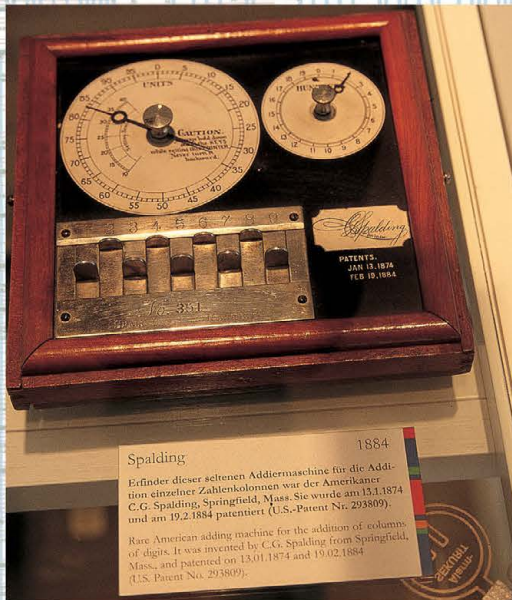


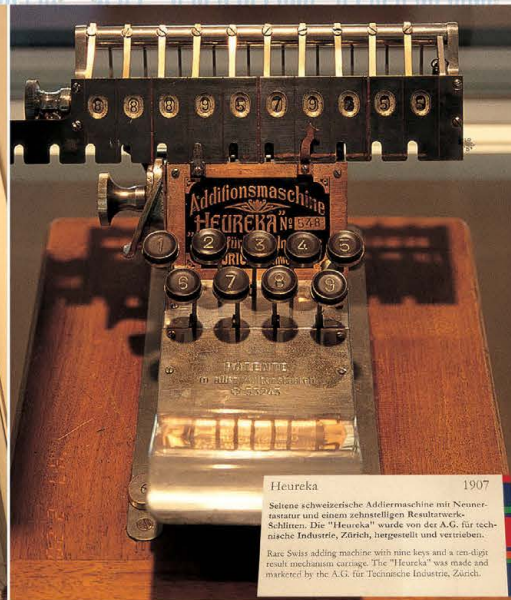


คอมพิวเตอร์

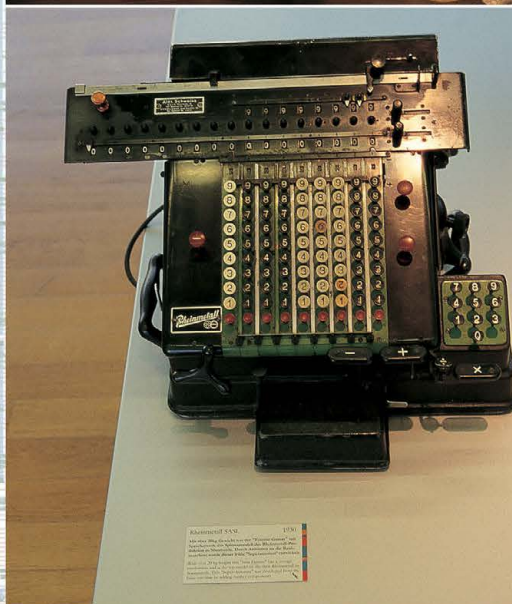
การเขียนโปรแกรมคอมพิวเตอร์ ภาษาซี



Spalding 1884
Erfinder dieser seltenen Addiermaschine für die Addition einzelner Zahlenkolonnen war der Amerikaner C.G. Spalding, Springfield, Mass. Sie wurde am 13.1.1874 und am 19.2.1884 patentiert (U.S.-Patent Nr. 293809).
Rare American adding machine for the addition of columns of digits. It was invented by C.G. Spalding from Springfield, Mass., and patented on 13.01.1874 and 19.02.1884 (U.S. Patent No. 293809).



Heureka 1907
Seltene schweizerische Addiermaschine mit Numeratortastatur und einem zehnstelligen Resultatwerk-Schlitzen. Die "Heureka" wurde von der A.G. für technische Industrie, Zürich, hergestellt und vertrieben.
Rare Swiss adding machine with nine keys and a ten-digit result mechanism carriage. The "Heureka" was made and marketed by the A.G. für Technische Industrie, Zürich.



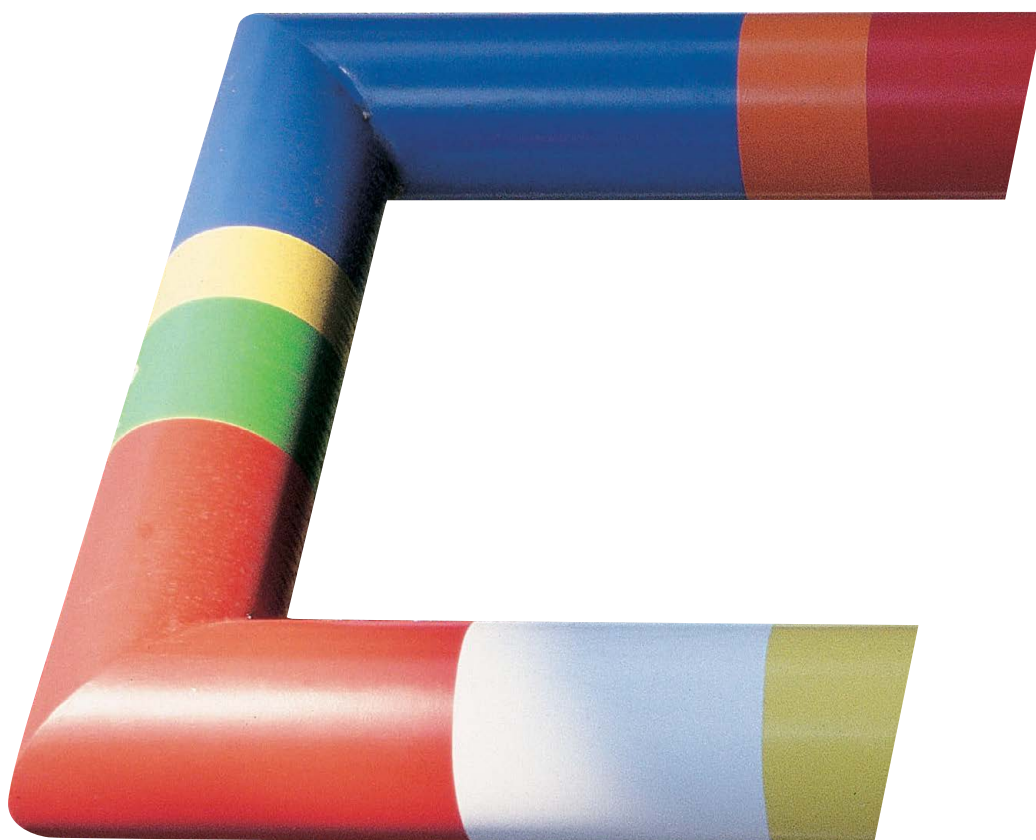
Babcock No. 100
The Babcock No. 100 is a rare and valuable addition machine. It was invented by the American Babcock & Wilcox Company, and was patented in 1909. It is a rare and valuable addition machine, and is a fine example of the work of the Babcock & Wilcox Company.



โครงการตำราวิทยาศาสตร์และคณิตศาสตร์มูลนิธิ สวทช.

คอมพิวเตอร์

(การเขียนโปรแกรมคอมพิวเตอร์ ภาษาซี)



โครงการตำราวิทยาศาสตร์และคณิตศาสตร์มูลนิธิ สอวน.



**มูลนิธิส่งเสริมโอลิมปิกวิชาการและพัฒนามาตรฐานวิทยาศาสตร์ศึกษา
ในพระอุปถัมภ์สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา
กรมหลวงนราธิวาสราชนครินทร์ (สอวน.)**

ความเป็นมา

ประเทศไทยส่งนักเรียนเข้าแข่งขันคณิตศาสตร์โอลิมปิกระหว่างประเทศครั้งแรกในปี พ.ศ. 2532 ที่ประเทศเยอรมนี โดยความร่วมมือระหว่างสมาคมวิทยาศาสตร์แห่งประเทศไทยในพระบรมราชูปถัมภ์ และสมาคมคณิตศาสตร์แห่งประเทศไทยในพระบรมราชูปถัมภ์ สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา (พระยศในขณะนั้น) ได้พระราชทานเงินส่วนพระองค์จำนวนหนึ่งเพื่อเป็นค่าใช้จ่าย

