การสร้างโปรแกรมคอมพิวเตอร์ช่วยการเขียน

ผ่าน บาลโพธิ์

ความนำ

ทักษะการเขียนเป็นทักษะที่ยาก โดยเฉพาะการเขียนภาษาต่างประเทศ และที่สำคัญยิ่งก็คือในการเรียนทักษะการเขียนนั้น ผู้เรียนควรได้ฝึกเขียนอย่างเพียงพอ จึงจะสามารถเขียนได้ตามจุดมุ่งหมายของบทเรียนและหลักสูตร แต่เป็นที่น่าเสียดายที่ผู้สอนจำนวนไม่น้อย อาจไม่สามารถให้ผู้เรียนเขียนได้มากนัก เนื่องจากต้องสอนหลายกลุ่ม และแต่ละกลุ่มก็มักมีผู้เรียนจำนวนมาก ประกอบกับมีผู้สอนอีกจำนวนหนึ่งที่จำเป็นต้องปฏิบัติภารกิจอื่นที่เกี่ยวข้องกับการเรียนการสอน จึงทำให้การตรวจงานเขียนของผู้เรียนเป็นภาระที่หนักมาก เพราะการตรวจงานอย่างละเอียดรอบคอบนั้นใช้เวลามาก และในบางกรณีผู้สอนบางท่านอาจไม่สามารถตรวจงานเขียนได้ทันเวลา นอกจากนี้ในการฝึกเขียนนั้น ผู้เรียนอาจพัฒนาทักษะนี้ได้ดียิ่งขึ้นหากผู้เรียนได้เรียนด้วยตนเองหลังจากที่ได้เรียนในห้องเรียนปกติไปแล้ว (Jones, 1998 : 378) แต่การที่ผู้เรียนจะเรียนหรือฝึกการเขียนด้วยตนเองได้ดีนั้น ต้องมีผู้คอยช่วยเหลือแนะนำ ซึ่งก็เป็นไปได้ยากอีกเช่นกันที่ผู้สอนจะสามารถให้ความช่วยเหลือผู้เรียนทุกคนนอกเวลาการเรียนการสอนได้อย่างทั่วถึง ดังนั้น เพื่อช่วยบรรเทาหรือแก้ไขปัญหาดังกล่าว การสร้างโปรแกรมคอมพิวเตอร์ช่วยการเขียนจะช่วยให้ผู้เรียนมีโอกาสได้ฝึกเขียนอย่างเพียงพอ และได้รับการตรวจอย่างรวดเร็ว เพราะโปรแกรมช่วยการเขียนที่สร้างขึ้นจะสามารถตรวจการเขียน ให้คำชี้แนะ อธิบายข้อผิดพลาดต่าง ๆ ได้ และสามารถบันทึกข้อผิดพลาดต่าง ๆ ของผู้เรียนได้อีกด้วย และในการเรียนกับโปรแกรมคอมพิวเตอร์ที่ผู้สอนสร้างขึ้นนั้น นอกจากผู้เรียนจะได้รับคำแนะนำและคำอธิบายต่าง ๆ ที่ผู้สอนบรรจุไว้ในโปรแกรมแล้ว ผู้เรียนยังมีโอกาสที่จะได้ทดลองใช้ภาษาหลายรูปแบบ บทความเรื่องนี้จึงเสนอแนะแนวทางการสร้างโปรแกรมคอมพิวเตอร์ช่วยการเขียน พอเป็นแนวทางที่ท่านจะนำไปสร้างโปรแกรมขึ้นใช้เองต่อไป และท้ายสุด บทความนี้จะให้โปรแกรมตัวอย่างที่ท่านสามารถนำไปปรับปรุงหรือขยายให้สมบูรณ์ต่อไป

การเริ่มต้นและการออกแบบโปรแกรม

ปัญหาที่สำคัญของผู้เขียนโปรแกรมมือใหม่จำนวนไม่น้อย คือการไม่ทราบว่าจะเริ่มต้นการสร้างโปรแกรมอย่างไรจึงจะสร้างโปรแกรมได้ง่ายและได้ผลดีที่สุด การเริ่มต้นที่ดีควรเริ่มจากการกำหนดจุดมุ่งหมายของโปรแกรมให้ชัดเจน เช่น จะใช้โปรแกรมกับใคร ระดับใด เพราะลักษณะของผู้ใช้โปรแกรม (target user) เป็นสิ่งที่ต้องนำมาพิจารณาในการกำหนดเนื้อหา (content) และวิธีสอน นอกจากนี้ ควรมีการกำหนดว่าหลังจากใช้โปรแกรมช่วยการเขียนแล้ว ผู้เรียนจะมีความรู้ความสามารถ หรือทักษะเฉพาะเรื่องใด จากนั้นจึงลองวาดภาพลักษณะที่สำคัญทั่วไปของโปรแกรมว่าควรมีลักษณะอย่างไร สิ่งที่จะช่วยให้ท่านสามารถวาดภาพโปรแกรมของท่านได้ก็คือจุดมุ่งหมายและลักษณะทั่วไปของโปรแกรมช่วยการเรียนภาษา (computer-assisted language learning program) จุดมุ่งหมายจะช่วยให้ทราบลักษณะที่สำคัญของผู้เรียนหรือผู้ใช้โปรแกรมที่ท่านต้องคำนึงถึงในทุก

ขั้นตอนของการสร้างโปรแกรม และลักษณะของโปรแกรมช่วยการเรียนภาษาจะช่วยให้ทราบลักษณะและขั้นตอนการทำงานของโปรแกรม

โดยทั่วไปแล้วโปรแกรมช่วยการเรียนภาษาจะเริ่มต้นจากการให้ผู้เรียนอ่านข้อความ ดูภาพ หรือฟังการบรรยาย การพูด การสนทนา หรือประกาศ หรือโปรแกรมอาจกำหนดหัวข้อเรื่อง (topic) แล้วจึงให้ผู้เรียนพิมพ์คำตอบ จากนั้นโปรแกรมจะตรวจและให้คำแนะนำที่จำเป็น ดังนั้นในการเขียนจุดมุ่งหมายของโปรแกรมท่านควรระบุให้ชัดเจนเสมอว่าต้องการให้ผู้เรียนเขียนจากสิ่งใด ท่านจึงจะมีข้อมูลเพียงพอที่จะออกแบบโปรแกรมต่อไป

อย่างไรก็ตามภาพของโปรแกรมที่ได้ในขั้นนี้เป็นเพียงภาพกว้าง ๆ ของโปรแกรมเท่านั้น แต่ก็เพียงพอที่จะนำไปสู่การพิจารณาความเป็นไปได้ของโปรแกรมว่าท่านจะสามารถสร้างโปรแกรมที่ท่านต้องการได้หรือไม่ เพราะแม้ท่านจะมีความคิดที่ดี แต่หากมีปัญหาทางเทคนิค เช่น เครื่องมือที่จะใช้สร้างโปรแกรมไม่สามารถทำงานตามที่ท่านต้องการ ท่านก็จะไม่สามารถสร้างโปรแกรมนั้นได้

