

# แผนการสอนประจำบทที่ 7

## พอยน์เตอร์ (Pointer)

### หัวข้อสำคัญ

1. ความหมายและการทำงานของพอยน์เตอร์
2. การประกาศตัวแปรพอยน์เตอร์
3. การแสดงตำแหน่ง และค่าของข้อมูลที่พอยน์เตอร์ชี้อยู่
4. พอยน์เตอร์กับอาร์เรย์
5. อาร์เรย์ของพอยน์เตอร์
6. พอยน์เตอร์ของพอยน์เตอร์

### วัตถุประสงค์เชิงพฤติกรรมการเรียนการสอน

1. ผู้เรียนสามารถอธิบายความหมายของพอยน์เตอร์และอาร์เรย์ได้
2. ผู้เรียนสามารถอธิบายการแสดงตำแหน่ง และค่าของข้อมูลของตัวแปรที่พอยน์เตอร์ชี้อยู่ได้
3. ผู้เรียนสามารถอธิบายความหมายอาร์เรย์ของพอยน์เตอร์ได้

### วิธีการสอนและกิจกรรมการเรียนการสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

### สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์
3. เครื่องฉายภาพนิ่ง

### การวัดและประเมินผล

1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

## บทที่ 7

### พอยน์เตอร์ (Pointer)

#### 7.1.พอยน์เตอร์ (Pointer)

เมื่อมีการประกาศตัวแปรใด ๆ ขึ้นในโปรแกรม คอมไพเลอร์ของ C จะจัดการจองพื้นที่ในหน่วยความจำ ณ ตำแหน่งที่ว่างเพื่อเป็นที่เก็บข้อมูลของตัวแปรนั้นปกติผู้พัฒนาจะไม่ทราบว่าตำแหน่งที่เก็บข้อมูลนั้นอยู่ ณ ตำแหน่งใด หากต้องการทราบตำแหน่งของข้อมูลในหน่วยความจำ สามารถทำได้โดยการใช้ตัวแปรประเภทตัวชี้หรือพอยน์เตอร์ (pointer) แสดงตำแหน่งของพื้นที่ในหน่วยความจำดังกล่าวได้

เมื่อมีการประกาศตัวแปรจำนวนเต็ม อักขระ และจำนวนจริง พร้อมกำหนดค่าเริ่มต้นดังต่อไปนี้

1. int number = 100;
2. char sex = 'M';
3. float gpa = 3.21;

การจำลองการจองพื้นที่ของคอมไพเลอร์และจำลองตำแหน่งของตัวแปรในหน่วยความจำแสดงดังภาพที่ 7.1 หมายเหตุ ใช้เครื่องหมาย # นำหน้าตำแหน่งบนหน่วยความจำ

ตัวแปร	ตำแหน่งบนหน่วยความจำ	ค่าข้อมูล
number	#1001	100
sex	#1005	'M'
gpa	#1006	3.21

ภาพ 7.1 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปร

#### 7.2.การประกาศตัวแปรพอยน์เตอร์

ชนิดข้อมูล เป็นชนิดของข้อมูลพอยน์เตอร์

ชื่อตัวแปร เป็นชื่อของตัวแปรพอยน์เตอร์

รูปแบบการประกาศตัวแปร Pointer	ตัวอย่าง
ชนิดข้อมูล *ชื่อตัวแปร;	int *pt_number; à ตัวแปรพอยน์เตอร์นี้จะชี้ไปยังชนิดข้อมูลที่เป็นจำนวนเต็ม char *pt_sex; à ตัวแปรพอยน์เตอร์นี้จะชี้ไปยังชนิดข้อมูลที่เป็นตัวอักษร float *pt_gpa; à ตัวแปรพอยน์เตอร์นี้จะชี้ไปยังชนิดข้อมูลที่เป็นจำนวนจริง

### 7.3.การกำหนดค่าให้กับตัวแปรพอยน์เตอร์

การประกาศตัวแปรพอยน์เตอร์เพื่อชี้ไปยังตำแหน่งบนหน่วยความจำของตัวแปรชนิดต่างๆ การทำงานของพอยเตอร์จะทำงานดังกล่าวนั้น จำเป็นต้องให้ตัวแปรพอยเตอร์และตัวแปรที่ต้องการชี้ตำแหน่งมีชนิดข้อมูลชนิดเดียวกันเท่านั้น โดยตัวแปรแต่ละชนิดมีการจองพื้นที่ในหน่วยความจำดังต่อไปนี้

