

วิชา การพัฒนาโปรแกรม (ภาษาซี)

รหัสวิชา ว31293

หน่วยการเรียนรู้ที่ 6 อาร์เรย์และฟังก์ชันจัดการสตริง

สาระการเรียนรู้

1. ชนิดของตัวแปรอาร์เรย์
2. ฟังก์ชันจัดการสตริง
3. การอ่านและแสดงค่าของสตริง
4. การแปลงข้อความที่เป็นตัวเลขเป็นค่าตัวเลขที่นำไปคำนวณได้

ผลการเรียนรู้

สามารถเลือกใช้ตัวแปรอาร์เรย์ในการเขียนโปรแกรมและประยุกต์ใช้งานฟังก์ชันเพื่อจัดการข้อความหรือสตริงได้อย่างถูกต้องและเหมาะสม

จุดประสงค์การเรียนรู้

1. เข้าใจหลักการอาร์เรย์ และสามารถนำไปประยุกต์ใช้งานได้อย่างถูกต้อง
2. เปรียบเทียบความแตกต่างระหว่างอาร์เรย์ 1 มิติ และอาร์เรย์ 2 มิติได้
3. สามารถเขียนโปรแกรมเพื่อจัดเก็บข้อมูลลงในตัวแปรอาร์เรย์ได้
4. สามารถเขียนโปรแกรมเพื่อจัดการกับข้อความหรือสตริงได้
5. สามารถแปลงข้อความที่เป็นตัวเลข นำไปใช้เพื่อการคำนวณได้



ใบความรู้ที่ 6 เรื่อง อาร์เรย์และฟังก์ชันจัดการสตริง

จัดทำโดย นางพรพนารัตน์ ชมภูษ

เมื่อเครื่องคอมพิวเตอร์มีการประกาศตัวแปรเกิดขึ้น ชื่อตัวแปรจะเป็นตัวแทนตำแหน่งหน่วยความจำของคอมพิวเตอร์ ตัวแปรหนึ่งตัวจะแทนหน่วยความจำหนึ่งตำแหน่ง สามารถเก็บข้อมูลได้หนึ่งค่า ซึ่งเดิมถ้าหากต้องการเก็บข้อมูลหลายค่าจะต้องประกาศตัวแปรขึ้นมาหลายตัว ในระบบการประมวลผลข้อมูลหน่วยความจำนั้น เราสามารถดำเนินการสร้างตัวแปรที่สามารถเก็บข้อมูลชนิดเดียวกันแต่มีหลายค่า โดยสร้างตัวแปรขึ้นมาเพียงตัวเดียว แต่สามารถเก็บค่าได้หลายค่า เราเรียกตัวแปรแบบนี้ว่า **ตัวแปรแถวลำดับ** หรือ **ตัวแปรอาร์เรย์ (Array)** และการเก็บข้อมูลแบบอาร์เรย์นั้น อาจมองอีกอย่างหนึ่งว่าเป็นการเก็บข้อมูลแบบตาราง นั่นเอง



วิธีการกำหนดตัวแปรชุด และการใช้งาน

อาร์เรย์หรือตัวแปรชุด เป็นตัวแปรประเภทหนึ่งที่เหมาะสมกับการนำไปใช้เพื่อประมวลผลกลุ่มชุดข้อมูลเดียวกัน ถ้าต้องการประกาศตัวแปรที่มีชนิดข้อมูลเดียวกันจำนวน 10 ตัว ที่มีชนิดข้อมูลเป็นชนิดตัวเลขจำนวนเต็ม (Integer) หากใช้วิธีประกาศตัวแปรแบบเดิม ๆ ก็จะต้องประกาศดังนี้

```
int a1 , a2 , a3 , a4 , a5 , a6 , a7 , a8 , a9 , a10;
```

จากตัวอย่างข้างต้น เห็นได้ว่า การประกาศตัวแปรแบบนี้ทำให้เกิดความยุ่งยากในการอ้างอิงตัวแปรเพื่อใช้งานค่าของตัวแปรดังกล่าวเมื่อมีข้อมูลมากขึ้น หรือเพื่อการเรียกใช้งานตัวแปรภายใต้เงื่อนไขใดเงื่อนไขหนึ่ง

ดังนั้น จึงมีแนวทางแก้ไขเพื่อให้เราสามารถอ้างอิงตัวแปร เพื่อนำมาใช้ได้ง่ายขึ้น ที่เรียกว่า **ตัวแปรอาร์เรย์** โดยอาร์เรย์จะดำเนินการเสมือนกับแบ่งหน่วยความจำออกตาราง และภายในตารางก็แบ่งออกเป็นช่องเซลล์ (**Elements**) และสามารถอ้างอิงถึงค่าในแต่ละช่องด้วยเลข **เลขดัชนี (index number)** หรือที่เรียกว่า **ซับสคริปต์ (subscript)** เป็นตัวชี้ตำแหน่งของช่องอิลิเมนต์นั้น ๆ ซึ่งภาษาซีจะใช้สัญลักษณ์ [n] เป็นตัวชี้ระบุตำแหน่ง

