บทที่ 14
ข้อมูลชนิด พอยเตอร์ และ รายการข้อมูลที่มีการชี้ต่อ

<u>วัตถุประสงค์ของบทนี้</u>

1. ทราบถึงความจำเป็นในการใช้ข้อมูลชนิดพอยเตอร์
2. ทราบถึงการกำหนดชนิดของตัวชี้
3. สามารถประกาศตัวแปรชนิดพอยเตอร์ได้
4. สามารถอ้างถึงตัวแปรชนิดพอยเตอร์ได้
5. เข้าใจคำสั่งพื้นฐานที่ใช้กับตัวแปรพอยเตอร์
6. มีทักษะในการเขียนโปรแกรมที่มีรายการข้อมูลมีการชี้ต่อได้
   - สามารถสร้างข้อมูลที่มีการชี้ต่อได้
   - สามารถลบรายการข้อมูลในโครงสร้างข้อมูลที่มีการชี้ต่อได้
   - สามารถแทรกรายการข้อมูลในโครงสร้างข้อมูลที่มีการชี้ต่อได้

การแก้ปัญหางานบางลักษณะเช่นการแทรกข้อมูล ในกรณีที่ข้อมูลถูกจัดเก็บเป็นเป็นโครงสร้างของข้อมูลชนิดแถว หรือเก็บเป็นระเบียนชนิดแถวก็ดี การแทรกข้อมูลนั้นมีผลให้ข้อมูลทั้งหมดตั้งแต่ ตำแหน่งที่แทรกเลื่อนไปเพิ่มจากตำแหน่งเดิมหนึ่งตำแหน่ง ซึ่งทำให้โปรแกรมทำงานช้าลง กรณีที่ข้อมูลมีจำนวนมากๆ อาจช้ามากถึงขั้นยอมรับการทำงานไม่ได้

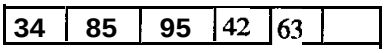
ในภาษาปาสคาลได้มีการแก้ปัญหาลักษณะนี้ โดยให้การแทรกข้อมูลเป็นไปอย่างอิสระไม่ส่งผลกระทบต่อข้อมูลอื่นๆ ให้ต้องขยับตำแหน่งไปจากเดิม การแก้ปัญหาดังกล่าวนี้คือการใช้ข้อมูลชนิดพอยเตอร์นั่นเอง

14.1 ตัวชี้ (pointer)

เป็นการกำหนดให้ข้อมูลมีโครงสร้างซึ่งชี้ไปยังข้อมูลตัวต่อไป เป็นแนวเส้นตรง(linear) สิ่งที่แตกต่างจากข้อมูลชนิดแถวคือ กรณีที่ต้องการเก็บตัวเลขจำนวนเต็มดังนี้

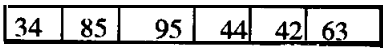
34 85 95 42 63

กรณีเก็บข้อมูลเป็นโครงสร้างข้อมูลชนิดแถวหนึ่งมิติ จะมีลักษณะดังนี้คือ



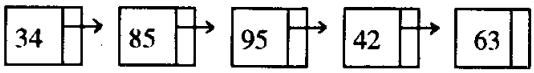
คือข้อมูลจะจัดเรียงกันตามลำดับ

กรณีต้องการแทรก 44 ระหว่างค่า 95 กับ 42

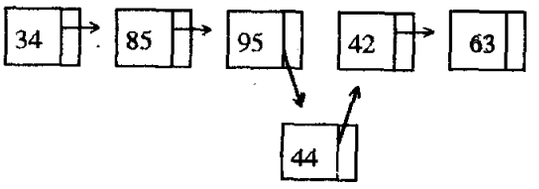


ข้อมูล 44 จะแทรกระหว่าง 95 กับ 42 โดยเลื่อนตำแหน่งของข้อมูล 42 และ 63 ให้เพิ่มหนึ่งช่อง ดังที่กล่าวแล้วว่ากรณีที่ข้อมูลหลังตำแหน่งที่แทรกมีจำนวนมากๆ ต้องเลื่อนตำแหน่งหมดทุกตัว ซึ่งไม่เป็นการดีในการทำงาน

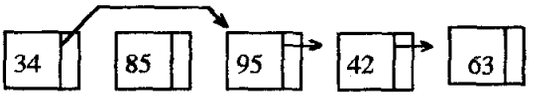
กรณีที่ให้ข้อมูลแต่ละตัวมีตัวชี้ ไปยังข้อมูลตัวถัดไปดังนี้



