

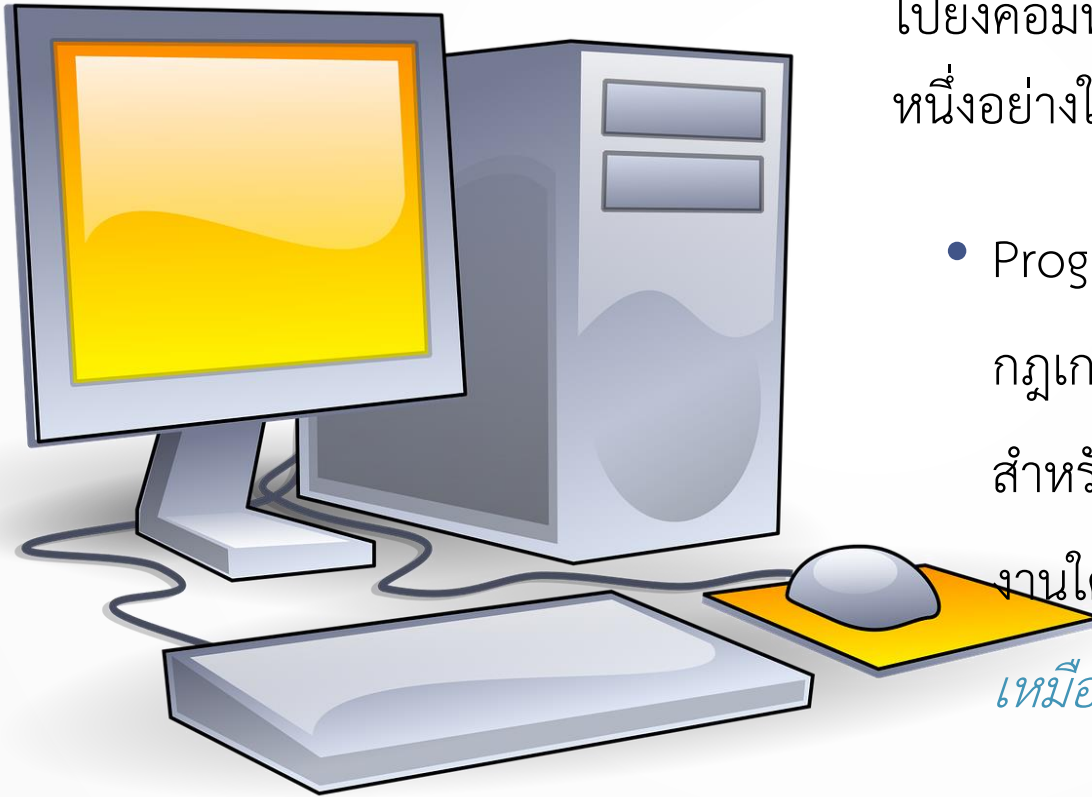


Introduction to Programming

Lecturer: Watcharin Sarachai, Ph.D.

The Program

- Program เป็นกลุ่มของชุดคำสั่งที่มีการสั่งงานไปยังคอมพิวเตอร์โดยตรง เพื่อทำงานอย่างหนึ่งอย่างใดและให้ผลลัพธ์ที่ผู้ใช้ต้องการ
- Programming language เป็นกลุ่มของกฎเกณฑ์ที่ถูกจัดไว้อย่างเป็นรูปแบบสำหรับเป็นหนทางบอกให้คอมพิวเตอร์ว่างานใดที่ต้องดำเนินการ *เปรียบได้เป็นเหมือนกับภาษาเฉพาะของคอมพิวเตอร์*



History of C Cont.