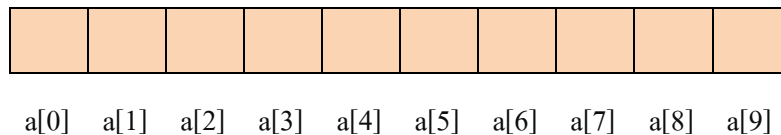
ตัวอย่างเช่น

a[1] หมายถึง การอ้างอิงถึงอิลิเมนต์ของอาร์เรย์ a ช่องหน่วยความจำที่ 1

ดังนั้น เมื่อต้องการประกาศตัวแปร a เป็นตัวแปรอาร์เรย์ที่มีชนิดข้อมูลเป็น integer ทำได้โดย

```
int a[10];
```

อย่างไรก็ตาม ตำแหน่งแรกของอาร์เรย์ในภาษาซีจะเริ่มตั้งแต่ตำแหน่งช่องที่ 0 เป็นต้นไป ดังนั้น อาร์เรย์จากตัวอย่างข้างต้นจะมีสมาชิกจำนวน 10 ตัว และอ้างอิงเลขดัชนีตั้งแต่ 0 – 9 ดังนี้



รูปที่ 7.1 อาร์เรย์ a ที่ประกาศไว้เพื่อจัดเก็บสมาชิก 10 ชุด

สรุปได้ว่า การกำหนดตัวแปรชุดมาใช้งาน สามารถทำได้เช่นเดียวกับการกำหนดตัวแปรธรรมดา เพียงแต่ต้องมีสัญลักษณ์ [] มาต่อท้าย (ซึ่งสามารถมีเครื่องหมายดังกล่าวต่อท้ายได้มากกว่าหนึ่งครั้ง) และภายในสัญลักษณ์ดังกล่าวต้องมีการระบุขนาดของชุดตัวแปรด้วย

ตัวอย่าง

```
Int    i;
int    j[5];
int    m[2][3];
float  L[3][4][20];
```

คำอธิบาย จากตัวอย่างจะเห็นได้ว่า

- i เป็นตัวแปรประเภท Integer
- j เป็นตัวแปรชุดประเภท integer มีขนาดความจุได้ 5 จำนวน
- m เป็นตัวแปรชุด(สองมิติ)ประเภท integer มีขนาดความจุได้ 2×3 จำนวน
- L เป็นตัวแปรชุด(สามมิติ) ประเภท float มีขนาดความจุได้ $3 \times 4 \times 20$ จำนวน



ชนิดของตัวแปรอาร์เรย์

อาร์เรย์สามารถประกาศใช้กับตัวแปรได้หลายลักษณะ แต่โดยส่วนใหญ่ อาร์เรย์ 1 มิติ และ อาร์เรย์ 2 มิติ จะถูกนำมาใช้งานมากที่สุด ดังนั้นจึงขอกล่าวรายละเอียดถึงอาร์เรย์ 1 มิติ และ อาร์เรย์ 2 มิติ เท่านั้น

1. อาร์เรย์ 1 มิติ (One Demension Array)

เป็นตัวแปรอาร์เรย์แบบมิติเดียว มีลักษณะเหมือนกับหน่วยความจำที่เรียงกันเป็นลำดับ โดยมีรูปแบบคำสั่งดังนี้

รูปแบบ การประกาศตัวแปรชุด 1 มิติ

```
Type Arrayname [size];
```

Type คือ ชนิดของตัวแปร เช่น int , char , float

Arrayname คือ ชื่อของตัวแปร array

size คือ ขนาดของตัวแปร

ตัวอย่าง

กรณีที่ 1 การประกาศตัวแปรชนิดอาร์เรย์ 1 มิติ ที่มีชนิดข้อมูลเป็น ตัวขระ หรือตัวอักษร

```
char name[20];
```

หมายถึง กำหนดให้ตัวแปร name จัดเก็บอักขระได้จำนวน 25 ตัวอักษร

การกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์ชนิดตัวอักขระ สามารถกำหนดในรูปแบบต่าง ๆ ดังนี้

แบบที่ 1 char name[15] = "ponpanarat";

แบบที่ 2 char name[15] = {'p', 'o', 'n', 'p', 'a', 'n', 'a', 'r', 'a', 't'};

name	p	o	n	p	a	n	a	r	a	t	\0	\0	\0	\0	\0
	name[0]	name[1]	name[2]	name[3]	name[4]	name[5]	name[6]	name[7]	name[8]	name[9]	name[10]	name[11]	name[12]	name[13]	name[14]

