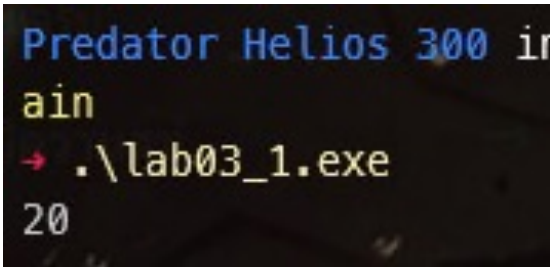


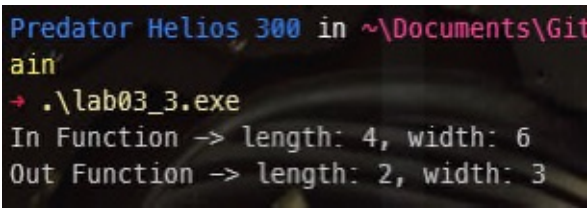
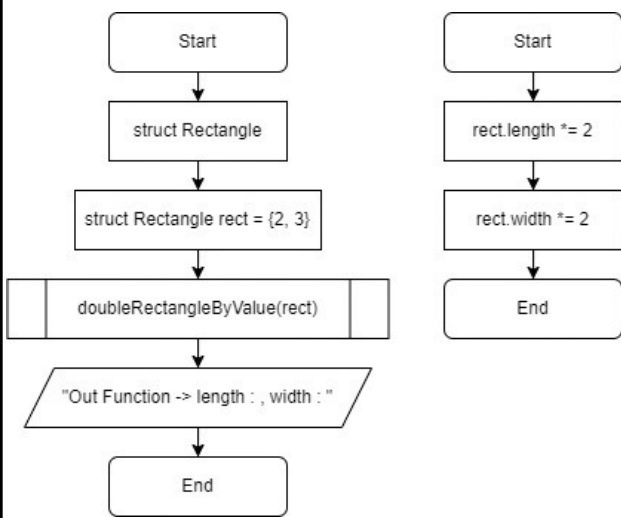
## ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

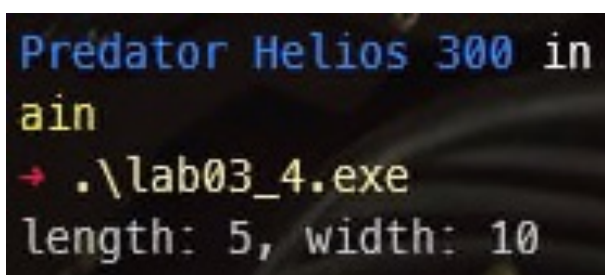
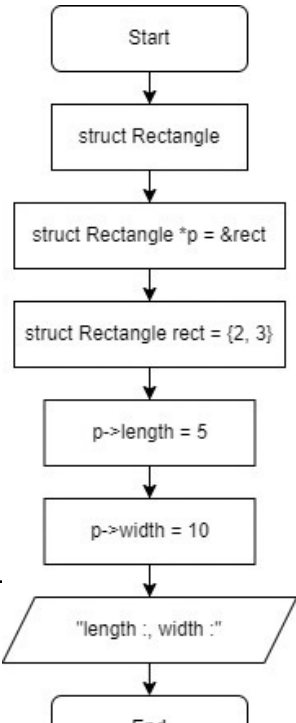
ข้อที่ 1 จงอธิบายความหมายของ Structure อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

