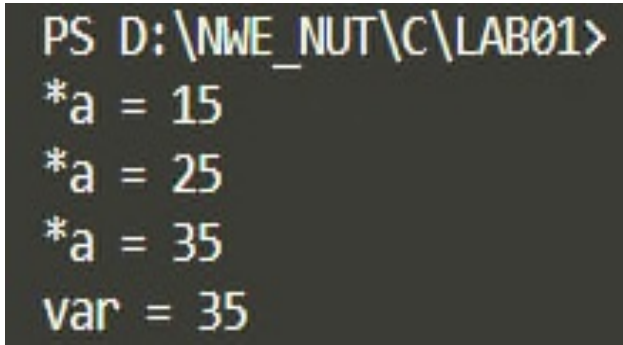
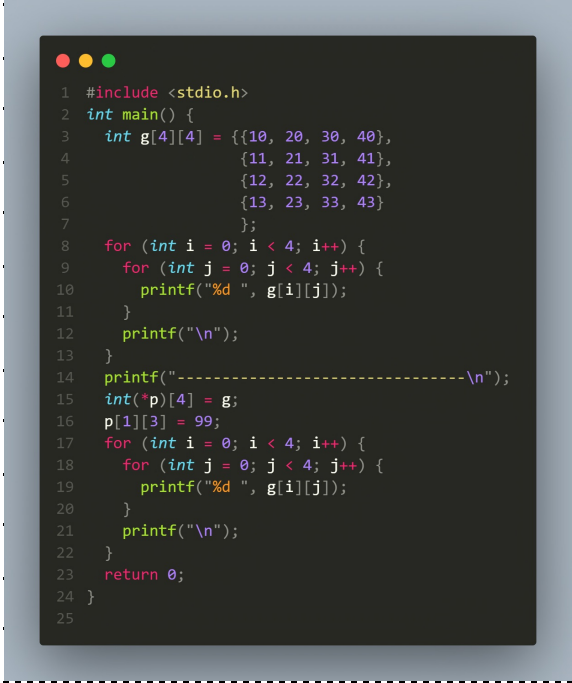
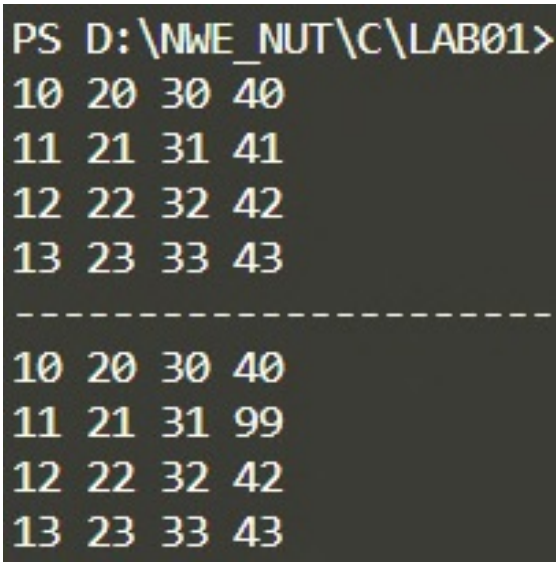
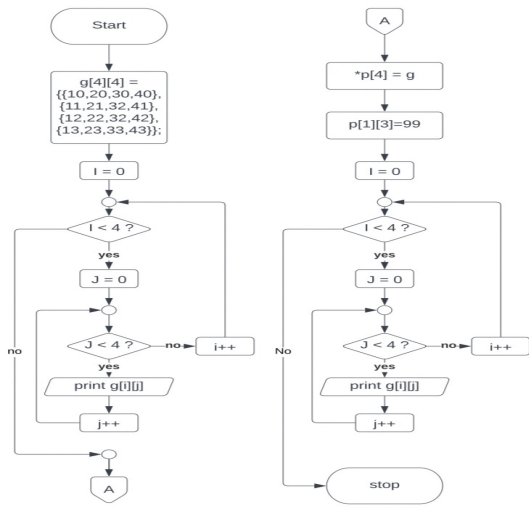