รูปที่ 7.2 ค่าที่บรรจุอยู่ในแต่ละอิลิเมนต์ของอาร์เรย์ name ที่ประกาศไว้ เพื่อจัดเก็บสมาชิก 15 ชุด

ศึกษาสังเกต

สำหรับการประกาศค่าตัวแปรให้กับตัวแปรอาร์เรย์ชนิดข้อความ (Character) นั้น อักขระแต่ละตัวจะถูกบรรจุในแต่ละช่องหน่วยความจำ โดยในแต่ละช่องหน่วยความจำ จะจัดเก็บได้เพียงหนึ่งอักขระเท่านั้น โดยจะมีรหัสพิเศษ คือ \0 (Null) ซึ่งเป็นตัวแปรที่บ่งบอกถึงจุดสิ้นสุดของข้อความ

กรณีที่ 2 การประกาศตัวแปรชนิดอาร์เรย์ 1 มิติ ที่มีข้อมูลชนิด ตัวเลข ได้แก่ integer , float , double เป็นต้น

```
int age[10];
```

หมายถึง กำหนดให้ตัวแปรชื่อ age จัดเก็บอักขระได้จำนวน 10 จำนวน

การกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์ชนิดตัวเลข สามารถกำหนดในรูปแบบต่าง ๆ ดังนี้

```
int age[10] = { 25 , 23 , 24 , 27 , 26 , 29 , 30 , 34 , 27 , 23 };
```

age	25	23	24	27	26	29	30	34	27	23
	age[0]	age[1]	age[2]	age[3]	age[4]	age[5]	age[6]	age[7]	age[8]	age[9]

รูปที่ 7.3 ค่าที่บรรจุอยู่ในแต่ละช่องอิลิเมนต์ของอาร์เรย์ age ที่ประกาศไว้ เพื่อจัดเก็บสมาชิก 10 ชุด

เราสามารถอ้างอิงถึงข้อมูลภายในตัวแปรข้อมูลชุดแบบ 1 มิติ ได้โดยใช้เลขดัชนี (Index) เป็นตัวระบุตำแหน่งของข้อมูล ดังตัวอย่างดังนี้

ตัวอย่าง

```
printf(“ %d ”,age[2]);
```

หมายถึง แสดงค่าภายในตัวแปรอาร์เรย์ชื่อ age ในตำแหน่งช่องอิลิเมนต์ที่ 2 นั่นคือ ค่าที่ได้เท่ากับ 24 นั่นเอง



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมเพื่อหาคะแนนเฉลี่ยของคะแนนเก็บในที่นี่จะเป็นการป้อนคะแนนเก็บ 3 ครั้ง ของนักเรียนคนหนึ่งโดยบันทึกโปรแกรม ชื่อ ex07_1.cpp ซึ่งประกอบด้วยตัวแปรดังต่อไปนี้

ข้อมูลนำเข้า ได้แก่ ชื่อ (name) , คะแนนเก็บ (score)

ผลลัพธ์ที่ต้องการ ได้แก่ คะแนนรวม (total) , คะแนนเฉลี่ย (average)

```

1 #include <stdio.h>
2 #include <conio.h>
3 char name[20];
4 int i , score[3] , total;
5 float average;
6 main()
7 {
8     printf("Input name : ");
9     scanf("%s",name);
10
11     for (i=0 ; i<=2 ; i++)
12     {
13         printf("score [%d] is :", i+1 );
14         scanf("%d", &score[i] );
15
16         total = total + score[i];
17         average = total / i+1;
18     }
19     printf("Total is : %d\n", total );
20     printf("Average is : %.2f", average );
21     getch( );
22 }
```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_1.exe

2. อาร์เรย์ 2 มิติ (Two Dimension Array)

ลักษณะของตัวแปรชุด 2 มิติจะเป็นในลักษณะของตาราง พิจารณาจากตัวอย่างการเก็บข้อมูลคะแนนสอบของนักเรียนในวิชาใดวิชาหนึ่ง เช่น ถ้าแบ่งการเก็บคะแนนออกเป็น 5 ครั้ง จะพบว่าในการเก็บข้อมูลคะแนนสอบของนักเรียนแต่ละคนสามารถใช้ตัวแปรชุดมิติเดียว ดังรูป

25	23	24	27	26
score[0]	score[1]	score[2]	score[3]	score[4]

รูปที่ 7.4 แสดงการเก็บคะแนนสอบของนักศึกษาคนหนึ่ง

แต่หากเก็บคะแนนข้อมูลของนักศึกษาทุกคนในชั้นนั้น จะต้องใช้ตัวแปรชุดหลายมิติเข้ามาเกี่ยวข้องตัวอย่างของข้อมูลที่เก็บดังรูปที่ 7.5

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5
นาย ก	23	24	22	21	30
นาย ข	22	33	27	19	25
...

