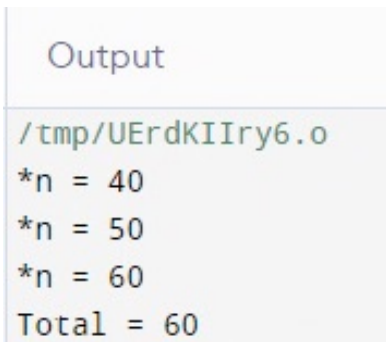
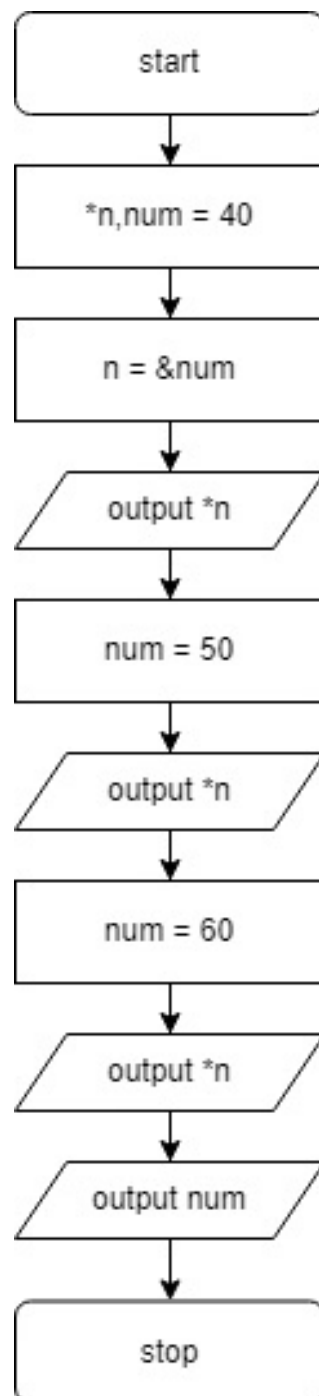


## ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

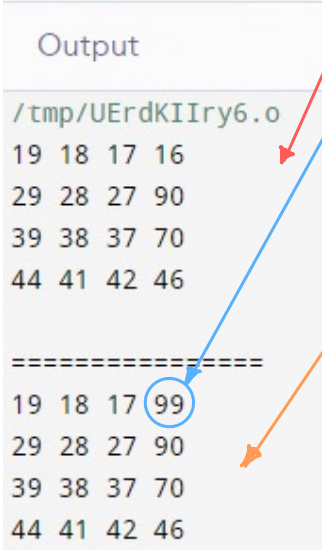
จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer	
คำอธิบาย	Code ตัวอย่าง
<p>Pointer คือ ตัวชี้ ถือเป็นลักษณะหนึ่งของ Operator ในภาษา C</p> <p>มีประโยชน์ในการจัดการข้อมูลแบบ Dynamic data structure</p> <p>ตัวแปรชนิด Pointer จะเก็บค่า ที่อยู่ ของหน่วยความจำ ( แตกต่างกับตัวแปรชนิดอื่นที่เก็บแค่ค่าจริง )</p> <p>รูปแบบการประกาศตัวแปร</p> <p>ชนิดข้อมูล * ชื่อตัวแปร เช่น <code>int * variable;</code></p>	<pre>#include &lt;stdio.h&gt;  int main () {      int *n, num = 40;      n = &amp;num;      printf( "*n = %d\n", *n );      num = 50;      printf( "*n = %d\n", *n );      *n = 60;      printf( "*n = %d\n", *n );      printf( "Total = %d\n", *n );       return 0;  }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <p>Output</p> <pre>/tmp/UErdKIry6.o *n = 40 *n = 50 *n = 60 Total = 60</pre>	<p>หน้าถัดไป</p> <p>↓</p>