1. char จองพื้นที่ในหน่วยความจำไป 1 bytes
2. int จองพื้นที่ในหน่วยความจำไป 4 bytes
3. float จองพื้นที่ในหน่วยความจำไป 4 bytes

จากข้อมูลข้างต้นและจากภาพที่ 7.1 เมื่อมีการประกาศตัวแปรพอยเตอร์ที่ชี้ไปยังข้อมูลทั้ง 3 ชนิดเพิ่มเติมอีก 3 คำสั่ง ได้ดังต่อไปนี้ โดยภาพที่ 7.2 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปรและตัวแปรพอยน์เตอร์ทั้ง 6 คำสั่งดังกล่าว

1. int number = 100;
2. int \*pt\_number;
3. char sex = 'M';
4. char \*pt\_sex;
5. float gpa = 3.21;
6. float \*pt\_gpa;

ตัวแปร	ตำแหน่งบนหน่วยความจำ	ชนิดข้อมูล	ค่าของข้อมูล	ตัวแปรพอยน์เตอร์
number	#1001	int	100	pt_number
sex	#1005	char	'M'	pt_sex
gpa	#1006	float	3.21	pt_gpa

ภาพ 7.2 จำลองการจองพื้นที่ในหน่วยความจำของตัวแปรและตัวแปรพอยน์เตอร์

รูปแบบกำหนดค่าให้กับตัวแปรพอยน์เตอร์	ตัวอย่าง
ตัวแปรพอยน์เตอร์ = &ชื่อตัวแปรที่ต้องการทราบตำแหน่ง;	<pre>pt_number = &amp;number; pt_sex = &amp;sex; pt_gpa = &amp;gpa;</pre>

หมายเหตุ ถ้าไม่ต้องการให้ตัวแปรพอยน์เตอร์ชี้ไปที่ตำแหน่งใดสามารถกำหนดให้พอยเตอร์มีค่าเท่ากับ NULL ได้ เช่น `int *pt_number = NULL`

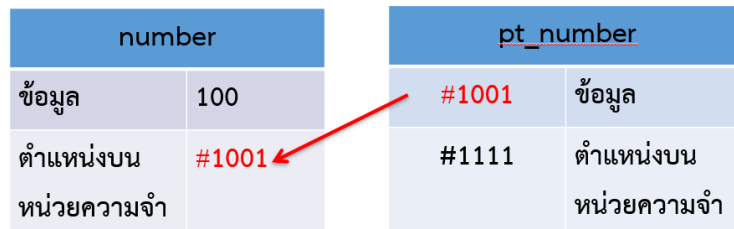
### 7.3.1 การกำหนดค่าตัวแปรโดยใช้ตัวแปรพอยน์เตอร์แบบทางตรง

เมื่อกำหนดให้ตัวแปรชนิดตัวนับเต็มชื่อ `number = 100` และตัวแปรพอยน์เตอร์ชื่อ `*pt_number` ชี้ไปยังที่อยู่ของตัวแปร `number` จะใช้คำสั่ง 3 คำสั่งดังนี้

1. `int number = 100;`
2. `int *pt_number;`
3. `pt_number = &number;`

โดยภาพจำลองการอ้างที่อยู่และข้อมูลตัวแปร `number` โดยใช้ตัวแปรพอยน์เตอร์ `*pt_number` แสดงดังภาพ 7.3

number		pt_number	
ข้อมูล	100	#1001	ข้อมูล
ตำแหน่งบนหน่วยความจำ	#1001	#1111	ตำแหน่งบนหน่วยความจำ



