



อาร์เรย์และพอยน์เตอร์ (Array and Pointer)

หน่วยที่ 3

3.4 ตัวแปรพอยน์เตอร์ (Pointer)

ในการประกาศตัวแปรแต่ละครั้ง ไม่ว่าจะเป็นตัวแปรชนิด int float char หรือชนิดใดก็ตาม ตัวแปรเหล่านี้จะถูกเก็บอยู่ในหน่วยความจำของคอมพิวเตอร์ ซึ่งตัวแปรแต่ละตัวก็จะมีตำแหน่งระบุเอาไว้ว่าอยู่ในส่วนของหน่วยความจำ เรียกว่า "แอดเดรส (Address) ของตัวแปร" ซึ่งในการใช้งานตัวแปรโดยทั่วไปนั้นจะกำหนดได้เฉพาะค่าของตัวแปรเท่านั้น แต่ไม่สามารถที่จะกำหนดแอดเดรสของตัวแปรได้ ตัวอย่างเช่น

มีตัวแปรอยู่ 2 ตัวคือ x และ y

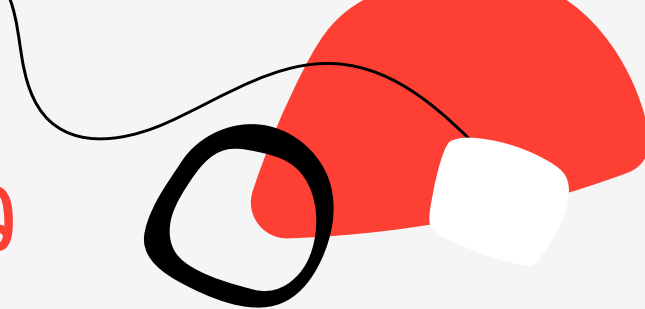
ตัวแปร x มีแอดเดรสอยู่ที่ตำแหน่ง 55900 และค่าของ x คือ 10

ตัวแปร y มีแอดเดรสอยู่ที่ตำแหน่ง 60000 และค่าของ y คือ 5

3.4 ตัวแปรพอยน์เตอร์ (Pointer)

หากต้องการที่จะรู้ว่าตัวแปรแต่ละตัวอยู่ที่ตำแหน่งใดในหน่วยความจำ ให้ใช้เครื่องหมาย & นำหน้าตัวแปร ตัวอย่างเช่น แอดเดรสของตัวแปร x เขียนแทนได้ด้วย &x และแอดเดรสของตัวแปร y เขียนแทนได้ด้วย &y เมื่อต้องการเก็บตำแหน่งในหน่วยความจำไว้ที่ตัวแปร จะมีตัวแปรอยู่ชนิดหนึ่งที่ใช้ในการเก็บค่าแอดเดรสของตัวแปรอื่น ตัวแปรชนิดนั้นคือ "พอยน์เตอร์ (Pointer)"

3.4.1 การประกาศตัวแปรพอยน์เตอร์



การประกาศตัวแปรพอยน์เตอร์จะใช้สัญลักษณ์ * เป็นการบ่งบอกว่าตัวแปรที่ประกาศนั้นเป็นพอยน์เตอร์

ตัวอย่างเช่น `int *numPtr;`

เป็นการประกาศตัวแปร `numPtr` เป็นชนิดพอยน์เตอร์ที่ใช้เก็บแอดเดรสของตัวแปรชนิด `int` ได้

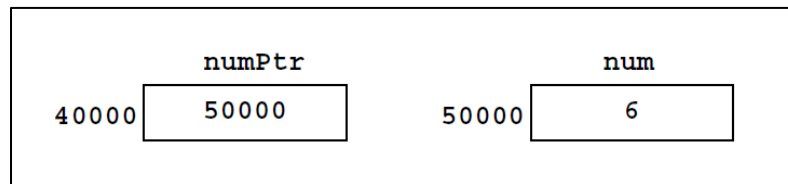


3.4.1 การประกาศตัวแปรพอยน์เตอร์

ตัวแปรพอยน์เตอร์จะใช้ในการเก็บค่าแอดเดรสของตัวแปรตัวอื่น ดังนั้น ในการกำหนดค่าของตัวแปรพอยน์เตอร์ ก็มักจะกำหนดด้วยแอดเดรสของตัวแปร