รูปที่ 7.5 แสดงตัวอย่างการเก็บข้อมูลคะแนนของนักศึกษา

ข้อมูลที่จะจัดเก็บในตัวแปรชุดจะต้องเป็นข้อมูลชนิดเดียวกันเสมอ ในที่นี้จะจัดเก็บเฉพาะส่วนของคะแนนสอบของนักเรียนเท่านั้น จากลักษณะความต้องการเก็บข้อมูลดังกล่าวจะต้องเตรียมตัวแปรชุดเพื่อเก็บข้อมูลในลักษณะ 2 มิติ โดยมีรูปแบบคำสั่งดังนี้

รูปแบบ การประกาศตัวแปรชุด/2 มิติ

```
Type Arrayname [Row][Column];
```

Type คือ ชนิดของตัวแปร เช่น int , char , float

Arrayname คือ ชื่อของตัวแปร array

Row คือ ขนาดของแถว

Column คือ ขนาดของสดมภ์ หรือคอลัมน์

ตัวอย่าง

การประกาศตัวแปรอาร์เรย์เพื่อเก็บข้อมูลคะแนนของนักเรียน จำนวน 4 คน และมีคะแนนสอบ 3 ครั้ง ในรูปแบบตัวแปรอาร์เรย์ 2 มิติ ดังนี้

```
int score [4] [3];
```

หมายถึง กำหนดตัวแปรชื่อ score ที่สามารถจัดเก็บข้อมูลชนิด integer โดยมีลักษณะของช่องอิลิเมนต์เท่ากับ 4 แถว 3 คอลัมน์

การกำหนดค่าเริ่มต้นให้กับตัวแปร หรือการรับค่าให้กับตัวแปร score ในรูปแบบ ตัวแปรอาร์เรย์ 2 มิติ สามารถทำได้โดยการระบุตำแหน่งของช่องอิลิเมนต์ที่ต้องการเก็บ โดยมีรูปแบบดังนี้

ตัวอย่าง การกำหนดค่าเริ่มต้นให้กับตัวแปร score[4][3]

```
int score[4][3] = { { 10, 12, 14},
                   { 15, 13, 16},
                   { 12, 15, 19},
                   { 11, 15, 18} };
```

หรือ กำหนดในลักษณะเรียงเป็นแถวเดียวก็ได้

```
int score[4][3] = { 10, 12, 14, 15, 13, 16, 12, 15, 19, 11, 15, 18};
```

กรณีการกำหนดค่าเริ่มต้นให้กับตัวแปรอาร์เรย์ 2 มิติ การใช้เครื่องหมาย { } ซ้อนย่อยเพื่อกำหนดข้อมูลจัดเก็บลงในแต่ละแถว จะทำให้อ่านได้ง่ายกว่า และจากการประกาศข้างต้น ในแต่ละอิลิเมนต์ จะถูกจัดเก็บข้อมูลไว้ในหน่วยความจำในตำแหน่งของอิลิเมนต์ต่างๆ ดังรูป

score	column 0	column 1	column 2
row 0	10	12	14
row 1	15	13	16
row 2	12	15	19
row 3	11	15	18

รูปที่ 7.6 แสดงตัวอย่างการเก็บข้อมูลคะแนนของนักเรียน

ตัวแปรอาร์เรย์ 2 มิติจะมองข้อมูลในลักษณะแถวและคอลัมน์ แถวของข้อมูลในที่นี้จะเป็นคะแนนที่นักเรียนแต่ละคนได้รับ ส่วนคอลัมน์จะเป็นคะแนนสอบแต่ละครั้งของนักเรียนเพราะฉะนั้นเมื่อเราอ้างอิงจุดใดจุดหนึ่งใน ตัวแปรชุด 2 มิติ ก็จะเป็นคะแนนที่นักเรียนแต่ละคน ดังนั้นในการอ้างอิงข้อมูลภายในตารางข้อมูล 2 มิติของตัวแปรอาร์เรย์ ดังตารางที่กล่าวมาข้างต้นนั้น สามารถทำได้โดย ระบุตำแหน่งของแถว และ คอลัมน์ ของตำแหน่งนักเรียนที่ต้องการทราบข้อมูล ดังตัวอย่างดังนี้

ตัวอย่าง

การรับค่าทางเป็นพิมพ์เพื่อบันทึกข้อมูลลงในตัวแปรอาร์เรย์ 2 มิติ ในตำแหน่งอิเลเมนต์ที่ต้องการ สามารถทำได้ดังตัวอย่างต่อไปนี้

```
scanf("%d", &score[0][0]);
```

หมายถึง การรับข้อมูลทางคีย์บอร์ดมาจัดเก็บไว้ในตัวแปรที่ชื่อ score ในตำแหน่งอิเลเมนต์แถวที่ 0 คอลัมน์ที่ 0 คู่อันดับดังรูป

score	column 0	column 1	column 2
row 0	10	12	14
row 1	15	13	16
row 2	12	15	19
row 3	11	15	18

รูปที่ 7.7 แสดงตัวอย่างการเก็บข้อมูลคะแนนของนักเรียนในตำแหน่งอิเลเมนต์ แถว 0 คอลัมน์ 0

และถ้าต้องการแสดงข้อมูลที่อยู่ภายในอิเลเมนต์ที่ต้องการ ก็สามารถทำได้โดยอ้างอิงถึงตำแหน่งของข้อมูลในอิเลเมนต์นั้น ดังตัวอย่างต่อไปนี้

ตัวอย่าง

```
printf("%d", &score[0][0]);
```

หมายถึง การแสดงข้อมูลที่จัดเก็บไว้ในตัวแปรที่ชื่อ score ในตำแหน่งอิเลเมนต์แถวที่ 0 คอลัมน์ที่ 0 นั่นก็คือ 10 (ดูรูปที่ 7.7)

ในกรณีที่ต้องการจัดเก็บข้อมูลชนิดอักขระ ในรูปแบบอาร์เรย์ 2 มิติ สามารถประกาศได้ เช่นเดียวกับ การประกาศตัวแปรอาร์เรย์ 2 มิติ ชนิดตัวเลข ซึ่งในแต่ละอิลิเมนต์ในหน่วยความจำจะสามารถจัดเก็บข้อมูลได้ 1 ตัวอักขระเท่านั้น ดังนั้น การประกาศตัวแปรดังกล่าวจึงไม่ใช่จัดเก็บชื่อของแต่ละคน แต่เป็นการจัดเก็บกลุ่มอักขระนั่นเอง ดังนั้นหากต้องการประกาศตัวแปรชนิดอักขระในรูปแบบ อาร์เรย์ 2 มิติ สามารถทำได้ ดังตัวอย่างต่อไปนี้

ตัวอย่าง

```
char name [3][10];
```

หมายถึง กำหนดตัวแปรชื่อ name ที่สามารถจัดเก็บข้อมูลชนิด อักขระ โดยมีลักษณะของช่องอิลิเมนต์ เท่ากับ 3 แถว 10 คอลัมน์

การกำหนดค่าเริ่มต้นให้กับตัวแปร หรือการรับค่าให้กับตัวแปร name ในรูปแบบ ตัวแปร อาร์เรย์ 2 มิติ สามารถทำได้โดย ไม่ต้องระบุตำแหน่งของช่องอิลิเมนต์ที่ต้องการเก็บ ซึ่งมีรูปแบบดังนี้

ตัวอย่าง การกำหนดค่าเริ่มต้นให้กับตัวแปร name[3][10]

```
char name[3][10] = {"MALI", "DOW", "SOMSAK"};
```

ซึ่งผลจากการประกาศพร้อมการกำหนดค่าดังกล่าว การจัดเก็บข้อมูลในแต่ละช่องอิลิเมนต์ ในแถวและคอลัมน์จะได้ผลลัพธ์ดังรูป

name	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
Row 0	M	A	L	I	\0	\0	\0	\0	\0	\0
Row 1	D	O	W	\0	\0	\0	\0	\0	\0	\0
Row 2	S	O	M	S	A	K	\0	\0	\0	\0

รูปที่ 7.8 แสดงตัวอย่างการเก็บข้อมูลชื่อของนักเรียนที่ประกาศไว้จำนวน 3 คน

สำหรับการอ้างถึงตัวแปรอาร์เรย์ 2 มิติ ชนิดอักขระนั้น สามารถทำได้โดยระบุเฉพาะ ตำแหน่งของแถวของตัวแปรเท่านั้น แต่การแสดงผลลัพธ์จะแสดงค่าในอิลิเมนต์ในแถว ๆ นั้นทุกอักขระ โดยอัตโนมัติ ดังตัวอย่างต่อไปนี้

ตัวอย่าง การอ้างถึงเพื่อแสดงข้อมูลในตัวแปร name[3][10] ทำได้ดังนี้

กรณีที่ 1 แสดงข้อมูลชื่อเป็นรายคน

```
printf( "%s", name[1] );
```

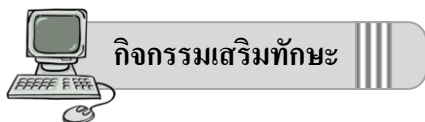
หมายถึง การแสดงข้อมูลที่จัดเก็บไว้ในตัวแปรที่ชื่อ score ในตำแหน่ง
 อิลิเมนต์แถวที่ 1 ผลลัพธ์ที่ได้ คือ DOW (ดูรูปที่ 7.9)

name	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10
Row 0	M	A	L	I	\0	\0	\0	\0	\0	\0	\0
Row 1	D	O	W	\0	\0	\0	\0	\0	\0	\0	\0
Row 2	S	O	M	S	A	K	\0	\0	\0	\0	\0

รูปที่ 7.9 แสดงตัวอย่างการเก็บข้อมูลชื่อของนักเรียนที่ตำแหน่งแถวที่ 1

กรณีที่ 2 แสดงข้อมูลชื่อทั้งหมดทุกคนที่จัดเก็บไว้

```
for (i=0 ; i<=2 ; i++)
    printf( "%s", name[i] );
```



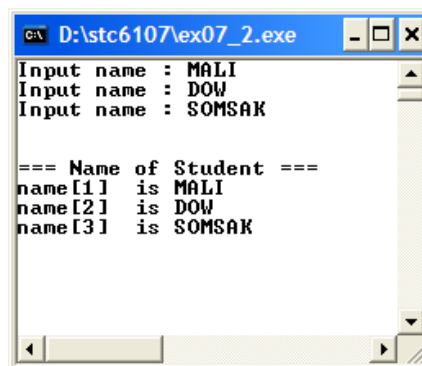
กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมเพื่อจัดเก็บชื่อนักเรียน โดยจัดเก็บข้อมูลของนักเรียนจำนวน 3 คน และนำข้อมูลที่จัดเก็บไว้นำมาแสดงดังรูป

```

1  #include <stdio.h>
2  #include <conio.h>
3  char name[3][15];
4  int i ,j ;
5  main()
6  {
7      for (i=0 ; i<=2 ; i++)
8      {
9          printf("Input name : ");
10         scanf("%s", name[i] );
11     }
12     printf("\n\n=== Name of Student ===\n");
13
14     for (j=0 ; j<=2 ; j++)
15     {
16         printf("name[%d] is %s \n", j+1 , name[j]);
17     }
18     getch();
19 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงาน
ของโปรแกรม ex07_2.exe



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมเพื่อจัดเก็บข้อมูลนักเรียน โดยจัดเก็บข้อมูลของนักเรียนจำนวน 3 คน และแต่ละคนมีคะแนนสอบ 2 ครั้ง และหาผลรวมของคะแนนเก็บของนักเรียนทั้ง 3 คน

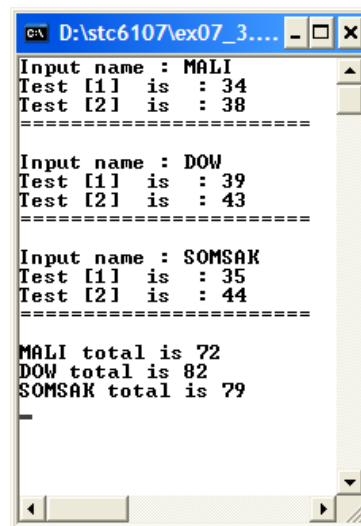
ข้อมูลนำเข้า ได้แก่ ชื่อ (name) , คะแนนเก็บ (score)

ผลลัพธ์ที่ต้องการ ได้แก่ คะแนนรวม (total)

```

1 #include <stdio.h>
2 #include <conio.h>
3 char name[3][15];
4 int i , j , a , b , c , score[3][2],total[3][1] ;
5 main()
6 {
7     for (i=0 ; i<=2 ; i++)
8     {
9         printf("Input name : ");
10        scanf("%s",name[i] );
11        for (j=0 ; j<=1 ; j++)
12        {
13            printf("Test [%d] is : ", j+1 );
14            scanf("%d", &score[i][j] );
15            total[i][0]=total[i][0]+score[i][j];
16        }
17        printf("=====\n\n");
18    }
19    for(a=0;a<=2;a++);
20    {
21        for (b=0;b<=2;b++)
22        {
23            printf("%s total is ",name[b]);
24            for (c=0;c<=0;c++)
25            {
26                printf("%d \n", total[b][0]);
27            }
28        }
29    }
30    getch();
31 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_2.exe



ฟังก์ชันจัดการสตริง

จากเนื้อหาที่ผ่านมาทำให้เข้าใจว่า สตริงหรือข้อความ จะถูกจัดเก็บในลักษณะของอาร์เรย์ของอักขระนั่นเอง อีกทั้งยังมีการใช้รหัส \0 ไว้ท้ายข้อมูลเพื่อให้ทราบว่าเป็นจุดสิ้นสุดของข้อความนั้น ๆ

สำหรับการประกาศชนิดข้อมูลพร้อมกำหนดค่าเริ่มต้นให้กับตัวแปรสตริงนั้น สามารถกระทำได้หลายรูปแบบดังนี้

```
Char school[10]= {'c', 'h', 'a', 'i', 'b', 'a', 'd', 'a', 'n', '\0'};
```

```
Char school[10]= "chaibadan",
```

```
Char school[] = {'c', 'h', 'a', 'i', 'b', 'a', 'd', 'a', 'n', '\0'};
```

```
Char school[] = "chaibadan",
```