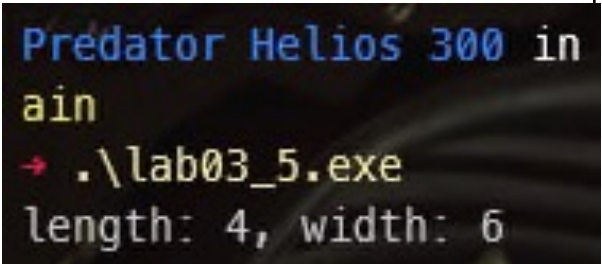
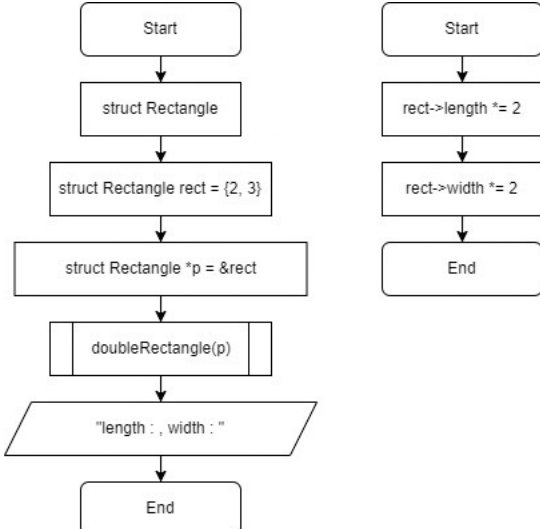
คำอธิบาย	Code ตัวอย่าง
<p>Structure คือการเก็บข้อมูลหลายๆ อย่างไว้ด้วยกัน สามารถสร้างได้หลายตัวแปร และสามารถเรียกใช้งานได้โดยการใช้ (.) เช่น Struct Name . Value</p>	<pre> 1  #include &lt;stdio.h&gt; 2 3  struct student { 4      char name[20]; 5      int age; 6      char sex; 7      float gpa; 8  }; typedef struct student stds; 9 10 int main() { 11     stds aboy ; 12     aboy.age = 20; 13     printf("%d\n", aboy.age); 14     return 0; 15 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD     Start([Start]) --&gt; struct_student[struct student]     struct_student --&gt; stds_aboy[stds aboy]     stds_aboy --&gt; aboy_age[aboy.age = 20]     aboy_age --&gt; display_age[/แสดงค่า aboy.age/]     display_age --&gt; End([End])     </pre>

ข้อที่ 2 จงแสดงวิธีการส่งผ่านค่าตัวแปรประเภท Struct แบบ Pass by reference และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Struct แบบ Pass by reference มีทั้ง Pointer ชี้ Struct เป็นแบบ Parameter และ แบบ Array เก็บ Struct เป็น Parameter</p> <p>ถ้ามีการเปลี่ยนแปลงสมาชิกใน Function และ Main</p>	<pre> 1 #include &lt;stdio.h&gt; 2 3 struct Rectangle { 4     int length; 5     int width; 6 }; 7 8 void doubleRectangleByReference(struct Rectangle *rect) { 9     rect-&gt;length *= 2; 10    rect-&gt;width *= 2; 11    printf("In Function -&gt; length: %d, width: %d\n", rect-&gt;length, rect-&gt;width); 12 } 13 14 int main() { 15     struct Rectangle rect = {2, 3}; 16     doubleRectangleByReference(&amp;rect); 17     printf("Out Function -&gt; length: %d, width: %d\n", rect.length, rect.width); 18     return 0; 19 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD     subgraph MainFlow [Main Function Flow]         Start1([Start]) --&gt; InitRect[struct Rectangle rect = {2, 3}]         InitRect --&gt; CallFunc[doubleRectangleByReference(&amp;rect)]         CallFunc --&gt; PrintOut[/"Out Function -&gt; length : , width : "/]         PrintOut --&gt; End1([End])     end      subgraph SubFuncFlow [Sub Function Flow]         Start2([Start]) --&gt; LenCalc[rect-&gt;length *= 2]         LenCalc --&gt; WidCalc[rect-&gt;width *= 2]         WidCalc --&gt; End2([End])     end </pre>