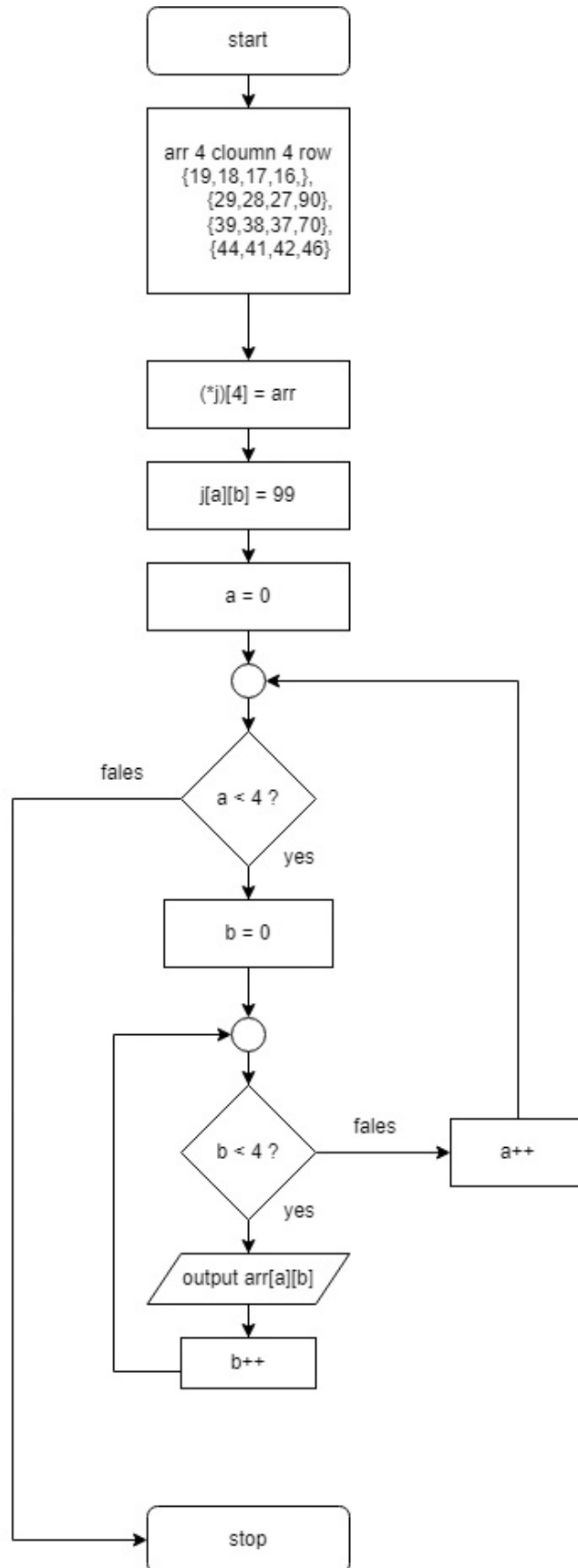
## Flow Chart vs Code Ex.

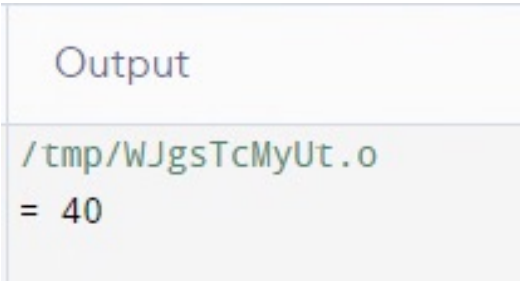
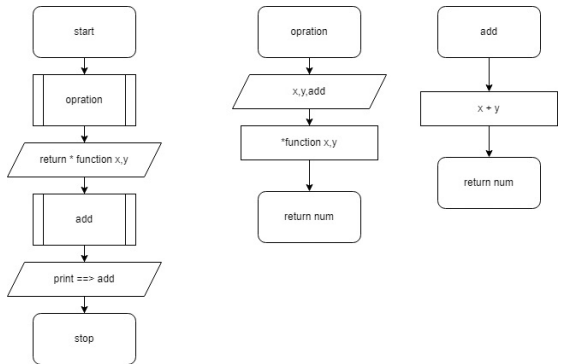