การคัดเลือกนักเรียนเพื่อไปแข่งขันคณิตศาสตร์โอลิมปิกระหว่างประเทศ ครั้งที่ 30 สมาคมคณิตศาสตร์แห่งประเทศไทยฯ ได้จัดคณะกรรมการมหาวิทยาลัยจำนวนหนึ่งมาช่วยเฟ้นกอบรวมเป็นเวลาประมาณสองเดือน เพื่อให้นักเรียนได้เรียนรู้เนื้อหาเพิ่มเติมครอบคลุมหลักสูตรที่จะใช้แข่งขัน ซึ่งอยู่ในระดับชั้นปีที่ 1-2 ของมหาวิทยาลัย จากผลสำเร็จในปีแรก ทำให้รัฐบาลเห็นความสำคัญจึงได้จัดสรรงบประมาณให้กับโครงการนี้ผ่านสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.) ตั้งแต่ปี พ.ศ. 2533 เป็นต้นมา สมทบกับเงินพระราชทานเป็นรายปีจากสมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ เพื่อสนับสนุนการส่งนักเรียนไทยไปแข่งขันโอลิมปิกวิชาการระหว่างประเทศ 5 สาขา จนถึงปัจจุบันรวม 14 ปี นักเรียนไทยได้ทำชื่อเสียงได้เหรียญทองรวม 17 เหรียญ เหรียญเงิน 53 เหรียญ เหรียญทองแดง 93 เหรียญ และเกียรติคุณประกาศ 35 ราย รวมทั้งสิ้น 198 รางวัล จากจำนวนนักเรียนที่ส่งไปแข่งขัน 276 คน (72%)

อย่างไรก็ตาม จากการที่ประเทศไทยดำเนินการจัดส่งนักเรียนเข้าร่วมการแข่งขันคณิตศาสตร์ วิทยาศาสตร์ โอลิมปิกระหว่างประเทศ ตั้งแต่ พ.ศ.2532 จนถึงปัจจุบัน ทำให้เห็นว่ามาตรฐานการศึกษาด้านคณิตศาสตร์และวิทยาศาสตร์ของไทยยังต่ำกว่ามาตรฐานสากล การเตรียมตัวของนักเรียนยังใช้เวลาน้อยเกินไป

เพื่อให้การส่งเสริมและสนับสนุน โครงการจัดส่งผู้แทนประเทศไทยไปแข่งขันโอลิมปิกวิชาการระหว่างประเทศมีประสิทธิภาพยิ่งขึ้นสมเด็จพระเจ้า พี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ จึงมีพระดำริให้จัดตั้งมูลนิธิ สอวน. และได้รับอนุมัติจากกระทรวงมหาดไทยเมื่อวันที่ 12 ตุลาคม 2542 โดยมีวัตถุประสงค์หลัก 2 ประการ

วัตถุประสงค์

1. ส่งเสริมให้นักเรียนในระดับมัธยมศึกษาตอนปลายทั่วประเทศที่มีความสามารถทางวิทยาศาสตร์และคณิตศาสตร์ มีโอกาสได้รับการพัฒนาศักยภาพทางด้านคณิตศาสตร์ คอมพิวเตอร์ เคมี ฟิสิกส์ และชีววิทยา ตามความถนัดทั้งด้านทฤษฎีและทักษะด้านปฏิบัติ ให้สามารถคิดวิเคราะห์และแก้ปัญหาที่ซับซ้อนได้ โดยจัดให้มีศูนย์อบรมทั่วประเทศเป็นการเพิ่มเวลาฝึกอบรม เพื่อช่วยให้นักเรียนมีความพร้อมที่จะเข้ารับการคัดเลือกไปแข่งขันโอลิมปิกวิชาการระหว่างประเทศให้ได้ผลดียิ่งขึ้น ตามพระดำริของสมเด็จพระเจ้าพี่นางเธอเจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ องค์ประธานมูลนิธิ สอวน.
2. เพื่อนำประสบการณ์ที่ได้จากการแข่งขันโอลิมปิกวิชาการระหว่างประเทศมาพัฒนามาตรฐานคณิตศาสตร์และวิทยาศาสตร์ศึกษาของไทยให้สูงขึ้นเทียบเท่าระดับสากล

โครงการตำราวิทยาศาสตร์และคณิตศาสตร์มูลนิธิ สอวน.

มูลนิธิส่งเสริมโอลิมปิกวิชาการและพัฒนามาตรฐานวิทยาศาสตร์ศึกษา ในพระอุปถัมภ์สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ (สอวน.) ได้ร่วมมือกับคณาจารย์มหาวิทยาลัยของรัฐ 20 แห่ง และกระทรวงศึกษาธิการ ดำเนินการจัดตั้งศูนย์ สอวน. ในภูมิภาค 12 ศูนย์ และในกรุงเทพฯ 1 ศูนย์ (5 โรงเรียน) เพื่อพัฒนาศักยภาพทางปัญญาของนักเรียนที่มีความพร้อมในด้านวิทยาศาสตร์ คณิตศาสตร์และคอมพิวเตอร์จากทั่วประเทศ เพื่อให้นักเรียนได้มีความรู้ความสามารถเทียบเท่ามาตรฐานสากล และพร้อมที่จะสอบคัดเลือกเป็นผู้แทนประเทศไทยไปร่วมการแข่งขันโอลิมปิกวิชาการระหว่างประเทศในสาขาวิชาต่างๆ ได้ และเพื่อขยายผลจากประสบการณ์ที่ได้เข้าแข่งขันโอลิมปิกวิชาการระหว่างประเทศมาพัฒนามาตรฐานวิทยาศาสตร์และคณิตศาสตร์ของไทยให้สูงขึ้นเทียบเท่าสากล

ในการอบรมนักเรียนของศูนย์ สอวน. มูลนิธิ สอวน. ได้พิจารณาเห็นว่าตำราที่มีความสมบูรณ์ของเนื้อหาตามหลักสูตรของ สอวน. จะช่วยพัฒนาศักยภาพของนักเรียนในศูนย์ สอวน. ให้สูงขึ้นได้ นอกจากนั้นตำราเรียนในวิชาวิทยาศาสตร์และคณิตศาสตร์ในระดับมัธยมศึกษาตอนปลายที่มีความสมบูรณ์ถูกต้องยังขาดแคลน นักเรียนและครูสามารถนำตำราที่พัฒนาขึ้นมานี้ไปใช้เป็นหนังสืออ้างอิงประกอบการเรียนการสอนได้อีกด้วย มูลนิธิ สอวน. จึงได้มี “โครงการจัดทำตำราส่งเสริมพัฒนาศักยภาพของนักเรียนด้านวิทยาศาสตร์ คณิตศาสตร์และคอมพิวเตอร์” ขึ้น เพื่อผลิตตำราคณิตศาสตร์ คอมพิวเตอร์ เคมี ชีววิทยาและฟิสิกส์ที่มีเนื้อหาหลักสูตรตามมาตรฐานสากล โดยมีวัตถุประสงค์

1. เพื่อเป็นที่ระลึกในมหามงคลสมัยที่สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ ทรงเจริญพระชนมายุ 80 พรรษา ซึ่งพระองค์ทรงมีพระมหากรุณาธิคุณต่อโครงการจัดส่งผู้แทนประเทศไทยไปแข่งขันโอลิมปิกวิชาการระหว่างประเทศและทรงสนับสนุนการดำเนินงานเพื่อพัฒนามาตรฐานการศึกษาวิทยาศาสตร์และคณิตศาสตร์ของไทยอย่างหาที่สุดมิได้

2. เพื่อผลิตหนังสือคณิตศาสตร์ คอมพิวเตอร์ เคมี ชีววิทยาและฟิสิกส์ระดับมัธยมศึกษาตอนปลายถึงระดับชั้นปีที่ 1 ของคณะวิทยาศาสตร์ในมหาวิทยาลัยให้มีคุณภาพเทียบเท่าสากล เรียบเรียงโดยคณาจารย์จากมหาวิทยาลัยของรัฐที่มีประสบการณ์สูงและที่มีส่วนร่วมในการฝึกอบรมนักเรียนในค่าย สอวน. ค่าย สสวท. และควบคุมนักเรียนไปแข่งขันโอลิมปิกวิชาการระหว่างประเทศ เนื้อหาในหนังสือเน้นกระบวนการคิดแบบวิทยาศาสตร์ เพื่อฝึกฝนให้นักเรียนสามารถคิดวิเคราะห์และแก้ปัญหาที่ซับซ้อนทั้งในเชิงทฤษฎีและประยุกต์ได้โดยจัดพิมพ์บนกระดาษอาร์ตอย่างดีและมีภาพสีประกอบคำบรรยาย

โครงการตำราวิทยาศาสตร์และคณิตศาสตร์ของมูลนิธิ สอวน. มีเป้าหมายในการผลิตตำราคณิตศาสตร์ รวม 5 เล่ม คอมพิวเตอร์ 3 เล่ม เคมี 3 เล่ม ชีววิทยา 5 เล่ม และฟิสิกส์ 3 เล่ม โดยหนังสือชุดแรกจะนำขึ้นทูลเกล้าถวายในวันที่ 6 มิถุนายน 2547 ซึ่งเป็นวันประชุมสามัญประจำปี พ.ศ. 2547 ของคณะกรรมการบริหาร และสมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ เสด็จเป็นองค์ประธานที่ประชุม ส่วนที่เหลือคาดว่าจะจัดพิมพ์ให้แล้วเสร็จภายในปี พ.ศ. 2547 นี้ นอกจากนั้นคณะกรรมการโครงการผลิตตำราวิทยาศาสตร์และคณิตศาสตร์มูลนิธิ สอวน. มีเป้าหมายจะผลิตตำราที่มีคุณภาพสูงนี้ในระดับมัธยมศึกษาต้นโดยคณาจารย์จากมหาวิทยาลัย เพื่อวางรากฐานวิทยาศาสตร์และคณิตศาสตร์พื้นฐานและเพื่อสร้างแรงบันดาลใจและความสนใจทางด้านวิทยาศาสตร์ คณิตศาสตร์แก่เยาวชนอีกด้วย หากได้รับความอนุเคราะห์ช่วยเหลือด้านงบประมาณจากภาครัฐและเอกชน

คณะจัดทำหนังสือคณิตศาสตร์ วิทยาศาสตร์หวังว่าโครงการตำราคณิตศาสตร์ วิทยาศาสตร์ของมูลนิธิ สอวน. จะมีส่วนร่วมในการปฏิรูปการเรียนรู้และยกระดับมาตรฐานวิทยาศาสตร์และคณิตศาสตร์ของไทยให้เทียบเท่ากับระดับสากลเพื่อสนองพระดำริของสมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์

*คำขอบคุณ : คณะกรรมการดำเนินงานโครงการตำราฯ ขอขอบคุณพิพิธภัณฑ ARTTHMEUM, Bonn ที่อนุญาต

ให้ประธานคณะกรรมการฯ เข้าถ่ายภาพในพิพิธภัณฑ เพื่อนำมาเผยแพร่

คณะกรรมการโครงการตำราวิทยาศาสตร์และคณิตศาสตร์มูลนิธิ สอวน.