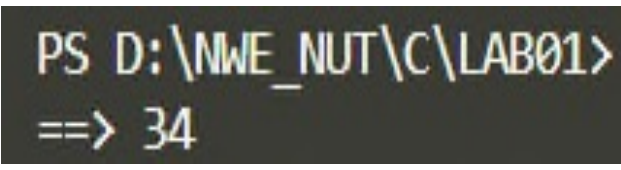
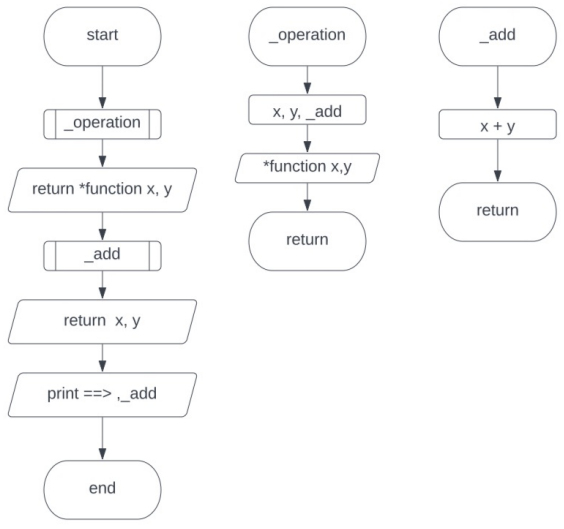
## ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

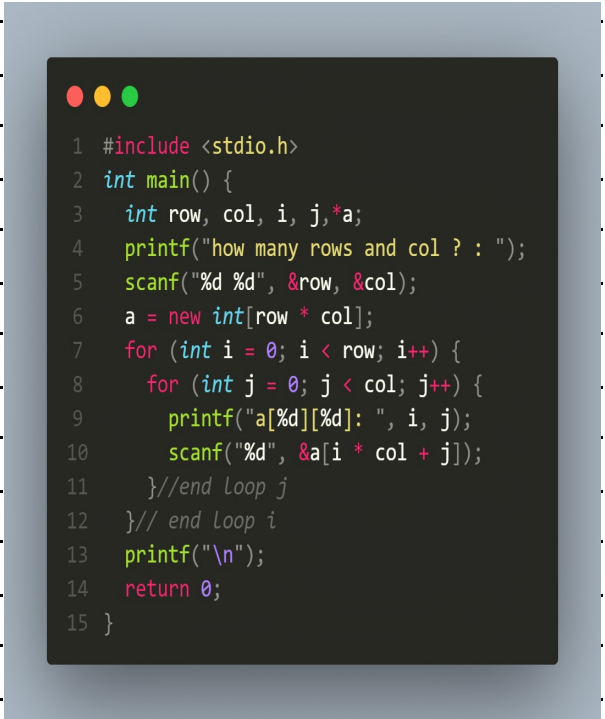
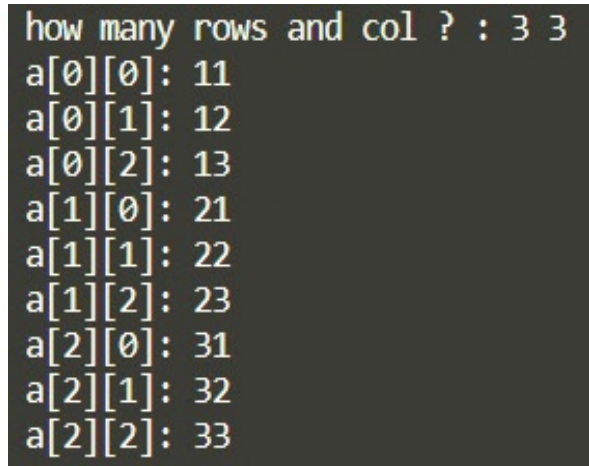
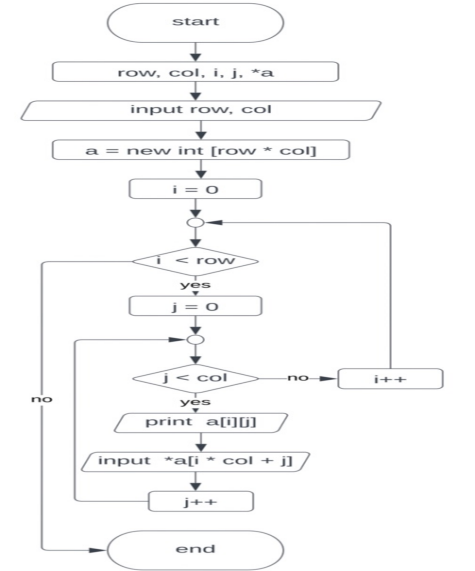
ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง								
<p>ใช้ Pointer ในการชี้ที่อยู่ของตัวแปร เลขที่ชี้ให้ถึง ข้อมูลได้เร็ว กว้างขวาง เมื่อจะทำการใช้ pointer ให้ใส่เครื่องหมาย * ไว้หน้า ตัวแปร จะทำให้ตัวแปรนั้นเป็นชนิด pointer</p> <table border="1"> <thead> <tr> <th>*a</th><th>a</th></tr> </thead> <tbody> <tr> <td>1001</td><td>26</td></tr> <tr> <td>1011</td><td>8</td></tr> <tr> <td>1100</td><td>65</td></tr> </tbody> </table> <p>Address value</p>	*a	a	1001	26	1011	8	1100	65	<pre> 1 #include &lt;stdio.h&gt; 2 int main() { 3     int *a; 4     int var = 15; 5     a = &amp;var; 6     printf("a = %d\n", *a); 7     var = 25; 8     printf("a = %d\n", *a); 9     *a = 35; 10    printf("a = %d\n", *a); 11    printf("var = %d\n", var); 12    return 0; 13 }</pre>
*a	a								
1001	26								
1011	8								
1100	65								
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง								
	<pre> graph TD     Start([Start]) --&gt; Init[*a, var = 0]     Init --&gt; Assign[a = &amp;var]     Assign --&gt; Print1[/OUTPUT *a/]     Print1 --&gt; ChangeVar[var = 25]     ChangeVar --&gt; Print2[/OUTPUT *a/]     Print2 --&gt; ChangeA[*a = 35]     ChangeA --&gt; Print3[/OUTPUT *a/]     Print3 --&gt; Print4[/OUTPUT var/]     Print4 --&gt; Stop([Stop])</pre>								