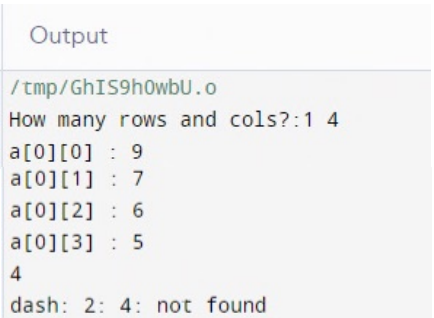
ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ใช้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
Pointer จะทำการบอกค่าที่อยู่ใน Array และทำการรวม/เปลี่ยนค่าใน Column / Row นั้นๆ	<pre>#include &lt;stdio.h&gt;  int main() {     int n = 4, arr[4][4] = {         { 19, 18, 17, 16 }, { 29, 28, 27, 90 },         { 39, 38, 37, 70 }, { 44, 41, 42, 46 }     };      for (int a = 0; a &lt; 4; a++) {         for (int b = 0; b &lt; 4; b++) {             printf("/.d ", arr[a][b]);         }         printf("\n");     } // End output old table.      int (*j)[4] = arr;     j[0][3] = 99;     for (int a = 0; a &lt; 4; a++) {         for (int b = 0; b &lt; 4; b++) {             printf("/.d ", arr[a][b]);         }         printf("\n");     } // End output now table.      return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	 <p>Output</p> <pre>/tmp/UErdKIIry6.o 19 18 17 16 29 28 27 90 39 38 37 70 44 41 42 46  ===== 19 18 17 99 29 28 27 90 39 38 37 70 44 41 42 46</pre> <p>Flow Chart ข้อ 2 หน้าถัดไป ↓</p>

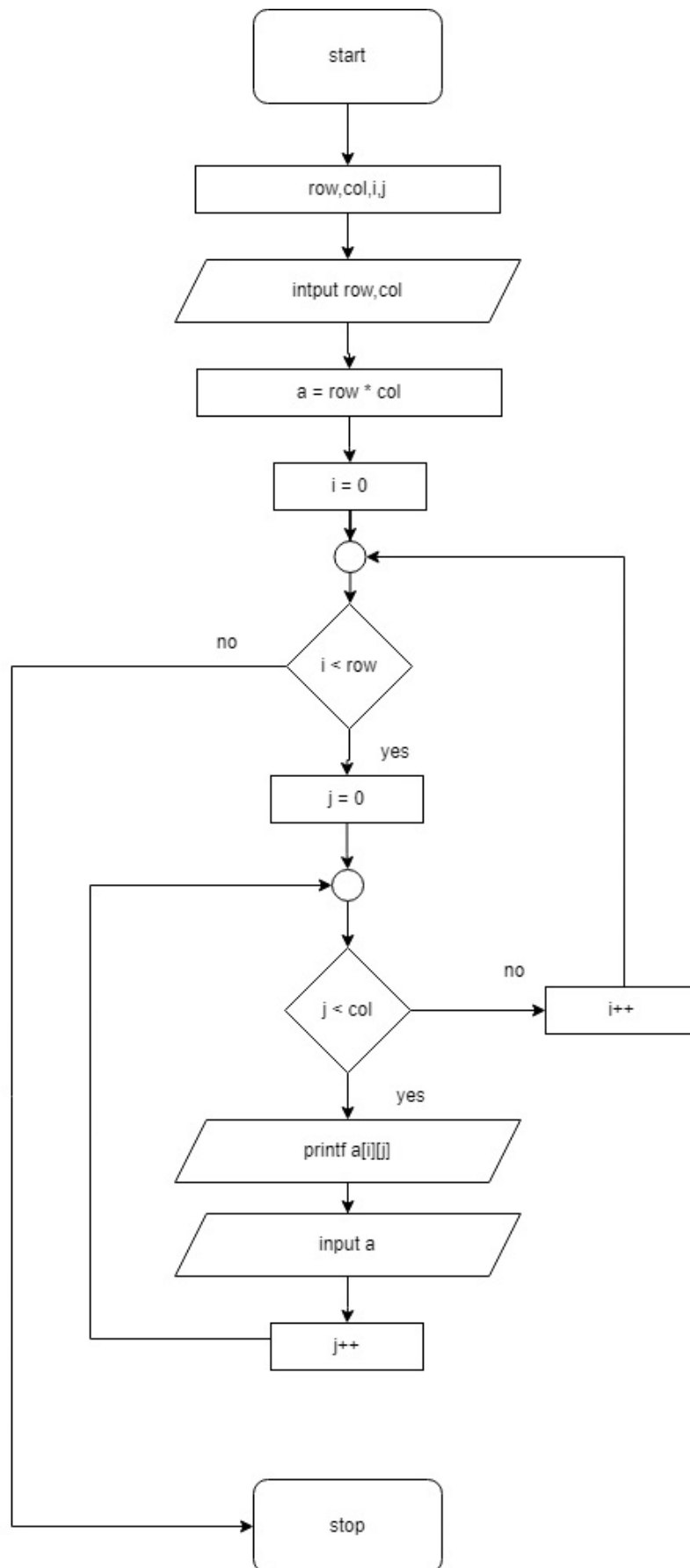
## Flow Chart no 2.