การแทรกข้อมูลสามารถกระทำได้โดย



กรณีนี้ไม่มีการขยับข้อมูลเก่าแต่อย่างใด ในทำนองเดียวกันถ้าต้องการลบข้อมูลออก ก็สามารถกระทำได้อย่างง่าย เช่นต้องการลบข้อมูล 85 ก็เพียงแต่เปลี่ยนตัวชี้จาก 34 เป็นชี้ไปที่ 95 แทนนั่นเอง



รายการที่มีการชี้ต่อด้วยตัวชี้ (pointer) ในลักษณะนี้เรียกว่า รายการที่มีการชี้ต่อ (linked list) ซึ่งตัวชี้เป็นข้อมูลชนิดหนึ่งซึ่งสามารถให้ค่าใหม่และเปรียบเทียบได้

การกำหนดชนิดของตัวชี้

ในภาษาปาสคาลการกำหนดชนิดของตัวชี้สามารถกระได้ดังนี้

```
Type   DataPointer = *Nodes ;
```

ตัวอย่างข้างต้นเป็นการกำหนดชนิดของตัวชี้ชื่อ DataPointer ซึ่งเป็นตัวชี้ไปยังโครงสร้างข้อมูลชนิด Nodes โครงสร้างข้อมูล Nodes นี้จะแทนรายการข้อมูลแต่ละรายการซึ่งประกอบด้วยตัวรายการและตัวชี้รายการถัดไป เช่น

```
Nodes = record
        n      : integer;
        next   : DataPointer;
        end;
```

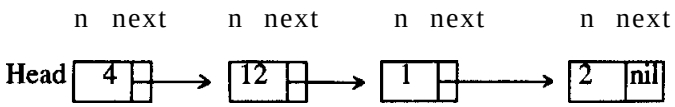
โครงสร้างข้อมูลชื่อ Nodes นี้ประกอบด้วย 2 ฟิลด์คือฟิลด์ที่เก็บเลขจำนวนเต็ม และฟิลด์ที่เป็นตัวชี้ตำแหน่งที่อยู่ของระเบียนถัดไป

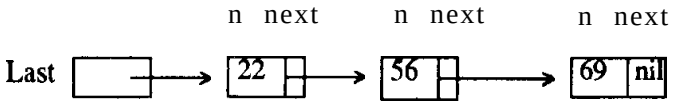
14.2 การประกาศตัวแปรชนิดพอยเตอร์

ในหัวข้อที่แล้วเป็นการกำหนดชนิดของข้อมูลพอยเตอร์ ซึ่งเป็นแบบในการกำหนดตัวแปร ในการประกาศตัวแปรชนิดพอยเตอร์สามารถประกาศได้ดังนี้คือ

```
VAR Head, Last : DataPointer ;
```

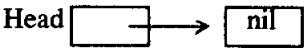
หมายความว่า ตัวแปรชื่อ Head และ Last เป็นตัวแปรพอยเตอร์ชนิด DataPointer ข้อมูลชนิดพอยเตอร์นี้จะมีจุดสิ้นสุดข้อมูลที่ nil ซึ่งเป็นตัวชี้รายการที่ไม่มีการชี้ต่อ





กรณีที่ไม่มีข้อมูลในรายการข้อมูลที่มีตัวชี้ (linked list) สามารถกำหนดได้โดย

```
Head := nil ;
```



หมายความว่าตัวแปรพอยเตอร์ Head ชี้ไปที่ nil คือไม่มีรายการใน Linked list นี้

14.3 การอ้างถึงตัวแปรชนิดพอยเตอร์

การอ้างถึงตัวแปรชนิดพอยเตอร์สามารถกระทำได้โดยมีลักษณะการอ้างถึงเหมือนตัวแปรชนิดเรคอร์ดทุกประการคือสามารถอ้างถึงได้โดยตรง เช่น

```
Head^.n  :=  78 ;
Head^.next := Last;
Last^.n  :=  67;
```

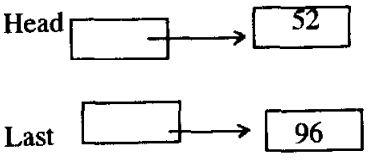
หรือสามารถอ้างถึงโดยใช้ With .. do ก็ได้

```
With   Head*   do
        begin
                n     := 78 ;
                next  := Last ;
        end:
```