ในการศึกษาและพิจารณาความเป็นไปได้ของการสร้างโปรแกรมที่ท่านต้องการนั้น หากท่านยังมีประสบการณ์ในการสร้างโปรแกรมไม่มากนัก ท่านควรปรึกษากับผู้เชี่ยวชาญด้านคอมพิวเตอร์ช่วยการเรียนภาษา หรือผู้ที่มีความรู้ความสามารถด้านคอมพิวเตอร์ จึงจะช่วยให้การพิจารณาถูกต้องยิ่งขึ้น เป็นต้นว่าสิ่งที่ท่านคิดว่าทำไม่ได้อาจไม่จริง เพราะอาจมีวิธีแก้ไขปัญหานั้น ในทำนองเดียวกัน สิ่งที่ท่านคิดว่าทำได้ก็อาจทำไม่ได้ เพราะอาจเป็นข้อจำกัดของเครื่องมือหรือโปรแกรมแม่ที่ใช้ สิ่งที่ควรปรึกษากับผู้เชี่ยวชาญได้แก่ขั้นตอนการทำงานของโปรแกรมที่ท่านจะสร้าง การตรวจคำตอบ การอธิบายหรือชี้แนะเมื่อผู้เรียนเขียนผิด การให้ความช่วยเหลือ (on-line help) และการบันทึกข้อมูลจากการฝึกปฏิบัติของผู้เรียน

ขั้นต่อไปเป็นการออกแบบโปรแกรม ซึ่งจะเป็นการกำหนดลักษณะ ขั้นตอน และรายละเอียดที่สำคัญอย่างน้อยสี่ประการ ประการแรก ท่านต้องกำหนดขั้นตอนการทำงานของโปรแกรม เช่น จะให้โปรแกรมเริ่มต้นอย่างไร ทำงานในขั้นต่อ ๆ ไปอย่างไร และจะจบอย่างไร เป็นต้นว่าก่อนจบจะเสนอรายงานผลการฝึกเขียนของผู้เรียน ซึ่งอาจเสนอทั้งประโยคที่ผิดและข้อผิดพลาดต่าง ๆ จากนั้นจะจบโปรแกรม และออกไปสู่เมนู (menu) และท้ายที่สุดจะออกไประบบปฏิบัติการ DOS หรือ Windows ประการที่สองเป็นการกำหนดลักษณะการติดต่อระหว่างผู้เรียนกับโปรแกรม เช่น จะให้ผู้เรียนพิมพ์ประโยคแล้วกดแป้น Enter หรือจะให้เลื่อนตัวชี้ตำแหน่งของเมาส์ไปที่ปุ่ม OK แล้วกดเมาส์ หรือในกรณีที่ท่านให้ผู้เรียนแต่งประโยคจากคำที่กำหนดให้ ท่านจะต้องกำหนดว่าจะให้ผู้เรียนพิมพ์ประโยคที่ต้องการ หรือจะให้คัดลอก (copy) คำที่ให้มา แล้วนำคำเหล่านั้นไปใส่ที่ช่องคำตอบ และอาจให้ผู้เรียนพิมพ์คำอื่นที่จำเป็นได้ด้วย ประการที่สามจะเป็นการกำหนดชนิดและลักษณะของข้อมูล (data type and data structure) สำหรับโปรแกรมช่วยการเขียนนั้น ในขั้นนี้จะเป็นการกำหนดว่าข้อความที่จะให้ผู้เรียนเขียนเป็นข้อความ (text) ในระดับคำ วลี ประโยค หรือย่อหน้า และข้อความนั้นจะมีความยาวเท่าใด เช่น มีความยาวไม่เกิน 10 ตัวอักษร (characters) รวมทั้งช่องว่าง (white space) 40 ตัวอักษร 80 ตัวอักษร หรือจะไม่จำกัดความยาวและจำนวนบรรทัด การกำหนดรายละเอียดนี้สำคัญยิ่งเพราะจะมีผลต่อการการรับคำตอบ และการแสดงคำตอบนั้น หากกำหนดไม่ถูกต้องโปรแกรมอาจบันทึกคำตอบไม่ครบถ้วน หรือเมื่อโปรแกรมแสดงผลข้อความที่ผู้เรียนพิมพ์อาจมีข้อความบางส่วนขาดหายไป หรืออาจทำให้ข้อความหรือภาพที่ส่วนบนของจอภาพหายไป สำหรับสิ่งสุดท้ายที่ท่านต้องกำหนดจะเป็นรายละเอียดเกี่ยวกับการบันทึกข้อมูลและการแสดงผลข้อมูล ซึ่งจะเป็นการกำหนดทั้งวิธีการและรูปแบบการแสดงผล เช่น การบันทึกข้อผิดพลาดต่าง ๆของผู้เรียน ก็ต้องกำหนดว่าจะบันทึกข้อผิดพลาดใดบ้าง และจะบันทึกอย่างไร จะเสนอรายงาน (report) ข้อผิดพลาดต่าง ๆเมื่อใด และจะเสนอที่ส่วนใดของจอภาพ

จุดมุ่งหมายของโปรแกรม

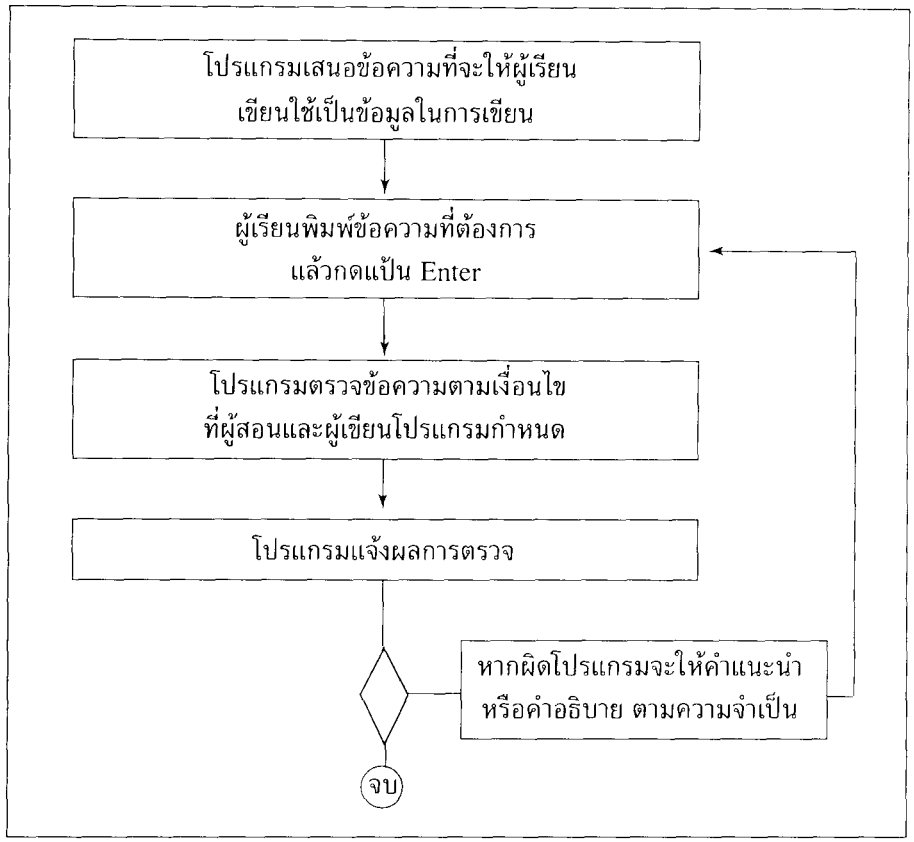
จุดมุ่งหมายที่สำคัญของโปรแกรมคอมพิวเตอร์ช่วยการเขียนได้แก่การแก้ไขปัญหาด้านการเขียนอย่างใดอย่างหนึ่งของผู้เรียน และการช่วยให้ผู้เรียนมีโอกาสได้ฝึกเขียนภาษาที่เรียนมากขึ้น ซึ่งจะเป็นการฝึกการใช้ภาษาตามความรู้ความสามารถของผู้เรียนแต่ละคนในบรรยากาศที่ผู้เรียนไม่ต้องวิตกกังวลเกี่ยวกับผู้สอนและเพื่อนร่วมชั้น ดังนั้น ในการกำหนดจุดม่งหมายของโปรแกรมจึงต้องมีการกำหนดรายละเอียดต่าง ๆ ที่เป็นรูปธรรมเพียงพอที่ผู้เขียนโปรแกรมจะสามารถนำไปปฏิบัติได้