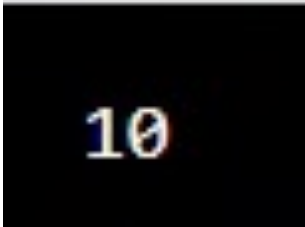


ข้อที่ 3 จงอธิบายเรื่อง Pointer Functionยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pointer Function เรียกใช้ Function จะใช้ค่าที่ส่งมา            จ้าง Function ค่าในคีย์แบบไม่ได้ แล้วจะเจอน Function ที่ส่งมา Return            ค่ากลับไปที่ Function นั้น</p>	<pre> main.c 1 // Online C compiler to run C program online 2 #include &lt;stdio.h&gt; 3 int add(int x,int y ){ 4     return x + y; 5 } 6 int operation( int x, int y, int (*function) (int,int) ){ 7     return (*function)(x,y); 8 } 9 int main(){ 10    printf("= %d\n",operation(25,15,add) ); 11    return 0; 12 }           </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	


ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p><i>Pointer → Array → Dynamic Array</i></p> <p><i>เมื่อใช้ Array ในคอมไพเลอร์จะทำการจองพื้นที่ที่จริงเกิดขึ้นที่หน่วย Dynamic Array นั่นเอง</i></p>	 <pre> main.c 1  #include &lt;stdio.h&gt; 2 3  int main() 4  { 5      int row,col,i,j; 6      printf("How many rows and cols?:"); 7      scanf("%d %d",&amp;row,&amp;col); 8      int a[row*col]; 9      for( i = 0; i &lt; row; i++) 10         for(j = 0; j &lt; col; j++) { 11             printf(" a[%d][%d] : ",i,j); 12             scanf("%d ", &amp;a[i*col+j]); 13         } 14         printf("\n"); 15     return 0; 16 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <pre> Output /tmp/GhIS9h0wbU.o How many rows and cols?: 1 4 a[0][0] : 9 a[0][1] : 7 a[0][2] : 6 a[0][3] : 5 4 dash: 2: 4: not found </pre>	

# Flow Chart 4



ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pass by reference. เป็นการส่งค่าผ่านทาง Argument ของ Function นั้นๆ ในกรณีส่ง โดยไม่มีการ Return ค่า</p>	<pre> #include &lt;iostream&gt; using namespace std;  void swapNums(int &amp;x, int &amp;y) {     int z = x;     x = y;     y = z; }  int main() {     int firstNum = 22;     int secondNum = 20;      cout &lt;&lt; "Before swap: " &lt;&lt; "\n";     cout &lt;&lt; firstNum &lt;&lt; secondNum &lt;&lt; "\n";      swapNums(firstNum, secondNum);      cout &lt;&lt; "After swap: " &lt;&lt; "\n";     cout &lt;&lt; firstNum &lt;&lt; secondNum &lt;&lt; "\n";      return 0; } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD     subgraph MainFlow [Main Flow]         Start([start]) --&gt; XY[x,y]         XY --&gt; Swap[swapNums]         Swap --&gt; Stop1([stop])     end      subgraph FunctionFlow [Function Flow: swapNums]         Start2([swapNums]) --&gt; Input[/input firstNum, secondNum/]         Input --&gt; Output[/output After swap/]         Output --&gt; Stop2([stop])     end </pre>



ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pass by value จะมีการส่งค่าตัวแปรกลับโดยไม่ได้ return โดยจะใช้ Function ช่วย ส่งค่ากลับหา Function main</p>	<pre> --- #include &lt;iostream&gt; --- using namespace std; --- --- int myFunction(int x, int y) { ---     return x + y; --- } --- --- int main() { ---     cout &lt;&lt; myFunction(5, 3); ---     return 0; --- } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD     subgraph "main Function"         start([start]) --&gt; xy[x,y]         xy --&gt; myFuncCall[myFunction]         myFuncCall --&gt; stop([stop])     end     subgraph "myFunction"         myFuncStart([myFunction]) --&gt; input[/input num1,num2/]         input --&gt; sum[num1+num2]         sum --&gt; return([return num])     end </pre>