1960

ALGOL

International Group

1967

BCPL

Martin Richards

1970

B

Ken Thompson

1972

Traditional C

Dennis Ritchie

1978

K&R C

Kernighan and Ritchie

1989

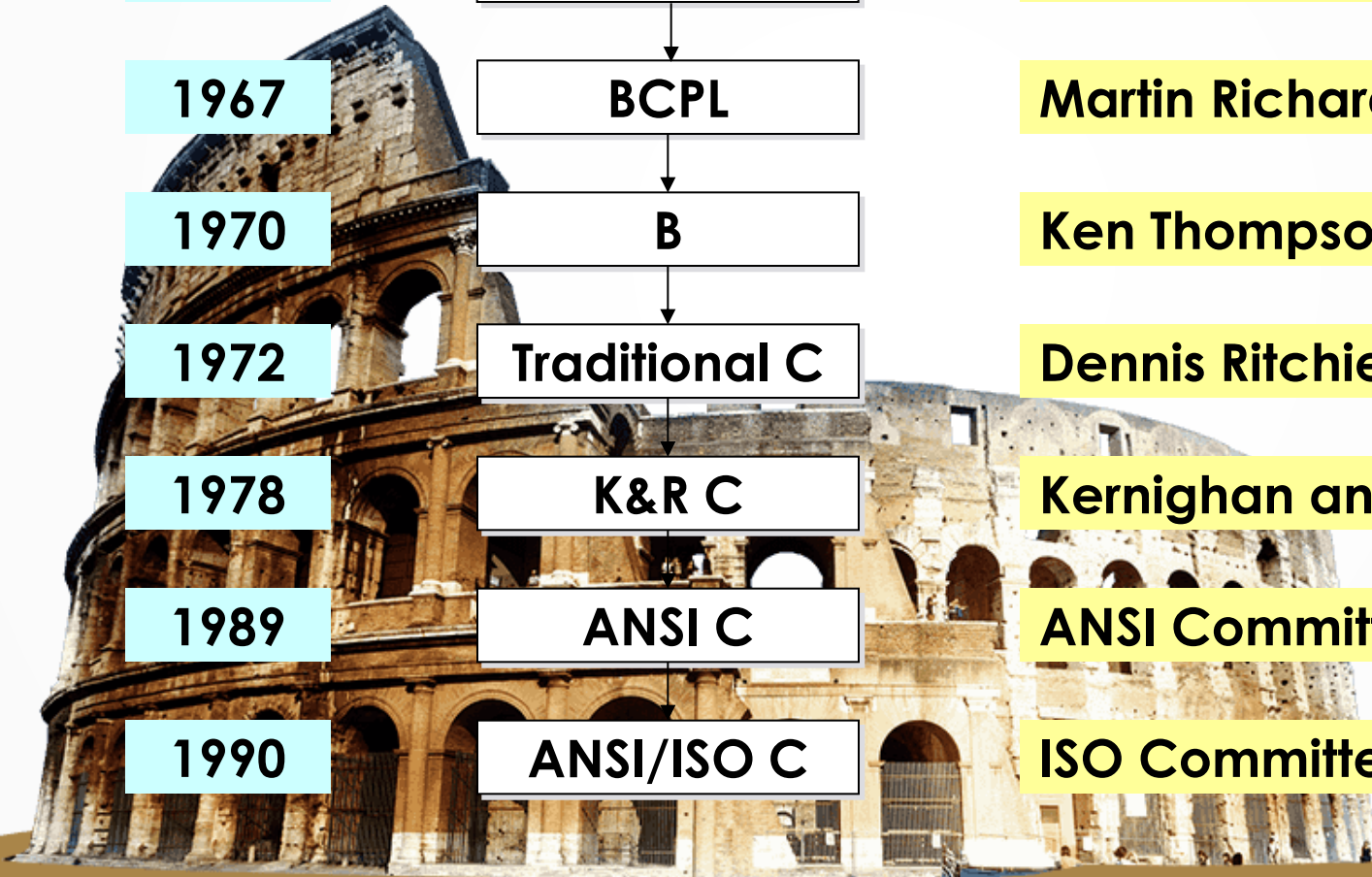
ANSI C

ANSI Committee

1990

ANSI/ISO C

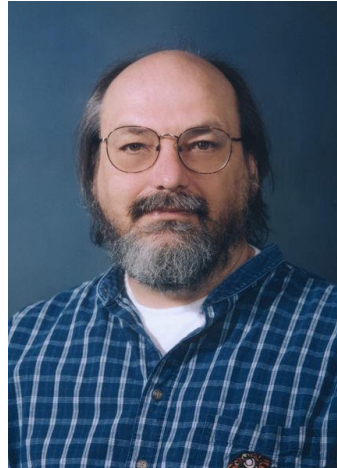
ISO Committee



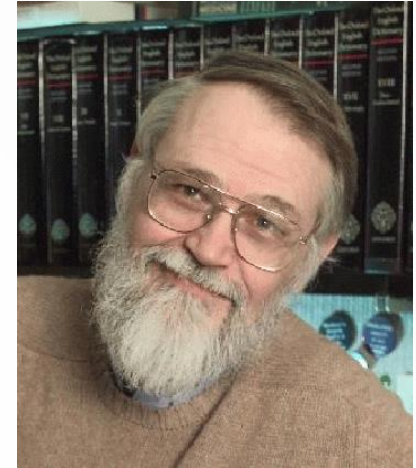