สิ่งสำคัญที่ควรกำหนดไว้ในจุดมุ่งหมายได้แก่ระดับของผู้เรียน ทักษะเฉพาะ หรือหน้าที่ทางภาษา (function) ที่ต้องการให้ผู้เรียนฝึก และระดับของข้อความที่จะให้ผู้เรียนเขียน เช่น

1. *ระดับคำ และวลี*
   - Use “make” or “do” in the following sentences.
   - Supply the following sentences with the correct prepositions.
   - Modify a noun with a prepositional phrase / clause.
   - Transform each sentence into a phrase.
   - Write a phrase to inform the reader of the source of information.
2. *ระดับประโยค*
   - Write a sensible sentence from the words given.
   - Write a good topic sentence from the information given.
   - Paraphrase the following sentences.
   - Write a sentence to express a similarity between two things / events / persons.
3. *ระดับย่อหน้า*
   - Write a paragraph of about 100 words to describe a person.
   - Write a paragraph to compare similarities between two things.
   - Write a paragraph to describe one’s daily activities.
   - Write a short paragraph to define a word, a concept, or an idea.
   - Write a paragraph to point out differences between two things.
   - Write a paragraph to describe a graph (e.g. bar graph, circle graph).

ลักษณะและขั้นตอนที่สำคัญของโปรแกรม

ในการสร้างโปรแกรมช่วยการเรียนภาษา ท่านควรทราบลักษณะที่สำคัญของโปรแกรมที่ต้องการอย่างชัดเจนและเป็นรูปธรรมเพียงพอก่อน จึงจะสามารถออกแบบ และกำหนดรายละเอียดต่าง ๆ ได้อย่างเหมาะสม ลักษณะและขั้นตอนที่สำคัญของโปรแกรมช่วยการเขียนมีดังนี้



รูปที่ 1 : ลักษณะและขั้นตอนการทำงานของโปรแกรม

จากรูปที่ 1 จะเห็นได้ว่าการทำงานของโปรแกรมไม่ยุ่งยากหรือสลับซับซ้อนแต่อย่างใด แต่เนื่องจากโปรแกรมช่วยการเรียนภาษาทั่วไปนั้นจะเป็นโปรแกรมที่ให้ผู้เรียนฝึกใช้ภาษาด้วยตนเอง (self-access learning) ดังนั้น การสร้างโปรแกรมช่วยการเขียนจึงต้องให้ผู้เรียนสามารถควบคุมการใช้โปรแกรมได้มาก และควรมีการให้ความช่วยเหลือ (help) ในระดับต่าง ๆ (Ross, 1985:111) ความช่วยเหลือนี้ผู้เรียนสามารถเรียกใช้ได้ตลอดเวลา (on-line help) ตัวอย่างของความช่วยเหลือก็เช่น ความช่วยเหลือเกี่ยวกับโครงสร้างประโยค (sentence structure) ศัพท์ การขยายนาม บุพบท รวมไปถึงการขอดูคำตอบที่ถูกต้อง

การเลือกเครื่องมือ

เครื่องมือที่จะใช้สร้างโปรแกรมช่วยการเขียนได้แก่โปรแกรมคอมพิวเตอร์ ซึ่งอาจเป็นโปรแกรมที่สร้างขึ้นเพื่อใช้สำหรับการสร้างบทเรียนโดยเฉพาะที่เรียกว่า *โปรแกรมช่วยสร้างบทเรียน* (authoring program) หรืออาจเป็นโปรแกรมภาษาคอมพิวเตอร์ (computer language)

โปรแกรมช่วยสร้างบทเรียน มีให้เลือกทั้งโปรแกรมสำเร็จรูปที่ผู้สอนเพียงพิมพ์ข้อมูลเข้าไปในโปรแกรมแม่ (ซึ่งได้แก่โปรแกรมช่วยสร้างบทเรียนนั่นเอง) ก็จะได้แบบฝึกหัดทันที แต่โปรแกรมชนิดนี้แก้ไขเปลี่ยนแปลงรูปแบบไม่ได้ (template) ดังนั้น แบบฝึกหัดที่สร้างขึ้นจากโปรแกรมชนิดนี้จึงมีลักษณะเหมือนกับแบบฝึกหัดตัวอย่างในโปรแกรมแม่ทุกประการ ตัวอย่างของโปรแกรมชนิดนี้ก็เช่น โปรแกรม Eclipse และ Storyboard โปรแกรมช่วยสร้างบทเรียนแบบที่สองยากกว่าแบบแรก แต่สามารถใช้งานได้กว้างขวางกว่าแบบแรก เป็นโปรแกรมกึ่งสำเร็จรูป

ในการใช้โปรแกรมชนิดนี้ ผู้สอนต้องออกแบบหน้าจอภาพ (screen) และกำหนดขั้นตอนการทำงานของโปรแกรมและการตรวจคำตอบด้วยตนเอง อย่างไรก็ตาม แม้ผู้สอนจะต้องออกแบบโปรแกรมด้วยตนเองก็จริงแต่เป็นการสร้างโปรแกรมของท่านภายใต้สภาพแวดล้อม (environment) ที่โปรแกรมแม่ได้จัดเตรียมไว้ให้เป็นอย่างดี และมีเครื่องมืออำนวยความสะดวกต่าง ๆ ให้เลือกใช้จำนวนมาก เช่น เมื่อต้องการเสนอข้อความทางจอภาพ ท่านเพียงเลือกเครื่องมือสำหรับพิมพ์ข้อความ (text) แล้วพิมพ์ข้อความที่ต้องการได้ทันที การใช้โปรแกรมชนิดหลังนี้ผู้สอนจึงไม่ต้องมีความรู้เรื่องการเขียนโปรแกรม (programming) ตัวอย่างโปรแกรมช่วยสร้างบทเรียนชนิดนี้ก็เช่นโปรแกรม Authorware หรือ Tool Book

