

บทที่เจ็ด

พอยน์เตอร์ (Pointer)

พอยน์เตอร์ หมายถึง ตัวแปรชนิดหนึ่งที่ทำหน้าที่เก็บตำแหน่งที่อยู่ (address) ของตัวแปรที่อยู่ในหน่วยความจำ

7.1 การประกาศชนิดของตัวแปรพอยน์เตอร์

ตัวแปรที่จะทำหน้าที่เป็นตัวแปรพอยน์เตอร์ได้จะต้องมีการประกาศไว้ก่อนตอนต้นโปรแกรมโดย

รูปแบบ

```
type * Variable - name
```

type หมายถึง ชนิดของตัวแปร

* เป็นเครื่องหมายที่แสดงว่า ตัวแปรที่ตามหลังเครื่องหมายนี้เป็นตัวแปรชนิดพอยน์เตอร์

variable - name เป็นชื่อตัวแปรที่ต้องการประกาศว่าเป็นชนิดพอยน์เตอร์โดยมีกฎการตั้งชื่อ

เหมือนกับการตั้งชื่อตัวแปรธรรมดา

ตัวอย่างที่ 1

```
char *prt;
```

```
int *temp, *str;
```

หมายความว่า ประกาศว่าตัวแปร prt เป็นตัวแปรพอยน์เตอร์ชนิด character และตัวแปร temp ตัวแปร str เป็นตัวแปรพอยน์เตอร์ชนิด integer

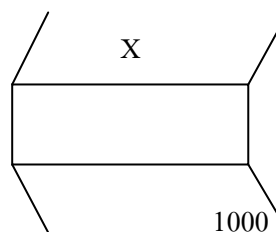
7.2 เครื่องหมายที่ใช้ในการทำงานกับตัวแปรพอยน์เตอร์

7.2.1 เครื่องหมาย &

เป็นเครื่องหมายที่ใช้เมื่อต้องการให้เอาค่าตำแหน่งที่อยู่ (address) ของตัวแปรที่เก็บไว้ใน

หน่วยความจำออกมาใช้เช่น

```
x = 10;
```



ตำแหน่งที่อยู่

หมายความว่า กำหนดให้ตัวแปร x ซึ่งอยู่ที่ตำแหน่ง 1000 มีค่า 10 เก็บอยู่

ตัวอย่างที่ 2

```
int *y, x = 10;
```

```
y = &x;
```

หมายความว่า ตัวแปร y ซึ่งประกาศเป็นตัวแปรพอยน์เตอร์ จะเก็บค่า 1000 ซึ่งเป็นตำแหน่งที่อยู่ของตัวแปร x เอาไว้

ตัวแปรที่มีเครื่องหมาย & นำหน้าจะไม่สามารถนำมาทำการคำนวณได้ เช่น

`&x = &x + 1;` ใช้ไม่ได้

7.2.2 เครื่องหมาย *

เป็นเครื่องหมายที่จะใช้เมื่อต้องการให้นำค่าที่อยู่ในตำแหน่งที่ตัวแปรพอยน์เตอร์นั้นชี้อยู่ออกมา
แสดงให้คุณ

ตัวอย่างที่ 3

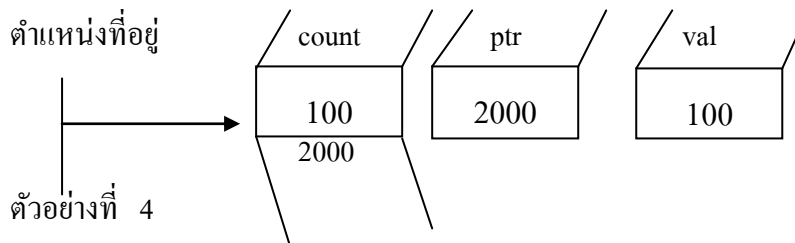
```
int *ptr, count, val;
```

```
count = 100;
```

```
ptr = &count;
```

```
val = *ptr;
```

