

Computer Engineering

วิศวกรรมคอมพิวเตอร์



บทที่ 9 ตัวแปรแบบตัวชี้ตัวแหน่ง (Pointers) และฟังก์ชัน 2

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

9.1 โครงสร้างของหน่วยความจำ และตัวชี้

01006012 Computer Programming

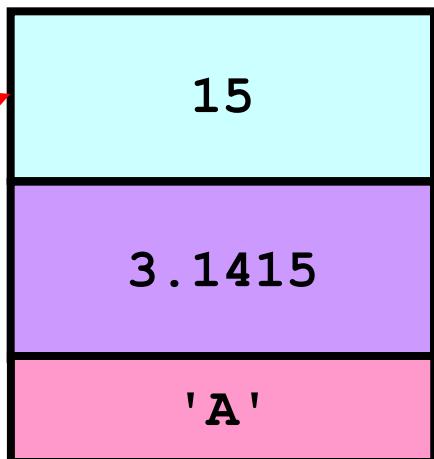
ที่อยู่ของข้อมูล
(Address)

0000

0004

0008

xxxx



ข้อมูล

ประเภท

ชื่อตัวแปร

ขนาด

int

count

4 bytes

float

pi

4 bytes

char

ch

1 bytes

ptr = &count
ตัวชี้ ptr ซึ่งไปที่อยู่ของ
ตัวแปรชื่อ count

9.1.1 การประกาศตัวชี้

01006012 Computer Programming

type *pointer_name

type คือ ชนิดของตัวแปรประเภทตัวชี้ (pointer)

***** คือ เครื่องหมายแสดงว่าเป็นตัวแปรประเภทตัวชี้

pointer_name คือ ชื่อของตัวแปรประเภทตัวชี้

- ตัวอย่างการใช้ตัวชี้

int * ptr_int; /* pointer to integer */

float * ptr_float; /* pointer to float */

char * ptr_char; /* pointer to char */

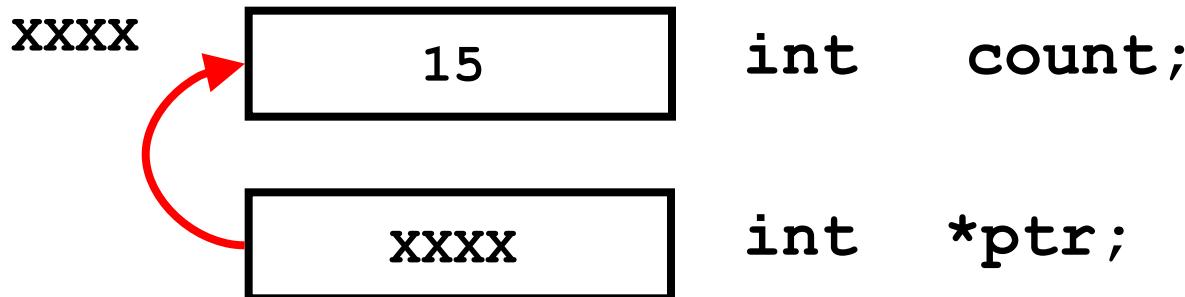
9.1.2 ตัวดำเนินการ (Reference Operator) “&”



01006012 Computer Programming

ตัวดำเนินการ & (Referencing Operator)

ตัวดำเนินการ & ส่งกลับเลขที่อยู่ (Address) ของวัตถุ



`ptr = &count; /* ptr มีค่าเป็น xxxx */`

9.1.2 ตัวดำเนินการ (Reference Operator) “&”



01006012 Computer Programming

ที่อยู่ของข้อมูล (Address)	ข้อมูล	ประเภท	ชื่อตัวแปร	ขนาด
0000	15	int	count	4 bytes
0004	3.1415	float	pi	4 bytes
0008	'A'	char	ch	1 bytes
	0008	char	*ptr_char = &ch;	
	0004	float	*ptr_float = π	
	0000	int	*ptr_int = &count;	

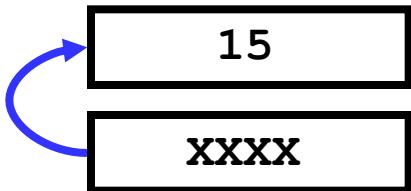
9.1.3 ตัวดำเนินการเชิงอ้อม (Indirect Operator) “*”



01006012 Computer Programming

เป็น ตัวดำเนินการเชิงอ้อม (Indirection)
หรือ กลับการอ้างอิง (Dereferencing)

ตำแหน่งที่ XXXX



```
int count;  
int * ptr;
```

```
int count = 15, y, z[10];  
int *ptr; /* ptr เป็นตัวชี้ int */  
  
ptr = &count; /* ptr ชี้ไปที่ count */  
y = *ptr; /* y มีค่า 15 */  
*ptr = 0; /* count มีค่า 0 */  
ptr = &z[0]; /* ptr ชี้ไปที่ z[0] */
```

9.2 การแสดงตัวชี้ด้วยฟังก์ชัน printf

- ฟังก์ชัน printf สามารถแสดงตำแหน่งที่อยู่ (address) ได้โดยใช้
 - เครื่องหมาย %p เพื่อแสดงตำแหน่งเป็นเลขฐานสิบหก
 - เครื่องหมาย %u เพื่อแสดงตำแหน่งเป็นเลขฐานสิบ
- ผลลัพธ์ที่ได้จะอยู่ในรูปแบบ XXXX:YYYY หรือ XXXX ขึ้นอยู่กับ memory model ที่ใช้

```

1 #include <stdio.h>
2 int main()
3 {
4     int i = 10;
5     int *p;
6     p = &i;
7     printf("%p %d\n", &i, i);
8     printf("%p %p %d\n", &p, p, *p);
9     printf("-----\n");
10    printf("%u %d\n", &i, i);
11    printf("%u %u %d\n", &p, p, *p);
12    return 0;
13 }
  
```

C:\WINDOWS\system32\cmd.exe

0012FF60	10	
0012FF54	0012FF60	10
<hr/>		
1245024	10	
1245012	1245024	10

Press any key to continue .

โปรแกรม 9.1 โปรแกรมแสดงข้อมูลผ่านทาง pointer



01006012 Computer Programming

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char      letter = 'D';
    int       num    = 19;
    float     point  = 26.09;
    char      *pt_letter;
    int       *pt_num;
    float     *pt_point;
    pt_letter = &letter;
    pt_num   = &num;
    pt_point = &point;
    printf("Address of letter = %p \n",pt_letter);
    printf("Address of num    = %p \n",pt_num);
    printf("Address of point  = %p \n",pt_point);
    return 0;
}
```

letter	num	point
D	19	26.09
0000	0004	0008

pt_letter	pt_num	pt_point
0000	0004	0008
????	????	????

โปรแกรม 9.1 โปรแกรมแสดงข้อมูลผ่านทาง pointer



01006012 Computer Programming

Address of letter = 0000
Address of num = 0004
Address of point = 0008

letter	num	point
D	19	26.09
0000	0004	0008

pt_letter	pt_num	pt_point
0000	0004	0008
????	????	????

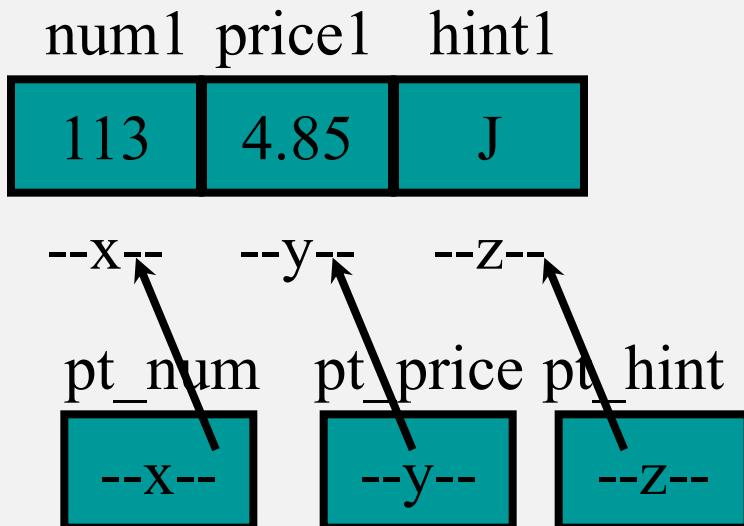
ໂປຣແກນ 9.2 ໂປຣແກນການໃໝ່ referencing ແລະ dereferencing

01006012 Computer Programming