เครื่องมือชนิดที่สอง ได้แก่ ภาษาคอมพิวเตอร์ เช่น ภาษาเบสิก (BASIC) ภาษาปาสกาล (Pascal) และภาษาซี (C) ภาษาคอมพิวเตอร์เหล่านี้ก็เก็บอยู่ในรูปโปรแกรมเช่นกัน โปรแกรมภาษาคอมพิวเตอร์มีให้เลือกหลายโปรแกรม เช่น QuickBASIC Turbo Pascal C^{++} และ Turbo C^{++} ในการใช้โปรแกรมภาษาคอมพิวเตอร์เขียนบทเรียน ผู้สอนต้องมีความรู้ความสามารถในการเขียนโปรแกรม และการเขียนคำสั่งต่าง ๆ (coding) เพราะไม่ว่าจะให้โปรแกรมทำสิ่งใด ท่านต้องเขียนคำสั่งเพื่อให้โปรแกรมทำสิ่งนั้น

ความแตกต่างระหว่างการสร้างโปรแกรมช่วยการเขียนจากโปรแกรมช่วยสร้างบทเรียน และจากโปรแกรมภาษาคอมพิวเตอร์

โปรแกรมช่วยสร้างบทเรียน (authoring program)	**โปรแกรมภาษาคอมพิวเตอร์ (computer language)**
1. ผู้ใช้ไม่ต้องมีความรู้ด้านการเขียนโปรแกรม	1. ผู้ใช้ต้องมีความรู้ด้านการเขียนโปรแกรม
2. โปรแกรมได้สร้างวัตถุ (object) หรือเครื่องมือต่าง ๆ ให้ผู้เรียนใช้ เช่น กรอบข้อความ การเรียกภาพ และการเรียกเสียง	2. ผู้ใช้ต้องสร้างสิ่งต่าง ๆ (เช่น function หรือ sub-routine) ขึ้นใช้เองเป็นส่วนใหญ่
3. ผู้ใช้สามารถกดเมาส์เพื่อเลือกและใช้เครื่องมือต่าง ๆ ที่โปรแกรมจัดไว้ให้ (drag and drop)	3. ผู้ใช้ต้องเขียนคำสั่งหลายคำสั่งจึงจะได้สิ่งที่ต้องการ
4. เป็นโปรแกรมขนาดใหญ่ อาจไม่สะดวกต่อการใช้และการเก็บรักษา	4. ขนาดของโปรแกรมขึ้นอยู่กับความต้องการของผู้เขียนและความสามารถของภาษา คอมพิวเตอร์แต่ละภาษา แต่โปรแกรมที่สร้างขึ้นจะมีขนาดไม่ใหญ่เกินไป
5. ในปัจจุบันโปรแกรมช่วยสร้างบทเรียนส่วนใหญ่เป็นโปรแกรมในระบบ Windows จึงใช้กับระบบ Windows เท่านั้น และบางครั้งโปรแกรมที่จะใช้ได้ต้องสอดคล้องกับรุ่น (version) ของระบบปฏิบัติการของ Windows	5. ส่วนใหญ่แล้วโปรแกรมภาษาคอมพิวเตอร์มีให้เลือกทั้งที่ใช้กับระบบปฏิบัติการ DOS และที่ใช้กับระบบ Windows
6. การจัดการเกี่ยวกับตัวแปรต่าง ๆ ทำได้ยาก หรือในบางกรณีอาจต้องใช้ผู้เชี่ยวชาญเท่านั้น	6. การจัดการเกี่ยวกับตัวแปรทำได้ง่ายและสะดวก
7. การพัฒนาความรู้ด้านการเขียนโปรแกรมให้สูงขึ้นทำด้วยยาก เพราะผู้ใช้มักเคยชินกับการเลือกสิ่งต่าง ๆ ที่โปรแกรมจัดไว้ให้	7. ผู้ใช้อาจสามารถพัฒนาความรู้ด้านการเขียนโปรแกรมให้สูงขึ้น เพราะผู้เขียนโปรแกรมจะเข้าใจระบบและการทำงานของโปรแกรมเป็นอย่างดี

จากความแตกต่างที่แสดงไว้ในตาราง จะเห็นได้ว่าโปรแกรมช่วยสร้างบทเรียน (authoring program) มีข้อดีหลายประการ เมื่อใช้เป็นแล้วจะใช้ได้ง่ายกว่าการเขียนโปรแกรมจากโปรแกรมภาษาคอมพิวเตอร์โดยตรง แต่ในขณะเดียวกันการจัดการเกี่ยวกับตัวแปรทำได้ยาก อย่างไรก็ตาม หลักสำคัญก็คือท่านควรเลือกใช้เครื่องมือที่เหมาะสมกับความรู้ความสามารถของท่านเป็นสำคัญ

การออกแบบหน้าจอ

เมื่อทราบจุดมุ่งหมายและลักษณะทั่วไปของโปรแกรมที่จะสร้าง และมีเนื้อหาพร้อมแล้ว ขั้นต่อไปจะเป็นการออกแบบหน้าจอภาพ (screen design) หลักทั่วไปในการออกแบบหน้าจอภาพได้แก่การออกแบบให้หน้าจอเหมาะสมกับเรื่องที่จะสอน สะดวกต่อการใช้งาน สวยงาม น่าสนใจ และน่าเรียน

ในการออกแบบจอภาพของโปรแกรมช่วยการเขียน ท่านอาจแบ่งเนื้อที่ของจอภาพออกเป็นส่วน ๆ เพื่อใช้งานต่าง ๆ เช่น

1. เนื้อที่สำหรับแสดงข้อมูลสำหรับให้ผู้เรียนเขียน ข้อมูลนี้อาจเป็นข้อความ หรือภาพก็ได้
2. เนื้อที่สำหรับให้ผู้เรียนพิมพ์คำตอบ ซึ่งต้องจัดไว้ให้พอกับข้อความที่ผู้เรียนจะพิมพ์ อาจมีเพียงบรรทัดเดียว หรือหลายบรรทัด ตามความจำเป็น
   เนื้อที่นี้อาจอยู่ส่วนล่างของจอภาพ
3. เนื้อที่สำหรับแจ้งผลการตรวจ ซึ่งอาจอยู่ที่ส่วนล่างสุดของจอภาพ หรืออยู่ด้านขวาของจอภาพ
4. เนื้อที่สำหรับแสดงข้อความหรือสัญลักษณ์เพื่อแสดงความช่วยเหลือต่าง ๆ
5. เนื้อที่สำหรับแสดงข้อความหรือปุ่มสำหรับให้ผู้เรียนควบคุมการทำงานของโปรแกรม เช่น ปุ่ม CONTINUE MENU EXIT

นอกจากเนื้อที่สำคัญทั้งห้าส่วนนี้แล้ว ท่านอาจแบ่งจอภาพเพื่อใช้งานอื่นๆได้อีกตามต้องการ นอกจากนี้เพื่อให้ผู้เรียนเห็นความแตกต่างของแต่ละส่วนชัดเจนขึ้น ท่านอาจตีกรอบและให้สีแต่ละส่วนแตกต่างกันเพื่อให้สังเกตได้ง่าย

