

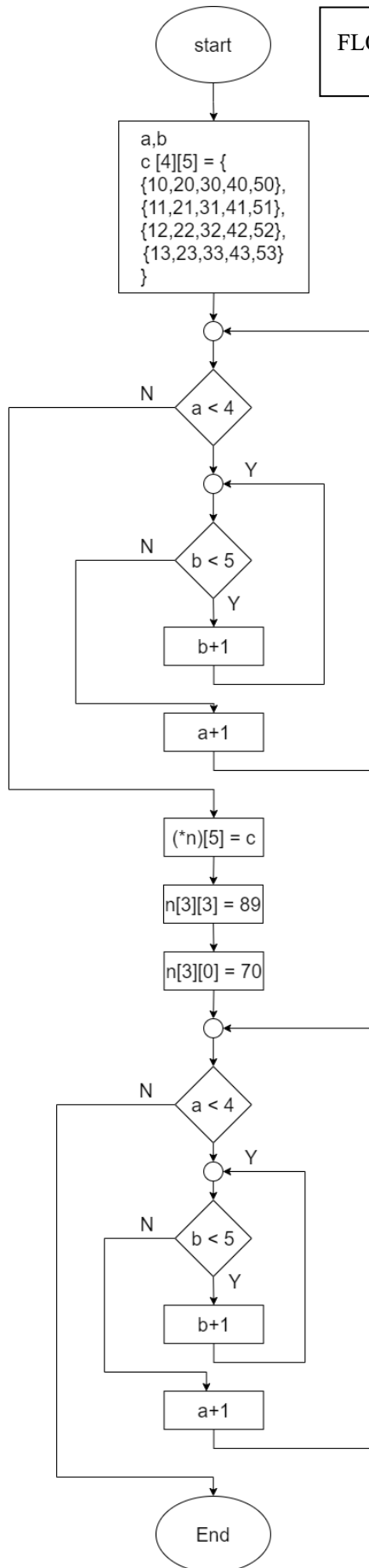
ผลลัพธ์ 1

```
C:\Users\phupa\Desktop\Computer Programming\Lab1\input1.exe
Enter your value :8
----OUTPUT----
*p = 8
v = 8
-----
Process exited after 3.385 seconds with return value 0
Press any key to continue . . .
```

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>ขอสร้าง array 2 มิติแบบไม่คงที่ และกำหนดค่าไว้</p> <p>แล้วใช้ for loop ใช้วนทั้งในแถว printf ทดสอบว่า pointer ใน array 2 มิติ แล้ว print แล้ว array ไม่จำกัด</p> <p>แถว แถวแรก (*p)[5] = 0; เราคิดว่า 0 ไม่พิมพ์</p> <p>ที่ (*p) เก็บจำนวน แล้วจะ print แล้วใน array 2 มิติ</p> <p>ใช้ for loop printf ค่าออกมาเรื่อยๆ โดย</p> <p>เราใช้ C[a][b] แสดงแถว a และ คอลัมน์ b</p> <p>ค่าที่เราใช้คือ 0 ถึง 10 เรียงตามแถว</p>	<pre>#include <stdio.h> int main() { int a, b; int c[4][5] = { {10, 20, 30, 40, 50}, {1, 21, 31, 41, 51}, {12, 22, 32, 42, 52}, {13, 23, 33, 43, 53} }; for (a = 0; a < 4; a++) { for (b = 0; b < 5; b++) { printf("%d ", c[a][b]); } printf("\n"); } }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart และ Code ตัวอย่าง
	<pre>printf("\n-----\n"); int (*p)[5] = 0; p[3][3] = 89; p[3][0] = 70; for (a = 0; a < 4; a++) { for (b = 0; b < 5; b++) { printf("%d ", c[a][b]); } printf("\n"); } return 0;</pre>

Flow chart และ code ตัวอย่าง

FLOW CHART 2



ผลลัพธ์ 2

```
Select C:\Users\phupa\Desktop\Computer Programming\Lab1\input2.exe
1020304050
1121314151
1222324252
1323334353

-----
1020304050
1121314151
1222324252
7023338953

-----
Process exited after 0.0535 seconds with return value 0
Press any key to continue . . .
```

