

## การทดลองที่ 9

### ตัวชี้ตำแหน่งและตัวชี้ตำแหน่งกับฟังก์ชัน

#### วัตถุประสงค์

1. เข้าใจการทำงานของตัวชี้
2. สามารถใช้งานตัวชี้ได้
3. เข้าใจการใช้งานตัวชี้ในการส่งค่าให้กับฟังก์ชันได้
4. สามารถสร้างฟังก์ชันที่มีการส่งค่าให้กับฟังก์ชันในรูปแบบต่าง ๆ ได้

#### ทฤษฎีโดยย่อ

##### ก. ตัวชี้ตำแหน่ง (Pointer)

ตัวชี้ตำแหน่งหรือตัวแปรแบบ Pointer เป็นตัวแปรที่มีลักษณะพิเศษและเป็นตัวแปรที่เป็นลักษณะเด่นอย่างหนึ่งของภาษาซี การทำงานของตัวแปรชนิดนี้จะมีลักษณะที่พิเศษจากตัวแปรชนิดอื่นในภาษาซีคือตัวแปรชนิดนี้จะทำหน้าที่เป็นจุดอ้างอิงไปยังตัวแปรชนิดอื่นหรือเป็นจุดอ้างอิงไปยังตำแหน่งใด ๆ ในหน่วยความจำ การสร้างตัวแปรชนิดนี้สามารถทำได้ดังนี้

```
type *pointer_name;
```

โดย type คือชนิดของข้อมูลใด ๆ ในภาษาซี และ pointer\_name คือชื่อตัวแปรที่ตั้งขึ้นตามหลักการตั้งชื่อของภาษาซี ทั้งนี้จะสังเกตเห็นว่าหน้าชื่อตัวแปรจะมีเครื่องหมาย \* อยู่ ซึ่งการมีเครื่องหมาย \* อยู่หน้าชื่อนี้คือการบอกภาษาซีว่าตัวแปรนี้เป็นตัวแปรแบบตัวชี้ตำแหน่ง

ในการใช้งานตัวแปรชนิดนี้จะประกอบไปด้วย 2 ขั้นตอน ขั้นตอนแรกจะเป็นการกำหนดให้ตัวชี้ตำแหน่งชี้ไปยังตัวแปรหรือตำแหน่งในหน่วยความจำที่ต้องการใช้งาน และขั้นที่ 2 คือการเรียกใช้งานตัวชี้ตำแหน่งเพื่อทำงาน (อ่าน หรือ เขียน) กับตำแหน่งที่ถูกชี้โดยตัวชี้ตำแหน่งนั้น

## ข. ตัวชี้กับฟังก์ชัน

ฟังก์ชันสามารถรับพารามิเตอร์เป็นตัวชี้ตำแหน่งได้ โดยจะเรียกการส่งพารามิเตอร์แบบนี้ว่า เป็นการส่งพารามิเตอร์แบบ Pass by reference ซึ่งการส่งพารามิเตอร์แบบนี้จะมีลักษณะพิเศษคือการเปลี่ยนแปลงค่าตัวแปรที่เป็นพารามิเตอร์ที่เป็นตัวชี้จะส่งผลการเปลี่ยนแปลงนั้นต่อตัวแปรที่ถูกชี้ด้วย ตัวชี้ นั่นหมายความว่าตัวแปรที่ถูกชี้จะอยู่คนละฟังก์ชันกันก็ตาม

### ตอนที่ 1 ตัวชี้ตำแหน่ง

#### 1.1 การสร้างและการใช้งานตัวชี้ตำแหน่ง

- 1) ทดลองประกาศตัวชี้ตำแหน่งตามตัวอย่างต่อไปนี้แล้วตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int *x;
    int y=10;

    x = &y;
    //01
    //02

    return 0;
}
```

- ก) ที่บรรทัด //01 พิมพ์คำสั่ง printf(“\*x = %d\n”, \*x); เมื่อสั่ง run ได้ผลลัพธ์บนหน้าจอ

คือ \_\_\_\_\_

- ข) ที่บรรทัด //01 พิมพ์คำสั่ง printf(“\*y = %d\n”, y); เมื่อสั่ง run ได้ผลลัพธ์บนหน้าจอ

คือ \_\_\_\_\_

- ค) ผลลัพธ์ในข้อ ก) และ ข) เหมือนกันหรือไม่ \_\_\_\_\_

นศ. คิดว่าเป็นเพราะเหตุใด \_\_\_\_\_

ง) หากเปลี่ยนค่า y เป็น 20 แล้วทำซ้ำข้อ ก) และ ข้อ ข) ผลลัพธ์ที่ได้จะเปลี่ยนแปลง

อย่างไร ให้ลองอธิบายความสัมพันธ์ระหว่างตัวแปร x และ y \_\_\_\_\_

ได้ค่าเป็น 20 เท่ากัน เพราะตัวชี้ x จะชี้ไปที่ข้อมูลของตัวแปร y

จ) ที่บรรทัด //01 พิมพ์คำสั่ง printf(“\*x = %p\n”, x); เมื่อสั่ง run ได้ผลลัพธ์บนหน้าจอ

คือ \_\_\_\_\_

ณ) ที่บรรทัด //01 พิมพ์คำสั่ง printf(“\*y = %p\n”, &y); เมื่อสั่ง run ได้ผลลัพธ์บนหน้าจอ

คือ \_\_\_\_\_

ข) จากข้อ จ) และ ณ) ให้สรุปความสัมพันธ์ของ x และ y \_\_\_\_\_

x เก็บที่อยู่ของ Y

2) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int *x;
    int y=10;

    x = &y;
    y++; //01

    printf("x = %d\n", *x); //02

    return 0;
}
```

ก) นศ. คาดว่าผลลัพธ์ของโปรแกรมนี้คือ x = 11

(ให้ตรวจสอบคำตอบโดยการ run โปรแกรมนี้)

ข) แก้ไขบรรทัด //01 เป็น (\*x)++; และแก้ไขบรรทัด //02 เป็น printf("y = %d\n",y);

นศ. คาดว่าผลลัพธ์ที่แสดงบนหน้าจอคือ \_\_\_\_\_

ค) จากข้อ ก) และ ข) นศ. สรุปความสัมพันธ์ระหว่าง x และ y \_\_\_\_\_

X ขึ้นที่ตัวแปร y ดังนั้นการกระทำใดๆบน x เช่น (\*x)++ ก็

คือการกระทำเพิ่มค่าข้อมูลไปอีกหนึ่งค่าของข้อมูลในตัวแปร y นั่นเอง

3) จากข้อ 1) และข้อ 2) ให้ นศ. สรุปความเข้าใจวิธีการใช้ตัวชี้ชี้ไปยังตัวแปร ความสัมพันธ์  
ระหว่างตัวชี้และตัวถูกชี้ \_\_\_\_\_

ตัวชี้หากชี้ไปที่ตัวแปรใดๆ ก็สามารถจะกระทำใดๆกับข้อมูลตัวแปรนั้นๆได้เช่นกัน

## 1.2 การใช้ตัวชี้กับตัวแปรแบบแถวลำดับ

1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int i[5] = {1,2,3,4,5};
    int *x;

    x = i;

    //00
    printf("i[0]=%d", i[0]); //01

    return 0;
}
```

ก) เมื่อ run โปรแกรมนี้จะแสดงผลพบบนหน้าจอคือ \_\_\_\_\_

ข) แก้ไขบรรทัด //01 เป็น printf("i[0] = %d", \*x); เมื่อ run แล้วจะแสดงผลเหมือนกับข้อ ก) หรือไม่ \_\_\_\_\_ เหมือนกัน

ค) เพิ่มคำสั่ง x++; ที่บรรทัด //00 ในโปรแกรมข้อ ข) นศ. คาดว่าเมื่อ run โปรแกรมแล้วจะแสดงผลเป็น \_\_\_\_\_

เหตุใดจึงได้ผลลัพธ์เช่นนั้น \_\_\_\_\_ ตัวชี้มีการขยับไปชี้ที่ข้อมูลช่องต่อไป

ง) จากข้อ ค) หากเปลี่ยนคำสั่งในบรรทัด //00 เป็น x+=3; นศ. คาดว่าผลลัพธ์จะเป็นอย่างไร \_\_\_\_\_

เพราะเหตุใด \_\_\_\_\_ มีการขยับตัวชี้ไปชี้ที่ช่องหมายเลข 3 ซึ่งข้อมูลก็คือ 4 นั่นเอง

2) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    int i[10] = {10,20,30,40,50,60,70,80,90,0};
    int j, *x;

    x = i;

    for(j=0; j<10; j++)
    {
        printf("i[%d] = %d\n", j, *x); //00
        x++;                          //01
    }
    printf("\n\n*x = %d\n", *x);
    return 0;
}
```

ก) ผลลัพธ์ที่ได้เมื่อ run โปรแกรมนี้คือ \_\_\_\_\_

---

---

---

---

ข) หากลบบรรทัด //01 และแก้ไขบรรทัด //00 เป็น printf("i[%d] = %d\n", j, \*(x+j));

เมื่อ run โปรแกรมแล้วผลลัพธ์ที่ได้จะเหมือนกับข้อ ก) หรือไม่ มีจุดใดที่แตกต่าง

และ นศ คิดว่าเป็นเพราะเหตุใด \_\_\_\_\_

```
i[0] = 10
i[1] = 30
i[2] = 50
i[3] = 70
i[4] = 90
i[5] = -858993460
i[6] = 6
i[7] = -858993460
i[8] = -858993460
i[9] = 12327432
```

ได้ค่าที่เก็บใน i[0...9] เหมือนกัน  
แต่จะมีค่าของ \*(x+j) จะชี้ข้ามช่องไปตาม  
ค่า j ฉะนั้น หากวนเกินค่าที่เราจองไว้ 10 ช่อง  
ก็จะมีค่าอื่นที่คาดเดาไม่ได้ ดังนี้

### 1.3 การใช้ตัวชี้กับตัวแปรโครงสร้าง

- 1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

int main()
{
    struct test {
        char name[30];
        int age;
        float gpa;
    };
    struct test x, *y;
}
```

- ก) หากต้องการให้ตัวชี้ \*y ชี้ไปยังตัวแปร x จะต้องใช้คำสั่งอย่างไร \_\_\_\_\_
- ข) หากต้องการรับค่าจากแป้นพิมพ์ไปเก็บในฟิลด์ข้อมูล name ของตัวแปร โครงสร้าง x ด้วยตัวชี้ \*y จะต้องสั่งอย่างไร \_\_\_\_\_
- ค) จากข้อ ข) หากต้องการใช้ตัวชี้ \*y แสดงค่าในฟิลด์ข้อมูล name จะต้องใช้คำสั่งอย่างไร \_\_\_\_\_ (\*y).name หรือ y->name

### 1.4 ตัวชี้กับฟังก์ชัน

- 1) ใช้โปรแกรมต่อไปนี้ตอบคำถามด้านล่าง

```
#include<stdio.h>

void test(int *p)
{
    *p = 99;
}

int main()
{
    int i=10;
    test(&i);
    printf("i = %d\n", i);
    return 0;
}
```

ก) เมื่อ run โปรแกรมนี้ จะได้ผลลัพธ์เป็น i=99

เหตุใดค่าตัวแปร i จึงเปลี่ยนแปลงไป \_\_\_\_\_

ที่อยู่ของ i ถูกถือป้ไปให้กับตัวชี้ p ในฟังก์ชัน test(int \*p)

2) หาก นศ. มีตัวแปรแถวลำดับ x ขนาด 10 ช่อง และ นศ. ต้องการส่งตัวแปรแถวลำดับ x นี้ไป

ประมวลผลยัง function โดย function นี้จะทำหน้าที่แสดงข้อมูลของแต่ละช่องของตัวแปร

แถวลำดับบนหน้าจอ นศ. จะต้องเขียนฟังก์ชันนี้อย่างไร

```
#include <stdio.h>
void test(int *p){
    int i;
    for(i=0;i<10;i++)
        printf("[%d] ",*p++);
}
void main(){
    int num[10]={0,1,2,3,4,5,6,7,8,9};
    test(num);

    printf("\n");
}
```