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pointer สามารถชี้ไปที่ Array ได้ เพื่อสร้าง Array แบบไม่จำกัดแถว และมีตัวชี้ข้อที่ชี้ค่า ต้องรู้ค่าหรือกำหนดค่าให้ ดอนลิมน์ทุกครั้ง เมื่อเริ่มสร้าง Pointer ในทางตอนที่ได้ชี้ Array ก่อน</p>	 <pre> 1 #include &lt;stdio.h&gt; 2 int main() { 3     int g[4][4] = {{10, 20, 30, 40}, 4                   {11, 21, 31, 41}, 5                   {12, 22, 32, 42}, 6                   {13, 23, 33, 43}}; 7 } 8 for (int i = 0; i &lt; 4; i++) { 9     for (int j = 0; j &lt; 4; j++) { 10        printf("%d ", g[i][j]); 11    } 12    printf("\n"); 13 } 14 printf("-----\n"); 15 int (*p)[4] = g; 16 p[1][3] = 99; 17 for (int i = 0; i &lt; 4; i++) { 18     for (int j = 0; j &lt; 4; j++) { 19        printf("%d ", g[i][j]); 20    } 21    printf("\n"); 22 } 23 return 0; 24 } 25 </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <pre> PS D:\NWE_NUT\C\LAB01&gt; 10 20 30 40 11 21 31 41 12 22 32 42 13 23 33 43 ----- 10 20 30 40 11 21 31 99 12 22 32 42 13 23 33 43 </pre>	 <pre> graph TD     Start([Start]) --&gt; Init1["g[4][4] = {{10,20,30,40}, {11,21,32,41}, {12,22,32,42}, {13,23,33,43}};"]     Init1 --&gt; I0["I = 0"]     I0 --&gt; I4{"I &lt; 4 ?"}     I4 -- yes --&gt; J0["J = 0"]     J0 --&gt; J4{"J &lt; 4 ?"}     J4 -- yes --&gt; Print1["print g[i][j]"]     Print1 --&gt; JInc["j++"]     JInc --&gt; J4     J4 -- no --&gt; IInc["i++"]     IInc --&gt; I4     I4 -- no --&gt; A1((A))     A1 --&gt; Init2["*p[4] = g"]     Init2 --&gt; P3["p[1][3]=99"]     P3 --&gt; I2["I = 0"]     I2 --&gt; I4_2{"I &lt; 4 ?"}     I4_2 -- yes --&gt; J2["J = 0"]     J2 --&gt; J4_2{"J &lt; 4 ?"}     J4_2 -- yes --&gt; Print2["print g[i][j]"]     Print2 --&gt; JInc2["j++"]     JInc2 --&gt; J4_2     J4_2 -- no --&gt; IInc2["i++"]     IInc2 --&gt; I4_2     I4_2 -- no --&gt; Stop([stop]) </pre>