ภาพ 7.3 จำลองภาพการอ้างที่อยู่ของตัวแปรพอยน์เตอร์ `*pt_number`

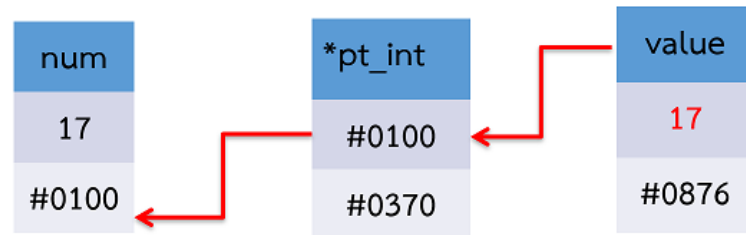
ตัวแปรพอยน์เตอร์ เป็น ตัวแปรชนิดพอยน์เตอร์ที่สร้างไว้ ชื่อตัวแปรที่ต้องการทราบตำแหน่ง โดย “ทั้งตัวแปรพอยน์เตอร์และตัวแปรที่ต้องการทราบตำแหน่งต้องเป็นประเภทข้อมูลชนิดเดียวกัน” และการแสดงผลข้อมูลใช้เครื่องหมาย `%p` เพื่อแสดงข้อมูลประเภทพอยน์เตอร์ (แสดงตำแหน่ง)

## ตัวอย่างการใช้การกำหนดค่าและการแสดงผลของตัวแปรพอยน์เตอร์

โปรแกรม
<pre> 1.  #include &lt;stdio.h&gt; 2.  main() 3.  { 4.      int number = 100; 5.      int *pt_number; 6.      pt_number = &amp;number; 7.      printf("address of counter is: %p\n", pt_number); 8. 9.      char sex = 'M'; 10.     char *pt_sex; 11.     pt_sex = &amp;sex; 12.     printf("address of sex is: %p\n", pt_sex); 13. 14.     float gpa = 3.21; 15.     float *pt_gpa; 16.     pt_gpa = &amp;gpa; 17.     printf("address of gpa is: %p", pt_gpa); 1.  }</pre>
ผลลัพธ์

## 7.3.2 การกำหนดค่าตัวแปรโดยใช้ตัวแปรพอยน์เตอร์แบบทางอ้อม

ตัวแปรพอยน์เตอร์ \*pt\_int ชี้ไปยังตัวแปร num ที่มีค่า 17 เมื่อต้องการให้ตัวแปร value มีค่าเท่ากับตัวแปร num ด้วย วิธีการคือกำหนดให้ value มีค่าเท่ากับ \*pt\_int โดยใช้คำสั่ง value = pt\_int; ดังภาพที่ 7.4 และตัวอย่างโปรแกรมด้านล่าง



ภาพ 7.4 จำลองภาพการกำหนดค่าตัวแปรโดยใช้ตัวแปรพอยน์เตอร์

#### ตัวอย่างการกำหนดค่าตัวแปรโดยใช้ตัวแปรพอยน์เตอร์

โปรแกรม
<pre> 1. #include &lt;stdio.h&gt; 2. main() 3. { 4.     int num = 17, value; 5.     int *pt_int; 6.     pt_int = &amp;num; 7.     printf("Value of num is %d\n", num ); 8.     printf("Address of num is %p\n", pt_int ); 9.     printf("Value pointer pt_int pointing is %d",*pt_int); 10.    value = *pt_int; 11.    printf("Value of value is %d\n", value ); 12. }</pre>
ผลลัพธ์

## 7.4.การแสดงค่าของข้อมูลที่ตัวแปรพอยน์เตอร์ชี้

ตัวแปรพอยน์เตอร์สามารถแสดงตำแหน่งข้อมูลของตัวแปรที่มีการชี้หรือมีการกำหนดค่าได้ นอกจากนี้ตัวแปรพอยน์เตอร์สามารถยังสามารถแสดงค่าที่อยู่ในตัวแปรที่พอยน์เตอร์ชี้ได้อีกด้วย ซึ่งการแสดงผลดังกล่าวต้องใช้คำสั่ง printf ควบคู่กันและกำหนดสัญลักษณ์การแสดงผลให้สอดคล้องกับชนิดของข้อมูลของตัวแปรด้วย

รูปแบบการแสดงค่าของข้อมูลที่ตัวแปรพอยน์เตอร์ชี้	ตัวอย่าง
<pre>printf("%format", *ตัวแปรพอยน์เตอร์);</pre> <p>format คือ รูปแบบของชนิดตัวแปร</p>	<pre>printf("%d", *pt_number); printf("%c", *pt_sex); printf("%.2", *pt_gpa);</pre>

### ตัวอย่างที่ 1 การแสดงค่าของข้อมูลที่ตัวแปรพอยน์เตอร์ชี้

โปรแกรม
<pre>1. #include &lt;stdio.h&gt; 2. main() 3. { 4.     int num = 17; 5.     int *pt_int; 6.     pt_int = &amp;num; 7.     printf("Value of num is %d\n", num); 8.     printf("Value pointer pt_int pointing is %d\n", *pt_int); 9.     printf("Address of num is: %p\n", pt_int); 10. }</pre>
ผลลัพธ์

**ตัวอย่างที่ 2 การแสดงค่าของข้อมูลที่ตัวแปรพอยน์เตอร์ชี้**

โปรแกรม
<pre>1. #include &lt;stdio.h&gt; 2. main() 3. { 4.   int val1 = 2, val2=3; 5.   int *pt_val = &amp;val1; 6.   printf("Pointer pt_val is points to val1 at address: %p\n", pt_val); 7.   printf("Value of val1 is %d and value of pt_val is %d too\n\n", val1, *pt_val); 8.   pt_val = &amp;val2; 9.   printf("Now pointer pt_val is point to val2 at address %p\n", pt_val); 10.  printf("Value of val2 is %d and now value of pt_val is %d too\n", val2, *pt_val); 11. }</pre>
ผลลัพธ์



## 7.5. ตัวแปรพอยน์เตอร์กับอาร์เรย์

พอยน์เตอร์สามารถชี้ไปยังข้อมูลในแต่ละ index ของอาร์เรย์ได้ โดยการระบุ index ของอาร์เรย์ที่ต้องการพอยน์เตอร์อ้างอิงถึง

เช่น `pt = &no[1];`

ถ้าต้องการให้พอยน์เตอร์ชี้ไปยัง index แรกของอาร์เรย์ (index=0) ทำได้โดยกำหนดชื่อของอาร์เรย์ให้กับตัวแปรพอยน์เตอร์โดยตรง คือ `pt = no;` หรือ `pt = &no[0];`

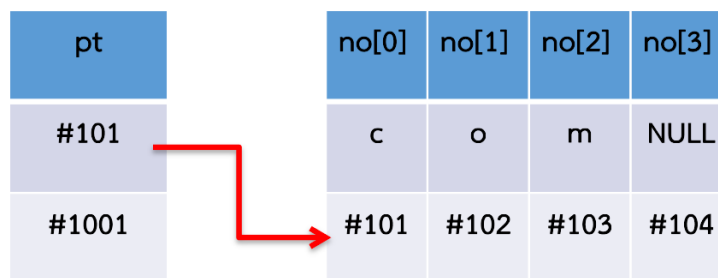
รูปแบบกำหนดค่าให้กับตัวแปรพอยน์เตอร์ชี้ไปยังอาร์เรย์	ตัวอย่าง
ตัวแปรพอยน์เตอร์ = &ชื่อตัวแปรอาร์เรย์[หมายเลขอินเด็กซ์];	<code>pt = &amp;no[1];</code> <code>pt = &amp;no[0];</code> หรือ <code>pt = no;</code> <code>pt = &amp;no[3*2];</code>

### 7.5.1 พอยน์เตอร์และการเข้าถึงข้อมูลอาร์เรย์

ถ้ามีการกำหนดค่าของตัวแปรพอยน์เตอร์และตัวแปรอาร์เรย์ ดังต่อไปนี้

1. `char no[4] = "com";`
2. `char *pt;`
3. `pt = no;` หรือ `pt = &no[0];`

จากคำสั่งด้านบนแสดงภาพจำลองการจองพื้นที่ในหน่วยความจำได้ดังภาพ 7.5 ด้านล่างนี้



ภาพ 7.5 จำลองภาพการจองพื้นที่ในหน่วยความจำของตัวแปรพอยน์เตอร์และอาร์เรย์

พอยน์เตอร์สามารถใช้การกระทำทางคณิตศาสตร์ เช่น `+`, `-`, `++`, `--` เข้ามาจัดการกับการเลื่อนตัวแปรพอยน์เตอร์ในอาร์เรย์ ดังนั้นการเลื่อนพอยน์เตอร์ไป 1 ตำแหน่ง คือ เลื่อนตำแหน่งในหน่วยความจำไปเท่ากับขนาดของชนิดตัวแปรนั้น การจำลองการเลื่อนตำแหน่งของพอยน์เตอร์แสดงดังตารางด้านล่าง ดังภาพ 7.6

คำสั่ง	ผลลัพธ์				
1. char *pt;	pt				
	#1001				
2. char no[4] = "com";	pt	no[0]	no[1]	no[2]	no[3]
		c	o	m	null
	#1001	#101	#102	#103	#104
3. pt = no;	pt	no[0]	no[1]	no[2]	no[3]
	#101	c	o	m	null
	#1001	#101	#102	#103	#104
4. pt = pt+2;	pt	no[0]	no[1]	no[2]	no[3]
	#103	c	o	m	null
	#1001	#101	#102	#103	#104
5. pt = pt-;	pt	no[0]	no[1]	no[2]	no[3]
	#102	c	o	m	null
	#1001	#101	#102	#103	#104

ภาพ 7.6 จำลองการเลื่อนตำแหน่งของตัวแปรพอยน์เตอร์และอาร์เรย์

## ตัวอย่างพอยน์เตอร์และการเข้าถึงข้อมูลอาร์เรย์-1

โปรแกรม
<pre>1. #include "stdio.h" 2. main() { 3.     char data[9] = "Computer"; 4.     char *pt = data;  5.     printf("First element of data at address %p\n", &amp;data[0]); 6.     printf("and value of first element is: %c\n\n", *pt); 7.     ++pt; 8.     printf("pt point to address %p: value %c\n", pt, *pt); 9.     --pt; 10.    printf("pt point to address %p: value %c\n", pt, *pt); 11.    pt = pt+5; 12.    printf("pt point to address %p: value %c\n", pt, *pt); 13. }</pre>
ผลลัพธ์

## ตัวอย่างพอยน์เตอร์และการเข้าถึงข้อมูลอาร์เรย์-2

โปรแกรม
<pre>1. #include "stdio.h" 2. main() { 3.     int data[9] = {10,20,30,40,50,60,70,80,90}; 4.     int *pt = data;  5.     printf("First element of data at address %p\n", &amp;data[0]); 6.     printf("and value of first element is: %d\n\n", *pt); 7.     ++pt; 8.     printf("pt point to address %p: value %d\n", pt, *pt); 9.     --pt; 10.    printf("pt point to address %p: value %d\n", pt, *pt); 11.    pt = pt+5; 12.    printf("pt point to address %p: value %d\n", pt, *pt); 13. }</pre>
ผลลัพธ์

### 7.3 อาร์เรย์ของพอยน์เตอร์

ตัวแปรพอยน์เตอร์ 1 ตัวแปร สามารถอ้างอิงไปยังตัวแปรอื่น ๆ ได้ ณ เวลาใดเวลาหนึ่งเพียงตัวแปรเดียวเท่านั้น หากต้องการให้พอยน์เตอร์ 1 ตัวแปร สามารถอ้างอิงไปยังตัวแปรอื่นๆ หลาย ๆ ตัวพร้อมกันในเวลาเดียวกัน สามารถทำได้โดยการประยุกต์ตัวแปรพอยน์เตอร์ให้อยู่ในรูปแบบของอาร์เรย์ ส่วนการอ้างอิงรายการสมาชิกของอาร์เรย์ของพอยน์เตอร์ก็ต้องใส่ [หมายเลขอินเด็กซ์] เหมือนตัวแปรอาร์เรย์ทั่วไป

รูปแบบการประกาศอาร์เรย์ของพอยน์เตอร์	ตัวอย่าง
ชนิดข้อมูล *ตัวแปรพอยน์เตอร์ [ขนาดของอาร์เรย์];	<p>int *pt[3]; -&gt;การประกาศตัวแปร pt เป็นตัวแปรอาร์เรย์ของพอยน์เตอร์ชนิดจำนวนเต็ม ขนาด 3 เซลล์</p> <p>char *pt[10]; -&gt;การประกาศตัวแปร pt เป็นตัวแปรอาร์เรย์ของพอยน์เตอร์ชนิดตัวอักษร ขนาด 10 เซลล์</p>

จากชุดคำสั่งของอาร์เรย์ของพอยน์เตอร์ประกอบด้วยการประกาศอาร์เรย์และการกำหนดค่าของอาร์เรย์ของพอยน์เตอร์แสดงดังภาพ 7.7

คำสั่ง	ผลลัพธ์															
1. int num = 1, val = 2;	<table><tr><th>num</th><th>val</th></tr><tr><td>1</td><td>2</td></tr><tr><td>#101</td><td>#105</td></tr></table>	num	val	1	2	#101	#105									
num	val															
1	2															
#101	#105															
2. int *pt[2];	<table><tr><th>pt[0]</th><th>pt[1]</th></tr><tr><td></td><td></td></tr><tr><td>#108</td><td>#112</td></tr></table>	pt[0]	pt[1]			#108	#112	<table><tr><th>num</th><th>val</th></tr><tr><td>1</td><td>2</td></tr><tr><td>#101</td><td>#105</td></tr></table>	num	val	1	2	#101	#105		
pt[0]	pt[1]															
#108	#112															
num	val															
1	2															
#101	#105															
3. pt[0] = &num;	<table><tr><th>pt[0]</th><th>pt[1]</th></tr><tr><td>#101</td><td></td></tr><tr><td>#108</td><td>#112</td></tr></table>	pt[0]	pt[1]	#101		#108	#112	<table><tr><th>num</th><th>val</th></tr><tr><td>1</td><td>2</td></tr><tr><td>#101</td><td>#105</td></tr></table>	num	val	1	2	#101	#105		
pt[0]	pt[1]															
#101																
#108	#112															
num	val															
1	2															
#101	#105															
4. pt[1] = &val;	<table><tr><th>pt[0]</th><th>pt[1]</th></tr><tr><td>#101</td><td>#105</td></tr><tr><td>#108</td><td>#112</td></tr></table>	pt[0]	pt[1]	#101	#105	#108	#112	<table><tr><th>num</th><th>val</th></tr><tr><td>1</td><td>2</td></tr><tr><td>#101</td><td>#105</td></tr></table>	num	val	1	2	#101	#105		
pt[0]	pt[1]															
#101	#105															
#108	#112															
num	val															
1	2															
#101	#105															

ภาพที่ 7.7 จำลองการประกาศและการกำหนดค่าของอาร์เรย์ของพอยน์เตอร์

## ตัวอย่างอาร์เรย์ของพอยน์เตอร์

โปรแกรม
<pre> 1. #include&lt;stdio.h&gt; 2. main(){ 3.     int data[5] = {1, 2, 3, 4, 5}; 4.     int *p_data[5]; 5.     int i; 6.     for(i=0; i&lt;5; i++){ 7.         p_data[i] = &amp;data[i]; 8.         printf("Address of data[%d]: %p\n", i, p_data[i]); 9.         printf("Value of data[%d]: %d\n", i, *p_data[i]); 10.    } 11. }</pre>
ผลลัพธ์

## 7.6.พอยน์เตอร์ของพอยน์เตอร์

พอยน์เตอร์ของพอยน์เตอร์ เรียกอีกอย่างว่า indirect pointer เป็นตัวแปรพอยน์เตอร์ตัวหนึ่งที่ทำหน้าที่ชี้ไปยังตัวแปรพอยน์เตอร์อีกตัวแทนที่จะชี้ไปยังตัวแปรอื่นโดยตรง หากต้องการทราบตำแหน่งของพอยน์เตอร์ของพอยน์เตอร์ต้องใช้ \* จำนวน 2 ตัว ตอนประกาศตัวแปร และหากต้องการทราบตำแหน่งของพอยน์เตอร์ของพอยน์เตอร์ของพอยน์เตอร์ต้องใช้ \* จำนวน 3 ตัว ตอนประกาศตัวแปร ดังนั้นเครื่องหมาย \* จะเป็นจำนวนครั้งที่ชี้ไปยังตัวแปร

รูปแบบการประกาศ Indirect pointer	ตัวอย่าง
ชนิดข้อมูล <b>**ตัวแปรพอยน์เตอร์</b>	<pre>int <b>**</b>pt_number; char <b>**</b>pt_sex; int <b>**</b>pt_gpa;</pre>

จากชุดคำสั่งของพอยน์เตอร์ของพอยน์เตอร์ประกอบด้วยการประกาศอาร์เรย์และการกำหนดค่าของพอยน์เตอร์แสดงดังภาพ 7.8

คำสั่ง	ผลลัพธ์		
char num = 'A';	<div>num</div> <div>A</div> <div>#200</div>		
char *pt1;	<div>pt1</div> <div></div> <div>#300</div>	<div>num</div> <div>A</div> <div>#200</div>	
char **pt2;	<div>pt2</div> <div></div> <div>#400</div>	<div>pt1</div> <div></div> <div>#300</div>	<div>num</div> <div>A</div> <div>#200</div>
pt1 = &num;	<div>pt2</div> <div></div> <div>#400</div>	<div>pt1</div> <div>200</div> <div>#300</div>	<div>num</div> <div>A</div> <div>#200</div>
pt2 = &pt1;	<div>pt2</div> <div>300</div> <div>#400</div>	<div>pt1</div> <div>200</div> <div>#300</div>	<div>num</div> <div>A</div> <div>#200</div>