คณะกรรมการที่ปรึกษา

1. ศาสตราจารย์ นายแพทย์ จรัส สุวรรณเวลา	รองประธานมูลนิธิ สอวน.
2. รองศาสตราจารย์ ดร.กำจัด มงคลกุล	กรรมการมูลนิธิฯ
3. ดร. คุณหญิงกษมา วรวรรณ ณ อยุธยา	กรรมการมูลนิธิฯ
4. รองศาสตราจารย์ ดร. คุณหญิงสุมณฑา พรหมบุญ	กรรมการมูลนิธิฯ
5. นายเชาว์ อรรถมานะ (อดีต)	กรรมการมูลนิธิฯ
6. นายสมนึก พิมลเสถียร	กรรมการมูลนิธิฯ
7. นายสุนทร อรุณานนท์ชัย	กรรมการมูลนิธิฯ
8. รองศาสตราจารย์บุญรักษา สุนทรธรรม	กรรมการมูลนิธิฯ

คณะกรรมการดำเนินงานโครงการตำราฯ

1. ศาสตราจารย์ศักดิ์ ศิริพันธุ์ (ราชบัณฑิต) เลขานุการมูลนิธิ สอวน.	ประธาน
2. รองศาสตราจารย์เย็นใจ สมวิเชียร	รองประธาน
3. รองศาสตราจารย์ ดร. ณรงค์ ปิ่นนัม	กรรมการ (วิชาคณิตศาสตร์)
4. รองศาสตราจารย์เย็น ภู่วรรณ	กรรมการ (วิชาคอมพิวเตอร์)
5. รองศาสตราจารย์ ดร. พินิติ รตะนานุกูล	กรรมการ (วิชาเคมี)
6. ผู้ช่วยศาสตราจารย์ ดร. วุทธิพันธุ์ ปรัชญพฤทธิ์	กรรมการ (วิชาฟิสิกส์)
7. ศาสตราจารย์อักษร ศรีเปล่ง	กรรมการ (วิชาชีพวิทยา ประเภทพืช)
8. รองศาสตราจารย์ ดร. อุษณีย์ ยศยิ่งยวด	กรรมการ (วิชาชีพวิทยา ประเภทสัตว์)

คณะผู้เขียนวิชาคอมพิวเตอร์

1. รองศาสตราจารย์เย็น ภู่วรรณ	ประธานคณะอนุกรรมการ
2. รองศาสตราจารย์ ดร. อนงค์นาฏ ศรีวิหก	อนุกรรมการ
3. ผู้ช่วยศาสตราจารย์อุมาพร ศิริธรรานนท์	อนุกรรมการ
4. ผู้ช่วยศาสตราจารย์กัลยาณี บรรจงจิตร	อนุกรรมการ
5. ผู้ช่วยศาสตราจารย์นงนุช สุขวารี	อนุกรรมการ
6. ผู้ช่วยศาสตราจารย์กรรณิกา คงสาคร	อนุกรรมการ
7. อาจารย์ศิริกร จันทร์นวล	อนุกรรมการ
8. อาจารย์พบสิทธิ์ กมลเวช	อนุกรรมการ
9. ดร. นวลวรรณ สุนทรภิชัย	อนุกรรมการ
10. ดร. สุขุมล กิตติสิน	อนุกรรมการ
11. อาจารย์พัชรี เลิศจิตรศิลป์	อนุกรรมการ
12. อาจารย์มารีสา มัยยะ	อนุกรรมการ

คำนำ

การเขียนโปรแกรมคอมพิวเตอร์ “ภาษาซี” เล่มนี้เหมาะสมสำหรับพื้นฐานให้ผู้สนใจการเขียนโปรแกรมแบบโครงสร้าง ทั้งผู้ที่เริ่มต้นศึกษาการเขียนโปรแกรมคอมพิวเตอร์และผู้ที่เคยศึกษาการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซีอื่นมาแล้ว โดยเนื้อหาประกอบด้วยทฤษฎีคำอธิบายและตัวอย่างโปรแกรม เริ่มต้นจากความรู้พื้นฐาน กฎเกณฑ์ การเขียนข้อความสั่งควบคุมการใช้แถวลำดับ ตัวชี้ รวมทั้งการประมวลผลเพิ่ม โดยได้พยายามลำดับเนื้อหาที่ละขั้นตอน เพื่อให้ผู้อ่านสามารถติดตาม เรียนรู้และเข้าใจง่าย คณะผู้เขียนหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะเป็นแนวทางในการศึกษาการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซีและเป็นประโยชน์ต่อผู้อ่าน

ที่ปรึกษา

รองศาสตราจารย์ยืน ภู่วรรณ

วศ.บ. (วิศวกรรมไฟฟ้าสื่อสาร) เกียรตินิยม, วศ. ม. (วิศวกรรมไฟฟ้า) จุฬาลงกรณ์มหาวิทยาลัย

M.Eng (Industrial Engineering and Management) at Asian Institute of Technology

ผู้เขียน

ผู้ช่วยศาสตราจารย์อุมาพร ศิริธรรณท์

M.S. (Mathematics and Computer Science) University of Illinois at Chicago

ผู้ช่วยศาสตราจารย์กัลยาณี บรรจงจิตร

พ.บ.ม. (สถิติประยุกต์) สถาบันบัณฑิตพัฒนบริหารศาสตร์

ดร. นวลวรรณ สุนทรภิษฐ์

วศ.ด. (วิศวกรรมคอมพิวเตอร์) จุฬาลงกรณ์มหาวิทยาลัย

ISBN 974-92235-2-7

สงวนลิขสิทธิ์

จัดพิมพ์โดย มูลนิธิ สอวน.

พิมพ์ครั้งที่ 2 พ.ศ. 2547 (2004)