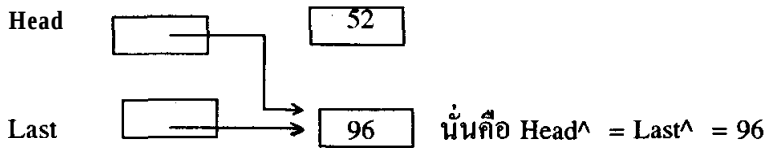
ความแตกต่างระหว่าง Head กับ Head^

สมมุติให้ Head และ Last เป็นตัวแปรพอยเตอร์ชนิด DataPointer

ให้ Head ชี้ไปยังตำแหน่งข้อมูล 52
ให้ Last ชี้ไปยังตำแหน่งข้อมูล 96

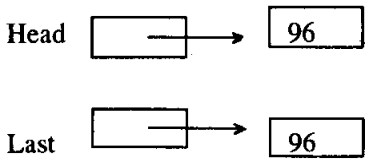


กรณีที่ 1 ถ้าใช้คำสั่ง Head := Last ;



นั่นคือ Head^ = Last^ = 96

กรณีที่ 2 ถ้าใช้คำสั่ง Head^ := Last^ ;



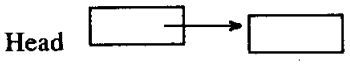
นั่นคือ Head^ = Last^ = 96

แต่ Head o Last

14.4 คำสั่งที่สร้างตำแหน่งที่เก็บข้อมูล และการยกเลิกตำแหน่งที่เก็บข้อมูล

ในหัวข้อที่แล้วมาเป็นเพียงประกาศตัวแปรชนิดพอยเตอร์เท่านั้นไม่ใช่การสร้างตำแหน่งที่เก็บข้อมูล การสร้างที่เก็บตัวแปรนั้นกระทำได้โดยคำสั่ง New เช่นต้องการสร้างที่เก็บข้อมูล โดยให้ตัวแปรพอยเตอร์ Head ชี้ไปยังตำแหน่งนี้ ซึ่งเป็นเนื้อที่ว่างๆ

NEW(Head) ;



ที่เก็บข้อมูลที่สร้างโดยคำสั่ง New

ที่เก็บข้อมูลนี้ถูกสร้างมาใหม่ยังไม่ได้กำหนดค่าของข้อมูลในฟิลด์ต่างๆ

ในกรณีที่สร้างตำแหน่งที่เก็บข้อมูลขึ้นมาแต่ไม่ได้ใช้ประโยชน์แล้ว เช่นการลบข้อมูลออกจาก Linked list ที่สร้างขึ้นมา สามารถกระทำได้โดยยกเลิกที่เก็บนั้นออกได้โดยคำสั่ง dispose

Dispose(Head) ;

คอมพิวเตอร์จะปล่อยเนื้อที่ส่วนนี้ที่ได้กำหนดโดยใช้คำสั่ง New ให้เป็นอิสระ ในกรณีที่ใช้คำสั่งนี้กับข้อมูลที่มีรายการชี้ต่อจะเป็นการยกเลิกข้อมูลรายการนั้นทั้งหมด รวมทั้งตัวชี้รายการถัดไปด้วย ดังนั้นการใช้คำสั่งนี้ผู้เขียนโปรแกรมจะต้องต่อตัวชี้ในรายการก่อนกับรายการหลังเข้าด้วยกันเอง

ตัวอย่างที่ 14.1

การสร้างข้อมูลชนิด Linked list ของกลุ่มรายการที่ประกอบด้วย ชื่อ และ อายุ

```
Program List1(input,output) ;
Type   DataLii = ^Data ;                          ....(1)
       Data    = Record                           ....(2)
                 Name : String[10] ;
                 Age  : 0..100 ;
                 Next : DataLii ;
                 End;
Var    Head, Last, Node : DataLii;                ....(3)
Begin
       New(Node) ;                                ....(4)
       Head := Node ; Last := Node ;              ....(5)
       readln(Node^.Name);                        ....(6)
       readln(Node^.Age);                         ....(7)
       While Node^.Age > 0 do                     ....(8)
             Begin
                   New(Node);                     ....(9)
                   readln(Node^.Name);            ....(10)
                   readln(Node^.Age);
                   Last^.Next := Node;            ....(n)
                   Last := Node ;                 ....(12)
                   Last^.Next := Nil ;            ....(13)
             End;
       Writeln (' Print data in Linked List ');
       Node := Head ;                             ....(14)
```

```
While Node o Nil do                                    .....(15)
    Begin
        Writeln(Node^.Name:20,Node^.Age:10);           .....(16)
        Node := Node^.Next ;                           .....(17)
    End;
End.
```