```

int main()
{
    int num1 = 113, num2;
    float price1 = 4.85;
    char hint1 = 'J', hint2;
    int *pt_num;
    float *pt_price;
    char *pt_hint;
    pt_num = &num1;
    pt_price = &price1;
    pt_hint = &hint1;
    num2 = *pt_num;
    hint2 = *pt_hint;
    printf ("Variable num1 = %d \n", num2);
    printf ("Variable price1 = %f \n", *pt_price);
    printf ("Variable hint2 = %c \n", hint2);
    return 0;
}

```



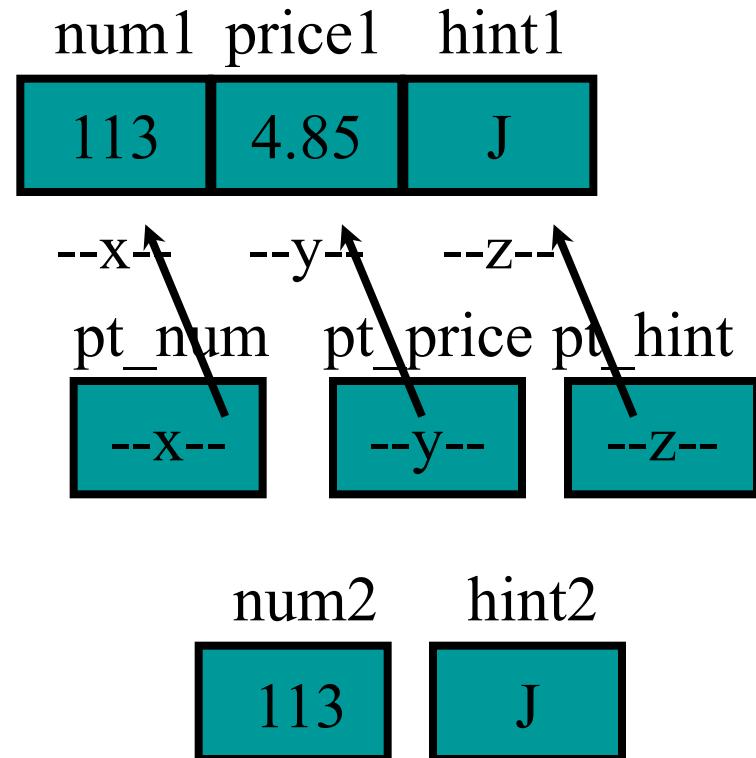
num2	hint2
113	J

โปรแกรม 9.2 โปรแกรมการใช้ referencing & dereferencing



01006012 Computer Programming

```
Variable num1 = 113  
Variable price1 = 4.850000  
Variable hint2 = J
```



ໂປຣແກຣມ 9.3 ໂປຣແກຣມການໃໝ່ referencing ແລະ dereferencing ມູນ 2

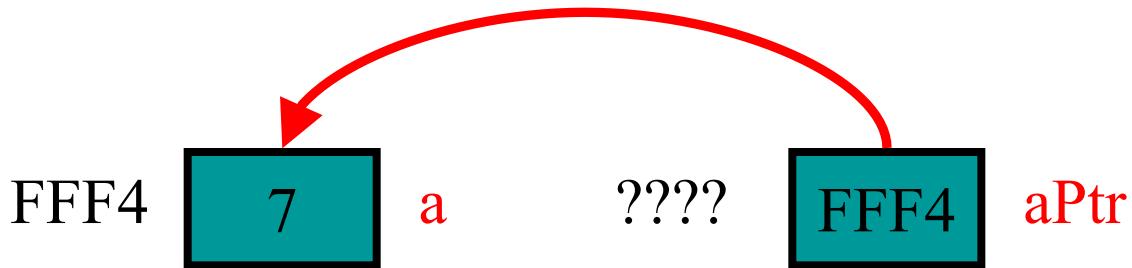
01006012 Computer Programming

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int    a;          /* a is an integer */
    int    *aPtr;      /* aPtr is a pointer to an integer */
    a = 7;
    aPtr = &a;        /* aPtr set to address of a */
    printf ("The address of a is %p\n"
            "The value of aPtr is %p\n\n", &a, aPtr);
    printf ("The value of a is %d\n"
            "The value of *aPtr is %d\n\n", a, *aPtr);
    return 0;
}
```

