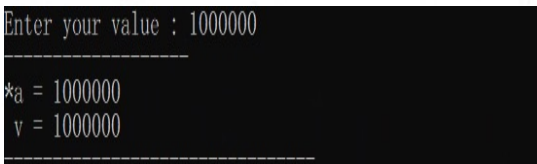
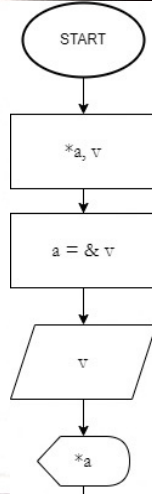


## ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง
<p>Pointer คือตัวชี้ที่เก็บค่าที่อยู่หน่วยความจำที่ไม่ใช่ค่าในหน่วยความจำ Dynamic data Structure หรือตัวชี้ที่เก็บค่าของตัวแปรอื่น</p> <p>โดยตัวชี้จะเก็บค่าที่อยู่หน่วยความจำของตัวแปรอื่น pointer นี้จะเก็บค่าของตัวแปรอื่นที่เก็บค่าของตัวแปรอื่น และเก็บค่าของตัวแปรอื่นที่เก็บค่าของตัวแปรอื่น</p> <p>pointer จะเก็บค่าของตัวแปรอื่นที่เก็บค่าของตัวแปรอื่น</p>	<pre>#include &lt;stdio.h&gt; int main() {     int* p;     int* v;     p = &amp;v;     printf("Enter your value:");     scanf("%d", &amp;v);      printf("---- OUTPUT ----\n");     printf("*p = %d\n", *p);     printf("v = %d", v);     return 0; }</pre>
ผลลัพธ์ของ Code (Capture พร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre>graph TD     START([START]) --&gt; Process1[*a, v]     Process1 --&gt; Process2[a = &amp;v]     Process2 --&gt; IO[v]     IO --&gt; Decision1{{*a}}     Decision1 --&gt; Decision2{{v}}     Decision2 --&gt; END([END])</pre>

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>จะสร้าง array 2 มิติ แถวละ 4 คอลัมน์ และใช้ for loop ชี้อันไหนให้ print ในแถวและคอลัมน์ใน array 2 มิติ และจะเอาตัวชี้ array ไม่จำกัดแถว แล้วใส่แถว (*n)[5]=c โดยค่า c ไม่เกิน 100 แล้วจะเอาตัวชี้ array ไม่จำกัดแถวใน array 2 มิติ ให้ for loop print ค่าออกมาแถวละ 4 คอลัมน์ โดยจะใส่ (*n)[2] แล้วให้เอาตัวชี้แถวกำหนดแถวจะใส่ตัวชี้แถวที่ชี้แถวเปลี่ยนแถว</p>	<pre> 1 #include &lt;stdio.h&gt; 2 int main(){ 3     int a , b ; 4     int g[4][5]={ 5         { 10, 20, 30, 40, 50 }, 6         { 11, 21, 31, 41, 51 }, 7         { 12, 22, 32, 42, 52 }, 8         { 13, 23, 33, 43, 53 } 9     }; 10    for (a = 0 ; a &lt; 4 ; a++){ 11        for(b=0 ; b &lt; 5 ; b++){ 12            printf("%d ", g[a][b]); 13            printf("\n"); 14        } 15    printf("-----\n"); 16    int (*n)[5] = g; 17    n[3][3] = 69 ; 18    n[2][0] = 96 ; 19 20    for (a = 0 ; a &lt; 4 ; a++){ 21        for(b = 0 ; b &lt; 5 ; b++){ 22            printf("%d ", g[a][b]); 23            printf("\n"); 24        } 25    } 26    return 0; 27                 </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
<div> <div> 10 20 30 40 50 11 21 31 41 51 12 22 32 42 52 13 23 33 43 53 </div> <div> 10 20 30 40 50 11 21 31 41 51 96 22 32 42 52 13 23 33 69 53 </div> </div>	<pre> graph TD     Start([start]) --&gt; Init[a, b g[4][5]={ { 10, 20, 30, 40, 50 }, { 11, 21, 31, 41, 51 }, { 12, 22, 32, 42, 52 }, { 13, 23, 33, 43, 53 } }]     Init --&gt; LoopA{a &lt; 4}     LoopA -- N --&gt; PrintG[("n)[5] = g n[3][3] = 69 n[2][0] = 96 }]     LoopA -- Y --&gt; LoopB{b &lt; 5}     LoopB -- N --&gt; LoopA     LoopB -- Y --&gt; IncB[b+1]     IncB --&gt; LoopB     LoopB --&gt; IncA[a+1]     IncA --&gt; PrintG     PrintG --&gt; LoopA     LoopA --&gt; LoopA2{a &lt; 4}     LoopA2 -- N --&gt; End([End])     LoopA2 -- Y --&gt; LoopB2{b &lt; 5}     LoopB2 -- N --&gt; LoopA2     LoopB2 -- Y --&gt; IncB2[b+1]     IncB2 --&gt; LoopB2     LoopB2 --&gt; IncA2[a+1]     IncA2 --&gt; LoopA2     </pre>