ภาพที่ 7.8 จำลองการประกาศและการกำหนดค่าของพอยน์เตอร์ของพอยน์เตอร์

## ตัวอย่างพอยน์เตอร์ของพอยน์เตอร์

โปรแกรม
<pre>1. #include&lt;stdio.h&gt; 2. main() { 3.   int num = 10; 4.   int *pt_num; 5.   int **pt_pt_num; 6.   pt_num = &amp;num; 7.   pt_pt_num = &amp;pt_num; 8.   printf("address of num is %p and value of num is %d\n", pt_num, *pt_num); 9.   printf("address of pt_num is %p and value of num is %d", pt_pt_num, **pt_pt_num); 10. }</pre>
ผลลัพธ์



### คำถามท้ายบทที่ 7

1. จงอธิบายความหมายและประโยชน์ของพอยน์เตอร์
2. จงอธิบายความหมายของสัญลักษณ์ %p ของพอยน์เตอร์
3. จงอธิบายความหมายพอยน์เตอร์ของพอยน์เตอร์
4. จงอธิบายความหมายของสัญลักษณ์ \* ของ พอยน์เตอร์ของพอยน์เตอร์ (indirect pointer)
5. จงประมวลผลโปรแกรมต่อไปนี้พร้อมทั้งแสดงผลลัพธ์ของโปรแกรมที่ปรากฏบนหน้าจอคอมพิวเตอร์

โปรแกรม	ผลลัพธ์ (แสดงผลลัพธ์ตามบรรทัดบนหน้าจอคอมพิวเตอร์)
<pre> 1. #include &lt;stdio.h&gt; 2. main() 3. { 4.     float arr[5] = {12.5, 10.0, 13.5, 90.5, 0.5}; 5.     float *ptr1 = &amp;arr[0]; 6.     float *ptr2 = ptr1 + 3; 7.     printf("%f ", *ptr2); 8.     printf("%d", ptr2 - ptr1); 9. }</pre>	<pre> 1. .... 2. .... 3. .... 4. .... 5. ....</pre>

6. จงแสดงผลลัพธ์ของโปรแกรกดังต่อไปนี้

โปรแกรม	ผลลัพธ์ (แสดงผลลัพธ์ตามบรรทัดบนหน้าจอคอมพิวเตอร์)
<pre> 1. #include &lt;stdio.h&gt; 2. main() 3. { 4.     char data[8] = "computer"; int i; 5.     char *ptr = &amp;data[0]; 6.     for (i=0;i&lt;8;i+=2) 7.         printf("pt point to address %p: value            %c\n", ptr, *ptr); 8.     ptr = ptr+1; } 9. }</pre>	<pre> 1. .... 2. .... 3. .... 4. .... 5. ....</pre>

## 7. จงแสดงผลลัพธ์ของโปรแกรกดังต่อไปนี้

โปรแกรม	ผลลัพธ์ (แสดงผลลัพธ์ตามบรรทัดบนหน้าจอคอมพิวเตอร์)
1. #include <stdio.h>	
2. main()	
3. {	1. ....
4.     int *ptr;	2. ....
5.     int num;	3. ....
6.	4. ....
7.     ptr = &num;	5. ....
8.     *ptr = 0;	6. ....
9.	7. ....
10.    printf("num = %d", num);	8. ....
11.    printf(" *ptr = %d\n", *ptr);	9. ....
12.	10. ....
13.    *ptr += 5;	11. ....
14.    printf("num = %d", num);	12. ....
15.    printf(" *ptr = %d\n", *ptr);	
16.	
17.    (*ptr)++;	
18.    printf("num = %d", num);	
19.    printf(" *ptr = %d\n", *ptr);	
20.	
21. }	

## 8. จงเติมโปรแกรมให้สมบูรณ์

```
First Element is at address: 00000000023FE30, value is: 10
Now pointer is point to address: 00000000023FE34, value is: 20
Now pointer is point to address: 00000000023FE38, value is: 30
Now pointer is point to address: 00000000023FE34, value is: 20
Now pointer is point to address: 00000000023FE3C, value is: 40
Now pointer is point to address: 00000000023FE30, value is: 10
```

1. #include<stdio.h>
2. main()
3. {
4.     int data[5] = {10,20,30,40,50};
5.     int \*pt\_data;
6.     pt\_data = &data[0];
7.     printf("First Element is at address: %p, value is: %d\n", pt\_data, \*pt\_data);
8.     \_\_\_\_\_;
9.     printf("Now pointer is point to address: %p, value is: %d\n", pt\_data, \*pt\_data);
10.    \_\_\_\_\_;
11.    printf("Now pointer is point to address: %p, value is: %d\n", pt\_data, \*pt\_data);
12.    \_\_\_\_\_;
13.    printf("Now pointer is point to address: %p, value is: %d\n", pt\_data, \*pt\_data);
14.    \_\_\_\_\_;
15.    printf("Now pointer is point to address: %p, value is: %d\n", pt\_data, \*pt\_data);
16.    \_\_\_\_\_;
17.    printf("Now pointer is point to address: %p, value is: %d\n", pt\_data, \*pt\_data);
18. }

9. จงเขียนโปรแกรมสำหรับวนรับค่าจากแป้นคีย์บอร์ดเป็นจำนวนเต็ม ชื่อ data[5] จำนวน 5 ค่าโดยใช้อาร์เรย์ และต้องใช้ ตัวแปรพอยน์เตอร์ ชื่อ pt\_data ในการชี้ไปยังตำแหน่งของจำนวนเต็มที่ได้รับค่ามา จากนั้นให้นิสิตแสดงผลลัพธ์เป็นค่าและตำแหน่ง ของจำนวนเต็มที่ได้รับเข้ามา โดยมีข้อกำหนดดังต่อไปนี้

- กำหนดให้มีการแสดงผลแบบย้อนกลับจากหลังสุดมายังหน้าสุด
- ใช้ตัวแปรพอยน์เตอร์ ชื่อ pt\_data ในการแสดงผลเท่านั้น
- รูปแบบการแสดงผลดังตัวอย่างหน้าจอด้านล่าง

```

Insert Number 1: 10
Insert Number 2: 20
Insert Number 3: 30
Insert Number 4: 40
Insert Number 5: 50

----- Result -----
position 5 => address: 000000000062FE30 and value is: 50
position 4 => address: 000000000062FE2C and value is: 40
position 3 => address: 000000000062FE28 and value is: 30
position 2 => address: 000000000062FE24 and value is: 20
position 1 => address: 000000000062FE20 and value is: 10

```

10. จงเขียนโปรแกรมสำหรับวนรับค่าจากแป้นคีย์บอร์ดเป็นจำนวนเต็ม ชื่อ `data[10]` จำนวน 10 ค่าโดยใช้อาร์เรย์และต้องใช้ ตัวแปรอาร์เรย์ของพอยน์เตอร์ ชื่อ `pt_data[10]` ในการชี้ไปยังตำแหน่งของจำนวนเต็มที่ได้รับค่ามา จากนั้นให้นิสิตแสดงผลลัพธ์เป็นค่าและตำแหน่ง ของจำนวนเต็มที่ได้รับเข้ามา โดยมีข้อกำหนดดังต่อไปนี้

- กำหนดให้มีการแสดงผลแบบย้อนกลับจากหลังสุดมายังหน้าสุด
- เลือกแสดงข้อมูลเฉพาะค่า index ของ `pt_data` ที่เป็นเลขคู่เท่านั้น
- ใช้ตัวแปรพอยน์เตอร์ ชื่อ `pt_data[10]` ในการแสดงผลเท่านั้น
- รูปแบบการแสดงผลดังตัวอย่างหน้าจอด้านล่าง

```

Insert Number Number 1: 1
Insert Number Number 2: 20
Insert Number Number 3: 30
Insert Number Number 4: 40
Insert Number Number 5: 50
Insert Number Number 6: 60
Insert Number Number 7: 70
Insert Number Number 8: 80
Insert Number Number 9: 90
Insert Number Number 10: 100

----- Result -----
position 10 => address: 000000000062FE44 and value is: 100
position 8 => address: 000000000062FE3C and value is: 80
position 6 => address: 000000000062FE34 and value is: 60
position 4 => address: 000000000062FE2C and value is: 40
position 2 => address: 000000000062FE24 and value is: 20

```