ๆ ในการศึกษาใช้งาน

ST2

ภาษานี้จะทำหน้าที่ควบคุมการนำเสนอบทเรียน

นั่น...! หมายถึง ผู้สอนจะสามารถใส่เทคนิคการสอนตามที่ตนเองต้องการ โดยใช้ภาษา TAL

ST2

บทเรียนสำหรับใช้กับระบบ TEACH จึงเป็นบทเรียนที่ประกอบด้วยภาษา TAL ซึ่งผู้สอนสามารถสร้างได้ง่ายๆ โดยจัดพิมพ์บทเรียนเป็นแฟ้มข้อมูล ด้วยโปรแกรมประมวลคำ (Word processor) และเมื่อจะใช้งานบทเรียน ก็สั่งให้บทเรียนทำงานโดยผ่านโปรแกรม TEACH

@(22,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(23,20)'> หรือกดแป้น ENTER เพื่อต่องค์ประกอบของบทเรียน '
BRANCH
'1'N:3

FRAME 5,E
SCREEN

บทเรียนที่สร้างสำเร็จแล้วจะประกอบด้วย

- แฟ้ม โครงสร้างบทเรียน ซึ่งเป็นแฟ้มที่ใช้บอกว่บทเรียนชุดนี้มีหัวข้อ แต่ละหัวข้อชื่ออะไร
- แฟ้ม เนื้อหาบทเรียน ซึ่งเป็นแฟ้มบรรจุเนื้อหาบทเรียนของแต่ละหัวข้อ โดยเนื้อหาของบทเรียนจะถูกกำกับด้วยคำสั่งภาษา TAL ซึ่งจะทำหน้าที่ควบคุมการนำเสนอบทเรียน เมื่อนำไปใช้กับระบบ TEACH

นอกจากการนำเสนอบทเรียนแล้วระบบ TEACH ยังทำหน้าที่จัดการ การเรียน โดยการบันทึกสรปผลการเรียน บันทึกพฤติกรรมการเรียนของผู้เรียน และจัดให้ผู้เรียน เข้าศึกษาในตำแหน่งที่ต่อ เนื่องจากครั้งก่อน หรือตามความ ประสงค์ของผู้เรียนด้วย

@(22,15)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(23,15)'> หรือกดแป้น ENTER เพื่อดูประโยชน์ที่ได้รับจากระบบ '
 BRANCH
 '1'N:4

FRAME 6,E

SCREEN

@(7,1)' '

ประโยชน์ที่ได้รับจากระบบนี้ คือ

* ผู้เรียน ได้เรียนบทเรียนด้วยตนเองที่มีปฏิสัมพันธ์กับผู้เรียนแต่ละคน
 โดยเฉพาะ

* ผู้สอน สามารถติดตามประเมินผลบทเรียน
 ศึกษาพฤติกรรมกรเรียนของผู้เรียน
 และผลการเรียนของผู้เรียนได้ จากข้อมูลที่ระบบบันทึกไว้

@(21,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(22,20)'> หรือกดแป้น ENTER เพื่อเรียนต่อไป '

BRANCH

'1'N:5

FRAME 7,E

SCREEN

ลักษณะของระบบ TEACH

ดังได้กล่าวแล้วว่า ระบบนี้มีลักษณะเป็นระบบโปรแกรมช่วยการเรียนการสอน
 แบบอัตโนมัติ (โปรแกรมที่อำนวยความสะดวกให้ครูสามารถสร้างบทเรียนบนไมโคร
 คอมพิวเตอร์ได้ด้วยตนเอง) ชนิดกรอบการเรียน (FRAME)

คือ บทเรียนที่จะนำมาใช้กับระบบจะถูกจัดแบ่งออกเป็นส่วนย่อย ๆ ที่เรียกว่า
 ว่างกรอบ (FRAME) แต่ละกรอบจะมีขนาดข้อมูลประมาณ 1 หน้าจอ (24 บรรทัด)
 หรือน้อยกว่า เพื่อสะดวกแก่การศึกษาของผู้เรียน

การเข้าเรียนของผู้เรียนโดยใช้ระบบ TEACH
 ผู้เรียนสามารถเข้าเรียนได้โดยการเรียกบทเรียนผ่านโปรแกรม TEACH
 ซึ่งเป็นโปรแกรมหนึ่งในระบบ

@(21,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(22,20)'> หรือกดแป้น ENTER เพื่อดูส่วนต่าง ๆ ของระบบ TEACH '
 BRANCH
 '1'N:6

FRAME 8,E
 SCREEN

ระบบ TEACH แยกได้เป็น 2 ส่วน คือ

 ส่วนที่ 1

มีโปรแกรมหลัก TEACH เป็นโปรแกรมควบคุมจัดการการเรียนการสอน ดำเนินการสอน การโต้ตอบ บันทึกพฤติกรรมการณ์เรียนของผู้เรียน บันทึกสรุปผลการเรียน

\$T2

มีแฟ้มข้อมูลที่เกี่ยวข้องในการประมวลผล 5 แฟ้มข้อมูลคือ

- แฟ้มข้อมูลโครงสร้างบทเรียน
- แฟ้มข้อมูลเนื้อหาวิชา
- แฟ้มข้อมูลสรุปผลการเรียน
- แฟ้มข้อมูลบันทึกพฤติกรรม
- แฟ้มข้อมูลข้อความโต้ตอบ

@(23,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(24,20)'> หรือกดแป้น ENTER เพื่อดูส่วนที่ 2 ของระบบ TEACH '
 BRANCH
 '1'N:7

FRAME 9,E
 SCREEN

ส่วนที่ 2

มีโปรแกรมหลัก ANALYSER สำหรับให้ผู้สอนใช้ดข้อมูลสรุปผลการเรียน และดข้อมูลพฤติกรรมกรเรียนของผู้เรียน ซึ่งเป็นประโยชน์ต่อการวิเคราะห์ ประสิทธิภาพของบทเรียน และให้คำปรึกษาแก่ผู้เรียน

ข้อมูลที่ใช้ในส่วนนี้ คือ ข้อมูลจากแฟ้มข้อมูลสรุปผลการเรียน

และแฟ้มข้อมูลบันทึกพฤติกรรม

ซึ่งเป็นผลลัพธ์จากส่วนที่ 1

@(21,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(22,20)'> หรือกดแป้น ENTER เพื่อตั้งงานระบบส่วนที่ 1 '

BRANCH

'1'N:8

FRAME 10,E

SCREEN

ส่วนที่ 1

```

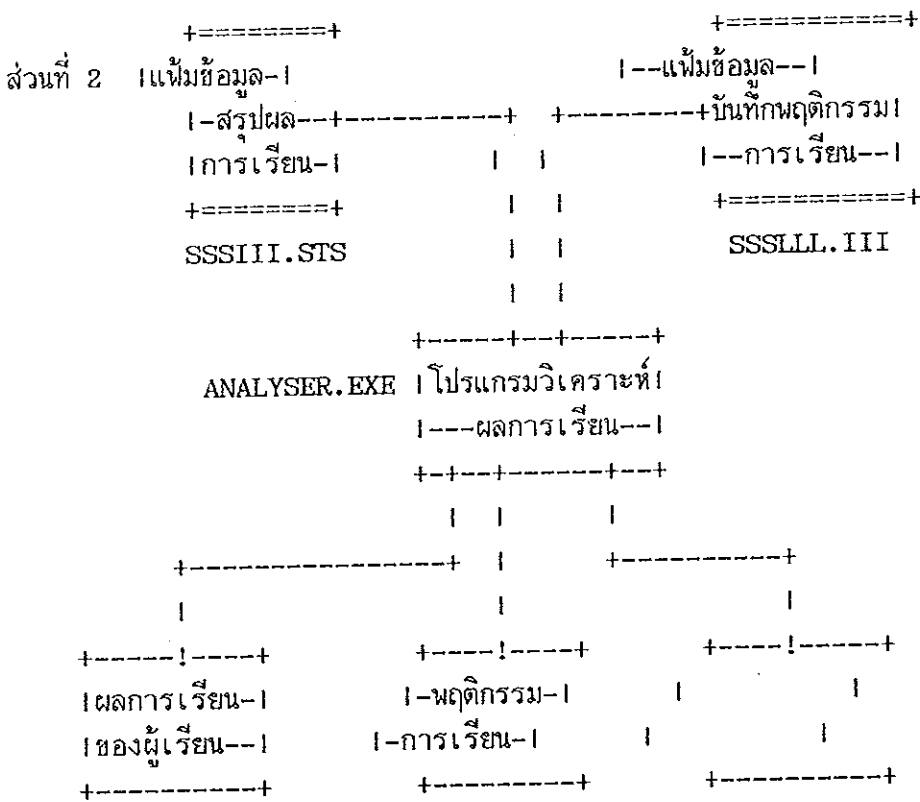
+-----+
| เริ่มต้น-|
+---+---+
SSS.STR          | TEACH.EXE  CAICOM.MSG
+=====+      +-----+-----+   +=====+
|แฟ้มข้อมูล-+-----+โปรแกรมผู้สอน-+-----+แฟ้มข้อมูล-|
|โครงสร้าง| |---TEACH---| |ข้อความ-|
|-บทเรียน-| +-----+---+ |โต้ตอบ--|
+=====+          |||          +=====+
+=====+          |||
|แฟ้มข้อมูล|SSSLLL.TXT |||
|-เนื้อหา-+-----+||
|---วิชา-|          ||
+=====+ +=====+  || +=====+
|แฟ้มข้อมูล-| |+---+--แฟ้มข้อมูล--|
SSSIII.STS |--สรุปผล-+-----+ |บันทึกพฤติกรรม| SSSLLL.III
|การเรียน-| |--การเรียน--|
+=====+          +=====+

```

ผังงานระบบของระบบ TEACH ส่วนที่ 1

@(23,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(24,20)'> หรือกดแป้น ENTER เพื่อดูผังงานระบบส่วนที่ 2 '
 BRANCH
 '1'N:9

FRAME 11,E
 SCREEN



ผังงานระบบของระบบ TEACH ส่วนที่ 2

@(23,13)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(24,13)'> หรือกดแป้น ENTER เพื่อดูอุปกรณ์ที่จำเป็นสำหรับระบบ TEACH '
 BRANCH
 '1'N:10

FRAME 12, E
 SCREEN

อุปกรณ์ที่จำเป็นสำหรับระบบ TEACH

ในการใช้งานระบบ TEACH จำเป็นต้องมีอุปกรณ์ที่เป็น
Hardware และ Software ดังต่อไปนี้

Hardware

\$T2

- 1 เครื่องไมโครคอมพิวเตอร์ 16 บิต (IBM PC, XT, AT หรือ Compatible) ที่มีหน่วยความจำไม่ต่ำกว่า 256 K

\$T2

- 2 Floppy disk drive 1 ตัว

\$T2

- 3 PC-DOS รุ่น 3.0 ขึ้นไป หรือ MS-DOS รุ่น 3.0 ขึ้นไป

@(23,18)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(24,18)'> หรือกดแป้น ENTER เพื่อดู Software ที่ระบบใช้งาน '
BRANCH
'1'N:11

FRAME 13,E

SCREEN

Software

การใช้งานระบบ TEACH จะต้องมีโปรแกรมดังต่อไปนี้

- 1 TEACH.EXE

\$T1

- 2 ANALYSER.EXE

\$T1

- 3 THAI.COM

\$T1

- 4 แฟ้มบทเรียนที่มีส่วนขยายเป็น .TXT

\$T1

- 5 แฟ้มโครงสร้างบทเรียนที่มีส่วนขยายเป็น .STR

\$T1

- 6 CAICOM.MSG แฟ้มข้อความโต้ตอบ

\$T2

เมื่อมีการเรียนจะมีแฟ้มบันทึกข้อมูลการเรียนเกิดขึ้นคือ

\$T1

7 เพิ่มข้อมูลสรุปผลการเรียนที่มีส่วนขยายเป็น .STS

\$T1

8 เพิ่มข้อมูลบันทึกพฤติกรรมกรเรียนที่มีส่วนขยายเป็น เลขรหัสผู้เรียน

\$T1

@(23,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(24,20)'> กดแป้น ENTER เพื่อไปสรุปการเรียน '

BRANCH

'1'N:12

FRAME 14,E

SCREEN

จากการเรียนที่ผ่านมา

หากมีส่วนใดที่ยังสงสัยอยู่ ท่านสามารถกลับเข้าเรียนใหม่

โดยเลือกหัวข้อ 1 ใหม่อีกครั้ง

หาก... ท่านได้เข้าใจถึง

ความเป็นมาของระบบ TEACH

จุดประสงค์

ลักษณะ

ประโยชน์

และอุปกรณ์ที่จำเป็นสำหรับระบบ TEACH แล้ว

ขอแนะนำให้ท่านเลือกศึกษาหัวข้อที่ 2

เรื่องการใช้งานระบบ TEACH ต่อไป

@(23,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(24,20)'> หรือกดแป้น ENTER เพื่อจบการเรียนหัวข้อที่ 1 นี้ '

BRANCH

'1'N:13

N:999

FRAME 20,E
SCREEN

- ในงานการเรียนการสอนมีบุคคลสำคัญ 2 ฝ่ายที่เกี่ยวข้องกัน
คือ ผู้สอน กับ ผู้เรียน
- ผู้สอนมีหน้าที่สร้างบทเรียนหรือนำบทเรียนมาสอน
และติดตามผลการเรียนของผู้เรียน
 - ผู้เรียนมีหน้าที่เรียน

TEACH AUTHORIZING SYSTEM เป็นระบบที่เกี่ยวข้องกับงานดังกล่าว

ในหัวข้อที่ 2 นี้ จึงขอกล่าวถึง.....

การใช้งาน TEACH AUTHORIZING SYSTEM
เฉพาะในส่วนการสร้างบทเรียน

INPUT @ (22,12) '> กดปุ่ม ENTER เพื่อไปยังการใช้งาน TEACH AUTHORIZING SYSTEM '

FRAME 1,E
SCREEN

การใช้งาน TEACH AUTHORIZING SYSTEM

การใช้งานระบบนี้ จะประกอบด้วยบุคคล 2 ฝ่าย คือ

ผู้สร้างบทเรียนหรือผู้สอน-----กับ-----ผู้เรียน

\$T2

งานการเรียนการสอนที่บุคคลทั้ง 2 ฝ่ายมีความสัมพันธ์ต่อกัน

คือ การสร้างบทเรียน

การเรียน

และการติดตามการเรียน

INPUT @ (23,15) '> กดปุ่ม ENTER เพื่อไปยังการสร้างบทเรียน '

FRAME 2,E

SCREEN

การสร้างบทเรียน

การสร้างบทเรียนกระทำโดยผู้สร้างบทเรียนหรือผู้สอน

ในที่นี้..... หมายถึง ครู อาจารย์ ผู้สอน หรือใครก็ตาม
ที่มีความประสงค์จะสร้างบทเรียนสำหรับใช้สอนบนเครื่องไมโครคอมพิวเตอร์

การสร้างบทเรียนจะเริ่มต้นที่....

1. ผู้สร้างจะต้องมีบทเรียนที่ตนเองต้องการจะสอนไว้แล้ว
2. ควรมีการจัดแบ่งบทเรียนที่จะใช้สอนกับระบบนี้เป็นหัวข้อ
3. ในแต่ละหัวข้อควรแบ่งเป็นกรอบการเรียน

ฉะนั้น....บทเรียนหนึ่ง ๆ อาจประกอบด้วย หัวข้อเดียว หรือหลายหัวข้อก็ได้

@(21,20)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(22,20)'> หรือกดแป้น ENTER เพื่อศึกษาว่าหัวข้อคืออะไร...? '

BRANCH

'1'N:1

FRAME 3,E

SCREEN

หัวข้อคืออะไร....?

ในที่นี้....อาจให้หมายถึง จุดประสงค์การเรียนหนึ่งจุดประสงค์

หรือบทเรียนหนึ่งบท

หรือส่วนการเรียนหนึ่งส่วน

แล้วแต่ผู้สร้างบทเรียนจะกำหนด

โดยมีเงื่อนไขว่า... บทเรียนหนึ่ง ๆ มี หัวข้อได้ไม่เกิน 10 หัวข้อ

และแต่ละหัวข้อจะมีกรอบการเรียน ได้ไม่เกิน 50 กรอบการเรียน

หรือไม่เกิน 500 บรรทัด
 @ (21, 18) '> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @ (22, 18) '> หรือกดแป้น ENTER เพื่อดูว่ากรอบการเรียนรู้คืออะไร..?'
 BRANCH
 '1'N:2
 FRAME 4,E
 SCREEN

กรอบการเรียนรู้คืออะไร..?

+++++

กรอบการเรียนรู้ คือ ขอบเขตของเนื้อหาบทเรียน
 ที่ผู้สอนต้องการจะนำเสนอให้ผู้เรียนเรียน

ขอบเขตของเนื้อหาขนาดใด ? จึงเหมาะสม....สำหรับ 1 กรอบการเรียนรู้

ขอแนะนำว่า... ขอบเขตที่เหมาะสมสำหรับ 1 กรอบการเรียนรู้
 ควรนำเสนอเนื้อหา 1 แนวคิด (concept)
 โดย ปริมาณข้อความควรมีความยาวไม่เกิน 1 หน้าจอภาพ
 (ไม่เกิน 24 บรรทัด) เพราะ... จะทำให้ง่ายต่อการควบคุม
 ลำดับขั้นตอนการนำเสนอ และ ง่ายต่อการทำความเข้าใจ
 บทเรียนของผู้เรียน

จุดประสงค์ของแต่ละกรอบขึ้นกับความต้องการของผู้สอนว่าจะกำหนดให้เป็น
 กรอบสำหรับสอนเนื้อหา (ชนิด E)
 กรอบสำหรับถามคำถาม (ชนิด Q)
 หรือกรอบสำหรับส่งเสริมเลือกกรอบการเรียนรู้อื่น ๆ มานำเสนอ (ชนิด R)

@ (23, 18) '> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @ (24, 18) '> หรือกดแป้น ENTER เพื่อดูขั้นตอนการสร้างบทเรียน '
 BRANCH
 '1'N:3
 FRAME 5,E
 SCREEN

การสร้างบทเรียนเพื่อนำไปใช้กับระบบ TEACH

มีขั้นตอนดังนี้

1. กำหนดหัวข้อต่าง ๆ ในบทเรียนแล้วนำไปจัดทำเป็นแฟ้มโครงสร้าง

2. จัดทำแฟ้มเนื้อหาบทเรียนแต่ละหัวข้อ
3. ทดสอบใช้งาน
4. แก้ไขและปรับปรุง
5. นำไปใช้งาน

@(21,18)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(22,18)'> หรือกดแป้น ENTER เพื่อดูตัวอย่างการสร้างบทเรียน '
 BRANCH
 '1'N:4
 FRAME 6,E
 SCREEN

ตัวอย่างการสร้างบทเรียนเรื่องการบวกเลขเพื่อนำไปใช้กับระบบ TEACH

มีขั้นตอนดังนี้

1. กำหนดหัวข้อต่าง ๆ ในบทเรียนแล้วนำไปจัดทำเป็นแฟ้มโครงสร้าง

หัวข้อที่กำหนดไว้-----จัดทำเป็น-----+

```
*****
** วิชาคณิตศาสตร์ประถม 1 *****
** 1 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด *
** 2 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด ***
** 3 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด **
** 4 การบวกเลขสองหลักสองจำนวนที่มีตัวทด ****
*****
```

X

แฟ้มโครงสร้างชื่อ MAT.STR (พิมพ์ด้วย WP)

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xx MAT000 วิชาคณิตศาสตร์ประถม 1 XXXXXXXXXXXXXXXXXXXX
xx MAT001 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด xx
xx MAT002 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด xxx
xx MAT003 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด xxx
xx MAT004 การบวกเลขสองหลักสองจำนวนที่มีตัวทด xxx
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

@(24,17)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
 INPUT @(25,17)'> หรือกดแป้น ENTER เพื่อดูการตั้งชื่อแฟ้มโครงสร้าง '

BRANCH
 '1'N:5
 FRAME 7,E

SCREEN

การตั้งชื่อแฟ้มโครงสร้างและรูปแบบข้อมูลในแฟ้มโครงสร้าง

=====

การตั้งชื่อแฟ้มโครงสร้าง มีข้อกำหนดว่า ชื่อประกอบด้วยอักขระ 3 ตัว
และมีส่วนขยายของชื่อแฟ้มต้องเป็น .STR

รูปแบบข้อมูลในแฟ้มโครงสร้าง มีข้อกำหนดว่า จะต้องประกอบด้วย 2 ส่วน
ที่แยกจากกันด้วยช่องว่าง มีรูปแบบดังนี้

ชื่อแฟ้มหัวข้อ ชื่อหัวข้อ

เมื่อ * ชื่อแฟ้มหัวข้อ คือ ชื่อแฟ้มบทเรียนของหัวข้อนั้น ชื่อจะประกอบด้วย
อักขระ 3 ตัวแรกของชื่อแฟ้มโครงสร้างตามด้วยตัวเลข 3 ตัว
ซึ่งเป็นลำดับหัวข้อ โดยไม่ต้องมีส่วนขยายชื่อแฟ้ม และกำหนดไว้เป็น
การเฉพาะให้ชื่อแฟ้มหัวข้อที่มีรหัสตามด้วยตัวเลข 000 เป็นชื่อวิชา
จากตัวอย่างข้างต้น ตั้งชื่อแฟ้มวิชาเป็น MAT.STR
หัวข้อต่าง ๆ ในแฟ้มโครงสร้างมีชื่อเป็น MAT000 MAT001 MAT002
MAT003 MAT004

* ชื่อหัวข้อ คือ ชื่อหัวข้อนั้น ๆ

> พิมพ์ 1 แล้วกดแป้น ENTER

เพื่อกลับไปดูตัวอย่าง ในกรอบที่ผ่านมามากครั้งถ้ายังไม่ชัดเจน

INPUT @ (24,10) '>' หรือกดแป้น ENTER เพื่อดูการจัดทำแฟ้มเนื้อหาบทเรียน '

BRANCH

'1'N:6

FRAME 8,E

SCREEN

2. จัดทำแฟ้มเนื้อหาบทเรียนแต่ละหัวข้อ

จำนวนแฟ้มเนื้อหาบทเรียนที่จะต้องสร้าง

จะมีจำนวนเท่ากับ จำนวนหัวข้อที่มีในแฟ้มโครงสร้าง

จากตัวอย่าง แฟ้มโครงสร้าง MAT.STR จะมีแฟ้มเนื้อหาบทเรียนจำนวน 4 แฟ้ม
ได้แก่ MAT001.TXT MAT002.TXT MAT003.TXT และ MAT004.TXT

ลักษณะของข้อมูลในแฟ้มเนื้อหาบทเรียน

ข้อมูลในแฟ้มเนื้อหาบทเรียน ก็คือ.....