ลักษณะการเก็บข้อมูลในหน่วยความจำจะเป็นดังนี้



```
int a, *prt, b, c, *d;
```

```
a = 25;
```

```
prt = &a;
```

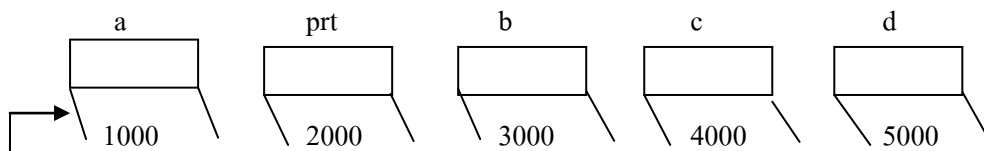
```
b = a;
```

```
c = *prt;
```

```
d = prt;
```

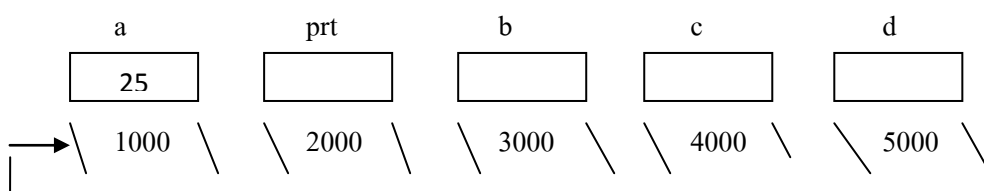
ลักษณะที่อยู่ภายในหน่วยความจำ

คำสั่งที่ 1 เครื่องจะเตรียมเนื้อหาที่ให้กับตัวแปรภายในหน่วยความจำตามตำแหน่งต่าง ๆ ดังนี้



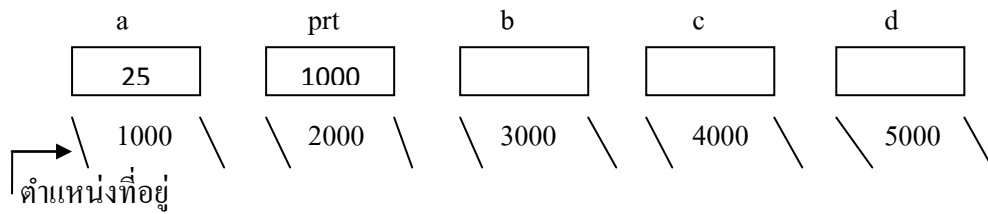
ตำแหน่งที่อยู่

คำสั่งที่ 2 เครื่องจะกำหนดค่าให้กับตัวแปร a โดยมีค่า 25 เก็บอยู่ ดังนี้

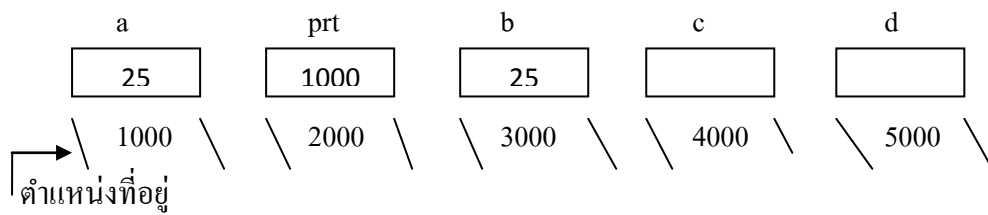


ตำแหน่งที่อยู่

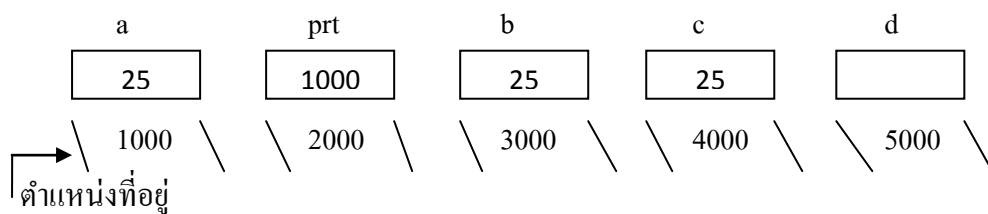
คำสั่งที่ 3 เครื่องจะเอาตำแหน่งที่อยู่ (address) ของตัวแปร a ไปเก็บไว้ในตัวแปรพอยน์เตอร์ ptr