Origins of C



Denis Ritchie



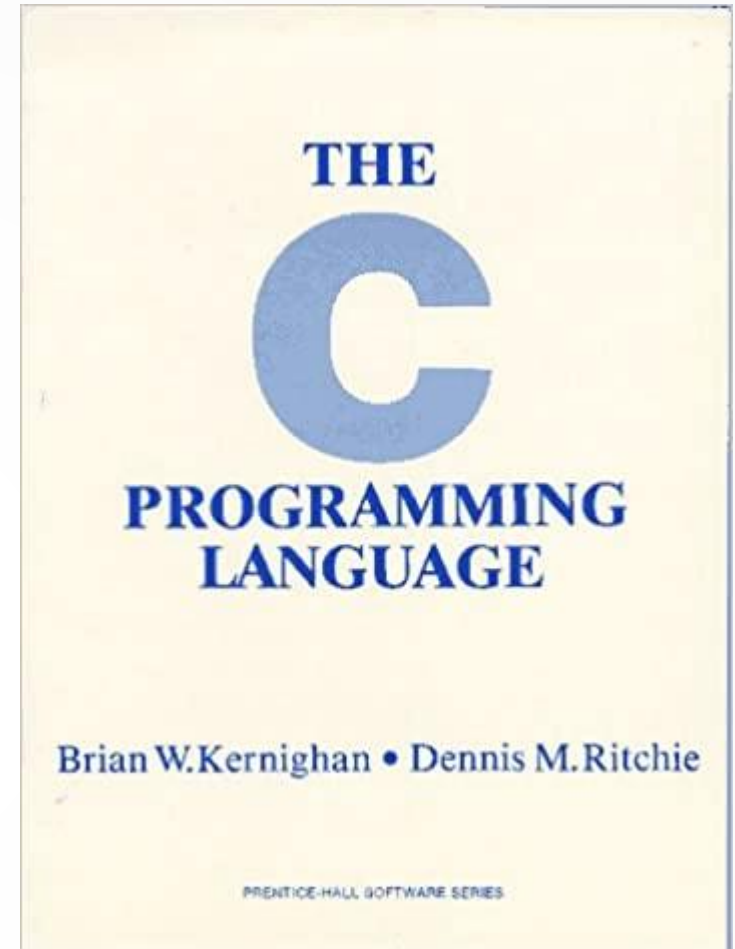
Ken Thompson



Bryan Kernighan

- ปี 1971 Ritchie และ Thompson ได้พัฒนาระบบยูนิคซ์ขึ้นโดยใช้ภาษาแอสเซมบลี แต่ระบบดังกล่าวไม่สามารถ Portable
- ดังนั้นจึงจำเป็นต้องพัฒนาจากภาษา BCPL โดยเพิ่มเติมความสามารถของภาษาและเรียกชื่อย่อ ๆ ว่า C