เนื้อหาบทเรียนที่มีภาษา TAL เป็นส่วนประกอบ

โครงสร้างของภาษา TAL จะมีลักษณะในการจัดแบ่งเนื้อหาเป็นกรอบการเรียน
และช่วยในการนำเสนอบทเรียนให้เป็นไปตามที่ผู้สอนต้องการ

การสร้างแฟ้มเนื้อหาบทเรียนทำได้ง่าย ๆ โดยการพิมพ์เนื้อหาบทเรียน
ที่มีภาษา TAL เป็นส่วนประกอบด้วยโปรแกรมประมวลคำทั่วไป
และตั้งชื่อแฟ้มเป็นชื่อเดียวกับชื่อแฟ้มหัวข้อที่อยู่ในแฟ้ม โครงสร้าง
ให้มีส่วนขยายของชื่อแฟ้มเป็น .TXT

> พิมพ์ 1 แล้วกดแป้น ENTER

เพื่อกลับไปดูตัวอย่างในกรอบที่ผ่านมาอีกครั้งถ้ายังไม่ชัดเจน

INPUT @(24,15)'> หรือกดแป้น ENTER เพื่อไปยังผังการสร้างแฟ้มเนื้อหาบทเรียน '

BRANCH

'1'N:7

FRAME 9,E

SCREEN

ผังการสร้างแฟ้มเนื้อหาบทเรียน

พิมพ์ด้วย WP

* เนื้อหาบทเรียน *

* เนื้อหาบทเรียน *****> ** ที่มีภาษา TAL *

*** ควบคุม ***

<<<<< ตัวอย่างการสร้างแฟ้มเนื้อหาบทเรียน

และผลที่ได้จากการสร้าง

ติดตามดูได้ในหัวข้อที่ 4

>>>>>

> พิมพ์ 1 แล้วกดแป้น ENTER

เพื่อกลับไปดูตัวอย่างในกรอบที่ผ่านมาอีกครั้งถ้ายังไม่ชัดเจน

INPUT @(21,15)'> หรือกดแป้น ENTER เพื่อจบการเรียนหัวข้อนี้ '

BRANCH

'1'N:8

N:999

FRAME 1,E
SCREEN

ในหัวข้อที่ 2 ได้กล่าวถึง.....

การใช้งาน TEACH AUTHORIZING SYSTEM
เฉพาะในส่วนการสร้างบทเรียนไปแล้ว

ในหัวข้อที่ 3 นี้ จึงขอกล่าวถึงในส่วนที่เหลือคือ.....

ส่วนการเรียน และ การติดตามการเรียน

INPUT @ (22,20) '> กดแป้น ENTER เพื่อไปยังส่วนการเรียน '

FRAME 10,E
SCREEN
@ (4,1) ' '

การเรียน

เมื่อผู้สอนจัดเตรียมบทเรียนเสร็จแล้ว
(มีแฟ้มโครงสร้างบทเรียนและแฟ้มเนื้อหาบทเรียนแล้ว)
ผู้สอนก็สามารถกำหนดให้ผู้เรียน ไปศึกษาบทเรียน
โดยผู้สอนจะเป็นผู้กำหนดว่า ผู้เรียนจะเรียนวิชาอะไร
และมีรหัสประจำตัวหมายเลขใด

แต่ในกรณีที่บทเรียนเปิด (บทเรียนที่ให้ผู้เรียนเข้าใช้
ตามความสนใจของตนเอง) ในที่นี้ ถ้าไม่สนใจว่าผู้เรียนเป็นใคร
ก็ให้ผู้เรียนกำหนดรหัสตัวเลข 3 ตัวขึ้นใช้ได้เองอย่างอิสระ

INPUT @ (23,20) '> กดแป้น ENTER เพื่อดูวิธีการเข้าสู่ระบบ TEACH '

FRAME 11,E
SCREEN

ผู้เรียนสามารถเข้าสู่ระบบ TEACH ได้ดังนี้

- 1 พิมพ์ A:TEACH แล้วกด Enter ผลจะได้ข้อความนำเข้าสู่ระบบ
- 2 หลังจากนั้นจอภาพจะปรากฏข้อความให้ผู้เรียนป้อนข้อมูล รหัสบทเรียนและรหัสผู้เรียน
- 3 เมื่อรหัสบทเรียนและรหัสผู้เรียนได้รับการยอมรับจากระบบแล้ว ระบบจะแสดงข้อมูลประวัติการเรียนของผู้เรียน เพื่อให้ผู้เรียนทราบผลการเรียนที่ผ่านมา และให้ผู้เรียนเลือกหัวข้อที่ต้องการจะศึกษาต่อไป
- 4 เมื่อผู้เรียนเลือกหัวข้อที่ต้องการจะศึกษาแล้ว ระบบจะนำเสนอบทเรียนในหัวข้อที่เลือกจนกระทั่งจบบทเรียน และรายงานผลการเรียน เมื่อผู้เรียนกด Enter ระบบจะแสดงประวัติการเรียนของผู้เรียนตามหัวข้อที่ 3 อีกครั้งหนึ่ง เพื่อให้ผู้เรียนเลือกศึกษาหัวข้ออื่นต่อไป หรือจะเลิกการเรียนโดยการพิมพ์ 999 ก็ได้ ซึ่งระบบจะให้ผู้ใช้เรียนยืนยันความต้องการเลิกใช้งานก่อน
- 5 เมื่อผู้เรียนต้องการเลิกเรียนจริง ระบบจะแสดงจอภาพการเลิกงาน

@(22,5)'> พิมพ์ 1 2 3 4 หรือ 5 แล้วกดเป็น ENTER เพื่อเลือกดูจอภาพในข้อนั้น ๆ'

@(23,5)'> หรือพิมพ์ 6 แล้วกดเป็น ENTER เพื่อกลับไปจอภาพที่ผ่านมา'

INPUT @(24,5)'> หรือกดเป็น ENTER เพื่อดูการติดตามการเรียน '

BRANCH

'1'N:12

'2'N:13

'3'N:14

'4'N:15

'5'N:16

'6'N:10

N:17

FRAME 12,E

SCREEN

จอภาพแสดงข้อความนำเข้าสู่ระบบ.....

```

+-----+
|              ยินดีต้อนรับ              |
|  สู่ระบบช่วยการเรียนการสอน "ผู้สอน"  |
+-----+

```

โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต
มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536

ที่ปรึกษาโครงการ : อาจารย์วุฒิพงษ์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
 รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา
 พัฒนาระบบโดย : นายนิพนธ์ ศุภศิริ

ข้อความนี้จะปรากฏบนจอภาพประมาณ 40 วินาที
 หรือถ้าผู้เรียนกดแป้นใด ๆ ก็จะไปยังจอภาพต่อไป

INPUT @ (24,10) '> กดแป้น ENTER เพื่อกลับไปยังการเข้าสู่ระบบ TEACH '
 BRANCH
 N:11

FRAME 13,E
 SCREEN

จอภาพให้ผู้เรียนป้อนข้อมูลรหัสบทเรียนและรหัสผู้เรียน

ระบบช่วยการเรียนการสอน 'ผู้สอน'

=====
 โปรดป้อนรหัสบทเรียน[SSSS]:

โปรดป้อนรหัสผู้เรียน[XXXX]:

ระบบจะตรวจสอบรหัสบทเรียนและรหัสผู้เรียนว่ามีรูปแบบที่ถูกต้องหรือไม่
 ถ้าผู้เรียนป้อนผิดเกิน 3 ครั้ง โปรแกรมจะหยุดยกเลิกการทำงาน

INPUT @ (24,10) '> กดแป้น ENTER เพื่อกลับไปยังการเข้าสู่ระบบ TEACH '
 BRANCH
 N:11

FRAME 14,E
 SCREEN

จอภาพแสดงข้อมูลประวัติการเรียนของผู้เรียน.....

วิชา.....	หัวข้อ ที่	เรื่อง	เรียนเมื่อ วัน-เดือน-ปี(เวลา)	รวม ใช้เวลา	ผลการเรียน คะแนน	เกรด
	1
	2
	3

โปรด ใส่เลขหัวข้อที่ต้องการศึกษา แล้วกด ENTER
 หรือ กด ENTER เข้าสู่การศึกษาหัวข้อที่...ต่อจากครั้งก่อน
 หรือ พิมพ์ 999 แล้วกด ENTER เพื่อเลิกการใช้งาน >

INPUT @(24,5)'> กดแป้น ENTER เพื่อกลับไปยังการเข้าสู่ระบบ TEACH '
 BRANCH
 N:11

FRAME 15,E
 SCREEN

เมื่อจบบทเรียน ระบบจะแสดงจอภาพรายงานผลการเรียนดังนี้.....