การตรวจคำตอบและการอธิบาย

สำหรับโปรแกรมช่วยการเขียน การตรวจคำตอบเป็นเรื่องสำคัญ เพราะแต่ละเรื่องอาจมีคำตอบได้หลายคำตอบ เช่น การบรรยายแต่ละส่วน (sector) ของกราฟวงกลมอาจมีคำตอบที่ถูกต้องได้ถึง 1,000 คำตอบ หรือมากกว่า และยังมีคำตอบผิดอีกจำนวนมากเช่นกัน ดังนั้น ท่านจึงต้องเลือกเครื่องมือที่จะใช้สร้างโปรแกรมให้เหมาะสม และต้องจัดระบบการตรวจให้มีประสิทธิภาพที่สุด

การตรวจคำตอบมีหลายแนวทาง แต่ละแนวทางมีอุปสรรค ปัญหา และความยากง่ายแตกต่างกัน ดังตัวอย่างต่อไปนี้

	วิธีตรวจ	ข้อดี	ปัญหา
1.	ผู้เขียนโปรแกรมพิมพ์ข้อความที่เป็นคำตอบถูกของแต่ละเงื่อนไข (case) ทุกคำ เพื่อให้โปรแกรมนำคำตอบหรือข้อความที่ผู้เรียนเขียนมาเปรียบเทียบ	ตรวจได้ถูกต้องแม่นยำ การตรวจวิธีนี้เหมาะกับโปรแกรมที่มีคำตอบไม่มาก	เปลืองเนื้อที่สำหรับเก็บคำตอบ ทำให้โปรแกรมใหญ่มากกว่าปกติ ซึ่งอาจมีผลทำให้ไม่สามารถกำหนดคำตอบได้ครบถ้วน
2.	แบ่งข้อความที่เป็นคำตอบถูกออกเป็นส่วน ๆ และเก็บคำตอบแต่ละส่วนไว้ในตัวแปรต่าง ๆ กัน ในการตั้งเงื่อนไขให้ใช้วิธีซ้อนเงื่อนไขเป็นชุดดังตัวอย่างต่อไปนี้ **(1) If ....** *(2) If ....* (3) If .... (3) else ... *(2) else ...* **(1) else ...**	ตรวจได้ถูกต้องแม่นยำ การตรวจวิธีนี้เหมาะกับโปรแกรมที่มีคำตอบจำนวนมาก	เปลืองเนื้อที่สำหรับเก็บคำตอบน้อยกว่าวิธีแรก แต่การตรวจด้วยวิธีนี้ยากกว่าการตรวจด้วยวิธีแรก
3.	เก็บคำหรือวลีไว้ในตัวแปรต่าง ๆ กัน เช่น Art1 = A Art2 = a Art3 = The Art4 = the Subject1 = boy Subject2 = young man VerbBe1 = is VerbBe2 = are VerbBe3 = was จากนั้น จึงกำหนดเงื่อนไขคำตอบถูกคล้ายกับวิธีที่สอง แต่เป็นการนำตัวแปรมารวมกันเป็นประโยค เช่น If (Art3 + Subject1 + VerbBe1 + ...) then ...	หากมีระบบการเก็บที่ดี วิธีนี้สามารถตรวจได้ถูกต้องแม่นยำ และไม่ต้องกำหนดเงื่อนไขสลับซับซ้อนเหมือนวิธีที่สอง การตรวจด้วยวิธีนี้เหมาะกับโปรแกรมที่มีคำตอบจำนวนมาก	การตรวจด้วยวิธีนี้ผิดพลาดได้ง่าย ท่านจึงต้องละเอียดรอบคอบมาก จึงจะตรวจได้ถูกต้อง

อย่างไรก็ตาม การจะใช้วิธีตรวจแบบใด ย่อมขึ้นอยู่กับเครื่องมือ และความรู้ความสามารถของท่านด้วยเช่นกัน นอกจากนี้ เครื่องมือหรือโปรแกรมแม่บางโปรแกรมอาจไม่เอื้อให้ท่านตั้งเงื่อนไข (case) จำนวนมาก เช่น 1,000 เงื่อนไข ดังนั้น ท่านจึงต้องศึกษาโปรแกรมแม่ที่จะใช้สร้างโปรแกรมของท่านให้ละเอียดก่อนเสมอ

การสร้างโปรแกรมต้นแบบ

โปรแกรมต้นแบบ (prototype) เป็นโปรแกรมที่ประกอบด้วยโครงสร้างและขั้นตอนหลักของโปรแกรมเท่านั้น จะยังไม่มีรายละเอียด เพราะรายละเอียดต่าง ๆ นั้นสามารถเพิ่มเติมได้ภายหลัง โปรแกรมต้นแบบสร้างขึ้นเพื่อศึกษาลักษณะและการทำงานของโปรแกรม และเป็นแบบ (model) ที่จะนำมาปรับปรุงเพื่อให้สามารถทำงานที่ท่านต้องการได้อย่างครบถ้วนต่อไป โปรแกรมต้นแบบอาจประกอบด้วยหน้าจอภาพที่แบ่งออกเป็นส่วน ๆ เช่น มีส่วนที่ใช้แสดงคำ ข้อความ หรือภาพที่กำหนดให้ และมีส่วนที่ให้ผู้เรียนป้อนคำตอบ มีตัวอย่างเงื่อนไขในการตรวจคำตอบทั้งคำตอบถูกและคำตอบผิด

ในการสร้างโปรแกรมต้นแบบ หากใช้โปรแกรมภาษาคอมพิวเตอร์ ท่านจะต้องเขียนคำสั่งต่าง ๆ ให้โปรแกรมทำงานหลัก ๆ ตามที่ได้ออกแบบไว้ หากใช้โปรแกรมช่วยสร้างบทเรียน (authoring program) เช่น โปรแกรม Authorware ท่านก็สามารถเริ่มต้นสร้างโปรแกรมที่ต้องการได้โดยไม่ต้องเขียนคำสั่ง แต่จะเป็นการนำไอคอน (icon) มาวางบนเส้น flow line และกำหนดรายละเอียดต่าง ๆ ภายในไอคอนแต่ละไอคอน

หลักที่สำคัญในการสร้างโปรแกรมต้นแบบมีดังนี้

1. พิจารณาว่าโปรแกรมที่จะสร้างขึ้นนั้นมีงาน (task) ที่สำคัญและจำเป็นใดบ้าง
2. ทดลองเขียนโปรแกรมให้ทำงานตามที่ได้จากข้อ 1
3. ทดลองใช้ (run) โปรแกรมต้นแบบ เพื่อตรวจสอบการทำงานของโปรแกรม
4. ปรับปรุงโปรแกรมต้นแบบให้สมบูรณ์ยิ่งขึ้น

เพื่อช่วยให้ท่านทราบวิธีสร้างโปรแกรมต้นแบบได้ชัดเจนขึ้น ขอให้ท่านศึกษาตัวอย่างต่อไปนี้ ซึ่งจะเป็นโปรแกรมช่วยการเขียนที่เขียนขึ้นจากโปรแกรมช่วยสร้างบทเรียนชื่อ Authorware และโปรแกรมที่เขียนขึ้นจากภาษาเบสิกและภาษาซี

