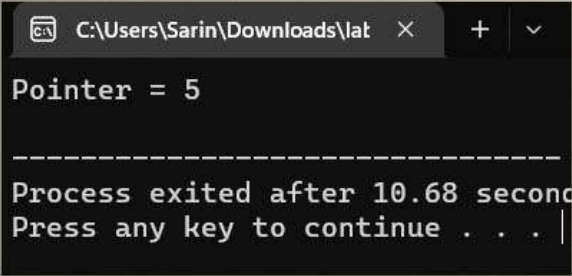


ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง
<p>pointer ช่วยทำให้ performance ของโปรแกรมดีขึ้นในการทำงานอย่างต่ำเราใช้ pointer จะทำได้เร็วกว่า หรือว่า หรือ การส่งค่าไปใน function หนึ่งจะแทนลงไปเลยซึ่งมันจะเป็นการถือค่าเพื่อสร้างตัวแปร ที่มีค่าเท่ากัน แล้วนำไปใช้ในฟังก์ชันอีกที แต่ pointer จะเป็นการเอาค่าที่จำแนกหนึ่งนั้นๆ มาใช้เลย ลดการตั้งค่าขึ้นแล้วลง และ คิดว่าประสิทธิภาพหรือการที่สะดวกของคอมพิวเตอร์ทำให้ค่าหนึ่ง (memory allocation) ก็จำเป็นที่ต้องใช้ Pointer นั่นเองก็ต้องประกาศตัวแปรก่อน แล้วรูปแบบของมันคือต้องมี (*) เหนือตัวแปรที่จะเป็น pointer of type เช่น *a, *b, *c</p>	<pre>//Pointer int main() { int *a ; int var = 5 ; a = &var ; printf("Pointer = %d\n", *a) ; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปรรูป)	Flow chart ของ Code ตัวอย่าง
	<pre>graph TD START([START]) --> Init[/ *a var = 5 /] Init --> Assign[a = &var] Assign --> Print[print *a] Print --> End[/ *a = 5 /] End --> END([END])</pre>

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว และละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย

Code ตัวอย่าง

ตัวแปรอาร์เรย์ b จะมีความเท่ากับ $8[b]$
เราจะเห็นได้ว่าตัวแปรอาร์เรย์มีลักษณะคล้าย
กับ พอยน์เตอร์ให้มองจากสีแถบค่า แอดเดรสหน่วย
กัน ผลักดันแถวของข้อมูลสามารถเปลี่ยน
แปลง ค่าของตัวแปร อาร์เรย์ให้เลื่อนไปข้างหน้า
เหมือน ลายตัวแปร พอยน์เตอร์ใช้ เช่น
จะเพิ่มค่าของ b โดย ให้คำสั่ง $b = b + x$ ใน
โปรแกรมทำได้ ดังนั้นตัวแปรอาร์เรย์จึงถูก
เรียกว่า Constant Pointer ซึ่งก็คือ
พอยน์เตอร์ที่ไม่มีการเปลี่ยนแปลงในเมื่อ ตัวแปร
อาร์เรย์จะขึ้นอยู่กับตัวที่แอดเดรสของ อีล
เมนต์แรก

```

//Pointer Array

/*int main() {
    int a[4][4] = {
        { 10, 20, 30, 40 } ,
        { 11, 21, 31, 41 } ,
        { 12, 22, 32, 42 } ,
        { 13, 23, 33, 43 }
    };
    int (*r)[4] = a ;
    //r[0][0] = 99 ;
    printf( " \n " ) ;

    for( int i = 0 ; i < 4 ; i++ ) {
        for( int j = 0 ; j < 4 ; j++ )
            printf( " %d ", a[i][j] ) ;
        printf( " \n " ) ;
    }
    return 0 ;
}*/

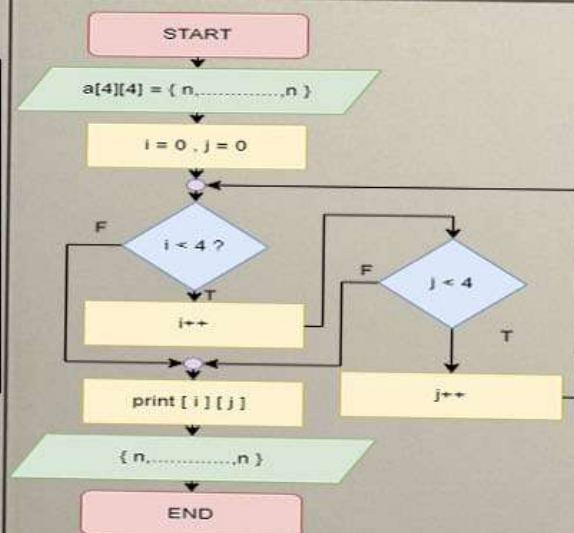
```

ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)

Flow chart ของ Code ตัวอย่าง

C:\Users\Sarin\Downloads\lat X + v

10	20	30	40
11	21	31	41
12	22	32	42
13	23	33	43



ข้อที่ 3 จงอธิบายเรื่อง Pointer Function ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย

Code ตัวอย่าง

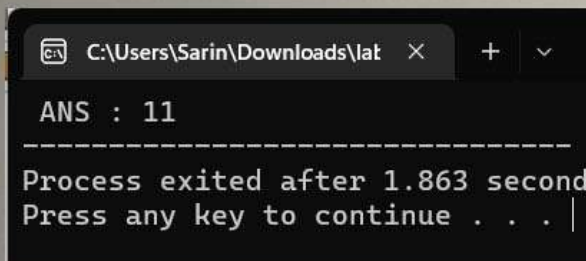
ฟังก์ชันที่ส่งค่าคืนค่าให้กับตัวแปรอื่น
เช่น ฟังก์ชันที่ส่งค่าคืนค่าให้กับตัวแปรอื่น
โดย ต้องมีตัวแปรที่รับค่าคืนค่าไว้
typename C* functionname(parameters)
การเขียนฟังก์ชันที่ส่งค่าคืนค่าให้กับตัวแปรอื่น
จากก่อนที่เราจะเขียนโปรแกรมเชิงวัตถุ
ช่วยให้เราสามารถเขียนโปรแกรมเชิงวัตถุ
และ เมื่อได้ผลลัพธ์แล้วสามารถส่งค่าคืนค่าให้กับตัวแปรอื่น

```
//Pointer Function*/
int main() {
    int i ;
    int x = 5 ;
    int y = 6 ;
    int *a = &x ;
    int **b = &a ;

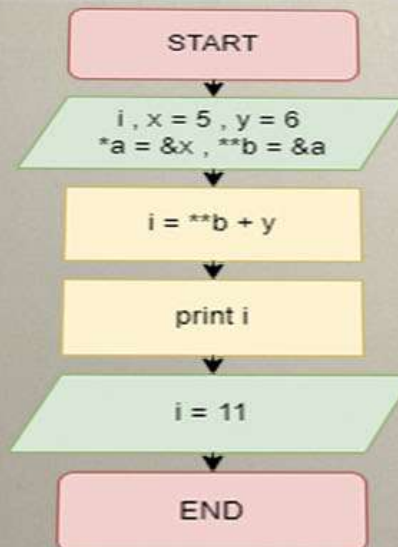
    i = **b + y ;
    printf( " ANS : %d ", i ) ;
}
/*
```

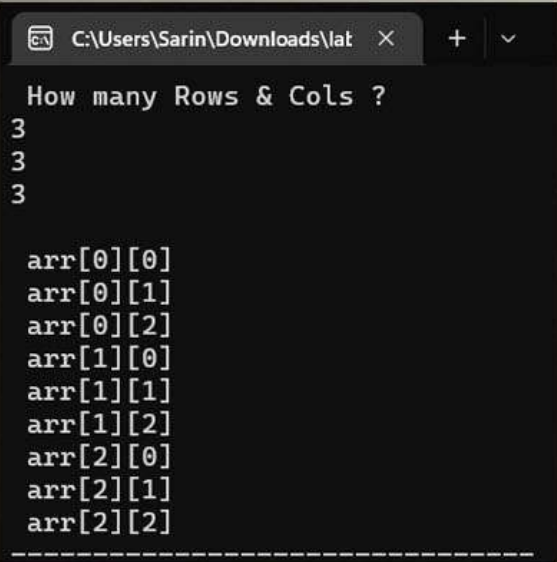
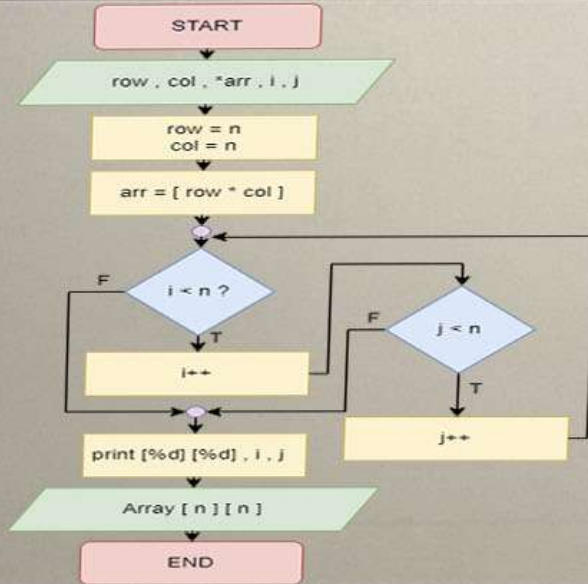
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)

