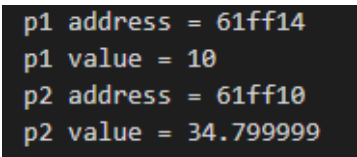
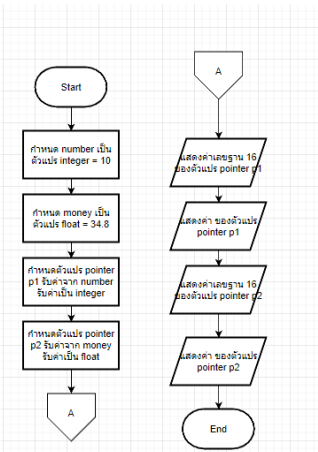
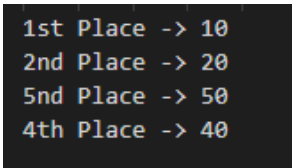
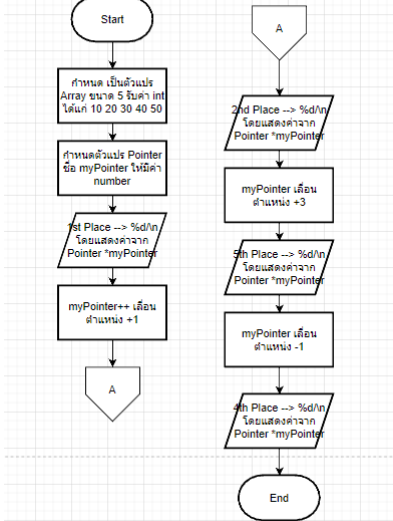


ตอนที่ 1 จงอธิบายความหมายพร้อมยกตัวอย่างประกอบ

จงเขียนคำอธิบาย ยกตัวอย่างประกอบ และวาดรูปประกอบตามความเข้าใจของคุณ

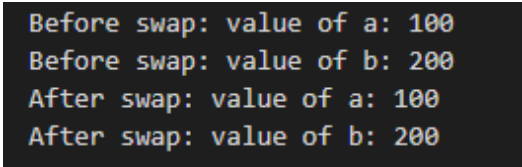
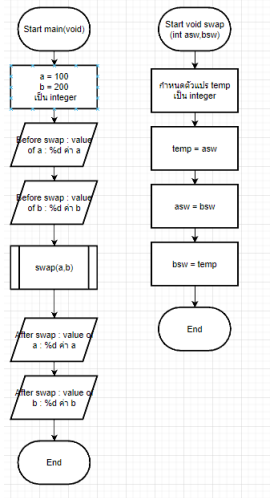
| ข้อที่ 1 จงอธิบายความหมายของ Pointer อย่างละเอียด และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน Pointer | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>Pointer พูดให้เข้าใจอย่างง่ายคือเป็นตัวที่คอยชี้ทางให้กับที่อยู่ของข้อมูล เป็นตัวแปรที่เก็บค่าของที่อยู่ของข้อมูลของตัวแปร</p> <p>ทำให้เราสามารถเข้าถึงข้อมูลได้โดยตรงโดยใช้ที่อยู่ของหน่วยความจำ</p> <p>โดยมีเครื่องหมายที่สำคัญในการใช้ Pointer ได้แก่</p> <ul style="list-style-type: none"> Address ที่อยู่ข้อมูล (&) Dereference Operator ตัวดำเนินการประกาศ (*) Declaration Operator ตัวดำเนินการอ้างอิง (*) | <pre> 1 #include <stdio.h> 2 3 int main() 4 { 5 int number = 10; 6 float money = 34.8; 7 8 int *p1 = &number; 9 float *p2; 10 p2 = &money; 11 12 printf("p1 address = %x\n", p1); //Output ของ p1 ในรูปแบบเลขฐาน 16 คือการใช้ %x 13 printf("p1 value = %d\n", *p1); 14 printf("p2 address = %x\n", p2); //Output ของ p2 ในรูปแบบเลขฐาน 16 คือการใช้ %x 15 printf("p2 value = %f\n", *p2); 16 return 0; 17 } </pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
|  <pre> p1 address = 61ff14 p1 value = 10 p2 address = 61ff10 p2 value = 34.799999 </pre> |  <pre> graph TD Start([Start]) --> A1[/A/] A1 --> P1Decl[/กำหนด number เป็น ตัวแปร integer = 10/] P1Decl --> P2Decl[/กำหนด money เป็น ตัวแปร float = 34.8/] P2Decl --> P1PtrDecl[/กำหนดตัวแปร pointer p1 รับค่าเป็น integer/] P1PtrDecl --> P2PtrDecl[/กำหนดตัวแปร pointer p2 รับค่าจาก money รับค่าเป็น float/] P2PtrDecl --> A2[/A/] A2 --> P1Print[/แสดงค่า ของตัวแปร pointer p1/] P1Print --> P2Print[/แสดงค่า ของตัวแปร pointer p2/] P2Print --> End([End]) </pre> |

