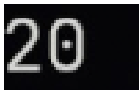

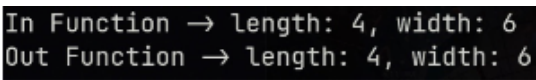
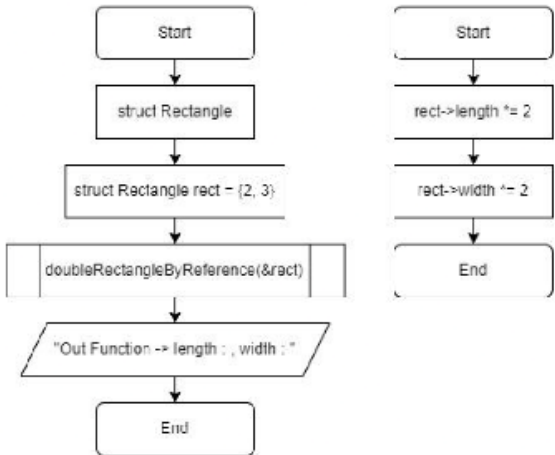


ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

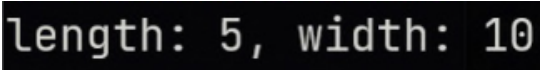
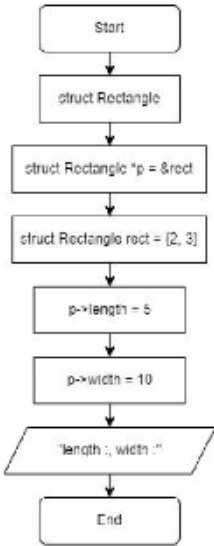
จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

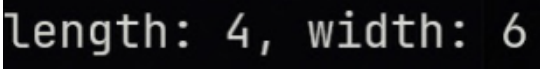
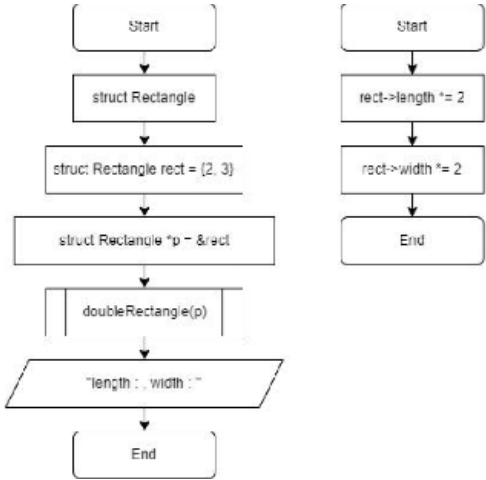
ข้อที่ 1 จงอธิบายความหมายของ Structure อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>structure คือการเก็บข้อมูลเป็นจุดโดยใช้ตัวแปรเดี่ยว โดยการเรียนรู้ Structure นั้นสามารถเรียกใช้ข้อมูลโดยใช้ (.) จุด ตามหลังชื่อของ Structure structName avalue</p>	<pre>#include <stdio.h> struct student { char name[20]; int age; char sex; float gpa; }; typedef struct student stds; int main() { stds aboy ; aboy.age = 20; printf("%d\n", aboy.age); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre>graph TD Start([Start]) --> Declaration[student student(stds)] Declaration --> Declaration2[stds aboy] Declaration2 --> Assignment[aboy.age = 20] Assignment --> Print[/printf (aboy.age)/] Print --> Stop([Stop])</pre>

ข้อที่ 2 จงแสดงวิธีการส่งผ่านค่าตัวแปรประเภท Struct แบบ Pass by reference และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>การส่งค่าตัวแปรประเภท struct แบบ Pass by reference นั้นจะทำการเปลี่ยนแปลงข้อมูล Argument ที่ส่งมา</p>	<pre>#include <stdio.h> struct Rectangle { int length; int width; }; void doubleRectangleByReference(struct Rectangle *rect) { rect->length *= 2; rect->width *= 2; printf("In Function -> length: %d, width: %d\n", rect->length, rect->width); } int main() { struct Rectangle rect = {2, 3}; doubleRectangleByReference(&rect); printf("Out Function -> length: %d, width: %d\n", rect.length, rect.width); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	