ผลการเรียนวิชา
 หัวข้อที่ ... เรื่อง
 เรียนเมื่อ วัน-เดือน-ปี เวลา.....น. รวมใช้เวลา(ชม. : นาที : วินาที).....
 ผลการเรียน ตอบถูก.....ข้อ ผิดข้อ สรุปได้ ../... คะแนน
 คิดเป็นร้อยละ.....ประเมินโดยใช้เกณฑ์.....ได้เกรด

[กด enter เพื่อสรุปทั้งวิชา]

INPUT @(24,10)'> กดแป้น ENTER เพื่อกลับไปยังการเข้าสู่ระบบ TEACH '
 BRANCH
 N:11

FRAME 16,E
 SCREEN

ผู้สอนติดตามดูพฤติกรรมการเรียนของผู้เรียนได้
โดยการใช้โปรแกรม ANALYSER ตามขั้นตอนต่อไปนี้

- 1 พิมพ์ A:ANALYSER แล้วกด Enter ผลจะได้ข้อความนำเข้าสู่ระบบ
- 2 หลังจากนั้นจอภาพจะปรากฏข้อความให้ป้อนข้อมูลรหัสบทเรียนและรหัสผู้เรียน
- 3 เมื่อรหัสบทเรียนและรหัสผู้เรียนได้รับการยอมรับจากโปรแกรมแล้ว
โปรแกรมจะค้นหาแฟ้มสรุปผลการเรียนของผู้เรียนที่ระบุ
 - > ถ้าไม่พบ...จอภาพแสดงข้อความว่า...ไม่มีข้อมูลการเข้าเรียนของผู้เรียนที่ระบุ
 - > หรือถ้าเปิดแฟ้มแล้วมีปัญหา...จอภาพแสดงข้อความว่า การอ่านแฟ้มบันทึกการเรียนของผู้เรียนมีปัญหา จากนั้นจะยุติการทำงานของโปรแกรม
 - > ถ้าพบแฟ้มสรุปผลการเรียน แสดงว่า...มีการเข้าใช้บทเรียน
โปรแกรมจะแสดงข้อมูลสรุปผลการเรียนตามลำดับการเข้าเรียนทั้งหมด
ที่ผู้เรียนเข้าเรียน โดยแสดงหน้าจอละ 3 ครั้งแล้วรอให้ผู้สอนกดแป้น ENTER
เพื่อต่อข้อมูลต่อไป
- 4 เมื่อผู้สอนเลือกหัวข้อที่ต้องการจะติดตามแล้ว
ข้อมูลพฤติกรรมการเรียนในหัวข้อที่เลือกจะถูกนำมาแสดงจนกระทั่งจบแฟ้ม
- 5 เมื่อจบการรายงานผลแล้ว
โปรแกรมจะแสดงสรุปผลการเรียนตามลำดับการเข้าเรียนตามหัวข้อที่ 3 อีกครั้งหนึ่ง
เพื่อให้ผู้สอนเลือกดูพฤติกรรมการเรียนหัวข้ออื่นต่อไป หรือจะเลิกการเรียกดู
โดยการพิมพ์ 999 ซึ่งโปรแกรมจะให้ผู้สอนยืนยันความต้องการเลิกใช้งานอีกครั้ง
เมื่อผู้สอนต้องการเลิกจริง จะแสดงจอภาพการเลิกงาน

@(22,8)'> พิมพ์ 1 2 3 4 หรือ 5 แล้วกดแป้น ENTER เพื่อเลือกดูจอภาพในข้อนั้น ๆ'

@(23,8)'> พิมพ์ 6 แล้วกดแป้น ENTER เพื่อกลับไปดูการติดตามการเรียน'

INPUT @(24,8)'> หรือกดแป้น ENTER เพื่อจบการเรียนหัวข้อที่ 3 นี้ '

BRANCH

'1'N:19

'2'N:20

'3'N:21

'4'N:22

'5'N:23

'6'N:17

N:999

FRAME 19,E

SCREEN

จอภาพแสดงข้อความนำเข้าสู่ระบบ.....

```

+-----+
|-----|

```

```

      | |              ยินดีต้อนรับ              | |
      | |              | |
      | |  สู่ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน' | |
      | +-----+ |
      +-----+

```

โครงการวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต
 มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2536
 ที่ปรึกษาโครงการ : อาจารย์วุฒินงค์ เตชะดำรงสิน ประธานกรรมการที่ปรึกษา
 รศ. ไพโรจน์ ตีรณานกุล กรรมการที่ปรึกษา
 พัฒนาระบบโดย : นายนิพนธ์ ศุภศรี

ข้อความนี้จะปรากฏบนจอานประมาณ 40 วินาทีหรือถ้าผู้เรียนกดแป้นใด ๆ
 ก็จะผ่านไปยังจอภาพต่อไป

INPUT @ (24,5)'> กดแป้น ENTER เพื่อกลับไปขั้นตอนการติดตามคุณพฤติกรรมการเรียน '
 BRANCH
 N:18

FRAME 20,E
 SCREEN

จอภาพให้ผู้สอนป้อนข้อมูลรหัสบทเรียนและรหัสผู้เรียน

ระบบวิเคราะห์การเรียนการสอน 'ผู้สอน'

=====
 โปรดป้อนรหัสบทเรียน[SSSS] :

โปรดป้อนรหัสผู้เรียน[XXX] :

โปรแกรมจะตรวจสอบรหัสบทเรียนและรหัสผู้เรียนว่ามีรูปแบบที่ถูกต้องหรือไม่
 ถ้าผู้สอนป้อนผิดเกิน 3 ครั้ง โปรแกรมจะยุติการทำงาน

INPUT @ (24,5)'> กดแป้น ENTER เพื่อกลับไปขั้นตอนการติดตามคุณพฤติกรรมการเรียน '
 BRANCH
 N:18

FRAME 21,E
 SCREEN

จอภาพจะแสดงข้อมูลสรุปผลการเรียนตามลำดับการเข้าเรียนทั้งหมด...
สรุปผลการเรียนตามลำดับการเข้าเรียนจำนวน ครั้ง
วิชา..... ของนักเรียนรหัส มีดังนี้

=====
ครั้งที่ หัวข้อที่ ... ชื่อแฟ้ม เรื่อง
เรียนเมื่อ ..-..-..(.....) รวมใช้เวลา วินาที
ตอบถูก ข้อ ผิด ข้อ ไม่ถูกไม่ผิด ข้อ คะแนนเต็ม คะแนน
ทำได้ คะแนน ใช้เกณฑ์ ได้ระดับคะแนน
.
.
.

=====
กดแป้นใด ๆ เพื่อดูลำดับการเข้าเรียนครั้งต่อไป
=====

เมื่อจบการรายงานสรุปผลการเรียนตามลำดับการเข้าเรียนแล้ว
จะปรากฏข้อความให้ผู้สอนป้อนหมายเลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการณ์เรียน
หรือหมายเลข 999 เพื่อเลิกการใช้โปรแกรม ดังนี้

โปรดใส่เลขหัวข้อที่ต้องการเรียกดูพฤติกรรมการณ์เรียน(1-X)
หรือ กด 999 เลิกการใช้งานโปรแกรม :

INPUT @ (24,5) '> กดแป้น ENTER เพื่อกลับไปขั้นตอนการติดตามดูพฤติกรรมการณ์เรียน '
BRANCH
N:18

FRAME 22,E
SCREEN

จอภาพรายงานพฤติกรรมการณ์เรียนแต่ละกรอบการเรียนมีรูปแบบดังนี้...
พฤติกรรมการณ์เรียนตามลำดับการเข้าใช้กรอบการเรียน
วิชา ของนักเรียนรหัส
หัวข้อที่ ... เรื่อง มีดังนี้

=====
ลำดับเฟรมที่
หมายเลขเฟรม: ชั้นที่: จำนวนเฟรมคำถามสะสม:
จำนวนครั้งที่ยอมให้เข้าใช้: ที่เคยเข้าใช้:
เวลา
จำนวนเวลาสะสมที่ผ่านมาในบทเรียน: วินาที
เวลาเริ่มเข้าใช้เฟรม:

FRAME 1,E

SCREEN

ภาษา TAL (TEACH Authoring Language)

ในหัวข้อนี้จะกล่าวถึง.....

1. แบบฟอร์มลวงรหัส ซึ่งอธิบายรูปแบบการพิมพ์เพิ่มเนื้อหาบทเรียน
2. คำสั่ง จะกล่าวถึงคำสั่งต่าง ๆ ที่สั่งไว้ใช้เป็นคำสั่ง
3. โครงสร้างของกรอบการเรียน ซึ่งเป็นโครงสร้างสำหรับกรอบการเรียน 1 กรอบ
4. คำสั่ง จะเป็นคำอธิบายคำสั่งภาษา TAL แต่ละคำสั่งอย่างละเอียด

5. คำแนะนำเพิ่มเติม

@(21,10)'> พิมพ์ 1 2 3 4 หรือ 5 แล้วกดแป้น ENTER เพื่อเลือกเรื่องที่ต้องการ'
INPUT @(22,10)'> หรือพิมพ์ 9 แล้วกดแป้น ENTER เพื่อเลิกการเรียน '

BRANCH , \$T3

'1'N:2

'2'N:3

'3'N:4

'4'N:6

'5'N:17

'9'N:999

N:1, 'โปรดพิมพ์ 1 2 3 4 5 หรือ 9 แล้วกด ENTER เท่านั้น'

FRAME 2,E

SCREEN

1. แบบฟอร์มลวงรหัส

การเขียนบทเรียนโปรแกรมด้วยภาษา TAL สามารถเขียนได้ดังนี้

- 1.1 เริ่มเขียนคำสั่งที่คอลัมน์ที่ 1

- 1.2 คำสั่งหรือข้อความในแต่ละบรรทัดจะมีความยาวได้ไม่เกิน 100 ตัวอักษร

- 1.3 บทเรียนโปรแกรมบทหนึ่ง จะแบ่งเนื้อหาในบทเรียนเป็นส่วนย่อย ๆ เรียกว่า กรอบการเรียน (FRAME) โดยในแต่ละบทเรียนมีความยาวได้ไม่เกิน 500 บรรทัด หรือ 50 กรอบ
- 1.4 กรอบการเรียนแต่ละกรอบจะมีโครงสร้างตามข้อ 3
- 1.5 การจัดเรียงกรอบการเรียนควรจัดให้เป็นไปตามลำดับที่ต้องการควบคุม เพื่อสะดวกต่อการนำเสนอบทเรียนและบันทึกผลการเรียนของผู้เรียน

```
INPUT @ (22, 15) '> กดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น '
```

```
BRANCH
```

```
N:1
```

```
FRAME 3,E
```

```
SCREEN
```

```
2. คำสงวน
```

```
-----
```

คำและเครื่องหมายต่อไปนี้ ถ้าไม่ใช่เป็นคำสั่ง สงวนไม่ให้พิมพ์ในคอลัมน์ที่ 1

```
FRAME      SCREEN      INPUT      BRANCH      EVALUATE
```

```
$T          @          '          "          ^
```

```
INPUT @ (22, 16) '> กดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น '
```

```
BRANCH
```

```
N:1
```

```
FRAME 4,E
```

```
SCREEN
```

```
3. โครงสร้างของกรอบการเรียน มีลักษณะดังนี้.....
```

```
COL ที่ 1
```

```
+-----+
```

```
|FRAME #no. [,frame_type] [,no_loop] |
```

```
|SCREEN ['label'] [,wait_time] |
```

```
|[string] |
```

```
|[["]string] |
```

```
|[[']string] |
```

```
|[^]string] |
```

```
|[@([line]-[col])] -[string'] |
```

```
+-----+
```



```

| $C |
| $T#no |
| . |
| INPUT @[([line][,col])][ 'message' ][,wait_time] |
| BRANCH [full_score][,wait_time] |
| [ 'value' [condition]] [:frame_no][, 'string' /message] |
| . |
| . |
+-----+

```

หมายเหตุ คำสั่งหรือข้อความภายในเครื่องหมายวงเล็บใหญ่จะมีหรือไม่มีก็ได้
 @(23,10)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น '