| ข้อที่ 2 จงสร้าง Pointer จำนวน 1 ตัวที่ชี้ Array ไม่จำกัดแถว แถวละ 4 Column และยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>การใช้ Pointer มีประโยชน์กับ Array เพราะ Array มีการเก็บข้อมูลเป็นชุดอันดับ และเรียงต่อกัน</p> <p>จากโค้ดเป็นการใช้ Pointer กับ ตัวแปรอาร์เรย์ ซึ่งด้วยคำสั่ง Pointer ทำให้เราสามารถเลื่อนตำแหน่งของค่าที่ถูกจัดเรียงตามลำดับใน Array ได้ โดยเรากำหนดตัวแปร Pointer ชื่อ myPointer ให้มีค่าเป็น number โดยสามารถเลื่อนตำแหน่งได้โดยการ + หรือ - ค่า Pointer และแสดงผลลัพธ์ในตำแหน่ง Array ที่เราต้องการ</p> | <pre> 1 #include <stdio.h> 2 3 int main() 4 { 5 int number[5] = {10, 20, 30, 40, 50}; 6 int *myPointer = &number; 7 8 printf("1st Place -> %d\n", *myPointer); 9 myPointer++; 10 printf("2nd Place -> %d\n", *myPointer); 11 myPointer += 3; 12 printf("5nd Place -> %d\n", *myPointer); 13 myPointer--; 14 printf("4th Place -> %d\n", *myPointer); 15 return 0; 16 } </pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
|  <pre> 1st Place -> 10 2nd Place -> 20 5nd Place -> 50 4th Place -> 40 </pre> |  <pre> graph TD Start([Start]) --> Init[กำหนด เป็นตัวแปร Array ขนาด 5 รับค่า int ได้แก่ 10 20 30 40 50] Init --> Assign[กำหนดตัวแปร Pointer ชื่อ myPointer ให้มีค่า number] Assign --> Print1[/1st Place -> %d\n โดยแสดงค่าจาก Pointer *myPointer/] Print1 --> Inc1[myPointer++ เลื่อนตำแหน่ง +1] Inc1 --> A1{A} A1 --> Print2[/2nd Place -> %d\n โดยแสดงค่าจาก Pointer *myPointer/] Print2 --> Inc2[myPointer += 3] Inc2 --> Print3[/5th Place -> %d\n โดยแสดงค่าจาก Pointer *myPointer/] Print3 --> Dec[myPointer-- เลื่อนตำแหน่ง -1] Dec --> Print4[/4th Place -> %d\n โดยแสดงค่าจาก Pointer *myPointer/] Print4 --> End([End]) </pre> |

| ข้อที่ 3 จงอธิบายเรื่อง Pointer Functionยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>..... เป็นฟังก์ชันที่มีโครงสร้างเหมือนฟังก์ชันอื่นทั่วไป</p> <p>..... แต่ประกาศชื่อฟังก์ชันเป็น Pointer โดยต้องมีวงเล็บ</p> <p>..... กรอบไว้ เป็น</p> <p>..... <code>typename(*functionname)(parameter);</code></p> <p>.....</p> <p>..... จากโค้ด</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> | <pre>#include <stdio.h> void fun(int a) { printf("ค่าที่ Pointer ชี้เป็น %d\n", a); } int main() { void (*fun_ptr)(int) = fun; // & removed fun_ptr(10); // * removed return 0; }</pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
|  |  <pre>graph TD Start([Start]) --> Connector(()) Connector --> DefinePtr[กำหนด Pointer Function ชื่อ fun_ptr รับค่า integer] DefinePtr --> SetVal[ค่า fun_ptr เป็น 10] SetVal --> CallFun[void fun รับค่า int a] CallFun --> Connector StartVoid([Start Void fun]) --> PrintPtr[/ค่าที่ Pointer ชี้เป็น a/] PrintPtr --> End([End])</pre> |

| ข้อที่ 4 จงอธิบายเรื่อง Dynamic Array ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>Dynamic Array คืออาร์เรย์ที่สามารถเปลี่ยนขนาดได้ ในขณะที่โค้ดกำลังทำงาน และสามารถเพิ่มองค์ประกอบได้อย่างต่อเนื่องที่ตำแหน่งสิ้นสุดของ Array Dynamic</p> | <pre style="background-color: #2e3436; color: #eeeeec; padding: 10px;">1 #include <iostream> 2 using namespace std; 3 int main() { 4 int p,num; 5 cout << "Enter the numbers: " << "\n"; 6 cin >> num; 7 int *Array = new int(num); 8 cout << "Enter " << num << " numbers " << endl; 9 for (p = 0; p < num; p++){ 10 cin >> Array[p]; 11 } 12 cout << "Your numbers are: "; 13 for (p = 0; p < num; p++){ 14 cout << Array[p] << " "; 15 } 16 cout << "\n "; 17 return 0; 18 }</pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
| <div style="background-color: #2e3436; color: #eeeeec; padding: 10px; margin-bottom: 10px;"> Enter the numbers: 4 Enter 4 numbers 2 3 4 1 Your numbers are: 2 3 4 1 </div> <div style="background-color: #2e3436; color: #eeeeec; padding: 10px;"> C:\tmp\CMGR7NkuCa.o Enter the numbers: 6 Enter 6 numbers 3 2 3 1 5 4 Your numbers are: 3 2 3 1 5 4 </div> | <pre> graph TD Start([Start]) --> Init[กำหนดตัวแปร p, num เป็น integer] Init --> Input[/Enter the numbers ป้อนค่าไว้ใน num/] Input --> Create[สร้างอาร์เรย์ Pointer ชื่อ Array รับค่า num] Create --> Input2[/Enter จำนวน numbers/] Input2 --> Decision1{p < num?} Decision1 -- Yes --> Store[รับค่าเก็บไว้ใน Array ชื่อ Array[p]] Store --> Output[/Your numbers are/] Output --> Print[/Array [p]/] Print --> Increment[p++] Increment --> Decision1 Decision1 -- No --> End([End]) </pre> |

