
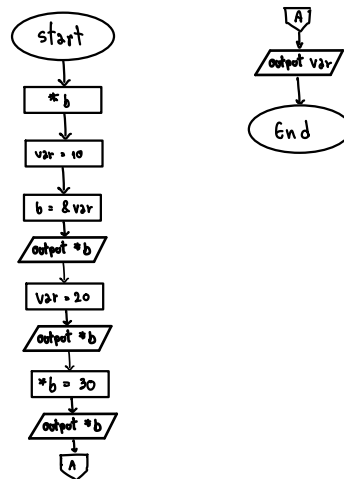


ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

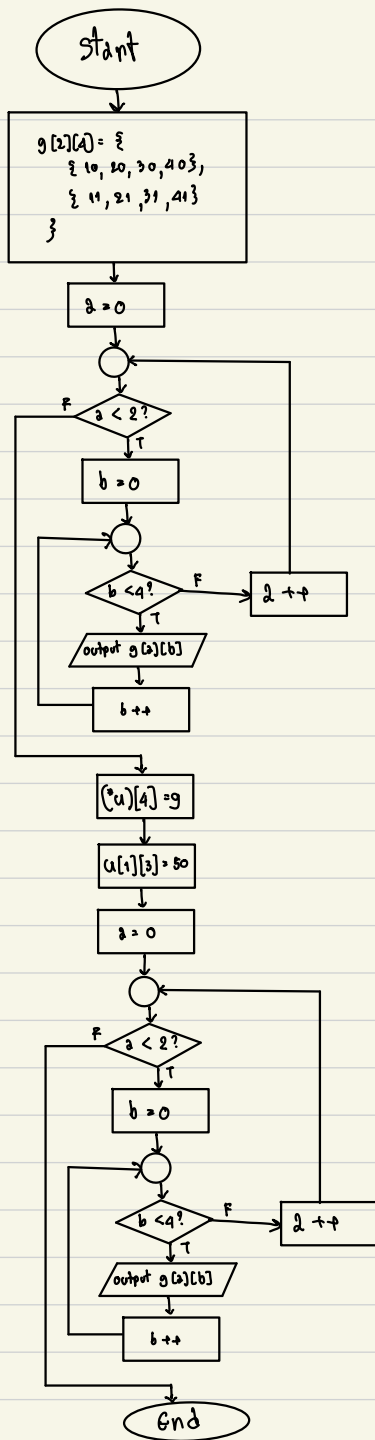
จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

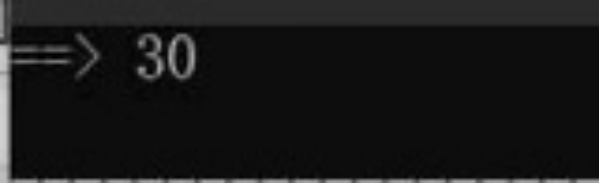
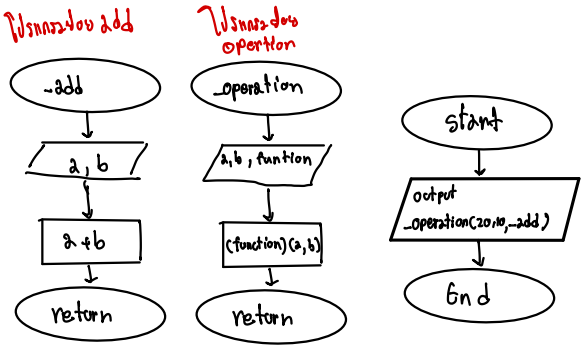
ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer	
คำอธิบาย	Code ตัวอย่าง
<p>Pointer เปลี่ยนเลขชี้อนันต์ เพราะเมื่อเราประกาศ Pointer 2 ตัว ตอนแรกจะชี้ใช้งานอะไรไม่ได้ จะใช้งานได้ก็ต่อเมื่อมันชี้ไปยังตัวแปรอื่นก่อนเท่านั้น ถ้าหากเราเขียน *a มันชี้มันยังไม่ถึงค่าไม่ได้ ตัวแปร var เป็นตัวเก็บค่าได้ ดังนั้นในโปรแกรมตัวชี้ *a จะไปชี้ var ก่อนถึงจะใช้งานได้</p>	<pre> #include <stdio.h> int main () { int *b ; int var = 10 ; b = &var ; printf ("*b = %d\n", *b); var = 20 ; printf ("*b = %d\n", *b); *b = 30 ; printf ("*b = %d\n", *b); printf ("var = %d\n", var); return 0; } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD start([start]) --> pb[*b] pb --> v10[var = 10] v10 --> b_ptr[b = &var] b_ptr --> o1[/output *b/] o1 --> v20[var = 20] v20 --> o2[/output *b/] o2 --> v30[*b = 30] v30 --> o3[/output *b/] o3 --> end([End]) </pre>