ออกแบบปก หน้ารองปกและหน้านำบทที่ 1 โดย ศาสตราจารย์ศักดิ์ ศิริพันธุ์

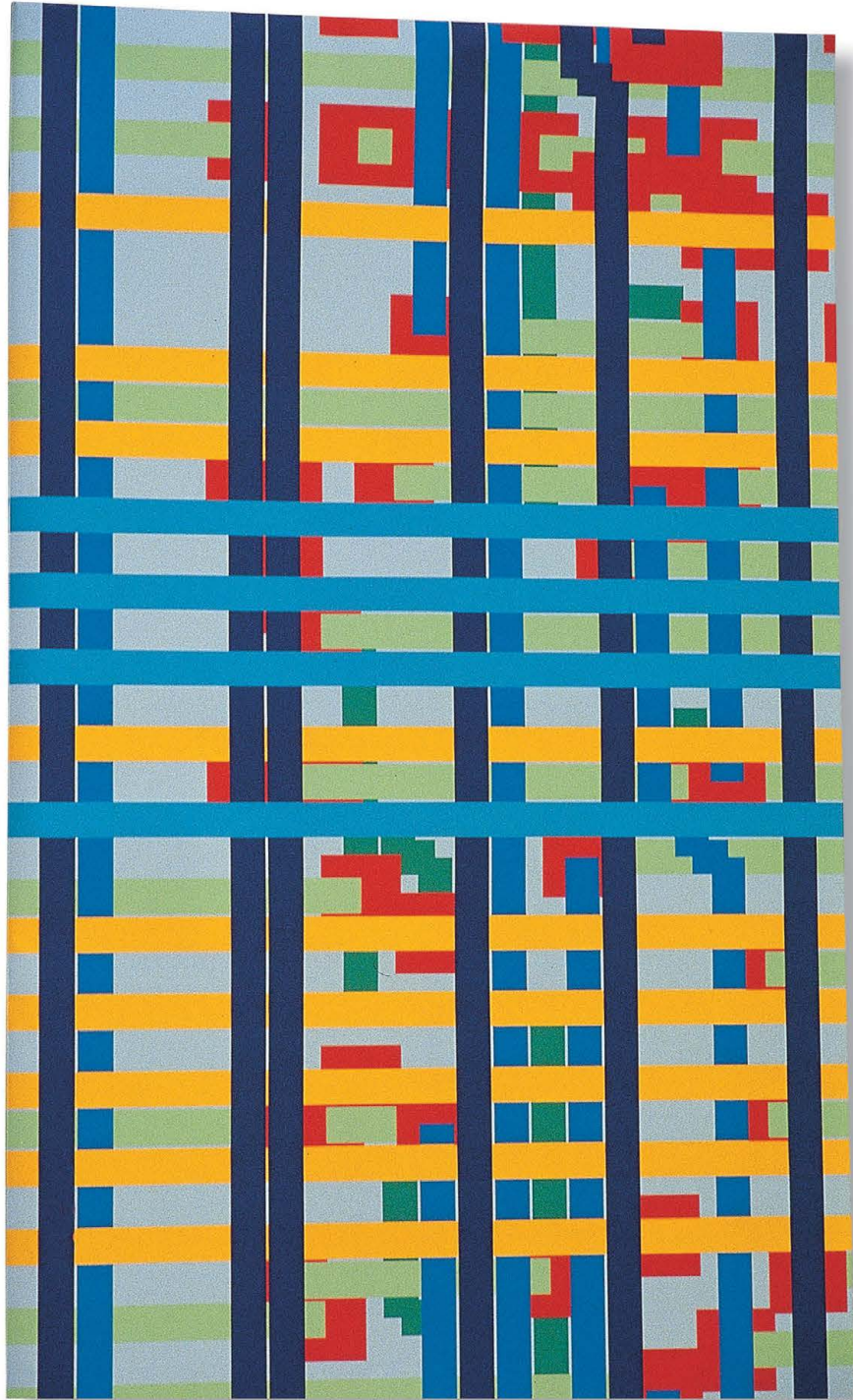
ศิลปกรรม: นายปรีชา ฉัตรเนตร

พิมพ์ที่ บริษัทด้านสุทธาการพิมพ์ จำกัด

แยกสี / เพลท เอ็นอาร์ फिल्म

บทที่ 1 ความรู้พื้นฐานโปรแกรมภาษาซี	1
1.1 โครงสร้างของโปรแกรมภาษาซี	1
1.2 ตัวแปร (variables)	5
1.3 ตัวคงที่ (constant)	12
1.4 การแสดงผลและการรับค่า	13
1.5 นิพจน์ (expressions)	19
1.6 ข้อความสั่งกำหนดค่า (assignment statement)	20
1.7 การคำนวณทางคณิตศาสตร์	21
1.8 ตัวดำเนินการเอกภาค (unary operator)	26
1.9 ตัวดำเนินการประกอบ (compound operator)	28
1.10 การแปลงชนิดข้อมูล (type cast)	30
1.11 การกำหนดค่าจากข้อมูลหลายชนิด (assignment with mixed types)	31
บทที่ 2 การจัดรูปแบบในการรับและแสดงผลข้อมูล	35
2.1 ฟังก์ชันรับข้อมูล	35
2.2 ฟังก์ชันในการแสดงผลข้อมูลออกทางจอภาพ	39
2.2.1 การแสดงเฉพาะข้อความ	39
2.2.2 การแสดงค่าของตัวแปรที่ได้จากการกระทำ	42
2.2.3 การแสดงทั้งข้อความและค่าที่เก็บในตัวแปร	50
บทที่ 3 การควบคุมโปรแกรม	57
3.1 ตัวดำเนินการสัมพันธ์ (relational operator)	57
3.2 ตัวดำเนินการตรรกะ (logical operator)	58
3.3 ลำดับในการดำเนินการ	59
3.4 ข้อความสั่งให้เลือกทำ	59
3.4.1 ข้อความสั่ง if	59
3.4.2 ข้อความสั่ง if-else	64
3.4.3 ข้อความสั่ง if ซ้อน	67
3.4.4 ข้อความสั่ง switch	74
3.4.5 ข้อความสั่ง break	76
3.5 การวนซ้ำ	80
3.5.1 ข้อความสั่ง for	80
3.5.2 ข้อความสั่ง while	85
3.5.3 ข้อความสั่ง do-while	86
3.6 การวนซ้ำซ้อน	88
บทที่ 4 ฟังก์ชัน	93
4.1 ฟังก์ชันที่เรียกใช้ฟังก์ชันอื่น	93
4.2 ฟังก์ชันเรียกซ้ำ (recursive function)	104
4.3 ฟังก์ชันจากคลัง (library function)	106
บทที่ 5 แกลลัดับ	113
5.1 การประกาศตัวแปรแกลลัดับ	113
5.2 การอ้างถึงสมาชิกแต่ละหน่วยของตัวแปรแกลลัดับ	114
5.3 ตัวแปรแกลลัดับเพื่อเก็บข้อมูลชนิดอักขระ	117
5.4 ตัวแปรแกลลัดับสองมิติ	120
5.5 ตัวแปรแกลลัดับสามมิติ	123
5.6 การกำหนดค่าเริ่มต้นให้กับแกลลัดับ	124

5.7	ฟังก์ชันเพื่อทำงานกับสายอักขระ	129
5.7.1	ฟังก์ชัน strcat()	129
5.7.2	ฟังก์ชัน strcmp()	130
5.7.3	ฟังก์ชัน strcpy()	131
5.7.4	ฟังก์ชัน strlen()	131
5.8	การส่งตัวแปรแถวลำดับไปยังฟังก์ชัน	132
5.8.1	การส่งหน่วยของตัวแปรแถวลำดับเป็นอาร์กิวเมนต์	132
5.8.2	การส่งตัวแปรแถวลำดับเป็นอาร์กิวเมนต์	134
บทที่ 6	ตัวแปรโครงสร้าง	143
6.1	การประกาศตัวแปรโครงสร้าง	143
6.2	การกำหนดค่าเริ่มต้นให้กับตัวแปรโครงสร้าง	149
6.3	ตัวแปรแถวลำดับที่มีสมาชิกคือตัวแปรโครงสร้าง	151
6.4	การส่งตัวแปรโครงสร้างไปยังฟังก์ชัน	153
6.5	ข้อความสั่ง typedef	158
บทที่ 7	ตัวชี้	163
7.1	การประกาศตัวแปรชนิดตัวชี้	163
7.2	ตัวดำเนินการสำหรับตัวชี้	164
7.2.1	ตัวดำเนินการเลขที่อยู่ (address operator)	164
7.2.2	ตัวดำเนินการอ้างอิง (dereferencing operator)	164
7.3	การจองเนื้อที่ในหน่วยความจำ	167
7.4	การคืนเนื้อที่ให้แก่หน่วยความจำ	167
7.5	การใช้ตัวแปรชนิดตัวชี้กับค่าคงที่	169
7.6	การเรียกใช้ฟังก์ชันที่มีการส่งตัวแปรแบบอ้างอิง	169
7.7	การใช้ตัวชี้กับตัวแปรชนิดแถวลำดับ	176
7.8	การใช้ตัวชี้กับสายอักขระ	180
บทที่ 8	แฟ้ม	185
8.1	การเปิดและปิดแฟ้ม	185
8.2	การอ่านและเขียนแฟ้ม	187
8.2.1	การอ่านและเขียนข้อมูลที่ละอักขระจากแฟ้ม	187
8.2.2	การอ่านและเขียนสายอักขระลงในแฟ้ม	193
8.2.3	การอ่านและเขียนข้อมูลแบบมีรูปแบบ	195
8.2.4	การอ่านและเขียนข้อมูลสำหรับแฟ้มชนิดไบนารี	197
8.3	การเข้าถึงข้อมูลในแฟ้มแบบสุ่ม (random access file)	200
8.4	ฟังก์ชันที่เกี่ยวข้องกับการประมวลผลแฟ้ม	203



บทที่ 1

ความรู้พื้นฐานโปรแกรมภาษาซี

โปรแกรมภาษาซีเป็นโปรแกรมที่มีความยืดหยุ่นและมีขีดความสามารถสูง โปรแกรมมีขนาดเล็กทำงานได้เร็ว ลักษณะของภาษาจะอยู่ในรูปแบบของฟังก์ชัน โปรแกรมหนึ่งอาจประกอบด้วยฟังก์ชันเดียวหรือหลายฟังก์ชัน เมื่อเขียนโปรแกรมใหม่ก็อาจนำเอาฟังก์ชันจากอีกโปรแกรมหนึ่งมาใช้ได้ ถ้าโปรแกรมทั้งสองมีการทำงานบางส่วนเหมือนกัน โปรแกรมที่ใช้ในคอมพิวเตอร์ระบบหนึ่ง ยังสามารถนำไปใช้กับคอมพิวเตอร์อีกระบบหนึ่งได้ โดยอาจมีการแก้ไขเพียงเล็กน้อย ด้วยเหตุนี้โปรแกรมภาษาซีจึงได้รับความนิยมจากนักเขียนโปรแกรมเป็นอย่างมาก

1.1 โครงสร้างของโปรแกรมภาษาซี

โปรแกรมภาษาต่างๆ จะมีรูปแบบหรือโครงสร้างเฉพาะที่แตกต่างกันไป สำหรับโปรแกรมภาษาซี มีโครงสร้างและลำดับการเขียนดังนี้

