

ภาษาซีและ Arduino



C & Arduino UNO

อ่านง่าย เข้าใจ ประยุกต์ใช้ได้จริง



สำนักพิมพ์ ส.ส.ท.
สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

ผู้ช่วยศาสตราจารย์ดอนสัน ปงผาบ

ภาษาซีและ Arduino



โดย... ผู้ช่วยศาสตราจารย์ดอนสัน ปงผาบ

ราคา 295 บาท

พิมพ์ครั้งที่ 1 กันยายน 2560 จำนวนพิมพ์ 2,000 เล่ม

ข้อมูลทางบรรณานุกรมของสำนักหอสมุดแห่งชาติ

ดอนสัน ปงผาบ.

ภาษาซีและ Arduino. -- กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2560.

216 หน้า.

1. ไมโครคอนโทรลเลอร์.

2. ซี (ภาษาคอมพิวเตอร์).

I. ชื่อเรื่อง.

629.89551

ISBN 978-974-443-715-0

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ (ฉบับเพิ่มเติม) พ.ศ. 2558 โดย สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใด ๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษร

จัดพิมพ์โดย

สำนักพิมพ์ ส.ส.ท.

สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

5-7 ซอยสุขุมวิท 29 ถนนสุขุมวิท แขวงคลองเตยเหนือ เขตวัฒนา กรุงเทพฯ 10110

โทร. 0-2258-0320 (6 เลขหมายอัตโนมัติ), 0-2259-9160 (10 เลขหมายอัตโนมัติ)

ติดต่อสำนักพิมพ์ book4u@tpa.or.th

ติดต่อสั่งซื้อหนังสือ www.tpabook.com

จัดจำหน่ายโดย บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)

เลขที่ 1858/87-90 ถนนบางนา-ตราด

แขวงบางนา เขตบางนา กรุงเทพฯ 10260

โทร. 0-2739-8000, 0-2739-8222 โทรสาร 0-2739-8356-9

www.se-ed.com



สมาคม รักชาติโลก

ร่วมใช้หมึกพิมพ์จากถั่วเหลือง

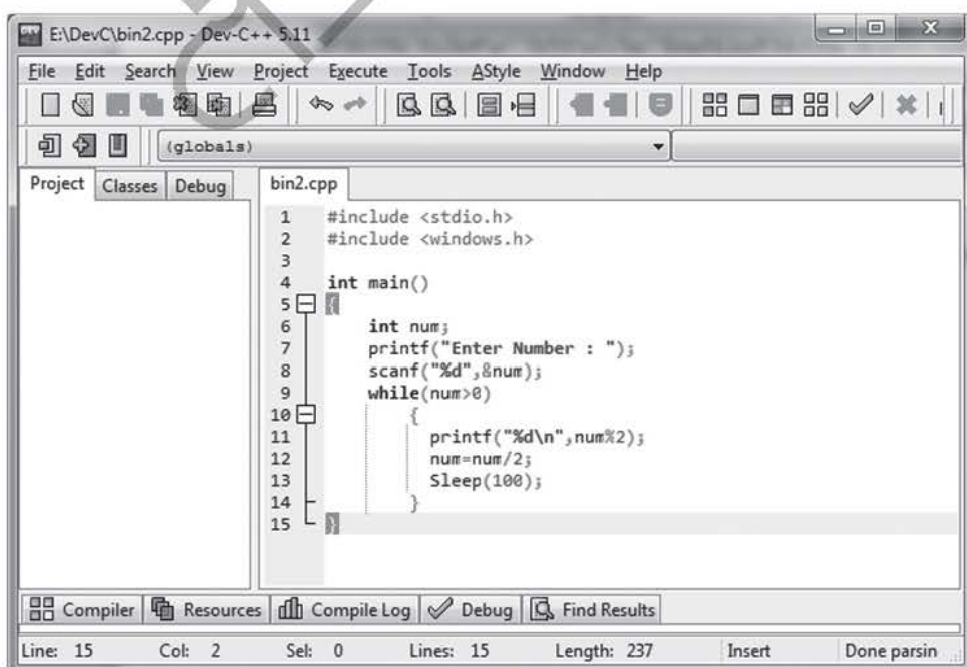


“ถ้าหนังสือมีข้อผิดพลาดเนื่องจากการพิมพ์ ให้นำมาแลกเปลี่ยนได้ฟรีที่สมาคมฯ” โทร. 0-2258-0320 ต่อ 1560, 1570

■ บรรณาธิการบริหาร สุกัญญา จารุกการ บรรณาธิการเล่ม พรรณพิมล กิจไพฑูรย์ กองบรรณาธิการ รินดา คันธวร, วิธภา สิริขตานนท์, แสงเงิน นาคพัฒน์ ออกแบบปก ธารินี คุณตะสิงค์ ออกแบบรูปเล่ม ประเทือง คชเสนีย์ ศิลปกรรม ศิริบุษ อิศรางกูร ณ อยุธยา ธุรกิจสำนักพิมพ์ อังคณา อรรถพงศ์ธร ■ พิมพ์ที่ : ห้างหุ้นส่วนจำกัด ที.เอส.บี. โปรดักส์

พื้นฐานภาษาซี

ภาษาซี (C Language) เป็นภาษาที่ได้รับความนิยมมาตั้งแต่อดีตจนถึงปัจจุบัน ถูกพัฒนา
มาจากภาษาบีโดย เดนนิส ริตชี (Dennis Ritchie) ที่ห้องปฏิบัติการเบลล์ (Bell Laboratories)
ในสหรัฐอเมริกาเมื่อปี ค.ศ. 1972 เป็นภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรม มีลักษณะ
ของภาษาที่เป็นโครงสร้าง การเขียนโปรแกรมเป็นแบบลำดับ และเป็นภาษาที่ใช้คอมไพเลอร์เป็น
ตัวแปลความหมาย ถูกจัดกลุ่มเป็นภาษาระดับกลางของภาษาคอมพิวเตอร์ ภาษาซีสามารถ
ประยุกต์ใช้งานในด้านต่าง ๆ ได้อย่างกว้างขวาง เช่น โปรแกรมระบบ โปรแกรมคำนวณ โดยเฉพาะ
ไมโครคอนโทรลเลอร์ตระกูลต่าง ๆ เช่น MCS-51 PIC และ Arduino จะใช้ภาษาซีในการพัฒนา
โปรแกรม ในบทที่ 1 ถึง 4 ของหนังสือเล่มนี้จะใช้โปรแกรม Dev-C++ เป็นเครื่องมือในการพัฒนา
โปรแกรม



รูปที่ 1.1 โปรแกรม Dev-C++

1.1 การเขียนโปรแกรมภาษาซีในงานควบคุม

การเขียนโปรแกรมภาษาซีเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ให้สามารถประยุกต์ใช้งานได้จะต้องเรียนรู้และเข้าใจใน 4 ส่วนดังนี้

ส่วนที่ 1) ภาษาซี เรียนรู้การใช้งานโปรแกรม โครงสร้างของภาษาซี ฟังก์ชัน คำสั่ง อัลกอริทึมในการแก้ปัญหา และฝึกทักษะการเขียนโปรแกรม

ส่วนที่ 2) ไมโครคอนโทรลเลอร์ Arduino เรียนรู้โครงสร้าง ขาสัญญาณ คุณสมบัติของ Arduino ที่ใช้งาน ซึ่ง Arduino ถูกออกแบบมาให้สามารถใช้งานได้ง่าย ไม่จำเป็นต้องเข้าใจถึงโครงสร้างภายในหรือจิสเตอร์เหมือนไมโครคอนโทรลเลอร์รุ่นก่อน