Origins of C



- ยูนิกซ์ได้ถูกเขียนขึ้นใหม่โดยใช้ C และถูกยอมรับอย่างแพร่หลายในเวลาต่อมา
- ปี 1978 Kernighan & Ritchie ได้เผยแพร่หนังสือ “The C Programming Language” (Traditional C) or K&R C

Programming Languages

High-level Language

$POS = VEL * (TIME + INC)$

Assembly Language

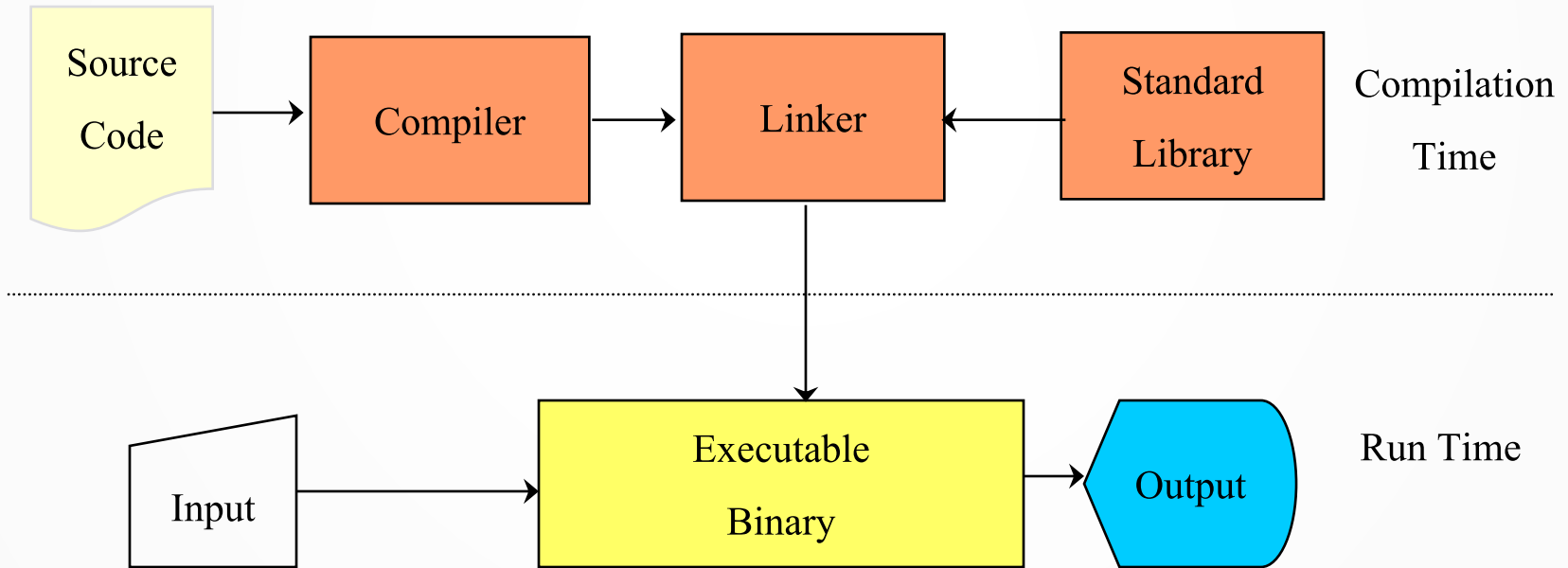
```
LDA TIME  
ADA INC  
MUL VEL  
STA POS
```

Machine Language

```
0000100100010010  
0010000100010100  
0011100100010000  
0001000100010110
```