การบันทึกและแสดงผลข้อมูล

ในการเขียนโปรแกรมโดยใช้ภาษาคอมพิวเตอร์ (computer language) นั้น เมื่อต้องการให้ผู้เรียนหรือผู้ใช้โปรแกรมพิมพ์คำตอบ หรือเลือกคำตอบด้วยการกด (click) เมาส์ที่คำ หรือข้อความที่ต้องการ ผู้เขียนโปรแกรมต้องเขียนคำสั่งสำหรับการป้อนข้อมูล (INPUT) คำสั่งดังกล่าวนี้แตกต่างกันไปตามภาษาคอมพิวเตอร์แต่ละภาษา

นอกจากการบันทึกคำตอบของผู้เรียนแล้วท่านอาจบันทึกข้อมูลอื่น โดยเฉพาะข้อผิดพลาดต่าง ๆ ของผู้เรียนเพื่อแจ้งให้ผู้เรียนทราบ และเพื่อที่ผู้สอนจะสามารถนำข้อมูลนั้นไปพิจารณาปรับปรุงการเรียนการสอน รวมทั้งการให้คำปรึกษาแก่ผู้เรียนเป็นรายบุคคล

ข้อผิดพลาดที่สำคัญด้านการเขียนมีหลายประการ Kroll (1991 : 145) แบ่งข้อผิดพลาดเป็นหมวดหมู่ดังนี้

1. Sentence structure (e.g. subject formation, verb missing, sentence fragment, run-on sentence, parallel structure, word order, etc.)

2. Verb-centered errors (e.g. noun-pronoun agreement, ambiguous/unlocatable referent, voice shift, etc.)

3. Word level choice (e.g. lexical/phrase choice, word form, singular-plural, preposition, etc.)

อย่างไรก็ตาม ท่านอาจแบ่งข้อผิดพลาดออกเป็นสองหมวดหมู่ดังนี้

1. Major errors (e.g. *subject-verb agreement, tense, voice*)
2. Minor errors (e.g. *article, preposition, punctuation*)

หลักทั่วไปในการบันทึกข้อมูลต่าง ๆมีดังนี้

1. กำหนดตัวแปร (variable) ที่จะใช้เก็บข้อมูล ตามกฎเกณฑ์ของภาษาคอมพิวเตอร์แต่ละภาษา
2. เปิดไฟล์ชนิดไฟล์ข้อมูล (text file/ASCII file)
3. เขียนคำสั่งบันทึกข้อมูลที่ต้องการลงในไฟล์ที่เปิดไว้แล้วในข้อ 2 เช่นในภาษาเบสิกอาจใช้คำสั่ง WRITE ตามด้วยหมายเลขไฟล์ และชื่อตัวแปรที่เก็บข้อมูล

วิธีการบันทึกข้อมูลต่าง ๆ ทำได้โดยการเขียนคำสั่งให้คอมพิวเตอร์บันทึกความผิดพลาดที่เกิดขึ้นว่าเป็นความผิดพลาดเรื่องใดไว้ในแต่ละเงื่อนไข (case) ขอให้ศึกษาตัวอย่างการบันทึกข้อมูลที่เขียนจากภาษาเบสิกต่อไปนี้

```
2
LOCATE 8, 1 : LINE INPUT "", Ans1$
SELECT UCASE$(Ans1$)
CASE "FOSSIL FUELS CONTAMINATE THE ENVIRONMENT."
      LOCATE 22, 28 : PRINT "VERY GOOD."
      score1 = score1 + 4
      OPEN "Scores1. wrt" FOR OUTPUT AS #1
            WRITE #1, score1
         WRITE #1, Ans1$
         CLOSE #1
         k$ = INPUT$(1)

CASE "...."
   CASE "...."
   CASE "FOSSIL FUELS CONTAMINATES THE ENVIRONMENT."
         LOCATE 22, 28 : PRINT "Check the subject-verb agreement."
         SVAgree1 = SVAgree1 + 1
      OPEN "SV1. wrt" FOR OUTPUT AS #2
```

⇨ ให้ผู้เรียนพิมพ์คำตอบ
⇨ ***เริ่มต้นการตรวจคำตอบ***
⇨ คำตอบถูกแรก
⇨ คำชม
⇨ ให้คะแนนข้อละ 4
⇨ เปิดไฟล์เพื่อบันทึกข้อมูล
⇨ ***บันทึกคะแนนและคำตอบถูกลงในไฟล์ชื่อ Scores1.wrt***
⇨ ปิดไฟล์
⇨ รอจนกว่าจะมีการกดแป้นใด ๆ (Press any key to continue)
⇨ คำตอบถูกอื่น ๆ ซึ่งเขียนเหมือนเงื่อนไขข้างบน
⇨ คำตอบผิดแรก
⇨ คำแนะนำ
⇨ รวมจำนวนครั้งที่ผิดเรื่องนี้
⇨ เปิดไฟล์บันทึกข้อผิดพลาด

WRITE #2, SVAgree1	⇨ ***บันทึกข้อผิดพลาดและจำนวน***
CLOSE #2	⇨ ปิดไฟล์
OPEN ""WrongAn1. wrt" FOR APPEND AS #3	⇨ เปิดไฟล์บันทึกคำตอบผิด
WRITE #3, Ans1$	⇨ ***บันทึกคำตอบผิด***
CLOSE #3	⇨ ปิดไฟล์
k$ = INPUT$(1)	⇨ รอจนกว่าจะมีการกดแป้นใด ๆ
GOTO 2	⇨ ย้อนกลับไปยังตำแหน่งที่ให้ผู้เรียนพิมพ์คำตอบ
CASE "...."	⇨ คำตอบผิดอื่น ๆ ที่ต้องการอธิบาย (เขียนเหมือนข้างบน)
CASE ELSE	⇨ คำตอบผิดอื่น ๆ นอกเหนือไปจากที่มีในเงื่อนไขข้างบน
.....	
.....	
END SELECT	⇨ ***จบการตรวจ***

จากตัวอย่างมีข้อที่ท่านควรทราบสามประการดังนี้

1. ในการเปิดไฟล์เพื่อบันทึกข้อมูล ท่านอาจบันทึกตัวแปรหลายตัวไว้ในไฟล์เดียว หรืออาจเปิดไฟล์เฉพาะสำหรับแต่ละตัวแปรก็ได้ จากตัวอย่างในเงื่อนไขคำตอบถูกจะมีการเก็บทั้งคะแนนและตัวคำตอบถูก (score1, Ans1$) ไว้ในไฟล์เดียวกัน แต่จากตัวอย่างคำตอบผิดนั้นจะเก็บข้อผิดพลาด (SVAgree1) และคำตอบผิด (Ans1$) ในไฟล์ต่างกัน เพราะจะสะดวกต่อการเรียกออกมาใช้ แต่วิธีหลังจะทำให้มีไฟล์จำนวนมาก

