ภาษาซี

ฉบับโปรแกรมเมอร์

(C Programming for Programmer)

- ✓ ภาษาซีเบื้องต้น
- ✓ คำสั่งควบคุม
- ✓ อาร์เรย์
- ✓ ฟังก์ชัน
- ✓ พอยเตอร์
- ✓ ตัวแปรโครงสร้างและยูเนียน
- ✓ แฟ้มข้อมูล
- ✓ สตริง
- ✓ การควบคุมคีย์บอร์ดและเมาส์
- ✓ กราฟิกในภาษาซี
- ✓ คู่มือเฮดเดอร์ในเทอร์โบซี

โดย เชาวลิต ขันคำ

คำนำ

หนังสือเล่มนี้ มีเนื้อหาเกี่ยวกับการใช้งานภาษาซี โดยนำเสนอ ฟังก์ชัน หลักการใช้งานฟังก์ชัน ตัวแปรแบบอาร์เรย์ ตัวแปรพอยเตอร์ สตริง การติดต่อกับแฟ้มข้อมูล การควบคุมีย์บอร์ด เมาส์ และการใช้งานกราฟิก ตั้งแต่ระดับพื้นฐานถึงระดับกลาง

ผู้ที่ควรจะศึกษาจากหนังสือเล่มนี้เป็นอย่างยิ่งคือ นักเรียน นักศึกษา ที่เป็นผู้ศึกษาการเขียนโปรแกรมภาษาซี และ ครู อาจารย์ ที่ต้องการสอนการเขียนโปรแกรม โดยเฉพาะอย่างยิ่งความรู้ภาษาซีระดับกลาง ก่อนที่จะเข้าสู่การเขียนโปรแกรมระดับสูงในระดับมืออาชีพต่อไป เนื่องจากเนื้อหาในหนังสือจะเน้นตัวอย่าง การใช้ฟังก์ชันที่สามารถศึกษา และนำไปสู่การประยุกต์เพื่อได้โดยง่าย

ท้ายนี้ผู้เขียนหวังเป็นอย่างยิ่งว่า ผู้ที่ได้ศึกษาจากหนังสือเล่มนี้ จะมีทัศนคติที่ดีต่อการเขียนโปรแกรม จนพัฒนาตนเองให้เป็นโปรแกรมเมอร์ที่เก่ง เพื่อพัฒนาประเทศไปสู่ความเป็นผู้นำทางด้านการพัฒนาซอฟต์แวร์ในอนาคตได้

เชาวลิต ขันคำ

3 มิถุนายน 2549

สารบัญ

หน้า

บทที่ 1
ภาษาซีเบื้องต้น

โปรแกรมภาษาซี เป็นโปรแกรมภาษาชั้นสูงที่มีความใกล้เคียงกับภาษาคอมพิวเตอร์ระดับต่ำ เพราะติดต่อกับภาษาแอสเซมบลี และเข้าถึงระบบการทำงานของเครื่องคอมพิวเตอร์ได้ง่ายกว่าภาษาอื่น ๆ โปรแกรมภาษาซีนำไปใช้เขียนโปรแกรมประยุกต์สำหรับงานด้านต่าง ๆ นอกจากนั้น ยังเป็นต้นแบบ หรือภาษาพื้นฐาน ในการสร้างโปรแกรมภาษาอื่น ๆ อีกด้วย ในบทนี้จะนำเสนอความรู้พื้นฐาน ที่จะเป็นสำหรับการศึกษาในต่อ ๆ ไป

ความเป็นมาของภาษาซี

ใน “มาเรียนภาษา C กันเถอะ” (2007) กล่าวว่า “การพัฒนาภาษาซีครั้งแรก เพื่อใช้เป็นภาษาสำหรับพัฒนาระบบปฏิบัติการยูนิกซ์แทนภาษาแอสเซมบลี ซึ่งเป็นภาษาระดับต่ำที่สามารถกระทำในระบบฮาร์ดแวร์ได้ด้วยความรวดเร็ว แต่จุดอ่อนของภาษาแอซเซมบลีก็คือความยุ่งยากในการโปรแกรม ความเป็นเฉพาะตัว และความแตกต่างกันไปในแต่ละเครื่อง เดนนิส ริตชี จึงได้คิดค้นพัฒนาภาษาใหม่นี้ขึ้นมา โดยการรวบรวมเอาจุดเด่นของแต่ละภาษาระดับสูงผนวกเข้ากับภาษาระดับต่ำ เรียกชื่อว่าภาษาซี”

ในเรื่อง “การรับข้อมูลและแสดงผล” (2007) อธิบายว่า “ภาษาซี เป็นภาษาคอมพิวเตอร์ ที่ใช้เขียนโปรแกรมคอมพิวเตอร์ ที่มีประสิทธิภาพสูงมาก มีผู้ผลิตหลายบริษัทที่ผลิตโปรแกรม(compiler)ที่ใช้เปลี่ยนชุดคำสั่งที่เขียนในรูปแบบของภาษาซีไปเป็นภาษาเครื่องที่สั่งให้คอมพิวเตอร์ทำงานตามต้องการได้ เช่น ไมโครซอฟต์ อินไพรส์(เดิมชื่อบอร์แลนด์) เป็นต้น และแต่ละบริษัทอาจมีในหลายชื่อหลายรุ่น ซึ่งแต่ละรุ่นอาจมีความแตกต่างกันบ้าง แต่ส่วนใหญ่จะเป็นไปตามมาตรฐานของสถาบัน

ANSI (American Standard National Institute) ซึ่งได้กำหนดมาตรฐานของภาษาซีไว้ ภาษาซีที่เป็นตามมาตรฐานทุกประการเรียกว่า ANSI C”

“พื้นฐานภาษาซี” (2007) ได้อธิบายเกี่ยวกับประวัติของโปรแกรมภาษาซีไว้ว่า “พัฒนาขึ้นมาในปี ค.ศ. 1970 โดย Dennis Ritchie แห่ง Bell Laboratories และได้ถูกใช้งานแต่ในห้องปฏิบัติการของ Bell จนกระทั่งปี 1978 นั้น Brian Kerninghan กับ Dennis Ritchie ได้ออกหนังสือ กำหนดมาตรฐานของภาษาซี ข้อกำหนดนี้คนมักเรียกขานกันว่า K&R C หลังจากนั้นปี 1980 ภาษาซี ก็ได้รับความนิยมมากขึ้น มีการพัฒนา comiler ภาษาซีออกมามากมาย”