ตัวอย่างเช่น

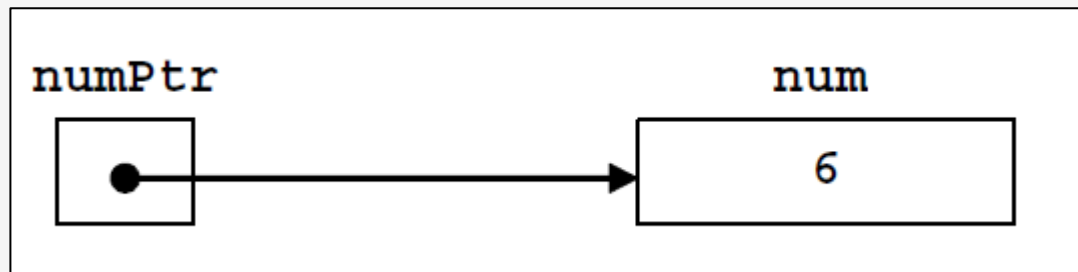
```
int num = 6;  
int *numPtr;  
numPtr = &num;
```



เป็นการกำหนดให้ numPtr มีค่าเท่ากับแอดเดรสของตัวแปร num ถ้าสมมติว่าตัวแปร num ถูกเก็บอยู่ที่ตำแหน่ง 50000 ตัวแปรพอยน์เตอร์ numPtr ก็จะมีค่าเท่ากับ 50000 ด้วย

3.4.1 การประกาศตัวแปรพอยน์เตอร์

ในการเขียนโปรแกรมนั้น ไม่จำเป็นจะต้องรู้ว่าตัวแปร `num` ถูกเก็บอยู่ที่ตำแหน่งไหน เพียงแค่รู้ว่าตัวแปร `numPtr` เป็นพอยน์เตอร์ที่ใช้ในการอ้างอิงไปยังตำแหน่งของตัวแปร `num` ได้ ซึ่งมักจะใช้คำพูดว่า "`numPtr` ชี้ไปยัง `num`" และในการอธิบายก็จะใช้แผนภาพตามภาพ



3.4.1 การประกาศตัวแปรพอยน์เตอร์

ในการอ้างถึงค่าในตำแหน่งที่พอยน์เตอร์ชี้อยู่นั้น ให้ใช้เครื่องหมาย * นำหน้าตัวแปร ซึ่งเรียกว่า การ Dereferencing ตัวอย่างเช่น จากคำสั่งต่อไปนี้

```
int num = 6;      int *numPtr;  
numPtr = &num;    cout << *numPtr;
```

ค่าที่พิมพ์ออกทางหน้าจอคือ 6 เนื่องจากในภาพที่ 3.8 นั้นจะเห็นว่า numPtr ชี้ไปยังตัวแปร num ดังนั้นค่าในตำแหน่งที่ numPtr ชี้อยู่ก็คือค่าของ num นั้นเอง ซึ่งก็คือ 6 หากสังเกตจะเห็นว่าสามารถแทนค่าของ *numPtr ด้วย num ได้เลย

3.4.1 การประกาศตัวแปรพอยน์เตอร์

นอกจากนี้หากต้องการแก้ไขค่าในตำแหน่งที่พอยน์เตอร์ชี้อยู่

ก็สามารถทำได้ในทำนองเดียวกันตัวอย่างเช่น จากคำสั่ง

```
*numPtr = 20;
```

เป็นการกำหนดให้ค่าในตำแหน่งที่ numPtr ชี้อยู่มีค่าเท่ากับ 20 ดังนั้นค่าของตัวแปร num

ก็จะถูกเปลี่ยนเป็น 20 ไปด้วย