

การเขียนโปรแกรมด้วยภาษา C (คู่มืออ้างอิง)		ตัวประมวลผลก่อน(คอมไพเลอร์)ในภาษาซี (C Preprocessors)	ชนิดของข้อมูล
โครงสร้างของโปรแกรมและฟังก์ชัน		\ สัญลักษณ์การต่อบรรทัด นำ ข้อความในบรรทัดลงมาต่อ การขอไฟล์ "ชื่อไฟล์" ซึ่งเป็น ไฟล์ของไลบรารี ที่ติดตั้งในระบบ มารวมในโปรแกรม ขอไฟล์ "ชื่อไฟล์" ซึ่งเป็น ไฟล์ ของผู้ใช้งาน มารวมในโปรแกรม	char 1 byte ตัวอักขระ (character) int 4 bytes ตัวเลขจำนวนเต็ม float 4 bytes เลขจำนวนจริง double 8 bytes เลขจำนวนจริงความละเอียดสูง short 2 bytes เลขจำนวนเต็มปริมาณน้อย long 4 bytes เลขจำนวนเต็ม Long long 8 bytes เลขจำนวนเต็มเก็บ 64 บิต signed ประกาศให้เป็นจำนวนที่ติดลบได้ unsigned ประกาศให้เป็นจำนวนบวกเท่านั้น int* หรือ float* พอยเตอร์ชี้ไปที่ข้อมูลชนิด int หรือ float
<pre>int main (void) { ประกาศตัวแปร ประโยคคำสั่ง return ค่าสถานะจบ; }</pre>	ฟังก์ชัน ชื่อ "main" ฟังก์ชันหลักในโปรแกรม แบบไม่มีอาร์กิวเมนต์ จะเขียน void หรือไม่เขียนก็ได้		enum ชนิดของตัวแปรลำดับ {name1=value1, ...}; ประกาศชนิดตัวแปรลำดับ เพื่อกำหนดตัวเลข (value) มาแทนชื่อ (name) ของสิ่งที่เป็นลำดับ เช่นสร้างชนิดตัวแปรใหม่ชื่อ weekend enum weekend {SAT=0, SUN=1}; เราสร้างตัวแปรให้มีชนิดเป็น weekend ได้ และกำหนดค่าเป็น SAT, SUN ได้
<pre>int main(int argc, char* argv[]) { ประกาศตัวแปร ประโยคคำสั่ง return ค่าสถานะจบ; }</pre>	ฟังก์ชัน "main" แบบมีอาร์กิวเมนต์ argc=จำนวนอาร์กิวเมนต์ argv=พอยเตอร์ชี้ไปยังรายการอาร์กิวเมนต์	#include <ชื่อไฟล์> #include "ชื่อไฟล์" #define ชื่อ(ตัวแปร) ข้อความ การประกาศมาโคร "ชื่อ" เช่น ประกาศมาโคร max ดังนี้ #define max(A,B) ((A)>(B) ? (A):(B)) ในโปรแกรมถ้าเขียนคำสั่งว่า max(x, y); จะถูกแทนที่กลายเป็น (x > y ? x:y); (A) ถูกแทนด้วย x , (B) ถูกแทนด้วย y	
ชนิดข้อมูล ชื่อตัวแปร;	การประกาศตัวแปร Variable Declaration	#define ชื่อ ค่า การแทนที่ "ชื่อ" ด้วย "ค่า" เช่น หากกำหนดให้ MAX เป็น 5 ด้วยคำสั่ง #define MAX 5 ทุกๆ จุดในโปรแกรมที่มีคำว่า "MAX" จะถูกแทนที่ด้วยเลข 5	const ชื่อตัวแปร=ค่าคงที่; ประกาศชื่อตัวแปรนั้นเป็นตัวแปรเก็บค่าคงที่ ที่ตั้งค่าได้ครั้งแรกครั้งเดียว
/* ข้อความอธิบาย */	คำอธิบายในโปรแกรม Comments	#undef ชื่อ ยกเลิกการแทนที่ หรือการแทนที่คำหรือมาโคร "ชื่อ" นับจากจุดนี้ในโปรแกรมเป็นต้นไป	void ข้อมูลไม่มีชนิด. การไม่ส่ง, ไม่มีค่า extern ชนิดข้อมูล ชื่อตัวแปร; ประกาศว่าตัวแปรตัวนี้นำมาจากไฟล์อื่น (ตัวแปรสร้างภายนอก)
ชนิดข้อมูล ชื่อฟังก์ชัน (ชนิดข้อมูล ₁ , ...);	การประกาศรูปแบบตัวอย่างของฟังก์ชัน Function Prototype	# ใช้แทนที่สตริงที่อยู่เหนือประกาศ เช่น #define msg(A) printf("%s = %d", #A, (A)) หากในโปรแกรมมีคำสั่ง msg("hello"); จะถูกแทนที่ด้วย printf("%s = %d", "hello", ("hello")); ซึ่ง #A แทนด้วยข้อความ แต่ (A) คือแอดเดรสของข้อความนั้น	typedef ชนิดข้อมูล ชื่อเรียกใหม่; สร้างชื่อเรียกใหม่ ไม่เรียกชนิดข้อมูลที่มียอยู่แล้ว
ชนิดข้อมูล ชื่อฟังก์ชัน (อาร์กิวเมนต์ ₁ , ...)	การนิยามฟังก์ชัน Function Definition การประกาศตัวแปรเพื่อใช้ในฟังก์ชัน (local variable definition)	#ifndef ชื่อการ์ด การ์ดมาโคร (Guarded macro) #define ชื่อการ์ด ใช้ประกาศใน include file (file.h) เพื่อกันไม่ให้ include file ตัวนั้น ถูกนำไป Include หลายครั้ง	sizeof ชื่อตัวแปร ขนาดของตัวแปรในหน่วยไบต์ sizeof (ชนิดข้อมูล) ขนาดของชนิดข้อมูลในหน่วยไบต์
ชนิดข้อมูล ₁ ชื่ออาร์กิวเมนต์ ₁ , ...	การประกาศอาร์กิวเมนต์ หากมีหลายตัวคั่นด้วย comma	#endif	static ชนิดข้อมูล ชื่อตัวแปร; ประกาศให้ตัวแปรนั้นอยู่ใน file scope
exit (รหัสสถานะการจบโปรแกรม);	คำสั่งให้โปรแกรมจบทันที		