ดังนั้นจึงสรุปได้ว่า ประวัติความเป็นมา เริ่มต้นที่เดนนิส ริสชี่ (Dennis Ritchie) แห่งห้องทดลองเบลล์ (Bell Labs) ได้สร้างภาษาซีขึ้นเมื่อ ค.ศ. 1971-1972 ซึ่งเขาและเคน ทอมพ์สัน (Ken Thompson) ได้ช่วยกันออกแบบระบบปฏิบัติการยูนิกส์มาแล้ว ภาษานี้ได้พัฒนามาจากภาษาบี (B) ต้นแบบเองมาจากภาษาบีซีพีแอล (BCPL) ในปี ค.ศ. 1967 ซึ่ง BCPL คือ (Basic Combined Programming Language) ที่ออกแบบและติดตั้งใช้งานในระบบยูนิกส์ เวอร์ชันแรก ๆ รู้จักกันดีใน เคแอนด์อาร์ซี (K&R C) และต่อมาในปี ค.ศ. 1983 ได้พัฒนาให้เป็นมาตรฐานคือ ANSI (American National Standards Institute) C

เนื่องจากภาษาซีได้รับความนิยมอย่างรวดเร็ว เพราะสามารถจัดการฮาร์ดแวร์ได้ดี และพัฒนาโปรแกรมบนเครือข่ายคอมพิวเตอร์ได้ ซึ่งระยะแรก ๆ จะใช้งานในระบบปฏิบัติการยูนิกส์ ต่อมามีการขยายลงมาใช้งานในเครื่องใช้งานทั่ว ๆ ไป จากความสามารถที่มีมากและความนิยมใช้มากขึ้น จึงทำให้มีผู้ผลิตโปรแกรมภาษาซีขึ้นมาอย่างมากมายหลากหลายบริษัท ทั้งบริษัทไมโครซอฟต์ บริษัทบอร์แลน และในยูนิกส์หรือ ลีนุกซ์ ซึ่งจะมีคอมไพเลอร์ของภาษาซีติดมาด้วย

ขั้นตอนในการเขียนโปรแกรมด้วยภาษาซี

วิทยา เรืองพรวิสุทธิ์ (2537, หน้า 16) กล่าวว่า ขั้นตอนการเขียนโปรแกรมทำได้ดังนี้คือ ขั้นตอนการเขียนโปรแกรมด้วยอิดิเตอร์ ขั้นตอนการคอมไพล์โดยคอมไพเลอร์ และขั้นตอนการลิงค์โดยลิงเกอร์

นุกูล กระจาย (2540,หน้า 37-38) กล่าวว่าขั้นตอนการเขียนโปรแกรมมีดังนี้ สร้างซอร์สโค้ดเป็นไฟล์นามสกุล .CPP คอมไพล์ ลิงค์กับไฟล์นามสกุล .H เพื่อสร้างเป็นอ็อบเจก์โค้ดคือไฟล์นามสกุล .EXE หลังจากนั้นจะนำไฟล์นามสกุล .EXE ไปรันเพื่อให้ทำงานเป็นขั้นสุดท้าย

การเขียนโปรแกรมด้วยภาษาซี จะทำได้หลังจากติดตั้งโปรแกรมภาษาซีก่อน ซึ่งหากเป็นยูนิกส์หรือลีนุกส์ จะใช้งานได้ทันที แต่ถ้าเป็นบอร์แลนซี หรือไมโครซอฟต์ซี จำเป็นต้องติดตั้งโปรแกรมเพื่อใช้งานก่อน ขั้นตอนที่จะอธิบายต่อไปนี้ เป็นขั้นตอนที่เกิดภายหลังจากการติดตั้งโปรแกรมส่วนที่เป็นตัวดำเนินการและคอมไพล์โปรแกรมไว้แล้ว การใช้งานมีขั้นตอนดังนี้

1. เขียนรหัสต้นฉบับ

ผู้เขียนโปรแกรมภาษาซีจะต้องเขียนรหัสต้นฉบับ (Source Code) โดยใช้โปรแกรมอิดิเตอร์ (Editor) ซึ่งอาจจะเป็นอิดิเตอร์ ของภาษาซีเอง หรือใช้อิดิเตอร์ตัวอื่น ๆ โดยรหัสต้นฉบับดังกล่าวควรจะกำหนดนามสกุล .CPP หรือ .C ทั้งนี้จะเขียนรหัสต้นฉบับได้จะต้องวิเคราะห์และออกแบบโปรแกรมมาแล้ว รวมถึงเรียนรู้คำสั่งและรูปแบบของการเขียนโปรแกรมด้วยภาษาซีด้วย

2. คอมไพล์และลิงค์

คอมไพล์และลิงค์ (Link) โปรแกรม เพื่อแปลจากรหัสต้นฉบับให้เป็นภาษาที่เครื่องคอมพิวเตอร์สามารถนำไปใช้งานได้ เนื่องจากโปรแกรมภาษาซีจะใช้คอมไพเลอร์ (Compiler) ในการแปลรหัสต้นฉบับ จึงต้องมีการคอมไพล์เพื่อแปลจากรหัสต้นฉบับ ให้เป็นอ็อบเจกต์โค้ด (Object Code) โดยจะมีนามสกุล OBJ ในขั้นตอนนี้ หากมีการใช้งานผิดรูปแบบ จะเกิดข้อผิดพลาดให้แก้ไขรหัสโปรแกรมให้ถูกต้องก่อนการคอมไพล์โปรแกรม หลังจากนั้นจะทำการลิงค์ โปรแกรมให้ เป็นโปรแกรมที่สามารถรันได้