| ข้อที่ 5 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by reference ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>Pass by Ref คือการส่งค่าไปยังฟังก์ชันที่ถูกเรียกใช้ โดยส่งเป็นค่าตำแหน่งที่อยู่ (Address) ของตัวแปรไปซึ่งหากภายในฟังก์ชัน มีการเปลี่ยนแปลงค่าของ Argument ที่ส่งไป ก็จะมีผลทำให้ค่านั้นเปลี่ยนไปด้วย</p> | <pre>#include <stdio.h> int main () { int a = 100; int b = 200; printf("Before swap, value of a : %d\n", a); printf("Before swap, value of b : %d\n", b); swap(&a, &b); printf("After swap, value of a : %d\n", a); printf("After swap, value of b : %d\n", b); return 0; } void swap(int *x, int *y) { int temp; temp = *x; //เก็บค่า pointer x ไว้ที่ temp *x = *y; //รับค่า pointer y เก็บไว้ที่ pointer x *y = temp; //รับค่า temp เก็บไว้ที่ pointer y เป็นการสลับค่ากัน return; }</pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
| <div>Before swap, value of a : 100 Before swap, value of b : 200 After swap, value of a : 200 After swap, value of b : 100</div> | <pre>graph TD subgraph Main Start([Start]) --> Init[กำหนดตัวแปร a = 100 b = 200] Init --> PrintBefore[Before swap, value of a : %d เป็นค่า a Before swap, value of b : %d เป็นค่า b] PrintBefore --> CallSwap[swap(&a, &b)] CallSwap --> PrintAfter[After swap, value of a : %d เป็นค่า a After swap, value of b : %d เป็นค่า b] PrintAfter --> EndMain([End]) end subgraph SwapFunction StartVoid([Start void swap กำหนดตัวแปร Pointer x และ y]) --> DeclTemp[กำหนดตัวแปร temp เป็น integer] DeclTemp --> AssignTemp[กำหนดค่า temp = pointer x] AssignTemp --> AssignX[กำหนดค่า pointer x = pointer y] AssignX --> AssignY[กำหนดค่า pointer y = temp] AssignY --> EndVoid([End]) end</pre> |

| ข้อที่ 6 จงอธิบายการส่งผ่านตัวแปรแบบ Pass by value ยกตัวอย่างการใช้งาน พร้อม Code ตัวอย่างการใช้งาน | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| คำอธิบาย | Code ตัวอย่าง |
| <p>Pass by value คือ การส่งค่าเป็น argument ของ function โดยค่าในฟังก์ชันจะไม่ส่งผลต่อตัวแปรนอกฟังก์ชัน</p> | <pre> 1 void swap(int asw, int bsw) 2 { 3 int temp; 4 5 temp = asw; 6 asw = bsw; 7 bsw = temp; 8 9 return; 10 } 11 12 int main(void) 13 { 14 int a = 100; 15 int b = 200; 16 17 printf("Before swap: value of a: %d\n", a); 18 printf("Before swap: value of b: %d\n", b); 19 20 swap(a, b); 21 22 printf("After swap: value of a: %d\n", a); 23 printf("After swap: value of b: %d\n", b); 24 25 return 0; 26 } </pre> |
| ผลลัพธ์ของ Code (Captureพร้อมแปะรูป) | Flow chart ของ Code ตัวอย่าง |
|  <pre> Before swap: value of a: 100 Before swap: value of b: 200 After swap: value of a: 100 After swap: value of b: 200 </pre> |  <pre> graph TD subgraph Main_Function [int main(void)] StartMain([Start main(void)]) --> InitMain[a = 100 b = 200 เป็น integer] InitMain --> PrintBeforeMain[/Before swap: value of a: %d คำ a/] PrintBeforeMain --> PrintBeforeMain2[/Before swap: value of b: %d คำ b/] PrintBeforeMain2 --> CallSwap[swap(a, b)] CallSwap --> PrintAfterMain1[/After swap: value of a: %d คำ a/] PrintAfterMain1 --> PrintAfterMain2[/After swap: value of b: %d คำ b/] PrintAfterMain2 --> EndMain([End]) end subgraph Swap_Function [void swap(int asw, int bsw)] StartSwap([Start void swap (int asw, bsw)]) --> TempInit[กำหนดตัวแปร temp เป็น integer] TempInit --> TempAssign[temp = asw] TempAssign --> AswAssign[asw = bsw] AswAssign --> BswAssign[bsw = temp] BswAssign --> EndSwap([End]) end </pre> |