Flow chart ของ Code ตัวอย่าง



```
ANS : 11
Process exited after 1.863 second
Press any key to continue . . . |
```



ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>อาร์เรย์ไดนามิกมีลักษณะเหมือนกับอาร์เรย์ธรรมดา แต่มันสามารถเปลี่ยนขนาดได้โดยอัตโนมัติ ไม่ต้องกังวลเรื่องขนาด ส่วนประกอบ 'Dynamic Array' นั้นมีลักษณะคล้ายคลึงกับอาร์เรย์ธรรมดา แต่เราสามารถปรับขนาดได้โดยอัตโนมัติ</p>	<pre style="font-family: monospace;"> //Dynamic Array*/ int main() { int row ; int col ; printf(" How many Rows & Cols ? \n") ; scanf(" %d ", &row) ; scanf(" %d ", &col) ; int *arr ; arr = new int [row * col] ; for(int i = 0 ; i < row ; i++) for(int j = 0 ; j < col ; j++) { printf(" \n arr[%d][%d] ", i , j) ; } return 0 ; } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	

ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย

Code ตัวอย่าง

Pass by Reference คือ
การส่งตัวแปร (variable) เข้า
argument ของฟังก์ชัน ดังนั้น ตัวแปร
ที่ส่งค่าเข้ามาในฟังก์ชัน จะส่งค่า
ในตัวแปรของฟังก์ชันกลับไปเปลี่ยนแปลง
ด้วย

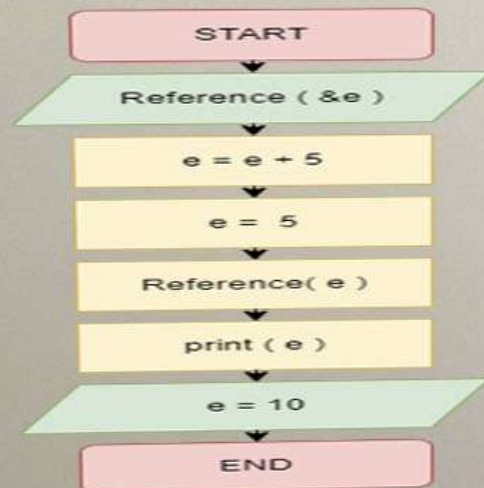
```
//Pass By Reference*/
void reference( int &e ) {
    e = e + 5 ;
}
int main() {
    int e = 5 ;
    reference( e ) ;
    printf("Reference : %d", e ) ;
    return 0 ;
}
/*
```

ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)

Flow chart ของ Code ตัวอย่าง

C:\Users\Sarin\Downloads\lat x + v
Reference : 10

Process exited after 1.305 second
Press any key to continue . . . |



ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

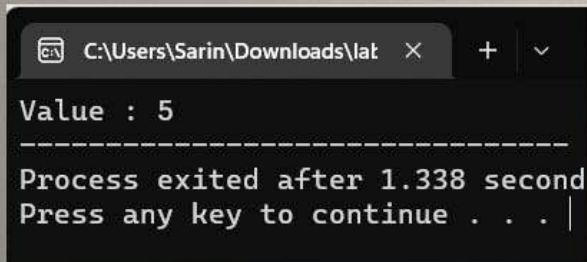
คำอธิบาย

Pass by Value คือ การส่งค่า (Value) ไปเป็น argument ของฟังก์ชัน ดังนั้นค่าที่ส่งไปนั้นฟังก์ชันจึงไม่สามารถส่งผลต่อตัวแปรนอกฟังก์ชัน

Code ตัวอย่าง

```
//Pass By Value*/
void value( int e ) {
    e = e + 5 ;
}
int main() {
    int e = 5 ;
    value( e ) ;
    printf("Value : %d", e ) ;
    return 0 ;
}
```

ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)



```
C:\Users\Sarin\Downloads\lat
Value : 5
-----
Process exited after 1.338 second
Press any key to continue . . .
```

Flow chart ของ Code ตัวอย่าง