โอเปอเรเตอร์ (Operators)				กลุ่ม	โอเปอเรเตอร์	ความหมาย	Associativity	การควบคุมลำดับการทำงาน (Flow of Control)	
1	postfix	โอเปอเรเตอร์แบบเติมหลัง		13	Ternary conditional		ขวาไปซ้าย	; สิ้นสุดประโยคคำสั่ง (statement terminator) { } ตัวคั่นบล็อก, บอกขอบเขตบล็อก break หลุดออกจาก switch, while, do, for continue บังคับให้วนรอบ ใน while, do และ goto ฉลาก; กระโดดไปยังคำสั่งที่ตำแหน่งปลายทาง ฉลาก: ประโยคคำสั่ง การระบุตำแหน่งปลายทาง (label) return นิพจน์; นำค่าจะนิพจน์ส่งคืนกลับให้ผู้เรียก ประโยคคำสั่งที่ใช้ควบคุมลำดับการทำงาน (Flow construction) โครงสร้างที่ควบคุมด้วยการตัดสินใจ (Decision case structure)	
		วงเล็บกลม (เรียกฟังก์ชัน)			? :				
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			Assignment โฆษ ใช้กำหนดค่าให้ตัวแปร		ขวาไปซ้าย		
		จุด (เข้าถึงสมาชิกของสตรัค)			= นำค่าด้านขวามาใส่ให้ด้านซ้าย				
		->			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
2	unary	โอเปรข้างเดียวแบบใช้นำหน้า		14	+= -= นำค่าทางขวามาคคูณ/หารแล้วใส่		ขวาไปซ้าย	&= &= ทวนเอาเศษ/AND ก่อนใส่ค่า ^= = XOR/OR บิตก่อนใส่ค่า <<= >>= เลื่อนทั้งทางซ้าย/ขวาก่อนใส่ค่า	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
3	unary	โอเปรข้างเดียวแบบใช้นำหน้า		15	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
4	unary	โอเปรข้างเดียวแบบใช้นำหน้า		16	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
5	unary	โอเปรข้างเดียวแบบใช้นำหน้า		17	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
6	unary	โอเปรข้างเดียวแบบใช้นำหน้า		18	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
7	unary	โอเปรข้างเดียวแบบใช้นำหน้า		19	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
8	unary	โอเปรข้างเดียวแบบใช้นำหน้า		20	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
9	unary	โอเปรข้างเดียวแบบใช้นำหน้า		21	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
10	unary	โอเปรข้างเดียวแบบใช้นำหน้า		22	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
11	unary	โอเปรข้างเดียวแบบใช้นำหน้า		23	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
12	unary	โอเปรข้างเดียวแบบใช้นำหน้า		24	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
13	unary	โอเปรข้างเดียวแบบใช้นำหน้า		25	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
14	unary	โอเปรข้างเดียวแบบใช้นำหน้า		26	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
15	unary	โอเปรข้างเดียวแบบใช้นำหน้า		27	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
16	unary	โอเปรข้างเดียวแบบใช้นำหน้า		28	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
17	unary	โอเปรข้างเดียวแบบใช้นำหน้า		29	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
18	unary	โอเปรข้างเดียวแบบใช้นำหน้า		30	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
19	unary	โอเปรข้างเดียวแบบใช้นำหน้า		31	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
20	unary	โอเปรข้างเดียวแบบใช้นำหน้า		32	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
21	unary	โอเปรข้างเดียวแบบใช้นำหน้า		33	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		->			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		++ --			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
		+= -=			*= /= นำค่าทางขวามาคคูณ/หารแล้วใส่				
22	unary	โอเปรข้างเดียวแบบใช้นำหน้า		34	Comma โอเปอเรเตอร์คอมม่า		ซ้ายไปขวา	if (expression) การตัดสินใจแบบเลือกทำ ทำ statement statement เมื่อ expression=True if (expression) การตัดสินใจแบบสองทาง ถ้า statement1 expression เป็นจริง ทำ else statement1 ไม่เช่นนั้น (else) ทำ statement2 statement2 if (expression1) การตัดสินใจแบบนำตก (Casted if) statement1 สองชั้น ถ้า exp1 เป็นจริงทำ statm1 else ไม่เช่นนั้น ถ้า exp1 เป็นเท็จ เจ็ด if (expression2) exp2 ถ้าเป็นจริงทำ statemnt2 ไม่ statement2 จัง ทำ sttmnt3 (default case) else statement3 switch (expression) การเลือกทำประโยคคำสั่ง ตามค่าของ { expression ในวงเล็บกลมหลัง case val1: switch ว่าตรงกับ val ในกรณี statement1 (case) ไດ หากไม่ตรงกับกรณีใดเลย break; จะทำประโยคคำสั่งที่อยู่ใกรณี case val2: default statement2 } break; default: statement3 default: statement3 }	
		วงเล็บเหลี่ยม (ช่องในอาร์เรย์)			+= -= นำค่าทางขวามาคบวก/ลบแล้วใส่				
		จุด (เข้าถึงสมาชิกของสตรัค)							