INPUT @(24,10)'> หรือกดแป้น ENTER เพื่อดูโครงสร้างของกรอบการเรียนต่อ '

BRANCH
 '1'N:1
 N:5

FRAME 5,E
 SCREEN

กรอบการเรียนแต่ละกรอบที่เป็นชนิดบรรยายและชนิดคำถาม จะเริ่มต้นด้วย
 คำสั่ง FRAME และตามด้วยคำสั่ง SCREEN บรรทัดต่อจาก SCREEN จะเป็นข้อความ
 แสดงผลบนจอภาพ

คำสั่งต่อจากนี้อาจเป็นคำสั่ง INPUT หรือ BRANCH หรือไม่มีทั้งคู่ก็ได้
 ถ้ามีคำสั่ง BRANCH จะต้องมีคำสั่งย่อยของ BRANCH ตามอย่างน้อย 1 บรรทัด
 ต่อจากนั้นจะเป็นกรอบการเรียนถัดไป

แต่ถ้ากรอบการเรียนนั้นเป็นชนิดลุ่ม บรรทัดต่อจากคำสั่ง FRAME
 จะต้องเป็นชุดหมายเลขกรอบที่จะถูกลุ่มขึ้นใช้งาน มีรูปแบบดังนี้

```

FRAME #no.,R
([nos-noe][,no])

```

```

@(21,13)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูกรอบที่ผ่านมา'
INPUT @(22,13)'> กดแป้น ENTER เพื่อเลือกดูเรื่องอื่น '
BRANCH
'1'N:4
N:1

```

FRAME 6,E

SCREEN

4. คำอธิบายคำสั่งที่ใช้ในภาษา TAL

จะเห็นได้ว่า ภาษา TAL ประกอบด้วยคำสั่งหลัก 4 คำสั่ง คือ

- 1 FRAME สำหรับบอก หมายเลขกรอบ
ชนิดของกรอบ
จำนวนครั้งที่ยอมให้เรียกใช้
- 2 SCREEN สำหรับกำหนดการเริ่มต้นแสดงข้อมูลออกทางจอภาพ
- 3 INPUT สำหรับการรอรับข้อมูลจากผู้ใช้นั้น
- 4 BRANCH สำหรับการแตกแขนงไปแสดงกรอบอื่น ๆ

และคำสั่งพิเศษอีก 1 คำสั่งคือ

- 5 EVALUATE สำหรับใช้กำหนดเกณฑ์การประเมินผลการเรียน

```
@(21,8)'> พิมพ์ 1 2 3 4 หรือ 5 แล้วกดแป้น ENTER เพื่อเลือกคำสั่งที่ต้องการ'
INPUT @(22,8)'> หรือพิมพ์ 9 แล้วกดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น '
BRANCH , $T2
'1'N:7
'2'N:10
'3'N:12
'4'N:13
'5'N:15
'9'N:1
N:6, 'โปรดพิมพ์ 1 2 3 4 5 หรือ 9 แล้วกดแป้น ENTER เท่านั้น'
```

FRAME 7,E

SCREEN

- 1 คำสั่ง FRAME เป็นคำสั่งใช้บอก
 - หมายเลขกรอบ (#no) มีได้ตั้งแต่ 1-99 สำหรับ 1 แฟ้มเนื้อหา
 - ชนิดของกรอบ (frame.type)
 - จำนวนครั้งที่ยอมให้ใช้งาน (no_loop) ถ้าใช้ครบจำนวนครั้งแล้วจะผ่านไปกรอบถัดไปทันที

คำแนะนำ จำนวนกรอบการเรียนใน 1 แฟ้มเนื้อหา
ไม่ควรเรียกใช้กรอบการเรียนเกิน 50 กรอบ
หรือ 1 แฟ้มเนื้อหาควรใช้เวลาเรียนประมาณ 20 นาที

รูปแบบของคำสั่ง

FRAME #no [,frame_type] [,no_loop]

ตัวอย่าง

FRAME 3,Q,2

หมายถึง กรอบหมายเลข 3 เป็นชนิดคำถาม
ยอมให้เรียกใช้งานได้ไม่เกิน 2 ครั้ง

INPUT @(24,21)'> กดแป้น ENTER เพื่อต่อไป '

BRANCH

N:8

FRAME 8,E

SCREEN

ชนิดของกรอบ(frame type)

แบ่งได้เป็น 3 ชนิด คือ

- * E (Explanation) กรอบชนิดคำบรรยาย สำหรับใช้อธิบายเนื้อหา
- * Q (Question) กรอบชนิดคำถาม
- * R (Random) กรอบชนิดสุ่มเลือกกรอบอื่นมานำเสนอ
โดยจะสุ่มตัวเลขจากกลุ่มตัวเลขที่อยู่ในบรรทัด
ต่อจากคำสั่ง FRAME กลุ่มตัวเลขนี้กำหนดให้มีได้
ไม่เกิน 20 จำนวน เขียนในลักษณะนี้
หรือเลขโดดก็ได้

ตัวอย่าง

FRAME 2,R,1

(1-4, 8, 10)

หมายถึง กรอบหมายเลข 2 เป็นชนิดลุ่ม ถูกเรียกใช้งานได้หนึ่งครั้ง
 โดยเอากรอบหมายเลข 1, 2, 3, 4, 8, 10 กรอบหนึ่งกรอบได้นำมาเสนอ
 @ (23, 12) ' > พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูความหมายของคำสั่ง FRAME'
 INPUT @ (24, 12) ' > หรือกดแป้น ENTER เพื่อดูข้อควรระวังในการใช้กรอบชนิดลุ่ม '
 BRANCH
 '1'N:7

FRAME 9, E
 SCREEN

ข้อควรระวังในการใช้กรอบการเรียนชนิดลุ่ม

- กรอบชนิดนี้จะไม่มีการสั่งหลักก่อนต่อท้าย
 เพราะเป็นการลุ่มกรอบขึ้นมาแทนที่กรอบปัจจุบันที่กำลังใช้งาน
- กรอบที่ถูกลุ่มควรกำหนดคำสั่ง BRANCH ให้เหมาะสม
 เพื่อไม่ให้ลำดับขั้นตอนการสอนสับสน และในกรอบที่ถูกลุ่มควร
 กำหนดการเข้าใช้เป็น 1 ถ้าไม่ต้องการให้ถูกลุ่มซ้ำ
- กรณีไม่มีการกำหนดทิศทางของกรอบ โปรแกรมจะเรียกกรอบที่อยู่ถัดจาก
 กรอบปัจจุบันมาทำงาน
- ห้ามลุ่มเลือกตัวเอง
- ถ้ากรอบที่ถูกลุ่มใช้หมดแล้ว โปรแกรมจะข้ามไปทำงานกรอบถัดไปที่

กรณีผู้สร้างบทเรียนไม่ระบชนิดของกรอบ และจำนวนครั้งที่ยอมให้เรียกใช้
 ถือว่าเป็นเช่นเดียวกับบรรยาย ยอมให้เรียกใช้ได้ไม่จำกัดจำนวนครั้ง
 @ (21, 12) ' > พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูชนิดของกรอบ'
 INPUT @ (22, 12) ' > หรือกดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่งอื่น '
 BRANCH
 '1'N:8
 N:6

FRAME 10, E
 SCREEN

2 คำสั่ง SCREEN
 เป็นคำสั่งกำหนดการเริ่มต้นแสดงข้อมูลทางจอภาพ

รูปแบบของคำสั่ง

```

SCREEN ['label'] [,wait_time]
[string]
[["]string]
[']string]
[^]string]
[@([line] [,col])] ['string']
%C
$T#no

```

เมื่อ Label เป็นข้อความที่จะนำเสนอในบรรทัดแรกกลางจอภาพ

wait_time จำนวนเวลาที่กำหนดให้แสดงข้อมูลบนจอภาพ โดยเริ่มจับเวลาเมื่อข้อความภายใต้คำสั่ง SCREEN ถูกนำเสนอหมด และจะผ่านไปดำเนินการสอนต่อไปเมื่อหมดเวลาที่กำหนด หรือผู้เรียนกด Enter

```

@ (23,10) '>' พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปเลือกคำสั่ง TAL อื่น ๆ
INPUT @ (24,10) '>' หรือกดแป้น ENTER เพื่อดูคำอธิบายเกี่ยวกับคำสั่ง SCREEN
BRANCH
'1'N:6

```

FRAME 11,E

SCREEN

ข้อความในบรรทัดต่อจากคำสั่ง SCREEN จะเป็นข้อความ[string] ที่ต้องการให้ถูกนำเสนอออกทางจอภาพตามลักษณะที่เห็นในบทเรียน

ถ้าไม่มีการกำหนดตำแหน่งแสดงข้อความโดยคำสั่งย่อย @(line,col) ข้อความนั้นไม่ต้องอยู่ระหว่างเครื่องหมาย '.....' และสามารถใส่ตัวนำหน้า " หรือ ' หรือ ^ เพื่อจัดข้อความชิดซ้าย หรือชิดขวา หรือกึ่งกลางบรรทัดได้ตามลำดับ

หากมีการใช้คำสั่งย่อยกำหนดตำแหน่งแสดงข้อความ ข้อความจะแสดงบนจอภาพในตำแหน่งที่กำหนด และหากมีข้อความในบรรทัดตามมา ข้อความนั้นจะแสดงในบรรทัดต่อจากบรรทัดที่กำหนดไว้

การจัดข้อความเพื่อแสดงออกทางจอภาพ มีคำสั่งย่อยสำหรับอำนวยความสะดวกดังนี้

@(line, col) 'string' กำหนดตำแหน่งแสดงข้อความ(string)บนจอภาพ

%C ลบข้อความทั้งหมดบนจอภาพ กำหนดตำแหน่งแสดงข้อความใหม่ ณ บรรทัดที่ 1

\$T#no ถ่วงเวลาตามจำนวนหมายเลขที่ให้ไว้ (วินาที)

@(23,10)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูความหมายของคำสั่ง SCREEN'
 INPUT @(24,10)'> หรือกดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่ง TAL อื่น ๆ '
 BRANCH
 '1'N:10
 N:6

FRAME 12,E

SCREEN

3 คำสั่ง INPUT
 เป็นคำสั่งสำหรับรอรับการโต้ตอบจากผู้ใช้นักเรียน

รูปแบบของคำสั่ง

INPUT @[([line][,col])]['message'][,wait_time]
 เมื่อ @[([line][,col]) กำหนดตำแหน่งแสดงข้อความรอรับการโต้ตอบ
 message ข้อความ
 wait_time กำหนดเวลาในการรอการโต้ตอบ(\$T)
 ถ้าหมดเวลาหรือผู้เรียนกด Enter จะผ่านไปยังกรอบถัดไป

ตัวอย่าง

INPUT @(8, 3) 'โปรดเลือก' , \$T10
 INPUT @(23,15)'> กดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่ง TAL อื่น ๆ '
 BRANCH
 N:6