Compilation Process

- **Compiler:** เป็นโปรแกรมที่ใช้สำหรับแปลงจากภาษาระดับสูงให้อยู่ในรูปของภาษาระดับเครื่อง (object code)



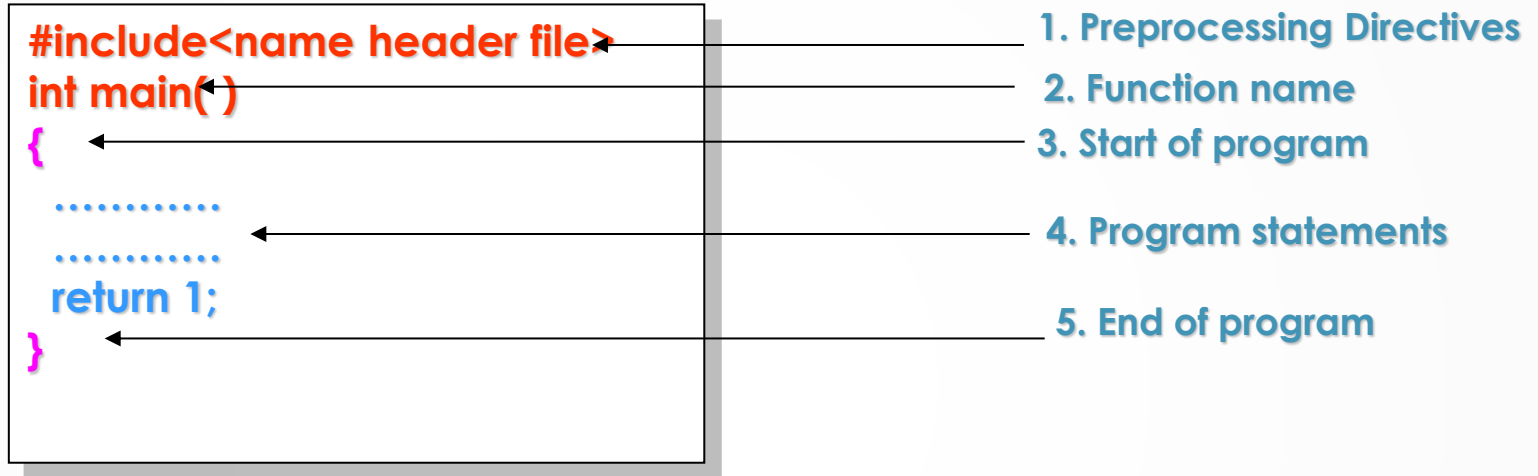
Compilers & Programs

- Source program:
 - เป็นรูปแบบของโปรแกรมที่ถูกเขียนขึ้นตามข้อกำหนดของภาษานั้น ๆ
 - ไฟล์ที่ถูกเขียนขึ้นโดยใช้ภาษาซี (C)จะมีนามสกุลเป็น '.c'
- Object program :
 - เป็นผลจากการทำงานของคอมไพเลอร์ ซึ่งถือเป็นภาษาระดับเครื่อง
 - ไฟล์แบบนี้มีนามสกุลเป็น '.obj'
- Executable program:
 - ไฟล์ .obj จะถูกเชื่อมต่อเข้ากับส่วนประกอบอื่น ๆ ที่จำเป็นในการทำงาน
 - ไฟล์แบบนี้จะมีนามสกุลเป็น '.exe'

โครงสร้างโปรแกรมภาษาซี

C Programming Structure

Format of Simple C Programs



```
#include <stdio.h>

int main() {
    printf("Hello World\n");
    return 1;
}
```

C program may contain one or more sections as shown below:

DOCUMENTATION SECTION

LINK SECTION

DEFINITION SECTION

GLOBAL DECLARATION SECTION

Main() Function section

{

Declaration part

Executable part

}

SUBPROGRAM SECTION

User defined functions

Statements

- ทุก ๆ โปรแกรมที่เขียนด้วยภาษาซีจะต้องประกอบด้วยฟังก์ชัน *main()* function definition เสมอ
- *main()* เป็นจุดเริ่มต้นการประมวลผลของโปรแกรม
- เช่นเดียวกับการกำหนดฟังก์ชันแบบอื่น ๆ ฟังก์ชัน *main()*
 - ถูกกำหนดโดยเครื่องหมาย { }
 - จะต้องประกอบไปด้วย statements ตั้งแต่หนึ่งหรือมากกว่า
- Statements เป็นการกระทำที่เกิดขึ้นจากคำสั่งในภาษา C
 - เช่น *printf* ทำการพิมพ์บางสิ่งไปยัง standard output.

C Source - #include ...

- เนื่องจากโปรแกรมตัวอย่างใช้ Standard I/O Library ในการนำเสนอข้อมูล “Hello World” ผ่านหน้าจอ ดังนั้นจึงจำเป็นต้องเรียกใช้ include <stdio.h>
- **#include <stdio.h>**
 - มีลักษณะการทำงานเป็น Preprocessor
 - เพื่อแจ้งให้คอมไพเลอร์โหลดข้อมูลที่มีอยู่แล้วจากที่ใด
 - **<stdio.h>** ระบุกลุ่มคำสั่งที่ทำงานเกี่ยวกับ standard input/output เช่น
 - Macros (ชื่อของค่าคงที่)
 - Prototypes functions (เช่น ฟังก์ชัน printf())

ตัวอักษรหรืออักขระในภาษา C (C Character Sets)

- ภาษาซีได้เตรียมกลุ่มอักขระต่าง ๆ ให้ไว้ใช้งานที่เรียกว่า Character Sets มีการแบ่งออกเป็น 2 กลุ่มใหญ่ ๆ คือ
 - 1. Basic Character Set ประกอบด้วยกลุ่มอักขระ
 - อักษรตัวพิมพ์ใหญ่ (Alphabets Upper Case)
ประกอบด้วยอักษร A-Z จำนวน 26 ตัว
 - อักษรตัวพิมพ์เล็ก (Alphabets Lower Case)
ประกอบด้วยอักษร a-z จำนวน 26 ตัว
 - ตัวเลข (Decimal Digital)
ประกอบด้วยตัวเลข 0-9- จำนวน 10

ตัวอักษรแบบกราฟิก (Graphic Characters)

, (Comma)	((left parenthesis)
; (Semicolon)	) (right parenthesis)
. (period)	[(left bracket)
? (Question mark)	] (right bracket)
! (exclamation)	{ (left brace)
: (colon)	} (right brace)
+(plus)	< (less than)
- (minus)	> (greater than)
* (asterisk)	= (equal sign)
/ (slash)	& (ampersand)
\ (backslash)	% (percent sign)
(vertical bar)	# (number sign)
' (single quot)	^ (caret)
“ (double quote)	_ (under score)

อักขระในภาษา C (C Character Sets)

- ตัวอักขระแบบช่องว่าง (Whitespace Characters)

ประกอบด้วยอักขระที่เป็นช่องว่างในลักษณะ ต่าง ๆ จำนวน 5 ตัว Blank space, horizontal tab, vertical tab, newline, form feed

2. Execution Character Set

- อักขระที่เป็นค่าว่าง (Null Character) อักขระที่เป็นค่าว่าง หรือค่า Null จะใช้สัญลักษณ์ `\0` (แบ็กสแลช แล้วตามด้วยเลขศูนย์) ซึ่งเป็นตัวกำหนดจุดสิ้นสุดของตัวอักษรสตริง (Character String)
- อักขระควบคุม (Escape Sequence) อักขระควบคุม เป็นรหัสที่ใช้สำหรับควบคุมการแสดงผลทั้งทางจอภาพและเครื่องพิมพ์

<code>\a</code> alert (bell)	<code>\\</code> backslash
<code>\b</code> backspace	<code>\?</code> backslash
<code>\f</code> form feed	<code>\'</code> single quote
<code>\n</code> newline	<code>\"</code> double quote
<code>\r</code> carriage return	<code>\ooo</code> octal number
<code>\t</code> horizontal tab	<code>\xhh</code> hexadecimal number
<code>\v</code> vertical tab	

C Source - printf...Comment

- ฟังก์ชัน “printf ()” พิมพ์ข้อความที่ต้องการออกทางหน้าจอ
- ข้อความที่อยู่ภายในวงเล็บ(“Hello World!\n”) จะถูกเรียกว่า Argument
- ส่วนสัญลักษณ์ “\n” ใช้สำหรับการส่งให้ขึ้นบรรทัดใหม่
- ทุก ๆ ประโยคคำสั่งในภาษาซีต้องสิ้นสุดด้วยเครื่องหมาย “;” เสมอ
- ส่วนที่ไม่ต้องการให้มีการคอมไพล์อาจจะบุ้ในรูปแบบของ comment ดังนี้

```
/* An example of a C comment */
```

หรือ

```
/* This is a multiline comment. Indentation is not  
compulsory
```

```
but it makes the code more readable */
```

```
// In some versions this comment can be used
```

C Source - printf...Comment

- เครื่องหมาย ‘\’ (Back Slash) เมื่อนำไปวางไว้หน้าตัวอักษรใด ๆ ในฟังก์ชัน printf() จะมีผลทำให้อักขรตัวนั้นไม่ถูกนำเสนผ่านหน้าจอ แต่จะใช้เป็นตัวบังคับให้โปรแกรมแสดงผลอย่างใดอย่างหนึ่งแทน เช่น สัญลักษณ์ ‘\n’ เมื่อประมวลผลแล้ว จะบังคับให้เคอร์เซอร์เลื่อนไปขึ้นบรรทัดใหม่ที่คอลัมน์แรกเสมอ

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

```
int main() {  
    printf("Computers, Computers, everywhere");  
    printf("\n\tAs far as I can C");  
    return 0;  
}
```

Output: Computers, Computers, everywhere

As far as I can C

Escape Sequence in C program

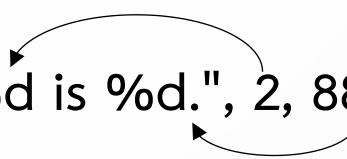
สัญลักษณ์	ชื่อของคาแรกเตอร์	ค่าเป็นตัวเลข
\0	Null Character	0
\a	Alert(bell) character	7
\b	Backspace	8
\t	Tab (horizontal)	9
\n	Newline (CR/LF pair)	10
\v	Tab (vertical)	11
\f	Form feed	12
\r	Carriage return (CR)	13
\"	Double Quotation mark	34
\'	Single Quotation mark	39
\\	Backslash	92

printf() : Format Specifier

- ฟังก์ชัน printf() นอกจากจะใช้แสดงข้อความที่ต้องการไปปรากฏบนหน้าจอแล้วยังสามารถใช้สำหรับแสดงผลลัพธ์ที่ได้จากการคำนวณทางคณิตศาสตร์ ไม่ว่าจะเป็นอยู่ในรูปของค่าคงที่หรือนิพจน์ทางคณิตศาสตร์ โดยการนำเสนอจะกระทำผ่านสัญลักษณ์ % นำหน้าตัวอักษรที่ได้กำหนดไว้แล้วที่มีลักษณะดังนี้

printf("%prefix", ค่าคงที่ หรือค่าตัวแปรหรือนิพจน์ทางคณิตศาสตร์);

printf("The program no.%d is %d.", 2, 88);



Output: The program no. 2 is 88.

printf() : Format Specifier

printf("The total of 10 and 18 is %d ", 10+18);

Output: **The total of 10 and 18 is 28**

printf("Division 2 number of 50 and 25 is %d ", 50/25);