ข้อที่ 3 จงแสดงวิธีการส่งผ่านค่าตัวแปรประเภท Struct แบบ Pass by value และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
การส่งค่า struct pass-by-value นั้นจะทำให้ค่าการเปลี่ยนแปลงข้อมูล argument ที่ส่งมาภายใน function ต้อง return คืนค่ากลับมา	<pre>#include <stdio.h> struct Rectangle { int length; int width; }; void doubleRectangleByValue(struct Rectangle rect) { rect.length *= 2; rect.width *= 2; printf("In Function -> length: %d, width: %d\n", rect.length, rect.width); } int main() { struct Rectangle rect = {2, 3}; doubleRectangleByValue(rect); printf("Out Function -> length: %d, width: %d\n", rect.length, rect.width); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
<div>In Function → length: 4, width: 6 Out Function → length: 2, width: 3</div>	<pre> graph TD subgraph Left_Path [] direction TB L_Start([Start]) --> L_Struct[struct Rectangle] L_Struct --> L_Rect["struct Rectangle rect = {2, 3}"] L_Rect --> L_Func[doubleRectangleByValue(rect)] L_Func --> L_Out[/Out Function -> length : , width :/] L_Out --> L_End([End]) end subgraph Right_Path [] direction TB R_Start([Start]) --> R_Length["rect.length *= 2"] R_Length --> R_Width["rect.width *= 2"] R_Width --> R_End([End]) end </pre>

ข้อที่ 4 จงอธิบายการใช้ Struct ร่วมกับ Pointer ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer	
คำอธิบาย	Code ตัวอย่าง
<p>นำ *p เก็บข้อมูล struct ที่ประกาศไว้แล้วแทนค่า struct ใหม่</p> <p>โดยใช้ *p จะพบว่าข้อมูลของ struct ที่เก็บไว้เปลี่ยนแปลง</p>	<pre>#include <stdio.h> struct Rectangle { int length; int width; }; int main() { struct Rectangle rect = {2, 3}; struct Rectangle *p = &rect; p->length = 5; p->width = 10; printf("length: %d, width: %d\n", rect.length, rect.width); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre>graph TD Start([Start]) --> A[struct Rectangle] A --> B[struct Rectangle *p = &rect] B --> C[struct Rectangle rect = {2, 3}] C --> D[p->length = 5] D --> E[p->width = 10] E --> F[/length : width :/] F --> End([End])</pre>

ข้อที่ 5 จงประยุกต์ใช้ Function ส่งผ่าน Struct แบบ Pointer ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer	
คำอธิบาย	Code ตัวอย่าง
นำ *p เก็บข้อมูล Struct ที่ประกาศไว้แล้วแทนค่า Struct ใหม่โดยใช้ *P แล้วทำให้ P เชื้อไปทำงานใน Function จะพบว่า struct ที่ประกาศไม่มีการเปลี่ยนแปลงตาม Function ที่คานาได้ที่ *p	<pre>#include <stdio.h> struct Rectangle { int length; int width; }; void doubleRectangle(struct Rectangle *rect) { rect->length *= 2; rect->width *= 2; } int main() { struct Rectangle rect = {2, 3}; struct Rectangle *p = &rect; doubleRectangle(p); printf("length: %d, width: %d\n", rect.length, rect.width); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD subgraph Left_Path [] direction TB L_Start([Start]) --> L_Struct[struct Rectangle] L_Struct --> L_Decl[struct Rectangle rect = {2, 3}] L_Decl --> L_Ptr[struct Rectangle *p = &rect] L_Ptr --> L_Func[doubleRectangle(p)] L_Func --> L_Out[/length : , width :/] L_Out --> L_End([End]) end subgraph Right_Path [] direction TB R_Start([Start]) --> R_Len[rect->length *= 2] R_Len --> R_Width[rect->width *= 2] R_Width --> R_End([End]) end </pre>