<p>การควบคุมลำดับการทำงาน (Flow of Control) ต่อ</p> <p>โครงสร้างการวนรอบหรือลูป (Loop)</p> <p>while (expression) statement ลูป while หาก expression เป็นจริง เข้าลูปทำ statement ซ้ำ และกลับมาเช็คเงื่อนไขใหม่ จนกว่า exp. จะเป็นเท็จ</p> <p>int x = 8; while (x>0) x--; ตัวอย่าง ประกาศตัวแปร x มีค่าเป็น 8 ระหว่างที่ (while) x ยังมีค่ามากกว่า 0 ลดค่า x ลงหนึ่ง วนรอบใหม่ (8 รอบ)</p> <p>do statement while (expression); do..while loop ทำประโยคคำสั่งหนึ่งรอบแล้วหาก expression ยังคงเป็นจริง วนเข้าลูปใหม่ จนกว่าจะเป็นเท็จ</p> <p>int x = 8; do x--; while (x>0); ตัวอย่าง ประกาศตัวแปร x มีค่า 8 เข้าลูป เพื่อลดค่า x ลงหนึ่ง จากนั้นถ้า x ยังมากกว่า 0 วนลูปไปจนกว่า x จะ <= 0</p> <p>for (initial; condition; update) statement เริ่มจากตั้งค่าตัวนับ แล้วทำประโยคคำสั่ง จากนั้นเลื่อนตัวนับ และวนรอบ หากเงื่อนไขเป็นจริง คำสั่งตัวอย่างจะพิมพ์ค่า i ตั้งแต่ 0 ถึง 7</p> <p>int i; for (i=0; i<8; i++) printf("i = %d\n", i);</p>	<p>การเก็บข้อมูลหลายชนิดในโครงสร้างเดียวกัน (Structure)</p> <p>struct ชื่อเรียก{ ประกาศชนิดข้อมูลใหม่เป็นสตริงค์ การประกาศตัวแปร }; struct ชื่อเรียก ชื่อตัวแปร; ชื่อตัวแปร . ชื่อสมาชิก ชื่อพอยเตอร์->ชื่อสมาชิก ตัวอย่าง (*p) . x หรือ p->x มีความหมายเดียวกัน</p> <p>การเก็บข้อมูลหลายชนิดโดยใช้เนื้อที่เก็บเดียวกัน (Union)</p> <p>union ชื่อเรียก{ ประกาศรายการตัวแปรที่จะใช้ที่เก็บเดียวกัน }; union ชื่อเรียก ชื่อตัวแปร; ชื่อตัวแปร . ชื่อสมาชิก ชื่อพอยเตอร์->ชื่อสมาชิก (*ชื่อพอยเตอร์) . ชื่อสมาชิก</p> <p>การประกาศสมาชิกของ struct และ union แบบกำหนดจำนวนบิต</p> <p>unsigned ชื่อสมาชิก : จำนวนบิต; สมาชิกมีจำนวนบิตที่ระบุ เช่น unsigned c:8; สมาชิกชื่อ c จะมีขนาดเก็บ 8 บิต</p>	<p>การรับข้อมูลเข้าและแสดงผล <stdio.h></p> <p>อินพุตจากคีย์บอร์ดและเอาต์พุตที่จอภาพ (Standard Input/Output หรือ Standard I/O)</p> <p>stdin สตริงที่ใช้เก็บข้อมูลจากคีย์บอร์ด stdout สตริงที่ใช้เก็บข้อมูลที่จะแสดงผลที่จอภาพ stderr สตริงที่ใช้เก็บ error ของโปรแกรม EOF สัญลัษณ์บอกจุดสิ้นสุดไฟล์ (เป็น int) getchar() อ่านตัวอักษรเข้ามาหนึ่งตัว รีเทิร์นให้ผู้ใช้เรียก putchar (ตัวอักษร) แสดงผลตัวอักษรไปที่จอภาพ puts (ข้อความ) แสดงผลข้อความไปที่จอภาพ</p> <p>printf ("รูปแบบ", ค่า1, ...) แสดงข้อมูลออกจอภาพ sprintf (s, "รูปแบบ", ค่า1, ...) พิมพ์ข้อมูลใส่สตริง s scanf ("รูปแบบ", &ตัวแปร1, ...) อ่านข้อมูลจากคีย์บอร์ด sscanf (s, "รูปแบบ", &ตัวแปร1, ...) นำข้อมูลจากคีย์บอร์ดใส่ให้สตริง s</p> <p>อินพุตและเอาต์พุตจากไฟล์ หรือเพิ่มข้อมูล (File Input/Output หรือ File I/O)</p> <p>FILE *fp; ประกาศตัวชี้ไฟล์ fp fopen ("ชื่อไฟล์", "โหมด: r(read), w(write), a(ต่อท้าย), b(binary)) เปิดไฟล์ด้วยโหมดนั้นส่งตัวชี้ไฟล์กลับ getc (fp) อ่านอักขระจากไฟล์ที่ชี้ด้วย fp putc (ตัวอักษร, fp) เขียนอักขระลงไฟล์ที่ชี้ด้วย fp</p> <p>fprintf (fp, "รูปแบบ", ค่า1, ...) พิมพ์ข้อมูลลงไฟล์ fp fscanf (fp, "รูปแบบ", ตัวแปร1, ...) อ่านข้อมูลจากไฟล์ fp fread (*ptr, eltsize, n, fp) อ่านเรคอร์ดขนาด eltsize จำนวน n เรคอร์ดจากไฟล์ที่ชี้ด้วย fp มาใส่ที่ ptr ชื่ออยู่ fwrite (*ptr, eltsize, n, fp) เขียนเรคอร์ดขนาด eltsize จำนวน n เรคอร์ดเนื้อหาที่ ptr ชื่ออยู่ลงในไฟล์ที่ fp ชื่ออยู่ fclose (fp) ปิดไฟล์ที่ fp ชื่ออยู่ feof (fp) ทดสอบว่าไฟล์ที่ fp ชื่ออยู่ ถึงจุดสิ้นสุดไฟล์หรือยัง ถ้าถึงแล้วรีเทิร์นค่าที่ไม่ใช่ 0</p> <p>fgets (s, max, fp) อ่านบรรทัดจากไฟล์มาใส่ s ขนาดไม่เกิน max fputs (s, fp) เขียนสตริง s ลงไฟล์ที่ fp ชื่ออยู่ ferror (fp) ทดสอบว่าไฟล์ที่ fp ชื่ออยู่ มีข้อผิดพลาดไหม ถ้ามี ferror จะรีเทิร์นค่าที่ไม่ใช่ 0</p>															
<p>พอยเตอร์ และอาร์เรย์ (Pointers and Array)</p> <p>ชนิดข้อมูล *ชื่อพอยเตอร์; ประกาศพอยเตอร์ชี้ไปยังชนิดข้อมูล void * ชื่อพอยเตอร์; ประกาศพอยเตอร์แบบไม่ระบุชนิด ชนิดข้อมูล *ชื่อฟังก์ชัน (); ประกาศฟังก์ชันที่ส่งพอยเตอร์กลับ ชนิดข้อมูล (*ชื่อพอยเตอร์) (); ประกาศพอยเตอร์ที่ชี้ไปยังฟังก์ชัน NULL ค่าพอยเตอร์เปล่า ไม่ชี้ไปที่ใดเลย *ชื่อพอยเตอร์ เข้าไปยังข้อมูลที่พอยเตอร์ชี้อยู่ &ชื่อตัวแปร แอดเดรสของตัวแปร ชนิดข้อมูล ชื่ออาร์เรย์ [ขนาด]; ประกาศสร้างอาร์เรย์หนึ่งมิติ ชนิดข้อมูล ชื่ออาร์เรย์ [][...][...]; ประกาศสร้างอาร์เรย์หลายมิติ ชื่ออาร์เรย์ แอดเดรสเริ่มต้นของตัวแปรอาร์เรย์ ชื่ออาร์เรย์ [ตำแหน่ง] เข้าไปยังข้อมูลในอาร์เรย์ช่องนั้น อาร์เรย์ในภาษา C เริ่มตำแหน่งที่ 0 จนถึง ขนาด-1 int a [4]; มี 4 ช่อง a[0], a[1], a[2], a[3]</p>	<p>ไลบรารีมาตรฐานของภาษาซี ที่สถาบันแอนไพกำหนดให้ทุกคอมพิวเตอร์ ต้องมีให้ผู้ใช้เขียนโปรแกรมใช้ (ANSI C Standard Library)</p> <table border="1"> <tr> <td><assert.h></td> <td><ctype.h></td> <td><errno.h></td> </tr> <tr> <td><float.h></td> <td><limits.h></td> <td><locale.h></td> </tr> <tr> <td><math.h></td> <td><setjmp.h></td> <td><signal.h></td> </tr> <tr> <td><stdarg.h></td> <td><stddef.h></td> <td><stdio.h></td> </tr> <tr> <td><stdlib.h></td> <td><string.h></td> <td><time.h></td> </tr> </table> <p>เมื่อนำเฮดเดอร์ไฟล์เหล่านั้นมารวมในโปรแกรมต้นฉบับ โปรแกรมเมอร์สามารถเรียกใช้ฟังก์ชันสำเร็จรูปที่มีให้ได้เลย เนื่องจากฟังก์ชันเหล่านั้นอยู่ใน C Standard Library (cstdlib) ที่ Link ให้อยู่แล้ว</p>	<assert.h>	<ctype.h>	<errno.h>	<float.h>	<limits.h>	<locale.h>	<math.h>	<setjmp.h>	<signal.h>	<stdarg.h>	<stddef.h>	<stdio.h>	<stdlib.h>	<string.h>	<time.h>	
<assert.h>	<ctype.h>	<errno.h>															
<float.h>	<limits.h>	<locale.h>															
<math.h>	<setjmp.h>	<signal.h>															
<stdarg.h>	<stddef.h>	<stdio.h>															
<stdlib.h>	<string.h>	<time.h>															