คำอธิบาย

1. เป็นการประกาศชนิดข้อมูลพอยเตอร์ชื่อ DataLink
2. ประกาศชนิดข้อมูล ชื่อ Data มีโครงสร้างเป็นลิสต์เชื่อมโยงประกอบด้วยฟิลด์ชื่อ ฟิลด์อายุ และฟิลด์ที่ชี้ไปยังโครงสร้างชนิดข้อมูล Data
3. ประกาศตัวแปรชื่อ Head, Last และ Node ให้มีชนิดข้อมูลเป็น DataLink
4. สร้างโครงสร้างใหม่ โดยให้ตัวแปรพอยเตอร์ชื่อ Node ชี้ไปยังโครงสร้างนี้
5. กำหนดให้ตัวแปรพอยเตอร์ Head และตัวแปรพอยเตอร์ Last ชี้ไปที่โครงสร้างใหม่ที่สร้างขึ้นในข้อที่ 4 ด้วย
6. รับชื่อทางแป้นพิมพ์ โดยเก็บไว้ในฟิลด์ชื่อในโครงสร้างใหม่ที่สร้างขึ้น
7. รับอายุทางแป้นพิมพ์ โดยเก็บไว้ในฟิลด์อายุในโครงสร้างใหม่ที่สร้างขึ้น
8. ในกรณีที่อายุที่รับทางแป้นพิมพ์มากกว่าศูนย์ให้ทำต่อในข้อ 9-13
9. สร้างโครงสร้างใหม่ โดยให้ตัวแปรพอยเตอร์ชื่อ Node ชี้ไปยังโครงสร้างนี้
10. รับชื่อและอายุทางแป้นพิมพ์ โดยเก็บไว้ในฟิลด์ชื่อและฟิลด์อายุในโครงสร้างใหม่ที่สร้างขึ้น
11. ให้ฟิลด์ที่เป็นตัวชี้ของตัวแปรพอยเตอร์ Last ชี้ไปยังโครงสร้างใหม่ที่สร้างขึ้นในข้อที่ 9
12. ให้ตัวแปรพอยเตอร์ Last ชี้ไปที่โครงสร้างใหม่ที่สร้างขึ้นในข้อที่ 9
13. ให้ฟิลด์ที่เป็นตัวชี้ของตัวแปรพอยเตอร์ Last ชี้ไปที่ Nil
14. กำหนดให้ตัวแปรพอยเตอร์ Node ชี้ไปที่โครงสร้างข้อมูลที่ตัวแปรพอยเตอร์ Head ชี้อยู่
15. ในขณะที่ตัวแปรพอยเตอร์ Node ไม่ได้ชี้ไปที่ Nil ให้ทำข้อ 16-17
16. พิมพ์ข้อมูลที่ตัวแปรพอยเตอร์ Node ชี้อยู่
17. กำหนดให้ตัวแปรพอยเตอร์ Node ชี้ไปที่โครงสร้างข้อมูลที่ฟิลด์ตัวชี้ของตัวแปรพอยเตอร์ Node ชี้อยู่ หรืออีกนัยหนึ่งคือ เลื่อนให้ตัวแปรพอยเตอร์ Node ชี้ไปยังโครงสร้างข้อมูลตัวถัดไป

ตัวอย่างที่ 14.2

การแทรกข้อมูล Linked list ของกลุ่มรายการที่ประกอบด้วย ชื่อ และ อายุ

```
Program List1(input,output) ;
Type   DataLink = ^Data ;
       Data     = Record
                  Name : String[10] ;
                  Age  : 0..100 ;
                  Next : DataLink ;
                  End;
Var    Head , Last , Node ,Front, Ins: DataLink;
       Post , Insert : String[10];
Begin
       New(Node) ; Head := Node ;  Last := Node ;
       readln(Node^.Name);   readln(Node^.Age);
       While Node^.Age > 0 do
             Begin
                   New(Node);
                   readln(Node^.Name);   readln(Node^.Age);
                   Last^.Next := Node;   Last := Node ;
                   Last^.Next := Nil ;
             End;
       Write('Add after which name '); readln(Post);          . . ...(l)
       New(h);                                                 . . ...(Z)
       Readln(Ins^.Name); Readln(Ins^.Age) ;                   . . ...(3)
       Front := Head ;                                         . . ...(4)
       While Front^.Name o Post do                             . . ...(5)
             Front := Front^.Next ;                            .....(6)
       Ins^.Next := Front^.Next ;                              . . ...(7)
       Front^.Next := Ins ;                                    . . . ..(I()
```

```
Writeln (' Print data in Linked List ');
Node := Head ;
While Node <> Nil do
        Begin
                Writeln(Node^.Name:20,Node^.Age:10);
                Node := Node^.Next ;
End;
End.
```

