



# แบบฝึกหัด:

## เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี

โครงสร้างของโปรแกรมภาษา

เล่ม

2



นางสาวสุชีรา มีอาษา  
ตำแหน่ง ครู วิทยฐานะ ครูชำนาญการ  
โรงเรียนนครนายกวิทยาคม  
สำนักงานเขตพื้นที่การศึกษามัธยมศึกษา เขต 7

## คำนำ

แบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เล่ม 2 โครงสร้างของโปรแกรมภาษา เป็นนวัตกรรมที่จัดทำขึ้นเพื่อใช้พัฒนาทักษะการออกแบบและเขียนโปรแกรมคอมพิวเตอร์ของนักเรียนชั้นมัธยมศึกษาปีที่ 5 ประกอบด้วย คำชี้แจง คำแนะนำการใช้สำหรับนักเรียนสาระสำคัญ ผลการเรียนรู้ จุดประสงค์การเรียนรู้ สาระการเรียนรู้ แบบทดสอบก่อนเรียน เนื้อหาเกี่ยวกับการออกแบบและเขียนโปรแกรมคอมพิวเตอร์แบบฝึกทักษะ แบบฝึกหัด แบบทดสอบหลังเรียน และบรรณานุกรม

กิจกรรมในแบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี จะเรียงลำดับจากเนื้อหาที่ง่ายไปหายาก นักเรียนจะได้ฝึกทักษะการออกแบบและเขียนโปรแกรมคอมพิวเตอร์อย่างเป็นขั้นตอน

หวังเป็นอย่างยิ่งว่าแบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เล่ม 2 โครงสร้างของโปรแกรมภาษา จะช่วยพัฒนาทักษะการออกแบบและเขียนโปรแกรมคอมพิวเตอร์ ส่งเสริมการเขียนโปรแกรม และการประยุกต์ใช้ในการพัฒนาโปรแกรมได้เป็นอย่างดี

ขอขอบคุณผู้ที่ให้ความช่วยเหลือทุกท่านที่ได้กรุณาแนะนำ ชี้แนะแนวทางการปรับปรุง จนแบบฝึกทักษะการเขียนโปรแกรมเบื้องต้นด้วยภาษาซีนี้สำเร็จ ลุล่วง มีประสิทธิภาพเป็นอย่างดี

สุชีรา มีอาษา

## สารบัญ

	หน้า
คำชี้แจง	1
คำแนะนำการใช้สำหรับนักเรียน	1
สาระสำคัญ	3
ผลการเรียนรู้	3
จุดประสงค์การเรียนรู้	4
สาระการเรียนรู้	5
แบบทดสอบก่อนเรียน	6
โครงสร้างของโปรแกรมภาษา	8
เริ่มต้นกับภาษาซี	8
การคอมไพล์และลิงค์โปรแกรมในภาษาซี	8
โครงสร้างโปรแกรมภาษาซี (Structure of C Programs)	12
ประเภทของข้อผิดพลาด (type of Errors)	17
แบบฝึกทักษะที่ 2.1	19
ชนิดข้อมูล ตัวแปร และค่าคงที่	21
ข้อมูล (Data)	21
ชนิดของข้อมูลในภาษาซี (Data Types)	21
ตัวแปร (Variables)	23
ค่าคงที่	29
แบบฝึกทักษะที่ 2.2	32
คำสั่งรับค่าและแสดงผล	34
คำสั่งแสดงผลโปรแกรมภาษาซี (Output Function)	34
คำสั่งรับค่าทางแป้นพิมพ์โปรแกรมภาษาซี (Input Function)	38
แบบฝึกทักษะที่ 2.3	44
นิพจน์และตัวดำเนินการ	50
นิพจน์ (Expression)	50
ตัวดำเนินการ (Operator)	52
แบบฝึกทักษะที่ 2.4	57
แบบทดสอบหลังเรียน	59
บรรณานุกรม	61

## คำชี้แจง

1. แบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เล่ม 2 โครงสร้างของโปรแกรมภาษา ใช้พัฒนาทักษะการออกแบบและเขียนโปรแกรมคอมพิวเตอร์ ประกอบด้วย คำชี้แจง คำแนะนำการใช้สำหรับนักเรียน สารระสำคัญ ผลการเรียนรู้ จุดประสงค์การเรียนรู้ สารการเรียนรู้ แบบทดสอบก่อนเรียน เนื้อหาเกี่ยวกับการออกแบบและเขียนโปรแกรมคอมพิวเตอร์ แบบฝึกทักษะ แบบฝึกหัด แบบทดสอบหลังเรียน และบรรณานุกรม

2. นักเรียนควรศึกษาเนื้อหาให้เข้าใจก่อนการทำแบบฝึกทักษะ และการทำแบบฝึกทักษะ นักเรียนต้องปฏิบัติตามคำชี้แจง ที่ระบุไว้ตามลำดับให้ครบทุกแบบฝึกหัด โดยแบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี ประกอบด้วยแบบฝึกทักษะ จำนวน 4 เล่ม ดังนี้

เล่ม 1 กระบวนการแก้ปัญหา

เล่ม 2 โครงสร้างของโปรแกรมภาษา

เล่ม 3 โครงสร้างแบบมีทางเลือก

เล่ม 4 โครงสร้างแบบทำซ้ำ

3. ในการทำแบบฝึกทักษะ นักเรียนจะต้องใช้ความรู้เกี่ยวกับการวิเคราะห์โจทย์ปัญหา ซึ่งการวิเคราะห์ปัญหาที่ถูกต้องจะส่งไปถึงการออกแบบและเขียนโปรแกรมคอมพิวเตอร์ได้อย่างถูกต้องด้วย

4. หากนักเรียนยังไม่เข้าใจเนื้อหาส่วนใดส่วนหนึ่งให้กลับไปศึกษาอีกครั้ง หรือขอคำแนะนำจากครู เพื่อให้เกิดความเข้าใจมากยิ่งขึ้น

5. ในการศึกษาและทำแบบฝึกทักษะ ให้นักเรียนทำด้วยความตั้งใจและมีความซื่อสัตย์ต่อตนเอง โดยไม่เปิดดูเฉลยก่อน

6. นักเรียนสามารถประเมินตนเองด้วยการตรวจเฉลย และศึกษาแนวคำตอบในเล่ม เพื่อดูพัฒนาการของตนเอง

### คำแนะนำการใช้สำหรับนักเรียน

แบบฝึกทักษะ เรื่อง การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เป็นแบบฝึกทักษะที่จัดทำขึ้น เพื่อให้นักเรียนมีความรู้ ความเข้าใจเรื่อง กระบวนการแก้ปัญหา โครงสร้างของโปรแกรมภาษา โครงสร้างแบบมีทางเลือก และโครงสร้างแบบทำซ้ำ นักเรียนควรปฏิบัติตามนี้





### สาระสำคัญ

โปรแกรมภาษาซีเป็นโปรแกรมที่ประกอบไปด้วยฟังก์ชันหลากหลายฟังก์ชันที่ทำงานต่อเนื่องกันเพื่อให้บรรลุผลตามที่นักเขียนโปรแกรมต้องการ โดยฟังก์ชันแต่ละฟังก์ชันก็จะทำงานในส่วนย่อย ๆ ที่ถูกแบ่งออกมาจากงานชิ้นใหญ่ การสร้างโปรแกรมภาษาซี มี 4 ขั้นตอนได้แก่ เขียนโปรแกรม คอมไพล์โปรแกรม เชื่อมโยงโปรแกรม และประมวลผล

ตัวแปร จึงเป็นชื่อของหน่วยความจำในตำแหน่งใด ๆ เมื่อนำข้อมูลไปเก็บไว้ในหน่วยความจำตำแหน่งนั้น จะกล่าวว่า ตัวแปรนั้นมีค่าเท่ากับข้อมูลที่เก็บไว้ ตัวแปรสามารถเก็บค่าชนิดต่าง ๆ ตามที่ได้ประกาศไว้ในโปรแกรมเท่านั้น และค่าที่เก็บไว้ในตัวแปรนี้สามารถเปลี่ยนค่าได้ตลอดเวลา ขึ้นกับข้อคำสั่งภายในโปรแกรม ค่าคงที่ (Constant) คือ ค่าของข้อมูลที่มีจำนวนแน่นอน เป็นค่าที่ไม่สามารถเปลี่ยนแปลงได้ขณะรันโปรแกรม

การส่งข้อความ หรือค่าในตัวแปรให้แสดงผลบนจอภาพ มีคำสั่งพื้นฐานที่ใช้ในโปรแกรมภาษาซี คือ printf โดยจะส่งค่าข้อมูลออกไปยังจอภาพ พร้อมทั้งส่งผลการทำงานกลับมา การแสดงค่าของตัวแปร ฟังก์ชัน scanf เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์ (keyboard) เข้ามาเก็บไว้ในตัวแปร

นิพจน์ คือ สิ่งๆี่เมื่อผ่านการประมวลผลแล้วสามารถให้ค่าเป็นตัวเลขได้ โดยแต่ละนิพจน์มีระดับความยากง่ายในการประมวลผลที่แตกต่างกัน ตัวดำเนินการมีหลายประเภท ได้แก่ ตัวดำเนินการทางคณิตศาสตร์ ตัวดำเนินการกำหนดค่า ตัวดำเนินการยูนิารี ตัวดำเนินการเปรียบเทียบและตรรกะ ตัวดำเนินการระดับบิต ตัวดำเนินการเงื่อนไข และ ตัวดำเนินการบอกขนาดของชนิดข้อมูล เมื่อนิพจน์ประกอบไปด้วยตัวดำเนินการหลายชนิด ในการประมวลผลนิพจน์ที่มีตัวดำเนินการมากกว่าหนึ่งชนิด ลำดับความสำคัญของตัวดำเนินการจึงต้องถูกนำมาพิจารณา

### ผลการเรียนรู้



1. อธิบายโครงสร้างและเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ได้
2. อธิบายและใช้ชนิดของข้อมูล ตัวแปร และค่าคงที่ในการเขียนโปรแกรมได้
3. เขียนโปรแกรมโดยใช้คำสั่งรับและแสดงผลได้
4. อธิบายและใช้นิพจน์ ตัวดำเนินการในการเขียนโปรแกรมได้

## จุดประสงค์การเรียนรู้



## ด้านความรู้ (Knowledge)

1. ระบุและอธิบายโครงสร้างโปรแกรมภาษาซีแต่ละส่วนได้
2. อธิบายขั้นตอนของการสร้างโปรแกรมด้วยภาษาซีได้
3. ระบุข้อผิดพลาดที่เกิดจากการเขียนโปรแกรมและขั้นตอนการแก้ไขได้
4. เข้าใจความหมายของข้อมูล และอธิบายความแตกต่างระหว่างข้อมูลแต่ละประเภท
5. ระบุชนิดข้อมูลที่เหมาะสมกับข้อมูลที่ต้องการเก็บได้
6. ระบุชนิด และอธิบายความแตกต่างของชนิดของข้อมูลที่ใช้ในภาษาซีได้
7. อธิบายความหมายและลักษณะของตัวแปรได้
8. อธิบายลักษณะและรูปแบบการแสดงผลในโปรแกรมภาษาซี
9. อธิบายลักษณะและรูปแบบการรับค่าในโปรแกรมภาษาซี
10. เข้าใจความหมายของนิพจน์
11. ระบุความสัมพันธ์ระหว่างตัวดำเนินการและตัวถูกดำเนินการในนิพจน์ที่กำหนดให้ได้
12. ระบุความแตกต่างระหว่างตัวดำเนินการตรรกะและตัวดำเนินการระดับบิตได้



## ด้านทักษะ/กระบวนการ (Process)

1. สร้างโปรแกรมภาษาซีและสั่งงานคอมพิวเตอร์โดยใช้รหัสต้นฉบับที่กำหนดให้ได้
2. ตั้งชื่อตัวแปรถูกต้องตามกฎการตั้งชื่อตัวแปร
3. เขียนโปรแกรมโดยใช้วิธีการประกาศ การกำหนดค่า และเรียกใช้ค่าที่เก็บในตัวแปรได้
4. สร้างโปรแกรมภาษาซีและสั่งงานคอมพิวเตอร์โดยใช้รหัสต้นฉบับที่กำหนดให้ได้
5. หาผลลัพธ์และสร้างนิพจน์จากตัวดำเนินการทางคณิตศาสตร์ได้
6. หาผลลัพธ์และสร้างนิพจน์จากตัวดำเนินการกำหนดค่า ตัวดำเนินการยูนิารีได้
7. หาผลลัพธ์และสร้างนิพจน์จากตัวดำเนินการเปรียบเทียบและตรรกะได้
8. หาผลลัพธ์และสร้างนิพจน์จากตัวดำเนินการเงื่อนไขได้
9. ใช้ตัวดำเนินการบอกชนิดของข้อมูลได้
10. หาผลลัพธ์และสร้างนิพจน์ที่มีตัวดำเนินการหลากหลายชนิดได้
11. เขียนโปรแกรมเพื่อสร้างนิพจน์โดยใช้ตัวดำเนินการต่าง ๆ ได้

### จุดประสงค์การเรียนรู้



#### ด้านคุณลักษณะ (Attitude)

1. มีคุณลักษณะที่ดีในการใช้คอมพิวเตอร์
2. ซื่อสัตย์สุจริต
3. มีวินัย
4. ใฝ่เรียนรู้
5. มุ่งมั่นในการทำงาน



#### สาระการเรียนรู้

1. โครงสร้างโปรแกรม
2. อักขระในภาษา
3. ขั้นตอนการสร้างโปรแกรมด้วยภาษาคอมพิวเตอร์
4. ประเภทของข้อผิดพลาด
5. ชนิดของข้อมูล
6. ลักษณะของตัวแปร
7. กฎการตั้งชื่อตัวแปร
8. การประกาศตัวแปร
9. คำสั่งแสดงผลในโปรแกรมภาษาซี (Output Function)
10. คำสั่งรับค่าในโปรแกรมภาษาซี (Input Function)
11. ลักษณะของนิพจน์ และตัวดำเนินการ
12. ตัวดำเนินการกำหนดค่า
13. ตัวดำเนินการทางคณิตศาสตร์
14. ตัวดำเนินการ
15. เอกภาค
16. ตัวดำเนินการเปรียบเทียบ
17. ตัวดำเนินการตรรกะ



## แบบทดสอบก่อนเรียน

## การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เรื่อง โครงสร้างของโปรแกรมภาษา

คำชี้แจง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว

- ข้อใดกล่าว ถูกต้อง เกี่ยวกับภาษาซี
  - ใช้ตัวแปลภาษาที่แปลทีเดียวกึ่งโปรแกรม เรียกว่า คอมไพเลอร์ (Compiler)
  - มีโครงสร้างคล้ายภาษาระดับสูง แต่ทำงานได้รวดเร็วเหมือนภาษาระดับต่ำ
  - เป็นภาษาระดับกลาง (Medium Level Language )
  - ถูกทุกข้อ
- ข้อใดคือฟังก์ชันหลักของโปรแกรมภาษาซี
  - `#include <stdio.h>`
  - `main()`
  - `int num1;`
  - `getch();`
- Compiler จะแจ้งข้อความ `syntax error` เมื่อ Run โปรแกรมแล้วพบว่ามีผิดพลาด ด้านใด
  - ประกาศตัวแปรไม่ถูกต้อง
  - กำหนดค่าไม่ถูกต้อง
  - ผลลัพธ์ของโปรแกรมไม่ถูกต้อง
  - ไวยากรณ์คำสั่งไม่ถูกต้อง
- รหัสควบคุมในข้อใด ใช้เพื่อขึ้นบรรทัดใหม่ (Newline)
  - `\a`
  - `\b`
  - `\n`
  - `\v`
- ข้อใดกล่าว ไม่ถูกต้อง
  - ค่าคงที่บูลีนมี 2 กรณี คือ ถ้าเป็นเท็จ มีค่าเป็น 0 ถ้าเป็นจริง มีค่าเป็น 1
  - ตัวดำเนินการทางคณิตศาสตร์ เช่น `+`, `-`, `*`, `/`, `%`
  - ตัวดำเนินการเปรียบเทียบ เช่น `>`, `<`, `==`, `!=`
  - ตัวดำเนินการกำหนดค่า เช่น `|=`, `^=`, `&&`, `!`

6. หลักในการกำหนดชื่อตัวแปรข้อใด ไม่ถูกต้อง

- ก. ต้องขึ้นต้นด้วยตัวเลข
- ข. ต้องขึ้นต้นด้วยตัวอักษร
- ค. สามารถใช้เครื่องหมาย underline ‘ \_ ’ ได้
- ง. ห้ามใช้เครื่องหมายทางคณิตศาสตร์ในชื่อตัวแปร

7. ข้อใดแสดงผลลัพธ์ของคำสั่งต่อไปนี้ได้ถูกต้อง

```
printf("Float number: %3.2f\n", 3.14159);
```

- ก. Float number: 3.2
- ข. Float number: %3.2f\n
- ค. Float number: 3.14
- ง. Float number: 3.14159

8. ข้อใดเป็นคำสั่งที่ใช้ในการรับข้อมูลเป็นตัวอักขระ 1 ตัวอักขระเข้ามาทางแป้นพิมพ์ โดยเมื่อป้อนข้อมูลเสร็จไม่ต้องกดปุ่ม enter และข้อมูลที่ป้อนจะไม่ปรากฏบนจอภาพ

- ก. getch()
- ข. getcher()
- ค. getch()
- ง. getchar()

9. ข้อใดใช้ฟังก์ชัน printf( ) ในการแสดงผลบวกของตัวแปร x กับตัวแปร y ได้ถูกต้อง

- ก. printf(“%d + %d = %d”,x , y , x+y );
- ข. printf (“x + y = %d”, x + y );
- ค. printf (“x + y = x+ y \n”, x ,y , x+y);
- ง. printf ( “x + y = ”,x+y);

10. ตัวแปร num เป็นตัวแปรชนิด int และ temp เป็นตัวแปรชนิด float ถ้าจะให้คำสั่งต่อไปนี้ทำงานได้อย่างถูกต้อง ต้องเติมข้อความใดลงในช่องว่างของคำสั่ง

```
scanf(“%f %d”, _____);
```

- ก. num , temp
- ข. &num , &temp
- ค. temp ,num
- ง. &temp , &num

## เริ่มต้นกับภาษาซี

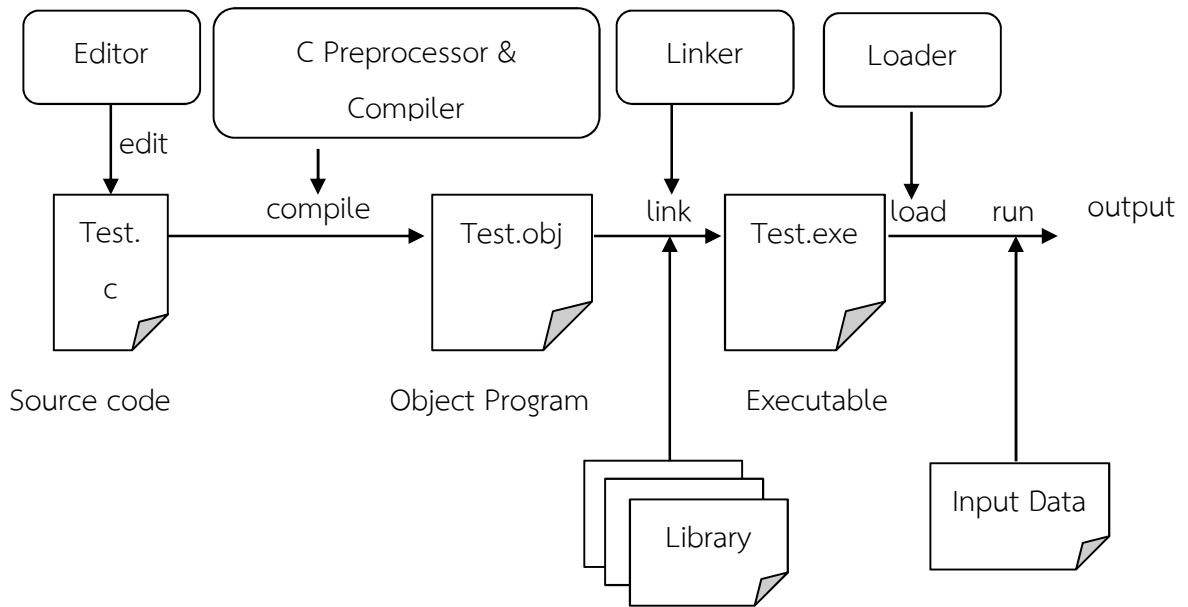
ภาษาซีเป็นภาษาโปรแกรมระดับสูง ที่ใช้สำหรับเขียนโปรแกรมประยุกต์ต่าง ๆ เช่นเดียวกันกับภาษาปาสคาล ภาษาเบสิก เป็นต้น นอกจากนี้ภาษาซียังใช้สำหรับเขียนโปรแกรมระบบ และโปรแกรมสำหรับควบคุมฮาร์ดแวร์บางส่วนที่ภาษาโปรแกรมระดับสูงหลายภาษาไม่สามารถทำได้ ดังนั้นภาษาซีจึงจัดเป็นภาษาโปรแกรมในระดับกลางด้วย

ก่อนที่โปรแกรมภาษาซีจะถูกรัน (run) จะต้องถูกแปลงให้อยู่ในรูปของอ็อบเจกต์โค้ด (object code) โดยการคอมไพล์ (compile) โปรแกรมภาษาซีที่เขียนโดยใช้คำสั่งตามมาตรฐานของ ANSI C สามารถนำไปคอมไพล์ และรันที่เครื่องคอมพิวเตอร์ต่างระบบกันได้ โปรแกรมที่เขียนขึ้นโดยใช้ภาษาโปรแกรมต่าง ๆ นั้น เราเรียกว่า รหัสต้นฉบับ (source code) ซึ่งอยู่ในรูปของข้อความตามหลักการเขียนโปรแกรมของภาษาโปรแกรมที่สามารถอ่าน และทำความเข้าใจได้โดยมนุษย์เท่านั้น ดังนั้น เพื่อให้เครื่องคอมพิวเตอร์สามารถเข้าใจโปรแกรมและปฏิบัติได้ จึงต้องนำรหัสต้นฉบับมาผ่านกระบวนการแปลงให้อยู่ในรูปของอ็อบเจกต์โค้ดที่ประกอบด้วยรหัสตัวเลข 0 และ 1 ก่อน เราเรียกกระบวนการแปลงดังกล่าวว่า การคอมไพล์โปรแกรม



## การคอมไพล์และลิงค์โปรแกรมในภาษาซี

เนื่องจากภาษาคอมพิวเตอร์ที่เป็นภาษาระดับสูงมีจุดประสงค์เพื่อให้มนุษย์สามารถเข้าใจจดจำ และสื่อสารในการเขียนโปรแกรมได้ง่ายขึ้น แต่เครื่องคอมพิวเตอร์ไม่สามารถทำงานตามภาษาระดับสูงนี้ได้โดยตรง ดังนั้นจึงต้องนำภาษาระดับสูงนี้มาผ่านกระบวนการแปลงเพื่อให้เป็นภาษาเครื่องซึ่งสามารถสั่งงานเครื่องคอมพิวเตอร์ได้โดยตรงเสียก่อน สำหรับการสร้างโปรแกรมด้วยภาษาซี มีขั้นตอนดังนี้



รูปการคอมไพล์และลิงค์โปรแกรมในภาษาซี

### ขั้นตอนที่ 1 เขียนรหัสต้นฉบับ (source code)

ใช้ Editor สร้างรหัสต้นฉบับเป็นตัวอักษรหรือเรียกว่า ซอร์สไฟล์ (Source file) โดยใช้โปรแกรมที่สามารถเขียนไฟล์ที่เก็บอักขระ (Editor) ใด ๆ ก็ได้ อักขระหรือ อักขระใด ๆ ซึ่งอยู่ในรูปแบบของโปรแกรมภาษาชั้นสูงที่สามารถอ่าน และทำความเข้าใจได้โดยมนุษย์ โดยรหัสต้นฉบับนี้จะได้รับการบันทึกเป็นไฟล์ที่มีนามสกุลเป็น .c หรือ .cpp ขึ้นมาก่อน

### ขั้นตอนที่ 2 คอมไพล์โปรแกรม (compile)

นำรหัสต้นฉบับจากขั้นตอนที่ 1 มาทำการคอมไพล์ เพื่อแปลจากภาษาที่มนุษย์เข้าใจ (ภาษาชั้นสูง) ไปเป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจได้ (ภาษาเครื่อง)

โดยก่อนที่คอมไพเลอร์ภาษาซีจะทำการแปลงรหัสต้นฉบับ ตัวประมวลผลก่อนซี (C preprocessor) จะทำงานโดยอัตโนมัติ ในขั้นตอนนี้จะมีการดำเนินการต่าง ๆ ตามตัวประมวลผลก่อนซี เช่น อ่าน header file ต่าง ๆ เพื่อให้ถูกประมวลผลร่วมกับโปรแกรม เมื่อมีการใช้คำสั่ง #include เป็นต้น

จากนั้นคอมไพเลอร์จะทำการตรวจสอบรหัสต้นฉบับว่ามีข้อผิดพลาดทางไวยากรณ์หรือไม่ ถ้าเกิดข้อผิดพลาด จะแจ้งให้ผู้เขียนโปรแกรมทราบ ผู้เขียนโปรแกรมจะต้องกลับไปแก้ไขโปรแกรม และทำการคอมไพล์โปรแกรมอีกครั้ง

หากไม่พบข้อผิดพลาด คอมไพเลอร์ของภาษาซี (C Compiler) จะทำการแปลงซอร์สไฟล์ (source code) จากอักขระใด ๆ ให้เป็นรหัสที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ เก็บไว้ในอีกไฟล์หนึ่งเรียกว่าไฟล์วัตถุ (Object file) ที่มีนามสกุล .obj

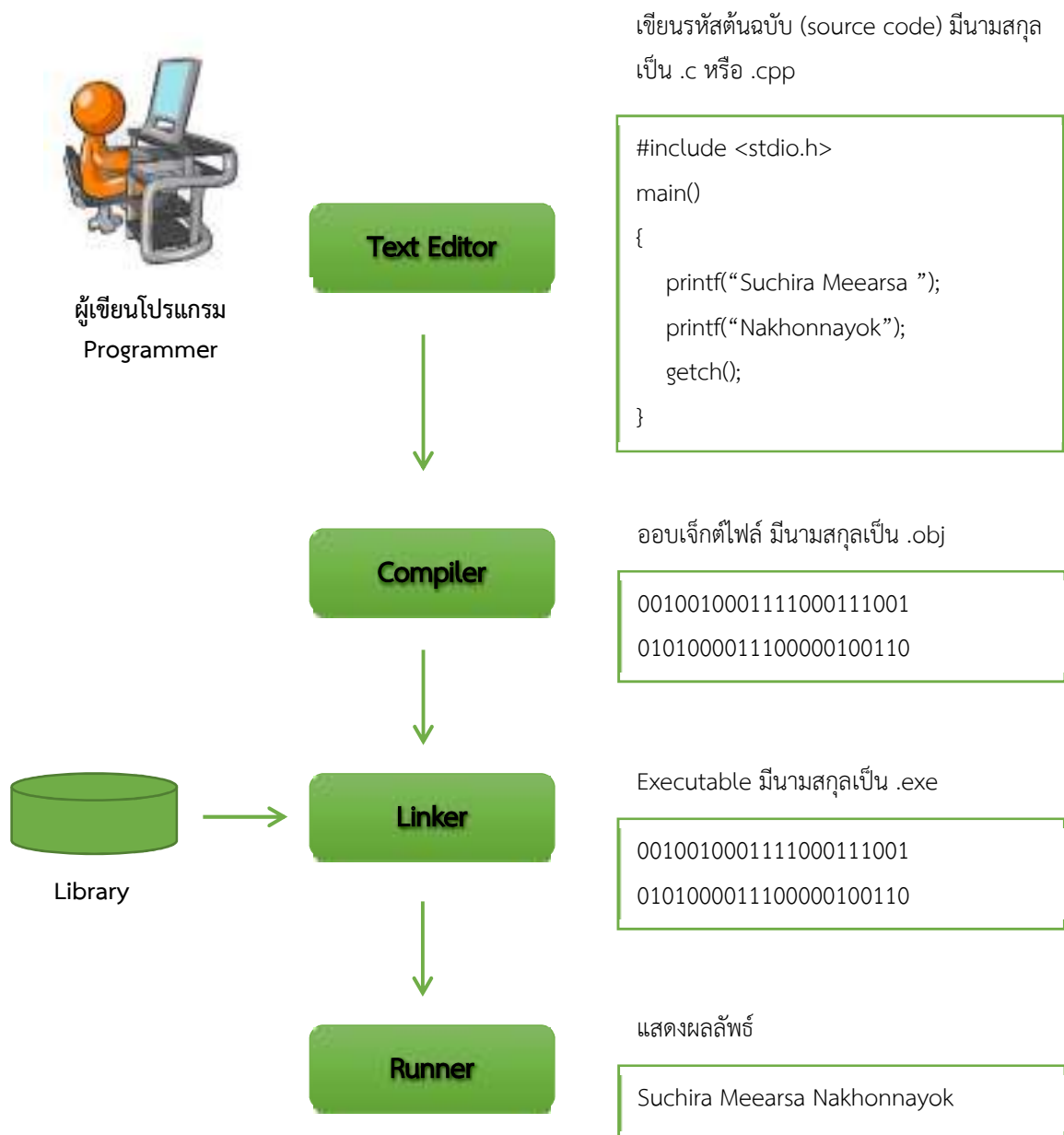
### ขั้นตอนที่ 3 เชื่อมโยงโปรแกรม (link)

การเขียนโปรแกรมภาษาซีนั้นผู้เขียนโปรแกรมไม่จำเป็นต้องเขียนคำสั่งต่าง ๆ ขึ้นใช้งานเอง เนื่องจากภาษาซีมีฟังก์ชันมาตรฐานที่เก็บไว้ใน library ให้ผู้เขียนโปรแกรมสามารถเรียกใช้งานได้ ด้วยเหตุนี้ ภาษาเครื่องที่ได้จากขั้นตอนที่ 2 จึงยังไม่สามารถนำไปใช้งานได้ แต่ต้องนำมาเชื่อมโยงเข้ากับ library ก่อน ซึ่งผลจากการเชื่อมโยงจะทำให้ได้ executable program ซึ่งมีนามสกุล .exe ที่สามารถนำไปใช้งานได้

### ขั้นตอนที่ 4 ประมวลผล (run)

ในขั้นตอนสุดท้าย executable program จากขั้นตอนที่ 3 จะถูกนำเข้าสู่หน่วยความจำของเครื่องคอมพิวเตอร์โดย Loader จากนั้นการรันโปรแกรมจึงเริ่มต้นขึ้น ซึ่งผลที่ได้จากการรันโปรแกรม (output) ขึ้นอยู่กับคำสั่งในโปรแกรมที่ปรากฏอยู่ในรหัสต้นฉบับที่เขียนขึ้นโดยนักเขียนโปรแกรมนั่นเอง

หากผลจากการรันไม่ถูกต้อง นักเขียนโปรแกรมต้องกลับไปแก้ไขรหัสต้นฉบับอีกครั้งหนึ่ง แล้วดำเนินการตามขั้นตอนต่อ ๆ มา จนได้ผลลัพธ์ตามที่นักเขียนโปรแกรมต้องการ



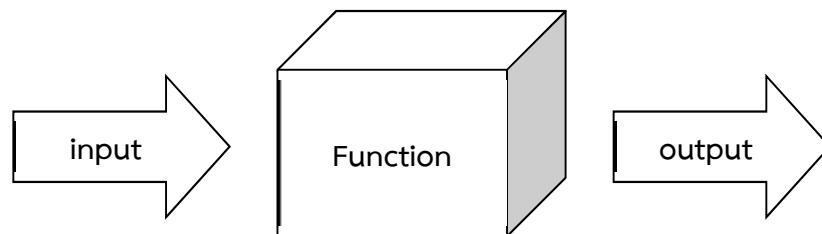
รูปแสดงขั้นตอนการทำงานของโปรแกรมภาษาซี

## โครงสร้างโปรแกรมภาษาซี (Structure of C Programs)

ภาษาซี (C Language) เกิดขึ้นจากการดัดแปลงภาษาบี (B Language) ภาษาซีเป็นภาษาที่คิดค้นโดย เดนนิส ริทช์ (Dennis Ritchie) ที่ศูนย์วิจัยเบล (Bell Laboratories) ในสหรัฐอเมริกา ราวปี ค.ศ. 1972 และเป็นภาษาที่ใช้เขียนระบบปฏิบัติการยูนิกซ์ (UNIX Operating System) ซึ่งเป็นระบบปฏิบัติการในระบบเครือข่ายคอมพิวเตอร์ที่แพร่หลายในปัจจุบัน



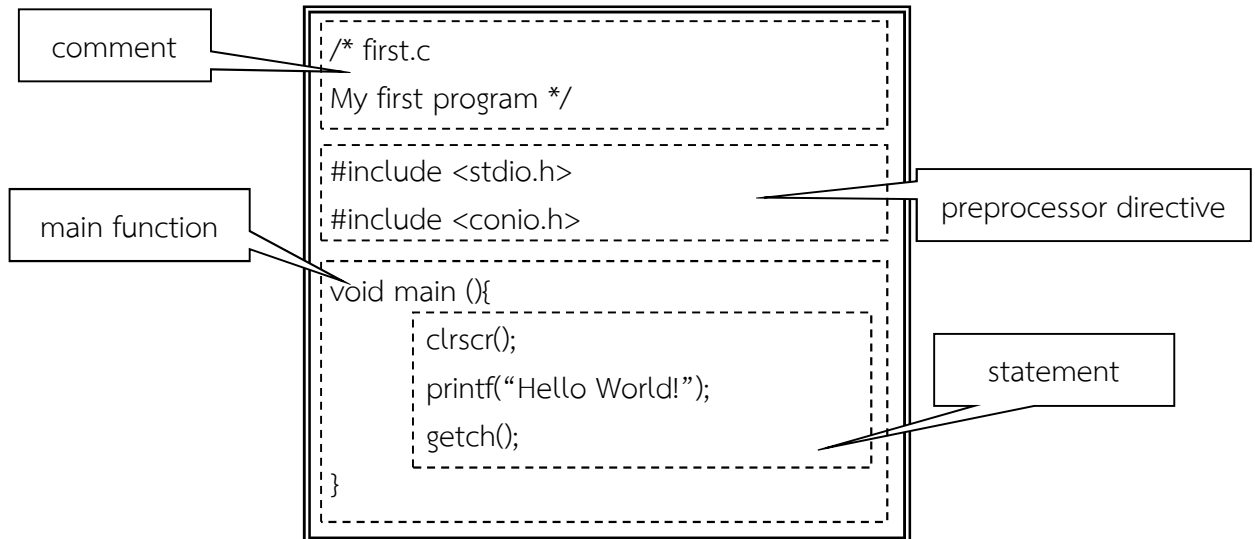
โปรแกรมภาษาซี เป็นโปรแกรมที่ประกอบไปด้วยฟังก์ชันหลากหลายฟังก์ชันที่ทำงานต่อเนื่องกัน เพื่อให้บรรลุผลตามที่นักเขียนโปรแกรมต้องการ โดยฟังก์ชันแต่ละฟังก์ชันก็จะทำงานในส่วนย่อย ๆ ที่ถูกแบ่งออกมากจากงานชิ้นใหญ่



ฟังก์ชัน (Function) คือ ชุดคำสั่งที่เขียนขึ้นเพื่อสั่งให้คอมพิวเตอร์ทำงาน สามารถรับข้อมูล (Input) ประมวลผล (Processes) และส่งออกผลข้อมูล (Output) ได้

โดยฟังก์ชันอาจจะถูกเขียนไว้แล้ว (Library Function) พร้อมใช้งาน (ฟังก์ชันเหล่านี้จะถูกจัดเก็บไว้ในไลบรารีมาตรฐาน (Standard Library)) หรืออาจจะเป็นฟังก์ชันที่ถูกเขียนขึ้นใหม่โดยนักเขียนโปรแกรมเอง (User-Defined Function) อย่างไรก็ตาม ในภาษาซีจะมีฟังก์ชันพิเศษคือ ฟังก์ชันหนึ่งที่ต้องมีไว้ในโปรแกรมเสมอ คือ ฟังก์ชัน main() ทั้งนี้ฟังก์ชันดังกล่าวจัดเป็นฟังก์ชันหลักที่นำมาใช้เป็นจุดเริ่มต้นของโปรแกรมเพื่อสั่งให้ทำงาน

แม้โปรแกรมภาษาซีแต่ละโปรแกรมจะมีทำงานต่างกัน แต่เราสามารถสรุปโครงสร้างหลักของโปรแกรมภาษาซีได้ดังนี้



### 1. ส่วนอธิบายโปรแกรม (Comments)

ใช้สำหรับเขียนอธิบายการทำงานของโปรแกรม ซึ่งข้อความในส่วนนี้ เมื่อผ่านการแปลคำสั่ง ตัวแปลคำสั่งจะไม่สนใจว่าเป็นส่วนหนึ่งของโปรแกรม เพื่อให้เข้าใจและอ่านโปรแกรมง่ายขึ้น การเขียนส่วนอธิบายโปรแกรมจะใช้เครื่องหมาย `//` หรือ `/*` และ `*/` คร่อมข้อความที่ต้องการอธิบาย ดังนี้

- 1) คำอธิบายบรรทัดเดียว ใช้เครื่องหมาย `//` นำหน้า ข้อความเริ่มตั้งแต่หลังเครื่องหมายนี้ตลอดไปจนสุดบรรทัดนั้นๆ ถือว่าเป็นคำอธิบาย

`// ..... ข้อความที่ต้องการอธิบาย`

- 2) คำอธิบายแบบหลายบรรทัด ใช้เครื่องหมาย `/*` เพื่อบ่งบอกจุดเริ่มต้น และ `*/` เพื่อบ่งบอกจุดสิ้นสุด คำอธิบายจะอยู่ระหว่างเครื่องหมายนี้ ไม่ว่าจะกี่บรรทัดก็ตาม จะเขียนได้ดังนี้

```

/* .....
..... ข้อความที่ต้องการอธิบาย....
..... */
    
```



การเขียนส่วนคำอธิบายโปรแกรมจะเขียนไว้ที่ไหนในโปรแกรมก็ได้ โดยมากจะนิยมเขียนอธิบายขั้นตอนการทำงานก่อนเขียนคำสั่ง หรือ อธิบายความหมายของคำสั่งไว้ข้างคำสั่งนั้นก็ได้

## 2. ส่วนพรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

เป็นส่วนที่ทุกโปรแกรมต้องมี ใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการ และกำหนดค่าต่าง ๆ ซึ่งจะต้องเริ่มต้นด้วยเครื่องหมายไดเรกทีฟ (#) เสมอ และตามด้วยชื่อที่ต้องการกำหนดค่า โดยในส่วนนี้จะได้รับการประมวลผลก่อนชุดคำสั่งภายใน โดย preprocessor directive ที่ใช้ในภาษาซี ได้แก่

#if	#ifdef	#ifndef	#else
#elif	#endif	#include	#define
#undef	#line	#error	#pragma

ตัวอย่าง Preprocessor directive ที่ใช้บ่อย ได้แก่ #include เป็นการบอกให้ตัวแปลคำสั่ง (compiler) อ่านไฟล์อื่นเข้ามาร่วมในการคอมไพล์ โดยไฟล์ที่อ่านนี้เรียกว่า header file

Header file เป็นไฟล์ชนิดข้อความ (Text file) ที่ภายในโปรแกรมจะมีการประกาศค่าตัวแปร และค่าคงที่ต่าง ๆ ซึ่งจะบรรจุในฟังก์ชันมาตรฐานต่าง ๆ รวมเข้าด้วยกันตามลักษณะงานเดียวกันไว้ และจัดเก็บลงในไลบรารี โดยจะถูกนำเอามาอ่านรวมกันกับชุดคำสั่งในโปรแกรมขณะคอมไพล์ โดย header file จะมีนามสกุลเป็น .h

มีรูปแบบดังนี้

```
#include<filename> หรือ
#include"filename"
```

เช่น #include<stdlib.h>

อ่านไฟล์ stdlib.h จากไดเรกทอรีที่กำหนด

```
#include"sample.h"
```

อ่านไฟล์ sample.h จากไดเรกทอรีปัจจุบันหรือที่กำหนด

o การใช้เครื่องหมาย <...> คร่อมชื่อ เพื่อบอกให้คอมไพเลอร์ค้นหาไฟล์จากไดเรกทอรีที่กำหนดใน Option directory แต่ถ้าใช้เครื่องหมาย "..." จะเป็นการกำหนดให้ค้นหาจากไดเรกทอรีปัจจุบัน และไดเรกทอรีที่กำหนดเส้นทางไว้

o ชื่อฟังก์ชันการทำงานพื้นฐานของภาษาซี ตัวแปร ค่าคงที่ และมาโครพื้นฐาน โดยปกติจะรวมกันเป็นกลุ่มในไฟล์ที่เรียกว่าไฟล์หัว (Header Files) คือ ไฟล์ที่มีนามสกุล .h ไฟล์หัวแต่ละไฟล์จะเก็บโมดูลแยกกันเป็นส่วนการทำงานแต่ละส่วน เพื่อให้ผู้เขียนโปรแกรมสามารถเรียกใช้งานได้ เช่น ในตัวอย่าง จะเห็นว่ามีการใช้ไฟล์ stdio.h ซึ่งในไฟล์ดังกล่าวมีการกำหนดคำสั่ง printf ซึ่งถูกเรียกใช้ในฟังก์ชัน main() และ ฟังก์ชัน process แต่ถ้าเราต้องการเรียกใช้ฟังก์ชัน system() ซึ่งไม่มีใน ไฟล์นี้ แต่มีในไฟล์ stdlib.h เราก็จะต้อง include ไฟล์ stdlib.h เข้ามาด้วย

ตัวอย่าง ไตรรกที่พ (Directive) การใช้งาน

#include	Include text from a file
#define	Define a macro
#undef	Undefined a macro
#if	Test if a compile-time condition holed
#ifdef	Test if a symbol is defined
#ifndef	Test if a symbol is not defined
#else	Indicate alternative if a test file fail
#line	Give a line number for compiler message

#define ทำหน้าที่ใช้กำหนดค่าคงที่ ที่เป็นชื่อแทน คำ นิพจน์ คำสั่ง หรือคำสั่งหลายคำสั่ง มีรูปแบบการใช้งานดังนี้

#define ชื่อตัวแปร (ชื่อที่ใช้แทน) ค่าที่ต้องการกำหนด

เช่น	#define TEN 10	กำหนดตัวแปร TEN แทนค่า 10
	#define PI 3.14	กำหนดตัวแปร PI แทนค่า 3.14

### 3. ส่วนฟังก์ชันหลัก (Main program)

เป็นส่วนที่โปรแกรมภาษาซีทุกโปรแกรมต้องมีในฟังก์ชัน main จะประกอบด้วยคำสั่งต่างๆ ที่ใช้ในการสั่งงานโปรแกรมเรียงต่อ ๆ กัน โดยขอบเขตของฟังก์ชันจะถูกกำหนดด้วยเครื่องหมาย “ { ” และเครื่องหมาย “ } ” นั่นคือ ส่วนคำสั่งของฟังก์ชันหลัก เริ่มที่ “ { ” และจบที่ “ } ” และในแต่ละประโยคคำสั่งจะจบลงด้วยเครื่องหมาย “ ; ” (semi colon)

ฟังก์ชันแรกสุดที่จะได้รับการประมวลผลก็คือ ฟังก์ชันหลัก นั่นคือเริ่มทำงานตามคำสั่งแรกที่อยู่ในฟังก์ชันหลัก เมื่อสิ้นสุดการทำงานตามคำสั่งแรกแล้ว ก็จะประมวลผลในคำสั่งถัดไปตามลำดับ และเสร็จสิ้นการทำงานเมื่อสิ้นสุดการทำงานตามคำสั่งสุดท้ายของฟังก์ชันหลักนี้

```
#include<stdio.h>
main()
{
    int a,b;
    scanf("%d",&a);
    scanf("%d",&b);
    printf("%d",a+b);
    getch();
}
```

} // main program

#### 4. ส่วนประโยคคำสั่ง (Statement)

เป็นชุดคำสั่งที่บรรจุอยู่ในฟังก์ชันนั้น ๆ ซึ่งอาจเป็น

- ❖ ประโยคคำสั่งสำหรับประกาศตัวแปร
- ❖ ประโยคนิพจน์คณิตศาสตร์
- ❖ ประโยคคำสั่งควบคุม
- ❖ การเรียกใช้ฟังก์ชันอื่น

โดยประโยคคำสั่งเหล่านี้จะถูกบรรจุอยู่ในภายใน { } และเมื่อสิ้นสุดประโยคใด ๆ จะต้องลงท้ายด้วย ; เสมอ

ตัวอย่างประโยคคำสั่ง ที่ใช้ในโปรแกรมนี้นี้ ได้แก่

➤ คำสั่ง clrscr() เป็นคำสั่งที่ถูกนิยามไว้ในไฟล์ conio.h เป็นคำสั่งที่ทำหน้าที่ลบการแสดงผลตัวอักขระบนหน้าจอที่แสดงอยู่ก่อนหน้าทั้งหมด แล้วกำหนดจุดเริ่มต้นของการแสดงผลตัวอักขระตัวถัดไปที่มุมบนด้านซ้ายมือ

➤ คำสั่ง printf() เป็นคำสั่งที่ถูกนิยามไว้ในไฟล์ stdio.h เป็นคำสั่งที่แสดงผลข้อความที่อยู่ระหว่างเครื่องหมาย “ และ ” ซึ่งอยู่ภายใน ( ) นอกจากตัวอักขระประเภท ตัวอักขระตัวเลข และอักขระพิเศษแล้ว ยังสามารถกำหนดการแสดงผลโดยใช้กลุ่มอักขระหลัก (escape

sequence) ซึ่งประกอบไปด้วยตัวอักษร “ \ ” แล้วตามด้วยตัวอักษรอีก 1 ตัว ซึ่งทำหน้าที่กำหนดลักษณะการแสดงผล เช่น \n หมายถึงการขึ้นบรรทัดใหม่ เป็นต้น

อักขระหลักอื่น ๆ ได้แก่

\a	Alert (bell)
\b	Back space
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab

\\	Back slash
\?	Question mark
\'	Single quote
\"	Double quote
\ooo	Octal number
\xhh	Hexadecimal number

➔ คำสั่ง getch() เป็นคำสั่งที่ถูกนิยามไว้ในไฟล์ conio.h เป็นคำสั่งที่ทำหน้าที่กำหนดให้รอรับการป้อนคีย์ใดก็ได้เพียงหนึ่งตัว

## ประเภทของข้อผิดพลาด (type of Errors)

สำหรับข้อผิดพลาดที่เกิดขึ้นจากการเขียนโปรแกรม สามารถแบ่งออกเป็น 3 ประเภท คือ

### 1. ข้อผิดพลาดที่เกิดจากไวยากรณ์ (Syntax Errors)

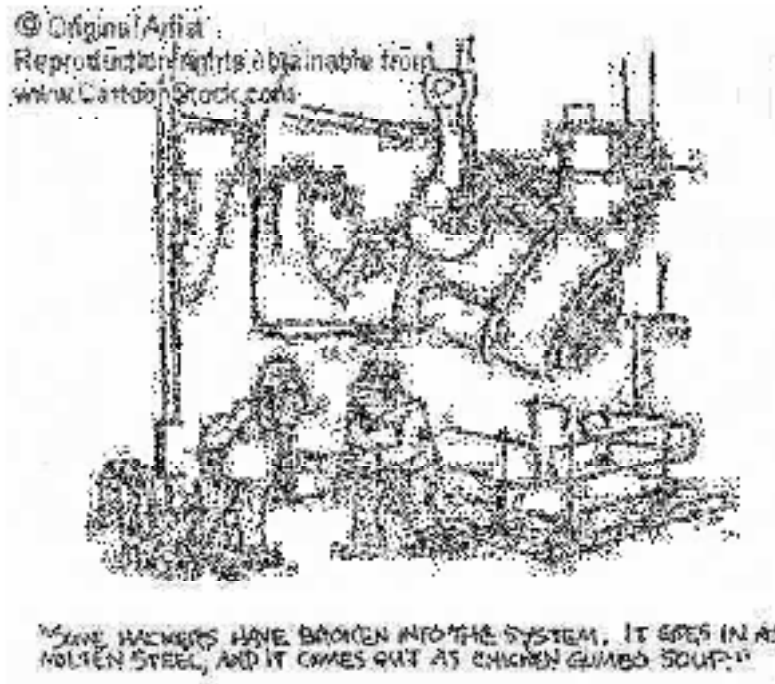
ข้อผิดพลาดชนิดนี้ เกิดจากการใช้ไวยากรณ์ หรือรูปแบบภาษาที่ผิด เช่น สะกดคำสั่งผิด เป็นต้น ซึ่งเมื่อผ่านการแปลแล้วตัวแปลภาษาก็จะไม่รู้จักคำสั่งดังกล่าว อย่างไรก็ตามข้อผิดพลาดที่เกิดจากไวยากรณ์นั้นคอมไพเลอร์สามารถตรวจพบ และแจ้งข้อผิดพลาดที่เกิดขึ้นให้ทราบได้

### 2. ข้อผิดพลาดที่เกิดจากตรรกะโปรแกรม (Logic Errors)

ข้อผิดพลาดชนิดนี้ จัดเป็นข้อผิดพลาดที่เกิดขึ้นจากตัวโปรแกรมเมอร์เอง เช่น การใช้เงื่อนไขผิดพลาด ทำให้ได้ผลลัพธ์ที่ผิดพลาด ข้อผิดพลาดชนิดนี้คอมไพเลอร์จะไม่สามารถตรวจพบได้ ดังนั้นตรรกะในการสร้างเงื่อนไข และสูตรการคำนวณต่าง ๆ จึงจำเป็นต้องได้รับการทดสอบความถูกต้องทุกครั้งก่อนนำไปใช้งานจริงเสมอ

### 3. ข้อผิดพลาดในขณะรันโปรแกรม (Runtime Errors)

กรณีพบข้อผิดพลาดชนิดนี้ในขณะที่รันโปรแกรมอยู่ โปรแกรมจะมีข้อความแจ้งให้ทราบ และจะหยุดทำงานไป เช่น เผลोकำหนดให้ตัวหรมีค่าเป็นศูนย์ ข้อผิดพลาดชนิดนี้ไม่สามารถ



ตรวจพบได้เวลาคอมไพล์ แต่เมื่อมีการรันโปรแกรมนี้เมื่อใด ก็จะมีข้อผิดพลาดแจ้งให้ทราบทางแก้ไขก็คือ จะต้องกลับไปแก้ไข source code ในโปรแกรมให้ถูกต้องเสียก่อน

### แบบฝึกทักษะที่ 2.1

1. จากโปรแกรม ให้นักเรียนอธิบายคำสั่งในโปรแกรมแต่ละบรรทัด ว่าใช้เพื่อสั่งให้คอมพิวเตอร์ทำงานอะไร

1:	#include<stdio.h>
2:	main()
3:	{
4:	printf("This is my first program");
5:	getch();
6:	}

#### อธิบายโปรแกรม

บรรทัดที่ 1 :

.....

บรรทัดที่ 2 :

.....

บรรทัดที่ 3 :

.....

บรรทัดที่ 4 :

.....

บรรทัดที่ 5 :

.....

บรรทัดที่ 6 :

.....



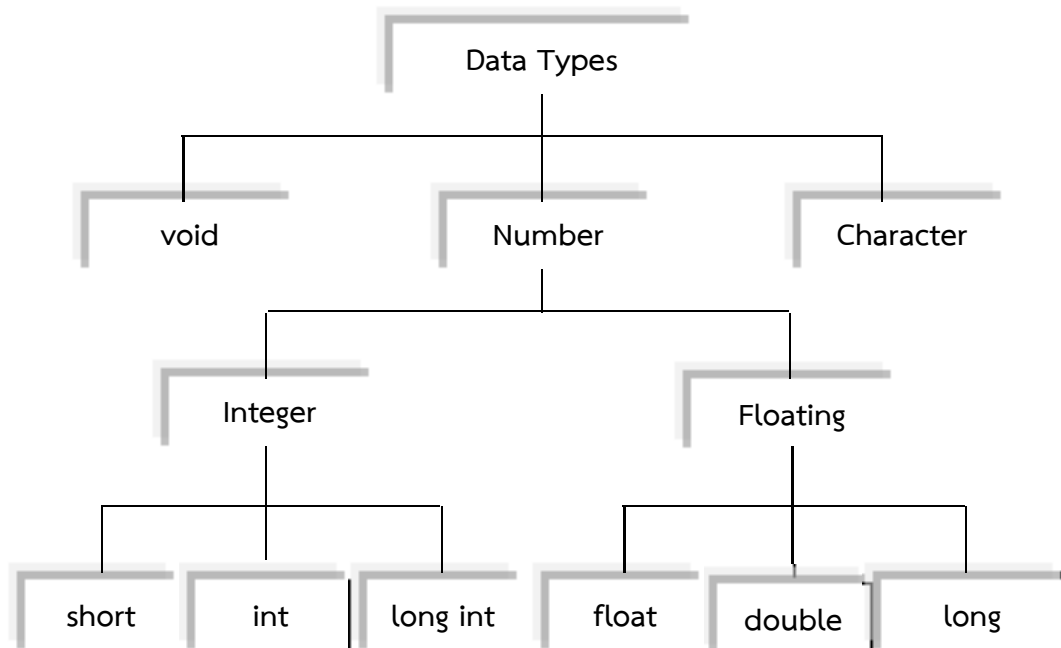
## ข้อมูล (Data)

ข้อมูล คือ ข้อเท็จจริงที่เราสนใจ ที่ผ่านการจัดเก็บ รวบรวม และป้อนเข้าสู่ระบบ เพื่อนำไปประมวลผล ให้ได้เป็นสิ่งที่ผู้ใช้ต้องการต่อไป

ข้อมูลจะอยู่ในรูปแบบใดก็ได้ เช่น ตัวเลข ตัวอักษร อักขระ ภาพ เสียง เป็นต้น

## ชนิดของข้อมูลในภาษาซี (Data Types)

ข้อมูลที่ใช้ในการเขียนโปรแกรมมีหลายชนิด ซึ่งนักเขียนโปรแกรมต้องเลือกใช้ตามความเหมาะสมกับการใช้งาน ข้อมูลมีขนาดที่แตกต่างกันไปตามชนิดข้อมูล นอกจากนี้แล้ว ชนิดข้อมูลยังอาจมีขนาดที่แตกต่างกัน โดยขึ้นอยู่กับเครื่องคอมพิวเตอร์และตัวแปรโปรแกรมที่ใช้ในการประมวลผล



### ❖ ข้อมูลชนิด void

ข้อมูลชนิดนี้จะไม่มีความหมาย ซึ่งเราจะใช้กำหนดไว้ในฟังก์ชันในกรณีไม่ต้องการให้ฟังก์ชันมีการรับค่าใด ๆ เข้ามา หรือส่งค่าใด ๆ กลับไป ใช้พื้นที่เก็บ 0 bit

### ❖ จำนวนเต็ม (Integer)

เป็นชนิดข้อมูลแบบเลขจำนวนเต็ม เช่น -10, -2, 0, 23, 164 เป็นต้น เรียกอีกอย่างว่า Integer Number ซึ่งภาษาซีแบ่งข้อมูลชนิดนี้ออกเป็น 3 ประเภท คือ



- 1) **short int** เป็นเลขจำนวนเต็มแบบสั้น
- 2) **int** เป็นเลขจำนวนเต็มแบบปกติ
- 3) **long int** เป็นเลขจำนวนเต็มแบบยาว

รายละเอียดของข้อมูลประเภทจำนวนเต็มชนิดต่าง ๆ สามารถสรุปได้ดังตาราง

ประเภท	จำนวนไบต์	ค่าต่ำสุด	ค่าสูงสุด
short int	2 (16 bits)	-32,786	32,767
int (16 bits)	2 (16 bits)	-32,768	32,767
int (32 bits)	4 (32 bits)	-2,147,483,648	2,147,483,647
long int	4 (32 bits)	-2,147,483,648	2,147,483,647

จะเห็นว่าข้อมูลแบบเลขจำนวนเต็มมีอยู่หลายประเภท การจะเลือกใช้ชนิดของข้อมูลประเภทใดนั้น ขึ้นอยู่กับความเหมาะสม เช่น หากข้อมูลที่ต้องการเก็บมีขนาดใหญ่ไม่มากนัก อาจเลือกชนิดข้อมูลประเภท short int มาใช้ในการเก็บข้อมูล แต่ถ้าหากข้อมูลที่ต้องการเก็บมีขนาดใหญ่เกินกว่าที่จะเก็บลงในข้อมูลประเภท short int ได้ ก็อาจเลือกเก็บลงในประเภท int หรือ long int แทน

#### ❖ เลขทศนิยม (floating point)

เป็นชนิดข้อมูลแบบเลขทศนิยม หรือบางที่เรียกว่า จำนวนจริง (Real Number) เช่น 10.0, 14.99, 29.000000000 เป็นต้น

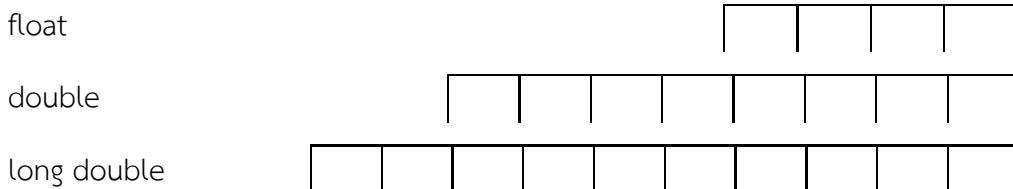
ภาษาซีแบ่งข้อมูลแบบตัวเลขทศนิยมออกเป็น 3 ประเภท คือ

- 1) float
- 2) double
- 3) long double

รายละเอียดของข้อมูลประเภทเลขทศนิยมชนิดต่าง ๆ สามารถสรุปได้ดังตาราง

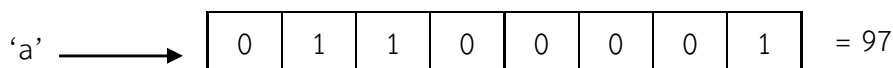
ประเภท	จำนวนไบต์	ค่าต่ำสุด	ค่าสูงสุด
float	4 (32 bits)	$3.4 \times 10^{-38}$	$3.4 \times 10^{38}$
double	8 (64 bits)	$1.7 \times 10^{-308}$	$1.7 \times 10^{308}$
long double	10 (80 bits)	$3.4 \times 10^{-4932}$	$3.4 \times 10^{4932}$

จะสังเกตเห็นได้ว่าทั้ง float, double และ long double ต่างก็เก็บข้อมูลในลักษณะของเลขทศนิยมเหมือนกัน เพียงแต่ long double สามารถเก็บข้อมูลได้ละเอียดกว่า double และ double สามารถเก็บข้อมูลที่มีช่วงค่าได้มากกว่าและละเอียดกว่า float



### ❖ ตัวอักษร (Character)

คือชนิดข้อมูลประเภท char ซึ่งชนิดของข้อมูลในรูปแบบนี้ใช้พื้นที่ 1 ไบต์ (8 บิต) ต่อข้อมูล 1 ตัวอักษรเท่านั้น ผู้เขียนโปรแกรมอาจคิดว่าคอมพิวเตอร์ทำการเก็บข้อมูลชนิดนี้ในรูปแบบของตัวอักษร เช่น 'a' แต่จริงๆ แล้วคอมพิวเตอร์จะเก็บตัวอักษร 'a' นี้ในลักษณะของรหัสแอสกี (ASCII : American Standard Code for Information Interchange) เช่น คอมพิวเตอร์จะเก็บตัว 'a' เป็น 0110 0001 ในระบบเลขฐานสอง (มีค่า 97 ในระบบเลขฐานสิบ) ซึ่งเป็นรหัสแอสกีของตัวอักษร a นั่นเอง



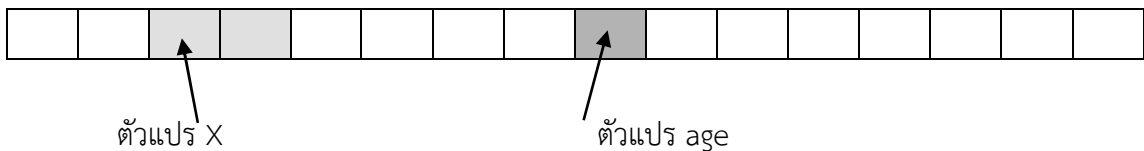
จากการที่คอมพิวเตอร์เก็บข้อมูลชนิดตัวอักษรในรูปแบบของรหัสแอสกี 1 ไบต์นั้น ทำให้ข้อมูลชนิดตัวอักษรเทียบได้กับข้อมูลชนิดตัวเลขจำนวนเต็ม มีค่าตั้งแต่ 0 ถึง 255 สำหรับ unsigned char และมีค่าตั้งแต่ -128 ถึง 127 สำหรับ signed char

### ตัวแปร (Variables)

คอมพิวเตอร์มีส่วนประกอบที่สำคัญส่วนหนึ่งคือ หน่วยความจำ หน่วยความจำเปรียบได้กับสมองของมนุษย์ทำหน้าที่เก็บข้อมูลในขณะที่ประมวลผล ในการประมวลผลแต่ละครั้งมักต้องใช้ข้อมูลจำนวนมาก เช่น ชื่อนักเรียน คะแนนเก็บ คะแนนสอบกลางภาค คะแนนสอบปลายภาค จำนวนหน่วยกิต คะแนนเฉลี่ย เป็นต้น ซึ่งจำเป็นจะต้องนำข้อมูลเหล่านี้ไปเก็บไว้ในหน่วยความจำ และเมื่อเก็บแล้วจะต้องทราบตำแหน่งที่นำข้อมูลเข้าไปเก็บไว้ภายในของหน่วยความจำด้วย เพื่อให้สามารถนำข้อมูลเหล่านั้นกลับมาประมวลผลได้ ในภาษาคอมพิวเตอร์ระดับสูง เพื่อความสะดวกในการเขียนโปรแกรม การจดจำตำแหน่งที่ใช้ในการเก็บข้อมูล จะทำโดยการตั้งชื่อให้กับตำแหน่ง

ของหน่วยความจำที่ใช้เก็บข้อมูลนั้น จากนั้นระบบปฏิบัติการจะช่วยจัดการในการหาตำแหน่งที่อยู่  
ที่แท้จริงของข้อมูลต่อไป

หน่วยความจำ



ตัวแปร จึงเป็นชื่อของหน่วยความจำในตำแหน่งใด ๆ เมื่อนำข้อมูลไปเก็บไว้ในหน่วยความจำ  
ตำแหน่งนั้น จะกล่าวว่า ตัวแปรนั้นมีค่าเท่ากับข้อมูลที่เก็บไว้ ตัวแปรสามารถเก็บค่าชนิดต่าง ๆ  
ตามที่ได้ประกาศไว้ในโปรแกรมเท่านั้น เช่น ตัวแปรที่ใช้เก็บข้อมูลชนิดจำนวนเต็ม ตัวแปรที่ใช้เก็บ  
ข้อมูลชนิดจำนวนจริง ตัวแปรที่ใช้เก็บข้อมูลชนิดตัวอักษร เป็นต้น ตัวแปรเหล่านี้จะไม่สามารถเก็บค่า  
ชนิดอื่น ๆ นอกเหนือจากชนิดที่ประกาศไว้ และค่าที่เก็บไว้ในตัวแปรนี้สามารถเปลี่ยนค่าได้  
ตลอดเวลา ขึ้นกับข้อคำสั่งภายในโปรแกรม

การกำหนดตัวแปรทำได้ 2 แบบ คือ

- 1) กำหนดไว้นอกกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Global Variable กำหนดไว้นอก  
ฟังก์ชันใช้งานได้ทั้งโปรแกรม มีค่าเริ่มต้นเป็น 0 (กรณีไม่ได้กำหนดค่าเริ่มต้น)
- 2) กำหนดไว้ในกลุ่มคำสั่ง หรือฟังก์ชัน เรียกตัวแปรนี้ว่า Local Variable กำหนดไว้ภายใน  
ฟังก์ชันใช้งานได้ภายในฟังก์ชันนั้น และไม่ถูกกำหนดค่าเริ่มต้นโดยอัตโนมัติ

#### ❖ กฎการตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรจำเป็นต้องคำนึงถึงกฎเกณฑ์ต่าง ๆ ดังรายละเอียดดังนี้

- 1) สามารถนำตัวอักษร ไม่ว่าจะเป็นอักษรตัวพิมพ์ใหญ่ A - Z อักษรตัวพิมพ์  
เล็ก a - z และตัวเลข 0 - 9 รวมถึงเครื่องหมาย underscore ( \_ ) มาใช้เพื่อการตั้งชื่อตัวแปรได้  
แต่มีข้อแม้ว่า อักษรตัวแรกจะต้องเป็นตัวอักษร หรือ underscore เท่านั้น
- 2) ชื่อที่ตั้งขึ้นเขียนเป็นตัวเล็ก ตัวใหญ่ หรือตัวใหญ่ปนตัวเล็กจะถือว่าเป็นคนละ  
ชื่อกันหมด เช่น count, COUNT, Count จะถือเป็น 3 ชื่อตัวแปรที่แตกต่างกัน
- 3) ชื่อตัวแปรจะต้องไม่ตรงกับคำสงวน (Reserved word)

คำสงวน หมายถึง คำที่สงวนไว้สำหรับเรียกใช้ตามวัตถุประสงค์ที่กำหนดไว้  
เฉพาะ เช่น คำที่ใช้ใน คำสั่งควบคุม และชนิดของข้อมูล เป็นต้น ซึ่งประกอบด้วย

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

สำหรับบางคอมไพเลอร์ ก็จะรวมคำสั่งบางตัวต่อไปนี้เข้าไปด้วย

ada	asm	entry	far	fortran
huge	near	pascal		

อย่างไรก็ตาม หากจำเป็นต้องใส่ชื่อให้ตรงกับคำสั่งอื่น ก็สามารถทำได้ด้วยการใช้ตัวอักษรตัวพิมพ์ใหญ่ช่วย เช่น Auto, Register เป็นต้น

4) สามารถกำหนดความยาวของตัวแปรไม่จำกัด แต่จะพิจารณาความแตกต่างของตัวอักษร 31 ตัวแรก

5) ชื่อตัวแปรไม่อนุญาตให้มีช่องว่าง ดังนั้นหากต้องการแยกคำ สามารถใช้เครื่องหมาย underscore ช่วยได้

6) ควรตั้งชื่อตัวแปรที่สามารถสื่อความหมาย เพื่อบอกให้รู้ว่าตัวแปรนั้น ๆ นำไปใช้เพื่อการใด

ชื่อตัวแปรที่ถูกต้อง	ชื่อตัวแปรที่ผิด
Count	1count
Num12	num !
m_mum	m...mun

#### ❖ การประกาศตัวแปร

ความสัมพันธ์ระหว่างตัวแปรและชนิดข้อมูลจะแสดงออกมาในรูปของการประกาศตัวแปร กล่าวคือ ในการประกาศตัวแปรขึ้นใช้งาน จำเป็นอย่างยิ่งที่จะต้องกำหนดชนิดของข้อมูลให้กับตัวแปรนั้นด้วยการจะกำหนดชนิดข้อมูลให้กับตัวแปรใด ๆ นั้น ต้องพิจารณาให้ได้ว่าต้องการ

เก็บข้อมูลในลักษณะใด แล้วเลือกชนิดข้อมูลให้เหมาะสม เพราะถ้าหากเลือกข้อมูลที่มีขนาดเล็กเกินไปก็จะเก็บข้อมูลได้ก็เกิดข้อผิดพลาด และหากเลือกชนิดข้อมูลใหญ่เกินไปกว่าขนาดของข้อมูลที่ต้องการเก็บจริง ๆ มาก ๆ ก็จะทำให้สิ้นเปลืองหน่วยความจำโดยเปล่าประโยชน์ ซึ่งจะเห็นว่าการเลือกกำหนดชนิดของข้อมูลให้กับตัวแปรนั้นมีความสำคัญมาก จึงควรพิจารณาอย่างรอบคอบ

รูปแบบการประกาศตัวแปร มีลักษณะดังนี้

ชนิดข้อมูล ชื่อตัวแปร[,ชื่อตัวแปร];

เช่น

```
char a;           // ประกาศตัวแปร a เป็นตัวแปรชนิดตัวอักษร
int b;           // ประกาศตัวแปร b เป็นเลขจำนวนเต็มประเภท int
float c,d;       // ประกาศตัวแปร c และ d เป็นจำนวนจริงประเภท float
```

**ข้อสังเกต** การกำหนดชนิดของตัวแปร มีสิ่งที่ควรพิจารณาอยู่ 2 ประการ คือ ตัวแปรนั้นจะต้องสามารถรับค่าได้ทุกค่าโดยไม่เกินขอบเขตของข้อมูลชนิดนั้น และตัวแปรจะต้องไม่ใช่หน่วยความจำมากเกินไปจนความจำเป็น เช่น ถ้าข้อมูลไม่เกินขอบเขตของ int ก็ไม่ควรกำหนดตัวแปรให้เป็น float

#### ❖ การกำหนดค่าให้กับตัวแปร

การกำหนดค่าให้กับตัวแปร เป็นการนำค่าที่ต้องการนำไปเก็บไว้ในหน่วยความจำที่ตัวแปรนั้น ๆ ครอบครองอยู่ โดยเมื่อตัวแปรถูกประกาศแล้ว เราสามารถกำหนดค่าให้กับตัวแปรได้ดังนี้

รูปแบบการกำหนดค่าให้กับตัวแปร มีลักษณะดังนี้

ชื่อตัวแปร = ค่าของตัวแปร;

เช่น

```
a = 'b';         // กำหนดให้ตัวแปร a มีค่าเท่ากับ ตัวอักษร b
b = 10;          // กำหนดให้ตัวแปร b มีค่าเท่ากับ 10
c = 4.5;         // กำหนดให้ตัวแปร c มีค่าเท่ากับ 4.5
```

เราสามารถประกาศตัวแปร และกำหนดค่าเริ่มต้นให้กับตัวแปรในคำสั่งเดียวได้ ดังนี้

รูปแบบการประกาศและกำหนดค่าเริ่มต้นให้กับตัวแปร มีลักษณะดังนี้

**ชนิดข้อมูล ชื่อตัวแปร = ค่าของตัวแปร;**

เช่น

```
char a = 'b';
```

```
/*ประกาศตัวแปร a เป็นตัวแปรชนิดตัวอักษร
และกำหนดให้ตัวแปร a มีค่าเท่ากับ ตัวอักษร
b */
```

```
int b = 10;
```

```
/*ประกาศตัวแปร b เป็นจำนวนเต็มประเภท int
และกำหนดให้ตัวแปร b มีค่าเท่ากับ 10 */
```

```
float c = 4.5,d = 2.0;
```

```
/* ประกาศตัวแปร c และ d เป็นจำนวนจริงประเภท
float แล้วกำหนดให้ตัวแปร c มีค่าเท่ากับ 4.5 และ
ตัวแปร d มีค่าเท่ากับ 2.0 */
```

การกำหนดค่าให้กับตัวแปรจะต้องกำหนดให้เป็นไปตามกฎ และต้องสัมพันธ์กับชนิดข้อมูลที่ประกาศไว้เป็นสำคัญ เช่น หากกำหนดตัวแปรเป็นชนิดข้อมูล int ค่าที่กำหนดให้กับตัวแปรก็ต้องเป็นเลขจำนวนเต็มเท่านั้น ไม่สามารถกำหนดให้เป็นเลขทศนิยมหรือข้อความได้ เป็นต้น

#### ➤ การกำหนดค่าชนิดจำนวนเต็ม กฎเกณฑ์การกำหนดค่าชนิดเลขจำนวนเต็ม มีดังนี้

- 1) ค่าตัวเลขจะต้องไม่มีทศนิยม
- 2) สามารถเป็นไปได้ทั้งค่าบวก หรือค่าลบ
- 3) สำหรับค่าบวกไม่จำเป็นต้องใส่เครื่องหมาย + นำหน้าค่า
- 4) ห้ามใช้เครื่องหมาย , หรือช่องว่าง กำกับระหว่างตัวเลข
- 5) ค่าของตัวเลขจะต้องไม่เกินช่วงค่าที่ชนิดของตัวแปรจะรับได้ เช่น ค่าที่สามารถกำหนดให้ตัวแปรประเภท short int มีค่าอยู่ระหว่าง - 32768 ถึง 32767 เป็นต้น
- 6) สามารถใช้ suffix ต่อท้ายค่าได้ เช่น ใช้ตัวอักษร L ต่อท้ายค่าเลขจำนวนเต็มแบบยาว เป็นต้น

ตัวอย่าง            n1 = 345;  
                          n2 = -1243;  
                          n3 = 45134L; //long int

➤ **การกำหนดค่าชนิดจำนวนจริง** กฎเกณฑ์การกำหนดค่าชนิดเลขจำนวนจริง มีดังนี้

- 1) ค่าตัวเลขสามารถมีจุดทศนิยมได้
- 2) สามารถเป็นไปได้ทั้งค่าบวก หรือค่าลบ
- 3) สำหรับค่าบวกไม่จำเป็นต้องใส่เครื่องหมาย + นำหน้าค่า
- 4) สามารถกำหนดค่าแบบเอ็กซ์โปเนนต์เซียลได้ด้วยการใส่ตัวอักษร E ต่อท้ายค่า เช่น 4.1E3
- 5) ค่าของตัวเลขจะต้องไม่เกินช่วงค่าที่ชนิดของตัวแปรจะรับได้
- 6) สำหรับค่าเลขจำนวนจริงชนิด double จะใช้ F ต่อท้ายค่า และใช้ L ต่อท้ายค่าที่กำหนดชนิดข้อมูลเป็น long double

ตัวอย่าง

```
r1 = 34.5;  
r2 = 3.4E-1;  
r3 = 3.0F;                //double  
r4 = 9.45L;              //long double
```

➤ **การกำหนดค่าชนิดตัวอักษร** กฎเกณฑ์การกำหนดค่าชนิดตัวอักษร มีดังนี้

- 1) สามารถกำหนดตัวอักษรได้โดยตรง แต่จะต้องเป็นอักขระเพียงตัวเดียว และจะต้องอยู่ในเครื่องหมาย single quote เช่น 'a'
- 2) สามารถกำหนดตัวอักษรที่เป็น escape sequence ได้ แต่จะต้องอยู่ในเครื่องหมาย single quote เช่น '\t'
- 3) สามารถกำหนดค่าเป็นจำนวนเต็มซึ่งมีค่าเท่ากับ ASCII ของตัวอักษรได้ เช่น 97 ซึ่งเป็นค่าของตัวอักษร a
- 4) ตัวแปรสามารถกำหนดค่าตัวอักษรได้สูงสุด 1 ตัวอักษรเท่านั้น

ตัวอย่าง

```
c1 = 'a';  
c2 = '\b';  
c3 = 65;                //'A'  
c4 = '3';
```

### ❖ การเรียกใช้ค่าของตัวแปร

เมื่อกำหนดค่าให้กับตัวแปรแล้ว เราสามารถเรียกใช้ค่าที่ถูกเก็บอยู่ในตัวแปรโดยใช้ชื่อของตัวแปร ได้โดยตรง อย่างไรก็ตาม ถ้าตัวแปรนั้นยังไม่เคยถูกกำหนดค่าเลย ค่าที่ถูกนำมาใช้จำเป็นค่าที่ค้างอยู่ในหน่วยความจำ ซึ่งเราไม่สามารถคาดเดาได้ ซึ่งมักเรียกว่า “ค่าขยะ” หรือ “Garbage value” เราจึงต้องกำหนดค่าตัวแปรก่อนที่จะมีการเรียกใช้ทุกครั้ง

### ค่าคงที่

ค่าคงที่ (Constant) คือ ค่าของข้อมูลที่มีจำนวนแน่นอน เป็นค่าที่ไม่สามารถเปลี่ยนแปลงได้ขณะรันโปรแกรม

การประกาศค่าคงที่มีลักษณะดังนี้

```
const ชนิดของข้อมูล ชื่อค่าคงที่ = ค่า;
```

### ตัวอย่าง

```
const float Pi = 3.1415;
```

หมายความว่า Pi เป็นค่าคงที่ชนิด float ซึ่งมีค่าเท่ากับ 3.1415

การประกาศค่าคงที่แบ่งตามชนิดต่าง ๆ ได้ดังนี้

#### 1) ค่าคงที่ชนิดจำนวนเต็ม

ค่าคงที่ชนิดจำนวนเต็ม (integer constant) เป็นค่าคงที่ชนิดตัวเลข ซึ่งอาจจะเป็นค่าบวก 0 หรือ ค่าลบ ค่าคงที่ชนิดนี้มี 3 ประเภท คือ ค่าคงที่อินทิจอร์ฐานสิบ ฐานแปดและฐานสิบหก ซึ่งมีรายละเอียดดังนี้

- ค่าคงที่ชนิดจำนวนเต็มฐานสิบ (decimal integer constant) หมายถึงชุดของเลขฐานสิบ ( 0 ถึง 9 ) ซึ่งเลขตัวแรกไม่เป็น 0 ดังตัวอย่างต่อไปนี้

```
const int price = 1000;
```

```
const int WeekDay = 7;
```



- ค่าคงที่ชนิดจำนวนเต็มฐานแปด (octal integer constant) หมายถึง ชุดของเลขฐานแปด (0 ถึง 7) ซึ่งเลขตัวแรกเป็น 0 ดังตัวอย่างต่อไปนี้

```
const int Oct1 = 01000;
```

```
const int Oct2 = 037;
```

- ค่าคงที่ชนิดจำนวนเต็มสิบหก (hexadecimal integer constant) หมายถึง ชุดของเลขฐานสิบหก (0 ถึง 9 และ A ถึง F) ซึ่งนำหน้าด้วย 0X หรือ 0x ดังตัวอย่าง ต่อไปนี้

```
const int Hex1 = 0xF78;
```

```
const int Hex2 = 0xFF;
```

## 2) ค่าคงที่ชนิดจำนวนจริง

ค่าคงที่ชนิดจำนวนจริง (floating point constant) เป็นค่าคงที่ชนิดตัวเลขที่มีทศนิยม ค่าคงที่ชนิดนี้มี 2 รูปแบบดังนี้

- เป็นชุดของตัวเลขที่ประกอบด้วยเลข 0 ถึง 9 และจุดทศนิยม ดังตัวอย่างต่อไปนี้

```
const float Pi = 3.1415;
```

```
const float Interest;
```

- เป็นเลขยกกำลังซึ่งมีส่วนประกอบ 2 ส่วนคือ ส่วนหนึ่งเป็น integer หรือ float อีกส่วนหนึ่งเป็นตัวคูณสิบยกกำลัง ซึ่งแสดงด้วยอักษร E (หรือ e) ตามด้วยเลขยกกำลังซึ่ง เป็น integer บวกหรือลบดังตัวอย่างต่อไปนี้

```
const float A = 1.0E2;
```

หมายถึง  $1.0 * 10^2$  ซึ่ง = 100.0

```
const float A = 1E+3;
```

หมายถึง  $1 * 10^3$  ซึ่ง = 1000.0

```
const float A = 1E-2;
```

หมายถึง  $1 * 10^{-2}$  ซึ่ง = 0.01

## 3) ค่าคงที่ชนิดอักษร

ค่าคงที่ชนิดอักษร(character constant) เป็นค่าคงที่ที่ประกอบด้วยอักษร 1 ตัวอยู่ในเครื่องหมาย ( ' ' ) ดังตัวอย่างต่อไปนี้

```
const char A = 'a';
```

```
const char B = 'b';
```

ค่าคงที่ชนิดอักษรมีชนิดเป็น char ค่าของค่าคงที่จะเป็นตัวเลขซึ่งมีค่าเท่ากับรหัส ASCII ของอักษรนั้น เช่น 'A' = 65, 'B' = 66

ค่าคงที่ชนิดอักษรอีกประเภทหนึ่งมีชื่อว่า เอสเคปซีควเอนซ์ (escape sequence) ซึ่งเป็นค่าคงที่ ที่ประกอบด้วยเครื่องหมาย \ และอักษรอยู่ในเครื่องหมาย ( ' ' ) เช่น '\n', '\t' ค่าคงที่ประเภทนี้มีจำนวนจำกัดและมีความหมายเฉพาะดังแสดงในตาราง

ตารางค่าคงที่แบบต่าง ๆ

escape sequence	ชื่อ	ความหมาย
\a	Alarm bell	เตรียมพร้อม (เสียงระฆัง)
\b	Backspace	เลื่อนกลับ
\f	Formfeed	ขึ้นหน้าใหม่
\n	Newline	ขึ้นบรรทัดใหม่
\r	Return	เคอเซอร์กลับไปอยู่ที่ต้นบรรทัด
\t	Tab	เว้นระยะในแนวระดับ
\\	Backslash	การกดยุ่เครื่องหมาย \
\'	Single quote	การกดยุ่เครื่องหมาย '
\"	Double quote	การกดยุ่เครื่องหมาย "
\?	Question mark	การกดยุ่เครื่องหมาย ?

## แบบฝึกทักษะที่ 2.2

1. ให้นักเรียนพิจารณาชื่อตัวแปรต่อไปนี้ ว่าถูกต้องตามกฎการตั้งชื่อตัวแปรของภาษาซีหรือไม่ ถ้าไม่ถูกต้องให้เหตุผลด้วยว่าเพราะอะไร

_____ 1.1) Golden Compass	เหตุผล _____
_____ 1.2) HELLO_	เหตุผล _____
_____ 1.3) Total-Number	เหตุผล _____
_____ 1.4) 7eleven	เหตุผล _____
_____ 1.5) float	เหตุผล _____
_____ 1.6) &volts	เหตุผล _____
_____ 1.7) _ADDR	เหตุผล _____
_____ 1.8) NUM#1	เหตุผล _____
_____ 1.9) X2	เหตุผล _____
_____ 1.10) int	เหตุผล _____

2. ครูบรรณีย์ต้องการเขียนโปรแกรมหาคะแนนรวมของนักเรียนแต่ละคน ที่เรียนวิชาการเขียนโปรแกรมภาษาซี โดยที่โปรแกรมจะต้องกำหนดให้มีการป้อนรหัสนักเรียนซึ่งเป็นเลข 5 หลัก ชื่อ - นามสกุล และคะแนนของนักเรียนคนนั้น ๆ ซึ่งประกอบด้วย คะแนนการบ้าน (เต็ม 60 คะแนน) คะแนนสอบกลางภาค (เต็ม 20 คะแนน) และคะแนนสอบปลายภาค (เต็ม 20 คะแนน) จากนั้นทำการประมวลผลหาคะแนนรวมซึ่งมีคะแนนเต็ม 100 คะแนน และแสดงผลข้อมูลต่าง ๆ เหล่านี้ รวมทั้งให้มีการแสดงผลการเรียนของนักเรียนคนนั้น โดยมีเงื่อนไขดังนี้

ถ้าคะแนนรวม > 50 คะแนน ให้แสดงข้อความ “You’re Pass”  
มิฉะนั้น ให้แสดงข้อความ “You’re Fail”

จากความต้องการของโปรแกรมหาดังกล่าวงานดังกล่าวให้นักเรียน กำหนดตัวแปร และชนิดของตัวแปรที่ใช้ในการเขียนโปรแกรมอย่างเหมาะสม พร้อมทั้งให้เหตุผลประกอบ

รายการข้อมูล	ชนิดข้อมูล	การประกาศตัวแปร	เหตุผล
รหัสนักเรียน	Char	Char std_id[6];	- รหัสนักเรียน เก็บข้อมูลเป็นชนิดตัวอักษร เพราะไม่ต้องนำไปใช้ในการคำนวณ - กำหนดให้มีขนาด 6 เพราะใช้เก็บรหัสนักเรียน 5 ตัว และอีก 1 ตัว ใช้เก็บ \0 - ตั้งชื่อตัวแปรว่า std_id เพื่อสื่อความหมายว่าเป็นรหัสนักเรียน เพราะ std ย่อมาจาก student และ id ย่อมาจาก identification

3. ให้นักเรียนกำหนดชนิดข้อมูล และชื่อตัวแปรที่เหมาะสม เพื่อใช้เก็บข้อมูลต่อไปนี้

รายการข้อมูล	ชนิดข้อมูลของตัวแปร	ชื่อตัวแปร
1) เลขจำนวนเต็ม เก็บค่าข้อมูลไม่เกิน 25,000		
2) ความสูง มีหน่วยเป็นเซนติเมตร		
3) นามสกุล ความยาวไม่เกิน 30 ตัวอักษร		
4) เลขที่บัตรประจำตัวประชาชน เป็นเลข 13 หลัก		
5) จำนวนพนักงานในบริษัทแห่งหนึ่ง ซึ่งมีไม่เกิน 150 คน		
6) คะแนนสอบของนักเรียน ซึ่งไม่เกิน 100 คะแนน		
7) ค่าใช้จ่ายประจำเดือน		
8) เกรดเฉลี่ยสะสม		
9) ดอกเบี้ยเงินฝากธนาคาร		
10) ยอดขายสินค้าของร้านค้าแห่งหนึ่ง		

## คำสั่งแสดงผลโปรแกรมภาษาซี (Output Function)

1. การส่งข้อความ หรือค่าในตัวแปรให้แสดงผลบนจอภาพ มีคำสั่งพื้นฐานที่ใช้ในโปรแกรมภาษาซี คือ printf() ซึ่งเป็นฟังก์ชันที่อยู่ในเฮดเดอร์ไฟล์ stdio.h มีรูปแบบคำสั่งคือ

```
int printf (<control>, <argument list>);
```

ฟังก์ชัน printf() จะส่งค่าข้อมูลออกไปยังจอภาพ พร้อมทั้งส่งผลการทำงานกลับมา การแสดงค่าของตัวแปร สามารถแสดงได้โดยกำหนดข้อความและรูปแบบการแสดงผลไว้ระหว่างเครื่องหมาย “ ” และเครื่องหมาย “ ” และระบุชื่อที่ต้องการแสดงผลไว้หลังเครื่องหมาย , เมื่อรันโปรแกรม ค่าของตัวแปรจะถูกนำมาแสดงแทนที่รูปแบบในตำแหน่งที่กำหนดไว้ เนื่องจากฟังก์ชันในภาษาซีไม่จำเป็นต้องมีตัวแปรมารับค่าผลในการทำงาน โดยทั่วไปเราจึงเห็นฟังก์ชันนี้อยู่ในรูปที่ไม่ต้องมีตัวแปรมารับค่า ซึ่งอยู่ในรูปของคำสั่ง(Statements) นั่นเอง

argument list	เป็นค่าตัวแปร ค่าคงที่ หรือนิพจน์ที่ต้องการนำมาแสดงผล ถ้ามีมากกว่า 1 ค่า จะแยกกันโดยใช้เครื่องหมายคอมมา ( , )
control	จะต้องเขียนอยู่ภายใต้เครื่องหมาย “ ” ซึ่งสามารถเขียนได้ 2 ลักษณะ คือ <ol style="list-style-type: none"> <li>1) เป็นข้อความที่ต้องการให้แสดงออกมา เช่น printf(“Sum of X = ”);</li> <li>2) เป็นรหัสรูปแบบ (Format Code) ที่ใช้ในการแสดงผลซึ่งทุกรหัส รูปแบบจะต้องอยู่ตามหลังเครื่องหมาย % รหัสรูปแบบข้อมูลที่นิยมใช้ ได้แก่</li> </ol>

รหัส	ความหมาย
% c	ใช้กับตัวแปรที่เก็บค่าที่เป็นตัวอักษรเพียงตัวเดียว
% s	ใช้กับตัวแปรที่เก็บค่าที่เป็นข้อความที่เก็บในตัวแปรชุด
% d	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็ม
% u	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็มบวก
% f	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยม
% e	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยมในรูปแบบ e ยกกำลัง
% x	ใช้กับตัวแปรที่เก็บค่าที่เป็นค่าเลขฐานสิบหก
% o	ใช้กับตัวแปรที่เก็บค่าที่เป็นค่าเลขฐานแปด
% p	ใช้กับตัวแปรที่เก็บค่าที่เป็นตัวชี้ตำแหน่ง (pointer)

ถ้าต้องการแสดงค่าของตัวแปรมากกว่า 1 ตัวแปร ให้คั่นระหว่างตัวแปรด้วยเครื่องหมาย , ในส่วนของข้อความและรูปแบบของการแสดงผลจะมีรูปแบบของการแสดงผลแทรกไว้ระหว่างข้อความในตำแหน่งที่เราต้องการให้แสดงผลค่าของนิพจน์นั้น ๆ ถ้ามีนิพจน์หลายนิพจน์ รูปแบบการแสดงผลก็จะมีจำนวนเท่าจำนวนของตัวแปร โดยรูปแบบแรกใช้สำหรับนิพจน์ที่หนึ่ง รูปแบบที่สองสำหรับนิพจน์ตัวที่สอง เป็นต้นจนครบทุกนิพจน์ ถ้าจำนวนรูปแบบน้อยกว่าจำนวนนิพจน์ จะมีนิพจน์จำนวนเท่ากับจำนวนรูปแบบเท่านั้นที่ถูกแสดงผล แต่ถ้าจำนวนของนิพจน์น้อยกว่า รูปแบบที่เกินมาจะถูกแทนที่ด้วย “ค่าขยะ” อันไม่พึงประสงค์

**ตัวอย่าง** การใช้คำสั่ง printf() ที่มีข้อความที่เป็นคำอธิบายอยู่ในรูปคำสั่ง

```
main( )
{
    int sum = 3;
    printf("The sum is %d",sum);
    getch();
}
```

**ผลลัพธ์**

The sum is 3

**ตัวอย่าง** การใช้คำสั่ง printf() รหัสรูปแบบ %d

```
main( )
{
    int a = 3, b = 5;
    printf("a=%d b=%d ",a, b);
    getch();
}
```

**ผลลัพธ์**

a=3 b=5

ตัวอย่าง การใช้คำสั่ง printf()

```
main( )
{
    int a = 3;
    float b = 2.4;
    char c = 'a';
    printf("result : %d %f %c",a,b,c);
    getch();
}
```

ผลลัพธ์

result : 3 2.4 a

ตัวอย่าง การใช้คำสั่ง printf() รหัสรูปแบบ %f และ %e โดยมีทศนิยม 2 ตำแหน่ง

```
main( )
{
    float x = 22.75;
    printf(" %.2f\n%.2e ",x,x);
    getch();
}
```

ผลลัพธ์

22.75

2.28e+01

**ตัวอย่าง** โปรแกรมคำนวณหาพื้นที่สามเหลี่ยม

```
main( )
{
    int base,height;           //ประกาศตัวแปร base และ heightเป็น
                                ประเภท int
    float area;                //ประกาศตัวแปร area เป็นประเภท float
    base = 4;                  // กำหนดให้ base มีค่าเท่ากับ 4
    height = 5;                // กำหนดให้ height มีค่าเท่ากับ 5
    area = 0.5*base*height;    //กำหนดให้ area มีค่าเท่ากับผลการคำนวณหา
                                พื้นที่สามเหลี่ยม

    printf("The area of this triangle is %f",area);
    getch();
}
```

### ผลลัพธ์

The area of this triangle is 10.000000

2. การส่งตัวอักษร หรือค่าในตัวแปรประเภทตัวอักษรให้แสดงผลบนจอภาพ มีคำสั่งพื้นฐานที่ใช้ในโปรแกรมภาษาซี คือ putchar() มีรูปแบบคำสั่ง คือ

```
int putchar (<character>);
```

**ฟังก์ชัน putchar()** จะส่งค่าข้อมูลออกไปยังจอภาพ พร้อมทั้งส่งผลการทำงานกลับมา เนื่องจากฟังก์ชันในภาษาซี ไม่จำเป็นต้องมีตัวแปรมารับค่าผลในการทำงาน โดยทั่วไปจึงเห็นฟังก์ชันนี้อยู่ในรูปที่ไม่ต้องมีตัวแปรมารับค่า ซึ่งอยู่ในรูปของคำสั่ง (Statements) นั่นเอง



ตัวอย่าง การใช้คำสั่ง putchar() ที่มีข้อความที่เป็นคำอธิบายอยู่ในรูปคำสั่ง

```
main( )  
{  
    int a = 65;  
    putchar(a);  
    putchar('a');  
    getch();  
}
```

ผลลัพธ์

Aa

## คำสั่งรับค่าทางแป้นพิมพ์โปรแกรมภาษาซี (Input Function)

1. ฟังก์ชัน scanf() เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์ (keyboard) เข้ามาเก็บไว้ในตัวแปร ซึ่งเป็นฟังก์ชันที่อยู่ในเฮดเดอร์ไฟล์ stdio.h

```
scanf (<control>, <argument list>);
```

argument list	เป็นตัวแปรที่จะรับค่าจากแป้นพิมพ์มาเก็บไว้ถ้ามีมากกว่าหนึ่งค่า จะต้องแยกกันด้วยเครื่องหมายคอมมา ( , ) และตัวแปรทุกตัวที่ใช้จะต้องนำหน้าด้วยเครื่องหมาย & ทุกตัว ยกเว้นตัวแปรชุด
control	เป็นรหัสรูปแบบที่ใช้ในการรับข้อมูลซึ่งจะต้องเขียนอยู่ภายในเครื่องหมาย “ ” และจะมีรหัสรูปแบบเหมือนกับรหัสรูปแบบที่ใช้ในคำสั่ง printf

การใช้ฟังก์ชันนี้มีลักษณะคล้ายกับฟังก์ชัน printf() โดยเราต้องระบุรูปแบบของการรับค่าระหว่างเครื่องหมาย “ และเครื่องหมาย ” โดยรูปแบบนี้มีลักษณะเดียวกับรูปแบบการแสดงผลของฟังก์ชัน printf() และระบุตัวแปรที่ต้องการเก็บค่าไว้หลังเครื่องหมาย , โดยมีเครื่องหมาย & นำหน้าตัวแปรแต่ละตัว โดยตัวแปรที่นำมารับค่านั้นจะต้องถูกประกาศแล้วในตอนต้นของโปรแกรม

ตัวอย่างการใช้งาน	ผลลัพธ์
scanf(“%d”,&a);	รับจำนวนเต็ม เพื่อเก็บไว้ในตัวแปร a
scanf(“%f”,&rate);	รับเลขทศนิยม เพื่อเก็บไว้ในตัวแปร rate
scanf(“%d %d”,&x,&y);	รับจำนวนเต็ม 2 จำนวน โดยจำนวนแรกเก็บไว้ในตัวแปร x และจำนวนที่สองเก็บไว้ในตัวแปร y

เมื่อรันโปรแกรม ฟังก์ชัน scanf() จะทำให้โปรแกรมหยุดเพื่อรอรับค่าจากผู้ใช้ เมื่อผู้ใช้พิมพ์ข้อมูลแล้วตามด้วยการกดแป้น Enter ข้อมูลจะถูกเก็บลงในตัวแปรตามรูปแบบที่กำหนดไว้

ถ้าในคำสั่งรับค่ามากกว่าหนึ่งค่า ผู้ใช้ต้องคั่นค่าที่ป้อนด้วยการกด space bar แล้วจึงกดแป้น enter หลังจากกรอกข้อมูลตัวสุดท้าย

**ตัวอย่าง** การใช้คำสั่ง scanf

```
main( )
{
    int num;
    scanf( “%d” ,&num);
    getch();
}
```

เป็นการรอรับข้อมูลเข้าทางแป้นพิมพ์ เป็นเลข จำนวนเต็ม 1 ค่า โดยเก็บไว้ในตำแหน่งของตัวแปร num

**ตัวอย่าง** การใช้คำสั่ง scanf

```
main( )
{
    int num;
    char ch;
    scanf( “ %c ” ,&ch);
    scanf( “ %d ” ,&num);
    printf(“ %d %c ” ,num, ch);
    getch();
}
```

ตัวอย่าง การใช้คำสั่ง scanf ของโปรแกรมคำนวณหาพื้นที่ของวงกลม

```
void main()
{
    int r;
    float area;
    printf("Enter a value for radius : ");
    scanf("%d",&r);
    area = 3.14*r*r;
    printf("The area of this circle is %f",area);
    getch();
}
```

ผลลัพธ์ : เมื่อผู้ใช้ป้อนค่า 7 ทางแป้นพิมพ์

Enter a value for radius : 7

The area of this circle is 153.8600000

2. ฟังก์ชัน `getchar ()` เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์ทีละ 1 ตัวอักษร โดยต้อง กด enter ทุกครั้งเมื่อสิ้นสุดข้อมูล และข้อมูลที่ป้อนจะปรากฏให้เห็นบนหน้าจอภาพ

รูปแบบ

`getchar ();`

3. ฟังก์ชัน `getch ()` เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์เป็นตัวอักษร 1 ตัวอักษร โดยเมื่อป้อนข้อมูลเสร็จไม่ต้องกดปุ่ม enter และข้อมูลที่ป้อนจะไม่ปรากฏบนหน้าจอภาพ

รูปแบบ

`getch ();`

4. ฟังก์ชัน `getche ()` เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์เป็นตัวอักษร 1 ตัวอักษร โดยเมื่อป้อนข้อมูลเสร็จไม่ต้องกดปุ่ม enter และข้อมูลที่ป้อนจะปรากฏบนหน้าจอภาพ

รูปแบบ

`getche ();`

## ตัวอย่างโปรแกรมการใช้คำสั่งรับ - แสดงผล

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    int x_value = 9;
    printf("x = %d\n",x_value);
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    int x,y,sum;
    x = 7;  y = 2;
    sum = x + y;
    printf("Total = %d\n",sum);
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    Int x, y;
    x = 7;  y = 4;
    --x;
    ++y;
    printf("x = %2d\n",x);
    printf("y = %6d\n",y);
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    Int profits, employees;
    profits = 9;
    employees = 2 ;
    printf("Each employee gets %.2f", (float) profits/employees);
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    char any_char;
    printf("Please type a character: ");
    scanf("c",&any_char);
    printf("Thank you, your character are %c",any_char);
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    printf("\n");
    printf("\xC9\xCD\xBB\n");
    printf("\xC8\xCD\xBC\n");
    getch();
}
```

## ตัวอย่าง

```
#include <stdio.h>
main()
{
    printf("\n****Using .1f****\n");
    printf("%.1f%.1f%.1f\n", 4.0, 16.5, 589.3);
    printf("%.1f%.1f%.1f", 400.0, 1600.5, 58900.3);
    printf("\n\n****Using 8.1f****\n");
    printf("%8.1f%8.1f%8.1f\n", 4.0, 16.5, 589.3);
    printf("%8.1f%8.1f%8.1f", 400.0, 1600.5, 58900.3);
    printf("\n\n**** Using -8.1f****\n");
    printf("%-8.1f%-8.1f%-8.1f\n", 4.0, 16.5, 589.3);
    printf("%-8.1f%-8.1f%-8.1f", 400.0, 1600.5, 58900.3);
    getch();
}
```













ข้อที่ 10	ผลลัพธ์ (OUTPUT)
<pre>#include&lt;stdio.h&gt; main() {     char ch;     printf("Please type a charater:");     scanf("%c",ch);     printf("Thank you,your character is %c", ch);     getch(); }</pre> <p>หมายเหตุ : กำหนดให้ผู้ใช้ป้อนข้อมูล character เป็น A</p>	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>

3. ให้นักเรียนเขียนโปรแกรมภาษาซี เพื่อคำนวณอายุ โดยรับข้อมูลชื่อ (name) และปีที่เกิด (birthyear) พร้อมกับแสดงอายุปัจจุบันออกทางหน้าจอ กำหนดให้

$$\text{อายุปัจจุบัน} = \text{ปี พ.ศ. ปัจจุบัน} - \text{ปีที่เกิด}$$

#### ข้อมูลนำเข้า

กำหนดให้มีข้อความประกอบการรับข้อมูลดังนี้

Enter Name : <ชื่อ>

Enter Birth day : <ปี พ.ศ. ที่เกิด>

Enter Current Year : <ปี พ.ศ. ปัจจุบัน>

#### การแสดงผล

ให้แสดงผลลัพธ์เป็นชื่อและอายุของผู้ใช้ โดยกำหนดให้มีข้อความประกอบการแสดงผลดังนี้

<ชื่อที่ป้อน> is <อายุที่คำนวณได้> year old.

#### ตัวอย่างข้อมูลนำเข้าและการแสดงผลลัพธ์

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างผลลัพธ์
Enter Name : Nina Enter Birth Year : 2527 Enter Current Year : 2554	Nina is 27 year old.

## นิพจน์

นิพจน์ (Expression) คือ การนำตัวแปร ค่าคงที่ มาสัมพันธ์กัน โดยใช้เครื่องหมายอย่างใดอย่างหนึ่งเป็นตัวเชื่อมโดยมีกฎเกณฑ์ทั่ว ๆ ไปในการเขียนนิพจน์ของภาษาซี มีดังนี้

- 1) เขียนตัวอักษรหลายตัวติดกันได้โดยไม่มีเครื่องหมายคั่น เช่น XY ถือเป็น 1 ตัวแปร
- 2) กรณีนิพจน์มีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกันในนิพจน์เดียวกัน กลไกของภาษาซี จะ เปลี่ยนชนิดของข้อมูลที่มีขนาดเล็กให้เป็นชนิดของข้อมูลที่ใหญ่ขึ้น ดังนั้น จึงควรระวังในการตั้งตัวแปรเพื่อ เก็บผลลัพธ์ที่ได้จากการดำเนินการของนิพจน์มีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกัน ซึ่งตัวแปรที่ตั้งขึ้นควรเป็นชนิดของข้อมูลที่ใหญ่ที่สุดในนิพจน์นั้น ดังตัวอย่าง

- หากนิพจน์ มี int กับ long กลไกของภาษาซี จะเปลี่ยนชนิดของข้อมูลเป็น long
- หากนิพจน์มี int กับ double กลไกของภาษาซี จะเปลี่ยนชนิดของข้อมูลเป็น double

นิพจน์คณิตศาสตร์	นิพจน์ภาษาซี
$2X^2$	$2*(X*X)$
$10X + 3XY + 10Y^2$	$10*X+3*X*Y+10*Y*Y$

นิพจน์แบ่งออกเป็นนิพจน์ต่างๆ ดังนี้

### 1) นิพจน์คณิตศาสตร์ (Arithmetic Expression)

หมายถึง การนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายคณิตศาสตร์เป็นตัวเชื่อม ผลที่ได้จากนิพจน์แบบนี้จะเป็นตัวเลข

		นิพจน์คณิตศาสตร์ในภาษา C
$3X + 5Y$	→	$3*X + 5*Y$
$X^2 - Y^2$	→	$X*X - Y*Y$

### 2) นิพจน์ตรรกะ (Logical Expression)

หมายถึง การนำตัวแปร ค่าคงที่ หรือนิพจน์ มาสัมพันธ์กันโดยใช้เครื่องหมายเปรียบเทียบและเครื่องหมายตรรกะเป็นตัวเชื่อม ผลที่ได้จะเป็นจริง หรือเท็จ คือจะให้ค่าเป็น 1 หรือ 0 ออกมาเป็น ผลลัพธ์ซึ่งสามารถนำไปคำนวณต่อได้

ตัวอย่าง ถ้า I มีค่าเป็น 3 J มีค่าเป็น 5 A มีค่าเป็น 3 ถ้าเขียนนิพจน์ดังนี้

I == J	ผลลัพธ์ ไม่จริง (0)
I == A	ผลลัพธ์ จริง (1)
I > J*5	ผลลัพธ์ ไม่จริง (0)
I+3 > J -2 && a*2 > 10	ผลลัพธ์ ไม่จริง (0)

#### ❖ นิพจน์ที่มีระดับการประมวลผลแบบง่าย ได้แก่ ค่าคงตัว ค่าคงที่ และตัวแปร

นิพจน์ที่มีลักษณะเป็น ค่าคงตัว (literal constant) ซึ่งก็คือ ข้อมูลที่ระบุเป็นค่าอย่างใดอย่างหนึ่ง เช่น

100	เป็นค่าคงตัวประเภท	จำนวนเต็ม
'g'	เป็นค่าคงตัวประเภท	ตัวอักษร
23.1	เป็นค่าคงตัวประเภท	จำนวนจริง

นิพจน์ที่เป็น ค่าคงที่ (constant) คือ ชื่อ ที่ใช้แทน ค่าคงตัว ซึ่งโดยปกติแล้ว มักใช้ตัวอักษรพิมพ์ใหญ่ทั้งหมด เช่น

# define VAT 7	กำหนดให้ ค่าคงที่ VAT แทนค่าคงตัว 7
# define PI 3.14	กำหนดให้ ค่าคงที่ แทนค่าคงตัว 3.14
const int MAX = 270;	ประกาศให้ ค่าคงที่ MAX เก็บค่าคงที่ประเภทจำนวนเต็ม และมีค่าเป็น 270
const char CH = 'm';	ประกาศให้ ค่าคงที่ CH เก็บค่าคงที่ประเภทตัวอักษร และมีค่าเป็น 'm'

นิพจน์ที่เป็นตัวแปร เช่น

int count;	ประกาศตัวแปรชื่อ count เก็บค่าจำนวนเต็ม
float amount;	ประกาศตัวแปรชื่อ amount เก็บค่าจำนวนจริง
char vertex;	ประกาศตัวแปรชื่อ vertex เก็บค่าตัวอักษร

❖ **นิพจน์ที่มีระดับการประมวลผลที่ซับซ้อน** เป็นนิพจน์ที่ประกอบด้วย นิพจน์ที่มีระดับการประมวลผลอย่างง่ายหลายนิพจน์เชื่อมต่อกันด้วยตัวดำเนินการ เช่น

$73 + 16$

$2 * \text{PI} * 12.5$

$\text{count} = 100 \% 5 + 12$

$(\text{amount} \geq 10.5) \ \&\& \ (\text{amount} \neq 0)$

## ตัวดำเนินการ

ตัวดำเนินการ (Operator) คือ การใช้ในการกำหนดค่าให้ตัวแปร การกระทำหรือลำดับขั้นตอนในการสร้างค่าใหม่ขึ้นเป็นผลลัพธ์ จากการรับค่าเข้าไปหนึ่งตัวหรือมากกว่า ค่าที่รับเข้าไบนั้น เราเรียกว่า **ตัวถูกดำเนินการ (operand)** ซึ่งตัวถูกดำเนินการนี้คือ นิพจน์ มีตัวดำเนินการ 3 ประเภทใหญ่ ๆ ดังนี้

- 1) ตัวดำเนินการเลขคณิต (arithmetic operators)
- 2) ตัวดำเนินการเปรียบเทียบ (relational operators)
- 3) ตัวดำเนินการตรรกะ (logical operators)

1) **ตัวดำเนินการเลขคณิต (arithmetic operators)** เป็นตัวดำเนินการทางด้านคณิตศาสตร์ ได้แก่ เครื่องหมายที่ใช้ในการบวก ลบ คูณ หาร ตัวเลข และอื่น ๆ ดังในตาราง

ตารางแสดงตัวดำเนินการของ C

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
-	การลบ	$x - y$
+	การบวก	$x + y$
*	การคูณ	$x * y$
/	การหาร	$x / y$
%	การหารจะเอาเฉพาะเศษไว้	$11 \% 3 = 3$ เศษ 2 ดังนั้น 2 เป็นผลลัพธ์
--	การลดค่าลงครั้งละ 1	$x--$ หรือ $--x$ เหมือนกับ $x = x - 1$
++	การเพิ่มค่าครั้งละ 1	$x++$ หรือ $++x$ เหมือนกับ $x = x + 1$

นอกจากตัวดำเนินการบวก ลบ คูณ หาร ธรรมดาแล้ว ยังมีโมดูลัส (modulus) คือ การหารเอาเฉพาะเศษ ดังในตาราง และในภาษาซี มีวิธีการรวบรัดการใช้ตัวดำเนินการเลขคณิต

ดังนี้ เช่น รวมสัญลักษณ์ (+) และ (=) เข้าด้วยกันกลายเป็น (+=) หรือรวมสัญลักษณ์ (-) และ(=) เข้าด้วยกัน กลายเป็น (-=) ดังตาราง

ตารางแสดงตัวอย่างตัวดำเนินการของภาษาซี

นิพจน์	ขั้นตอนการประมวลผล
$y = x + a++;$	1. $y = x + a$ 2. $a = a + 1$
$y = x + ++a;$	1. $a = a + 1$ 2. $y = x + a$
$y += x;$	1. $y = y + x$
$y += -x;$	1. $x = x - 1$ 2. $y = y + x$
$y -= 9;$	1. $y = y - 9$
$y *= 7 * x++;$	1. $y = y * 7 * x$ 2. $x = x + 1$
$y /= x;$	1. $y = y / x$
$y \% = x;$	1. $y = y \% x$

### ขั้นตอนการทำงานของตัวดำเนินการในภาษาซี

ในบางครั้งนิพจน์ประกอบด้วยตัวดำเนินการมากมาย ทำให้เกิดความยุ่งยากในการพิจารณาขั้นตอนการทำงานของตัวดำเนินการ จึงได้ตั้งกฎเกี่ยวกับลำดับการทำงานก่อนหลังของตัวดำเนินการ (Operator) ดังตาราง

ตารางแสดงลำดับการทำงานของตัวดำเนินการ

ตัวดำเนินการ	ลำดับที่
( )	1 (สูงสุด)
++ -- (unary)	2
* / %	3
+ -	4
+= -= *= = /= %=	5 (ต่ำสุด)

ตัวดำเนินการที่มีความสำคัญอยู่ในระดับเดียวกัน ให้ทำงานตามขั้นตอนจากซ้ายไปขวา เป็นหลักต่อไปนี้เป็นตัวอย่างพอสั่งเซป ดังตาราง



ตารางแสดงตัวอย่างการทำงานของตัวดำเนินการ

$5 + 6 * 2$	♣ ตัวดำเนินการ * อยู่ลำดับสูงกว่า + จึงต้องคูณเลขก่อน แล้วบวกเลข 5 ภายหลัง
$2 * 3 - 14 / 7 + 5$	♣ ตัวดำเนินการ * และ / อยู่ลำดับเดียวกัน ให้ทำจากซ้ายไปขวา คือ คูณเลข แล้วหารเลข
	♣ ตัวดำเนินการ - และ + อยู่ในลำดับเดียวกัน ทำจากซ้ายมือก่อน คือ ลบเลข แล้วจึงบวกเลขในภายหลัง

ในกรณีของนิพจน์ที่มีระดับการประมวลผลที่ซับซ้อน จะมีวิธีพิจารณาลำดับการดำเนินการ ดังนี้

- 1) ดำเนินการนิพจน์ที่อยู่ในวงเล็บที่อยู่ในสุดก่อน หากในระดับเดียวกันมีวงเล็บหลายคู่ การประมวลผลจะดำเนินการจากนิพจน์ที่อยู่ในวงเล็บที่อยู่ซ้ายมือสุด ไล่ไปจนถึงนิพจน์ที่อยู่ในวงเล็บขวามือสุดของระดับเดียวกัน
- 2) ดำเนินการ \*, / และ % เป็นตัวดำเนินการมีระดับความสำคัญเท่ากัน โดยจะได้รับการประมวลผลก่อน ตัวดำเนินการ + และ - ซึ่งมีระดับความสำคัญเท่ากัน
- 3) ตัวดำเนินการคำนวณที่มีระดับความสำคัญเท่ากัน การประมวลผลจะดำเนินการจากตัวดำเนินการตัวที่อยู่ซ้ายมือสุด ไล่ไปจนถึงตัวดำเนินการที่อยู่ขวามือสุด

ชนิดของมูลหลังการดำเนินการ ขึ้นอยู่กับชนิดข้อมูลของตัวถูกดำเนินการ โดยหากตัวถูกดำเนินการเป็นจำนวนเต็มทั้งหมด ผลลัพธ์ที่ได้จะมีชนิดข้อมูลเป็นจำนวนเต็ม แต่ถ้าหากมีตัวใดตัวหนึ่งหรือทั้งหมดเป็นจำนวนจริง ผลลัพธ์ที่ได้จะมีชนิดข้อมูลเป็นจำนวนจริง

เช่น	ผลลัพธ์ของ $3 + 5$	คือ	8	ซึ่งเป็นจำนวนเต็ม
	ผลลัพธ์ของ $3.0 + 5$	คือ	8.0	ซึ่งเป็นจำนวนจริง

2) ตัวดำเนินการเปรียบเทียบ (relational operators) หมายถึง เครื่องหมายที่ใช้ในการเปรียบเทียบและตัดสินใจ ผลการเปรียบเทียบจะให้ค่าเป็น 1 และ เท็จจะให้ค่าเป็น 0 เครื่องหมายที่ใช้ มีดังในตาราง

ตารางแสดงตัวดำเนินการเปรียบเทียบ

เครื่องหมาย	ความหมาย	ตัวอย่าง
>	มากกว่า	$A > B$ (A มากกว่า B)
>=	มากกว่าหรือเท่ากับ	$A >= B$ (A มากกว่าหรือเท่ากับ B)
<	น้อยกว่า	$A < B$ (A น้อยกว่า B)
<=	น้อยกว่าหรือเท่ากับ	$A <= B$ (A น้อยกว่าหรือเท่ากับ B)
= =	เท่ากับ	$A = = B$ (A เท่ากับ B)
! =	ไม่เท่ากับ	$A ! = B$ (A ไม่เท่ากับ B)

3) **ตัวดำเนินการตรรกะ (logical operators)** เครื่องหมายตรรกะมีจุดประสงค์ใช้ในการเปรียบเทียบเพื่อตัดสินใจ ซึ่งผลที่ได้จากการเปรียบเทียบจะได้ผลลัพธ์ 2 อย่าง คือ ถ้าได้ผลลัพธ์เป็นจริง จะมีค่าเป็น 1 และถ้าได้ผลลัพธ์เป็นเท็จจะมีค่า 0 เป็นเครื่องหมายตรรกะที่ใช้ในภาษาซี มีดังนี้

คำสั่ง && (AND) เป็นการนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกันซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	Y	X && Y
0	0	0
0	1	0
1	0	0
1	1	1

คำสั่ง || (OR) เป็นการนำเงื่อนไข 2 เงื่อนไขมาเปรียบเทียบกันซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	Y	X    Y
0	0	0
0	1	1
1	0	1
1	1	1

คำสั่ง ! (NOT) เป็นการทำนิเสธซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้

X	! X
0	1
1	0

ตัวอย่าง จงเขียนโปรแกรมเพื่อรับค่ารัศมีวงกลมจากผู้ใช้ แล้วคำนวณและแสดงผลค่าของเส้นรอบวง และค่าของพื้นที่ของวงกลมนั้น

#### วิเคราะห์ปัญหา

ข้อมูลเข้า	รัศมี (r)
ข้อมูลออก	เส้นรอบวง (c) และ พื้นที่ (a)
ขั้นตอนวิธี	ขั้นที่ 1 เริ่มต้น
	ขั้นที่ 2 รับค่า r
	ขั้นที่ 3 คำนวณค่าเส้นรอบวง $c = 2 * PI * r$
	ขั้นที่ 4 คำนวณค่าพื้นที่ $a = PI * r * r$
	ขั้นที่ 5 แสดงผลค่า c และ ค่า a
	ขั้นที่ 6 จบ

```
#include<stdio.h>
#include<conio.h>
void main()
{
    const float PI = 3.14;           // ประกาศให้ตัวแปร PI เป็นค่าคงที่
    float r,c,a;                     // ประกาศตัวแปร r,c และ a เป็นประเภท
                                    // จำนวนจริง
    printf("Enter a value for radius : "); // แสดงข้อความ
    scanf("%f",&r);                 // รับข้อมูลประเภทจำนวนจริงจากผู้ใช้เพื่อ
                                    // เก็บในตัวแปร r
    c = 2*PI*r;                      // คำนวณค่าความยาวเส้นรอบวงแล้วเก็บใน
                                    // ตัวแปร c
    a = PI*r*r;                      // คำนวณค่าพื้นที่แล้วเก็บในตัวแปร a
    printf("The circumference of this circle is %.2f\n",c);
                                    // แสดงค่าความยาวเส้นรอบวงในรูปแบบ
                                    // ทศนิยม 2 ตำแหน่ง
    printf("The area of this circle is %.2f",a);
                                    // แสดงค่าพื้นที่ในรูปแบบทศนิยม 2 ตำแหน่ง
    getch();
}
```



4. จงระบุว่า นิพจน์ที่กำหนดให้ต่อไปนี้เป็นการกำหนดนิพจน์แบบใด และมีค่าเท่าใด

ข้อที่	นิพจน์	ค่าคงตัว	ค่าคงที่	ตัวแปร	ซับซ้อน	ค่า
0.	75	✓				75
1.	'9'					
2.	int mark = 50+4;					
3.	const float MIN = 0.0;					
4.	1+2-3*4/5					
5.	#define BASE 100					

## แบบทดสอบก่อนเรียน

## การเขียนโปรแกรมเบื้องต้นด้วยภาษาซี เรื่อง โครงสร้างของโปรแกรมภาษา

คำชี้แจง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว

- ข้อใดกล่าว ถูกต้อง เกี่ยวกับภาษาซี
  - ใช้ตัวแปลภาษาที่แปลทีเดียวกึ่งโปรแกรม เรียกว่า คอมไพเลอร์ (Compiler)
  - มีโครงสร้างคล้ายภาษาระดับสูง แต่ทำงานได้รวดเร็วเหมือนภาษาระดับต่ำ
  - เป็นภาษาระดับกลาง (Medium Level Language )
  - ถูกทุกข้อ
- ข้อใดคือฟังก์ชันหลักของโปรแกรมภาษาซี
  - `#include <stdio.h>`
  - `main()`
  - `int num1;`
  - `getch();`
- Compiler จะแจ้งข้อความ `syntax error` เมื่อ Run โปรแกรมแล้วพบว่ามีผิดพลาด ด้านใด
  - ประกาศตัวแปรไม่ถูกต้อง
  - กำหนดค่าไม่ถูกต้อง
  - ผลลัพธ์ของโปรแกรมไม่ถูกต้อง
  - ไวยากรณ์คำสั่งไม่ถูกต้อง
- รหัสควบคุมในข้อใด ใช้เพื่อขึ้นบรรทัดใหม่ (Newline)
  - `\a`
  - `\b`
  - `\n`
  - `\v`
- ข้อใดกล่าว ไม่ถูกต้อง
  - ค่าคงที่บูลีนมี 2 กรณี คือ ถ้าเป็นเท็จ มีค่าเป็น 0 ถ้าเป็นจริง มีค่าเป็น 1
  - ตัวดำเนินการทางคณิตศาสตร์ เช่น `+`, `-`, `*`, `/`, `%`
  - ตัวดำเนินการเปรียบเทียบ เช่น `>`, `<`, `==`, `!=`
  - ตัวดำเนินการกำหนดค่า เช่น `|=`, `^=`, `&&`, `!`

6. หลักในการกำหนดชื่อตัวแปรข้อใด ไม่ถูกต้อง

- ก. ต้องขึ้นต้นด้วยตัวเลข
- ข. ต้องขึ้นต้นด้วยตัวอักษร
- ค. สามารถใช้เครื่องหมาย underline ‘ \_ ’ ได้
- ง. ห้ามใช้เครื่องหมายทางคณิตศาสตร์ในชื่อตัวแปร

7. ข้อใดแสดงผลลัพธ์ของคำสั่งต่อไปนี้ได้ถูกต้อง

```
printf("Float number: %3.2f\n", 3.14159);
```

- ก. Float number: 3.2
- ข. Float number: %3.2f\n
- ค. Float number: 3.14
- ง. Float number: 3.14159

8. ข้อใดเป็นคำสั่งที่ใช้ในการรับข้อมูลเป็นตัวอักขระ 1 ตัวอักขระเข้ามาทางแป้นพิมพ์ โดยเมื่อป้อนข้อมูลเสร็จไม่ต้องกดปุ่ม enter และข้อมูลที่ป้อนจะไม่ปรากฏบนจอภาพ

- ก. getch()
- ข. getcher()
- ค. getch()
- ง. getchar()

9. ข้อใดใช้ฟังก์ชัน printf ( ) ในการแสดงผลบวกของตัวแปร x กับตัวแปร y ได้ถูกต้อง

- ก. printf(“%d + %d = %d”,x , y , x+y);
- ข. printf (“x + y = %d”, x + y );
- ค. printf (“x + y = x+ y \n”, x ,y , x+y);
- ง. printf ( “x + y = ”,x+y);

10. ตัวแปร num เป็นตัวแปรชนิด int และ temp เป็นตัวแปรชนิด float ถ้าจะให้คำสั่งต่อไปนี้ทำงานได้อย่างถูกต้อง ต้องเติมข้อความใดลงในช่องว่างของคำสั่ง

```
scanf(“%f %d”, _____);
```

- ก. num , temp
- ข. &num , &temp
- ค. temp ,num
- ง. &temp , &num

### บรรณานุกรม

กระทรวงศึกษาธิการ. หลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551. กรุงเทพมหานคร :

โรงพิมพ์ชุมนุมสหกรณ์การเกษตรแห่งประเทศไทย, 2551.

ธนาวุฒิ ประกอบผล. การเขียนโปรแกรมเบื้องต้นด้วยภาษา C . กรุงเทพมหานคร : ซีเอสเอส มีเดีย,  
2554.

ธีรวัฒน์ ประกอบผล. คู่มือการเขียนโปรแกรมภาษา C . กรุงเทพมหานคร : ซิมพลิฟาย, 2554.

ประภาพร ช่างไม้. ฝึกเขียนโปรแกรมจาก รวมโจทย์แบบทดสอบภาษา C .นนทบุรี : ไอดีซี ฯ, 2549.

ยีน ภู่วรรณ และคณะ.การเขียนโปรแกรมคอมพิวเตอร์ภาษาซี.กรุงเทพมหานคร : มูลนิธิ สอวน,  
2547.

Behrouz A. Forouzan , Richard F. Gillberg, “Computer Science : A structured  
Programming Approach Using C”,Third Edition, Brooks/Cole , 2007.



A person's hands are shown typing on a white laptop keyboard. The image is framed by large, colorful, overlapping geometric shapes in shades of teal, orange, and purple. A code snippet is overlaid on the keyboard area. The background features a faint, light-colored circuit board pattern.

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

```
int main(void)  
{  
    printf ("hello, world!\n");  
    return 0;  
}
```