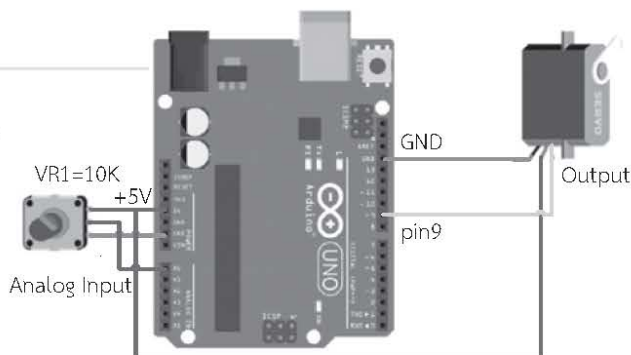
ส่วนที่ 3) การเชื่อมต่อกับอุปกรณ์อินพุต เอาต์พุต และการประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ Arduino ถูกออกแบบมาให้เชื่อมต่อกับอุปกรณ์อินพุตและเอาต์พุตได้โดยง่าย ทั้งแบบดิจิทัลและแอนะล็อก มีไลบรารีฟังก์ชันให้ใช้พัฒนาระบบได้อย่างรวดเร็ว

ส่วนที่ 4) ใจรัก เพียงแค่ใจรักก็จะสำเร็จแล้วครึ่งหนึ่ง เพราะจะทำให้เราเรียนรู้ภาษาซี และไมโครคอนโทรลเลอร์อย่างมีความสุข หากขยันหมั่นฝึกฝนก็จะประสบความสำเร็จในการเรียน

1) ภาษาซี



2) ไมโครคอนโทรลเลอร์ Arduino



3) การเชื่อมต่อกับอุปกรณ์อินพุต เอาต์พุต และการประยุกต์ใช้งาน



4) ใจรัก

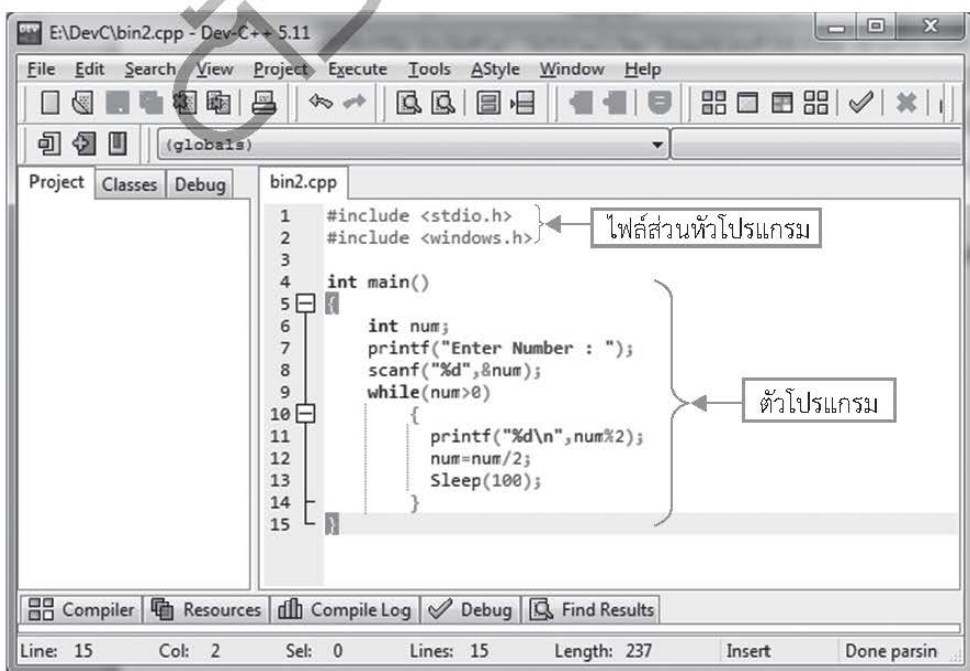
รูปที่ 1.2 ส่วนประกอบของการเขียนโปรแกรมภาษาซีในงานควบคุม

1.2 โครงสร้างของภาษาซี

โครงสร้างของภาษาซีประกอบด้วย 2 ส่วนใหญ่ ๆ คือ ไฟล์ส่วนหัวโปรแกรม (Header Files) และตัวโปรแกรม

1) **ไฟล์ส่วนหัวโปรแกรม** เป็นไฟล์ที่มีส่วนขยายเป็น *.h ใช้เก็บไลบรารีของภาษาซีเพื่อใช้ร่วมในการคอมไพล์โปรแกรม อย่างเช่น `stdio.h` เป็นไฟล์ที่เก็บไลบรารีมาตรฐานเกี่ยวกับการรับข้อมูลและการแสดงผล ซึ่งฟังก์ชัน `printf()`; ถูกนิยามไว้ใน `stdio.h` ดังนั้นโปรแกรมใดที่มีการเรียกใช้ฟังก์ชัน `printf()`; ต้องประกาศไฟล์ `stdio.h` เพื่อใช้ร่วมในการคอมไพล์โปรแกรมด้วย

2) **ตัวโปรแกรม** เริ่มต้นด้วยฟังก์ชัน `int main()` ซึ่งเป็นฟังก์ชันหลัก มีเครื่องหมายปีกกาเปิดเป็นเครื่องหมายเริ่มต้นการเขียนโปรแกรม และเครื่องหมายปีกกาปิดเป็นเครื่องหมายจบโปรแกรม ภายในฟังก์ชัน `int main()` ประกอบไปด้วยชุดคำสั่งและฟังก์ชันต่าง ๆ ซึ่งเกือบทั้งหมดจะปิดท้ายด้วยเครื่องหมายเซมิโคลอน การเขียนโปรแกรมภาษาซีจะแยกความแตกต่างระหว่างอักษรตัวพิมพ์เล็กและอักษรตัวพิมพ์ใหญ่ ในส่วนของคำอธิบายจะคั่นด้วยเครื่องหมาย `/*` และ `*/` ตามลำดับ หรือนำหน้าด้วยเครื่องหมาย `//` โดยส่วนที่เป็นคำอธิบายการทำงานของโปรแกรมจะไม่มีผลต่อการคอมไพล์