คำอธิบาย

การสร้างและการพิมพ์ข้อมูล เหมือนตัวอย่างที่ 14.1 ทุกประการเพียงแต่ตัวอย่างนี้เป็นการแทรกข้อมูลที่ต้องการเข้าไปในโครงสร้าง Linked list ที่สร้างขึ้นมา

(1) รับชื่อซึ่งเป็นข้อมูลที่ ข้อมูลใหม่จะอยู่ต่อจากโครงสร้างที่ชื่อนี้เก็บอยู่
(2) สร้างโครงสร้างใหม่ขึ้นมาให้ตัวแปรพอยเตอร์ Ins ชี้อยู่
(3) รับชื่อ และอายุ เก็บไว้ในโครงสร้างใหม่ที่สร้างขึ้นในข้อที่ 2
(4) กำหนดให้ตัวแปรพอยเตอร์ Front ชี้ไปที่ที่เดียวกับที่ ตัวแปรพอยเตอร์ Head ชี้อยู่ อีกนัยหนึ่งคือ ชี้ไปที่ต้น Linked list หรือ ชี้ไปที่ข้อมูลตัวแรกนั่นเอง
(5) ในขณะที่ ข้อมูลในฟิลด์ชื่อที่ตัวแปรพอยเตอร์ Front ชี้อยู่ไม่ใช่ ชื่อที่ต้องการให้ทำ (6) ถ้าใช่ทำ (7)
(6) เลื่อนให้ตัวแปรพอยเตอร์ชี้ไปยังโครงสร้างข้อมูลถัดไป
(7) กำหนดให้โครงสร้างข้อมูลที่จะแทรกนี้ชี้ไปยังโครงสร้างที่ โครงสร้างก่อนชี้อยู่ก่อน
(8) กำหนดให้โครงสร้างข้อมูลที่อยู่ลำดับก่อนชี้มายังโครงสร้างใหม่ที่สร้างขึ้นมาแทน

ตัวอย่างที่ 14.3

การตัดข้อมูลออกจาก List

```
Program  Linked-List-3  (Input,Output);
Type     DataLink  =  ^Data ;
         Data      =  Record
                      Name :  String[10] ;
                      Age  :  0..100 ;
                      Next :  DataLink ;
                      End;
Var      Head , Girl , Last :  DataLink ;
         Del :  String[10] ;
Begin
         Head := Nil ;                                              . . ...(l)
         New(Girl); Readln(Girl^.Name); Readln(Girl^.Age)           . . . ...(2)
         While  Girl^.name o '/' do                                 . ...(3)
                  Begin
                           Girl^.Next := Head ;                     . . ...(4)
                           Head := Girl;                            . . ...(5)
                           New(Girl); readln(Girl^.name); readln(Girl^.Age);   . . ...(6)
                  End;
         Girl := Head ;                                             . . ...(7)
         Write('Which name do you want to delete');
         Readln(Del);                                               . ...(8)
         While  Girl^.Name o Del do                                 . . ...(9)
                  Begin
                           Last := Girl ;                           ...(10)
                           Girl := Girl^.Next ;                     . . ...(n)
                  End;
         Last^.Next := Girl^.Next ;                                 ...(12)
         Dispose(Girl)                                              ...(13)
```

```
Gil := Head ;                                                    ....(14)
While Gil o Nil da
        Begin
                Writeln(Girl^.Name:20,Girl^.Age:6);
                Girl := Girl^.Next ;
        End;
End.
```