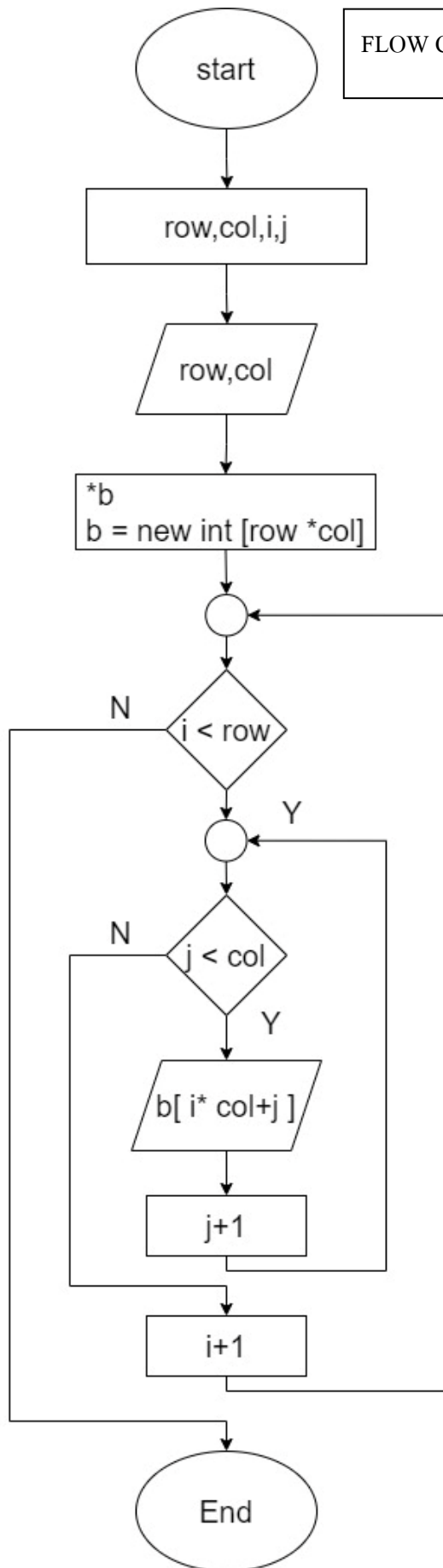

ข้อที่ 3 จงอธิบายเรื่อง Pointer Function ตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>เริ่มต้นด้วยตัวชี้ของตัวชี้ก่อนแล้วค่อยๆไป ดูฟังก์ชันของ Pointer แล้วใช้ตัวชี้ไปชี้ที่ตัวชี้ แล้วฟังก์ชันก็ใช้ตัวชี้ไปชี้ แล้วจึงค่อยๆไป ดูจนจบ ฟังก์ชันก็จะชี้ไปที่ตัวชี้ที่เราไปชี้ไว้ จนจบ ฟังก์ชันก็จะชี้ไปที่ตัวชี้ที่เราไปชี้ไว้</p>	<pre>#include <stdio.h> 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 (5, 3, 9, result)); return 0; }</pre>
ผลลัพธ์ของ Code (Capture หรือแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD START([START]) --> Num1[num (5, 3, 9, result)] Num1 --> Num2[num (a, b, c, *func)] Num2 --> Result[result (a, b, c)] Result --> END([END]) </pre>

ผลลัพธ์3

```
C:\Users\phupa\Desktop\Computer Programming\Lab1\input3.exe
OUTPUT = 32
-----
Process exited after 0.07809 seconds with return value 0
Press any key to continue . . .
```

ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
เป็น อาร์เรย์ 1 มิติ ที่ใช้จัดเก็บข้อมูลของตัวแปร ที่ตอนแรกเรา กำหนดให้ขนาดเป็น 0 คือขนาดที่มาก แต่สามารถใส่ค่าได้เรื่อยๆ 10 อันดับแรก แล้วเราสามารถเพิ่มขนาดให้เรื่อยๆ เราเพิ่มค่าเรื่อยๆ อาร์เรย์ จะปรับขนาด ให้เหมาะสมกับที่เราใส่ค่าเพิ่ม ให้ขนาดตัว แปร ที่ถูกใส่ค่าอยู่ ให้เป็นขนาดนั้น จะเพิ่มให้เท่าไรก็ได้ เราใส่ ทวีคูณเพิ่มเรื่อยๆ แล้วขนาดที่เราเพิ่ม ขึ้นอยู่กับข้อที่เรา ต้องการ	<pre>#include <stdio.h> int main() { int row, col, i, j; printf("Input:"); scanf("%d %d", &row, &col); int *b; b = new int [row * col]; for (i = 0; i < row; i++) for (j = 0; j < col; j++) { printf("a [%d] [%d] : ", i, j); scanf("%d", &b[i * col + j]); } return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง

FLOW CHART 4



ผลลัพธ์4

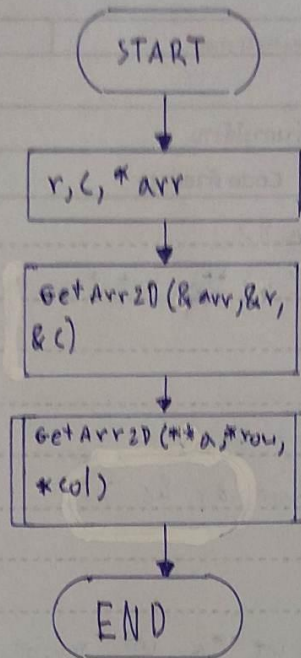
```
C:\Users\phupa\Desktop\Computer Programming\Lab1\input 4.exe
INPUT :2 3
a[0][0] : 2
a[0][1] : 4
a[0][2] : 3
a[1][0] : 5
a[1][1] : 4
a[1][2] : 1

-----
Process exited after 23.99 seconds with return value 0
Press any key to continue . . .
```