ข้อที่ 3 จงอธิบายเรื่อง Pointer Functionยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>pointer function. ใช้ในการส่งค่าจาก function หนึ่งไปยังอีก function หนึ่ง โดยที่ function แรก จะรับค่ามาจาก main function แล้วจะส่งค่าจาก function ใหม่ออกมา</p> <p>โปรแกรมนี้ จะทำการส่งค่ากลับออกไป</p>	<pre> 1 #include &lt;stdio.h&gt; 2 int _add(int x, int y) { 3     return x + y; 4 } 5 int _operation(int x, int y, int (*function)(int, int)){ 6     return (*function)(x, y); 7 } 8 int main(){ 9     printf("==&gt; %d\n", _operation(26, 8, _add)); 10    return 0; 11 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD     Start([start]) --&gt; Op1[_operation]     Op1 --&gt; Ret1[/return *function x, y/]     Ret1 --&gt; Add1[_add]     Add1 --&gt; Ret2[/return x, y/]     Ret2 --&gt; Print[/print ==&gt; _add/]     Print --&gt; End([end])      Op2[_operation] --&gt; Arg2[x, y, _add]     Arg2 --&gt; Fnc2[/*function x,y/]     Fnc2 --&gt; Ret3([return])      Add2[_add] --&gt; Sum2[x + y]     Sum2 --&gt; Ret4([return]) </pre>

ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>Dynamic Array คือ Array ที่ไม่จำกัดขนาด เราสามารถเพิ่มพื้นที่ได้เรื่อยๆ เราจึงใช้ Pointer มาทำเป็น Array เราจึงเรียกมันว่า Dynamic Array</p>	 <pre> 1 #include &lt;stdio.h&gt; 2 int main() { 3     int row, col, i, j, *a; 4     printf("how many rows and col ? : "); 5     scanf("%d %d", &amp;row, &amp;col); 6     a = new int[row * col]; 7     for (int i = 0; i &lt; row; i++) { 8         for (int j = 0; j &lt; col; j++) { 9             printf("a[%d][%d]: ", i, j); 10            scanf("%d", &amp;a[i * col + j]); 11        } //end loop j 12    } // end loop i 13    printf("\n"); 14    return 0; 15 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <pre> how many rows and col ? : 3 3 a[0][0]: 11 a[0][1]: 12 a[0][2]: 13 a[1][0]: 21 a[1][1]: 22 a[1][2]: 23 a[2][0]: 31 a[2][1]: 32 a[2][2]: 33 </pre>	 <pre> graph TD     Start([start]) --&gt; Decl[decl row, col, i, j, *a]     Decl --&gt; Input[input row, col]     Input --&gt; Alloc[a = new int [row * col]]     Alloc --&gt; Init[i = 0]     Init --&gt; LoopI{ }     LoopI --&gt; CondI{i &lt; row}     CondI -- yes --&gt; InitJ[j = 0]     CondI -- no --&gt; IncI[i++]     InitJ --&gt; LoopJ{ }     LoopJ --&gt; CondJ{j &lt; col}     CondJ -- yes --&gt; Print[print a[i][j]]     Print --&gt; InputElem[input *a[i * col + j]]     InputElem --&gt; IncJ[j++]     CondJ -- no --&gt; IncI     IncJ --&gt; LoopJ     IncI --&gt; LoopI     LoopJ --&gt; End([end]) </pre>

ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย

Pass by reference จะเป็นการส่งค่าและรับค่าผ่านทง Argument เช่น ใน function main จะเรียกใช้ function ช่อง 1 และจะใช้ Argument ของ function นั้นๆ ในการรับส่งค่า ในของ function ช่อง 2 จะไม่ส่งค่า Return

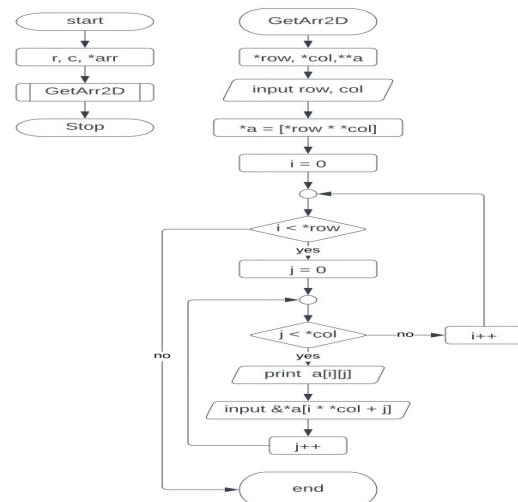
Code ตัวอย่าง

```
1 #include <stdio.h>
2 void GetArr2D(int **a, int *row, int *col) {
3     printf("How many rows and columns ? : ");
4     scanf("%d %d", row, col);
5     *a = new int[*row * *col];
6     for (int i = 0; i < *row; i++) {
7         for (int j = 0; j < *col; j++) {
8             printf("a[%d][%d]: ", i, j);
9             scanf("%d", &(*a)[i * *col + j]);
10        } // end Loop j
11    } // end Loop i
12 } // end Getarr2d
13 void GetArr2D(int **a, int *row, int *col); // prototype
14 int main() {
15     int r, c, *arr;
16     GetArr2D(&arr, &r, &c);
17     return 0;
18 }
```

ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)

```
PS D:\NWE_NUT\C> cd "d:\NWE_NUT\C\LAB01\"
How many rows and columns ? : 3 4
a[0][0]: 11
a[0][1]: 12
a[0][2]: 13
a[0][3]: 14
a[1][0]: 21
a[1][1]: 22
a[1][2]: 23
a[1][3]: 24
a[2][0]: 31
a[2][1]: 32
a[2][2]: 33
a[2][3]: 34
```

Flow chart ของ Code ตัวอย่าง



ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pass by value คือการส่งค่าตัวแปรหรือค่าที่เก็บไว้ในหน่วยความจำไปให้ Function ใช้ เช่น Function main จะเรียกใช้ Function ชื่อว่า Function ชื่อจะส่งค่ากลับคืนมา Function main จะใช้ Return</p>	<pre> 1 #include &lt;stdio.h&gt; 2 3 int *GetArr2D(int *row, int *col) { 4     printf("How many row and col ? : "); 5     scanf("%d %d", row, col); 6     int *arr; 7     arr = new int[*row * *col]; 8     for (int i = 0; i &lt; *row; i++) { 9         for (int j = 0; j &lt; *col; j++) { 10             printf("a[%d][%d]: = ", i, j); 11             scanf("%d", &amp;arr[i * *col + j]); 12         } // end Loop j 13     } // end Loop i 14     return arr; 15 } // end Getarr2d 16 17 int *GetArr2D(int *row, int *col); // prototype 18 19 int main() { 20     int r, c, *arr; 21     arr = GetArr2D(&amp;r, &amp;c); 22     delete[] arr; 23     return 0; 24 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง