Lab 09

L0901 ให้นักศึกษาเขียนโปรแกรมเพื่อทดลองการสร้างและใช้งาน Pointer เบื้องต้น

กำหนดให้มีตัวแปร 3 ตัว ดังนี้

- ตัวแปร apple เป็นตัวแปรเก็บข้อมูลเลขจำนวนเต็ม มีค่าเริ่มต้นเป็น 17
- ตัวแปร meter เป็นตัวแปรเก็บข้อมูลเลขทศนิยม มีค่าเริ่มต้นเป็น 22.38
- ตัวแปร order เป็นตัวแปรเก็บข้อมูลตัวอักษร 1 ตัว มีค่าเริ่มต้นเป็น 'k'

ให้นักศึกษาทำการสร้างตัวแปร Pointer เพื่อทำการเก็บเลข Address ของตัวแปรทั้ง 3 ตัว อย่างเหมาะสม โดยชื่อของ Pointer นักศึกษาสามารถตั้งชื่อได้อย่างอิสระ

จากนั้นให้นักศึกษาพิมพ์ค่าและเลข Address ของตัวแปรทั้ง 3 ตัวด้วย Pointer นักศึกษาสามารถใช้ %p เพื่อช่วยในการพิมพ์เลข Address ของตัวแปร

ตัวอย่างผล

```
apple is 17, stored at 000000000061FE04
meter is 22.38, stored at 000000000061FE00
order is k, stored at 000000000061FDFF
```

ผลลัพท์

```c
#include <stdio.h>
#include <stdlib.h>

int main() {

    int apple = 17;
    float meter = 22.38;
    char order = 'k';

    int *Apple = &apple;
    float *Meter = &meter;
    char *Order = &order;

    printf("apple is %d, stored at %p\n", *Apple,Apple);
    printf("meter is %.2f, stored at %p\n", *Meter,Meter);
    printf("order is %c, stored at %p\n", *Order,Order);

    return 0;
}
```

L0902 ให้นักศึกษาเขียนโปรแกรมเพื่อทดลองการสร้างและใช้งาน Pointer เบื้องต้น

กำหนดให้มีตัวแปร 2 ตัวสำหรับเก็บข้อมูลเลขจำนวนเต็ม ได้แก่

- ตัวแปร melon มีค่าเริ่มต้นเป็น 21
- ตัวแปร banana มีค่าเริ่มต้นเป็น 14

ให้นักศึกษาสร้างตัวแปร Pointer ขึ้นมา 2 ตัว โดยให้ Pointer ทั้ง 2 ตัวชี้ไปที่ melon ทั้งคู่ โดย Pointer ตัวหนึ่งจะทำการชี้ไปที่ melon ด้วยการเก็บ Address ของ melon โดยตรง และ Pointer อีกตัวจะทำการชี้ไปที่ melon ด้วยการ Assign มาจาก Pointer ตัวแรก

Step 1: ทำการพิมพ์ค่าที่อยู่ใน Address ที่ Pointer ทั้ง 2 ตัวชี้อยู่

Step 2: เปลี่ยนค่าที่ถูกเก็บใน Address ที่ Pointer ตัวแรกชี้ ให้กลายเป็น 77 แล้ว ทำการพิมพ์ค่าที่อยู่ใน Address ที่ Pointer ทั้ง 2 ตัวชี้อยู่

Step 3: ให้ Pointer ตัวที่ 2 ชี้ไปที่ Banana แล้วพิมพ์ค่าที่อยู่ใน Address ที่ Pointer ทั้ง 2 ตัวชี้อยู่

Step 4: เปลี่ยนค่าที่ถูกเก็บใน Address ที่ Pointer ตัวที่สองชี้ ให้มีค่าเท่ากับค่าเริ่มต้นของ melon (Assign ค่าเข้าเป็นตัวเลขโดยตรงได้) แล้วพิมพ์ค่าที่อยู่ใน Address ที่ Pointer ทั้ง 2 ตัวชี้อยู่

ตัวอย่างผล

```
step 1
pointer1 has 21 in its stored address
pointer2 has 21 in its stored address

step 2
pointer1 has 77 in its stored address
pointer2 has 77 in its stored address

step 3
pointer1 has 77 in its stored address
pointer2 has 14 in its stored address

step 4
pointer1 has 77 in its stored address
pointer2 has 21 in its stored address
```

ผลลัพท์

```
09 > C L0902.c > ...
#include <stdio.h>
#include <stdlib.h>

int main() {

    int melon = 21;
    int banana = 14;

    int *pointer1 = &melon ;
    int *pointer2 = &melon ;
    printf("step 1");
    printf("\npointer1 has %d in its stored address",*pointer1);
    printf("\npointer2 has %d in its stored address",*pointer2);

    printf("\n");

    *pointer1 = 77 ;
    printf("\nstep 2");
    printf("\npointer1 has %d in its stored address",*pointer1);
    printf("\npointer2 has %d in its stored address",*pointer2);

    printf("\n");

    pointer2 = &banana ;
    printf("\nstep 3");
    printf("\npointer1 has %d in its stored address",*pointer1);
    printf("\npointer2 has %d in its stored address",*pointer2);

    printf("\n");

    *pointer2 = 21 ;
    printf("\nstep 3");
    printf("\npointer1 has %d in its stored address",*pointer1);
    printf("\npointer2 has %d in its stored address",*pointer2);

    return 0 ;

}
```