FRAME 13,E

SCREEN

4 คำสั่ง BRANCH เป็นคำสั่งสำหรับตรวจสอบคำตอบ และให้ข้อความโต้ตอบ
 แล้วแตกแขนงไปกรอบอื่น ๆ โดยข้อความโต้ตอบมีได้ไม่เกิน 80 ตัวอักษร

รูปแบบของคำสั่ง

BRANCH [full_score][,wait_time]
 ['value'[condition]] [:frame_no] [, 'string'/message]

เมื่อ full_score กำหนดคะแนนเต็มเมื่อเงื่อนไขเป็น C

wait_time กำหนดเวลาในการรอผู้เรียนอ่านข้อความโต้ตอบ(\$T)
 ถ้าหมดเวลาหรือผู้เรียนกด Enter จะผ่านไปยังการสอนอื่นถัดไป

value คือ คำตอบของผู้เรียน หรือข้อมูลที่ผู้เรียนป้อนให้เครื่อง

condition คือ การกำหนด ค่าคะแนนที่ให้ ความถูกต้อง ผิด หรือไม่ถูกต้อง
สำหรับ value ที่ผู้เรียนป้อน สามารถกำหนดได้ดังนี้

V#no - ให้คะแนน #no คะแนน C - ถูก จะได้คะแนนเต็ม

W - ผิด จะได้ 0 คะแนน N - ไม่ถูกไม่ผิด

frame_no หมายถึงกรอบที่จะถูกเรียกมาทำงานต่อไป

ถ้าเว้นไว้จะเรียกกรอบที่อยู่ถัดจากกรอบนี้มาทำงาน

'string'/message ข้อความโต้ตอบกลับ จะใช้ข้อความ(string) หรือ \$M#no ซึ่ง
เป็นข้อความจากแฟ้มโต้ตอบก็ได้ ข้อความนี้จะแสดงกึ่งกลาง
บรรทัด และมีความยาวไม่เกิน 80 ตัวอักษร

ถ้าไม่มีการกำหนดหมายเลข FRAME ที่จะไป รวมทั้งไม่มีคำสั่ง BRANCH โปรแกรมจะ
กำหนดให้ไปยังกรอบที่อยู่ถัดจากกรอบปัจจุบันเสมอ

@(23,10)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปเลือกคำสั่ง TAL อื่น ๆ'

INPUT @(24,10)'> หรือกดแป้น ENTER เพื่อดูคำอธิบายเกี่ยวกับแฟ้มโต้ตอบ '

BRANCH

'1'N:12

FRAME 14,E

SCREEN

แฟ้มข้อมูลข้อความโต้ตอบ แฟ้มนี้ชื่อเฉพาะว่า CAICOM.MSG

จุดประสงค์เพื่อ เป็นแฟ้มข้อมูลสำหรับผู้สอนสร้างข้อความโต้ตอบที่ใช้บ่อย
เก็บไว้เลือกไปใช้งาน ดังนั้นจะใช้ข้อความโต้ตอบได้ เมื่อมี
การสร้างแฟ้มข้อมูลไว้ก่อนหน้าแล้ว โดยใช้โปรแกรมประมวลคำ

มีรูปแบบดังนี้ หมายเลขรหัสเรียก ข้อความโต้ตอบ

โดยมีข้อกำหนด... ให้มีข้อความได้ไม่เกิน 50 ข้อความ ข้อความละ 100 อักขระ

การเรียกใช้ข้อความ... เรียกโดยใช้คำสั่งใน \$M#no

(เมื่อ #no = หมายเลขรหัสเรียก)

ตัวอย่าง

1 เก่งมาก

2 ถูกต้อง

4 ต่ำมาก

@(23,15)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูคำสั่ง BRANCH'

INPUT @(24,15)'> หรือกดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่ง TAL อื่น ๆ '

BRANCH

'1'N:13

N:6

FRAME 15,E

SCREEN

5 คำสั่งพิเศษ EVALUATE

ใช้กำหนดเกณฑ์การประเมินผลการเรียนว่าผ่านการเรียนหรือไม่

TEACH AUTHORIZING SYSTEM ได้กำหนดเกณฑ์
การประเมินผลการเรียนทั่วไปไว้ดังนี้

คะแนนไม่น้อยกว่า 60% ของคะแนนเต็ม และได้รับเกรดการเรียนดังนี้

ได้คะแนน	100%-80%	ให้เกรด A
ได้คะแนน	79%-70%	ให้เกรด B
ได้คะแนน	69%-60%	ให้เกรด C
ได้คะแนน	59%-50%	ให้เกรด D
ได้คะแนน	49%- 0%	ให้เกรด E

ในการกำหนดเกณฑ์การประเมินผลให้เป็นอย่างอื่น
สามารถเปลี่ยนแปลงได้โดยใช้คำสั่ง EVALUATE
โดยมีรูปแบบคำสั่งดังนี้....

EVALUATE (เกรดที่1:คะแนนต่ำสุดของเกรดที่1, เกรดที่2:คะแนนต่ำสุดของ
เกรดที่2,....., เกรดที่ก:คะแนนต่ำสุดของเกรดที่ก,)

@(23,15)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่ง TAL อื่น ๆ '

INPUT @(24,15)'> หรือกดแป้น ENTER เพื่อดูตัวอย่าง '

BRANCH

'1'N:6

FRAME 16,E

SCREEN

ตัวอย่าง

EVALUATE(A:80,B:70,C:60,D:50,E:0)

EVALUATE(A:90,B:80,C:70,D:60,E:0)

EVALUATE(G:80,P:50,F:0)

คำสั่ง EVALUATE จะควบคุมการประเมินผลเฉพาะหัวข้อการเรียนนั้น ๆ
ต้องกำหนดให้อยู่หลังคำสั่ง FRAME ซึ่งเป็นกรอบที่ผู้เรียนจะต้องใช้
เพื่อไม่ให้เกิดความผิดพลาดควรวัดไว้ตั้งแต่เริ่มต้นบทเรียน


```

@ (23,15)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูคำสั่ง EVALUATE'
INPUT @ (24,15)'> หรือกดแป้น ENTER เพื่อกลับไปเลือกดูคำสั่ง TAL อื่น ๆ '
BRANCH
'1'N:15
N:6
FRAME 17,E
SCREEN

```

คำแนะนำเพิ่มเติม

1. ในการเรียนหรือทดสอบสามารถเลือกไปยังกรอบการเรียนหมายเลขใด ก็ได้ โดยใส่คำตอบเป็น //FRAME#no เมื่อ #no คือ หมายเลขกรอบ
2. ถ้าต้องการหยุดการทดสอบที่กรอบใด ๆ ให้ใส่คำตอบเป็น //END
3. การควบคุมบทเรียนให้เป็นแบบเส้นตรง หรือแบบแตกแขนง หรือแบบสุ่มเลือก สามารถกระทำได้โดยการควบคุมการแตกแขนงในคำสั่ง BRANCH หรือเลือกใช้กรอบชนิดสุ่มเลือก

```

@ (23,15)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น'
INPUT @ (24,15)'> หรือกดแป้น ENTER เพื่อดูคำแนะนำเพิ่มเติมต่อไป '
BRANCH
'1'N:1
FRAME 18,E
SCREEN

```

4. ตัวอย่างกรอบการเรียนเต็มรูป 1 กรอบพร้อมคำอธิบายผลลัพธ์ เป็นดังนี้

```

+-----+
|FRAME x,E,y                                     |<----(1)
|EVALUATE(G:80,P:50,F:0)                         |<----(2)
|SCREEN 'สาริตการให้ภาษา TAL', $Tx             |<----(3)
|ข้อความ                                         |
|"ข้อความชิตซ้าย                               |
|'ข้อความชิตขวา                               |
|^ข้อความจัดกึ่งกลาง                             |
|@(10,1)'ข้อความแสดง ณ บรรทัดที่ 10 คอลัมน์ที่ 1' |
| $C                                             |<----(4)
| $Tx                                           |<----(5)
|INPUT @ (20,10)'เมื่อใส่ข้อความแล้วกด Enter หรือรอให้หมดเวลา?', $Tx |
+-----+ คำสั่งสำหรับ 1 กรอบการเรียนยังมีต่อ...+

```

<<<< คำอธิบาย >>>>

- (1) กรอบที่ x ชนิดคำบรรยาย เข้าใช้ได้ไม่เกิน y ครั้ง
- (2) กำหนดเกณฑ์การประเมินผล
- (3) แสดงข้อความภายใน '....' ณ บรรทัดที่ 1 กึ่งกลางจอภาพ
และเมื่อแสดงข้อความต่าง ๆ แล้ว จะรอคอย x วินาที ก่อนถามคำถาม
- (4) ลบจอภาพทั้งหมด ถ้ามีการแสดงข้อความต่อจากนี้ให้เริ่มแสดง ณ บรรทัดที่ 1
- (5) รอคอย x วินาที

@(23,5)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูคำแนะนำเพิ่มเติมที่ผ่านมา'
INPUT @ (24,5)'> หรือกดแป้น ENTER เพื่อดูคำสั่ง 1 กรอบการเรียนส่วนที่เหลือ '
BRANCH
'1'N:17

FRAME 19,E

SCREEN

```

+-----+
|BRANCH Vx,$Ty                                |<--(6)
|'คำตอบ1'C:2,'ถูกต้อง... เก่งมาก'           |<--(7)
|'คำตอบ2'Vy:3,$Mz                             |<--(8)
|'คำตอบ3'W:4,'พยายามใหม่'                   |<--(9)
|'เลิกเรียน'N:999,'สวัสดิ์'                  |<--(10)
|W:5,'อื่น ๆ ไม่ถูกต้อง...'                  |<--(11)

```

+-----จบคำสั่งสำหรับ 1 กรอบการเรียน-----+

<<<< คำอธิบาย >>>>

- คำสั่งย่อยใน BRANCH ใช้กำกับการตอบของผู้เรียนว่าตอบอะไร ให้คะแนนเท่าไร ผิด หรือถูก หรือไม่ถูกไม่ผิด แยกแขนงไปกรอบใด แสดง ข้อความ ได้ตอบอะไร
- (6) ถ้าตอบถูกได้คะแนนเต็ม x คะแนน ข้อความได้ตอบแสดงนาน y วินาที
 - (7) ถ้าตอบว่า คำตอบ1 ให้ถูก ได้คะแนนเต็ม x คะแนน บวกคะแนนเต็มด้วย x แสดงข้อความว่าถูกต้อง... เก่งมาก นาน y วินาที แล้วไปกรอบที่ 2
 - (8) ถ้าตอบว่า คำตอบ2 จะเป็นคำตอบที่ได้คะแนน y คะแนน บวกคะแนนเต็มด้วย x นำข้อความรหัส z จากแฟ้มข้อความได้ตอบมาแสดง นาน y วินาที แล้วไปกรอบที่ 3
 - (9) ถ้าตอบว่า คำตอบ3 ให้ผิด ได้คะแนน 0 คะแนน บวกคะแนนเต็มด้วย x แสดงข้อความว่า พยายามใหม่ นาน y วินาที แล้วไปกรอบที่ 4
 - (10) ถ้าตอบว่า เลิกเรียน ให้ไม่มีถูกไม่ผิด ไม่บวกคะแนนได้และคะแนนเต็ม แสดงข้อความว่า สวัสดิ์ นาน y วินาที แล้วไปจบการเรียน แสดงสรุปผลการเรียน
 - (11) คำตอบอื่นๆ นอกจากข้างต้น ให้ผิด ได้ 0 คะแนน บวกคะแนนเต็มด้วย x แสดงข้อความว่า อื่น ๆ ไม่ถูกต้อง... นาน y วินาที แล้วไปกรอบที่ 5