คำอธิบาย

(1) กำหนดให้ตัวแปรพอยเตอร์ Head ชี้ไปที่ Nil

(2) กำหนดให้ตัวแปรพอยเตอร์ Girl ชี้ไปยังโครงสร้างใหม่ที่สร้างขึ้น
รับชื่อและอายุ เก็บในฟิลด์ชื่อ และ ฟิลด์อายุ ของโครงสร้างใหม่ที่สร้างขึ้น

(3) ขณะที่ชื่อที่รับเข้ามาไม่ใช่ / ให้ทำข้อ 4-6 ถ้าใช่ให้ทำข้อ 7

(4) กำหนดให้ฟิลด์ที่เป็นตัวชี้ของโครงสร้างใหม่ที่สร้างขึ้น ให้ชี้ไปที่ Head
อีกนัยหนึ่ง คือ โครงสร้างที่สร้างขึ้นใหม่นี้จะอยู่ลำดับก่อนโครงสร้างเก่าที่มีอยู่ก่อนนั่นเอง

(5) ให้ตัวแปรพอยเตอร์ Head ชี้ไปยังโครงสร้างที่ตัวแปรพอยเตอร์ Girl ชี้อยู่

(6) สร้างโครงสร้างใหม่ โดยรับชื่อ และอายุ เก็บในฟิลด์ของโครงสร้างใหม่นี้
โดยมีตัวแปรพอยเตอร์ ชื่อ Girl ชี้อยู่

(7) กำหนดให้ตัวแปรพอยเตอร์ Girl ชี้ไปยังโครงสร้างเดียวกันกับที่ตัวแปรพอยเตอร์ Head ชี้อยู่

(8) รับชื่อทางแป้นพิมพ์ ซึ่งเป็นข้อมูลที่ต้องการลบออกจาก list

(9) ในขณะที่ ชื่อที่ตัวแปรพอยเตอร์ Girl ชี้อยู่ไม่ใช่ชื่อที่ต้องการลบให้ทำข้อ 10-11
กรณีที่ใช่ ให้ทำข้อ 12

(10) ให้ตัวแปรพอยเตอร์ Last ชี้ไปยังโครงสร้างข้อมูลที่ตัวแปรพอยเตอร์ Girl ชี้อยู่

(11) เลื่อนให้ตัวแปรพอยเตอร์ Girl ชี้ไปยังโครงสร้างข้อูลตัวถัดไป

(12) ให้ฟิลด์ตัวชี้ที่ตัวแปรพอยเตอร์ Last ชี้อยู่ให้ชี้ไปยังโครงสร้างข้อมูลที่ Girl^.Next ชี้
ซึ่งคำสั่งนี้จะเป็นการลบโครงสร้างข้อมูลที่ Girl ชี้อยู่ออกไปจาก List

(13) คืนเนื้อที่ในหน่วยความจำที่ตัวแปรพอยเตอร์ Girl ชี้อยู่ให้แก่ระบบ

(14) กำหนดให้ ตัวแปรพอยเตอร์ Girl ชี้ไปยังโครงสร้างข้อมูลที่ Head ชี้อยู่
ในขณะที่ตัวแปรพอยเตอร์ Girl ไม่ได้ชี้ไปที่ Nil ให้พิมพ์ชื่อ และ อายุ ของโครงสร้าง
ข้อมูลที่ Girl ชี้อยู่ แล้วเลื่อนให้ตัวแปรพอยเตอร์ Girl ชี้ไปยังโครงสร้างข้อมูลตัวต่อไป
ซ้ำจนกระทั่งชี้ไปที่ Nil

แบบฝึกหัดท้ายบท

1. ให้สร้างข้อมูลชนิดที่มีการชี้ต่อของ 2 Linked list ซึ่งเก็บชื่อ และ เงินเดือนของพนักงาน 2 บริษัท โดยข้อมูลที่เก็บใน Linked list นี้จะเรียงลำดับตามชื่อจากน้อยไปมาก

2. จงเขียนโปรแกรมหา Union และ Intersetion ของ 2 Linked list ที่เก็บเลขจำนวนเต็ม
กำหนดให้ A และ B เป็นตัวแปรพอยเตอร์ที่ชี้ไปยังโครงสร้างข้อมูลที่เก็บเลขจำนวนเต็มและตัวชี้ต่อ ผลลัพธ์ของการทำงานจะได้เป็นสมาชิกทั้งหมดและสมาชิกที่ซ้ำของ 2 linked list นี้

3. จงเขียนโปรแกรมนับจำนวน โหนด ทั้งหมดที่มีอยู่ใน Linked list หนึ่ง