ข้อที่ 3 จงอธิบายเรื่อง Pointer Function ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>เมื่อส่งค่าตัวแปรไปให้ฟังก์ชันที่รับค่าเป็น pointer เราจะได้ค่าที่ส่งกลับมาจากฟังก์ชันนั้นมาใช้งานต่อได้</p>	<pre>#include &lt;stdio.h&gt; int result(int a, int b, int c) {     return a + b * c; } int num(int a, int b, int c, int (*func)(int, int, int)) {     return (*func)(a, b, c); } int main() {     printf("output = %d", num(     10, 20, 30, result));     return 0; }</pre>

ผลลัพธ์ของ Code (Capture พร้อมแปะรูป)      Flow chart ของ Code ตัวอย่าง

```

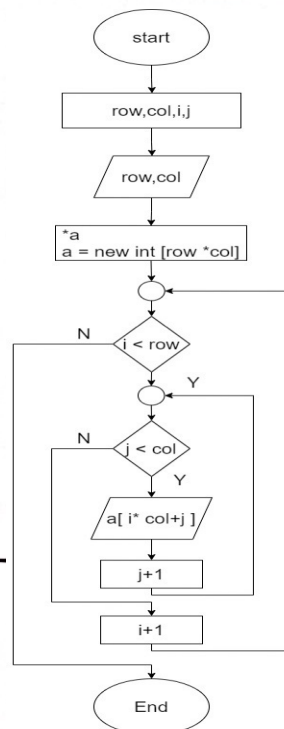
graph TD
    Start([start]) --> Num[num(10,20,30,result)]
    Num --> Num2[num(a,b,c,(*func))]
    Num2 --> Result[result(a,b,c)]
    Result --> End([End])
    
```

คำอธิบาย

**Code ตัวอย่าง**

**ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)**

Flow chart ของ Code ตัวอย่าง





ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย

Pass by reference เป็นการส่งค่าของ Argument หรือค่าที่ส่งมาให้กับฟังก์ชัน โดยที่ค่าที่ส่งมาจะเก็บไว้ใน pointer ของตัวแปรในฟังก์ชัน และเมื่อฟังก์ชันทำงานเสร็จแล้ว ค่าที่เก็บไว้ใน pointer นั้นจะส่งกลับไปยังตัวแปรต้นฉบับ