ໂປຣແກຣມ 9.3 ໂປຣແກຣມການໃຊ້ referencing ແລະ dereferencing ໂດຍ 2



01006012 Computer Programming



The address of **a** is **FFF4**

The value of **aPtr** is **FFF4**

The value of **a** is **7**

The value of ***aPtr** is **7**

สรุปการดำเนินการของตัวแปรประเภทตัวชี้

01006012 Computer Programming

การประกาศตัวแปรพอยน์เตอร์

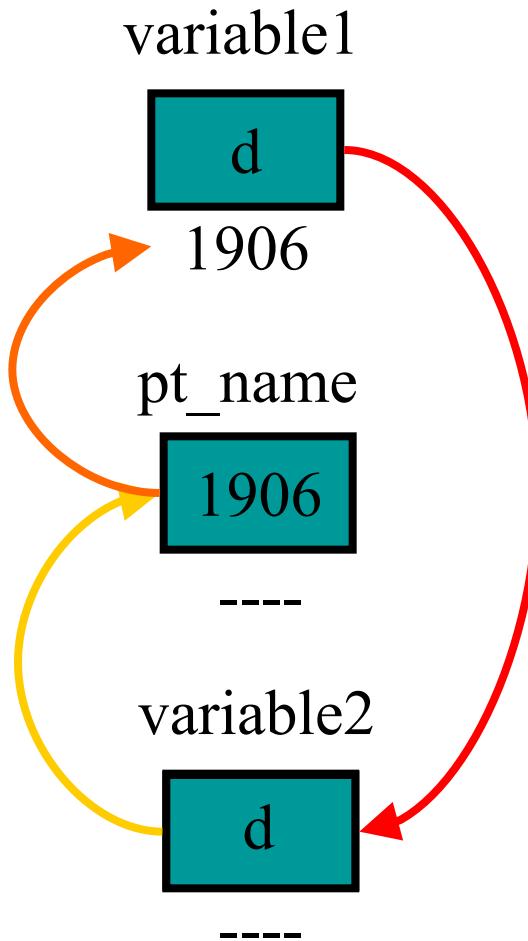
```
type *pt_name;
```

แสดงตำแหน่งข้อมูลด้วย &

```
pt_name = &variable1;
```

แสดงข้อมูลด้วย *

```
variable2 = *pt_name;
```



9.3 Pointer ชี้อัน Pointer

01006012 Computer Programming

```
type **ptt_name;
```

Type คือ ชนิดของตัวแปร พอยน์เตอร์

** คือ เครื่องหมายที่แสดงว่าเป็นตัวแปร พอยน์เตอร์ชี้อันพอยน์เตอร์

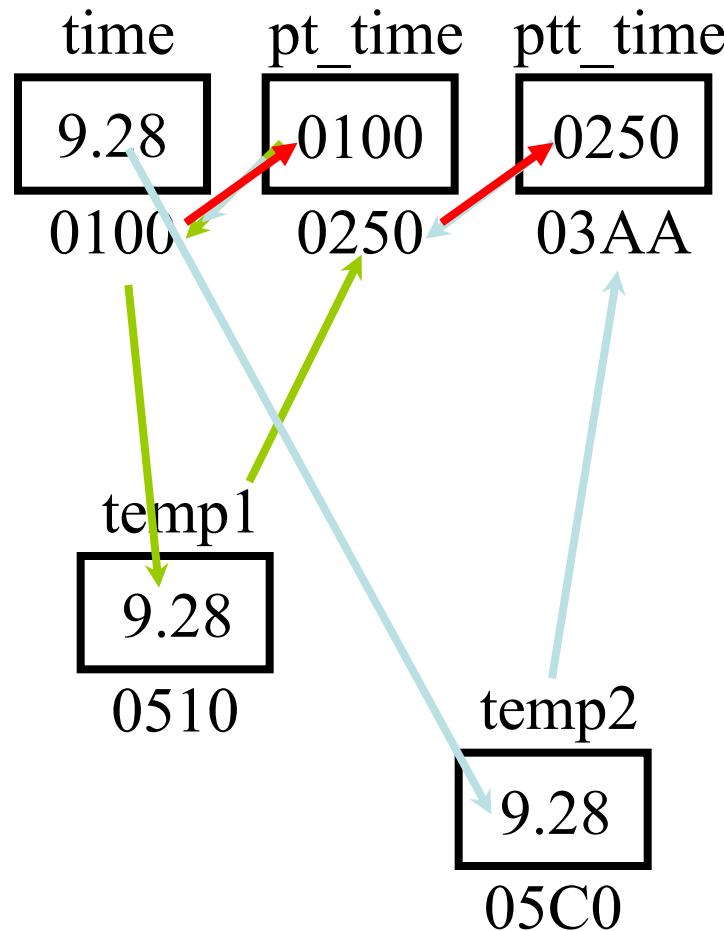
ptt_name คือ ชื่อของตัวแปร พอยน์เตอร์ชี้อันพอยน์เตอร์

9.3 Pointer ڦوں Pointer

01006012 Computer Programming

```

float time = 9.28;
float *pt_time;
float **ptt_time;
pt_time = &time;
ptt_time = &pt_time;
float temp1;
temp1 = *pt_time;
float temp2;
temp2 = **ptt_time;
    
```



9.4 ตัวชี้และแவ่ลำดับ

01006012 Computer Programming

ตัวชี้และแவ่ลำดับในภาษาซีนั้นมีความใกล้ชิดกันอย่างมาก การประกาศ

float v[5]

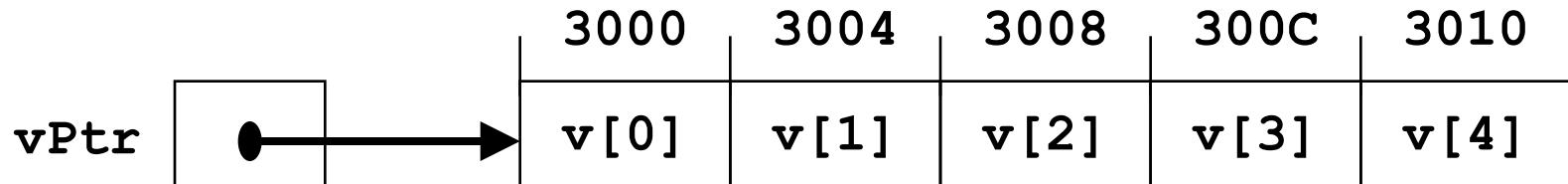
เป็นการกำหนดแவ่ลำดับ **v** ขนาด 5 นั่นคือกลุ่มของวัตถุติดกัน 5 ชิ้น มีชื่อว่า
v[0] , v[1] , v[2] , v[3] , v[4]

3000 3004 3008 300C 3010

v[0]	v[1]	v[2]	v[3]	v[4]
-------------	-------------	-------------	-------------	-------------

9.4 ตัวชี้และแเรวลำดับ

01006012 Computer Programming



การประกาศ **float *vPtr** และกำหนดให้ **vPtr** ชี้ไปยังตัวแปรแล้ว
ลำดับ **v** สามารถทำได้สองวิธีคือ

วิธีที่ 1

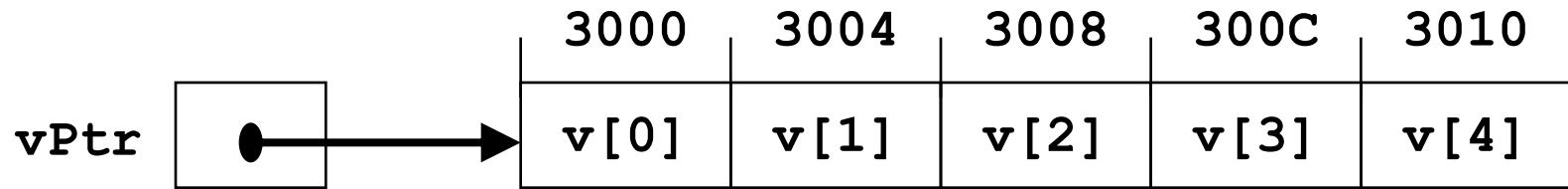
```
vPtr = v;
```

วิธีที่ 2

```
vPtr = &v[0];
```

9.4 ตัวชี้และแคล็บ

01006012 Computer Programming



การกำหนด **x = *vPtr** คือการสำเนาค่าใน **v[0]** มาอยู่ **x**

9.4.1 การย้าย Pointer ที่ชี้ตัวแปรอาร์เรย์ ด้วย += และ -=



01006012 Computer Programming

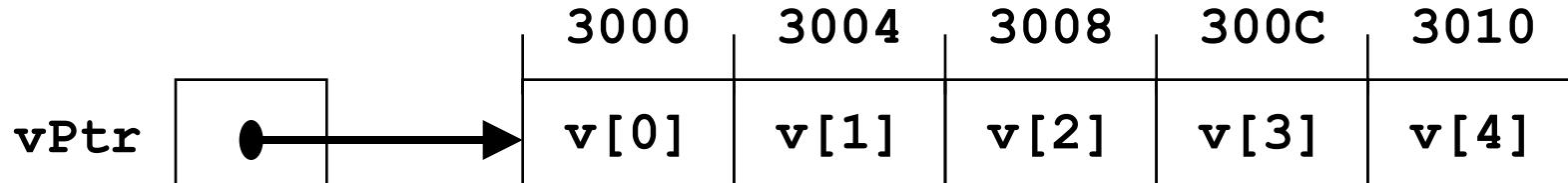
ตัวชี้สามารถนำมารคำนวณทางเลขคณิตได้ ดังนี้

- เพิ่ม (++, increment) หรือลด (--, decrement)
- จำนวนเต็มสามารถบวกกับตัวชี้ได้ (+ หรือ +=)
หรือลบกับตัวชี้ได้ (- หรือ -=)
- ตัวชี้ตัวหนึ่งสามารถลบกับตัวชี้อีกตัวหนึ่งได้

9.4.1 การย้าย Pointer ที่ชี้ตัวแปรอาร์เรย์ ด้วย += และ -=



01006012 Computer Programming

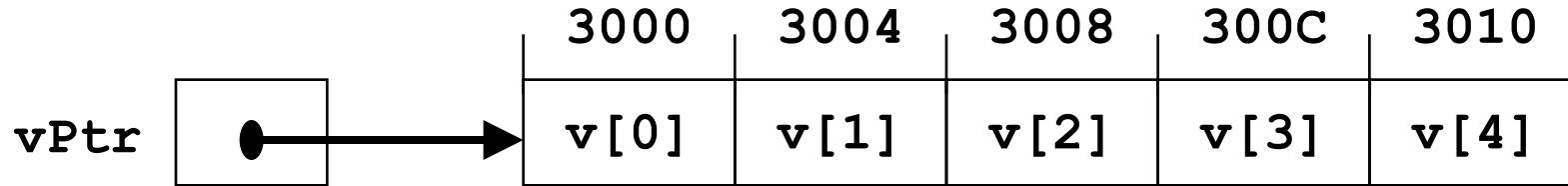


- เมื่อบวกหรือลบจำนวนเต็มกับตัวชี้แล้ว
 - ค่าของตัวชี้มีได้เพิ่มหรือลดลงตามตัวเลขจำนวนนั้น
 - ค่าของตัวชี้เพิ่มหรือลดตามตัวเลขจำนวนนั้นคุณกับขนาดของวัตถุที่ตัวชี้นั้นชื่อยู่
 - ขนาด (ไบท์) ขึ้นกับประเภทของข้อมูลที่ใช้ในวัตถุนั้น
- ตัวอย่าง (กำหนดขนาดของวัตถุ ชนิด float คือ 4 ไบท์)
 - $vPtr += 2; // vPtr = vPtr + (2 * 4)$
// or $vPtr = 3000 + (2 * 4)$

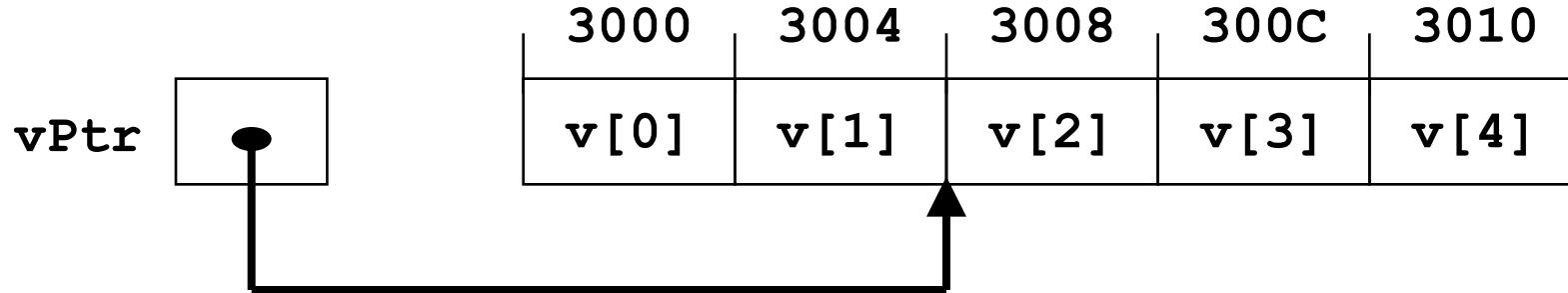
9.4.1 การย้าย Pointer ที่ชี้ตัวแปรอาร์เรย์ ด้วย += และ -=



01006012 Computer Programming



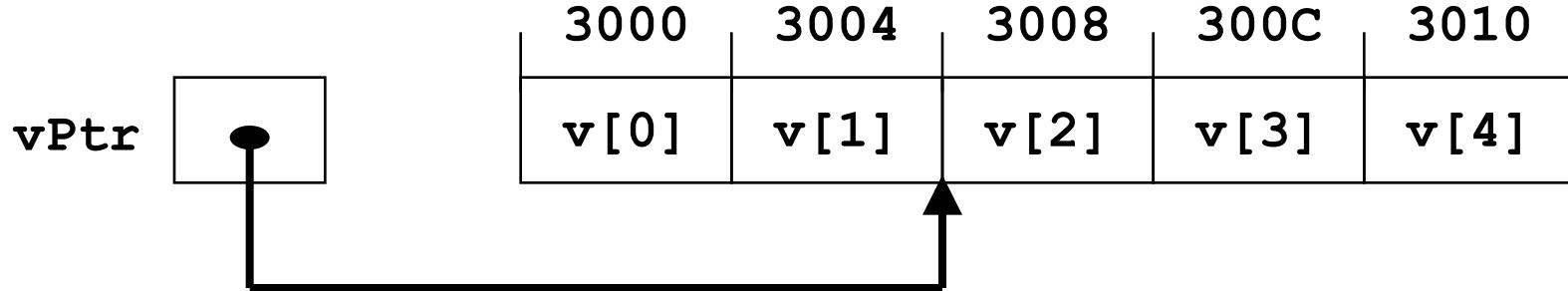
- **vPtr += 2;** // $vPtr = vPtr + (2 * 4)$
 // or $vPtr = 3000 + (2 * 4)$
- หลังจากคำสั่งข้างต้นแล้ว **vPtr** จะชี้ไปที่ **v[2]**



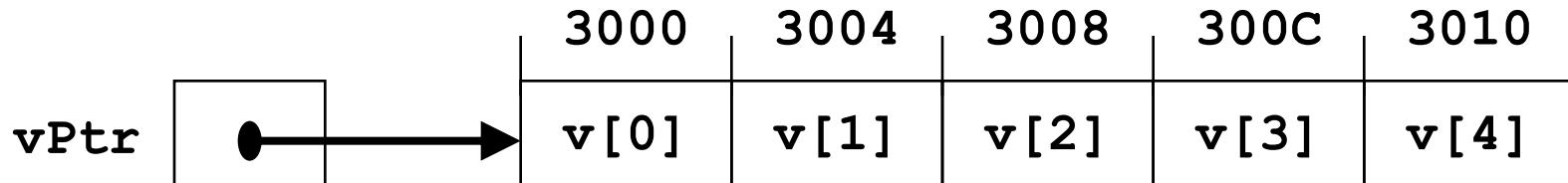
9.4.1 การย้าย Pointer ที่ชี้ตัวแปรอาร์เรย์ ด้วย += และ -=



01006012 Computer Programming



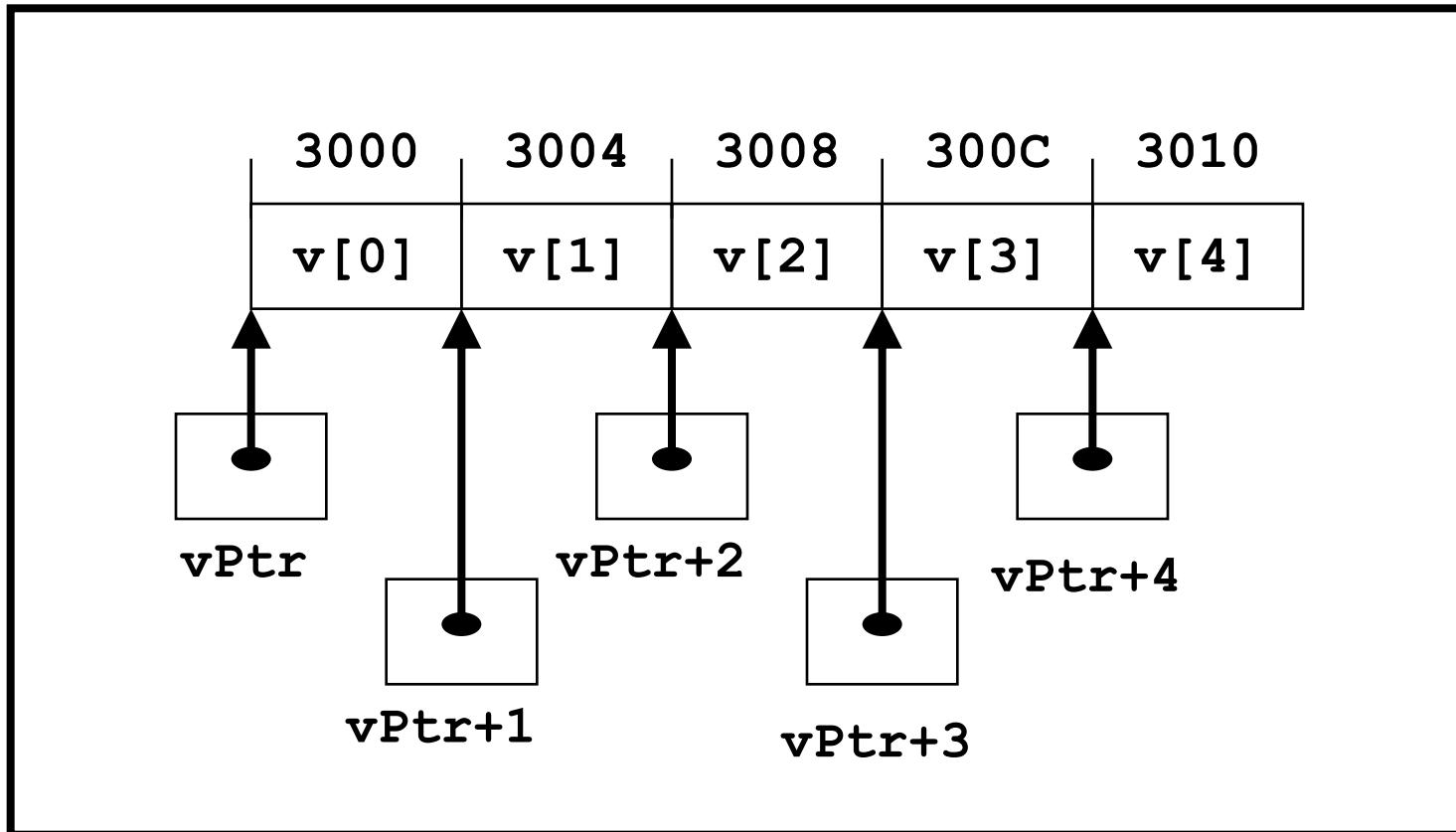
- **vPtr -= 2;** // **vPtr = vPtr - (2 * 4)**
 // or **vPtr = 3008 - (2 * 4)**
- หลังจากคำสั่งข้างต้นแล้ว **vPtr** จะชี้ไปที่ **v[0]**



9.4.2 การย้าย Pointer ที่ชี้ตัวแปรอาร์เรย์ ด้วย + และ -



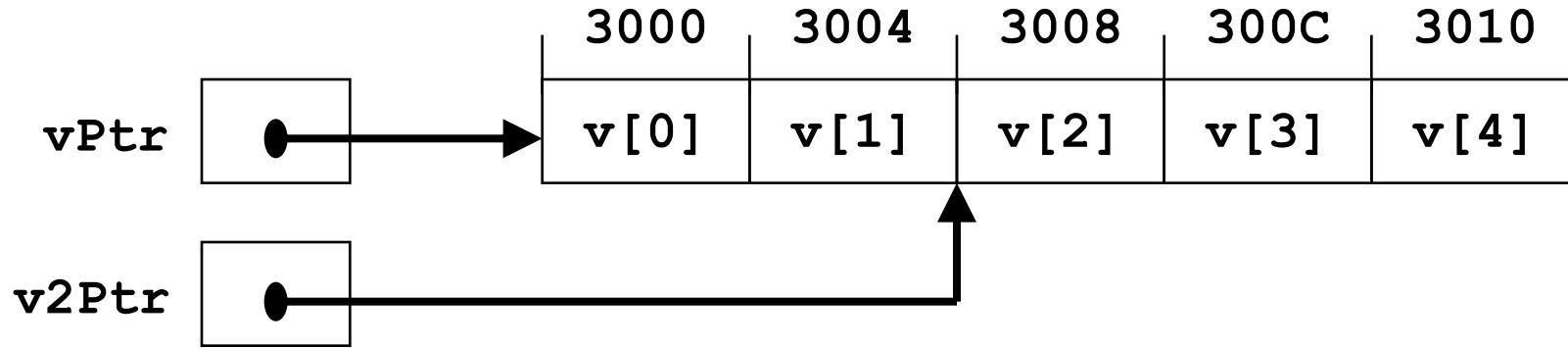
01006012 Computer Programming



9.4.3 การคำนวณจำนวน element ด้วย pointer



01006012 Computer Programming



- **vPtr = &v[0];** // **vPtr = 3000**
- **v2Ptr = &v[2];** // or **v2Ptr = 3008**
- **x = v2Ptr - vPtr;** // **x = ?**

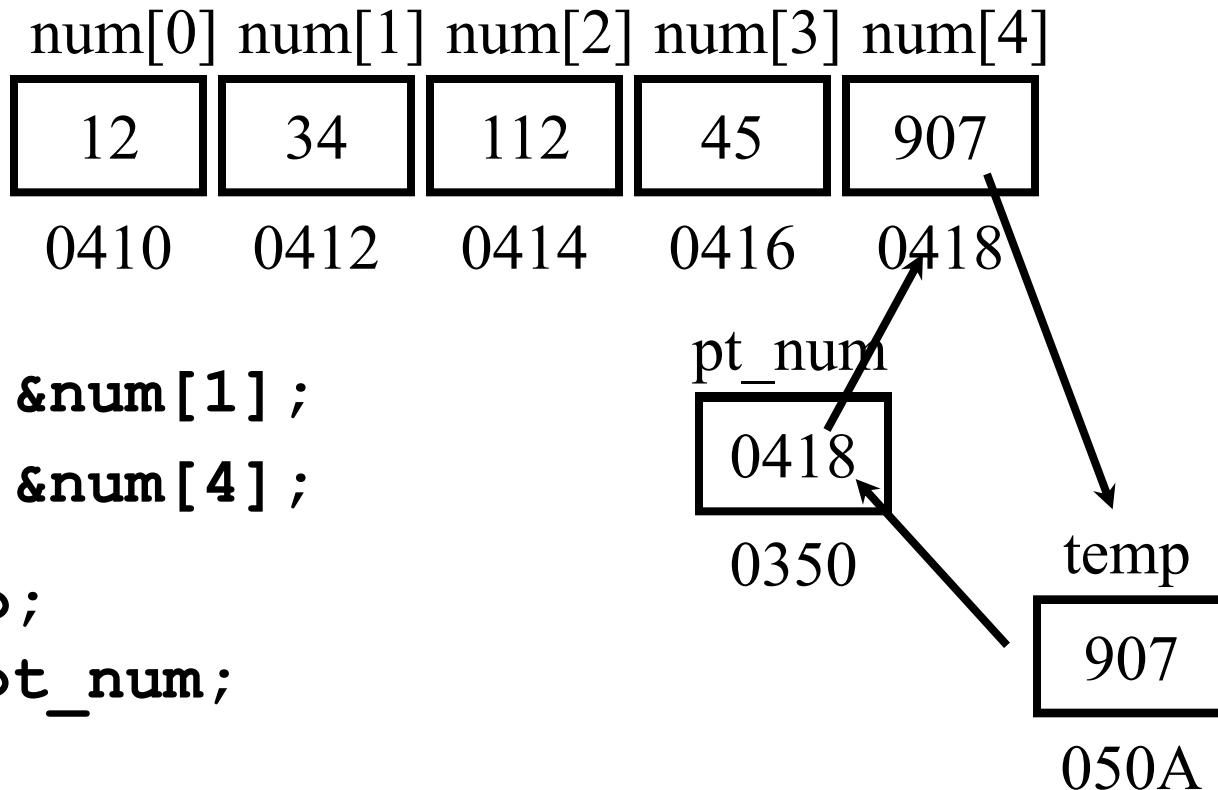
ค่าที่ **x** ได้รับคือจำนวนหน่วย (element) ของตัวแปรແຄວลำดับนั้น
จาก **vPtr** ถึง **vPtr2** ในกรณีนี้คือ 2

ໂປຣແກຣມ 9.4 ໂປຣແກຣມແສດງການໃໝ່ pointer ກັບ array



01006012 Computer Programming

```
int num[5] = {12, 34, 112, 45, 907};  
int *pt_num;
```



```
pt_num = &num[1];  
pt_num = &num[4];  
  
int temp;  
temp = *pt_num;
```

```
type name[10];
type *pt_name;
pt_name = name;           //pt_name = &name[0]
```