- ข้อความสั่งตัวประมวลผลก่อน (preprocessor statements)
- รหัสต้นฉบับ (source code) มีลำดับการเขียนดังนี้
 - ข้อความสั่งประกาศครอบคลุม (global declaration statements)
 - ต้นแบบฟังก์ชัน (function prototypes)
 - ฟังก์ชันหลัก (main function) มีเพียงฟังก์ชันเดียว
 - ฟังก์ชัน (functions) มีได้หลายฟังก์ชัน
 - ข้อความสั่งประกาศตัวแปรเฉพาะที่ (local declaration statements)
- หมายเหตุ (comment) สามารถแทรกไว้ที่ใดก็ได้ ภายในโปรแกรม

ข้อความสั่งตัวประมวลผลก่อน (preprocessor statements)

ข้อความสั่งตัวประมวลผลก่อนขึ้นต้นด้วยเครื่องหมาย # เช่น

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

หมายความว่า ให้ตัวประมวลผลก่อนไปอ่านข้อมูลจากแฟ้ม stdio.h ซึ่งเป็นแฟ้มที่มีอยู่ในคลัง

เมื่อโปรแกรมมีการใช้ข้อความสั่งอ่านและบันทึก จะต้องใช้ข้อมูลจากแฟ้ม stdio.h

ข้อความสั่งตัวประมวลผลก่อนจะต้องเขียนไว้ตอนต้นของโปรแกรม

รหัสต้นฉบับ (source code)

รหัสต้นฉบับ หมายถึง ตัวโปรแกรมที่ประกอบด้วยข้อความสั่งและตัวฟังก์ชันต่าง ๆ

ข้อความสั่งประกาศครอบคลุม (global declaration statements)

ข้อความสั่งประกาศครอบคลุมใช้ประกาศตัวแปรส่วนกลาง โดยที่ตัวแปรส่วนกลางนี้จะสามารถถูกเรียกใช้ จากทุกส่วนของโปรแกรม

ต้นแบบฟังก์ชัน (function prototypes)

ต้นแบบฟังก์ชันใช้ประกาศฟังก์ชัน เพื่อบอกให้ตัวแปลโปรแกรมทราบถึง ชนิดของค่าที่ส่งกลับ และ ชนิดของค่าต่างๆ ที่ส่งไปกระทำการในฟังก์ชัน

ฟังก์ชันหลัก (main function หรือ function main())

เมื่อสั่งให้กระทำการโปรแกรม ฟังก์ชันหลักจะเป็นจุดเริ่มต้นของการกระทำการ ภายในฟังก์ชันหลักจะประกอบด้วยข้อความสั่งและข้อความสั่งที่เรียกใช้ฟังก์ชัน เมื่อมีการทำงานตามข้อความสั่งและฟังก์ชันต่างๆ แล้ว จะมีการส่งค่าและกลับมาทำงานที่ฟังก์ชันหลักจนจบฟังก์ชัน

ฟังก์ชัน (functions)

ฟังก์ชัน หมายถึง กลุ่มของข้อความสั่งที่ทำงานใดงานหนึ่งโดยเป็นอิสระจากฟังก์ชันหลัก แต่อาจมีการรับส่งค่าระหว่างฟังก์ชันและฟังก์ชันหลัก การเขียนฟังก์ชัน ขึ้นต้นด้วย ชนิดข้อมูลที่ส่งกลับ ชื่อฟังก์ชัน วงเล็บ และตามด้วยเครื่องหมายปีกกา ภายในเครื่องหมายปีกกาประกอบด้วยข้อความสั่งภาษาซี

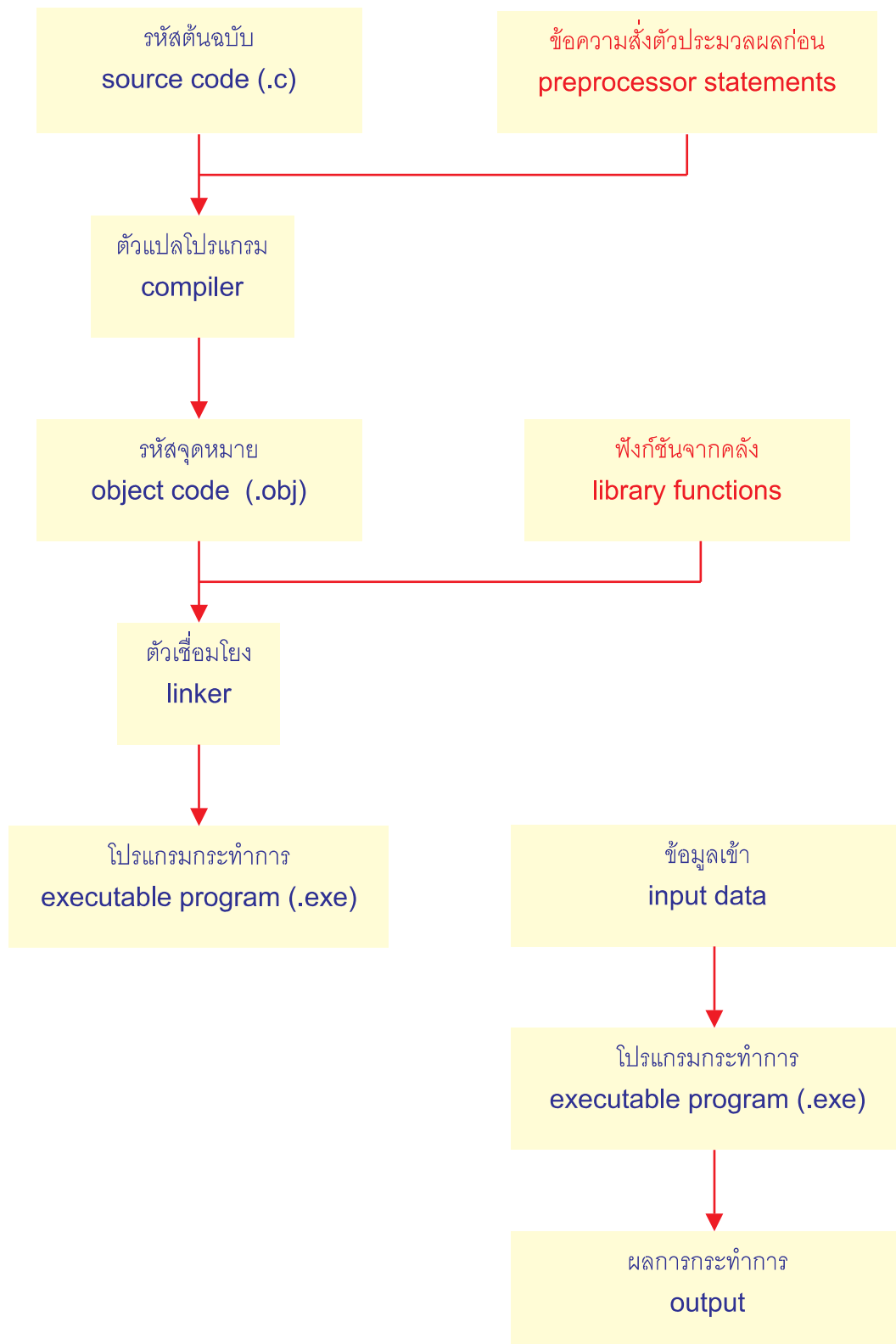
ข้อความสั่งประกาศตัวแปรเฉพาะที่ (local declaration statements)

ข้อความสั่งประกาศตัวแปรเฉพาะที่ ใช้ประกาศตัวแปรเฉพาะที่ โดยที่ตัวแปรเฉพาะที่จะสามารถถูกเรียกใช้เฉพาะภายในฟังก์ชันนั้น

การแปลและกระทำการโปรแกรม (program compilation and execution)

เมื่อได้เขียนและป้อนข้อความสั่งตัวประมวลผลก่อน (preprocessor statements) และรหัสต้นฉบับ (source code) ลงในโปรแกรมบรรณาธิการ (editor program) เสร็จแล้ว จะต้องเรียกตัวแปลโปรแกรม (compiler) มาเพื่อให้แปลภาษาซีให้เป็นภาษาเครื่อง (machine language) หากโปรแกรมเขียนได้ถูกต้องตรงตามกฎของภาษาซี ตัวแปลโปรแกรมแปลจะแปลโปรแกรมภาษาซีให้เป็นภาษาเครื่อง แล้วนำไปเก็บไว้ในแฟ้มชื่อเดียวกันแต่ส่วนขยายเป็น โอบีเจ (ชื่อแฟ้ม.obj) จากนั้นตัวเชื่อมโยง (linker) จะต้องนำฟังก์ชันจากคลัง (library function) ต่างๆที่โปรแกรมได้เรียกใช้ มารวมเข้ากับแฟ้มจุดโอบีเจ แล้วนำไปเก็บไว้ในแฟ้มชื่อเดิม แต่ส่วนขยายเป็น อีเอ็กซี (ชื่อแฟ้ม.exe) ซึ่งแฟ้มนี้จะเป็นแฟ้มที่พร้อมสำหรับกระทำการ (execute) เมื่อต้องการกระทำการโปรแกรมก็สามารถป้อนข้อมูลเข้า (input data) ให้กับโปรแกรม ซึ่งจะได้ผลการกระทำการ (output)

ฟังก์ชันจากคลัง เป็นฟังก์ชันสำเร็จรูปที่มีอยู่แล้วในตัวแปลโปรแกรม ซึ่งผู้เขียนโปรแกรมสามารถเรียกใช้ได้ด้วยการเขียนชื่อฟังก์ชันไว้ในโปรแกรม



รูปที่ 1.1 การแปลและกระทำการโปรแกรม

ตัวอย่าง 1.1

แสดงโครงสร้างของโปรแกรม

โปรแกรมประกอบด้วยฟังก์ชัน main() และฟังก์ชัน sum ()

ฟังก์ชัน main() ทำหน้าที่รับค่ามาเก็บไว้ในตัวแปร a และตัวแปร b แล้วส่งค่าของตัวแปรทั้งสองไปยังฟังก์ชัน sum() เพื่อคำนวณหาผลรวม เมื่อคำนวณผลรวมแล้ว จะส่งค่าของผลรวมกลับไปยังฟังก์ชัน main() จากนั้นฟังก์ชัน main() จะแสดงค่าของผลรวม

```
#include <stdio.h>                                // คำสั่งตัวประมวลผลก่อน
int a, b, c;                                       // คำสั่งประกาศครอบคลุม
int sum(int x, int y);                             // ต้นแบบฟังก์ชัน
void main()                                       // ฟังก์ชัน main()
{                                                 // เริ่มต้นฟังก์ชัน main()
    scanf("%d", &a);                             // คำสั่งในฟังก์ชัน main()
    scanf("%d", &b);                             // คำสั่งในฟังก์ชัน main()
    c = sum (a, b);                               // คำสั่งในฟังก์ชัน main()
    printf("\n%d + %d = %d", a, b, c);           // คำสั่งในฟังก์ชัน main()
}                                                 // จบฟังก์ชัน main()
int sum(int x, int y)                             // ฟังก์ชัน sum()
{                                                 // เริ่มต้นฟังก์ชัน sum()
    return (x + y);                               // คำสั่งในฟังก์ชัน sum()
}                                                 // จบฟังก์ชัน sum()
```

ข้อความสั่งหมายเหตุ (comment statement)

ข้อความสั่งหมายเหตุ คือ ข้อความที่เขียนไว้ภายในโปรแกรม เพื่อใช้อธิบายโปรแกรม โดยตัวแปลโปรแกรมจะไม่แปลข้อความสั่งหมายเหตุให้เป็นภาษาเครื่อง

การเขียนข้อความสั่งหมายเหตุในโปรแกรมทำได้ 2 แบบ ได้แก่

1. // หมายเหตุ ใช้เครื่องหมาย // หน้าข้อความหมายเหตุ ใช้ได้กับหมายเหตุที่มีขนาดความยาวไม่เกิน 1 บรรทัด
2. /* หมายเหตุ */ เขียนหมายเหตุไว้ระหว่าง /* และ */ ใช้ได้กับหมายเหตุที่มีขนาดความยาวตั้งแต่ 1 บรรทัดขึ้นไป

1.2 ตัวแปร (variables)

คอมพิวเตอร์มีส่วนประกอบที่สำคัญส่วนหนึ่งคือ หน่วยความจำ หน่วยความจำเปรียบได้กับสมองของมนุษย์ทำหน้าที่เก็บข้อมูลในขณะที่ยังประมวลผล ในการประมวลผลแต่ละครั้งมักต้องใช้ข้อมูลจำนวนมาก อาทิ ชื่อ นิสิต คะแนนสอบ วิชาคณิตศาสตร์ คะแนนสอบวิชาการเขียนโปรแกรมคอมพิวเตอร์ จำนวนหน่วยกิต คะแนนเฉลี่ย เป็นต้น ซึ่งจำเป็นจะต้องนำข้อมูลเหล่านี้ไปเก็บไว้ในหน่วยความจำ และเมื่อเก็บแล้ว จะต้องทราบตำแหน่งที่นำข้อมูลเข้าไปเก็บไว้ภายในของหน่วยความจำด้วย เพื่อให้สามารถนำข้อมูลเหล่านั้นกลับมาประมวลผลได้ ในภาษาคอมพิวเตอร์ระดับสูง เพื่อความสะดวกในการเขียนโปรแกรม การจดจำตำแหน่งที่ใช้ในการเก็บข้อมูล จะทำโดยการตั้งชื่อให้กับตำแหน่งของหน่วยความจำที่ใช้เก็บข้อมูลนั้น จากนั้นระบบปฏิบัติการจะช่วยจัดการในการหาตำแหน่งที่อยู่แท้จริงของข้อมูลต่อไป

ตัวแปร เป็น ชื่อของหน่วยความจำที่ตำแหน่งใดๆ เมื่อนำข้อมูลไปเก็บไว้ในหน่วยความจำตำแหน่งนั้น จะกล่าวว่า ตัวแปรนั้นมีค่าเท่ากับข้อมูลที่เก็บไว้ ตัวแปรสามารถเก็บค่าชนิดต่างๆ ตามที่ได้ประกาศไว้ในโปรแกรมเท่านั้น เช่น

- ตัวแปรที่ใช้เก็บข้อมูลชนิดจำนวนเต็ม
- ตัวแปรที่ใช้เก็บข้อมูลชนิดจำนวนจริง
- ตัวแปรที่ใช้เก็บข้อมูลชนิดอักขระ
- ตัวแปรที่ใช้เก็บข้อมูลชนิดสายอักขระ

ตัวแปรเหล่านี้จะไม่สามารถเก็บค่าชนิดอื่นนอกเหนือจากชนิดที่ประกาศไว้ และค่าที่เก็บไว้ในตัวแปรนี้สามารถเปลี่ยนค่าได้ตลอดเวลา ขึ้นกับข้อคำสั่งภายในโปรแกรม

กฎการตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรมีข้อกำหนดดังนี้

- ประกอบด้วย a ถึง z, 0 ถึง 9 และ _ เท่านั้น
- อักขระตัวแรกต้องเป็น a ถึง z และ _
- ห้ามใช้ชื่อเฉพาะ
- ตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก มีความหมายที่แตกต่างกัน
- ยาวสูงสุด 31 ตัวอักษร

หากตั้งชื่อตัวแปรไม่ตรงตามข้อกำหนด ก็จะทำให้ตัวแปลโปรแกรมไม่เข้าใจ ไม่ทราบว่า ชื่อนั้นคืออะไร ซึ่งจะทำให้เกิดข้อผิดพลาดขึ้นในระหว่างการแปลโปรแกรม

ชนิดข้อมูล (data types)

ข้อมูลที่ใช้ในโปรแกรมมีหลายชนิด ซึ่งนักเขียนโปรแกรมต้องเลือกใช้ตามความเหมาะสมกับการใช้งาน ข้อมูลมีขนาดที่ต่างกันไปตามชนิดข้อมูล นอกจากนี้แล้ว ชนิดข้อมูลยังอาจมีขนาดที่ต่างกันโดยขึ้นกับเครื่องคอมพิวเตอร์ และตัวแปลโปรแกรมที่ใช้ในการประมวลผล แต่โดยทั่วไปแล้วในไมโครคอมพิวเตอร์ ชนิดข้อมูลมีการใช้ในโปรแกรม และขนาดดังนี้

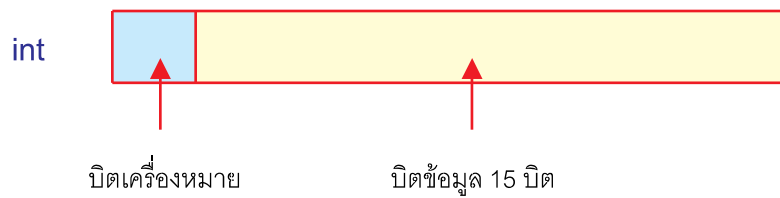
ชนิดข้อมูล	การใช้ในโปรแกรม	ขนาดข้อมูล (ไบต์)	ช่วงข้อมูล
character	char	1	-128 ถึง 127
integer	int	2	-32768 ถึง 32767
short integer	short	2	-32768 ถึง 32767
long integer	long	4	-2147483648 ถึง +2147483647
unsigned character	unsigned char	1	0 ถึง 255
unsigned integer	unsigned int	2	0 ถึง 65535
unsigned short integer	unsign short	2	0 ถึง 65535
unsigned long integer	unsign long	4	0 ถึง 4294967295
single-precision floating-point	float	4	1.2×10^{-38} ถึง 3.4×10^{38}
double-precision floating-point	double	8	2.2×10^{-308} ถึง 1.8×10^{308}

ตัวแปรชนิดตัวเลข (numeric variable types)

ตัวแปรชนิดตัวเลขแบ่งได้เป็น 2 กลุ่ม คือ

1. ตัวแปรจำนวนเต็ม (integer variables) หมายถึง ตัวแปรที่ใช้เก็บค่าที่เป็นจำนวนเต็ม ได้แก่ char, int, short, long, unsigned char, unsigned int, unsigned short และ unsigned long

ตัวอย่าง ลักษณะการจัดเก็บตัวแปรจำนวนเต็ม ใช้เนื้อที่ 16 บิต ประกอบด้วย บิตเครื่องหมาย (sign bit) 1 บิต และบิตข้อมูล (data bits) 15 บิต

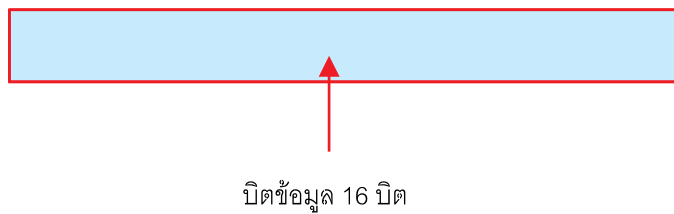


เช่น

- +15
- -2789
- 12
- -9

ตัวอย่าง ลักษณะการจัดเก็บตัวแปรจำนวนเต็มไม่มีเครื่องหมาย ใช้เนื้อที่เก็บข้อมูล 16 บิต โดยไม่มีการเก็บเครื่องหมาย

unsigned int



เช่น

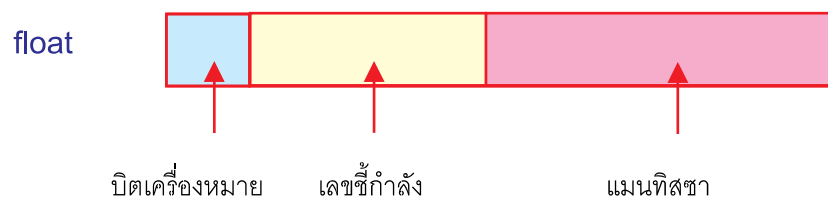
- 1578
- 2789
- 12
- 0

2. ตัวแปรจำนวนจริง (real variables) หรือ ตัวแปรจุดลอยตัว (floating-point variables) หมายถึง ตัวแปรที่ใช้เก็บค่าที่เป็นจำนวนจริง

ซึ่งเป็นตัวเลขที่เป็นเศษส่วน หรือมีจุดทศนิยม ได้แก่

- ชนิด float ใช้เนื้อที่เก็บข้อมูล 64 บิต

ตัวอย่าง ลักษณะการจัดเก็บตัวแปรจำนวนจริงชนิด float ใช้เนื้อที่เก็บข้อมูล 32 บิต ประกอบด้วย บิตเครื่องหมาย (sign bit) 1 บิต เลขชี้กำลัง (exponent) 8 บิต และแมนทิสซา (mantissa) 23 บิต



- ชนิด double ใช้เนื้อที่เก็บข้อมูล 64 บิต

ตัวแปรจำนวนจริง สามารถเขียนได้หลายแบบดังนี้

- แบบจุดตรึง (fixed point)

เช่น

- 35.058
- -400.75
- 49.
- 0.0
- +2.

- แบบจุดลอยตัว (floating point)

เช่น

- | | |
|--------------|------------------------------|
| • 1.234e6 | (1.234×10^6) |
| • 4.53e-8 | (4.53×10^{-8}) |
| • 1.8732e3 | (1.8732×10^3) |
| • 5.65421e-2 | (5.65421×10^{-2}) |

การประกาศตัวแปร

การประกาศตัวแปรทำได้โดย เขียนข้อความสั่งขึ้นต้นด้วยชนิดข้อมูล ตามด้วยชื่อตัวแปร และจบข้อความสั่งประกาศตัวแปรด้วยเครื่องหมายอัฒภาค (;) ดังนี้

ชนิดข้อมูล ชื่อตัวแปร;

ถ้าต้องการประกาศตัวแปรชนิดเดียวกันหลายตัว ต้องคั่นระหว่างตัวแปรด้วยเครื่องหมายจุลภาค (,) และจบข้อความสั่งประกาศตัวแปรด้วยเครื่องหมายอัฒภาค (;) ดังนี้

ชนิดข้อมูล ชื่อตัวแปร1, ชื่อตัวแปร2, ... ;

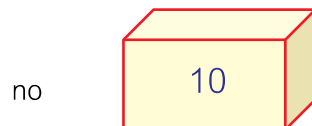
เช่น

```
int count;           // ประกาศตัวแปรชื่อ count ใช้เก็บข้อมูลชนิด integer
int m, n;            // ประกาศตัวแปรชื่อ m และ n ใช้เก็บข้อมูลชนิด integer
int no = 10;         /* ประกาศตัวแปรที่ใช้เก็บข้อมูลชนิด integer ชื่อ no และเก็บค่า 10
                     ไว้ในตัวแปรดังกล่าว */
long number;         // ประกาศตัวแปรชื่อ number ใช้เก็บข้อมูลชนิด long integer
float percent, total; // ประกาศตัวแปรชื่อ percent และ total ใช้สำหรับเก็บข้อมูลชนิด float
```

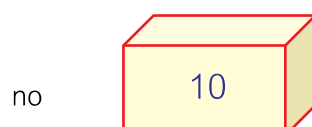
การกำหนดค่าให้ตัวแปรชนิดตัวเลข

การกำหนดค่าให้ตัวแปร ทำได้โดยการประกาศตัวแปรและกำหนดค่าให้กับตัวแปรไว้ในคำสั่งเดียวกัน หรือ อาจทำได้อีกวิธีหนึ่งคือ ประกาศตัวแปรก่อน จากนั้นจึงกำหนดค่าให้กับตัวแปรในอีกข้อความสั่งหนึ่ง เช่น

```
int no = 10; /* ประกาศตัวแปรที่ใช้เก็บข้อมูลชนิด integer ชื่อ no และ
             เก็บค่า 10 ไว้ในตัวแปรดังกล่าว */
```

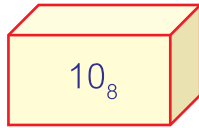


```
int no;           // ประกาศตัวแปรที่ใช้เก็บข้อมูลชนิด integer ชื่อ no
no = 10;          // เก็บค่า 10 ไว้ในตัวแปร no
```



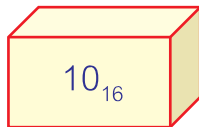
```
int num;           // ประกาศตัวแปรชื่อ num เพื่อใช้เก็บข้อมูลชนิด integer
num = 010;         /* กำหนดให้ค่า 108 เก็บไว้ใน num
                   ตัวเลขที่ขึ้นต้นด้วย 0 (ศูนย์) หมายถึง เลขฐาน 8 */
```

num



```
int num;           // ประกาศตัวแปรชื่อ num เพื่อใช้เก็บข้อมูลชนิด integer
num = 0x10;         /* 0x10 หมายถึง 1016
                   กำหนดให้ค่า 1016 เก็บไว้ใน num
                   ตัวเลขที่ขึ้นต้นด้วย 0x หรือ 0X (ศูนย์เอ็กซ์) หมายถึง เลขฐาน 16 */
```

num



ตัวแปรชนิดอักขระ (character variable types)

ตัวแปรชนิดอักขระ (char) ถูกจัดเก็บไว้ในหน่วยความจำในรูปแบบจำนวนเต็ม ขนาด 1 ไบต์ ดังนั้น ตัวแปรชนิดอักขระจึงสามารถใช้งานได้ทั้งแบบอักขระ และ จำนวนเต็ม

ตัวอย่าง 1.2

การใช้ตัวแปรชนิดอักขระ

```
#include <stdio.h>
void main()
{
    printf("%c %d\n", 'a', 'a');
}
```

ผลการกระทำการ

a 97

printf("%c %d\n", 'a', 'a'); เป็นฟังก์ชันที่ใช้สำหรับให้แสดงอักขระ a ในรูปแบบ char (%c) และการสั่งให้แสดงอักขระ a ในรูปแบบของ จำนวนเต็ม (%d) ส่วน \n หมายถึง ให้เลื่อนตัวชี้ตำแหน่ง (cursor) ไปที่ต้นบรรทัดใหม่

ตัวอย่าง 1.3

การใช้ตัวแปรชนิดอักขระ

```
#include <stdio.h>

void main()
{
    char c1, c2;                // ประกาศตัวแปรชื่อ c1 และ c2 ให้เป็นชนิดอักขระ
    c1 = 'a';                   // นำ a ไปเก็บไว้ใน c1
    c2 = 'A';                   // นำ A ไปเก็บไว้ใน c2
    printf("\n c1 = %c ", c1);  /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดง
                                ข้อความ c1 = a (ค่าของ c1 คือ a) */
    printf("\n c2 = %c ", c2) ; /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดง
                                ข้อความ c2 = A (ค่าของ c2 คือ A) */
}
```

ตัวอย่าง 1.4

การใช้ตัวแปรชนิดอักขระ

```
#include <stdio.h>

void main()
{
    char lc, uc;                // ประกาศตัวแปรชื่อ lc และ uc ให้เป็นชนิดอักขระ
    lc = 'a';                   // ให้นำ a ไปเก็บไว้ใน lc
    uc = lc - 32;               // นำ 32 ไปลบออกจากค่าที่เก็บใน lc แล้วเก็บผลไว้ใน uc
    printf("\nThe lower case character is %c ", lc);
                                /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดงข้อความ
                                The lower case character is a (ค่าของ lc คือ a) */
    printf("\n Its ASCII code is %d ", lc);
                                /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดงข้อความ
                                Its ASCII code is 97 (ค่าของ lc คือ a ซึ่งมีค่าเท่ากับ 97) */
    printf("\nThe upper case character is %c ", uc);
                                /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดงข้อความ
                                The uppercase character is A (ค่าของ lc คือ A) */
    printf("\n Its ASCII code is %d ", uc);
                                /* เลื่อนตัวชี้ตำแหน่งไปที่ต้นบรรทัดใหม่ แล้วแสดงข้อความ
                                Its ASCII code is 65 (ค่าของ uc คือ A ซึ่งมีค่าเท่ากับ 65) */
}
```

1.3 ตัวคงที่ (constant)

ตัวคงที่มีลักษณะคล้ายตัวแปร แตกต่างจากตัวแปรตรงที่ ค่าที่เก็บในตัวคงที่จะคงเดิมไม่มีการเปลี่ยนแปลงจนกระทั่งจบโปรแกรม แต่ค่าที่เก็บในตัวแปรสามารถเปลี่ยนแปลงได้ตลอดเวลา

การประกาศตัวคงที่

การประกาศตัวคงที่ทำได้ 2 วิธี ดังนี้

1. ใช้คำหลัก `const` ตามรูปแบบดังนี้

`const` ชนิดข้อมูล ชื่อตัวแปร = ค่าที่เก็บในตัวแปร;

ตัวอย่าง 1.5

การประกาศตัวคงที่โดยใช้คำหลัก `const`

```
const int count = 120;           // กำหนดให้ count เป็นตัวคงที่ชนิด int และเก็บค่า 120
const float vat = 0.07;         // กำหนดให้ vat เป็นตัวคงที่ชนิด float และเก็บค่า 0.07
const float pi = 3.14159;       // กำหนดให้ pi เป็นตัวคงที่ชนิด float และเก็บค่า 3.14159
```

2. ใช้ตัวประมวลผลก่อน ตามรูปแบบดังนี้

`#define` ชื่อตัวคงที่ ค่าคงที่

ตัวอย่าง 1.6

การประกาศตัวคงที่ โดยใช้ข้อความสั่งตัวประมวลผลก่อน

```
#define COUNT 120                // กำหนดให้ COUNT เป็นตัวคงที่ชนิด int และเก็บค่า 120
#define VAT 7                    // กำหนดให้ VAT เป็นตัวคงที่ชนิด float และเก็บค่า 0.07
#define PI 3.14159              // กำหนดให้ PI เป็นตัวคงที่ชนิด float และเก็บค่า 3.14159
```

1.4 การแสดงผลและการรับค่า

ฟังก์ชัน printf()

ฟังก์ชัน printf() เป็นฟังก์ชันจากคลัง ที่มาพร้อมกับตัวแปลโปรแกรมภาษาซี ใช้สำหรับการแสดงผล มีรูปแบบดังนี้

```
printf("สายอักขระควบคุม", ตัวแปร);
```

โดยที่ สายอักขระควบคุม ประกอบด้วย 3 ส่วน คือ

- ตัวอักขระที่จะแสดง
- รูปแบบการแสดงผล ขึ้นต้นด้วยเครื่องหมายเปอร์เซ็นต์ (%)
- ลำดับหลัก (escape sequence)

ตัวแปร คือ ชื่อของตัวแปรที่จะแสดงผล

รูปแบบการแสดงผล (format specifiers)

การกำหนดรูปแบบการแสดงผล

- ขึ้นต้นด้วยเครื่องหมายเปอร์เซ็นต์ (%)
- ตามด้วยอักขระ 1 ตัว หรือหลายตัว โดยที่อักขระนั้นมีความหมายดังนี้

อักขระ	ชนิดข้อมูล	รูปแบบการแสดงผล
c	char	อักขระเดียว
d	int	จำนวนเต็มฐานสิบ
o		จำนวนเต็มฐานแปด
x		จำนวนเต็มฐานสิบหก
f	float	จำนวนที่มีทศนิยม ในฐานสิบ

ตัวอย่าง 1.7

การใช้ฟังก์ชัน printf()

```
#include <stdio.h>
void main()
{
    int n;
    float score;
    n = 10;
    score = 100;
    printf("No : %d    Score : %f", n, score);
}
```

ผลการกระทำการ

No : 10 Score : 100.000000

แสดงค่าที่เก็บในตัวแปร n ในรูปแบบจำนวนเต็ม และแสดงค่าที่เก็บใน score ในรูปแบบจำนวนจริงที่มีทศนิยม

ตัวอย่าง 1.8

การใช้ฟังก์ชัน printf()

```
#include <stdio.h>
void main()
{
    int n;
    float score;
    n = 10;
    score = 100;
    printf("No : %d", n);
    printf("    Score : %f", score);
}
```

ผลการกระทำการ

No : 10 Score : 100.000000

ลำดับหลัก (escape sequence)

ในการแสดงผล บางสิ่งบางอย่างที่จะแสดง อาจไม่ใช่ตัวอักษร จึงไม่สามารถที่จะเขียนสิ่งที่จะแสดงไว้ในโปรแกรมได้ เช่น ต้องการเขียนโปรแกรมให้ส่งเสียง (แสดงผลเป็นเสียง) หรือต้องการให้เลื่อนขึ้นบรรทัดใหม่ก่อนแสดงข้อความ ดังนั้นในการเขียนโปรแกรมเพื่อแสดงผลสิ่งที่ไม่ใช่ตัวอักษรปกติ จะต้องใช้ลำดับหลัก เพื่อช่วยในการกำหนดอักขระพิเศษหรือสิ่งที่ไม่ใช่อักขระ ที่ต้องการให้โปรแกรมแสดง

ลำดับหลัก จะเขียนขึ้นต้นด้วยเครื่องหมายทับกลับหลัง (/) แล้วตามด้วยอักขระ ในการทำงาน เครื่องหมายทับกลับหลังจะบอกให้เครื่องคอมพิวเตอร์ทราบว่า ให้หลีกเลี่ยงการตีความอักขระที่ตามหลังมานี้ในลักษณะปกติ เพราะอักขระเหล่านี้จะมีความหมายพิเศษแตกต่างออกไป

ลำดับหลักที่ใช้ในการแสดงผลสิ่งที่ไม่ใช่อักขระปกติ ได้แก่

ลำดับหลัก	ผลการกระทำ
\n	ขึ้นบรรทัดใหม่ (new line)
\t	เลื่อนไปยังจุดตั้งระยะ (tab) ถัดไป
\a	เสียงกระดิ่ง (bell)
\b	ถอยไปหนึ่งทีว่าง (backspace)
\f	ขึ้นหน้าใหม่ (form feed)
\\	แสดงเครื่องหมายทับกลับหลัง (backslash)
\'	แสดงเครื่องหมายฝนทอง (single quote)
\"	แสดงเครื่องหมายพินทุ (double quote)

ตัวอย่าง 1.9

การใช้ลำดับหลัก \n

```
#include <stdio.h>                                // ข้อความสั่งตัวประมวลผลก่อน
void main()
{
    printf("Welcome to C!\n");                      // ฟังก์ชันจากคลัง ใช้ในการแสดงข้อความที่อยู่ใน " "
}
```

ผลการกระทำ

Welcome to C!

—

แสดงข้อความ Welcome to C! ณ ตำแหน่งที่ตัวชี้ตำแหน่งอยู่ แล้วตัวชี้ตำแหน่งเลื่อนไปที่ต้นบรรทัดใหม่

ตัวอย่าง 1.10

```

การใช้ลำดับหลัก \f และ \n
#include <stdio.h>                                // ข้อความสั่งตัวประมวลผลก่อน

void main()
{
    printf("\fWelcome ");    // ฟังก์ชันจากคลังใช้ในการแสดงข้อความที่อยู่ในพินทุ (" ")
    printf("to C!\n");       /* ฟังก์ชันจากคลังใช้ในการแสดง
                               ข้อความและอักขระที่อยู่ใน พินทุ (" ") */
}

ผลการกระทำการ
Welcome to C!

-
เลื่อนไปหน้าใหม่ แล้วแสดงข้อความ Welcome to C! ที่สคริปต์แรกของบรรทัดแรก แล้วตัวชี้ตำแหน่งเลื่อนไปที่
ต้นบรรทัดใหม่ ดังนี้

```

ตัวอย่าง 1.11

```

การใช้ลำดับหลัก \n
#include <stdio.h>                                // ข้อความสั่งตัวประมวลผลก่อน

void main()
{
    printf("Welcome\n to\n C!\n");               /* ฟังก์ชันจากคลังใช้ในการแสดง
                                                    ข้อความและอักขระที่อยู่ใน พินทุ (" ") */
}

ผลการกระทำการ
Welcome
to
C!

-
แสดงข้อความ Welcome ณ ตำแหน่งที่ตัวชี้จอภาพอยู่ แล้วตัวชี้ตำแหน่งเลื่อนไปที่ต้นบรรทัดใหม่
ที่ต้นบรรทัดใหม่ แสดงข้อความ to แล้วตัวชี้ตำแหน่งเลื่อนไปที่ต้นบรรทัดใหม่
ที่ต้นบรรทัดใหม่ แสดงข้อความ C! แล้วตัวชี้ตำแหน่งเลื่อนไปที่ต้นบรรทัดใหม่

```