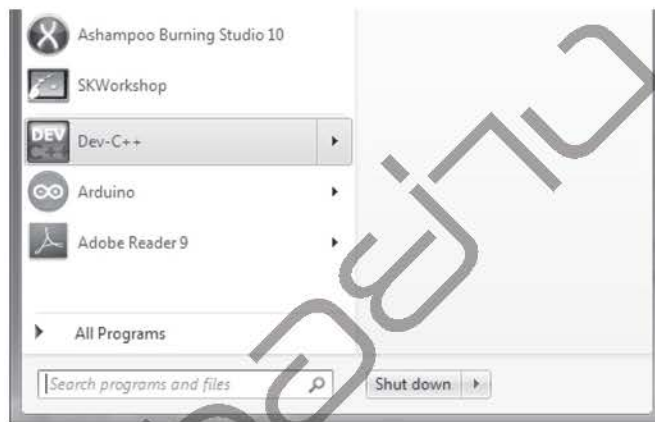


รูปที่ 1.3 ส่วนประกอบของภาษาซี

1.3 ขั้นตอนการใช้งานโปรแกรม Dev-C++

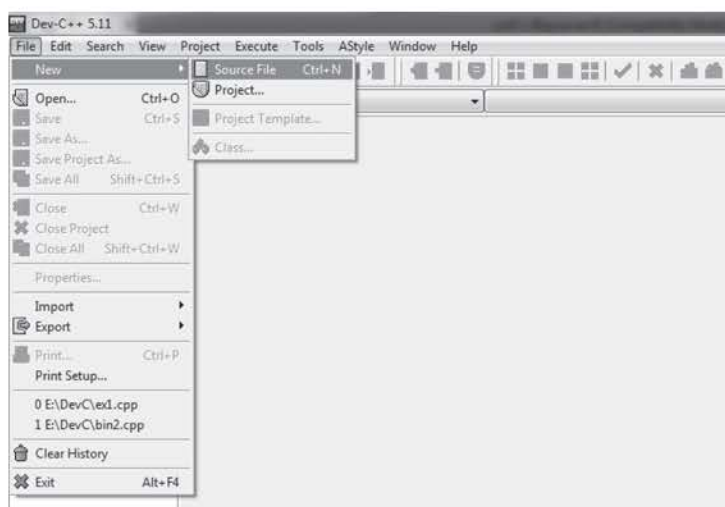
การพัฒนาโปรแกรมภาษาซีเริ่มจากเขียนโปรแกรม (Source Program) ซึ่งต้องเขียนตามลักษณะโครงสร้างของภาษาซี แล้วบันทึกไฟล์ให้มีนามสกุลเป็น *.cpp จากนั้นทำการคอมไพล์และรันโปรแกรม ถ้ามีข้อผิดพลาด ให้กลับไปแก้ไขที่โปรแกรม แต่ถ้าไม่มีข้อผิดพลาดจะได้ไฟล์ *.exe และแสดงผลของการรันโปรแกรมบนหน้าจอ ซึ่งมีขั้นตอนตามลำดับดังนี้

ลำดับที่ 1 เข้าโปรแกรม Dev-C++



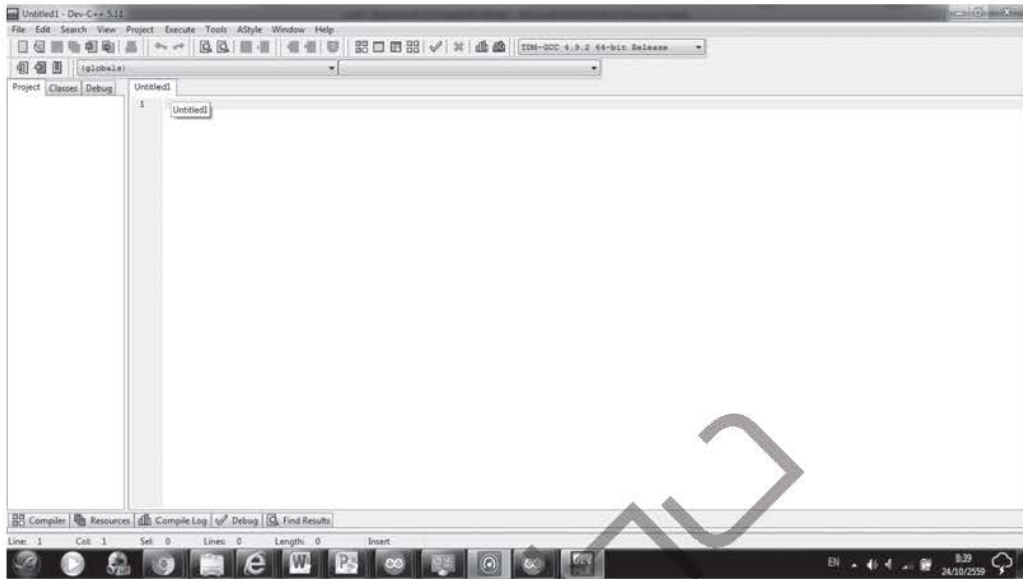
รูปที่ 1.4 การเข้าโปรแกรม Dev-C++

ลำดับที่ 2 สร้างไฟล์โดยคลิกที่ New และ Source File



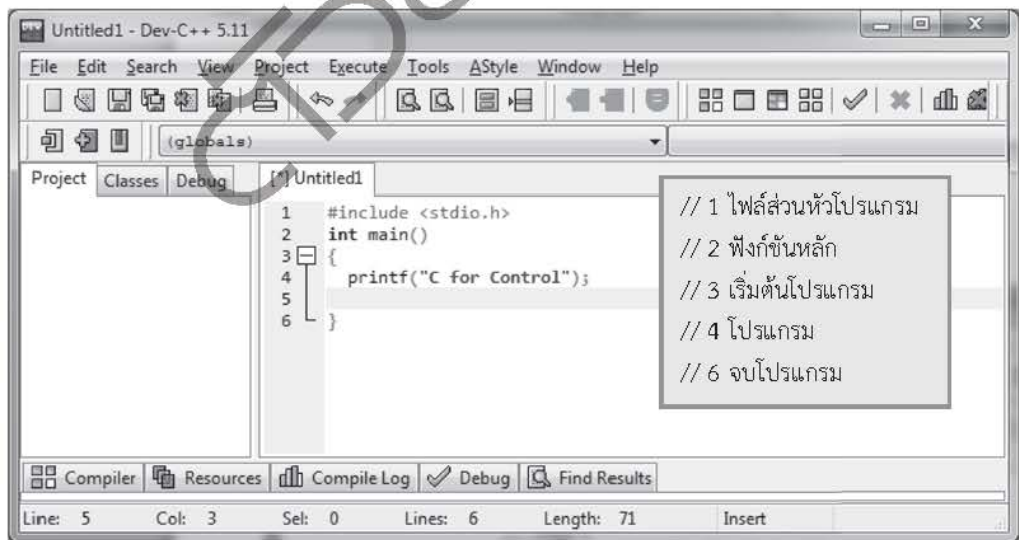
รูปที่ 1.5 การสร้างไฟล์ใหม่

ลำดับที่ 3 จะได้หน้าต่างของไฟล์ใหม่เพื่อเขียนโปรแกรม



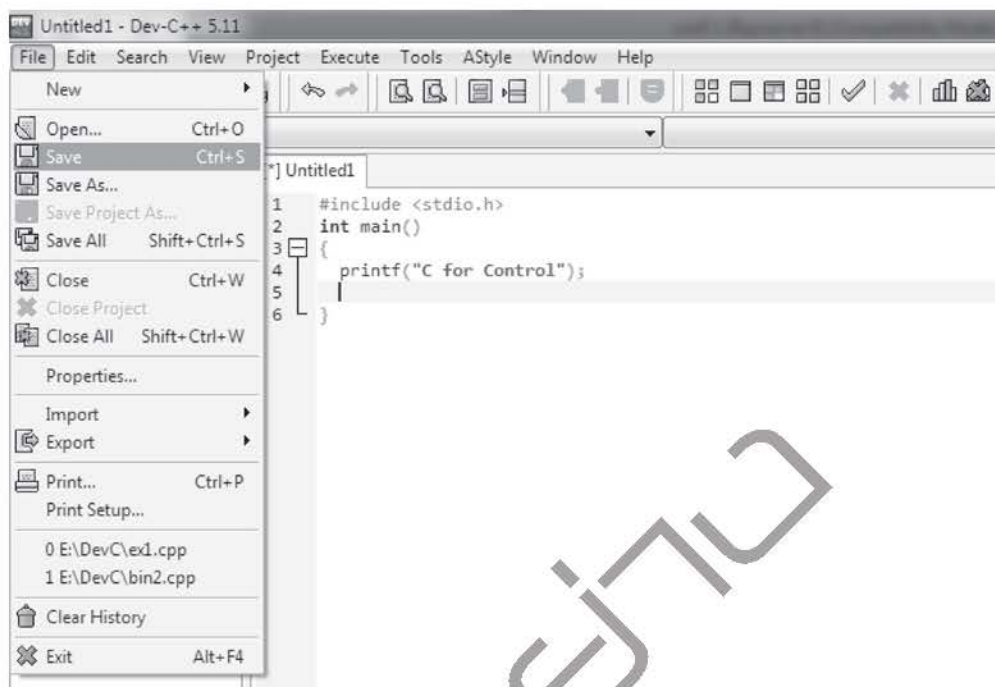
รูปที่ 1.6 หน้าต่างไฟล์ใหม่

ลำดับที่ 4 พิมพ์โปรแกรมตามตัวอย่าง



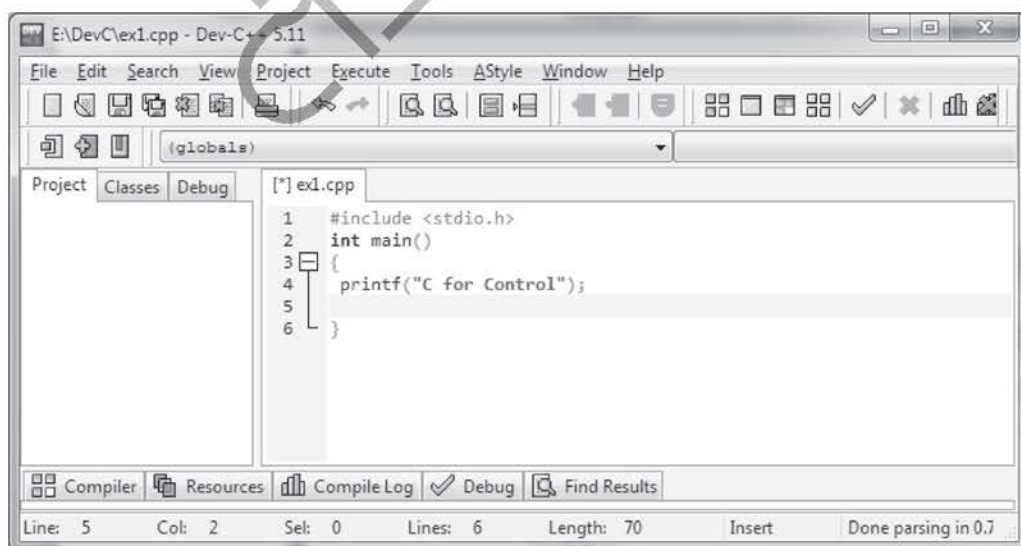
รูปที่ 1.7 โปรแกรมแสดงข้อความ C for Control

ลำดับที่ 5 ทำการบันทึกไฟล์ ex1.cpp



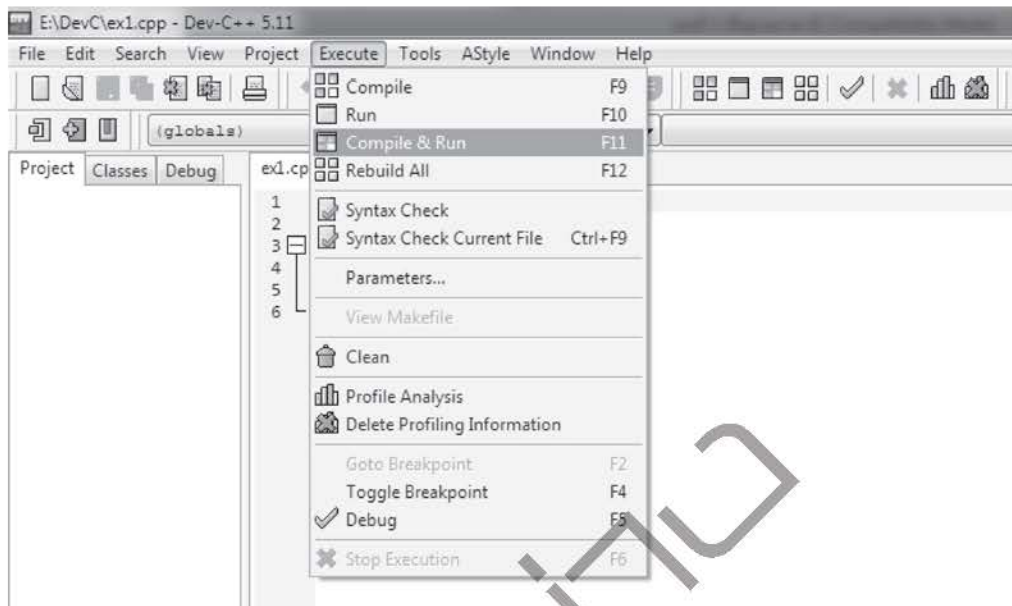
รูปที่ 1.8 การบันทึกไฟล์

ลำดับที่ 6 เมื่อบันทึกไฟล์ ex1.cpp เสร็จแล้ว



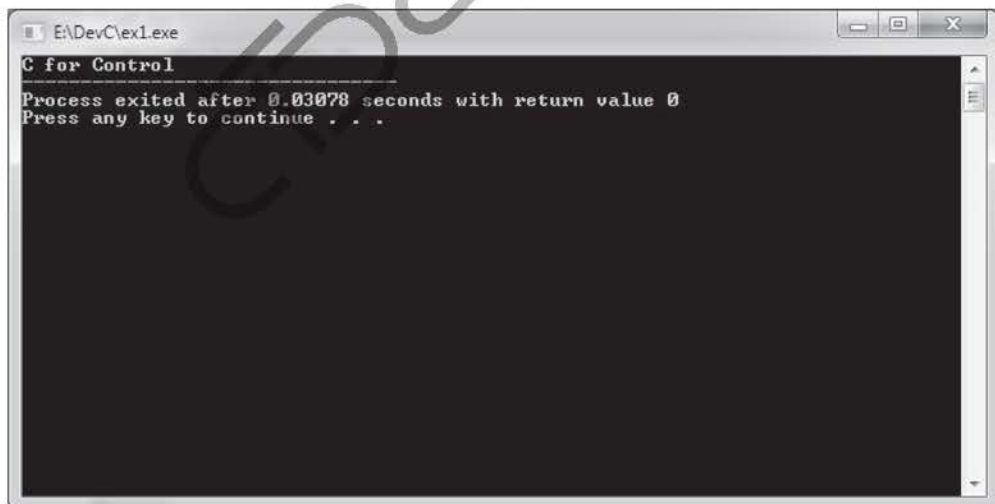
รูปที่ 1.9 ไฟล์ ex1.cpp

ลำดับที่ 7 ทำการ Compile และ Run โปรแกรม



รูปที่ 1.10 การ Compile และ Run โปรแกรม

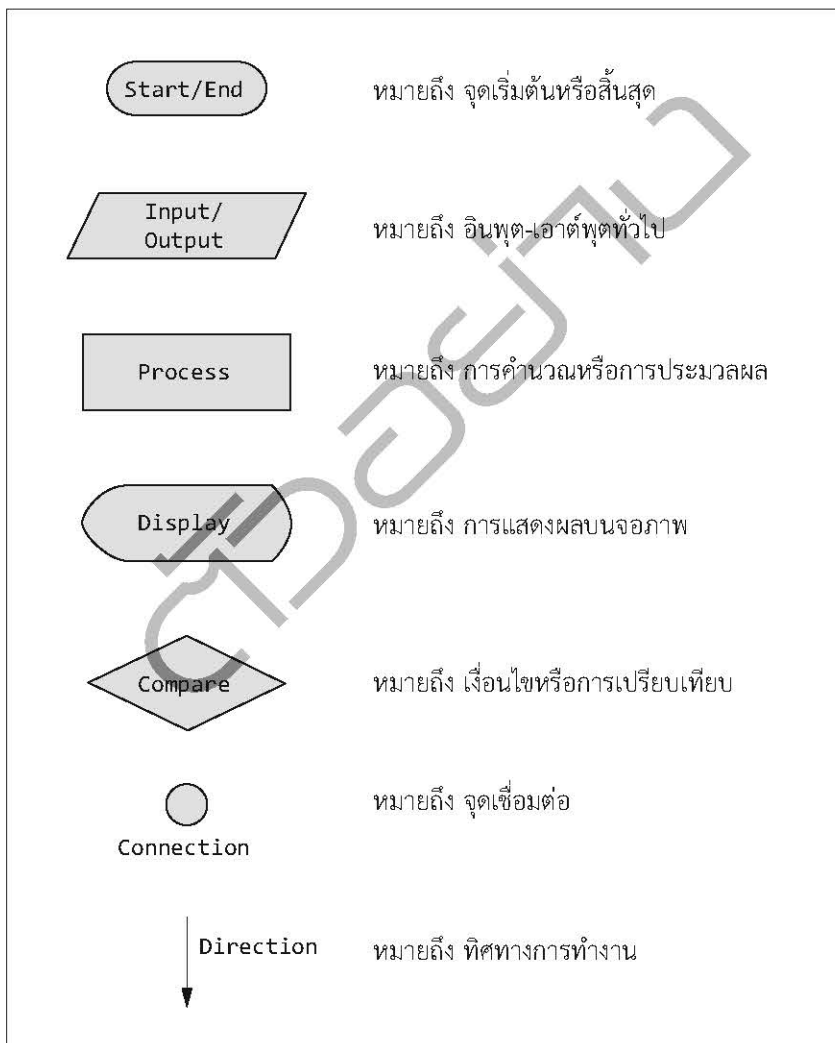
ลำดับที่ 8 ผลของการรันโปรแกรม หน้าจอจะแสดงข้อความ C for Control



รูปที่ 1.11 ผลการรันโปรแกรมแสดงข้อความ C for Control

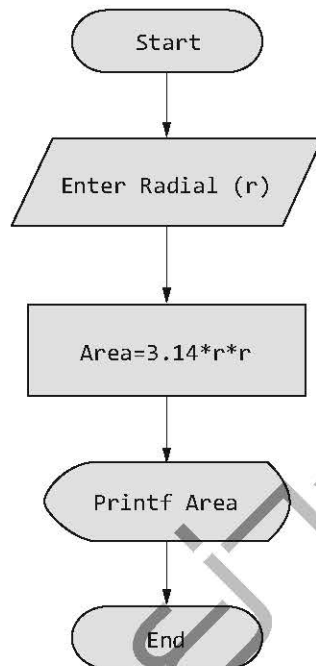
1.4 โฟลว์ชาร์ต

โฟลว์ชาร์ต (Flowchart) คือการเขียนแผนภาพที่แสดงลำดับขั้นตอนการทำงานของโปรแกรมหรือระบบงาน เพื่อให้ง่ายต่อการทำความเข้าใจและการแก้ไขปัญหา หลักการเขียนโฟลว์ชาร์ตควรเขียนจากบนลงล่าง มีจุดเริ่มต้นและสิ้นสุดเพียงจุดเดียว สัญลักษณ์พื้นฐานของการเขียนโฟลว์ชาร์ตแสดงดังรูปที่ 1.12



รูปที่ 1.12 สัญลักษณ์พื้นฐานของการเขียนโฟลว์ชาร์ต

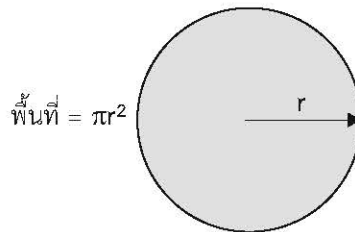
ตัวอย่างที่ 1.1 โฟลว์ชาร์ตการคำนวณหาพื้นที่วงกลม



รูปที่ 1.13 โฟลว์ชาร์ตการคำนวณหาพื้นที่วงกลม

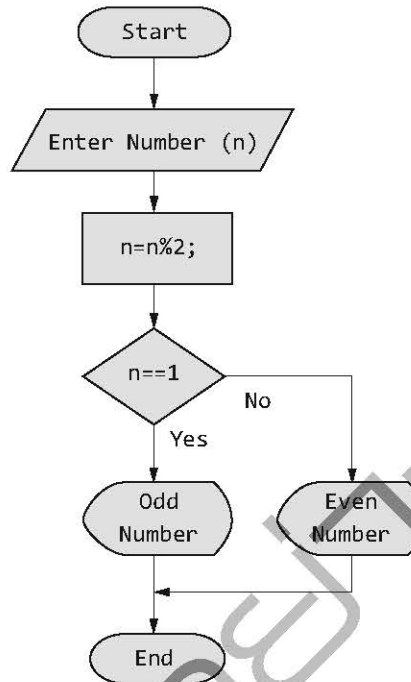
คำอธิบาย

- 1) เริ่มการทำงาน
- 2) รอรับค่ารัศมีมาเก็บไว้ที่ตัวแปร r
- 3) คำนวณหาพื้นที่ของวงกลมตามสูตร พื้นที่ = πr^2 หรือ $3.14 * r * r$



- 4) แสดงค่าพื้นที่ของวงกลมที่ได้จากการคำนวณ
- 5) จบการทำงาน

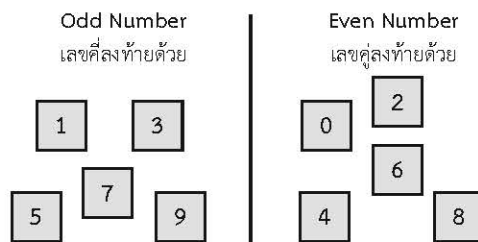
ตัวอย่างที่ 1.2 โปรแกรมตรวจสอบเลขคู่-เลขคี่



รูปที่ 1.14 โปรแกรมตรวจสอบเลขคู่-เลขคี่

คำอธิบาย

- 1) เริ่มการทำงาน
- 2) รอรับค่าจำนวนเต็มมาเก็บไว้ที่ตัวแปร n
- 3) คำนวณค่า n โดย n เท่ากับเศษที่ได้จากการหาร n ด้วย 2 ซึ่งผลลัพธ์คือ 0 หรือ 1
- 4) เปรียบเทียบค่าตัวแปร n กับ 1
 - ถ้าค่าของตัวแปร n เท่ากับ 1 แสดงว่าเป็นเลขคี่ ให้พิมพ์ Odd Number
 - ถ้าไม่ใช่ (ค่าของตัวแปร n เท่ากับ 0) แสดงว่าเป็นเลขคู่ ให้พิมพ์ Even Number



- 5) จบการทำงาน

1.5 ฟังก์ชัน printf();

ฟังก์ชัน printf(); เป็นฟังก์ชันที่ใช้พิมพ์หรือแสดงข้อความออกทางจอภาพ ถูกนิยามไว้ใน stdio.h

รูปแบบที่ 1

```
printf("ข้อความ");
```

ตัวอย่าง

```
printf("THAILAND");
```

ผลการรันโปรแกรม จะแสดงข้อความ THAILAND ออกทางจอภาพ

รูปแบบที่ 2

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

ตัวอย่าง

```
A=10;
```

```
printf("%d",A);
```

หมายถึง การนำค่าของตัวแปร A มาแสดงผลบนจอภาพในรูปของจำนวนเต็ม

ผลการรันโปรแกรม จะแสดงเลข 10 ออกทางจอภาพ

รูปแบบที่ 3

```
printf("ข้อความ ตัวกำหนดชนิดข้อมูล",ตัวแปร);
```

ตัวอย่าง

```
A=20;
```

```
printf("Data A = %d",A);
```

ผลการรันโปรแกรม จะแสดงข้อความ Data A = 20 ออกทางจอภาพ

1.6 ฟังก์ชัน scanf();

ฟังก์ชัน scanf(); เป็นฟังก์ชันที่ใช้รับค่าข้อมูลจากแป้นพิมพ์มาเก็บไว้ในตัวแปรเพื่อใช้สำหรับการคำนวณและการประมวลผล ซึ่งถูกนิยามไว้ใน stdio.h

รูปแบบ

```
scanf("ตัวกำหนดชนิดข้อมูล",ตัวแปร);
```

ตัวอย่าง

```
scanf("%d",&A);
```

หมายถึง รับรับค่าข้อมูลเป็นเลขจำนวนเต็มมาเก็บไว้ที่ตัวแปร A (&A คือที่อยู่ของตัวแปร A)

```
scanf("%f",&num);
```

หมายถึง รับรับค่าข้อมูลเป็นเลขทศนิยมมาเก็บไว้ที่ตัวแปร num

1.7 ตัวกำหนดชนิดข้อมูลของภาษาซี

ตารางที่ 1.1 ตัวกำหนดชนิดข้อมูลของภาษาซี

ตัวกำหนดชนิดข้อมูล	ความหมาย
%c	แสดงข้อมูลในรูปตัวอักษร
%d	แสดงข้อมูลในรูปจำนวนเต็ม
%e	แสดงข้อมูลในรูปเอกซ์โพเนนเชียล
%f	แสดงข้อมูลในรูปตัวเลขทศนิยม
%o	แสดงข้อมูลในรูปตัวเลขฐาน 8
%x	แสดงข้อมูลในรูปตัวเลขฐาน 16
%u	แสดงข้อมูลในรูปจำนวนเต็มบวก
%s	แสดงข้อมูลในรูปข้อความ
%p	แสดงข้อมูลในรูปที่อยู่ของตัวแปร

1.8 การประกาศตัวแปรในภาษาซี

การใช้งานตัวแปรในภาษาซีต้องมีการประกาศตัวแปรหรือการตั้งชื่อตัวแปรและกำหนดชนิดของตัวแปรจึงจะสามารถนำตัวแปรนั้นมาใช้งานได้ การตั้งชื่อตัวแปรมีข้อกำหนดดังนี้

- 1) ชื่อตัวแปรตัวแรกต้องขึ้นต้นด้วยตัวอักษรหรือเครื่องหมาย “_” เช่น num, max, _VR
- 2) ไม่มีการเว้นวรรคแต่สามารถใช้เครื่องหมาย “_” คั่นได้ เช่น sw_1, stop_motor
- 3) ถัดจากตัวแรกแล้วจะเป็นตัวเลขหรือเครื่องหมาย “_” ได้ เช่น a1, i_sensor, VR_1
- 4) อักษรตัวพิมพ์เล็กและพิมพ์ใหญ่ถือว่าแตกต่างกัน เช่น a1, A1 เป็นคนละตัวแปร
- 5) ห้ามตั้งชื่อซ้ำกับคำสงวนในภาษาซี อย่างเช่น if, do, while, printf, else ฯลฯ

ตัวอย่างการตั้งชื่อตัวแปรที่ถูกต้อง

```
int a1, max, num;
char c, VR_1, sw1;
```

ตัวอย่างการตั้งชื่อตัวแปรที่ผิด

```
int 1a, ma x, 2num;
char else, for, data 1;
```

1.9 ชนิดของตัวแปรในภาษาซี

ตารางที่ 1.2 ชนิดของตัวแปรในภาษาซี

ชนิดของตัวแปร	ขนาด	ช่วงของข้อมูล	การกำหนดชนิดตัวแปร
1) Character	8 บิต	-128 ถึง +127	char
2) Unsigned Character	8 บิต	0 ถึง 255	unsigned char
3) Integer	16 บิต	-32,768 ถึง +32,767	int
4) Unsigned Integer	16 บิต	0 ถึง 65,535	unsigned int
5) Long Integer	32 บิต	-2,147,483,648 ถึง 2,147,483,647	long int
6) Floating Point	32 บิต	3.4×10^{-38} ถึง $3.4 \times 10^{+38}$	float
7) Double	64 บิต	1.7×10^{-308} ถึง $1.7 \times 10^{+308}$	double

1) **Character** ตัวแปรชนิดตัวอักษรหรือจำนวนเต็มขนาด 8 บิต ใช้เก็บข้อมูลที่เป็นตัวอักษรหรือจำนวนเต็มที่มีค่าอยู่ในช่วง -128 ถึง +127

2) **Unsigned Character** ตัวแปรชนิดตัวอักษรหรือจำนวนเต็มขนาด 8 บิต แบบไม่คิดเครื่องหมาย ใช้เก็บข้อมูลที่เป็นตัวอักษรหรือจำนวนเต็มบวกที่มีค่าอยู่ในช่วง 0 ถึง 255

3) **Integer** ตัวแปรชนิดจำนวนเต็มขนาด 16 บิต ใช้เก็บข้อมูลที่เป็นจำนวนเต็มที่มีค่าอยู่ในช่วง -32,768 ถึง +32,767

4) **Unsigned Integer** ตัวแปรชนิดจำนวนเต็มขนาด 16 บิต แบบไม่คิดเครื่องหมายหรือจำนวนเต็มบวก ใช้เก็บจำนวนเต็มบวกที่มีค่าอยู่ในช่วง 0 ถึง 65,535

5) **Long Integer** ตัวแปรชนิดจำนวนเต็มขนาด 32 บิต ใช้เก็บข้อมูลจำนวนเต็มที่มีค่าอยู่ในช่วง -2,147,483,648 ถึง +2,147,483,647

6) **Floating Point** ตัวแปรชนิดเลขทศนิยมขนาด 32 บิต ใช้เก็บข้อมูลเลขทศนิยมที่มีค่าอยู่ในช่วง 3.4×10^{-38} ถึง $3.4 \times 10^{+38}$

7) **Double** ตัวแปรชนิดเลขทศนิยมขนาด 64 บิต ใช้เก็บข้อมูลเลขทศนิยมที่มีค่าอยู่ในช่วง 1.7×10^{-308} ถึง $1.7 \times 10^{+308}$

ตัวอย่างการประกาศตัวแปร

`char key;` หมายถึง การประกาศตัวแปร `key` เป็นตัวแปรชนิดตัวอักษรหรือจำนวนเต็มขนาด 8 บิต

`int A, B;` หมายถึง การประกาศตัวแปร `A` และ `B` เป็นตัวแปรชนิดจำนวนเต็มขนาด 16 บิต

`float f1, f2;` หมายถึง การประกาศตัวแปร `f1` และ `f2` เป็นตัวแปรชนิดเลขทศนิยมเพื่อใช้คำนวณตัวเลขที่มีจุดทศนิยม

`long int l1, num;` หมายถึง การประกาศตัวแปร `l1` และ `num` เป็นตัวแปรชนิดจำนวนเต็มขนาด 32 บิต ใช้ในการคำนวณตัวเลขที่มีค่ามาก ๆ

การกำหนดชนิดของตัวแปรควรกำหนดให้สอดคล้องกับการใช้งาน อย่างเช่น ถ้ามีการคำนวณตัวเลขที่เป็นจุดทศนิยม ต้องกำหนดชนิดของตัวแปรเป็นแบบ `float` หรือการกำหนดตัวแปรเพื่อใช้ในการวนรอบที่ไม่เกิน 127 รอบ ควรกำหนดตัวแปรเป็นแบบ `char` เป็นต้น ทั้งนี้เพื่อลดขนาดของโปรแกรมให้เล็กลงและทำให้การประมวลผลเร็วขึ้น

1.10 ตัวดำเนินการทางคณิตศาสตร์

ในการเขียนโปรแกรมที่มีการคำนวณทางคณิตศาสตร์ จำเป็นต้องทราบความหมายของตัวดำเนินการทางคณิตศาสตร์ตามลักษณะโครงสร้างของภาษาซี เพื่อให้สามารถเขียนโปรแกรมคำนวณได้อย่างถูกต้อง

ตารางที่ 1.3 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์	ความหมาย
+	การบวก
-	การลบ
*	การคูณ
/	การหาร
%	การหารที่คิดเฉพาะเศษ
++	การเพิ่มค่าขึ้น 1
--	การลดค่าลง 1

ตัวอย่างการใช้ตัวดำเนินการทางคณิตศาสตร์

$a = b + 5;$ หมายถึง a เท่ากับ b บวก 5

$c = a - 2;$ หมายถึง c เท่ากับ a ลบ 2

$x = a * b;$ หมายถึง x เท่ากับ a คูณ b

$y = a / b;$ หมายถึง y เท่ากับ a หาร b

$z = 7 \% 2;$ หมายถึง z เท่ากับ เศษที่ได้จากการหาร 7 ด้วย 2
ซึ่งจะได้ผลลัพธ์คือ 1

$a++;$ หรือ $a = a + 1;$ หมายถึง เพิ่มค่า a ครั้งละ 1

$a--;$ หรือ $a = a - 1;$ หมายถึง ลดค่า a ครั้งละ 1

ตัวอย่างที่ 1.3 โปรแกรมแสดงค่าตามตัวกำหนดชนิดข้อมูล

```
1  #include <stdio.h>
2  int main()
3  {
4      char x=65;
5      printf("char x = %c \n",x);
6      printf("int x = %d \n",x);
7      printf("Hex x = %x \n",x);
8      printf("float x = %f \n",x);
9  }
```

ผลการรันโปรแกรม

โปรแกรมจะแสดงค่าตามตัวกำหนดชนิดข้อมูลดังนี้

```
char x = A           // แสดงเป็นตัวอักษร
                        // (65 เลขฐานสิบจะมีค่าเท่ากับตัวอักษร A ตามรหัสแอสกี)
int x  = 65          // แสดงเป็นตัวเลข
Hex x  = 41          // แสดงเป็นเลขฐาน 16
float x = 0.000000    // แสดงเป็นเลขทศนิยม จะแสดงได้ก็ต่อเมื่อประกาศตัวแปร float
```

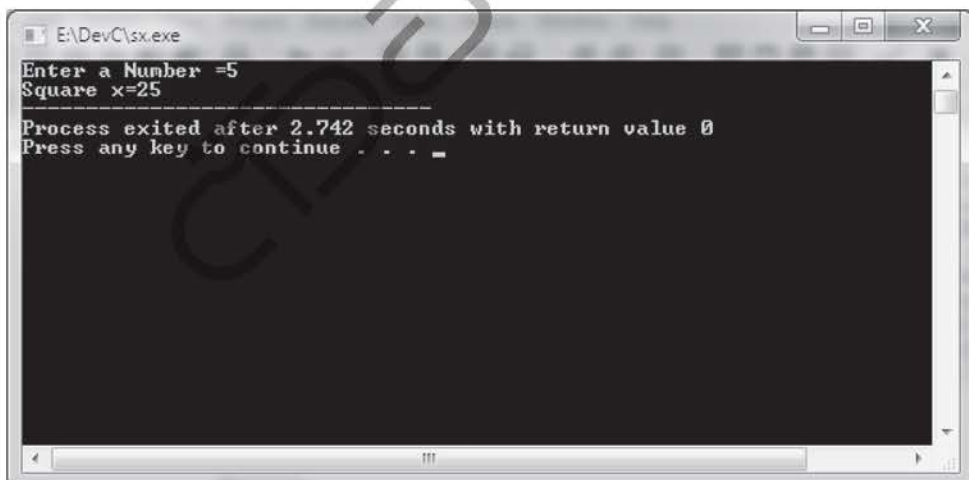
หมายเหตุ : รหัสแอสกี (ASCII : American Standard Code for Information Interchange) เป็นรหัสมาตรฐานของข้อมูลเพื่อใช้สื่อสารในระบบคอมพิวเตอร์ ซึ่งประกอบไปด้วยตัวอักษร ตัวเลข เครื่องหมาย และสัญลักษณ์ต่าง ๆ โดยแต่ละรหัสจะแทนด้วยหนึ่งอักขระ ตัวอย่างเช่น 65 (เลขฐานสิบ) ใช้แทนอักษร A สามารถดูรายละเอียดเพิ่มเติมได้จากตารางรหัสแอสกี

ตัวอย่างที่ 1.4 โปรแกรมคำนวณค่าของเลขยกกำลัง

<pre> 1 #include <stdio.h> 2 int main() 3 { 4 int x,sx; 5 printf("Enter a Number ="); 6 scanf("%d",&x); 7 sx=x*x; 8 printf("Square x=%d",sx); 9 }</pre>	<pre> // ไฟล์ส่วนหัวโปรแกรม // ฟังก์ชันหลักของโปรแกรม // ปีกกาเปิด เริ่มต้นเขียนโปรแกรม // ประกาศตัวแปร x, sx เป็นชนิดจำนวนเต็ม // พิมพ์หรือแสดงข้อความ // รอรับค่าตัวเลขมาเก็บไว้ในตัวแปร x // คำนวณค่ายกกำลัง โดย $sx = x * x$ // แสดงข้อความและผลของการยกกำลัง // ปีกกาปิด จบโปรแกรม</pre>
--	--

ผลการรันโปรแกรม

โปรแกรมจะรอรับค่าตัวเลขจากแป้นพิมพ์ จากนั้นนำตัวเลขมายกกำลังหรือคูณกันแล้วแสดงผลการคำนวณบนจอภาพ



รูปที่ 1.15 ผลการรันโปรแกรมคำนวณเลขยกกำลัง

ตัวอย่างที่ 1.5 โปรแกรมคำนวณหาพื้นที่วงกลม

```

1  #include <stdio.h>
2  int main()
3  {
4      float area,r;
5      printf("Enter a Radial =");
6      scanf("%f",&r);
7      area=3.14*r*r;
8      printf("Area a Circle =%.2f",area);
9  }
```

คำอธิบาย

บรรทัดที่ 1 ประกาศไฟล์ส่วนหัวโปรแกรม เพื่อให้สามารถเรียกใช้ฟังก์ชันต่าง ๆ ของ `stdio.h` ได้ เช่น `printf()`; `scanf()`;

บรรทัดที่ 2 ฟังก์ชันหลักของโปรแกรม

บรรทัดที่ 3 ปีกกาเปิด เริ่มต้นเขียนโปรแกรม

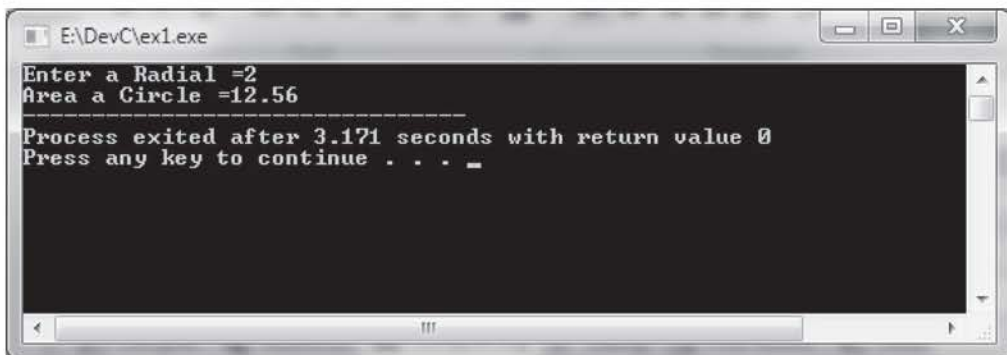
บรรทัดที่ 4 ประกาศตัวแปร `area`, `r` เป็นแบบทศนิยม

บรรทัดที่ 6 รอรับค่ารัศมีมาเก็บไว้ในตัวแปร `r`

บรรทัดที่ 7 คำนวณหาพื้นที่วงกลมตามสูตร $\text{พื้นที่} = 3.14 \times r \times r$

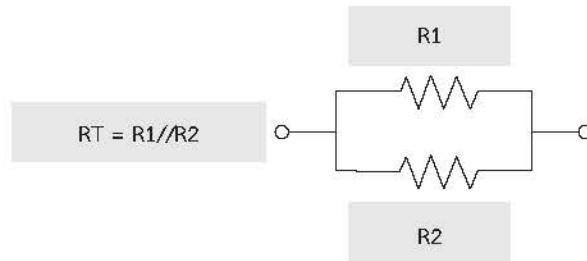
บรรทัดที่ 8 แสดงข้อความและผลการคำนวณพื้นที่วงกลมเป็นทศนิยมสองตำแหน่ง (`%.2f`)

ผลการรันโปรแกรม



รูปที่ 1.16 ผลการรันโปรแกรมคำนวณหาพื้นที่วงกลม

ตัวอย่างที่ 1.6 โปรแกรมคำนวณหาค่าความต้านทาน R1 ขนานกับ R2



```

1  #include <stdio.h>
2  int main()
3  { float rt,r1,r2;
4    printf("Enter R1 R2 =");
5    scanf("%f %f",&r1,&r2);
6    rt=r1*r1/(r1+r2);
7    printf("RT=R1//R2=%.2f",rt);
8  }

```

คำอธิบาย

- บรรทัดที่ 3 ประกาศตัวแปร rt, r1, r2 เป็นแบบทศนิยม
- บรรทัดที่ 5 รอรับค่าความต้านทานมาเก็บไว้ในตัวแปร r1 และ r2
- บรรทัดที่ 6 คำนวณหาค่าความต้านทานรวมตามสูตร
- บรรทัดที่ 7 แสดงข้อความและผลการคำนวณค่าความต้านทานรวม

ผลการรันโปรแกรม



รูปที่ 1.17 ผลการรันโปรแกรมคำนวณหาค่าความต้านทาน R1 ขนานกับ R2

1.11 สรุป

ภาษาซี เป็นภาษาที่ได้รับความนิยมมาตั้งแต่อดีตจนถึงปัจจุบัน จัดเป็นภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรม มีลักษณะของภาษาที่เป็นโครงสร้าง และการเขียนโปรแกรมเป็นแบบลำดับ ภาษาซีสามารถเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์ในตระกูลต่าง ๆ ได้ เช่น MCS-51 PIC และ Arduino ส่วนประกอบของภาษาซีแบ่งออกได้เป็น 2 ส่วนใหญ่ ๆ คือ ไฟล์ส่วนหัวโปรแกรมและตัวโปรแกรม

1) ไฟล์ส่วนหัวโปรแกรม เป็นไฟล์ที่มีส่วนขยายเป็น *.h อย่างเช่น stdio.h เป็นไฟล์ที่ใช้เก็บไลบรารีมาตรฐานของภาษาซีเกี่ยวกับการรับข้อมูลและการแสดงผล ซึ่งต้องประกาศไฟล์ stdio.h เพื่อใช้ร่วมในการคอมไพล์โปรแกรม

2) ตัวโปรแกรม เริ่มต้นด้วยฟังก์ชัน int main() ซึ่งเป็นฟังก์ชันหลัก โดยมีเครื่องหมายปีกกาเปิดและปิดแสดงการเริ่มต้นและจบโปรแกรม ภายในฟังก์ชัน int main() จะประกอบด้วยชุดคำสั่งและฟังก์ชันต่าง ๆ

การเขียนโปรแกรมภาษาซีเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ให้สามารถทำงานได้จะต้องเรียนรู้เกี่ยวกับโปรแกรมภาษาซี พื้นฐานของไมโครคอนโทรลเลอร์ Arduino และการเชื่อมต่อกับอุปกรณ์อินพุต เอาต์พุต จึงจะสามารถประยุกต์ใช้ในงานระบบควบคุมต่าง ๆ ได้

คำถามท้ายบทที่ 1

1. จงบอกประวัติของภาษาซี
2. จงบอกส่วนประกอบของภาษาซี
3. จงอธิบายความแตกต่างของตัวแปร unsigned char และ char
4. จงอธิบายความแตกต่างของตัวแปร int และ float
5. $a=5\%2$; และ $a=5/2$; ต่างกันอย่างไร
6. จงอธิบายหลักการสลับข้อมูลของตัวอย่างที่ 1.7
7. จงเขียนโฟลว์ชาร์ตการตัดเกรด A B C D และ E
8. จงเขียนโปรแกรมหาค่าพื้นที่สี่เหลี่ยมคางหมู

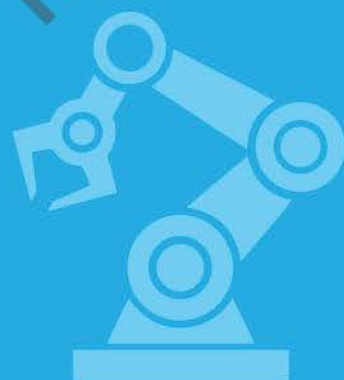
ภาษาซีและ Arduino

ลำดับเนื้อหาเข้าใจง่าย อธิบายสรุปเป็นขั้นตอน

พร้อมภาพประกอบและตัวอย่างมากมายเพื่อความเข้าใจที่มากขึ้น

เนื้อหาประกอบด้วย

- พื้นฐานภาษาซี
- การเขียนโปรแกรม DEV-C++
- ไมโครคอนโทรลเลอร์ Arduino UNO
- ดิจิทัลและแอนะล็อกอินพุต-เอาต์พุต
- การควบคุมมอเตอร์ เซอร์โวมอเตอร์
- ระบบควบคุมแบบ PID
- พื้นฐานหุ่นยนต์และแขนกล



พบหนังสือออกใหม่ • เสนอผลงานเขียน/แปล ได้ที่

www.tpa.or.th/tpapublishing

ร่วมสังคมออนไลน์กับเรา www.facebook.com/technologybook



ISBN 978-974-443-715-0



9 789744 437150

295 บาท

หมวดคอมพิวเตอร์