ในกรณีที่ต้องการกำหนดค่าเริ่มต้นให้กับตัวแปรชนิดอักขระ โดยปกติจะสามารถเก็บได้เพียง 1 อักขระเท่านั้น หากต้องการเก็บอักขระมากกว่า 1 อักขระ จำเป็นต้องกำหนดให้ตัวแปรนั้น ๆ อยู่ในรูปแบบของตัวแปรชนิดข้อความชุด หรือที่เรียกว่า ตัวแปรอาร์เรย์ (Array) นั่นเอง และในการทำเกี่ยวกับตัวแปรที่มีชนิดเป็นแบบ สตริง หรือข้อความ ยังมีฟังก์ชันที่ช่วยในการออกแบบโปรแกรมให้ทำงานได้สะดวกยิ่งขึ้น

ฟังก์ชันที่ใช้งานเกี่ยวกับสตริง (string functions) ในภาษาซี มีฟังก์ชันที่ใช้จัดการเกี่ยวกับค่าคงที่สตริง และตัวแปรสตริงอยู่หลายฟังก์ชันดังนี้

- ฟังก์ชัน strcpy()
- ฟังก์ชัน strcmp()
- ฟังก์ชัน strlen()
- ฟังก์ชัน strcat()

โดยก่อนที่จะใช้ฟังก์ชันเหล่านี้จะต้องมีการใช้คำสั่ง #include<string.h> เข้ามาในโปรแกรมด้วยจึงจะสามารถทำงานได้

1. ฟังก์ชัน strcpy()

เป็นฟังก์ชันที่นำมาใช้เพื่อคัดลอกข้อความไปเก็บไว้ในตัวแปร หรือคัดลอกจากตัวแปรสตริงหนึ่งไปเก็บไว้ยังตัวแปรของอีกสตริงหนึ่ง มีรูปแบบการใช้ฟังก์ชันดังนี้

รูปแบบ

Strcpy (Destination , Source)

Destination หมายถึง ตัวแปรที่ใช้จัดเก็บค่าข้อมูลใน source

Source หมายถึง ข้อความหรือตัวแปรสตริงที่จะนำไปจัดเก็บไว้ในตัวแปรที่ระบุไว้ใน Destination

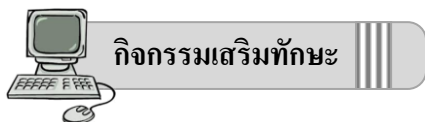
ตัวอย่าง

strcpy (bonus, "Television");

หมายถึง การคัดลอกข้อความคำว่า "Television" ไปเก็บไว้ในตัวแปรสตริงที่ชื่อว่า bonus

strcpy (award , bonus);

หมายถึง การคัดลอกข้อความจากตัวแปรสตริงที่ชื่อว่า bonus ไปเก็บไว้ในตัวแปรสตริงที่ชื่อว่า award

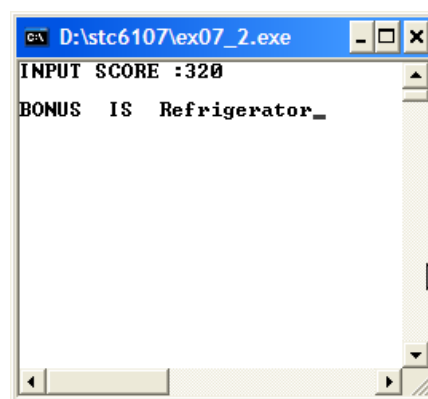


ให้นักเรียนทดลองเขียนโปรแกรมเพื่อเลือกผลลัพธ์การให้รางวัล (bonus) จากคะแนน (score) ที่ป้อนเข้าไปในโปรแกรมแล้วแสดงข้อความออกมา ในรูปแบบของข้อความชื่อของรางวัลที่จะได้รับ โดยใช้ฟังก์ชัน strcpy()

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  char bonus[15];
5  int score;
6  main( )
7  {
8      printf ("INPUT SCORE : ");
9      scanf("%d", &score);
10     if (score<=200)
11         strcpy(bonus, "Electric Fan");
12     else if (score<=300)
13         strcpy(bonus, "Television");
14     else if (score<=400)
15         strcpy(bonus, "Refrigerator");
16     else
17         strcpy(bonus, "Motorcycle");
18     printf("\nBONUS IS %s", bonus);
19     getch( );
20 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_4.exe

2. ฟังก์ชัน strlen()

เป็นฟังก์ชันที่นำมาใช้นับจำนวนตัวอักษร ที่บรรจุอยู่ในตัวแปรอาร์เรย์ มีรูปแบบการใช้ฟังก์ชันดังนี้

รูปแบบ

Strlen (StringVariable)

StringVariable หมายถึง ตัวแปร ตัวแปรชนิดสตริง

ตัวอย่าง

```
strcpy (bonus, "Television");
printf( "%d", strlen(bonus));
```

หมายถึง การคัดลอกข้อความคำว่า "Television" ไปเก็บไว้ที่ตัวแปรสตริง
ที่ชื่อว่า bonus และผลลัพธ์ที่ได้จากการนับจำนวนอักขระในตัวแปร
สตริงที่ชื่อว่า bonus เท่ากับ 10 ตัวอักษร



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมรับชื่อของนักเรียน เพื่อทำการนับอักขระทั้งหมดและจำแนกนับอักขระตัวอักษร 'a', 'e', 'i', 'o', 'u' จากชื่อการป้อนชื่อของนักเรียนโดยใช้ฟังก์ชัน strlen()

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  char name[30];
5  int i, j, n1, n2, n3, n4, n5;
6  main( )
7  {
8      printf("Input name : ");
9      scanf("%s", name);
10     printf("Number is : %d\n", strlen(name));
11     for (i=0 ; i<=strlen(name) ; i++)
12     {
13         if (name[i]=='a')
14             n1++;
15         else if(name[i]=='e')
16             n2++;
17         else if(name[i]=='i')
18             n3++;
19         else if(name[i]=='o')
20             n4++;
21         else if(name[i]=='u')
22             n5++;
23     }
24     printf("Number of 'a' is %d \n", n1);
25     printf("Number of 'e' is %d \n", n2);
26     printf("Number of 'i' is %d \n", n3);
27     printf("Number of 'o' is %d \n", n4);
28     printf("Number of 'u' is %d \n", n5);
29     getch( );
30 }

```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_5.exe

3. ฟังก์ชัน strcmp()

เป็นฟังก์ชันที่นำมาใช้เพื่อเปรียบเทียบสตริง 2 ตัว ว่าตรงกันหรือไม่ โดยหาผลตรวจสอบมีผลตรงกันก็จะรีเทิร์นค่าเป็น 0 แต่ถ้าไม่ตรงกัน ก็จะรีเทิร์นค่าที่เกิดจากผลต่างของตัวอักษร ซึ่งเป็นผลต่างระหว่างเลขรหัสเอสกีของค่าทั้งสองนั่นเอง มีรูปแบบการใช้ฟังก์ชันดังนี้

รูปแบบ

Strcmp (String1, String2)

String 1 และ String2 หมายถึง ข้อความหรือตัวแปร ที่นำมาเปรียบเทียบว่าตรงกันหรือไม่



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมทดสอบการใช้งานฟังก์ชัน strcmp()

```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <string.h>
4 char data1[]="ABC";
5 char data2[]="ABF";
6 int a , b , c;
7 main( )
8 {
9     a=strcmp(data1,"ABC");
10    b=strcmp(data1,data2);
11    c=strcmp(data2,data1);
12
13    printf(" Compare %s and %s = %d\n",data1,"ABC" , a);
14    printf(" Compare %s and %s = %d\n",data1,data2, b);
15    printf(" Compare %s and %s = %d\n",data2,data1, c);
16    getch( );
17 }
```

```

D:\stc6107\ex7_6.exe
Compare ABC and ABC = 0
Compare ABC and ABF = -1
Compare ABF and ABC = 1
```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_5.exe

4. ฟังก์ชัน strcat()

เป็นฟังก์ชันที่นำมาใช้เพื่อผนวกสตริงเข้าด้วยกัน โดยผลที่ได้จะเก็บไว้ที่ตัวแปรตัวแรก ข้อควรระวังคือ การเชื่อมสตริงจะทำให้เกิดความยาวของข้อความมากขึ้น ดังนั้นขนาดความกว้างของตัวแปรที่จัดเก็บ จะต้องมีความยาวเพียงพอต่อการจัดเก็บข้อความเพิ่มเติมได้ รูปแบบการใช้ฟังก์ชันมีรูปแบบดังนี้

รูปแบบ

Strcat (Target, Source)

Target	หมายถึง	ข้อความหรือตัวแปร ที่นำมาเปรียบเทียบกับตรงกันหรือไม่
Source	หมายถึง	ข้อความหรือตัวแปรสตริง ที่ต้องการนำมาผนวกต่อท้ายรวมกับตัวแปรใน Target



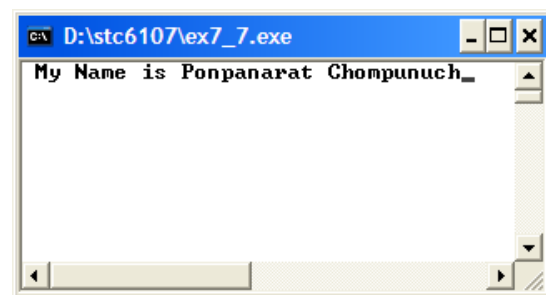
กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมทดสอบการใช้งานฟังก์ชัน strcat()

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  char name[ ]="Ponpanarat ";
5  char surname[ ]="Chompunuch";
6  main()
7  {
8      strcat(name , surname);
9      printf(" My Name is %s", name);
10     getch();
11 }

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex07_7.exe