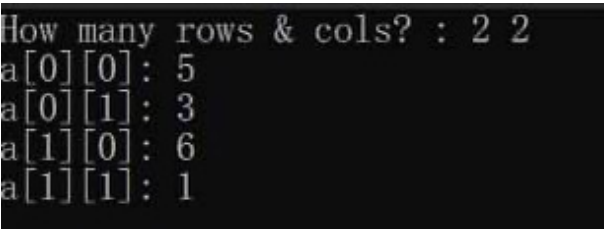
ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน				
คำอธิบาย		Code ตัวอย่าง		
เราใช้ pointer ชี้ Array เพื่อสร้าง Array แบบไม่จำกัดแถว เป็น Array 2 มิติ		#include <stdio.h>		
การประกาศใช้		int main () {		
int (*a)[4]; หรือ a = new int[n][4];		int g[2][4] = {		
ค่าที่ได้คือ		{ 10, 20, 30, 40 },		
		{ 11, 21, 31, 41 }		
		};		
		for (int a = 0; a < 2; a++) {		
		for (int b = 0; b < 4; b++) {		
		printf ("%d", g[a][b]);		
		printf ("\n");		
		}		
		printf ("~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~");		
		int (*u)[4] = g;		
		u[1][3] = 50;		
		for (int a = 0; a < 2; a++) {		
		for (int b = 0; b < 4; b++) {		
		printf ("%d", g[a][b]);		
		printf ("\n");		
		}		
		return 0;		
		}		
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)				
<div><div>10 20 30 40</div><div>11 21 31 41</div><div>10 20 30 40</div><div>11 21 31 50</div></div>				

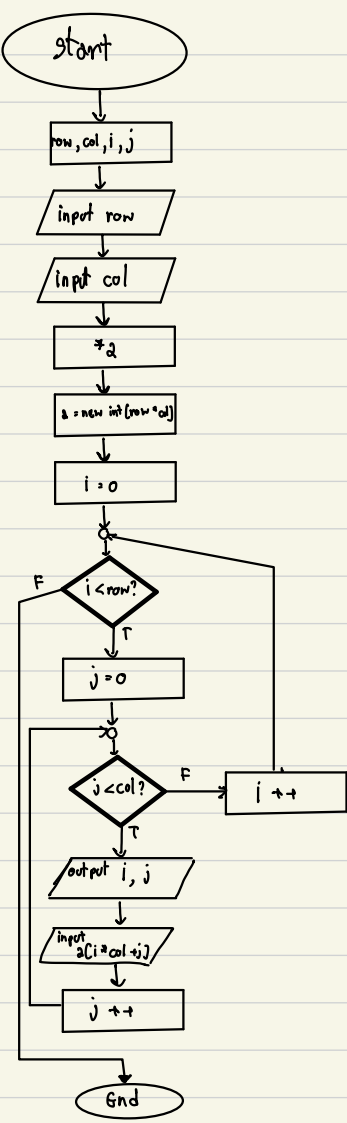
Flow chart อยู่อีกหน้าละ

q10 2

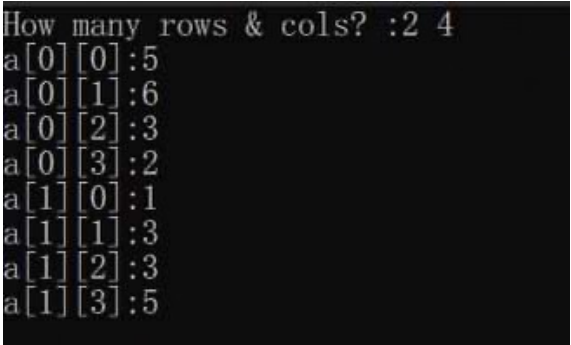


ข้อที่ 3 จงอธิบายเรื่อง Pointer Function ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>Pointer Function เป็น Function ที่มีโครงสร้างทั่วไป แต่ประเภท Function เป็น Pointer โดยต้องมีวงเล็บชี้ตรงไว้</p> <p>เช่น <code>return (*function)(a,b);</code></p> <p>ถ้าในกรณีส่งค่าจาก ฟังก์ชันส่งไปนาอีก ฟังก์ชันส่งโดยที่ส่งมาจาก ฟังก์ชัน ส่งตรงไปหา ฟังก์ชันส่งส่งไปบนมือ ฟังก์ชันส่งที่ส่งจะค่ามา</p> <p>กรณีใช้กับค่า ที่ได้มาจาก ฟังก์ชัน ส่งตรง หากส่ง ฟังก์ชัน ส่งที่ส่งจะ: รั่วรั่ว ส่งกลับไป</p>	<pre>#include <stdio.h> int _add(int a, int b) { return a + b; } int _operation(int a, int b, int (*function)(int, int)) { return (*function)(a, b); } int main() { printf("=> %d\n", _operation(20, 10, _add)); return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	