@(23,5)'> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูคำสั่ง 1 กรอบการเรียนที่ผ่านมา'
INPUT @ (24,5)'> หรือกดแป้น ENTER เพื่อกลับไปเลือกดูเรื่องอื่น '

BRANCH

'1'N:18

N:1

FRAME 1,E
SCREEN

หัวข้อที่ 5 ตัวอย่างการสร้างบทเรียนสำหรับใช้กับระบบ TEACH

ตัวอย่างนี้จะแสดงการสร้างบทเรียนเรื่องการบวกเลข

ทั้งในส่วนการสร้างแฟ้มโครงสร้าง และแฟ้มเนื้อหาบทเรียน

โดยจะเริ่มต้นจากบทเรียนสำเร็จรูปซึ่งผู้สอนสร้างไว้บนกระดาษ

แล้วเพิ่มเติมคำสั่งภาษา TAL ให้เป็นบทเรียนโปรแกรมที่ใช้กับ

ระบบ TEACH

สำหรับการทำงานของบทเรียนจากตัวอย่างนี้

ให้ดูการสาธิตได้ในหัวข้อที่ 6

INPUT @ (24,10) '> กดแป้น ENTER เพื่อดูตัวอย่างการสร้างแฟ้มโครงสร้าง '
BRANCH
'1'N:1

FRAME 2,E

SCREEN 'ตัวอย่างการสร้างแฟ้มโครงสร้าง'

บทเรียนสำเร็จรูปซึ่งผู้สอนสร้างไว้บนกระดาษ ได้แบ่งบทเรียนออกเป็นหัวข้อย่อย
ดังนี้....

- หัวข้อที่ 1 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด
- หัวข้อที่ 2 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด
- หัวข้อที่ 3 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด
- หัวข้อที่ 4 การบวกเลขสองหลักสองจำนวนที่มีตัวทด

จากหัวข้อต่าง ๆ ให้พิมพ์ด้วยโปรแกรมประมวลคำจัดทำเป็นแฟ้มโครงสร้าง
ซึ่งเป็นแฟ้มข้อความที่มีส่วนขยายของชื่อแฟ้มเป็น .STR ที่มีรูปแบบดังนี้

ชื่อแฟ้มหัวข้อ, ชื่อหัวข้อ (โดยชื่อแฟ้มที่มีรหัส 000 จะเป็นชื่อวิชา)

บทเรียนนี้ตั้งชื่อแฟ้มโครงสร้างเป็น MAT.STR ข้อความในแฟ้มมีดังนี้

MAT000 วิชาคณิตศาสตร์ประถม 1
 MAT001 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด
 MAT002 การบวกเลขหลักเดียวสองจำนวนที่มีตัวทด
 MAT003 การบวกเลขสองหลักสองจำนวนที่ไม่มีตัวทด
 MAT004 การบวกเลขสองหลักสองจำนวนที่มีตัวทด
 @ (23, 15) '> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปดูค่านำ'
 INPUT @ (24, 15) '> หรือกดแป้น ENTER เพื่อเตรียมเพิ่มเนื้อหาบทเรียน '
 BRANCH
 '1'N:1

FRAME 3,E

SCREEN ' การเตรียมเพิ่มเนื้อหาบทเรียน '

สำหรับเนื้อหาของแต่ละหัวข้อให้บรรจุลงในแฟ้มข้อความ
 ที่มีส่วนขยายของชื่อแฟ้มเป็น .TXT
 และชื่อแฟ้มเป็นชื่อเดียวกับชื่อแฟ้มหัวข้อที่อยู่ในแฟ้มโครงสร้าง

ในตัวอย่างนี้.....

จะสร้างเฉพาะหัวข้อที่ 1 เรื่องการบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด
 สำหรับหัวข้อที่เหลือสามารถจัดเป็นกรอบการเรียนได้ในลักษณะเดียวกัน

การเตรียมบทเรียนสำหรับใช้บนไมโครคอมพิวเตอร์
 ให้พิมพ์เป็นแฟ้มข้อความด้วยโปรแกรมประมวลคำ (word processor)
 เช่น RW หรือ CW

ตัวอย่างหัวข้อที่ 1 จะต้องตั้งชื่อแฟ้มเป็น MAT001.TXT
 เนื้อหาภายในหัวข้อนี้เขียนโดยใช้ภาษา TAL (Teach Authoring Language)
 ซึ่งเป็นภาษาสำหรับควบคุมการนำเสนอบทเรียนอย่างง่าย ๆ

@ (23, 7) '> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อกลับไปตัวอย่างการสร้างแฟ้มโครงสร้าง'
 INPUT @ (24, 7) '> หรือกดแป้น ENTER เพื่อเตรียมเขียนบทเรียนโปรแกรม '
 BRANCH
 '1'N:2

FRAME 4,E

SCREEN

หัวข้อที่ 1 การบวกเลขหลักเดียวสองจำนวนที่ไม่มีตัวทด
 บทเรียนนี้จัดแบ่งเป็น 6 กรอบการเรียน มีเนื้อหาในแต่ละกรอบดังนี้.....

| INPUT @ (20,35) 'กด Enter ไปยังหน้าต่อไป' |

+-----+

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ
และเมื่อผู้เรียนกด Enter จะผ่านไปทำงานในกรอบที่ 2

INPUT @ (22,5) '> กดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น '
BRANCH
N:4

FRAME 6,E
SCREEN

กรอบที่ 2 เมื่อเขียนสำหรับใช้กับระบบ TEACH ได้ดังนี้

```
+-----+
|FRAME 2,E
|SCREEN 'กรอบที่ 2 ตัวอย่างที่ 1'
|
| ตัวอย่างเช่น 1+1 มีค่าเท่ากับ 2
|INPUT @ (20,35) 'กด Enter ไปยังหน้าต่อไป'
+-----+
```

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ
และเมื่อผู้เรียนกด Enter จะผ่านไปทำงานในกรอบที่ 3

INPUT @ (22,5) '> กดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น '
BRANCH
N:4

FRAME 7,E
SCREEN

กรอบที่ 3 เมื่อเขียนสำหรับใช้กับระบบ TEACH ได้ดังนี้

```
+-----+
|FRAME 3,E
|SCREEN 'กรอบที่ 3 ตัวอย่างที่ 2'
|
| ตัวอย่างเช่น 2+3 มีค่าเท่ากับ 5
+-----+
```

| INPUT @(20,35) 'กด Enter ไปยังหน้าต่อไป' |

+-----+

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ
และเมื่อผู้เรียนกด Enter จะผ่านไปทำงานในกรอบที่ 4

INPUT @(22,5)'> กดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น '
BRANCH
N:4

FRAME 8,E

SCREEN

กรอบที่ 4 เมื่อเขียนสำหรับใช้กับระบบ TEACH ได้ดังนี้

+-----+

| FRAME 4,Q,2 |

| SCREEN 'กรอบที่ 4 โจทย์ทดสอบที่ 1' |

| | |

| ผลบวกของ 2 กับ 3 มีค่าเท่าไร |

| ก. 4 |

| ข. 5 |

| ค. 6 |

| ง. 7 |

| จ. 8 |

| INPUT @(20,10) 'เลือกคำตอบที่ถูกโดยกด ก ข ค ง หรือ จ แล้วกด Enter ? ' |

| BRANCH V2,\$T10 |

| 'ข'C:5,'ถูกต้อง... เก่งมาก' |

| W:4,'พยายามใหม่อีกครั้ง' |

+-----+

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ และให้ผู้เรียนใส่ตัวเลข
ซึ่งถ้าตอบผิด จะมีข้อความโต้ตอบว่า พยายามใหม่อีกครั้ง

แล้วให้ผู้เรียนมายังกรอบที่ 4 อีกครั้ง โดยให้โอกาสตอบไม่เกิน 2 ครั้ง

ถ้าตอบถูก จะมีข้อความโต้ตอบว่า ถูกต้อง... เก่งมาก แล้วผ่านไปยังกรอบที่ 5

กรณีที่ตอบไม่ถูกทั้ง 2 ครั้งจะผ่านไปยังกรอบที่ 5

INPUT @(24,5)'> กดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น '

BRANCH

N:4

FRAME 9,E

SCREEN

กรอบที่ 5 เมื่อเขียนสำหรับใช้กับระบบ TEACH ได้ดังนี้

```

+-----+
|FRAME 5,Q,2|
|SCREEN 'กรอบที่ 5 โจทย์ทดสอบที่ 2'|
|   ผลบวกของ 4 กับ 2 มีค่าเท่าไร|
|   ก. 4|
|   ข. 5|
|   ค. 6|
|   ง. 7|
|   จ. 8|
|INPUT @(20,10) 'เลือกคำตอบที่ถูกโดยกด ก ข ค ง หรือ จ แล้วกด Enter ?'|
|BRANCH V2,$T10|
|'ค'C:6,'เก่งมาก'|
|W:5,'คิดอีกครั้งซิ...'|
+-----+

```

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ และให้ผู้เรียนใส่ตัวเลขซึ่งถ้าตอบผิด จะมีข้อความโต้ตอบว่า คิดอีกครั้งซิ... และให้ผู้เรียนกลับมายังกรอบที่ 5 อีกครั้ง โดยให้โอกาสตอบไม่เกิน 2 ครั้ง ถ้าตอบถูกจะมีข้อความโต้ตอบว่า เก่งมาก แล้วผ่านไปยังกรอบที่ 6 กรณีที่ตอบไม่ถูกทั้ง 2 ครั้งจะผ่านไปยังกรอบที่ 6

```

INPUT @(24,5) '> กดปุ่ม ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น'
BRANCH
N:4

```

FRAME 10,E

SCREEN

กรอบที่ 6 เมื่อเขียนสำหรับใช้กับระบบ TEACH ได้ดังนี้

```

+-----+
|FRAME 6,Q,2|
|SCREEN 'กรอบที่ 6 โจทย์ทดสอบที่ 3'|
|   |
|   |
|   ผลบวกของ 5 กับ 3 มีค่าเท่าไร|
|   |
|   ก. 4|
|   ข. 5|
|   ค. 6|
|   ง. 7|
|   จ. 8|
|INPUT @(20,10) 'เลือกคำตอบที่ถูกโดยกด ก ข ค ง หรือ จ แล้วกด Enter ?'|

```



```

| BRANCH V2,$T10 |
| 'จ'C:999,'ยอดเยี่ยม...' |
| W:6,'ลองใหม่คะ...' |

```

กรอบนี้จะนำเสนอข้อความภายใต้คำสั่ง SCREEN บนจอภาพ และให้ผู้เรียนใส่ตัวเลข
ซึ่งถ้าตอบผิด จะมีข้อความโต้ตอบว่า ลองใหม่คะ...