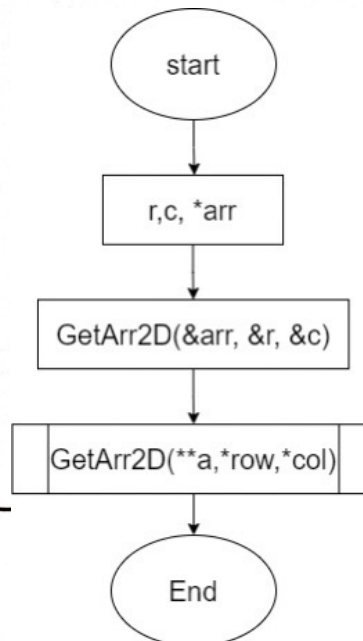
Code ตัวอย่าง

```
1 #include <stdio.h>
2
3 void GetArr2D(int **a, int *row, int *col);
4
5 int main(){
6     int r,c,*arr;
7     GetArr2D(&arr,&r,&c);
8     return 0;
9 }
10
11 void GetArr2D(int **a, int *row, int *col){
12     int i,j;
13     printf
14     ("INPUT : ");
15     scanf("%d %d", row, col);
16     *a = new int[*row * *col];
17     for( i = 0 ; i < *row ; i++){
18         for( j = 0 ; j < *col ; j++){
19             printf("a[%d][%d] = ", i, j);
20             scanf("%d", &(*a)[i* *col + j]);
21         }
22     }
23 }
```

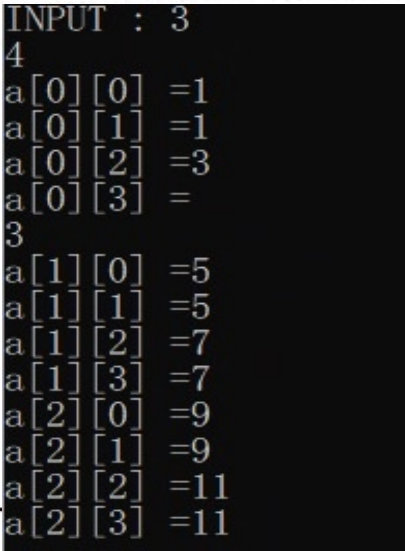
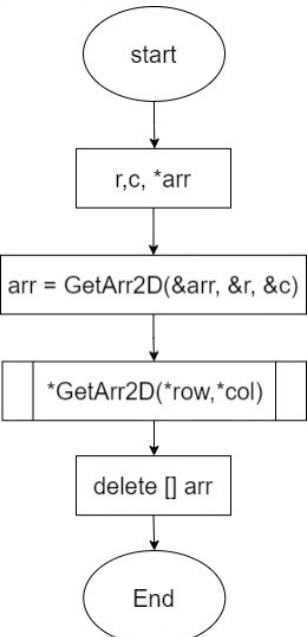
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)

```
INPUT : 2
2
a[0][0] = 1
a[0][1] = 1
a[1][0] = 2
a[1][1] = 2
```

Flow chart ของ Code ตัวอย่าง



ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>ตัวอย่างการส่งผ่านค่าได้ 2 แบบ Pass by Value 1 แบบ Pass by Reference 2 แบบ</p>	<pre> 1  #include &lt;stdio.h&gt; 2  int *GetArr2D(int *row, int *col); 3  int main(){ 4      int r, c, *arr; 5      arr = GetArr2D(&amp;r,&amp;c); 6      delete [] arr; 7      return 0; 8  } 9 10 int *GetArr2D(int *row, int *col){ 11     int *a; 12     int i, j; 13     printf("INPUT : " ); 14     scanf("%d %d", row,col); 15     a = new int[*row * *col]; 16     for (i = 0 ; i &lt; *row ; i++){ 17         for (j = 0 ; j &lt; *col ; j++){ 18             printf("a[%d][%d] = ", i ,j); 19             scanf("%d", &amp;a[i * *col + j]); 20         } 21     } 22     return a;                 </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD     Start([start]) --&gt; Init[r,c, *arr]     Init --&gt; Call[arr = GetArr2D(&amp;r, &amp;c)]     Call --&gt; Func[*GetArr2D(*row,*col)]     Func --&gt; Delete[delete [] arr]     Delete --&gt; End([End])                 </pre>