<div><stdio.h> ต่อ</div> <div>รหัสสำหรับระบุรูปแบบการรับ/แสดงผล</div> <div><div>%-+ 0w.pmc"</div><div>- เครื่องหมายลบ บังคับขีดซ้าย</div><div>+ ถ้าเป็นจำนวนบวกพิมพ์ + จำนวนลบพิมพ์ - นำ</div><div>ช่องว่าง พิมพ์ช่องว่างหากข้อมูลเป็นจำนวนบวก</div><div>0 เติมเลข 0 นำหน้าจำนวนให้เต็มหลัก</div><div>w จำนวนหลักทั้งหมด รวมจำนวนหลังจุดทศนิยม</div><div>p จำนวนหลักหลังจุดทศนิยม (ความละเอียด)</div><div>m รหัสแปลงข้อมูลก่อนพิมพ์</div><div>h=short, l=long, L=long double</div><div>c รหัสบอกชนิดข้อมูล</div><div><div><div>d, i integer u unsigned</div><div>c character s string</div><div>f double (ใน printf) e,E exponential</div><div>f float (ใน scanf) lf double(scanf)</div><div>o Octal (ฐานแปด) x,X hex(ฐานสิบหก)</div><div>p พอยเตอร์ n จำนวนตัวอักษร</div></div><div>g, G เหมือน f หรือ e,E ขึ้นอยู่กับจำนวน exponent</div></div></div>	<div>ฟังก์ชันอรรถประโยชน์ <stdlib.h></div> <div>(Standard Utilites Functions)</div> <div><div><div>abs(n) หาค่าสัมบูรณ์ของ int n</div><div>labs(n) หาค่าสัมบูรณ์ของ long n</div><div>div(n,d) หาผลหารของ int n หาร d</div><div>ส่งผลลัพธ์เป็น struct ที่ div_t.quot คือผลหาร div_t.rem คือเศษ</div><div>ldiv(n,d) หาผลหารของ long n หาร d</div><div>ส่งผลลัพธ์เป็น struct ที่ ldiv_t.quot คือผลหาร ldiv_t.rem คือเศษ</div><div>rand() การสร้างสุ่มตัวเลข int ขึ้นมา</div><div>srand(n) กำหนด n เป็น random seed</div><div>exit(status) จบโปรแกรม บอกเลข status</div><div>system(s) ส่ง OS ให้ทำตามข้อความ s</div></div><div>แปลงชนิดข้อมูล</div><div><div><div>atof(s) แปลงสตริง s เป็น double</div><div>atoi(s) แปลงสตริง s เป็น integer</div><div>atol(s) แปลงสตริง s เป็น long</div><div>strtod(s, &endp) แปลงส่วนหน้าของสตริง s เป็น double แล้วชี้ endp ไปที่อักขระตัวแรกหลังเลขนั้น</div><div>strtol(s, &endp,b) แปลงส่วนหน้าของสตริง s เป็นเลขจำนวนเต็มฐาน b แบบ long แล้วชี้ endp ไปที่อักขระแรกท้ายเลขนั้น</div><div>strtoul(s, &endp,b) เหมือนกันแต่แปลงเป็นunsigned</div></div><div>การจองพื้นที่หน่วยความจำเพื่อใช้งาน</div><div><div><div>malloc(size) จองที่เมมฯขนาด size ไบต์ส่ง addr กลับ</div><div>calloc(n,size) จองที่เมมฯ n บล็อก ละ size ไบต์</div><div>newptr=realloc(ptr,size);</div><div>เปลี่ยนขนาดเนื้อที่เมมฯ ที่ ptr ซ้ำอยู่ เป็น size ไบต์ ส่ง addr กลับ</div><div>free(ptr); คืนที่เมมฯ ที่ ptr ซ้ำอยู่</div></div><div>ฟังก์ชันจัดการอาร์เรย์</div><div><div><div>bsearch(key,array,n,size,cmpf) ค้นหา key จาก array ที่มี n ช่องๆ ละ size ไบต์ ใช้ฟังก์ชัน cmpf เป็นตัวเทียบ</div><div>qsort(array,n,size,cmpf) เรียงข้อมูลใน array</div></div></div></div></div></div>	<div>การทดสอบชนิดของ character <ctype.h></div> <div><div><div>isalnum(c) True ถ้า c เป็นอักขระตัวเลข false ถ้าไม่ใช่</div><div>isalpha(c) True ถ้า c เป็นตัวอักษร false ถ้าไม่ใช่</div><div>iscntrl(c) ทดสอบว่า c เป็น Control Character หรือไม่</div><div>isdigit(c) ทดสอบว่า c เป็นเลขโคตฐานสิบหรือไม่</div><div>isgraph(c) c เป็นอักขระที่มองเห็นได้หรือไม่ (ไม่รวม space)</div><div>islower(c) c เป็นตัวพิมพ์เล็กหรือไม่</div><div>isprint(c) c เป็นอักขระที่มองเห็นได้หรือไม่ (รวม space)</div><div>ispunct(c) c เป็นอักขระที่ไม่ใช่ space, ตัวเลข, ตัวอักษร</div><div>isspace(c) c เป็น space, formfeed, /n, /r, /t, /vt?</div><div>isupper(c) c เป็นตัวพิมพ์ใหญ่ หรือไม่</div><div>isxdigit(c) c เป็นเลขโคตฐานสิบหกหรือไม่</div><div>tolower(c) แปลง c เป็นตัวพิมพ์เล็ก</div><div>toupper(c) แปลง c เป็นตัวพิมพ์ใหญ่</div></div><div>การจัดการข้อความหรือสตริง <string.h></div><div>(String operations)</div><div><div><div>strlen(s) ส่งค่าความยาวของสตริง s</div><div>strcpy(s,ct) ก๊อปปี้ ct ใส่ s</div><div>strcat(s,ct) นำ ct มาต่อท้าย s</div><div>strcmp(cs,ct) เทียบ cs กับ ct เท่ากันรีเทิร์น 0</div><div>strncmp(cs,ct,n) เทียบ cs กับ ct แค่ n อักขระแรก</div><div>strchr(cs,c) รีเทิร์นพอยเตอร์ที่ชี้ไปยังอักขระแรกของสตริง c ที่ประกอบในสตริง cs รีเทิร์น NULL ถ้าไม่พบ</div><div>strrchr(cs,c) รีเทิร์นพอยเตอร์ที่ชี้ไปยังอักขระสุดท้ายของสตริง c ที่ประกอบอยู่ในสตริง cs รีเทิร์น NULL ถ้าไม่พบ</div><div>memcpy(s,ct,n) ก๊อปปี้อักขระ n ตัวจาก ct ไปใส่ที่ s</div><div>memmove(s,ct,n) ก๊อปปี้อักขระ n ตัวจาก ct ไปใส่ที่ s (อาจจะเกิดการซ้อนทับ)</div><div>memcmp(cs,ct,n) เปรียบเทียบอักขระ n ตัวระหว่าง cs กับ ct</div><div>memchr(cs,c,n) รีเทิร์นพอยเตอร์ที่ชี้ไปยังสตริง c ที่อยู่ใน n อักขระแรก ของสตริง cs</div><div>memset(s,c,n) กำหนดค่าให้ n ตัวแรก ของสตริง s มีค่าเป็นอักขระ c</div></div></div></div>
<div>ฟังก์ชันคำนวณทางคณิตศาสตร์<math.h></div> <div>อาร์กิวเมนต์และค่าผลลัพธ์ที่ส่งกลับ เป็น double</div> <div><div><div>sin(x),cos(x),tan(x) ฟังก์ชันคำนวณตรีโกณฯ</div><div>asin(x),acos(x),atan(x) ตรีโกณฯย้อนกลับ</div><div>atan2(y,x) ค่า arctan(y/x)</div><div>sinh(x),cosh(x),tanh(x) hyperbolic ตรีโกณฯ</div><div>exp(x),log(x),log10(x) ยกกำลังและถอดลอกล็อก</div><div>ldexp(x,n), frexp(x,&e) ยกกำลังในฐานสอง</div><div>modf(x,ip), fmod(x,y) การหารและการหาเศษ</div><div>pow(x,y) x ยกกำลัง y</div><div>sqrt(x) รากที่สองของ x</div><div>ceil(x) ปัดเศษขึ้น</div><div>floor(x) ปัดเศษลง</div><div>fabs(x) หาค่า absolute ของ x</div></div></div>		

ฟังก์ชันและข้อมูลเกี่ยวกับวันและเวลา <time.h>

ชนิดข้อมูลสำหรับเก็บเวลาและวันปฏิทิน

clock_t	ชนิดข้อมูลที่เก็บจังหวะสัญญาณนาฬิกา
time_t	ชนิดข้อมูลที่เก็บเวลาปฏิทิน
struct tm	ชนิดข้อมูลสตริงที่เก็บเวลาปฏิทิน มีสมาชิกคือ
tm_sec	วินาที (จำนวนวินาที เมื่อหักนาทีออกแล้ว)
tm_min	นาที (จำนวนนาทีที่เหลือ เมื่อหักชั่วโมงแล้ว)
tm_hour	ชั่วโมง (จำนวนชั่วโมงนับจากเที่ยงคืน)
tm_mday	วันที่ นับจากวันเริ่มต้นของเดือนเป็นวันที่ 1
tm_mon	เดือน นับจากเดือนมกราคมเป็นเดือนที่ 1
tm_year	ปี เริ่มนับจากปี 1900
tm_wday	วันในสัปดาห์ นับจากวันอาทิตย์เป็นวันที่ 1
tm_yday	วันในปี นับจากวันที่ 1 ม.ค.เป็นวันแรก
tm_isdst	กำหนดให้ปรับเวลาออมแสง (day light saving time) True=ปรับเวลาออมแสง

ฟังก์ชันเกี่ยวกับวันและเวลา (Date and time functions)

clock()	ใช้เก็บ time stamp จากซีพียู ณ จุดที่เรียก ฟังก์ชันวัดเวลาของซีพียู เป็นจำนวน clock ticks หากจะอ่านเป็นวินาทีจะต้องส่ง ต.ย. <code>clock()/CLOCK_PER_SEC</code>
time()	วัดวินาทีเวลาปฏิทิน (วัน,เดือน,ปี)
difftime(t2, t1)	วัดวินาทีว่า t2 ต่างจาก t1 กี่วินาที (t2-t1) ผลลัพธ์ต่างมีชนิดเป็น double
mktime(tp)	แปลง tp ซึ่งเป็น struct tm ให้เป็นชนิด time_t วัดวินาทีข้อมูลชนิด time_t กลับ
asctime(tp)	แปลง tp ซึ่งเป็น struct tm เป็นสตริง (char *) วัดวินาทีสตริงกลับ ในรูปแบบ Www Mmm dd hh:mm:ss yyyy เช่น MON JUN 20 13:17:18 2010
ctime(tp)	แปลง tp ชนิด time_t เป็นสตริง
gmtime(tp)	แปลง tp ชนิด time_t เป็นสตริง tm ที่เก็บ เวลาสากลเชิงพิกัด (UTC หรือ GMT)
localtime(tp)	แปลง tp ชนิด time_t เป็นสตริง tm
strftime(s, smax, "รูปแบบ", tp)	แปลง tp ชนิดสตริง tm เป็นสตริง ตามรูปแบบที่กำหนด บันทึกสตริงลงใน s ขนาดไม่เกิน smax ไบต์

ขอบเขตการเก็บข้อมูลชนิด Integer <limits.h>

ตัวเลขในวงเล็บเป็นค่าที่มักจะใช้กำหนดเป็นค่าคงที่ ในระบบยูนิกซ์ 32 บิต

CHAR_BIT	จำนวนบิตที่ใช้แทน 1 char	(8)
CHAR_MAX	ค่าสูงสุดที่เก็บได้ในชนิด char	(SCHAR_MAX หรือ UCHAR_MAX)
CHAR_MIN	ค่าต่ำสุดที่เก็บได้ในชนิด char	SCHAR_MIN หรือ 0
SCHAR_MAX	ค่าสูงสุดของ signed char	(+127)
SCHAR_MIN	ค่าต่ำสุดของ signed char	(-128)
SHRT_MAX	ค่าสูงสุดที่เก็บได้ใน short	(+32,767)
SHRT_MIN	ค่าต่ำสุดที่เก็บได้ใน short	(-32,768)
INT_MAX	ค่าสูงสุดที่เก็บได้ใน int	(+2,147,483,647)
INT_MIN	ค่าต่ำสุดที่เก็บได้ใน int	(-2,147,483,648)
LONG_MAX	ค่าสูงสุดที่เก็บได้ใน long	(+2,147,483,647)
LONG_MIN	ค่าต่ำสุดที่เก็บได้ใน long	(-2,147,483,648)
UCHAR_MAX	ค่าสูงสุดของ unsigned char	(255)
USHRT_MAX	ค่าสูงสุดของ unsigned short	(65,535)
UINT_MAX	ค่าสูงสุดของ unsigned int	(4,294,967,295) (65535)
ULONG_MAX	ค่าสูงสุดของ unsigned long	(4,294,967,295)

ขอบเขตการเก็บข้อมูลชนิด Float <float.h>

ตัวเลขในวงเล็บเป็นค่าที่มักจะใช้กำหนดเป็นค่าคงที่ ในระบบยูนิกซ์ 32 บิต

FLT_RADIX	ฐานของเลขยกกำลัง	(2)
FLT_ROUNDS	กำหนดให้ทำงานในโหมดพิเศษค่า Float	
FLT_DIG	ความละเอียดจุดทศนิยมของ float	(6)
FLT_EPSILON	ค่า x ต่ำสุดที่ทำให้ $1.0f+x \neq 1.0f$ (ระยะห่างจาก 1 ที่น้อยที่สุด)	($1.1E-7$)
FLT_MANT_DIG	จำนวนบิตในแมนทิซซา (Mantissa)	
FLT_MAX	ค่าสูงสุดที่เก็บได้ใน float	($3.4E38$)
FLT_MAX_EXP	ค่าสูงสุดของเลขชี้กำลัง (Exponent) ใน float	
FLT_MIN	ค่าต่ำสุดที่เก็บได้ใน float	($1.2E-38$)
FLT_MIN_EXP	ค่าต่ำสุดของเลขชี้กำลัง (Exponent) ใน float	
DBL_DIG	ความละเอียดจุดทศนิยมของ double	(12)
DBL_EPSILON	ค่า x ต่ำสุดที่ทำให้ $1.0+x \neq 1.0$ (ระยะห่างจาก 1 ที่น้อยที่สุด)	($2.2E-16$)
DBL_MANT_DIG	จำนวนบิตในแมนทิซซา (Mantissa) ของ dbl	
DBL_MAX	ค่าสูงสุดที่เก็บได้ใน double	($1.8E308$)
DBL_MAX_EXP	ค่าสูงสุดของเลขชี้กำลัง (Exponent) ใน dbl	
DBL_MIN	ค่าต่ำสุดที่เก็บได้ใน double	($2.2E-308$)
DBL_MIN_EXP	ค่าต่ำสุดของเลขชี้กำลัง (Exponent) ใน dbl	

การจัดการตัวแปรอาร์กิวเมนต์ <stdarg.h> (Variable Argument Lists)

va_list	ชื่อพอยเตอร์ที่ไปยังอาร์; ประกาศพอยเตอร์ไปยังอาร์กิว	หากมีการใช้ va_start เมื่อใช้เสร็จต้องเคลียร์พอยเตอร์
va_start	(พอยเตอร์ชี้อาร์กิวเมนต์, ชื่ออาร์กิวตัวสุดท้าย); กำหนดค่าเริ่มต้นให้พอยเตอร์ชี้อาร์กิวเมนต์	ที่กำหนดได้ด้วยฟังก์ชัน va_end
va_arg	(พอยเตอร์ชี้อาร์กิวเมนต์, ชนิดของอาร์กิวเมนต์นั้น); เข้าถึงค่าของอาร์กิวเมนต์ผ่านพอยเตอร์ โดยระบุชนิด	va_end (พอยเตอร์ชี้อาร์กิวเมนต์); เคลียร์ค่าของพอยเตอร์ชี้อาร์กิวเมนต์ที่เคยกำหนดด้วย va_start ไว้

คำสงวน (Keywords หรือ Reserve words) อาจเป็นคำหนึ่งคำใด ใน 36 คำดังนี้

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	inline	i_startnt
long	register	restrict	return	short	signed
sizeof	static	struct	switch	typedef	union
unsigned	void	volatile	while	_Bool	_Complex

ตารางรหัสแอสกี (ASCII - American Standard Code for Information Interchange)

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	!	64	40	@	96	60	'	128	80	Ç	160	A0	ā	192	C0	Ł	224	E0	α	256	100	Δ
^A	1	01		SOH	33	21	!	65	41	A	97	61	a	129	81	ü	161	A1	ī	193	C1	ł	225	E1	Β	257	101	Ε
^B	2	02		STX	34	22	"	66	42	B	98	62	b	130	82	ë	162	A2	ō	194	C2	Ł	226	E2	Γ	258	102	Θ
^C	3	03		ETX	35	23	#	67	43	C	99	63	c	131	83	ä	163	A3	ó	195	C3	ł	227	E3	Π	259	103	Ξ
^D	4	04		EOT	36	24	\$	68	44	D	100	64	d	132	84	å	164	A4	ñ	196	C4	—	228	E4	Σ	260	104	ς
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e	133	85	ä	165	A5	ñ	197	C5	+	229	E5	σ	261	105	μ
^F	6	06		ACK	38	26	&	70	46	F	102	66	f	134	86	ä	166	A6	ä	198	C6	+	230	E6	μ	262	106	υ
^G	7	07		BEL	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	o	199	C7	+	231	E7	Υ	263	107	ϕ
^H	8	08		BS	40	28	(72	48	H	104	68	h	136	88	ê	168	A8	ô	200	C8	+	232	E8	ϕ	264	108	ω
^I	9	09		HT	41	29)	73	49	I	105	69	i	137	89	ë	169	A9	·	201	C9	+	233	E9	θ	265	109	Ω
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j	138	8A	ë	170	AA	·	202	CA	+	234	EA	Ω	266	110	δ
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k	139	8B	ë	171	AB	½	203	CB	+	235	EB	δ	267	111	ϕ
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l	140	8C	ë	172	AC	¼	204	CC	+	236	EC	φ	268	112	φ
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m	141	8D	ë	173	AD	ı	205	CD	=	237	ED	Φ	269	113	Φ
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n	142	8E	ë	174	AE	«	206	CE	+	238	EE	ε	270	114	ε
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o	143	8F	ë	175	AF	»	207	CF	+	239	EF	ε	271	115	ε
^P	16	10		DLE	48	30	0	80	50	P	112	70	p	144	90	ë	176	B0	·	208	D0	+	240	F0	≡	272	116	≡
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q	145	91	ë	177	B1	·	209	D1	+	241	F1	±	273	117	±
^R	18	12		DC2	50	32	2	82	52	R	114	72	r	146	92	ë	178	B2	·	210	D2	+	242	F2	±	274	118	±
^S	19	13		DC3	51	33	3	83	53	S	115	73	s	147	93	ë	179	B3	·	211	D3	+	243	F3	±	275	119	±
^T	20	14		DC4	52	34	4	84	54	T	116	74	t	148	94	ë	180	B4	·	212	D4	+	244	F4	±	276	120	±
^U	21	15		NAK	53	35	5	85	55	U	117	75	u	149	95	ë	181	B5	·	213	D5	+	245	F5	±	277	121	±
^V	22	16		SYN	54	36	6	86	56	V	118	76	v	150	96	ë	182	B6	·	214	D6	+	246	F6	±	278	122	±
^W	23	17		ETB	55	37	7	87	57	W	119	77	w	151	97	ë	183	B7	·	215	D7	+	247	F7	±	279	123	±
^X	24	18		CAN	56	38	8	88	58	X	120	78	x	152	98	ë	184	B8	·	216	D8	+	248	F8	±	280	124	±
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y	153	99	ë	185	B9	·	217	D9	+	249	F9	±	281	125	±
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z	154	9A	ë	186	BA	·	218	DA	+	250	FA	±	282	126	±
^[27	1B		ESC	59	3B	:	91	5B	[123	7B	{	155	9B	ë	187	BB	·	219	DB	+	251	FB	±	283	127	±
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	}	156	9C	ë	188	BC	·	220	DC	+	252	FC	±	284	128	±
^]	29	1D		GS	61	3D	=	93	5D]	125	7D	}	157	9D	ë	189	BD	·	221	DD	+	253	FD	±	285	129	±
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~	158	9E	ë	190	BE	·	222	DE	+	254	FE	±	286	130	±
^-	31	1F	▼	US	63	3F	?	95	5F	—	127	7F	ÿ	159	9F	ë	191	BF	·	223	DF	+	255	FF	±	287	131	±

* ASCII code 127 has the code DEL. Under MS-DOS, this code has the same effect as ASCII 8 (BS). The DEL code can be generated by the CTRL + BKSP key.

[http://i.msdn.microsoft.com/9hxt0028.Local_-567447678_vc38qp1\(en-US,VS.80\).gif](http://i.msdn.microsoft.com/9hxt0028.Local_-567447678_vc38qp1(en-US,VS.80).gif)

[http://i.msdn.microsoft.com/60ecse8t.Local_-567447811_vc38qp1\(en-US,VS.80\).gif](http://i.msdn.microsoft.com/60ecse8t.Local_-567447811_vc38qp1(en-US,VS.80).gif)