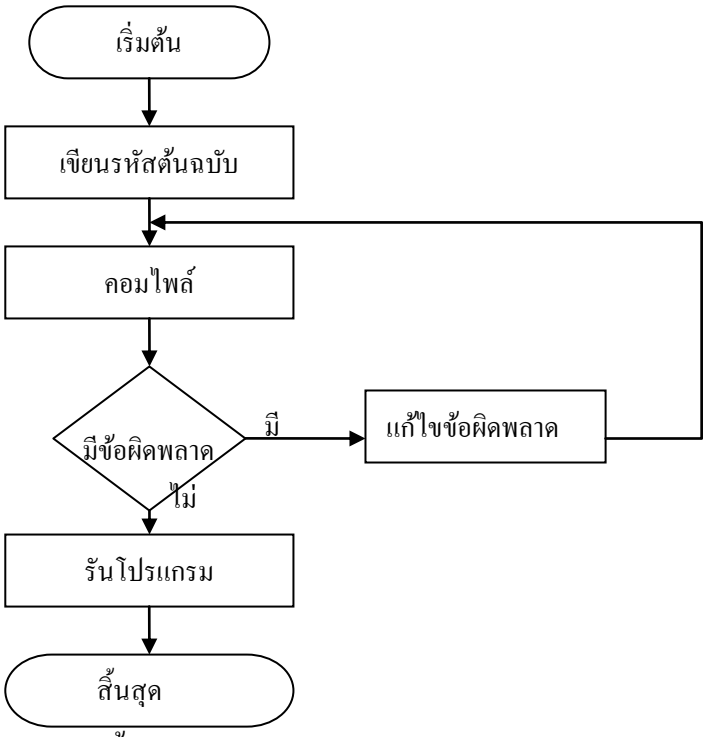
เรียกว่า เอ็กซีคิว้เทเบิลโค้ด (Executable Code) ซึ่งระบบวินโดว์จะมีนามสกุล .EXE และสามารถนำไปใช้งานได้ทันที หรือในระบบยูนิกส์หรือลีนุกซ์ก็จะเป็นแฟ้มที่สามารถเรียกใช้งานได้เช่นกัน

ขั้นตอนนี้มีความสำคัญยิ่งเพราะโปรแกรมจะตรวจสอบข้อผิดพลาดจากการเขียนโปรแกรมด้วยหากมีข้อผิดพลาดก็จะกลับไปแก้ไข ให้ถูกต้องและคอมไพล์ใหม่อีกครั้ง วิธีการจัดการกับข้อผิดพลาดเบื้องต้นแสดงในภาคผนวก ก

3. รันโปรแกรม

หลังจากคอมไพล์และลิงค์โปรแกรมแล้วไม่มีข้อผิดพลาด ขั้นตอนสุดท้ายคือการรันโปรแกรม โดยการเรียกใช้งานแฟ้มที่ลิงค์ให้แล้ว เช่น ในระบบวินโดว์จะเป็นแฟ้มนามสกุล .EXE

ขั้นตอนดังกล่าวสามารถแสดงได้ดังภาพที่ 3.1



ภาพที่ 1.1 แสดงขั้นตอนของการเขียนโปรแกรมด้วยภาษาซี

รูปแบบของโปรแกรมที่เขียนด้วยภาษาซี

เนื่องจากโปรแกรมภาษาซี เป็นโปรแกรมภาษาที่มีโครงสร้าง ดังนั้นจำเป็นอย่างมาก ที่ผู้เขียนโปรแกรมต้องศึกษารูปแบบโครงสร้าง ตลอดจนรูปแบบบังคับของภาษาซี ให้ชัดเจนและแม่นยำ สิ่งสำคัญของนั้นภาษาซีจะมองว่า คำสั่งทุกคำสั่งอยู่ในรูปแบบของฟังก์ชัน กล่าวคือ โปรแกรมย่อยหรืองานย่อยต่าง ๆ หรือคำสั่ง จะถูกเรียกว่า ฟังก์ชันทั้งหมด

1. โครงโปรแกรม

โครงโปรแกรม คือ ส่วนประกอบโดยรวมที่ทุก ๆ โปรแกรมต้องมี ธีรวัฒน์ ประกอบผล (2545, หน้า 9-10) กล่าวว่า โครงสร้างโปรแกรมประกอบด้วย ส่วนพรีโพรเซสเซอร์ไดเร็กทีฟ ส่วนประกาศ ส่วนฟังก์ชันหลัก ส่วนกำหนดฟังก์ชันขึ้นเอง และส่วนอธิบายโปรแกรม

“แนะนำภาษาซี” (2007) ได้อธิบายว่าโครงสร้างของโปรแกรมภาษาซีประกอบด้วยดังนี้

#include <library> หรือ header file

ส่วนประกาศ หรือ declaration part ส่วนนี้ใช้ประกาศตัวแปร ค่าคงที่ ชนิดของข้อมูล แบบ global

main() ฟังก์ชันหลักของโปรแกรม

```
{
        คำสั่งต่าง ๆ ค่าคงที่ ฯลฯ
}
```

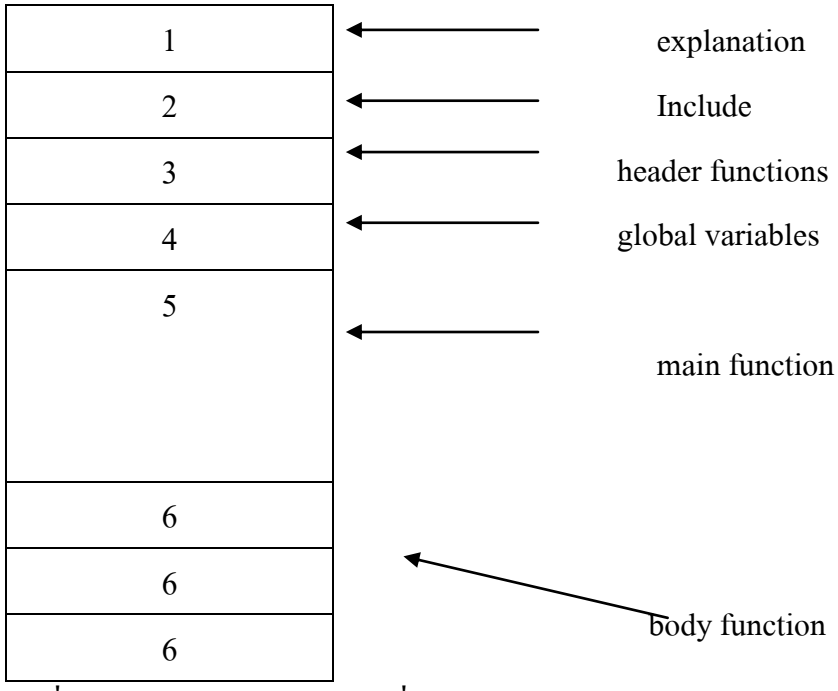
ฟังก์ชันย่อย (sub function)

```
{
        คำสั่งต่าง ๆ ค่าคงที่ ฯลฯ
}
```

ดังนั้นโครงโปรแกรมควรมีรูปแบบ ดังนี้

1. ส่วนคำอธิบาย (explanation) เกี่ยวกับโปรแกรม
2. ส่วนการอินคลู้ด (include) เพื่อเรียกใช้ฟังก์ชันจากไลบรารีหรือเรียกว่าเฮดเดอร์ไฟล์ (header files)
3. ส่วนการกำหนดส่วนหัวฟังก์ชัน (header functions)
4. ส่วนการประกาศตัวแปรสาธารณะ (public variables)
5. ฟังก์ชันของโปรแกรมหลัก (main function)
6. ส่วนรายละเอียดฟังก์ชัน (body functions)

สามารถอธิบายรายละเอียดต่าง ๆ ได้ดังภาพที่ 3.7



ภาพที่ 1.2 แสดงโครงโปรแกรมที่เขียนด้วยภาษาซี

อย่างไรก็ตามส่วนต่าง ๆ ดังกล่าวนั้นสามารถที่จะยืดหยุ่นได้บ้าง เช่นส่วนที่ 6 อาจจะนำไปเขียนไว้ในส่วนที่ 3 ได้ หากนำไปเขียนไว้ส่วนที่ 3 ก็ไม่จำเป็นต้องประกาศเฮดเดอร์ฟังก์ชันก็ได้ หรือตัวแปรสาธารณะอาจประกาศไว้ที่ส่วนที่อยู่นอกฟังก์ชัน ทั้งหมดก็สามารถทำได้เช่นกัน ตัวอย่างโปรแกรมตามส่วนต่าง ๆ ดังกล่าว แสดงได้ดังนี้

```
/*  an example program for input name and show name
    Wrote by mr.example
     Start date  25/03/2007
     Last update 24/04/2007
-------------------------------------------------*/
```

explanation

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

include

```
void Inputdata();
void Displaydata();
```

header functions

```
char name[20];
int score;
```

public variables

```
void main(){
    clrscr();
     Inputdata();
     Displaydata();
}
```

main function

```
void Inputdata(){
cprintf("Enter Name and Score");
scanf("%s%d",name,score);
}
void Displaydata(){
printf("This is Name:%s  Score :%d",name,score);
}
```

body function

2. รูปแบบการเขียน

แต่ละส่วนของโปรแกรมจะมีเครื่องหมายเฉพาะ เรียกกว่า รูปแบบการเขียนซึ่งอธิบายได้ดังนี้

รูปแบบและเครื่องหมายที่ใช้ในการเขียนโปรแกรม

{ ใช้สำหรับเริ่มการใช้งานกลุ่มคำสั่ง หรือฟังก์ชัน

} ใช้สำหรับจบหรือปิดกลุ่มคำสั่งหรือฟังก์ชัน

; ใช้สำหรับจบคำสั่ง ซึ่งในการเขียนโปรแกรมคำสั่งทุกคำสั่งจะต้องจบด้วยเครื่องหมายนี้เสมอ

( ) ใช้สำหรับวงเล็บครอบเพื่อผ่านค่าตัวแปรหรือค่าต่าง ๆ ไปยังฟังก์ชัน หรือครอบกลุ่มของตัวแปรในการคำนวณ

เรียกว่า พรีโพรเซสเซอร์ (preprocessor) ใช้สำหรับนำหน้าการ เรียกฟังก์ชันไลบรารีมาใช้งาน หรือกำหนดรูปแบบข้อมูลชนิด (tpyedef) เพื่อใช้ประมวลผลการดำเนินการคำสั่งอื่นในโปรแกรม

// ใช้สำหรับแสดงหมายเหตุเป็นบรรทัดกลุ่มคำสั่งที่อยู่ในบรรทัดนั้น โดยคอมไพเลอร์จะไม่นำไปคอมไพล์

/* */ ใช้สำหรับแสดงหมายเหตุเป็นกลุ่มคำสั่งขนาดยาว ๆ หรือมากกว่า 1 บรรทัด

3. รหัสบังคับการแสดงผล การรับข้อมูล และรหัสหลุดพ้น

รหัสบังคับการแสดงผลและรับข้อมูลจะช่วยในการจัดการเกี่ยวกับการติดต่อกับผู้ใช้ให้สามารถรู้จักสิ่งที่จะป้อนและสิ่งที่แสดง แสดงดังตารางที่ 3.1 และ 3.2 ตามลำดับ

ตารางที่ 1.1 แสดงรูปแบบของรหัสบังคับ การแสดงและการรับข้อมูล

รหัสบังคับ	ความหมาย
%c	แสดงหรือรับตัวอักขระ
%d	แสดงหรือรับเลขจำนวนเต็ม
%e	แสดงหรือรับเลขจำนวนจริงที่เป็นสัญลักษณ์ทางวิทยาศาสตร์
%f	แสดงหรือรับเลขจำนวนจริง
%g	แสดงหรือรับเลขเช่นเดียวกับ %c และ %f แต่สั้นกว่า
%o	แสดงหรือรับเลขฐานแปด (ไม่มีเครื่องหมาย)
%s	แสดงหรือรับค่าสตริง (string)
%u	แสดงหรือรับเลขจำนวนเต็มที่ไม่มีเครื่องหมาย
%p	แสดงค่าของพอยเตอร์ (pointer)
%x	แสดงหรือรับค่าเลขฐานสิบหก (ไม่มีเครื่องหมาย) ของ a-f
%X	แสดงรับเลขฐานสิบหก (ไม่มีเครื่องหมาย) A-F

การใช้งานรหัสบังคับนี้มักจะใช้ในฟังก์ชัน printf() และ scanf() ตัวอย่างที่ 3.1 แสดงการใช้งานรหัสบังคับดังกล่าว

ตัวอย่างที่ 1.1

จงเขียนโปรแกรมรับชื่อ และคะแนน หลังจากนั้นคำนวณ 80 เปอร์เซ็นต์ของคะแนนที่ป้อนแล้วแสดงผลทั้งหมดบนจอภาพ

```
#include <stdio.h>                    // นำ  header file  เข้ามารวมเพื่อเรียกใช้ฟังก์ชันมาตรฐาน
        #include <conio.h>            // นำ header file เข้ามารวมเพื่อเรียกใช้งานฟังก์ชันในการแสดงผล
        void main(void){               // เริ่มต้นโปรแกรมหลัก
            char name[80];             // กำหนดตัวแปร name เป็นชนิด อักขระ มีความยาว 80 อักขระ
            float score,percent;       // กำหนดตัวแปร score และ percent เป็นชนิดทศนิยม
             clrscr();                 // ฟังก์ชันที่ใช้ในการลบจอภาพ
            cprintf("Enter Your Name :");      //แสดงข้อความให้รับชื่อ
```

```
            scanf("%s",name);                    //ให้ผู้ใช้ป้อนข้อความชื่อของตนเอง
              cprintf("Enter  Score :");         //แสดงข้อความให้รับคะแนน
              scanf("%.2f",&score);              //ให้ผู้ใช้ป้อนคะแนนเป็นเลขทศนิยม 2 ตำแหน่ง
              percent = (score*100)/80;          // คิดค่า เปอร์เซ็นต์ จากคะแนนเต็ม 80 คะแนน
            printf("Name :%s   Score:%.2f   Percent :%.2f",name,score,percent);        / * คำสั่ง
แสดงผล ค่า เปอร์เซ็นต์ เป็นทศนิยม 2 ตำแหน่ง  */
              getch();                           // รอรับค่าการกดแป้นพิมพ์จากผู้ใช้
          }                                      // จบโปรแกรม
```

ผลของการรันโปรแกรมนี้ โปรแกรมจะให้ผู้ใช้ป้อนชื่อและคะแนนของตัวเอง โดยผู้ใช้จะต้องป้อนคะแนนไม่เกิน 80 คะแนน หลังจากนั้นโปรแกรมจะคำนวณคะแนนเป็นร้อยละ แล้วแสดงผลข้อมูลที่ป้อนและคำนวณค่าร้อยละของคะแนนที่ป้อนเข้าไปออกมาทางจอภาพ แล้วรอรับค่าการกดแป้นพิมพ์จากผู้ใช้ 1 ครั้ง

ตารางที่ 1.2 แสดงรหัสบังคับหลุดพ้น (Escape Character)

ค่าคงที่ตัวอักขระ	ชื่อรหัส ASCII	ความหมาย
\n	LF	ขึ้นบรรทัดใหม่(newline)
\t	HT	เลื่อนแท็บ(tab)
\v	VT	เลื่อนแท็บแนวตั้ง(vertical tab)
\b	BS	เลื่อนกลับ(backspace)
\r	CR	การรีเทอร์น(return)
\f	FF	ขึ้นหน้าใหม่(formfeed)
\\		การกดเครื่องหมาย backslash
\'		การกดเครื่องหมาย single qoute

ตัวอย่างที่ 1.2

จงเขียนโปรแกรมแสดงรายงานโดยมีหัวตารางคือ Name Score Percent หลังจากนั้นแสดงข้อความ Somchai 40.00 50.00 ทางจอภาพ

```
#include <stdio.h>                    // นำ header file มารวมกับแฟ้มข้อมูลเพื่อใช้งานฟังก์ชัน
 #include <conio.h>
```

```
void main(void){                          // เริ่มโปรแกรมหลัก
      clrscr();                           // ลบจอภาพ
      printf("\t\t Name \t\t Score \t\t Perctnt\t\t\t");          //สั่งพิมพ์หัวรายงาน
     printf("\n \t\t Somchai \t\t  40.00 \t\t  50.00\t\t");    //แสดงการรายงานชื่อและคะแนนพร้อมร้อยละ
     getch();                                                    // รอรับค่าการแป้นพิมพ์จากผู้ใช้
}                                                                // จบโปรแกรม
```

ผลการรันโปรแกรม โปรแกรมจะแสดง ชื่อ คะแนน ร้อย ออกมาทางจอภาพ และ รอให้ผู้ใช้กดแป้นพิมพ์ 1 ครั้ง แสดงดังภาพ

ตัวแปรในภาษาซี

วิทยา เรืองพรวิสุทธิ์ (2537, หน้า 21) กล่าวว่า ตัวแปรภาษาซีเป็นอักษรหรือตัวเลขที่ประกอบกันเป็นจำนวนตั้งแต่หนึ่งตัวขึ้นไปโดยไม่ใช้ตัวเลขเป็นตัวแรกของชื่อ ขณะที่ นุกูล กระจาย (2540, หน้า 93) กล่าวว่า แวเรียเบิล (variable) หมายถึงชื่อที่ใช้สำหรับเก็บค่าของข้อมูล และ Kris Jamsa และ Lars Klander (1998, p.23) กล่าวว่า การกำหนดตัวแปรให้กับโปรแกรม ซึ่งคือการให้ชื่อที่เกี่ยวของโปรแกรมและกำหนดตำแหน่งของหน่วยความจำ หมายถึง การเก็บค่าที่โปรแกรมใช้โดยสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดช่วงชีวิตของโปรแกรม

ในเรื่อง “เครื่องหมายและการคำนวณ” (2007) กล่าวว่า “การจองพื้นที่ในหน่วยความจำของคอมพิวเตอร์สำหรับเก็บข้อมูลที่ต้องใช้ในการทำงานของโปรแกรม โดยมีการตั้งชื่อเรียกหน่วยความจำในตำแหน่งนั้นด้วย เพื่อความสะดวกในการเรียกใช้ข้อมูล ถ้าจะใช้ข้อมูลใดก็ให้เรียกผ่านชื่อของตัวแปรที่เก็บเอาไว้”

ดังนั้น ตัวแปร หมายถึง ชนิดของข้อมูลที่มีใช้ในโปรแกรมภาษาซี หรือชนิดของข้อมูลที่สามารถใช้ได้ นำมาคำนวณหรือรับค่าแสดงผล ต่าง ๆ ในโปรแกรม ซึ่งการใช้งานนั้นจะต้องมีการประกาศตัวแปรขึ้นมาใช้งาน ลักษณะของตัวแปรที่จะใช้งานนั้นจะเป็นลักษณะคล้ายกับการใช้งานในคณิตศาสตร์ แต่จะมีข้อแตกต่างก็คือตัวแปรใดก็ตามที่ประกาศขึ้นมาใช้จะต้องระบุด้วยว่าตัวแปรตัวนั้นมีลักษณะของข้อมูลเป็นเช่นไร เช่น เป็นตัวเลข อักขระ หรือ จำนวนจริง เป็นต้น

1. ชนิดของตัวแปร

ชนิดของตัวแปร คือ ประเภทของข้อมูลต่างๆ ที่ผู้เขียนโปรแกรมสามารถกำหนดขึ้นมาได้เพื่อใช้ในการเขียนโปรแกรม โดยจะนำมาใช้สำหรับเก็บข้อมูลที่จะป้อนให้กับคอมพิวเตอร์ นำมาไว้ใช้สำหรับแสดงผล หรือช่วยในการคำนวณผลแบบชั่วคราวก็ได้ การเขียนโปรแกรมจะต้องมีความเข้าใจและสามารถใช้งานตัวแปรประเภทต่างๆ ดังกล่าวได้อย่างชำนาญ ชนิดของตัวแปรแสดงได้ดังนี้

ชนิดตัวแปร	ชนิดของข้อมูล
char	ข้อมูลชนิด อักขระ (character)
int	ข้อมูลชนิดตัวเลขจำนวนเต็ม (integer)
short int	ข้อมูลชนิดเลขจำนวนเต็มแบบสั้น (short integer)
long int	ข้อมูลชนิดเลขจำนวนเต็มแบบยาว (long integer)
unsigned int	ข้อมูลชนิดตัวเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย (unsigned
float	integer)
float	ข้อมูลชนิดจำนวนจริง (floating point)
double	ข้อมูลชนิดจำนวนจริงที่มีความแม่นยำเป็นสองเท่า (double-
	precision floating point)
long double	ข้อมูลชนิดจำนวนจริงที่มีความแม่นยำเป็นสองเท่าแบบยาว (long
	double-precision floating point)
	(ที่มา : Borland International, 1992)

การประกาศตัวแปรดังข้างต้น จะต้องใช้เนื้อที่หน่วยความจำของเครื่อง ซึ่งการใช้หน่วยความจำดังกล่าว มีเนื้อที่ดังต่อไปนี้

ชนิดของตัวแปร	ใช้หน่วยความจำ(Bit)	ช่วงค่าของตัวแปร(Range)
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	3.4 * (10**-38) to 3.4 * (10**+38)
double	64 bits	1.7 * (10**-308) to 1.7 * (10**+308)
long double	80 bits	3.4 * (10**-4932) to 1.1 * (10**+4932)

(ที่มา : Borland International, 1992)

หมายเหตุ ** หมายถึงเลขยกกำลัง เช่น 10**-38 หมายถึง ยกสิบกำลังลบสามสิบแปด เป็นต้น

2. การประกาศตัวแปร

การประกาศตัวแปรนั้นจะต้องประกาศตัวดังนี้

ชนิดของตัวแปร ชื่อตัวแปร;

ตัวอย่าง

int x; หมายความว่า ประกาศตัวแปรเป็นชนิดตัวเลขจำนวนเต็ม ชื่อตัวแปร x

float y,z; หมายความว่า ประกาศตัวแปรเป็นชนิด จำนวนจริง โดยมีตัวแปร 2 ตัว คือ y และ z

การประกาศตัวแปรอาจจะกำหนดค่าเริ่มต้นให้กับตัวแปรก็ได้เช่น

int x=50,y=45;

หมายความว่า ประกาศตัวแปรเป็นชนิดเลขจำนวนเต็ม โดยมีตัวแปร 2 ตัวคือ x มีค่าเท่ากับ 50 และ y มีค่าเท่ากับ 45 เป็นต้น

3. ค่าคงที่

ธีรวัฒน์ ประกอบผล (2545, หน้า 39) ระบุไว้ว่า ค่าคงที่เป็นค่าในหน่วยความจำที่มีค่าคงที่ตลอดโปรแกรม ในการประกาศค่าคงที่จะเป็นการกำหนดชื่อให้กับค่าคงที่ ถ้าในโปรแกรมส่วนใดเรียกชื่อที่ประกาศไว้ก็จะได้ข้อมูลตามที่กำหนด

วิทยา เรืองพรวิสุทธิ์ (2537,หน้า 24) กล่าวว่า ตัวแปรที่ถูกกำหนดค่าภายนอกโปรแกรมนั้นเรียกว่าค่าคงที่เชิงสัญลักษณ์ (symbolic constant) ซึ่งปกติมักเขียนด้วยอักษรตัวใหญ่ และการกำหนดค่าเช่นนี้เรียกว่า การกำหนดแบบแมโคร (macro defintion) โดยใช้เครื่องหมาย #

ดังนั้น ค่าคงที่ (Constant) หมายถึงค่าที่กำหนดขึ้นมาแล้วคงอยู่ตลอดในการใช้งานโปรแกรมและไม่สามารถเปลี่ยนแปลงค่าได้ตลอดการใช้งานในโปรแกรมนั้นๆ

การกำหนดตัวแปรค่าคงที่ในโปรแกรมภาษาซีจะต้องนำหน้าด้วยเครื่องหมาย #define เสมอและควรจะกำหนดไว้หัวโปรแกรมเสมอ

ตัวอย่างการกำหนดค่าคงที่ เช่น

```
#define PI 3.141
#define BOOLEAN_YES 1
#define BOOLEAN _NO 0
#define TRUE 1
#define MAX 50
#define BELL '\007'
```

ตัวอย่างที่ 1.3

จงเขียนโปรแกรมคิดหาพื้นที่ของรูปวงกลมโดยป้อนค่ารัศมีของวงกลมให้โปรแกรมคำนวณ

```
#define PI 3.141                // กำหนดค่าคงที่ PI
#include <stdio.h>              // เรียก header file เข้ามารวมในโปรแกรมเพื่อใช้งานฟังก์ชัน
#include <conio.h>
```

```
void main(void){                // เริ่มต้นโปรแกรมหลัก
   float radius,area;           // กำหนดตัวแปรชนิดจำนวนจริง radius แทนรัศมี area แทนพื้นที่วงกลม
   clrscr();                    // ลบหน้าจอ
   cprintf("Enter Radius :");  //แสดงข้อความให้ป้อนค่ารัศมี
   scanf("%f",&radius);         // ให้ผู้ใช้ป้อนรัศมีของวงกลม
   area=PI*radius*radius;     //คำนวณพื้นที่วงกลม
   printf("\nThis area :%.2f",area);   // แสดงค่าพื้นที่วงกลมที่คำนวณได้
   getch();                             //รับค่าการกดแป้นพิมพ์
}                                       //จบโปรแกรม
```

ผลการรันโปรแกรมจะให้ผู้ใช้ป้อนค่าเส้นรัศมีของวงกลมแล้วโปรแกรมจะคำนวณเส้นรอบวงมาให้โดยคิดจากค่า PI ที่กำหนดได้

ฟังก์ชันการแสดงผล

คำสั่งที่ใช้สำหรับการแสดงผลเบื้องต้น เพื่อให้ผู้เรียนได้มีความรู้เรื่องของรูปแบบการเขียนโปรแกรม การแสดงผลออกทางจอภาพตามที่ต้องการ ส่วนแรกจะนำเสนอฟังก์ชันสำหรับแสดงผลที่มาจาก conio.h ซึ่งใช้ในการควบคุมการแสดงผลทางคอนโซลมาตรฐาน หลังจากนั้นจะนำเสนอฟังก์ชันที่มาจาก stdio.h ที่ใช้เฉพาะการแสดงผลออกทางจอภาพ นอกจากนั้นยังนำเสนอเกี่ยวกับการแสดงรหัสแอสกีและเสียงจากลำโพงมาตรฐานที่มีในเครื่องอีกด้วย

1. ฟังก์ชันการแสดงผลจาก CONIO.H

conio.h เป็นโปรโตไทป์หรือเฮดเดอร์ ที่บรรจุฟังก์ชันที่ใช้ในการแสดงผล และรับข้อมูล กลุ่มของฟังก์ชันที่เป็นส่วนแสดงผลข้อมูลจะใช้สำหรับการแสดงผลออกทางคอนโซลมาตรฐานซึ่งในที่นี้ได้แก่จอภาพนั้นเอง ฟังก์ชันทั้งหมดที่นำเสนอนี้ได้นำมาจากโปรแกรมเทอร์โบซี 3.0 ตัวอย่างฟังก์ชัน เช่น cprintf, clrscr, textcolor, textbackground เป็นต้น สำหรับอีกกลุ่มคือกลุ่มของการรับข้อมูลจากอินพุตมาตรฐานซึ่งคือแป้นพิมพ์ ตัวอย่างฟังก์ชันกลุ่มนี้ เช่น kbhit, getche, getch เป็นต้น นอกจากสองกลุ่มดังกล่าวอาจมีการส่งข้อมูลออกพอร์ตมาตรฐานอีกด้วย เช่น ฟังก์ชัน outport, outportb เป็นต้น คำสั่งแสดงดังกล่าวนี้ใช้ได้ในโหมดเท็กซ์ (Text Mode) ซึ่งโดยปกติทำงานในโหมดเท็กซ์ จะต้องคำนึงถึงรายละเอียดเกี่ยวกับจอภาพที่สามารถแสดงได้ 25

บรรทัด แต่ละบรรทัดแสดงได้ 80 คอลัมน์ ถ้าใช้ในโหมดของระบบปฏิบัติการดอสจะได้จำกัดเท่านั้นแต่หากในระบบปฏิบัติการของวินโดว์เอ็กซ์พีอาจแสดงจำนวนบรรทัดได้มากถึง 50 บรรทัดได้ แนวคิดของจำนวนบรรทัดและคอลัมน์แสดงในภาพที่ 4.2

การแสดงผลในโหมดนี้จะใช้ได้เฉพาะในส่วนของเทอร์โบซีที่ใช้สำหรับดอสโหมดเท่านั้น ซึ่งหากโปรแกรมภาษาซีอื่น ๆ อาจมีการจัดการที่แตกต่างกันขึ้นอยู่กับแต่ละค่ายของโปรแกรม ในการอ้างอิงการเขียนโปรแกรมในบทนี้ จะเน้นการแสดงผลในโหมดนี้เท่านั้น

ฟังก์ชันที่ใช้ในการแสดงผลเบื้องต้นสำหรับผู้ที่เริ่มเขียนโปรแกรม ได้เริ่มต้นศึกษามีดังนี้ clrscr() cprintf() gotoxy() textbackground() และ textcolor()

1.1 ฟังก์ชัน clrscr

หน้าที่ : ลบหน้าจอในโหมดเท็กซ์

การเรียกใช้ : void clrscr(void);

(ที่มา : Borland International, 1992)

1.2 ฟังก์ชัน cprintf

หน้าที่ : เป็นฟังก์ชันที่ทำหน้าที่ในการแสดงผลข้อมูลออกทางจอภาพ

รูปแบบการประกาศใช้ : void cprintf("ข้อความ");

(ที่มา : Borland International, 1992)

ตัวอย่างการเรียกใช้งานเช่น หากต้องการพิมพ์ข้อความ “hello” ออกจอภาพจะแสดงได้คือ cprintf(“hello”); เป็นต้น

1.3 ฟังก์ชัน gotoxy

หน้าที่ : เป็นฟังก์ชันที่สั่งให้การทำงานของตำแหน่งปัจจุบันไปยังตำแหน่งต่าง ๆ บนจอภาพ ซึ่งปกติแล้ว จะเริ่มที่มุมบนซ้ายของจอภาพ จาก ตำแหน่งที่ 1,1 ถึงมุมล่างขวาสุดของจอภาพ ตำแหน่ง 25,80

รูปแบบ : gotoxy(ตำแหน่งแกน x,ตำแหน่งแกน y);

(ที่มา : Borland International, 1992)

ตัวอย่างเช่น ถ้ากำหนด gotoxy(40,20); หมายถึงให้ตำแหน่งเคอร์เซอร์ ไปที่ตำแหน่ง คอลัมน์ 40 บรรทัดที่ 20 เป็นต้น

1.4 ฟังก์ชัน textcolor,textbackground

หน้าที่ : textcolor() เป็นฟังก์ชันที่สั่งให้แสดงผลข้อความเป็นสีต่าง ๆ ปกติจะกำหนดได้ตั้งแต่ 0-15 หรือ BLACK ถึง WHITE

: textbackground() เป็นฟังก์ชันที่สั่งให้พื้นของตัวอักษรเป็นสีต่าง ๆ ลักษณะของสีจะเหมือนกับ textcolor()

เฮดเดอร์ไฟล์ : CONIO.H

รูปแบบ : textcolor(ค่าของสี);

: textbackground(ค่าของสี);

ค่าที่คือหลังการเรียกใช้งาน : ไม่มี

ความสามารถในการใช้งาน : เทอร์โบซีด้วยดอส

(ที่มา : Borland International, 1992)

ตัวอย่างเช่น textcolor(BLACK); textbackground(WHITE); หรือ textcolor(0); textbackground(15); ซึ่งสามารถใช้ได้ทั้งสองกรณี ซึ่งภาพที่ 4.12 แสดงค่าคงที่ของสีที่สามารถกำหนดได้

Constant	Value	Back-grnd?	Fore-grnd?
BLACK	0	Yes	Yes
BLUE	1	Yes	Yes
GREEN	2	Yes	Yes
CYAN	3	Yes	Yes
RED	4	Yes	Yes
MAGENTA	5	Yes	Yes
BROWN	6	Yes	Yes
LIGHTGRAY	7	Yes	Yes
DARKGRAY	8	No	Yes
LIGHTBLUE	9	No	Yes
LIGHTGREEN	10	No	Yes
LIGHTCYAN	11	No	Yes
LIGHTRED	12	No	Yes
LIGHTMAGENTA	13	No	Yes
YELLOW	14	No	Yes
WHITE	15	No	Yes
BLINK	128	No	***

ภาพที่ 1.3 ค่าคงที่ของสี สำหรับใช้งานในฟังก์ชันส่วนของโหมดเท็กซ์

(ที่มา : Borland International, 1992)

การกำหนดค่าของสีนั้นจะระบุว่าสีที่เป็นสีพื้นตั้งแต่ 0-7 เท่านั้น ส่วนสีที่สามารถกำหนดเป็นตัวอักษรได้นั้นจะได้ตั้งแต่ 0-15 และหากต้องการให้ตัวอักษรที่แสดงทางจอภาพให้บวกค่าของสีด้วย 128 จะทำให้ตัวอักษรนั้นกระพริบได้ขณะแสดงผล ตัวอย่างเช่น textcolor(1+128); ผลลัพธ์ของการแสดงผลของอักขระต่อจากคำสั่งนี้จะกระพริบทุกอักขระ เป็นต้น

ตัวอย่างที่ 1.4

จงเขียนโปรแกรมแสดงข้อความ มุมซ้ายของจอภาพแสดงข้อความ My name is Praram , มุมขวาแสดง My name is Hero , กลางจอภาพแสดงข้อความ Hello world , มุมล่างซ้ายแสดงข้อความ My name is Zoro , มุมล่างขวาแสดงข้อความ My name is Rambo

```
#include <conio.h>                  //นำ header file เข้ามารวมเพื่อเรียกใช้งานฟังก์ชัน
void main(void){                    //เริ่มต้นโปรแกรมหลัก
clrscr();                           //ลบหน้าจอ
gotoxy(1, 1);                       //สั่งตำแหน่งการทำงานไปที่ตำแหน่ง คอลัมน์ 1 บรรทัดที่  1
cprintf("My name is Praram");       //แสดงข้อความ
gotoxy(60, 1);                      //สั่งตำแหน่งการทำงานไปที่ตำแหน่ง คอลัมน์ 60 บรรทัดที่  1
cprintf("My name is Hero");         //แสดงข้อความ
gotoxy(35, 12);                     //สั่งตำแหน่งการทำงานไปที่ตำแหน่ง คอลัมน์ 35 บรรทัดที่ 12
cprintf("Hello world");             //แสดงข้อความ
gotoxy(1, 25);                      //สั่งตำแหน่งการทำงานไปที่ตำแหน่ง คอลัมน์ 1 บรรทัดที่ 25
cprintf("My name is Zoro");         //แสดงข้อความ
gotoxy(60, 25);                     //สั่งตำแหน่งการทำงานไปที่ตำแหน่ง คอลัมน์ 60 บรรทัดที่ 25
cprintf("My name is Rambo");        //แสดงข้อความ
getch();                            //รอรับการกดแป้นพิมพ์ 1 ครั้ง
}                                   //จบโปรแกรม
```

ผลการรันโปรแกรมจะพิมพ์ข้อความบนจอภาพทั้ง 4 มุมของจอภาพโดยกลางจอภาพจะแสดงข้อความ Hello world แล้วรอรับการกดแป้นพิมพ์จากผู้ใช้ 1 ครั้ง

2. ฟังก์ชันการแสดงผลจาก STDIO.H

ฟังก์ชันการใช้งานในส่วนนี้จะเป็นการรับและการแสดงผลมาตรฐานโดยทั่วไป จะสามารถใช้งานได้กับภาษาซีในทุก ๆ ค่ายของผู้ผลิต ไม่ว่าจะเป็นบอร์แลน ไมโครซอฟต์หรือแม้กระทั่งภาษาซีที่อยู่บนยูนิกซ์หรือลีนุกซ์ก็ตาม ดังนั้นในหัวข้อนี้จะแสดงรายละเอียดเฉพาะส่วนการแสดงผลเท่านั้นส่วนการรับข้อมูลจะแสดงรายละเอียดในบทต่อไป รายละเอียดของฟังก์ชันที่มีใน stdio.h

ฟังก์ชันที่จะแสดงผล ที่สำคัญที่จะกล่าวในบทนี้ได้ แก่ printf putc และ putchar สำหรับการแสดงผลข้อมูลต่าง ๆ นั้น จะต้องกำหนดรูปแบบของการแสดงผล ซึ่งในภาษาซีเรียกว่ารหัสบังคับการแสดงผลหรือรับข้อมูล ซึ่งกล่าวถึงแล้วในบทที่ผ่านมา สำหรับในบทนี้จะนำทั้งรหัสหลุดพ้นและรหัสบังคับการรับหรือแสดงผลมาใช้ร่วมด้วย

2.1 ฟังก์ชัน printf

หน้าที่ : เป็นฟังก์ชันที่ใช้ในการแสดงผลข้อมูลซึ่งอาจเป็นข้อความหรือการตัวแปร

รูปแบบ : printf("รูปแบบบังคับการแสดงผล",ตัวแปรที่1,ตัวแปรที่2,…);

(ที่มา : Borland International, 1992)

ตัวอย่างการใช้งานเช่น

```
printf("%c",j);       // คำสั่งแสดงผล ค่า j ในรูปแบบของอักขระ หรือ
printf(“%d”,’A’);    //หมายถึงแสดง อักขระ A ในรูปแบบตัวเลข
print("Your name:%s\n",name);     // แสดงข้อความสตริงจาก ตัวแปร name ที่ผู้ใช้ป้อนเข้าไปออกมาทางจอภาพแล้วขึ้นบรรทัดใหม่ จากรหัสหลุดพ้น \n
printf("\t\t Name \t\t Score \t\t Perctnt\t\t");  //สั่งเลื่อนแท็บ 2 แท็บก่อนพิมพ์ข้อความ Name แล้วเลื่อนไปอีก 2 แท็บ พิมพ์ข้อความ Score และเลื่อนอีก 2 แท็บแล้วจึงพิมพ์ข้อความ Percent หลังจากนั้นจึงเลื่อนอีก 2 แท็บ เป็นต้น
```