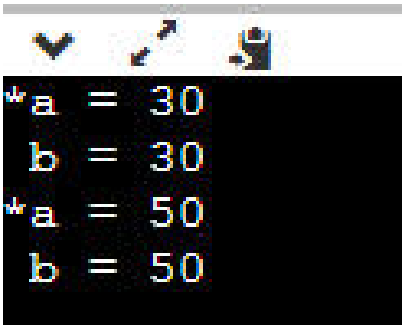
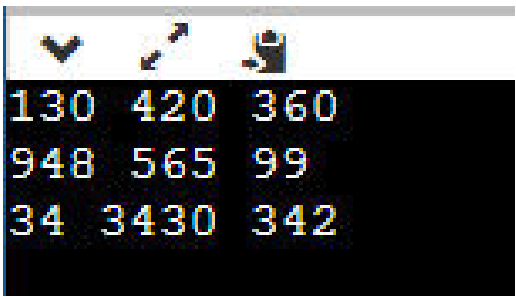
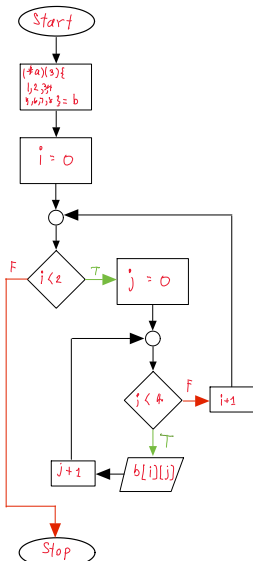


ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

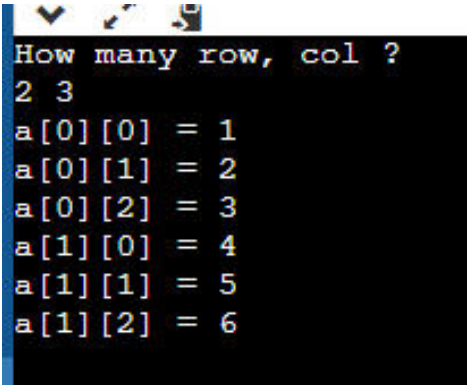
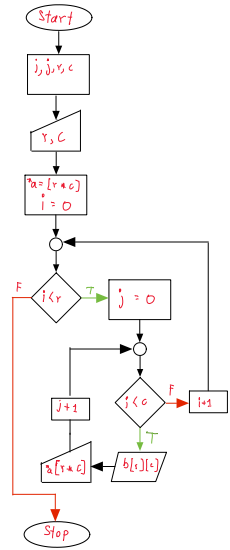
จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

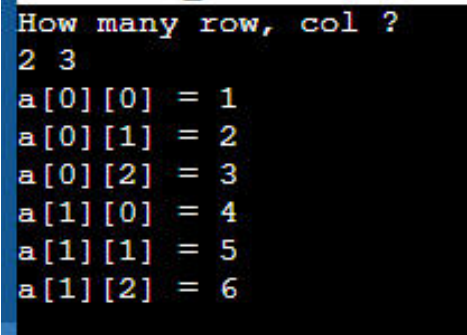
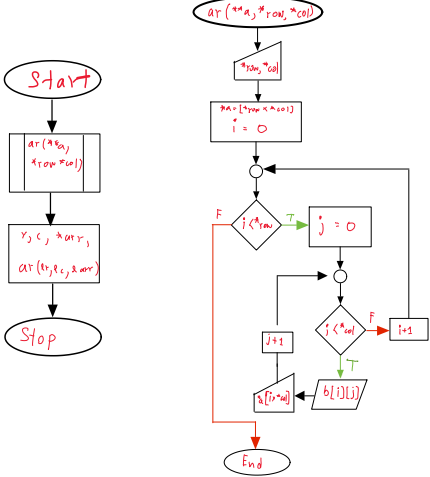
ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer

คำอธิบาย	Code ตัวอย่าง
<p>pointer แปลว่า ชี้ได้ไปใช้ได้</p> <p>pointer ต้นแบบจริง คือแปรที่เก็บค่าของตัวแปรอื่นไว้ที่ตัวแปรของตัวแปรนั้นอีกตัว</p> <p>แปรชนิดอื่น ๆ ของ pointer ชี้ไปยังตัวแปรที่ชี้ไปยังตัวแปรอื่น ๆ แทนที่จะเป็นตัวแปรจริง</p>	<pre> 1 #include <stdio.h> 2 3 int main() 4 { 5 int *a; 6 int b = 30; 7 a = &b; 8 printf("*a = %d\n", *a); 9 printf(" b = %d\n", b); 10 11 *a = 50; 12 13 printf("*a = %d\n", *a); 14 printf(" b = %d\n", b); 15 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD Start([Start]) --> Init[*, b] Init --> Assign[b = 30, *a = b] Assign --> Print1[/b, *a/] Print1 --> Assign2[*a = 50] Assign2 --> Print2[/b, *a/] Print2 --> Stop([Stop]) </pre>

ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>การชี้ pointer ไปที่ array ไม่จำกัดแถว แถวละ 4 Column</p> <p>Array. ฟังก์ชันสามารถใส่ค่าได้แค่ 4 ช่องแถว</p> <p>สามารถเพิ่มค่าเรื่อยๆ ตามแถวจนครบทุกแถวก็ได้</p> <p>ใส่ค่าให้ครบทุกแถวจนครบทุกแถวได้</p>	<pre> 1 #include <stdio.h> 2 3 int main() 4 { 5 int data[3][3] = { 6 {130, 420, 360}, 7 {948, 565, 340}, 8 {34, 3430, 342}, 9 }; 10 11 int (*j)[3] = data; 12 13 j[1][2] = 99; 14 15 for(int i = 0 ; i < 3 ; i++){ // start loop array [*j] 16 for(int j = 0 ; j < 3 ; j++){ // start loop array [3] 17 printf("%d ", data[i][j]); 18 } // start loop array [3] 19 printf("\n"); 20 } // start loop array [*j] 21 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	 <pre> graph TD Start([Start]) --> Init[(*j)[3] { j[2] = 99; }] Init --> i0[i = 0] i0 --> i3{i < 3} i3 -- T --> j0[j = 0] j0 --> j3{j < 3} j3 -- T --> Print[/b[i][j]/] Print --> jinc[j++] jinc --> j3 j3 -- F --> jinc1[j++] jinc1 --> iinc[i++] iinc --> i3 i3 -- F --> Stop([Stop]) </pre>

ข้อที่ 3 จงอธิบายเรื่อง Pointer Functionยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>ให้ function ด้สโ้ทวสร้าง แล้ function อื่นๆ ที่ไป แต่ จ. ด้. ด้. ด้. function ด้. ด้. ด้. Pointer</p>	<pre> 8 9 #include <stdio.h> 10 11 int op(int x, int y, int (*fun) (int, int)){ 12 return (*fun)(x, y); 13 } 14 15 int sum(int x, int y){ 16 return x + y; 17 } 18 19 int main() 20 { 21 printf("%d\n",op(5, 5, sum)); 22 23 return 0; 24 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	<pre> graph TD Start([Start]) --> Op[op] Op --> Sum[sum] Sum --> Main[main] Main --> Stop([Stop]) subgraph op_function [op] op_in([x, y, *fun]) --> op_calc[x + y] op_calc --> op_return([return]) end subgraph sum_function [sum] sum_in([x, y]) --> sum_calc[x + y] sum_calc --> sum_return([return]) end subgraph main_function [main] main_in([main]) --> main_op[op(5, 5)] main_op --> main_sum[sum] main_sum --> main_return([return]) end </pre>

ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>ให้เขียนฟังก์ชัน Array ที่สามารถขยายได้โดยตามขนาดที่ user ใช้งานให้ลงใหม่โดยขนาดของ Dynamic ที่ให้ Array นั้นให้ขนาดที่มากตัวแปรจะเพิ่มขึ้นทุกครั้งที่มีการเพิ่มค่าใหม่ จะมีการลบค่าของพื้นที่ที่ไม่สามารถนำมาใช้ได้อีกต่อไป</p>	<pre> main.cpp 1 #include <stdio.h> 2 3 int main() 4 { 5 int row, col, i, j; 6 printf("How many row, col ?\n"); 7 scanf("%d %d", &row, &col); // input max array[row][col] 8 9 int *a; 10 a = new int[row * col]; 11 12 for(int i = 0 ; i < row ; i++){ // start loop row 13 for(int j = 0 ; j < col ; j++){ // start loop col 14 printf("a[%d][%d] = ", i, j); 15 scanf("%d", &a[i * col + j]); // input cost position row = i, col = j 16 } 17 } 18 return 0; 19 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
 <pre> How many row, col ? 2 3 a[0][0] = 1 a[0][1] = 2 a[0][2] = 3 a[1][0] = 4 a[1][1] = 5 a[1][2] = 6 </pre>	 <pre> graph TD Start([Start]) --> Input[/i, j, c/] Input --> Init["a = [r*c] i = 0"] Init --> LoopStart(()) LoopStart --> Cond1{"i < r"} Cond1 -- F --> Stop([Stop]) Cond1 -- T --> LoopStart2(()) LoopStart2 --> Cond2{"j < c"} Cond2 -- F --> LoopStart Cond2 -- T --> Input2[/a[i*c+j]/] Input2 --> IncJ["j++"] IncJ --> Cond2 Cond2 -- F --> IncI["i++"] IncI --> LoopStart </pre>

ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>อินพุตส่ง row Array ไปยังฟังก์ชัน 6 turn ไปยัง 6 ให้ *a และ *b main ทำหน้าที่จาก function ar ที่ Code ar ที่ส่งค่าไปให้ *a และ *b *row → r และ *col → c</p>	<pre> 1 #include <stdio.h> 2 3 void ar(int **a, int *row, int *col){ //function ar 4 printf("How many row, col ?\n"); 5 scanf("%d %d", row, col); // input max array[row][col] 6 *a = new int[*row * *col]; // pointer array dynamic array 7 for(int i = 0 ; i < *row ; i++){ // start loop row 8 for(int j = 0 ; j < *col ; j++){ // start loop col 9 printf("a[%d][%d] = ", i, j); 10 scanf("%d", &a[i * *col + j]); // input cost position row = i, col = j 11 } // end loop col 12 } // end loop row 13 } //end function ar 14 15 void ar(int **a, int *row, int *col); 16 17 int main() 18 { 19 int r, c, *arr; 20 ar(&arr, &r, &c); //use function ar in position row and col in **a 21 return 0; 22 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
	

ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน	
คำอธิบาย	Code ตัวอย่าง
<p>แตกต่างจากการส่งผ่านแบบ Pass by reference เพราะมัน</p> <p>ป้: my function มี Pointer และ ส่งตัวแปร *a มาให้คนอื่น</p> <p>array และ return ค่า a ออกมา โดเมน main ที่เรียกใช้รับค่า</p> <p>ตัวส่งค่าส่งมาในตัวแปร arr เพื่อให้ arr มีค่าและไม่ต้องส่งค่าคืนค่า</p> <p>ค่าของ array</p>	<pre> 1 #include <stdio.h> 2 3 int *ar(int *row, int *col){ 4 int *a; 5 printf("How many row, col ?\n"); 6 scanf("%d %d", row, col); 7 a = new int[*row * *col]; 8 for(int i = 0 ; i < *row ; i++){ 9 for(int j = 0 ; j < *col ; j++){ 10 printf("a[%d][%d] = ", i, j); 11 scanf("%d", &a[i * *col + j]); 12 } 13 } 14 return a; 15 } 16 17 int *ar(int *row, int *col); 18 19 int main() 20 { 21 int r, c, *arr; 22 arr = ar(&r, &c); 23 delete [] arr; 24 return 0; 25 } </pre>
ผลลัพธ์ของ Code (Captureพร้อมแปะรูป)	Flow chart ของ Code ตัวอย่าง