Output: **Division 2 number of 50 and 25 is 2**

% prefix Symbol

- นอกจากสัญลักษณ์ %d แล้วในภาษาซี ยังมีรูปแบบสัญลักษณ์ที่กำหนดไว้อื่น ๆ ที่ใช้แสดงค่าตามชนิดของตัวแปรที่กำหนดได้ ดังต่อไปนี้
 - %c ใช้แสดงค่าตัวแปรแบบอักขระ(Character)
 - %s ใช้แสดงค่าตัวแปรแบบสตริง (String)
 - %e ใช้แสดงค่าตัวแปรทศนิยมตามรูปแบบของ Exponential
 - %f ใช้แสดงค่าตัวแปรแบบทศนิยมคงที่
 - %u ใช้แสดงค่าแบบตัวเลขไม่คิดเครื่องหมาย (Unsigned Integer)
 - %x ใช้แสดงค่าตัวแปรในรูปของเลขฐาน 16
 - %o ใช้แสดงค่าตัวแปรในรูปของเลขฐาน 8
 - %ld ใช้แสดงค่าตัวแปรแบบเลขจำนวนเต็มแบบยาว (long integer)
 - %lf ใช้แสดงค่าตัวแปรแบบทศนิยมแบบละเอียด (double precision)

Using % prefix Symbol

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

```
main() {
```

```
    printf("%s is %d km. from CHIANG-MAI ", "BANGKOK", 750);
```

```
}
```

Output: **BANGKOK is 750 km. from CHIANG-MAI**

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

```
main() {
```

```
    printf("The decimal Value of letter %c is %d", 'a', 'a');
```

```
}
```

Output: **The decimal Value of letter a is 97**

Using % prefix Symbol : Review

A	g	e		i	s				2	1	.
---	---	---	--	---	---	--	--	--	---	---	---

- การกำหนดตำแหน่งสามารถทำได้โดยการใส่ค่าตัวเลขจำนวนเต็มบวกหรือลบในระหว่างเครื่องหมาย % และตัวอักษรที่แสดงชนิดของตัวแปร ซึ่งค่าตัวเลขจะหมายถึงตำแหน่งของการแสดงผลซึ่งจะถูกจองไว้ตามจำนวนตัวเลขที่กำหนด

```
#include <stdio.h>

main()    {
    int age = 21;
    printf("Age is %4d.", age);
}
```

Using % prefix Symbol : Review

A	g	e		i	s		2	1			.
---	---	---	--	---	---	--	---	---	--	--	---

- ในกรณีที่ตัวเลขเป็นบวกหรือลบจะมีผลต่อการกำหนดจุดเริ่มต้นในการแสดงผล นั่นคือค่าที่เป็นบวกการแสดงผลจะเริ่มต้นจากตำแหน่งด้านขวามือสุดเป็นหลัก ส่วนในกรณีที่ค่าเป็นลบการแสดงผลจะเริ่มจากตำแหน่งซ้ายมือสุดของช่องตำแหน่งที่ถูกจองไว้นั่นเอง

```
#include <stdio.h>

main()    {
    int age = 21;
    printf("Age is %-4d.", age);
}
```

Using % prefix Symbol : Review

- โดยปกติค่าตัวเลขแบบทศนิยมคงที่ เมื่อใช้กับสัญลักษณ์ %f จะแสดงค่าทศนิยม จำนวน 6 หลักเสมอ หากต้องการแสดงจำนวนหลักทศนิยมให้มีจำนวนมากกว่าหรือน้อยกว่า 6 หลัก ก็สามารถใช้หลักการกำหนดตำแหน่งของการนำเสนอค่ามาประยุกต์ใช้ได้

```
main() {  
    printf("%8.1f %8.1f %8.1f \n", 3.0,12.5,546.8);  
    printf("%25.20f %30.2f",3.0,15.5);  
}
```

Q&A