2. จากตัวอย่างมีไฟล์ข้อมูล (text file) ที่เปิดใช้สองชนิด คือ Output file และ Append File สำหรับไฟล์ชนิดแรกนั้นเมื่อมีการบันทึกข้อมูลลงในไฟล์แล้ว หากมีการบันทึกข้อมูลใหม่ซ้ำอีก ข้อมูลใหม่จะทับข้อมูลเดิมทั้งหมด ซึ่งจะทำให้ข้อมูลเดิมหายไป แต่ไฟล์ชนิดหลังเป็นไฟล์ที่ให้มีการบันทึกซ้ำ ๆ ได้ โดยข้อมูลใหม่จะอยู่ในบรรทัดถัดลงไปตามลำดับ ดังนั้น จึงเหมาะกับการบันทึกคำตอบผิด เพราะหากผู้เรียนตอบผิดหลายครั้ง คำตอบผิดแต่ละครั้งก็จะปรากฏให้เห็นทุกคำตอบ คำตอบผิดเหล่านี้เป็นข้อมูลที่มีประโยชน์มากสำหรับการศึกษาลักษณะความผิดพลาดในการเขียนของผู้เรียน ซึ่งจะเป็นประโยชน์ทั้งกับการให้คำชี้แนะแก่ผู้เรียนแต่ละคนโดยตรง การปรับปรุงการสอนของผู้สอนเอง และการปรับปรุงหลักสูตร และหากต้องการศึกษาข้อผิดพลาดของผู้เรียนให้ลึกซึ้งยิ่งขึ้น ท่านอาจสั่งให้คอมพิวเตอร์บันทึกเวลาที่ผู้เรียนตอบผิดแต่ละครั้ง ซึ่งทำได้ด้วยการใช้คำสั่งเกี่ยวกับเวลา เช่น ภาษาเบสิกใช้ TIME$ ดังตัวอย่างต่อไปนี้

```
OPEN ""WrongAn1. wrt" FOR APPEND AS #3
    WRITE #3, Ans1$, TIME$
CLOSE #3
```

คำสั่งให้บันทึกเวลา

เมื่อนำไฟล์ข้อมูลนี้ไปเปิดในโปรแกรมประมวลผลคำ (word processing program) ก็จะเห็นคำตอบผิดต่าง ๆ ของผู้เรียนดังตัวอย่างต่อไปนี้

คำตอบผิด	เวลา	สิ่งที่ผิด
Fossil fuels contaminating environment	10:20a	◆ ขาดเครื่องหมาย Full stop
Fossil fuels contaminating environment..	10:22a	◆ Verb form
Fossil fuels contaminates environment..	10:25a	◆ Subject-verb agreement
Fossil fuels contaminate environment..	10:30a	◆ Article

ประโยคต่อจากประโยคสุดท้ายจะเป็นคำตอบถูก

Fossil fuels contaminate the environment..

3. ในการเปิดไฟล์สำหรับบันทึกข้อมูล ท่านอาจเปิดไฟล์ไว้ภายในเงื่อนไขแต่ละเงื่อนไข ดังที่แสดงไว้ในตัวอย่าง หรือท่านอาจเปิดไฟล์ไว้นอกเงื่อนไข (case) ก็ได้ วิธีหลังนี้ดีกว่าวิธีแรก เพราะจะช่วยประหยัดเวลาในการเขียน และช่วยประหยัดเนื้อที่สำหรับเก็บโปรแกรมอีกด้วย ในภาษาเบสิกท่านอาจใช้โปรแกรมย่อยเฉพาะงาน (subroutine) แต่สำหรับภาษาซีท่านอาจใช้ function ที่สร้างขึ้นเพื่อการนี้โดยเฉพาะ

โปรแกรมตัวอย่าง

เพื่อช่วยให้ท่านเข้าใจวิธีเขียนโปรแกรมช่วยการเขียนดีขึ้น ขอให้ศึกษาโปรแกรมตัวอย่างที่เขียนจากโปรแกรม QuickBASIC ต่อไปนี้

```
1                                                    ⇨ เริ่มโปรแกรม
COLOR 7, 0                                           ⇨ กำหนดให้จอภาพเป็นสีปกติ
CLS                                                  ⇨ ลบจอภาพเดิม
COLOR 0, 3                                           ⇨ กำหนดสีคำสั่ง
LOCATE 3, 5 : PRINT "Write a sensible sentence from the words given."
                                                     ⇨ คำสั่งให้พิมพ์ประโยค
COLOR  15, 1                                         ⇨ กำหนดสีคำที่กำหนดให้
LOCATE 5, 5 : PRINT "fossil / fuels / contaminate / environment"
                                                     ⇨ คำที่กำหนดให้ใช้
2
COLOR 7, 0                                           ⇨ กำหนดให้จอภาพเป็นสีปกติ
FOR col = 1 TO 80                                    ⇨ ลบคำตอบเดิม
   LOCATE 8, col : PRINT" "                          ⇨ ลบคำแนะนำ
   LOCATE 22, col : PRINT" "                         ⇨ ลบการแจ้งผลและคำอธิบาย
NEXT col                                             ⇨ กำหนดคำตอบเป็นสีเหลือง
COLOR 14, 0                                          ⇨ ให้ผู้เรียนพิมพ์คำตอบ
LOCATE 8, 1 : LINE INPUT"", Ans1$                    ⇨ ชุดตรวจคำตอบ
SELECT UCASE$(Ans1$)                                 ⇨ คำตอบถูก
   CASE "FOSSIL FUELS CONTAMINATE THE ENVIRONMENT."  ⇨ ให้สีอักษรขาวบนพื้นม่วง
      COLOR 15, 5                                    ⇨ คำชม
                                                     ⇨ ให้คะแนนข้อละ 4
```

```
      LOCATE 22, 28 : PRINT "VERY GOOD."
      score1 = score 1 + 4
      OPEN "Scores1. wrt" FOR OUTPUT AS #1
         WRITE #1, score1
         WRITE #1, Ans1$
      CLOSE #1
      GOSUB Anykey
   CASE "FOSSIL FUELS CONTAMINATES THE ENVIRONMENT."
      COLOR 12, 0
      LOCATE 22, 28 : PRINT "Check the subject-verb agreement."
      SVAgree = SVAgree1 + 1
      OPEN "SV1. wrt" FOR OUTPUT AS #2
         WRITE #2, SVAgree1
      CLOSE #2
      GPSUB WrongAns
      GOSUB Anykey
      GOTO 2
   CASE "...."
   CASE "...."
    CASE ELSE
      COLOR 12, 0
      LOCATE 22, 28 : PRINT "Check your spelling and grammar."
      Other1 = Other1 + 1
      OPEN "Others1. wrt" FOR OUTPUT AS #4
         WRITE #4, Other1
      CLOSE #4
      GOSUB WrongAns
      GOSUB Anykey
      GOTO 2
 END SELECT
END
'-------------- End of program --------------
Anykey:
   COLOR 0, 3
   LOCATE 23, 28 : PRINT "Press an key to continue"
   k$ = input$(1)
   COLOR 7, 0
   LOCATE 23, 28 : PRINT "                    "
RETURN
WrongAns:
OPEN "WrongAnl.wrt" FOR APPEND AS#3
         WRITE#3, Ans 1$
      CLOSE#3
RETURN
```

