

ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวิเคราะห์ประกอบตามความเข้าใจของคุณ

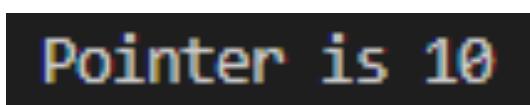
ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง
<p>Pointer เป็นตัวบันทึกที่บันทึกตำแหน่งในหน่วยความจำที่ใช้จราจรทางเรียกชื่อพื้นที่</p> <p>ตัวบันทึกที่บันทึกหน่วยความจำของแต่ละแบบ</p> <p>จุดเริ่มต้นของ Pointer</p> <p>ampersand (&) เป็นตัวลักษณะที่บันทึกไว้ในภาษา C</p> <p>Dereference Operator (*) ตัวดำเนินการประมวลผล</p> <p>Declaration Operator (?) ตัวดำเนินการประกาศ</p>	<pre>C Lab_1.c X C: > Users > RYZEN > Desktop > C Lab_1.c > main() ● 1 #include <stdio.h> 2 3 int main() 4 { 5 int number = 10; 6 float money = 34.8; 7 8 int *p1 = &number; 9 float *p2; 10 p2 = &money; 11 12 printf("p1 address = %x\n", p1); 13 printf("p1 value = %d\n", *p1); 14 printf("p2 address = %x\n", p2); 15 printf("p2 value = %f\n", *p2); 16 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	<p>Flow chart ของ Code ตัวอย่าง</p> <pre> graph TD START([START]) --> InitInt[int'ก้าวนด money เป็น ตัวแปร integer = 10'] InitInt --> InitFloat[int'ก้าวนด money เป็น ตัวแปร float = 34.8'] InitFloat --> P1[int'ก้าวนด pointer p1 รันค่า integer'] P1 --> P2[int'ก้าวนด pointer p2 รันค่า float'] P2 --> A{A} A --> PrintP1[/แสดงค่าเลขฐาน 16 ของ p1/] PrintP1 --> PrintP1Val[/แสดงค่าตัวแปร ของ p1/] PrintP1Val --> PrintP2[/แสดงค่าเลขฐาน 16 ของ p2/] PrintP2 --> PrintP2Val[/แสดงค่าตัวแปร ของ p2/] PrintP2Val --> END([END]) </pre>
<pre>p1 address = 61ff14 p1 value = 10 p2 address = 61ff10 p2 value = 34.799999 PS C:\Users\RYZEN\Desktop></pre>	

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดແຄວ ແຄລະ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>จงต้องการให้คือ รีบุค ให้คือ ต้องการที่จะต้อง Pointer ให้คือ ต้องการที่จะต้อง</p> <p>Pointer ต้องการที่จะต้อง ต้องการที่จะต้อง ต้องการที่จะต้อง</p> <p>Pointer ต้องการที่จะต้อง ต้องการที่จะต้อง ต้องการที่จะต้อง</p>	<pre>C: > Users > RYZEN > Desktop > LAB_อาจารย์เต็ม > C LAB_1_2.c > main() 1 #include<stdio.h> 2 3 int main(){ 4 int number[5] = {0, 10, 20, 30, 40}; 5 int *Pointer = number; 6 7 printf("1st Place -> %d\n", *Pointer); 8 Pointer++; 9 printf("2nd Place -> %d\n", *Pointer); 10 Pointer += 3; 11 printf("5nd Place -> %d\n", *Pointer); 12 Pointer--; 13 printf("4th Place -> %d\n", *Pointer); 14 15 }</pre>
ผลลัพธ์ของ Code (Captureภาพแบบรูป)	Flow chart ของ Code ตัวอย่าง
<pre>1st Place -> 0 2nd Place -> 10 5nd Place -> 40 4th Place -> 30 PS C:\Users\RYZEN\Desktop\LAB_อาจารย์เต็ม></pre>	<pre> graph TD START((START)) --> Init[กำหนดตัวแปร Array ขนาด 5 ค่า 0,10,20,30,40] Init --> SetP[กำหนดตัวแปร Pointer ให้มีค่า number] SetP --> Print1{1st Place -> %d ได้ผลลัพธ์จาก Pointer *Pointer} Print1 --> Inc1[Pointer++] Inc1 --> Print2{2nd Place -> %d ได้ผลลัพธ์จาก Pointer *Pointer} Print2 --> Inc2[Pointer += 3] Inc2 --> Print5{5nd Place -> %d ได้ผลลัพธ์จาก Pointer *Pointer} Print5 --> Dec1[Pointer--] Dec1 --> Print4{4th Place -> %d ได้ผลลัพธ์จาก Pointer *Pointer} END((END)) </pre>

ข้อที่ 3 จะอธิบายเรื่อง Pointer Function กดตัวอย่างการใช้งาน พิจารณาพร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
เขียนฟังก์ชันที่รับค่าแบบพุ่งเป้ามา แล้วพิมพ์ค่าที่ได้รับมา ตามรูปแบบ	C: > Users > RYZEN > Desktop > LAB_อาจารย์ต้ม > C LAB_1_3.c 1 #include <stdio.h> 2 3 void fun(int a) 4 { 5 printf("Pointer is %d\n", a); 6 } 7 8 int main() 9 { 10 void (*fun_pointer)(int) = fun; 11 fun_pointer(10); 12 return 0; 13 }
ผลลัพธ์ของ Code (Captureพร้อมภาพรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD START((START)) --> P1(()) P1 --> P2[กำหนด Pointer Function ชื่อ fun_pointer รับค่าจาก main] P2 --> P3[ตัว fun_pointer เป็น 10] P3 --> P4[Void fun รับค่า int a] P4 --> END((END)) </pre> <p>Flow chart ของ Code ตัวอย่าง</p> <p>Flowchart description:</p> <ul style="list-style-type: none"> START Flow to a decision point (empty circle). From the decision point, flow to a process box: "กำหนด Pointer Function ชื่อ fun_pointer รับค่าจาก main". From this process, flow to another process box: "ตัว fun_pointer เป็น 10". From this process, flow to a final process box: "Void fun รับค่า int a". From this final process, flow to END.

ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>Dynamic Array คือตัวแปรที่สามารถเปลี่ยนขนาดได้โดยอัตโนมัติ สำหรับการจัดเก็บข้อมูล</p> <p>เพื่อจัดเก็บข้อมูลที่ไม่แน่นอนจำนวน</p>	<pre> 1 #include <stdio.h> 2 3 int main(){ 4 5 int row,col; 6 printf("Input your row and column: "); 7 scanf("%d %d",&row,&col); 8 9 int *a; 10 a = new int(row*col) ; 11 12 for(int i = 0; i < row; i++){ 13 for(int j = 0; j < col; j++){ 14 printf("a[%d][%d] = ",i,j); 15 scanf("%d", &a[i * col + j]); 16 } 17 } 18 for(int i = 0; i < row * col ; i++){ 19 printf("%d\t", a[i]); 20 if((i+1)%col == 0){ 21 printf("\n"); 22 } 23 } 24 return 0 ; 25 }</pre>
ผลลัพธ์ของ Code (Captureภาพมุมแปรรูป)	Flow chart ของ Code ตัวอย่าง
<p>Input your row and column: 2</p> <p>2</p> <p>a[0][0] = 1</p> <p>a[0][1] = 2</p> <p>a[1][0] = 3</p> <p>a[1][1] = 4</p> <p>1 2</p> <p>3 4</p>	<pre> graph TD Start([Start]) --> InputRow[/Input your row and column/] InputRow --> InputNum[/Input num/] InputNum --> InitP[/Initialize p/] InitP --> DecisionA{p < num?} DecisionA -- Yes --> Pplus1[/p++/] Pplus1 --> PrintP[/Print array[p]/] PrintP --> DecisionA DecisionA -- No --> YourNumbers[/Your numbers are/] YourNumbers --> End([End]) </pre>

ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p><u>Pass by Ref.</u> คือการส่งตัวแปรโดยนำที่อยู่ของตัวแปรไปยังที่อื่น</p> <p>ที่อยู่ (Address) ของตัวแปรจะถูกเปลี่ยนแปลงตามที่ต้องการ</p> <p>ตัวอย่าง</p>	<pre> 1 #include <stdio.h> 2 3 int main(){ 4 int a = 100; 5 int b = 200; 6 7 printf("Before swap, value of a ; %d\n", a); 8 printf("Before swap, value of b ; %d\n", b); 9 10 swap(&a, &b); 11 12 printf("After swap, value of a ; %d\n", a); 13 printf("After swap, value of b ; %d\n", b); 14 15 return 0; 16 } 17 void swap(int *x, int *y){ 18 19 int temp; 20 temp = *x; 21 *x = *y; 22 *y = temp; 23 24 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมภาพรูป)	<p>Flow chart ของ Code ตัวอย่าง</p> <pre> graph TD Start((Start)) --> InitA[กำหนดค่า a = 100] InitA --> PrintAB[Before swap, value of a ; 100 Before swap, value of b ; 200] PrintAB --> Swap[swap(&a,&b)] Swap --> PrintAB2[Before swap, value of a ; 200 Before swap, value of b ; 100] PrintAB2 --> End((End)) StartVoidSwap((Start void swap กำหนดค่า x และ y)) --> TempInt[กำหนดค่า temp เป็น integer] TempInt --> PointerX[กำหนดค่า pointer x =] PointerX --> PointerY[กำหนดค่า pointer y = temp] PointerY --> End </pre>

Before swap, value of a ; 100
 Before swap, value of b ; 200
 After swap, value of a ; 200
 After swap, value of b ; 100
 PS C:\Users\RYZEN\Desktop\LAB

ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>Pass by value คือการส่งตัวแปรไปยัง function โดย function จะมีการcopy ค่าของตัวแปรมาไว้ในพื้นที่ที่ต้องการ</p> <p>ผลลัพธ์</p>	<pre> 1 void swap(int asw, int bsw) 2 { 3 int temp; 4 5 temp = asw; 6 asw = bsw; 7 bsw = temp; 8 9 return; 10 } 11 12 int main(void) 13 { 14 int a = 100; 15 int b = 200; 16 17 printf("Before swap: value of a: %d\n", a); 18 printf("Before swap: value of b: %d\n", b); 19 20 swap(a, b); 21 22 printf("After swap: value of a: %d\n", a); 23 printf("After swap: value of b: %d\n", b); 24 25 return 0; 26 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	<p>Flow chart ของ Code ตัวอย่าง</p> <pre> graph TD StartMain([Start main(void)]) --> SetA[a = 100 b = 200 เป็น integer] SetA --> BeforeSwapA[Before swap : value of a: %d คือ a] BeforeSwapA --> Swap[swap(a,b)] Swap --> AfterSwapA[After swap : value of a: %d คือ a] AfterSwapA --> EndMain([End]) StartSwap([Start void swap (int asw,bsw)]) StartSwap --> TempInit[temp = asw] TempInit --> AswSet[asw = bsw] AswSet --> BswSet[bsw = temp] BswSet --> AfterSwapB[After swap : value of b: %d คือ b] AfterSwapB --> EndSwap([End]) </pre>

Before swap: value of a: 100
 Before swap: value of b: 200
 After swap: value of a: 100
 After swap: value of b: 200