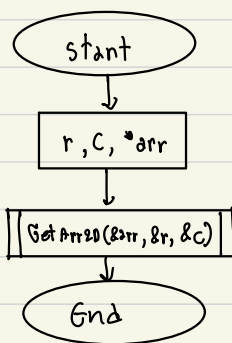
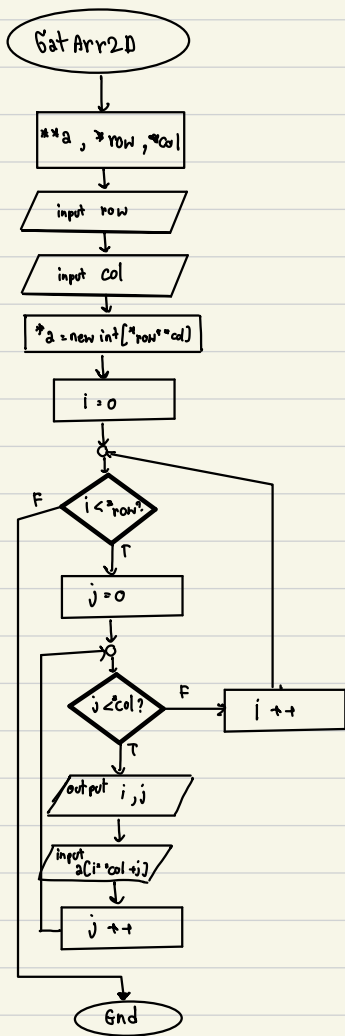
ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>เป็นคอมไพเลอร์ Array คือใช้สำหรับจัดเก็บข้อมูลที่เปลี่ยนแปลงจำนวนที่แน่นอน หรือ ทรง แต่มันคือ Dynamic Array เพื่อปรับขนาดจากบรรทัดต่อ บรรทัด</p> <p>คือโปรแกรม Array จะปรับขนาดจำนวนสมาชิกได้ เท่ากับ จำนวน ตัวเลขที่ คุณกำหนดไว้</p> <p>ในขณะนั้น ใช้ได้ สำหรับใช้กับทั้ง ระบบ เพื่อคอมไพเลอร์เท่านั้น</p> <p>ตัวอย่าง Dynamic Array คือ คุณใช้โปรแกรม จำนวน สมาชิกใน array</p>	<pre>#include <stdio.h> int main() { int row, col, i, j; printf("How many rows & cols? :"); scanf("%d %d", &row, &col); int *a; a = new int[row * col]; for (int i = 0; i < row; i++) { for (int j = 0; j < col; j++) { printf("a[%d][%d]: ", i, j); scanf("%d", &a[i * col + j]); } } return 0; }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	



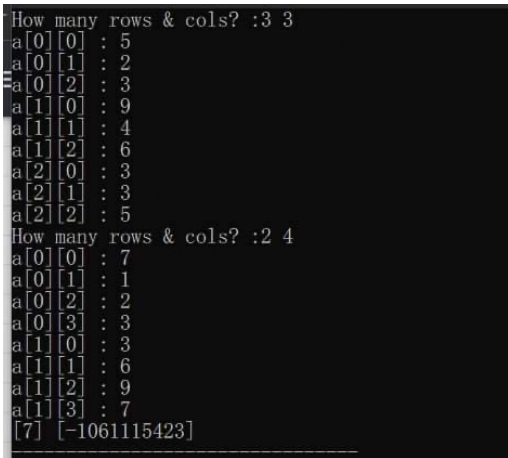
ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>pass by reference คือ การส่งค่าไปของพ้อยเตอร์ที่เก็บค่าไว้ โดยส่งเป็นค่าตำแหน่งที่อยู่ ของตัวแปรไป ซึ่งค่าของพ้อยเตอร์นั้นจะตรงกับค่าของ argument ที่ส่งไป ก็คือ ผลลัพธ์ที่ได้ค่าของ argument นั้นในโปรแกรม ที่ใช้เรียกใช้ฟังก์ชันนั่นเอง</p>	<pre>#include <stdio.h> void GetArr2D(int **a, int *row, int *col){ printf("How many rows & cols? :"); scanf("%d %d", &row & col); *a = new int[*row* *col]; for (int i=0; i<*row; i++) { for (int j=0; j<*col; j++) { printf("a[%d][%d]:", i, j); scanf("%d", &(*a)[i* *col + j]); } } }</pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	<pre>void GetArr2D(int **a, int *row, int *col) int main() { int r, c, *arr; GetArr2D(&arr, &r, &c); return 0; }</pre>
	

7/5/2020



ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน

คำอธิบาย	Code ตัวอย่าง
<p>pass by value คือ การส่งค่า (value) เป็น argument ของฟังก์ชัน ดังนั้นค่าที่แก้ไข ฟังก์ชันจึงไม่ส่งผลต่อตัวแปรของฟังก์ชัน</p>	<pre> #include <stdio.h> int *GetArr2D(int *row, int *col){ printf("How many rows & cols? :"); scanf("%d %d", row, col); int *a; a = new int[*row* *col]; for (int i = 0; i < *row; i++) { for (int j = 0; j < *col; j++) { printf("a[%d][%d] : ", i, j); scanf("%d", &a[i* *col + j]); } } return a; } </pre>
<p>ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)</p>	<pre> int *GetArr2D(int *row, int *col) int main() { int r, c, *arr; arr = GetArr2D(&r, &c); print("a[%d][%d] : ", GetArr2D(&r, &c)); delete[] arr; return 0; } </pre>
	

Q6

jsunssjow