ข้อที่ 3 จงแสดงวิธีการส่งผ่านค่าตัวแปรประเภท Struct แบบ Pass by value และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Struct แบบ Pass by value ข้อมูลเข้าได้อย่างเดียว ถ้ามีการเปลี่ยนแปลงภายใน Function หรือใน Main ก็จะไม่เปลี่ยนแปลงตาม</p>	<pre> 1  #include &lt;stdio.h&gt; 2 3  struct Rectangle { 4      int length; 5      int width; 6  }; 7 8  void doubleRectangleByValue(struct Rectangle rect) { 9      rect.length *= 2; 10     rect.width *= 2; 11     printf("In Function -&gt; length: %d, width: %d\n", rect.length, rect.width); 12 } 13 14 int main() { 15     struct Rectangle rect = {2, 3}; 16     doubleRectangleByValue(rect); 17     printf("Out Function -&gt; length: %d, width: %d\n", rect.length, rect.width); 18     return 0; 19 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <pre> Predator Helios 300 in ~\Documents\Git ain' → .\lab03_3.exe In Function -&gt; length: 4, width: 6 Out Function -&gt; length: 2, width: 3 </pre>	 <pre> graph TD     subgraph Left_Process [ ]         direction TB         L_Start([Start]) --&gt; L_Struct[struct Rectangle]         L_Struct --&gt; L_Rect["struct Rectangle rect = {2, 3}"]         L_Rect --&gt; L_Func["doubleRectangleByValue(rect)"]         L_Func --&gt; L_Print[/"Out Function -&gt; length : , width : "/]         L_Print --&gt; L_End([End])     end     subgraph Right_Process [ ]         direction TB         R_Start([Start]) --&gt; R_Length["rect.length *= 2"]         R_Length --&gt; R_Width["rect.width *= 2"]         R_Width --&gt; R_End([End])     end     L_End --&gt; Final_End([End])     R_End --&gt; Final_End </pre>

**ข้อที่ 4** จงอธิบายการใช้ Struct ร่วมกับ Pointer ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง
<p>การชี้ Pointer ใน Struct คือการใช้ Pointer รับค่าของตัวแปร และ สามารถนำค่า Pointer ไปใช้ใน ฟังก์ชันต่างๆได้</p>	<pre>1 #include &lt;stdio.h&gt; 2 3 struct Rectangle { 4     int length; 5     int width; 6 }; 7 8 int main() { 9     struct Rectangle rect = {2, 3}; 10    struct Rectangle *p = &amp;rect; 11    p-&gt;length = 5; 12    p-&gt;width = 10; 13    printf("length: %d, width: %d\n", rect.length, rect.width); 14    return 0; 15 }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	

ข้อที่ 5 จงประยุกต์ใช้ Function ส่งผ่าน Struct แบบ Pointer ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer	
คำอธิบาย	Code ตัวอย่าง
<p>ขั้นแรกสร้างตัวแปรเก็บ struct ที่ประกาศไว้ จากนั้นนำค่าไปคำนวณใน function เพิ่มขนาดพื้นที่โดยคูณ 2 เข้าไปจากหน้าค่ากลับมาแสดงใน main</p>	<pre> 1  #include &lt;stdio.h&gt; 2 3  struct Rectangle { 4      int length; 5      int width; 6  }; 7 8  void doubleRectangle(struct Rectangle *rect) { 9      rect-&gt;length *= 2; 10     rect-&gt;width *= 2; 11 } 12 13 int main() { 14     struct Rectangle rect = {2, 3}; 15     struct Rectangle *p = &amp;rect; 16 17     doubleRectangle(p); 18     printf("length: %d, width: %d\n", rect.length, rect.width); 19     return 0; 20 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD     subgraph Left_Path         Start1([Start]) --&gt; RectDef[struct Rectangle]         RectDef --&gt; RectInit[struct Rectangle rect = {2, 3}]         RectInit --&gt; PtrAssign[struct Rectangle *p = &amp;rect]         PtrAssign --&gt; FuncCall[doubleRectangle(p)]         FuncCall --&gt; Print[/length : , width : /]         Print --&gt; End1([End])     end     subgraph Right_Path         Start2([Start]) --&gt; LenMult[rect-&gt;length *= 2]         LenMult --&gt; WidMult[rect-&gt;width *= 2]         WidMult --&gt; End2([End])     end </pre>