ข้อที่ 5 จงอธิบายการส่งผ่านค่าแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pass by Reference โดยส่งค่าที่อยู่ที่ Argument และ Argument ที่ถูกป้อนด้วย จะไม่ได้อะไร array เป็น Pointer ดังนั้น array ของเรา เป็น Pointer ของ ตัวแปรที่เราจะป้อนข้อมูลของเราเอง มีค่าที่เก็บอยู่ 7024</p>	<pre>#include <stdio.h> void GetArr2D (int **a, int *row, int *col); int main () { int r, c, *arr; GetArr2D (&arr, &r, &c); return 0; } void GetArr2D (int **a, int *row, int *col) { int i, j; printf ("How many row, col? "); scanf ("%d %d", row, col); *a = new int [*row * *col]; for (i = 0; i < *row; i++) for (j = 0; j < *col; j++) {</pre>
<p>ผลลัพธ์ของ Code (Captureพร้อมแป้นพิมพ์)</p>	<p style="text-align: center;">Flow chart and Code ตัวอย่าง</p> <pre>printf ("a[%d][%d] = ", i, j); scanf ("%d", &(*a)[i * *col + j]); } }</pre>

Flow chart →

FLOW CHART 5



ผลลัพธ์ 5

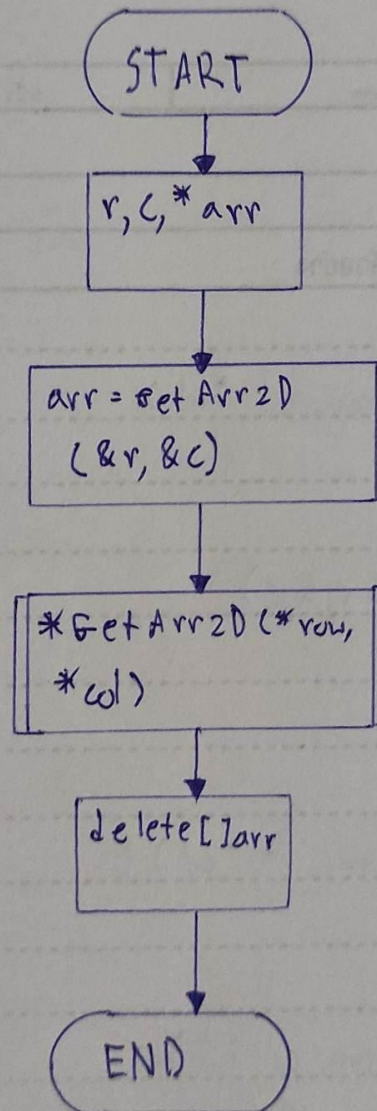
```
C:\Users\phupa\Desktop\Computer Programming\Lab1\input 5.exe
How many row, col ? 2 3
a[0][0] = 1
a[0][1] = 5
a[0][2] = 4
a[1][0] = 3
a[1][1] = 2
a[1][2] = 6

-----
Process exited after 10.97 seconds with return value 0
Press any key to continue . . .
```

ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pass by Value การส่งผ่านค่าตัวแปรแบบ Pass by Value นั้นจะ by Reference 2 ตัว</p> <p>Argument Return</p> <p>col ←</p> <p>row ←</p> <p>Get Array 2D</p> <p>→ a[i]</p>	<pre> #include <stdio.h> int *GetArr2D(int *row, int *col); int main() { int r, c; arr = GetArr2D(&r, &c); delete [] arr; return 0; } int *GetArr2D(int *row, int *col) { int *a; int i, j; printf("How many row, col? "); scanf("%d %d", row, col); a = new int[*row * *col]; for(i = 0; i < *row; i++) for(j = 0; j < *col; j++) { printf("a[%d][%d] = ", i, j); scanf("%d", &a[i * *col + j]); } return a; } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> for(i = 0; i < *col; i++) { printf("a[%d][%d] = ", i, j); scanf("%d", &a[i * *col + j]); } return a; } </pre>

Flow chart →

FLOW CHART 6



ผลลัพธ์ 6

```
C:\Users\phupa\Desktop\Computer Programming\Lab1\input 6.exe
How many row, col ? 2 3
a[0][0] = 2
a[0][1] = 4
a[0][2] = 5
a[1][0] = 6
a[1][1] = 1
a[1][2] = 3

-----
Process exited after 9.551 seconds with return value 0
Press any key to continue . . .
```