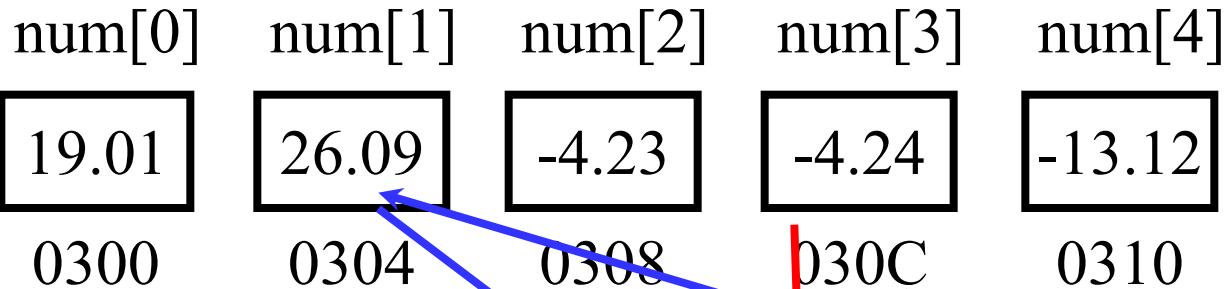
```
float num[] = {19.01, 26.09, -4.23, -4.24, -13.12}
```

num[0]	num[1]	num[2]	num[3]	num[4]
19.01	26.09	-4.23	-4.24	-13.12
0300	0304	0308	030C	0310

โปรแกรม 9.5 โปรแกรมแสดงการใช้ pointer กับ array



01006012 Computer Programming



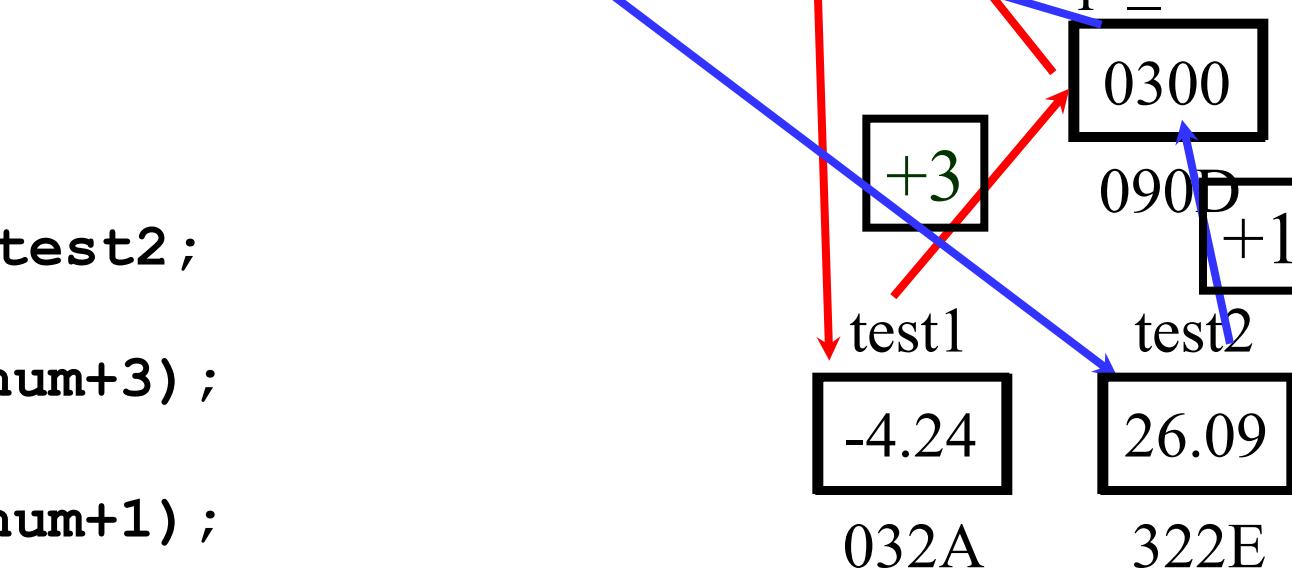
```
float *pt_num;
```

```
pt_num = num;
```

```
float test1, test2;
```

```
test1 = *(pt_num+3);
```

```
test2 = *(pt_num+1);
```



9.5 ความสัมพันธ์ระหว่าง Pointer กับ Array



01006012 Computer Programming

- เห็นได้ว่าตัวชี้และตัวแปรແ胄ດຳນັບມີຄວາມສັນພັນຮັກນ ແລະ ສາມາດໃຊ້ແທນກັນໄດ້ເກືອບທຸກກຣົມ

```
int b[5];
int *bPtr;
bPtr = b; //equivalent to bPtr = &b[0];
```

&b[3] equivalent to **bPtr+3**
b[3] equivalent to ***(bPtr+3)**

9.5.1 ການນິ້ນໍ້າ ໃຫ້ array ປົກຕີ

01006012 Computer Programming

```
int main()
{
    int i, offset, b[] = {10, 20, 30, 40};
    int *bPtr = b; /* set bPtr to point to array b */

    printf("Array b printed with:\n"
           "Array subscript notation\n");
    for (i=0; i<=3; i++)
        printf("b[%d] = %d\n", i, b[i]);
    return 0;
}
```

Array b printed with:

Array subscript notation

b[0] = 10

b[1] = 20

b[2] = 30

b[3] = 40

9.5.2 กรณีที่ 2 ใช้ array เวิญแบบ pointer

01006012 Computer Programming

```
int main()
{
    int i, offset, b[] = {10, 20, 30, 40};
    int *bPtr = b; /* set bPtr to point to array b */

    printf("Pointer/offset notation where\n"
           "the pointer is the array name\n");
    for (offset = 0; offset<=3 ; offset++)
        printf("* (b + %d) = %d\n", offset, *(b + offset));
    return 0;
}
```

Pointer/offset notation where
The pointer is the array name
 $\ast(b + 0) = 10$
 $\ast(b + 1) = 20$
 $\ast(b + 2) = 30$
 $\ast(b + 3) = 40$

9.5.3 กรณีที่ 3 ใช้ pointer เขียนแบบ array

01006012 Computer Programming

```
int main()
{
    int i, offset, b[] = {10, 20, 30, 40};
    int *bPtr = b; /* set bPtr to point to array b */

    printf("Pointer subscript notation\n");
    for (i=0 ; i<=3 ; i++)
        printf("bPtr[%d] = %d\n", i, bPtr[i]);
    return 0;
}
```

Pointer subscript notation

bPtr[0] = 10
bPtr[1] = 20
bPtr[2] = 30
bPtr[3] = 40

9.5.4 กรณีที่ 4 ใช้ pointer ปกติ

01006012 Computer Programming

```
int main()
{
    int i, offset, b[] = {10, 20, 30, 40};
    int *bPtr = b; /* set bPtr to point to array b */

    printf("Pointer/offset notation\n");
    for (offset=0; offset<=3; offset++)
        printf("* (bPtr + %d) = %d\n", offset, *(bPtr + offset));
    return 0;
}
```

Pointer/offset notation

```
* (bPtr + 0) = 10
* (bPtr + 1) = 20
* (bPtr + 2) = 30
* (bPtr + 3) = 40
```

9.6 ตัวชี้กับตัวแปรโครงสร้าง

01006012 Computer Programming

เราสามารถใช้งานสมาชิกของตัวแปรแบบโครงสร้างได้อยู่สองวิธี

- ใช้เครื่องหมายจุด (**struc_var.struc_member**) (structure member operator, or dot operator)
- ใช้เครื่องหมาย -> เมื่อเป็นตัวชี้ไปยังตัวแปรแบบโครงสร้าง เราจะใช้ **structure pointer operator** (**struc_Ptr->struc_member**)

9.6 ตัวชี้กับตัวแปรของสร้าง

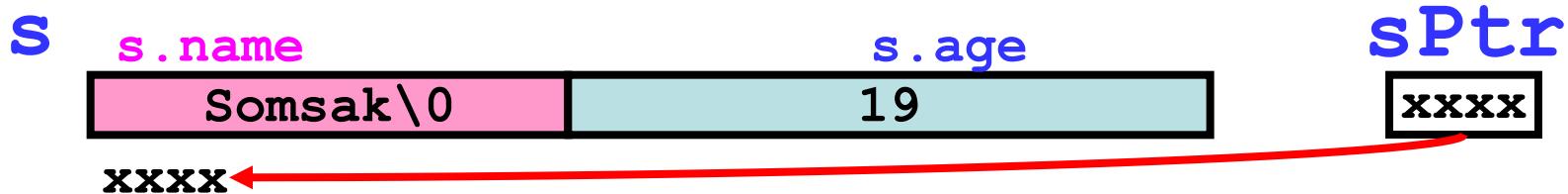
01006012 Computer Programming

```

struct student{
    char name[40];
    int age;
};

struct student s;
struct student *sPtr;
strcpy(s.name,"Somsak");
s.age = 19;
sPtr = &s;

```



หากต้องการพิมพ์ว่าชื่อนักศึกษา สามารถทำได้สองวิธีคือ

- **printf("%s", s.name);**
- **printf("%s", sPtr->name);**

9.6 ตัวชี้กับตัวแปรของสร้าง

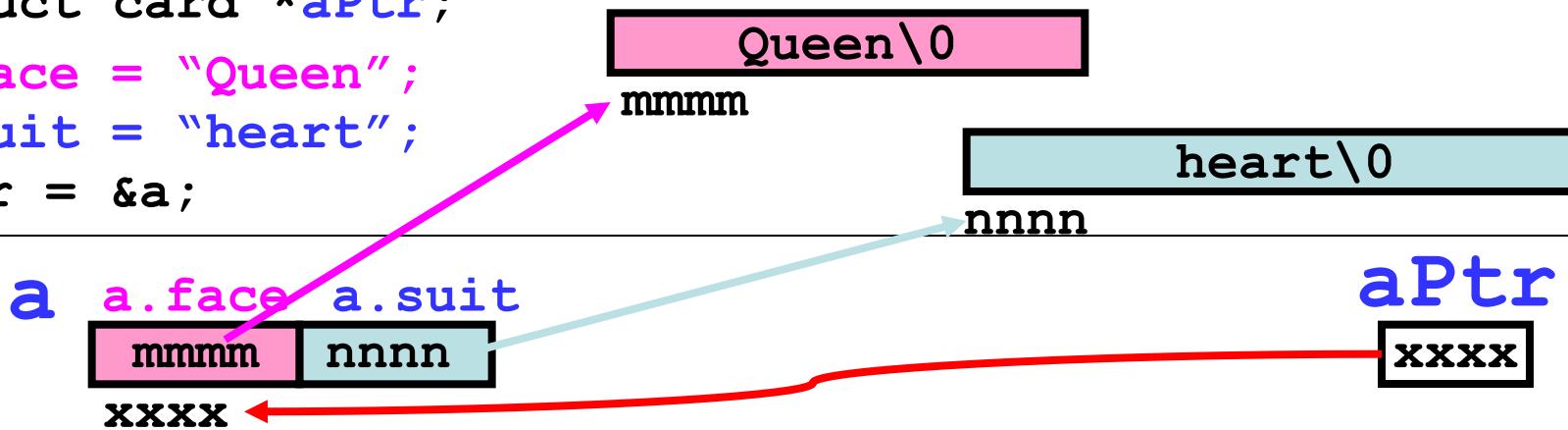
01006012 Computer Programming

```

struct card {
    char *face;
    char *suit;
};

struct card a;
struct card *aPtr;
a.face = "Queen";
a.suit = "heart";
aPtr = &a;

```



หากต้องการพิมพ์ว่าไฟฟ้าในนี่อยู่ในชุดไหน สามารถทำได้สองวิธีคือ

- `printf("%s", a.suit);`
- `printf("%s", aPtr->suit);`

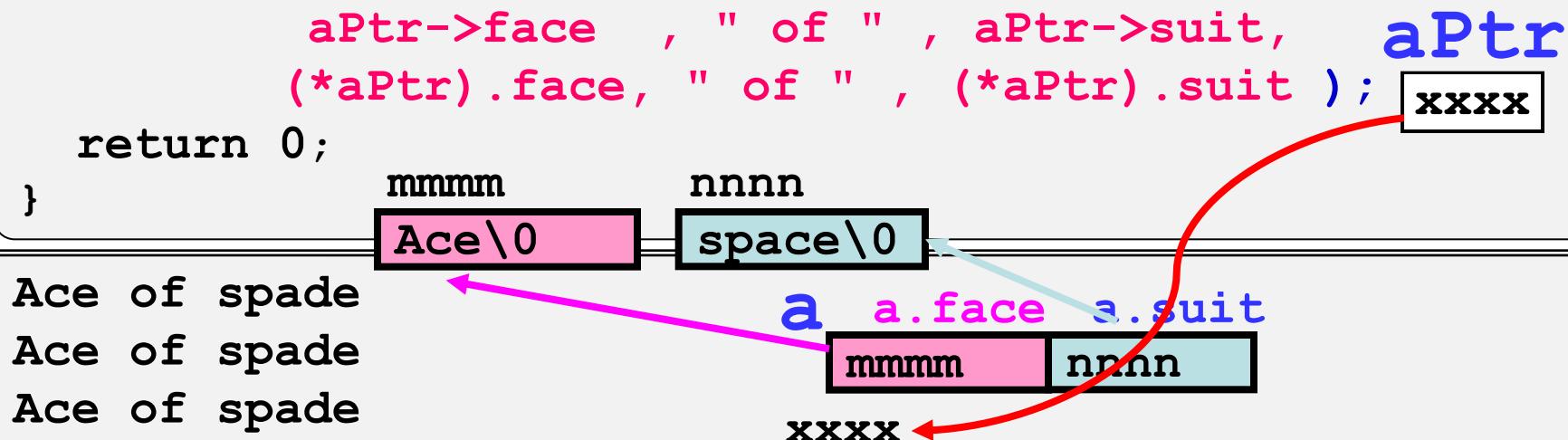
9.6 ตัวชี้กับตัวแปรคงสร้าง ตัวอย่าง

01006012 Computer Programming

```

int main(){
    struct card {
        char *face; // 2,3,...,9,J,Q,K,A
        char *suit; // space, heart, diamond, club
    };
    struct card a; struct card *aPtr;
    a.face = "Ace"; a.suit = "spade"; aPtr = &a;
    printf( "%s%s%s\n%s%s%s\n%s%s%s\n",
            a.face , " of " ,a.suit,
            aPtr->face , " of " , aPtr->suit,
            (*aPtr).face, " of " , (*aPtr).suit );
    return 0;
}

```



โปรแกรม 9.6 โปรแกรมเก็บข้อมูลนศ. 10 คน



01006012 Computer Programming

จะเขียนโปรแกรมเก็บข้อมูลนักศึกษาจำนวน 10 คน โดยมีรายละเอียดดังนี้

- ข้อมูลประกอบด้วย ชื่อ กับ อายุ
- รับข้อมูลนักศึกษาจากคีย์บอร์ด
- โปรแกรมจะต้องใช้ตัวชี้ที่ชี้ไปยังข้อมูลประเภทโครงสร้าง
- เมื่อป้อนข้อมูลเสร็จ โปรแกรมจะค้นหานักเรียนที่อายุมากกว่า 20 ปี
แสดงชื่อ ออกของภาพ

ໂປຣແກຣມ 9.6 ໂປຣແກຣມເກີບຂໍ້ມູນຄ. 10 ຄນ



01006012 Computer Programming

Student[0]	
name : joy	
age : 12	
Student[1]	
name : boy	
age : 20	
Student[2]	
name : jo	
age : 23	
Student[3]	
name : pat	
age : 21	
Student[4]	
name : ple	
age : 13	
Student[5]	
name : tom	
age : 11	

Student[6]	
name : tu	
age : 25	
Student[7]	
name : tee	
age : 34	
Student[8]	
name : bat	
age : 44	
Student[9]	
name : phon	
age : 33	
	jo, 23
	pat, 21
	tu, 25
	tee, 34
	bat, 44
	phon, 33

โปรแกรม 9.6 โปรแกรมเก็บข้อมูลนศ. 10 คน



01006012 Computer Programming

```
#include<stdio.h>
#include<conio.h>
int main()
{
    struct profile{
        char name[20];
        int age;
    } s[10];

    int i;
    struct profile *sPtr;

    sPtr = s;
```

ໂປຣແກຣມ 9.6 ໂປຣແກຣມເກີບຂໍ້ມູນຄ. 10 ຄນ



01006012 Computer Programming

```
for (i=0; i<10; i++)
{
    printf("Student # %d\n\tName : ", i+1 );
    scanf("%s",sPtr->name);
    printf("\tAge:");
    scanf("%d", &(sPtr->age));
    sPtr++;
}
sPtr == 10;
for (i=0; i<10; i++)
{
    if ((sPtr->age) > 20)
        printf("\n%s, %d", sPtr->name, sPtr->age);
    sPtr++;
}
return 0;
}
```

9.7 ตัวชี้และตัวแปรข้อความ (Strings)

01006012 Computer Programming

- **ข้อความ (String)** ในภาษาซี คือ อาร์เรย์ของตัวอักษรที่ลงท้ายด้วยรหัส null character ซึ่งคือเลข 0 รหัสฐานสอง 8 ตัว ($0000\ 0000$)₂ และใช้เนื้อที่ 1 byte
- รหัสตัวอักษรนั้น Nul Character เขียนแทนด้วย escape sequence ‘\0’
- ตัวอักษรเลขศูนย์ เขียนแทนด้วย ‘0’ แทนด้วยรหัสฐานสอง ($0011\ 0000$)₂
- **ตัวแปรตัวชี้ (Pointer)** ที่ไม่ได้ชี้ไปที่ตำแหน่งใดๆ สามารถกำหนดได้ด้วยค่านั้น **NULL** ซึ่งค่า **NULL** ได้กำหนด (#defined) อยู่ใน header file ของคอมไพล์เตอร์ภาษาซี (NULL เป็นชื่อของ macro)
- ค่าที่เก็บจริงๆ ในตัวแปร **Null Pointer** จะจะมีค่าเท่าหรือไม่เท่ากับ 0 ก็ได้ ทั้งนี้ขึ้นอยู่กับระบบหรือสภาพแวดล้อมที่พัฒนาโปรแกรม

9.7 ตัวชี้และตัวแปรข้อความ (Strings)

01006012 Computer Programming

ตัวอย่างการเก็บข้อความชื่อ **Ted** ในแบบต่างๆ ลงท้ายด้วย **nul character**

```
char myString[40];
```

```
myString[0] = 'T' ;
```

```
myString[1] = 'e' ;
```

```
myString[2] = 'd' ;
```

```
myString[3] = '\0' ;
```

```
char myString[40] = { 'T', 'e', 'd', '\0' };
```

```
char myString[40] = "Ted";
```

ตัวอย่างการเช็คค่าตัวแปรตัวชี้ให้มีค่าเป็น **NULL** หรือนำไปใช้เปรียบเทียบ

```
ptr = NULL;
```

```
if (ptr == NULL) ...
```

9.7 ตัวชี้และตัวแปรข้อความ (Strings)



01006012 Computer Programming

ตัวอย่างโปรแกรมสำเนาข้อความโดยการใช้ตัวชี้

```
#include <stdio.h>
char strA[80] =
    "A string to be used for demonstration purposes";
char strB[80];

int main(void)
{
    char *pA;
    char *pB;
    puts(strA);
    pA = strA;
    puts(pA);
    pB = strB;
    putchar('\n');
```

9.7 ตัวชี้และตัวแปรข้อความ (Strings)



01006012 Computer Programming

```
while (*pA != '\0')
{
    *pB++ = *pA++;
}
*pB = '\0';
puts(strB);

return 0;
}
```

9.8 Function การส่งค่าแบบ Pass by Reference

01006012 Computer Programming

- เป็นการส่งค่า Address ของตัวแปรเข้าสู่ฟังก์ชัน
- การเปลี่ยนแปลงค่าต่างๆ ของพารามิเตอร์จะส่งผลไปยังตัวแปรในโปรแกรมหลัก
- ลักษณะของ Pass by Reference Function จะส่งค่า Address (หน้าตัวแปรจะมีเครื่องหมาย * หน้าตัวแปรเสมอ)
- ตัวอย่าง
 - int max(int *a, int *b)
 - void increase(int *count)

Pass by Reference Function

01006012 Computer Programming

```
void func2(int *pa)
{
    *pa = *pa +1;
    printf ("In function *pa = %d\n", *pa);

}

int main()
{
    int x;
    x = 10;
    printf ("Before call function x = %d\n", x);
    func2 (&x) ;
    printf ("After call function x = %d\n", x);
    return 0;
}
```

Output

01006012 Computer Programming

```
Before call function x = 10
In function *pa = 11
After call function x = 11
```

|

สรุปการเปลี่ยนแปลงค่าตัวแปรในฟังก์ชันหลัก



01006012 Computer Programming

การทำงานในฟังก์ชัน สามารถเปลี่ยนแปลงค่าตัวแปรในฟังก์ชัน
หลัก ได้สองกรณี

1. ตัวแปรที่ส่งค่าเข้าสู่ฟังก์ชัน เป็นตัวแปรแบบ Global Variable
2. การฟังก์ชันทำงานแบบ Pass by Reference

ตัวอย่าง พังก์ชันกับอะเรย์

01006012 Computer Programming

```
#include <stdio.h>

int findmax( int * );
int main()
{
    int i, a[10];
    printf ("Input 10 number ");
    for ( i=0; i<10; i++ )
        scanf ( "%d", &a[i] );
    printf ( “ \nmax number = %d “ ,
            findmax( a ) );
    return 0;
}
```

```
int findmax( int *p )
{
    int i, temp;
    for (i=0;i<9;i++) {
        if ( *(p+1) < *p ) {
            temp = *p;
            *p = *(p+1);
            *(p+1) = temp;
        }
        p++;
    }
    return *p;
}
```

Q&A

01006012 Computer Programming

```
#include <stdio.h>
#include <conio.h>
int a;
void func(int x)
{
    x=10;
    printf("x = %d\n",x);
}

int main()
{
    int b;
    a = 3;
    b = 5;
    func(b);
    printf ("a = %d\n",a);
    printf ("b = %d\n",b);
    return 0;
}
```

OUTPUT

```
x = 10
a = 3
b = 5
```

Q&A

01006012 Computer Programming

```
#include <stdio.h>
#include <conio.h>
int a;
void func(int *x)
{
    *x=10;
    printf("x = %d\n",*x);
}

int main()
{
    int b;
    a = 3;
    b = 5;
    func(&b);
    printf ("a = %d\n",a);
    printf ("b = %d\n",b);
    return 0;
}
```

OUTPUT

x = 10
a = 3
b = 10

Q&A

01006012 Computer Programming

```
#include <stdio.h>
#include <conio.h>
int a;
void func(int x)
{
    a=10;
    printf("x = %d\n",x);
}

int main()
{
    int b;
    a = 3;
    b = 5;
    func(b);
    printf ("a = %d\n",a);
    printf ("b = %d\n",b);
    return 0;
}
```

OUTPUT

x = 5
a = 10
b = 5

9.8 คำสั่งที่อยู่บุฟเฟต์

01006012 Computer Programming

1. จงสร้างอาร์เรย์ A เป็นอาร์เรย์ขนาด 3×3 เก็บข้อมูลตัวเลข 1-9 และแสดงผลลัพธ์ตัวเลขในอาร์เรย์โดยใช้การอ้างอิงค่าตัวเลขโดยใช้ Pointer เท่านั้น
2. x และ y เป็นตัวแปรแบบทศนิยม จงเขียนโปรแกรม ทำการสลับค่า x และ y โดยใช้ pointer เท่านั้น
3. ให้ S เป็น Structure ที่เก็บข้อมูลตัวเลข 3 ตัว จงเขียนโปรแกรมให้เก็บข้อมูลตัวเลขตัวที่ 1 และ 2 ลงใน Structure และหาผลรวมลงในตัวเลขตัวที่ 3 โดยใช้การอ้างอิงข้อมูลโดย Pointer เท่านั้น