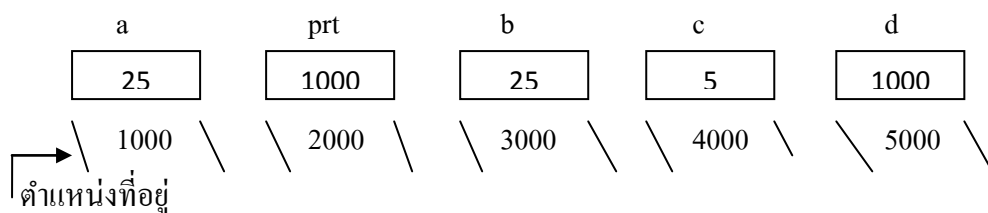
คำสั่งที่ 4 เครื่องจะเอาค่าในตัวแปร a ไปเก็บในตัว b ดังนี้



คำสั่งที่ 5 เครื่องจะเอาค่าที่อยู่ในตำแหน่งที่ ptr ชี้ไปยังเก็บไว้ในตัวแปร c ดังนี้



คำสั่งที่ 6 เครื่องจะย้ายค่าที่เก็บในตัวแปรพอยน์เตอร์ ptr ไปเก็บไว้ในตัวแปรพอยน์เตอร์ d ดังนี้



ตัวอย่างที่ 5 จงเขียนโปรแกรมแสดงการพิมพ์ค่าตำแหน่งที่อยู่ของตัวแปร c[0], i[0], f[0] ซึ่งเก็บค่า a, 0 และ 0. ตามลำดับเอาไว้แล้วพิมพ์ค่าตำแหน่งที่อยู่อีกตำแหน่งละ 1 ค่า พร้อมทั้งพิมพ์ค่าที่อยู่ในตำแหน่งนั้นๆออกมาด้วย

```
main ()
char *cp
int *ip;
float *fp;
int i[5] = {0,1,2,3,4};
char c[5] = {"abcd"};
float f[5] = {0.,1.,2.,3.,4.};
cp = &c[0];
```

```

ip = &i[0];
fp = &f[0];
printf ("%u %u %u \n", cp, ip, fp);
printf ("%c %d %f \n", *cp, *ip, *fp);

cp ++;
ip ++;
fp ++;

printf ("%u %u %u \n", cp, ip, fp);
printf ("%c %d %f \n", *cp, *ip, *fp);

cp ++;
ip ++;
fp ++;

printf ("%u %u %u \n", cp, ip, fp);
printf ("%c %d %f \n", *cp, *ip, *fp);
}

```

```

65478      65468      65484
a          0          0.000000
65479      65470      65488
b = 1.000000
65480      65472      65492
c          2          2.000000

```

ข้อสังเกต ตัวแปรพอยน์เตอร์ที่เป็นชนิด character เมื่อเพิ่มค่าครั้งละ 1 จะเปลี่ยนตำแหน่งที่อยู่ไปครั้งละ 1 ไบท์

ตัวแปรพอยน์เตอร์ชนิด integer เมื่อเพิ่มค่าครั้งละ 1 จะเปลี่ยนตำแหน่งที่อยู่ไปครั้งละ 2 ไบท์

ตัวแปรพอยน์เตอร์ชนิด float เมื่อเพิ่มค่าครั้งละ 1 จะเปลี่ยนตำแหน่งที่อยู่ไปครั้งละ 4 ไบท์

7.8 การใช้ตัวแปรพอยน์เตอร์กับตัวแปรชุด

ตัวแปรพอยน์เตอร์จะใช้อ้างถึงค่าที่เก็บไว้ในตัวแปรชุดได้ดังนี้

ตัวอย่างที่ 6

```

char str[80], *pl;
pl = str;

```

คำสั่งที่ 1 จะเป็นการประกาศว่า str เป็นตัวแปรชนิด character 1 มิติ ขนาด 80 สมาชิก และ p1 เป็นพอยน์เตอร์

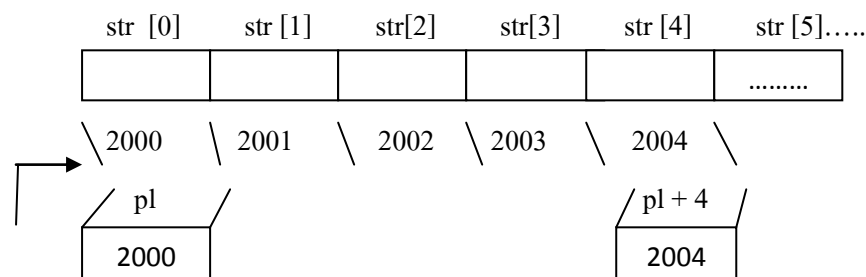
คำสั่งที่ 2 เป็นการอ้างถึงข้อมูลที่เก็บไว้ในตัวแปร str โดยการนำตำแหน่งที่อยู่ของตัวแปร str [0] ซึ่งเป็นตัวแปรตัวแรกในชุดไปเก็บไว้ในตัวแปรพอยน์เตอร์ p1 เหมือนกับการเขียนคำสั่ง

```
p1 = & str [0];
```

ดังนั้นการอ้างถึงตัวแปรชุด จึงสามารถอ้างถึงได้โดยการเพิ่มหรือลดตัวแปรพอยน์เตอร์ได้เช่น ถ้าต้องการอ้างถึงตัวแปร str [4] โดยการอ้างถึงตำแหน่งที่อยู่ ก็สามารถทำได้โดยการเพิ่มตัวแปรพอยน์เตอร์ดังนี้

```
* (p1 + 4)
```

เมื่อ p1 เป็นตัวแปรพอยน์เตอร์ที่ชี้ตำแหน่งที่อยู่ของตัวแปรชุด str ในหน่วยความจำดังนี้



ตัวอย่างที่ 7 แสดงการอ้างถึงข้อมูลที่อยู่ในตัวแปร ซึ่งทำได้หลายวิธี เช่น อ้างถึงชื่อตัวแปรนั้นเลยหรือใช้ตัวแปรพอยน์เตอร์แสดงตำแหน่งของตัวแปรนั้นก็ได้

```
main ()  
{  
    int a[5] = {1,2,3,4};  
    int *ptr;  
    ptr = a;  
    printf ("%d \n", a[0]);  
    printf ("%d \n", *ptr);  
    printf ("%d \n", *(ptr+1));  
    printf ("%d \n", *(a+2));  
    printf ("%d \n, ptr [3]");  
}
```

ผลที่ได้จากโปรแกรม

0

0

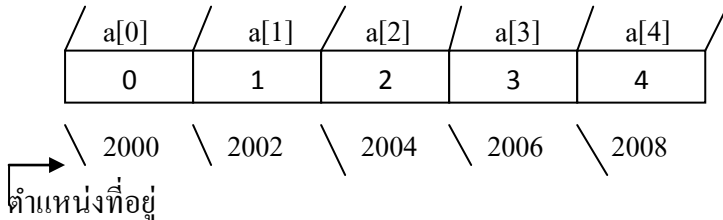
1

2

3

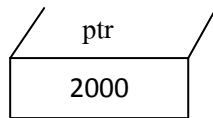
ลักษณะการทำงานของแต่ละคำสั่ง

คำสั่งที่ 1 เป็นการกำหนดค่าให้กับตัวแปรชุด a ที่สมมุติว่าอยู่ในหน่วยความจำที่ตำแหน่ง 2000 – 2008 (ตัวแปรชนิด int จะใช้ที่ 2 byte ต่อ 1 ชื่อตัวแปร)



คำสั่งที่ 2 เป็นการประกาศให้ตัวแปร ptr เป็นตัวแปรพอยน์เตอร์

คำสั่งที่ 3 เป็นการเอาค่าตำแหน่งที่อยู่ของตัวแปร a[0] ไปเก็บในตัวแปรพอยน์เตอร์ ptr



คำสั่งที่ 4 เป็นคำสั่งให้พิมพ์ข้อมูลที่อยู่ในตัวแปร a[0] ออกมาทำให้ได้ผลลัพธ์เป็น 0 ในบรรทัดแรก

คำสั่งที่ 5 เป็นคำสั่งให้พิมพ์ข้อมูลที่อยู่ในตำแหน่งที่ตัวแปร ptr ซึ่งก็คือค่าที่อยู่ในตำแหน่ง 2000

ออกมา ซึ่งก็เป็นตำแหน่งเดียวกับตัวแปร a[0] ผลลัพธ์จึงได้เป็น 0 ในบรรทัดที่สองนั่นเอง

คำสั่งที่ 6 แสดงการเพิ่มค่าของตัวแปรพอยน์เตอร์ ซึ่งเป็นตัวชี้ตำแหน่งที่ต้องการนำข้อมูลที่อยู่ในตำแหน่งนั้นออกมา นั่นคือตัวแปร ptr ซึ่งที่ตำแหน่ง 2000 ซึ่งเป็นตำแหน่งของตัวแปร a[0] เมื่อเพิ่มค่าขึ้นอีก 1 หมายความว่าตำแหน่งที่อยู่ของข้อมูลจะเปลี่ยนไปเป็น 2002 จึงทำให้ผลลัพธ์ได้ค่า 1 ที่เก็บอยู่ใน a[0] ออกมา

คำสั่งที่ 7 เป็นการอ้างถึงตำแหน่งที่อยู่ของข้อมูล โดยใช้ชื่อตัวแปรชุด a โดยการเพิ่มค่าของตำแหน่งที่อยู่ไปยังตำแหน่งที่ต้องการได้ เช่น a+2 จะหมายถึงตำแหน่งที่อยู่ที่ 2004 ดังนั้นเมื่อต้องการข้อมูลที่อยู่ภายในตำแหน่งที่ 2004 จึงเขียนได้ดังนี้ *(a+2) ทำให้ผลที่พิมพ์ออกมาคือ 2 ซึ่งเป็นค่าที่เก็บในตำแหน่ง 2004 ที่เดียวกับตัวแปร a[2]

คำสั่งที่ 8 เป็นการอ้างถึงตำแหน่งที่อยู่ของข้อมูล โดยใช้ตัวแปรพอยน์เตอร์ ptr ทำหน้าที่คล้ายกับตัวแปรชุด a โดยถือว่าตัวแปร ptr เริ่มต้นที่ตำแหน่งเดียวกับตัวแปรชุด a ดังนั้น ptr [3] ก็หมายถึงตำแหน่งของ a[3]

7.4 การใช้ตัวแปรพอยน์เตอร์กับข้อมูลที่เป็นข้อความ (string)

โดยปกติการเก็บข้อมูลที่เป็นข้อความจะใช้ตัวแปรชุดในการเก็บโดยกำหนดให้ตัวแปรชุด 1 ตัวเก็บข้อมูล 1 ตัวอักษร ดังนั้นจะต้องมีการเตรียมที่ของตัวแปรชุดไว้เท่ากับจำนวนตัวอักษรที่ประกอบเป็นข้อความนั้นบวกหนึ่ง สำหรับเครื่องหมาย \0 ซึ่งแสดงการเสร็จสิ้นสุดของข้อความ เพราะเรา

สามารถใช้ตัวแปรพอยน์เตอร์มาระบุถึงตำแหน่งเริ่มต้นของข้อมูลชนิดข้อความที่อยู่ในหน่วยความจำได้ ดังนี้

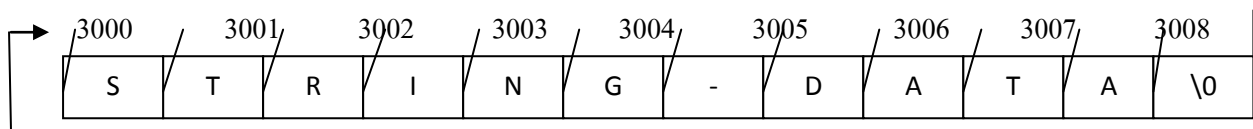
ตัวอย่างที่ 8

```
main ()
{
    char *s;
    s = "STRING - DATA";
    printf (s);
}
```

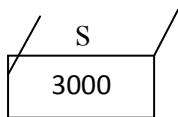
ผลลัพธ์

STRING - DATA

ลักษณะในหน่วยความจำ



ตำแหน่งที่อยู่



ตัวอย่างที่ 9 แสดงการรับข้อมูลชนิดข้อความจากแป้นพิมพ์มาเก็บไว้ในหน่วยความจำโดยใช้ตัวแปรพอยน์เตอร์เป็นตัวแปรระบุตำแหน่งของข้อมูลตัวแรก แล้วนำค่าออกมาพิมพ์จนถึง \0 จึงหยุดการพิมพ์

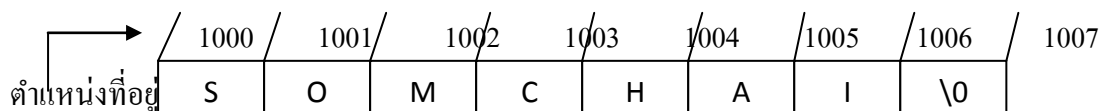
```
main ()
{
    char * a
    printf ("PRINT YOUR NAME:");
}
```

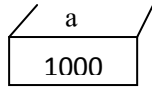
ผลลัพธ์

PRINT YOUR NAME : SOMCHAI

SOMCHAI

ลักษณะในหน่วยความจำ





7.5 การกำหนดตัวแปรชนิดพอยน์เตอร์

ในกรณีที่ต้องใช้ตัวแปรพอยน์เตอร์หลาย ๆ ตัว สามารถกำหนดตัวแปรพอยน์เตอร์เหล่านั้นได้ โดยประกาศให้ตัวแปรชนิดที่ต้องการนั้นมีลักษณะตัวแปรพอยน์เตอร์ เช่น

ตัวอย่างที่ 10

```
int *x[10];
```

หมายความว่า ประกาศให้ตัวแปรชนิด x เป็นตัวแปรพอยน์เตอร์ชนิด int จำนวน 10 ตัวแปร คือ x[0], x[1], ..., x[9] จะเป็นตัวแปรพอยน์เตอร์ทั้งหมด

ตัวอย่างที่ 11

```
int *x[10];
```

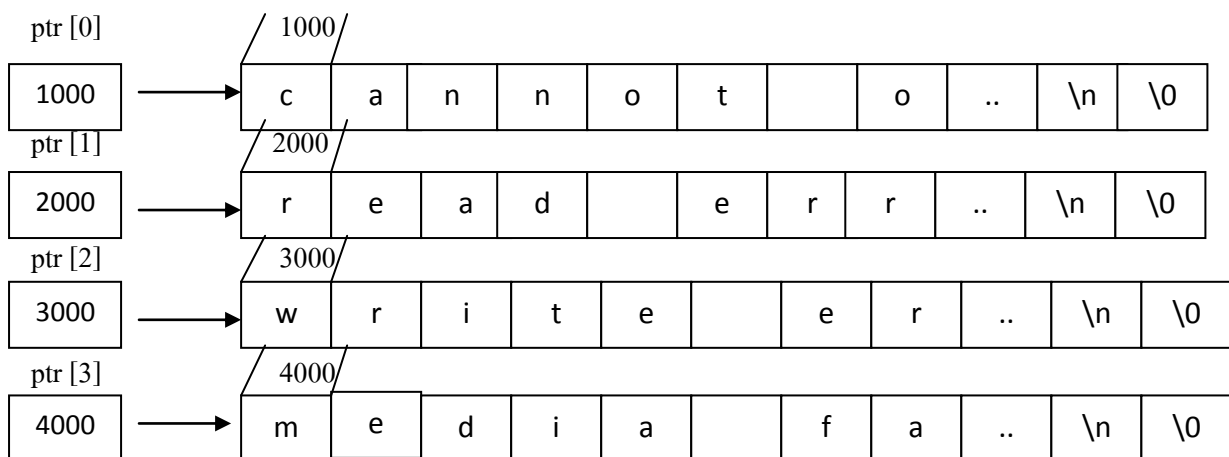
```
x[2] = &rate;
```

หมายความว่า ตัวแปรพอยน์เตอร์ x[2] จะทำหน้าที่เก็บตำแหน่งที่อยู่ตัวแปร rate

ตัวอย่างที่ 12

```
char *ptr[4] = {
    "cannot open file\n"
    "read error \n"
    "write error \n"
    "media failure \n"
};
```

ลักษณะที่เก็บในหน่วยความจำ



ดังนั้นถ้าใช้คำสั่ง

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


จะได้ผลลัพธ์เป็น

read error

ตัวอย่างที่ 13 แสดงการใช้ตัวแปรพอยน์เตอร์ทำหน้าที่ชี้ตำแหน่งของข้อมูลที่ได้รับเข้ามาทางแป้นพิมพ์

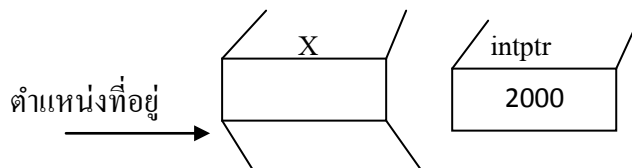
```
main () {  
    int x, *intptr;  
    intptr = & x;  
    printf ("enter an integer:");  
    scanf ("%d", intptr);  
    printf ("the value enter is %d \n", x);  
}
```

ผลลัพธ์

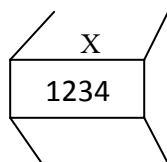
enter an integer : 1 2 3 4

the value entered is 1 2 3 4

ลักษณะในหน่วยความจำ



คำสั่ง scanf จะทำให้รับค่า 1 2 3 4 จากแป้นพิมพ์เข้ามาเก็บไว้ยังหน่วยความจำที่ตำแหน่ง 2000 ซึ่งเป็นตำแหน่งของตัวแปร x ดังนี้



ดังนั้นเมื่อใช้คำสั่ง printf เพื่อพิมพ์ค่า x จึงได้ค่า 1 2 3 4 ออกมา

ตัวอย่างที่ 14 แสดงความสัมพันธ์ระหว่างตัวแปร กับตัวแปรพอยน์เตอร์

```
main () {  
    int x;  
    int *pointer;  
    pointer = &x;  
    x = 15;  
    printf ("The value of X = %d \n", x);  
    printf ("The value of *pointer = %d \n"), *pointer);  
    *pointer = 32;
```

```
printf("x = %d \n", x);
printf("*pointer = %d \n", *pointer);
}
```

ผลลัพธ์

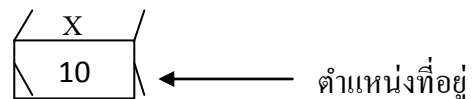
The value of x = 15
 The value of *pointer = 15
 x = 32
 *pointer = 32

7.6 ตัวแปรพอยน์เตอร์ที่ทำหน้าที่เป็นพอยน์เตอร์

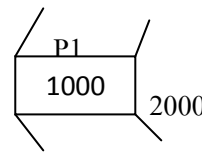
เป็นวิธีการกำหนดตัวแปรพอยน์เตอร์ให้ทำหน้าที่เก็บตำแหน่งที่อยู่ของตัวแปรพอยน์เตอร์อีกตัวหนึ่ง (บางที่เรียกตัวแปรพอยน์เตอร์นี้ว่า indirect pointer)

ลักษณะในหน่วยความจำ

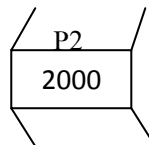
```
x = 10;
X
```



```
p1 = &x;
```



```
p2 = &p1
```



p2 เป็นตัวแปรพอยน์เตอร์ที่เรียกว่า Indirect Pointer

7.6.1 การประกาศตัวแปรให้เป็นตัวแปร Indirect Pointer

ตัวแปรที่จะทำหน้าที่เป็นตัวแปร Indirect Pointer จะต้องมีการประกาศหรือบอกให้รู้ไว้ก่อนในตอนต้นโปรแกรมโดยใช้

รูปแบบ

```
type **name - of - pointer
```

name - of - pointer คือตัวแปรที่ต้องการใช้เป็น

ตัวแปร Indirect Pointer

ตัวอย่างที่ 15

```
int i;
```

```
int *ip;
int **pip;
```

หมายความว่าประกาศให้ตัวแปร pip เป็นตัวแปร Indirect Pointer

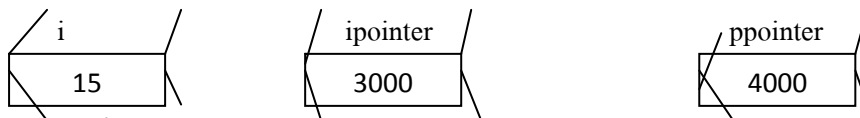
7.6.2 การใช้ตัวแปร Indirect Pointer

จะใช้เหมือนกับลักษณะของตัวแปรพอยน์เตอร์ธรรมดา

ตัวอย่างที่ 16

```
int i;
int *ipointer;
int **ppointer;
i = 15;
ipointer = &i;
ppointer = &ipointer;
```

ลักษณะในหน่วยความจำ



ดังนั้นการอ้างถึงค่า 15 ที่เก็บในตัวแปร i สามารถอ้างได้หลายแบบ คือ i หรือ *ipointer หรือ **ppointer

ตัวอย่างที่ 17

```
main ()
{
    int x, *p, **q;
    x = 10;
    p = &x;
    q = &p;
    printf ("%d", **q); /*print the value of x */
}
```

ผลลัพธ์

10

แบบฝึกหัด

1. จงเขียนส่วนของโปรแกรมให้ m เก็บค่า 20 ให้ p เก็บตำแหน่งที่อยู่ของ m และ q เก็บตำแหน่งที่อยู่ของ p เอาไว้อีกหนึ่งให้แสดงค่า m ค่าที่อยู่ใน p และค่าที่อยู่ใน q ออกมาทางจอภาพ
2. จงเขียนโปรแกรมรับข้อมูลชนิดข้อความ เข้ามาทางแป้นพิมพ์ความยาวไม่เกิน 100 ตัว จากนั้นให้แสดง
 1. จำนวนอักขระที่รับเข้ามา
 2. จำนวนคำในข้อความ
 3. จำนวนคำที่ยาวที่สุดและสั้นที่สุดให้ใช้พอยน์เตอร์ อ้างถึงตำแหน่งข้อมูลในการเขียนโปรแกรม

3. จากส่วนของโปรแกรม

```
main ()
```

```
char s1[0] = "SUMMARY"
```

```
char s2[10] = "position";
```

ให้เขียนโปรแกรมให้สมบูรณ์ เพื่อสลับข้อมูลระหว่างตัวแปรชุด s1 กับ s2