L0903 ให้นักศึกษาเขียนโปรแกรมเพื่อทดลองการสร้างและใช้งาน Pointer และ Function เบื้องต้น

กำหนดให้มี Function ที่มี Function Prototype ดังนี้

```
float speedDistance(float speed, int *time);
```

โดย Function นี้ทำการคำนวณค่าระยะทาง (Distance) ด้วยสูตร Distance = Speed × Time แล้วให้ผลลัพธ์ออกมาเป็นค่า Distance ที่คำนวณได้ แต่ Function นี้จะมีการเพิ่มค่า Time ขึ้น 3 เมื่อคำนวณค่า Distance เสร็จสิ้นด้วย

ให้นักศึกษาเขียนโปรแกรมเพื่อเรียกใช้งาน Function นี้เป็นจำนวน 3 ครั้ง (ใช้ Loop ได้ตามสะดวก) โดยแต่ละครั้งที่เรียกใช้ Function ที่ชื่อ **`speedDistance()`** นี้ ค่า Speed ที่ใช้ ให้โปรแกรมทำการรับค่ามาจากผู้ใช้โปรแกรมผ่านทางคีย์บอร์ด เมื่อใช้งาน **`speedDistance()`** เสร็จสิ้นในแต่ละรอบ ให้พิมพ์ค่าของ Distance ที่คำนวณได้และค่า Time ออกมาด้วย

กำหนดให้ค่า Time เริ่มต้นที่ 1

ตัวอย่างผล

```
Distance calculator
-----------------------
Enter speed: 15
step 1 => distance 15.00, time 4
Enter speed: 7
step 2 => distance 28.00, time 7
Enter speed: 9
step 3 => distance 63.00, time 10
```

```
Distance calculator
-----------------------
Enter speed: 124.5
step 1 => distance 124.50, time 4
Enter speed: 42.3
step 2 => distance 169.20, time 7
Enter speed: 12.77
step 3 => distance 89.39, time 10
```

ผลลัพท์

```
09 > C L0904.c > main()
#include <stdio.h>
#include <stdlib.h>

int main() {

    int arr[12] = {4,5,6,7,8,9,10,11,12,13,14,15} ;
    int *p = arr ;

    printf("original ::: ");
    for ( int i = 0; i < 12; i++)
    {
        printf(" %d ",*(p + i));
    }

    printf("\nomultiplied ::: ");
    for ( int i = 0; i < 12; i++)
    {
        printf(" %d ",*(p + i) * 24);
    }

    return 0 ;

}
```

L0904 ให้นักศึกษาเขียนโปรแกรมเพื่อทดลองการสร้างและใช้งาน Pointer และ Array เบื้องต้น

กำหนดให้มี Array 1 ตัวที่สามารถเก็บข้อมูลเลขจำนวนเต็มได้ 12 ค่า

ให้นักศึกษาเขียนโปรแกรมเพื่อแสดงผลค่าผลคูณของสมาชิกแต่ละตัวใน Array กับ 24 โดยใช้ Pointer ในการช่วยชี้ไปที่สมาชิกแต่ละตัว

ให้นักศึกษาแสดงผลค่าเดิมของสมาชิกแต่ละตัวด้วย Pointer ประกอบไปด้วยเช่นกัน

ค่าของสมาชิกแต่ละตัวใน Array นักศึกษาสามารถกำหนดได้อย่างอิสระ ไม่จำกัดวิธีการในการกำหนดค่าให้ Array จะกำหนดโดยตรง กำหนดเป็นค่าเริ่มต้น หรือจะทำการรับค่ามาจากผู้ใช้ สามารถทำได้ทั้งหมด

ตัวอย่างผล

```
original ::: 4 5 6 7 8 9 10 11 12 13 14 15
multiplied ::: 96 120 144 168 192 216 240 264 288 312 336 360
```

ผลลัพท์

```c
#include <stdio.h>
#include <stdlib.h>

float speedDistance(float speed, int *time)
{
        float Distance =  speed * (*time) ;
        *time+=3 ;
        return Distance ;
}
int main () {

    int step ;
    float speed , Distance ;
    int time=1 ;


    printf(" Distance calculator ");
    printf("\n----------------------\n");
    for ( step = 0 ; step <= 3 ; step++ )
    {
        printf(" Enter speed : ");
        scanf("%f",&speed);
        Distance = speedDistance(speed,&time) ;

        printf("step %d => distance %.2f, time %d\n",step + 1, Distance, time);

    }

    return 0;

}
```

L090O1 ให้นักศึกษาเขียนโปรแกรมประยุกต์การใช้ความรู้จากหัวข้อเรื่อง Pointer

เขียนโปรแกรมจำลองการต่อสู้ระหว่างการ์ด 2 ใบคือการ์ดของผู้เล่นและคู่ต่อสู้ การ์ดแต่ละใบจะมีค่าพลังโจมตี (ATK) และพลังป้องกัน (DEF) การต่อสู้นั้น จะวัดกันที่ค่าพลัง ATK ของแต่ละฝ่าย ใคร ATK สูงกว่าจะชนะและทำลายอีกฝ่ายได้

แต่การ์ดของผู้เล่นมีคุณสมบัติพิเศษคือ สามารถสลับค่าพลังโจมตีและพลังป้องกันได้ โดยโปรแกรมจะทำการถามว่าต้องการจะสลับค่าเหล่านี้ของการ์ดผู้เล่นหรือไม่ ถ้าใช่ ให้สลับค่าทั้ง 2 ค่านี้

การสลับค่า ให้เขียนเป็น Function ที่มี Parameter เป็น Pointer 2 ตัวสำหรับรับ Address ของข้อมูลที่ต้องการจะสลับค่ากัน

ตัวอย่างผลลัพธ์

```
Card Battle!

Player
--------
ATK :: 1100     DEF :: 2500

Opponent
--------
ATK :: 800      DEF :: 1700

Switch player's ATK and DEF? (y/n): y

Player
--------
ATK :: 2500     DEF :: 1100

Attack calculating ATK vs ATK
Opponent destroyed!
```

```
Card Battle!

Player
--------
ATK :: 2300     DEF :: 3400

Opponent
--------
ATK :: 1400     DEF :: 200

Switch player's ATK and DEF? (y/n): n

Attack calculating ATK vs ATK
Opponent destroyed!
```

```
Card Battle!

Player
--------
ATK :: 2500     DEF :: 600

Opponent
--------
ATK :: 2300     DEF :: 1300

Switch player's ATK and DEF? (y/n): y

Player
--------
ATK :: 600      DEF :: 2500

Attack calculating ATK vs ATK
Player destroyed!
```

```
Card Battle!

Player
--------
ATK :: 800      DEF :: 1000

Opponent
--------
ATK :: 600      DEF :: 300

Switch player's ATK and DEF? (y/n): x
Error input

Switch player's ATK and DEF? (y/n): n

Attack calculating ATK vs ATK
Opponent destroyed!
```

ผลลัพท์

L080O2 ให้นักศึกษาเขียนโปรแกรมประยุกต์การใช้ความรู้จากหัวข้อเรื่อง Pointer

ให้นักศึกษาเขียนเกมบันไดงู จำลองกระดานขนาด 5×5 (25 ช่อง)

ในเกมบันไดงูนี้ ผู้เล่นจะทำการทอยลูกเต๋าจำนวน 1 ลูก จากนั้นผู้เล่นจะทำการเดินไปเป็นจำนวนช่องตามที่ลูกเต๋ากำหนด (เช่น ตอนแรกอยู่ช่องที่ 1 ทอยลูกเต๋าได้ 3 ผู้เล่นจะเดินไปช่องที่ 4)

ถ้าผู้เล่นเดินตกลงไปยังช่องที่มีเลขลงท้ายด้วย 4 (4, 14, 24) ผู้เล่นจะต้องถอยหลังไปเป็นจำนวน 3 ช่อง ถ้าผู้เล่นเดินถึงหรือเลยเส้นชัย (ช่องที่ 25) จะถือว่าจบเกม

ผู้เล่นจะเริ่มต้นเกมที่ช่องที่ 1 เสมอ

ช่องที่ผู้เล่นอยู่ ให้มีเครื่องหมายวงเล็บครอบ ช่องที่เป็นหลุมให้มีเครื่องหมาย _ ล้อม ในกรณีที่ผู้เล่นถึงเส้นชัยแล้ว ไม่ว่าจะถึงพอดีหรือเลยก็ตาม ให้วงเล็บครอบที่ช่องสุดท้าย (ช่อง 25)

Note: สามารถใช้ Pointer กับ Array ช่วยในการเดินและวาดกระดาน

ตัวอย่างผลลัพธ์

```
Snake and Ladder

Board :: (1) 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 2
Board :: 1 2 (3) _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25

Board :: 1 2 (3) _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 6
Board :: 1 2 3 _4_ 5 6 7 8 (9) 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25

Board :: 1 2 3 _4_ 5 6 7 8 (9) 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 1
Board :: 1 2 3 _4_ 5 6 7 8 9 (10) 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25

Board :: 1 2 3 _4_ 5 6 7 8 9 (10) 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 3
Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 (13) _14_ 15 16 17 18 19 20 21 22 23 _24_ 25

Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 (13) _14_ 15 16 17 18 19 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 4
Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 (17) 18 19 20 21 22 23 _24_ 25

Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 (17) 18 19 20 21 22 23 _24_ 25
Roll dice?(y): n
Error input

Roll dice?(y): y
You got 2
Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 (19) 20 21 22 23 _24_ 25

Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 (19) 20 21 22 23 _24_ 25
Roll dice?(y): y
You got 6
Board :: 1 2 3 _4_ 5 6 7 8 9 10 11 12 13 _14_ 15 16 17 18 19 20 21 22 23 _24_ (25)

FINISH!
```

ผลลัพท์