- ⇨ เปิดไฟล์บันทึกข้อมูล
- ⇨ ***บันทึกคะแนน และคำตอบถูก***
- ⇨ ปิดไฟล์
- ⇨ รอการกดแป้นใด ๆ
- ⇨ คำตอบผิด
- ⇨ ให้สีอักษรแดงบนพื้นดำ
- ⇨ คำแนะนำ
- ⇨ นับจำนวนครั้งที่ตอบผิด
- ⇨ เปิดไฟล์บันทึกข้อมูล
- ⇨ ***บันทึกข้อผิดพลาด***
- ⇨ ปิดไฟล์
- ⇨ รอการกดแป้นใด ๆ
- ⇨ ให้ผู้เรียนตอบใหม่
- ⇨ คำตอบผิดอื่นที่ท่านต้องการอธิบาย
- ⇨ คำตอบผิดอื่น ๆ นอกเหนือไปจากที่มีในเงื่อนไขข้างบน
- ⇨ คำแนะนำ
- ⇨ จำนวนครั้งที่ตอบผิด
- ⇨ เปิดไฟล์บันทึกข้อมูล
- ⇨ ***บันทึกข้อผิดพลาด***
- ⇨ ปิดไฟล์
- ⇨ ***บันทึกประโยคคำผิด***
- ⇨ รอการกดแป้นใด ๆ
- ⇨ ให้ผู้เรียนตอบใหม่
- ⇨ จบการตรวจ
- ⇨ **จบโปรแกรมหลัก**

- ⇨ *โปรแกรมย่อยเฉพาะงาน*
- ⇨ สีอักษรดำบนพื้นฟ้าอมเขียว
- ⇨ ให้ผู้เรียนกดแป้นใด ๆ
- ⇨ รอการกดแป้นใด ๆ
- ⇨ กำหนดให้จอภาพเป็นสีปกติ
- ⇨ ลบข้อความ Press any key
- ⇨ จบโปรแกรมย่อยเฉพาะงานแรก
- ⇨ โปรแกรมย่อยเฉพาะงาน
- ⇨ เปิดไฟล์บันทึกข้อมูล
- ⇨ ***บันทึกประโยคผิด***
- ⇨ ปิดไฟล์
- ⇨ จบโปรแกรมย่อยเฉพาะงานที่สอง

ตัวอย่างนี้เป็นเพียงตัวอย่างสั้น ๆ ที่ยังไม่สามารถนำไปใช้ได้จริง เป็นเพียงตัวอย่างที่มุ่งให้เห็นแนวคิดและกระบวนการในการบันทึกข้อมูลเท่านั้น แต่ท่านสามารถขยายโปรแกรมตัวอย่างนี้ให้เป็นโปรแกรมที่ใช้งานได้จริงด้วยการเพิ่มเติมสิ่งต่าง ๆ เช่น การเพิ่มกรณีหรือเงื่อนไข (case) ในการตรวจ กล่าวคือต้องเพิ่มเงื่อนไขคำตอบถูกให้ครบ การเขียนคำสั่งต่าง ๆ ก็เขียนให้เหมือนกับตัวอย่างที่มีอยู่แล้ว และต้องเพิ่มคำตอบผิดที่ท่านต้องการอธิบายให้มากที่สุดเท่าที่ท่านสามารถคิดได้

การที่จะนำข้อมูลที่บันทึกไว้นี้มาใช้ในโปรแกรมได้นั้นจะต้องทำการเปิดไฟล์ที่บันทึกข้อมูลไว้ เช่น อาจเขียนคำสั่งดังนี้

```
OPEN "SV1. wrt" FOR INPUT AS #2
    READ #2, SVAgree1
CLOSE #2
```

จากนั้นจึงเรียกข้อมูลมาใช้เมื่อต้องการ เช่น LOCATE 15, 20 : PRINT SVAgree1

แต่หากท่านต้องการอ่านหรือแก้ไขข้อมูล ให้ท่านนำไฟล์ข้อมูลที่บันทึกไว้แล้วไปเปิดในโปรแกรมประมวลผลคำ

ในกรณีการใช้โปรแกรมช่วยสร้างบทเรียน การบันทึกข้อมูล และการนำข้อมูลออกมาใช้ทำได้ในลักษณะเดียวกันกับที่กล่าวไปแล้ว แต่ท่านต้องศึกษาว่าโปรแกรมที่ใช้นั้นเปิดโอกาสให้ท่านกำหนดตัวแปรต่าง ๆ ได้หรือไม่ หากได้ มีวิธีการอย่างไร

สรุป

โปรแกรมช่วยการเขียนมีประโยชน์ต่อทั้งผู้สอนและผู้เรียน การสร้างโปรแกรมชนิดนี้ไม่ยากเกินไป ท่านสามารถสร้างโปรแกรมช่วยการเขียนได้จากโปรแกรมภาษาคอมพิวเตอร์ หรือภาษาคอมพิวเตอร์โดยตรงแต่การจัดการเกี่ยวกับตัวแปร และการบันทึกข้อมูลอาจไม่สะดวกเท่ากับการใช้โปรแกรมภาษาคอมพิวเตอร์

การสร้างโปรแกรมควรเริ่มจากการกำหนดจุดมุ่งหมายของโปรแกรมอย่างชัดเจน และควรทดลองสร้างโปรแกรมต้นแบบเพื่อศึกษาความเป็นไปได้ในการสร้างโปรแกรมที่ต้องการ ในการสร้างโปรแกรมช่วยการเขียนท่านควรเลือกโปรแกรมแม่ (อาจเป็นโปรแกรมช่วยสร้างบทเรียน หรือโปรแกรมภาษาคอมพิวเตอร์) ที่ท่านสามารถใช้ได้ดีที่สุด และท่านควรมีที่ปรึกษาจึงจะประสบความสำเร็จได้ง่ายขึ้น

ในโปรแกรมช่วยการเขียนที่ท่านสร้างขึ้น ควรมีการบันทึกข้อมูลที่สำคัญต่าง ๆ โดยเฉพาะข้อผิดพลาดในเรื่องต่าง ๆ ของผู้เรียน คำตอบผิดทุกคำตอบ พร้อมกับเวลาที่เขียนประโยคผิดแต่ละประโยค และคะแนนจากคำตอบถูก

เอกสารอ้างอิง

Jones, Francis R., "Self-Instruction and Success: A Learner Profile Study" *Applied Linguistics* 19:3 (September, 1998). pp. 378-406.

Kroll, Babara, "What does Time Buy? ESL Student Performance on Home Versus Class Compositions" In Babara Kroll, *Second Language Writing: Research Insights for the Classroom*, Cambridge: Cambridge University Press, 1991. pp. 140-154.

Robat, Nico, On Developing a Sentence Analysis Program: A Progress Report in Theo Bongaerts, Pieter de Haan, Sylvia Lobbe and Herman Wekker (eds.), *Computer Applications in Language Learning*, Dordrecht, Holland, 1988. pp. 135-145.

Ross, Donald, "Realities of Computer Analysis of Composition" in James L. Collins and Elizabeth A. Sommers (eds.), *Writing On-Line: Using Computers in the Teaching of Writing*, Upper Monclair, New Jersey: Boynton/Cook Publishers, 1985. pp. 105-113.