และให้ผู้เรียนกลับมายังกรอบที่ 6 อีกครั้งหนึ่ง โดยให้โอกาสตอบไม่เกิน 2 ครั้ง

ถ้าตอบถูก จะมีข้อความโต้ตอบว่า ยอดเยี่ยม... แล้วเลิกการเรียน

กรณีที่ตอบไม่ถูกต้องทั้ง 2 ครั้งก็จะผ่านไปเลิกการเรียน

INPUT @ (24,10) '> กดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น'

BRANCH

N:4

FRAME 11,E

SCREEN

จะเห็นได้ว่าภาษา TAL ช่วยเพิ่มเทคนิคการสอน

และจัดลำดับขั้นตอนการนำเสนอบทเรียน

ทำให้สามารถสร้างความสนใจและจูงใจผู้เรียนที่จะเรียนรู้มากขึ้น

TAL เป็นภาษาที่มีรูปแบบและโครงสร้างง่าย ๆ

สามารถทำความเข้าใจเพื่อใช้งานได้ในเวลาไม่นาน

ดังนั้นการสร้างบทเรียน CAI จึงไม่ใช่เรื่องยากอีกต่อไป

@ (23,5) '> พิมพ์ 1 แล้วกดแป้น ENTER เพื่อไปเลือกดูการเขียนบทเรียนโปรแกรมกรอบอื่น'

INPUT @ (24,5) '> หรือกดแป้น ENTER จบการเรียนหัวข้อนี้ '

BRANCH

'1'N:4

N:999

FRAME 1,E

SCREEN 'กรอบที่ 1 คำอธิบายเกี่ยวกับการบวก'

การบวกเลขสองจำนวน คือ การนำเลขทั้งสองจำนวนมารวมกัน

INPUT @ (20,35) 'กด Enter ไปยังหน้าต่อไป'

FRAME 2,E

SCREEN 'กรอบที่ 2 ตัวอย่างที่ 1'

ตัวอย่างเช่น $1+1$ มีค่าเท่ากับ 2

INPUT @ (20,35) 'กด Enter ไปยังหน้าต่อไป'

FRAME 3,E

SCREEN 'กรอบที่ 3 ตัวอย่างที่ 2'

ตัวอย่างเช่น $2+3$ มีค่าเท่ากับ 5

INPUT @ (20,35) 'กด Enter ไปยังหน้าต่อไป'

FRAME 4,Q,2

SCREEN 'กรอบที่ 4 โจทย์ทดสอบที่ 1'

ผลบวกของ 2 กับ 3 มีค่าเท่าไร

ก. 4

ข. 5

ค. 6

ง. 7

จ. 8

INPUT @ (20,10) 'เลือกคำตอบที่ถูกต้องโดยกด ก ข ค ง หรือ จ แล้วกด Enter ? '

BRANCH V2,\$T10

'ข' C:5, 'ถูกต้อง... เก่งมาก'

W:4, 'พยายามใหม่อีกครั้ง'

FRAME 5,Q,2

SCREEN 'กรอบที่ 5 โจทย์ทดสอบที่ 2'

ผลบวกของ 4 กับ 2 มีค่าเท่าไร

ก. 4

ข. 5

ค. 6

ง. 7

จ. 8

INPUT @ (20,10) 'เลือกคำตอบที่ถูกโดยกด ก ข ค ง หรือ จ แล้วกด Enter ? '

BRANCH V2,\$T10

'ค'C:6, 'เก่งมาก'

W:5, 'คิดอีกครึ่งซี...'

FRAME 6,Q,2

SCREEN 'กรอบที่ 6 โจทย์ทดสอบที่ 3'

ผลบวกของ 5 กับ 3 มีค่าเท่าไร

ก. 4

ข. 5

ค. 6

ง. 7

จ. 8

INPUT @ (20,10) 'เลือกคำตอบที่ถูกโดยกด ก ข ค ง หรือ จ แล้วกด Enter ? '

BRANCH V2,\$T10

'จ'C:999, 'ยอดเยี่ยม...'

W:6, 'ลองใหม่คะ...'

ภาคผนวก จ.

สรุปการประเมินผล

การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

สรุปการประเมินผล
การใช้ TEACH : ระบบสร้างบทเรียนด้วยตนเองบนไมโครคอมพิวเตอร์

สรุปผลความคิดเห็นการใช้ TEACH ของกลุ่มตัวอย่าง จำแนกเป็นรายชื่อ
 ตามความถี่จากมาก(5) ไปหาน้อย(1) หรือจากยาก(5) ไปทางง่าย(1) เป็นร้อยละ ได้ดังนี้

ความคิดเห็นการใช้ TEACH	5	4	3	2	1
1. ท่านสามารถศึกษาทำความเข้าใจระบบ TEACH เพื่อนำไปสร้างบทเรียนบนไมโครคอมพิวเตอร์ ได้ยากหรือง่ายเพียงใด	14.29	78.57	7.14	0.00	0.00
2. ท่านเห็นวาระบบ TEACH มีความซับซ้อนมากน้อยเพียงใด	0.00	64.29	35.71	0.00	0.00
3. ท่านสามารถสร้างบทเรียนบนไมโครคอมพิวเตอร์ เพื่อใช้กับระบบ TEACH ได้ยากหรือง่ายเพียงใด	21.43	64.29	14.29	0.00	0.00
4. ท่านใช้เวลาในการศึกษาระบบ TEACH มากน้อยเพียงใด จึงสามารถใช้งานได้	28.57	57.14	14.29	0.00	0.00
5. ท่านเห็นว่าการนำระบบ TEACH ไปใช้ช่วยการเรียนการสอนมีความยากง่ายเพียงใด	18.18	36.36	36.36	9.09	0.00
6. ท่านเห็นว่าการนำระบบ TEACH ไปใช้ช่วยการเรียนการสอนสามารถใช้ได้ผลตามความต้องการมากน้อยเพียงใด	0.00	7.14	50.00	28.57	14.29
7. ท่านเห็นวาระบบ TEACH จัดแบ่งเนื้อหาบทเรียนเป็นกรอบการเรียนมีความเหมาะสมมากน้อยเพียงใด	16.67	33.33	25.00	25.00	0.00
8. ท่านเห็นว่าการทดสอบบทเรียนทำได้ยากง่ายเพียงใด	0.00	14.29	50.00	28.57	7.14
9. ท่านเห็นว่าผลที่ได้จากบันทึกพฤติกรรมการเรียนเป็นประโยชน์มากน้อยเพียงใด	57.14	28.57	14.29	0.00	0.00
10. ท่านเห็นว่าความสามารถในการสรุปผลการเรียนเป็นประโยชน์มากน้อยเพียงใด	57.14	35.71	7.14	0.00	0.00
11. ท่านเห็นว่าภาษา TAL ที่ใช้ในระบบ TEACH มีความเหมาะสมมากน้อยเพียงใด	0.00	23.08	61.54	15.38	0.00
12. ท่านเห็นว่าการใช้ภาษา TAL มีความยากง่ายมากน้อยเพียงใด	0.00	14.29	28.57	57.14	0.00
13. ท่านใช้เวลาในการศึกษาภาษา TAL เพื่อใช้งานมากน้อยเพียงใด	7.69	30.77	38.46	15.38	7.69

ความเห็นการใช้ TEACH	5	4	3	2	1
14. ท่านเห็นว่า ภาษา TAL มีความเหมาะสมกับบทเรียนสำเร็จรูปมากน้อยเพียงใด	14.29	42.86	28.57	7.14	7.14
15. ท่านเห็นว่า การใช้ภาษา TAL ควบคุมลำดับขั้นตอนการนำเสนอเนื้อหาบนจอภาพทำได้ยากง่ายมากน้อยเพียงใด	21.43	35.71	35.71	7.14	0.00
16. ท่านเห็นว่า ภาษา TAL ให้ผลการควบคุมการเรียนการสอนเป็นไปตามความต้องการมากน้อยเพียงใด	0.00	0.00	50.00	35.71	14.29
17. ท่านเห็นว่า การแยกบทเรียนกับโปรแกรมควบคุมในระบบ TEACH มีความเหมาะสมมากน้อยเพียงใด	21.43	35.71	28.57	7.14	7.14
18. ท่านเห็นว่า ภาษา TAL สามารถควบคุมการแตกแขนงได้มากน้อยเพียงใด	21.43	35.71	42.86	0.00	0.00
19. ท่านเห็นว่า ประโยชน์ของระบบ TEACH ในการนำไปใช้สร้างบทเรียนมีมากน้อยเพียงใด	0.00	28.57	14.29	42.86	14.29
20. ท่านเห็นว่า การนำบทเรียนไปใช้กับผู้เรียนทำได้ยากง่ายเพียงใด	14.29	28.57	28.57	14.29	14.29
21. ท่านเห็นว่า ระบบ TEACH เอื้อความสะดวกในการติดตามพฤติกรรมการเรียนมากน้อยเพียงใด	7.14	28.57	50.00	14.29	0.00
22. ท่านเห็นว่า ระบบ TEACH เอื้อความสะดวกในการดูสรุปผลการเรียนของผู้เรียนมากน้อยเพียงใด	28.57	21.43	42.86	7.14	0.00
23. ข้อเสนอแนะอื่น ๆ					
1. การสร้างบทเรียนด้วยโปรแกรม TEACH ยังมีข้อจำกัด ควรทำโปรแกรมให้ใช้ตารางได้ และถ้ามีรูปภาพยิ่งดี ข้อดีของโปรแกรม TEACH คือ สามารถใช้บททวนและทำแบบฝึกหัด					
2. ควรจะหาหรือนำโปรแกรมที่มีรูปภาพมาใช้ร่วมกับโปรแกรม TEACH เพราะจะช่วยให้เกิดความสนใจในเนื้อหาได้มากยิ่งขึ้น โดยโปรแกรม TEACH เหมาะสำหรับการใช้บททวน และเป็นข้อสอบที่สามารถกระทำได้ เพราะเกิดความรวดเร็วสำหรับผู้ทำ และประเมินผลได้อย่างรวดเร็ว					

ประวัติผู้เขียน

ชื่อ	นายรินทร์ ศุภศรี	
วัน เดือน ปีเกิด	15 ตุลาคม 2499	
วุฒิการศึกษา	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ (นิสิกส์)	มหาวิทยาลัยรามคำแหง	2522
ทุนการศึกษา	ทุนมูลนิธิเพื่อการสื่อสารและคอมพิวเตอร์	
ตำแหน่งและสถานที่ทำงาน	วิทยากร 2 สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ศึกษา สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี	