ลิงค์ลิสต์ 1

ผศ.ดร.อุไร ทองหัวไผ่

ตัวชี้ (Pointer)

- ตัวชี้(Pointer) เป็นชนิดของข้อมูลหนึ่งที่เก็บเลขที่อยู่ของข้อมูล
- **รูปแบบการประกาศตัวแปรชนิดตัวชี้**

type * Variable;

การประกาศตัวแปรชนิดตัวชี้

float *B;

เป็นการประกาศตัวแปรชนิดตัวชี้ชื่อ B ให้ตัวแปลภาษาได้รับรู้ โดยตัวแปรนี้เก็บเลขที่อยู่ของข้อมูลที่เป็นเลขทศนิยม โดยระบบไม่จัดสรรเนื้อที่ในหน่วยความจำแต่อย่างใด

B = 3000;

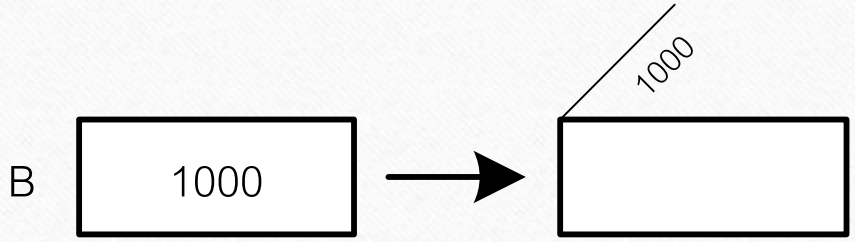
คำสั่งนี้เป็นการนำค่า 3000 ไปเก็บในตัวแปรตัวชี้ชื่อ B โดยค่าที่เก็บเป็นเลขที่อยู่โดยยังไม่มีการจัดสรรเนื้อที่เพื่อเก็บเลขทศนิยมใดๆดังรูป

การจองเนื้อที่ให้กับตัวแปรตัวชี้

new type;

ดังนั้นถ้ากำหนดคำสั่งดังนี้ B = new float;

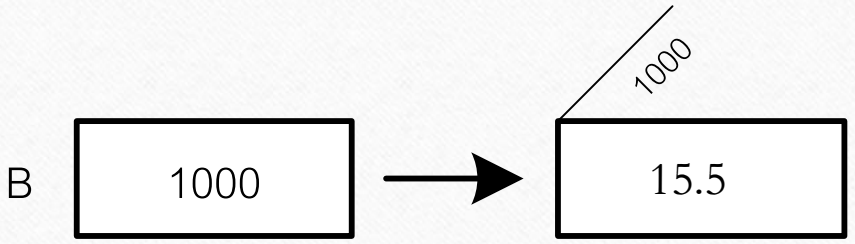
เป็นการจองเนื้อที่ใหม่เพื่อเก็บเลขทศนิยม 1 จำนวน โดยให้ตัวแปร B ชี้ที่เนื้อที่ใหม่



อ้างถึงค่าของข้อมูลที่ตัวแปรตัวชี้

ใช้สัญลักษณ์ * หรือ Asterisk หรือที่เรียกว่า Indirection operator

```
float   *B;
 B  =  new float;
*B  =  15.5;
```

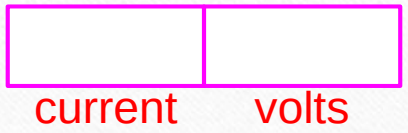


เป็นการนำค่าเลขทศนิยม 15.5 ไปจัดเก็บเลขที่อยู่ที่ตัวแปร B ชี้อยู่

Pointer to Structs

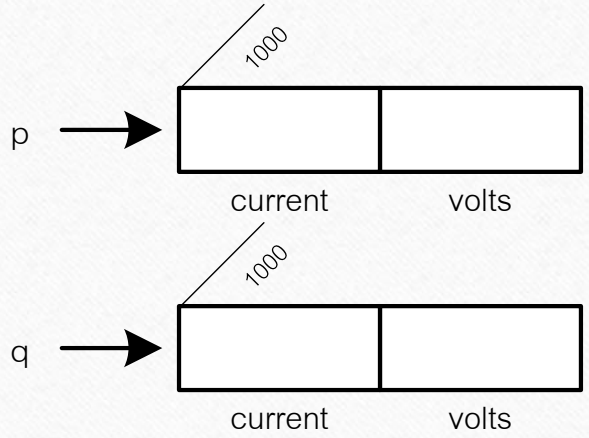
ประกาศโครงสร้างข้อมูล ดังนี้

```
struct electric
{
    string current;
    int  volts;
};
electric   *p, *q;
```



การจัดสรรเนื้อที่สำหรับเก็บข้อมูลนั้นต้องใช้คำสั่ง new ดังนี้

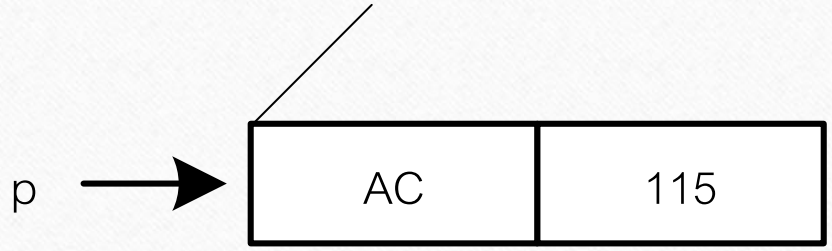
```
p  =  new electric;
q  =  new electric;
```



การนำข้อมูลเก็บในโครงสร้างที่จัดสรรไว้ สามารถใช้คำสั่งดังนี้

```
(*p).current = "AC";
(*p).volts = 115;
```

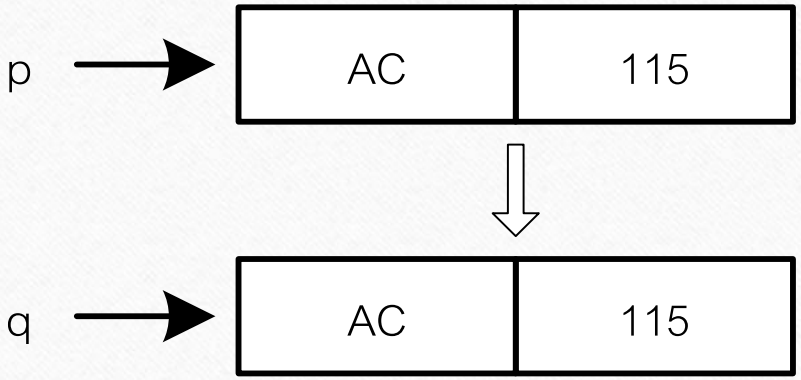
```
p -> current = "AC";
p -> volts = 115;
```



การคัดลอกข้อมูล

ถ้าต้องการคัดลอกข้อมูลทั้งหมดจากโหนดข้อมูลที่ตัวแปร p ชี้อยู่ให้กับโหนดข้อมูลที่ตัวแปร q ชี้ใช้คำสั่ง

```
*q = *p;
```

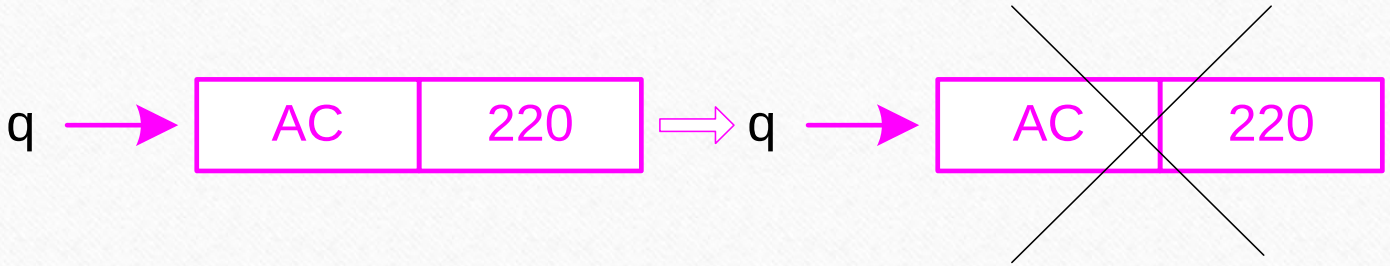


การลบโหนดข้อมูล

การลบโหนดข้อมูลเลขที่อยู่ที่ตัวแปรตัวชี้ชี้อยู่ใช้คำสั่ง delete มีรูปแบบดังนี้

รูปแบบ **delete Variable;**

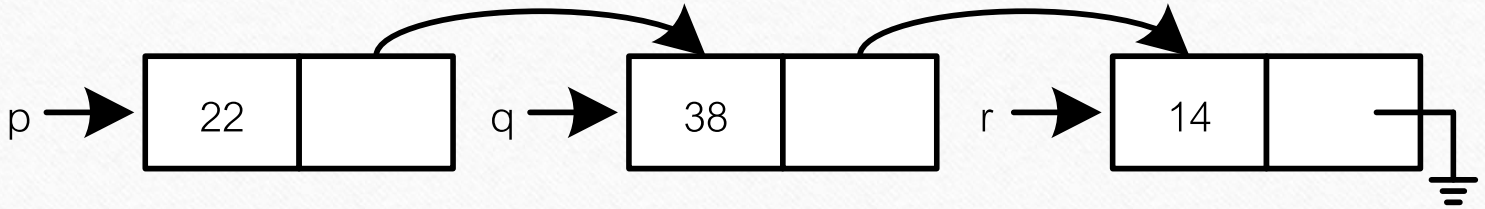
ตัวอย่าง delete q;



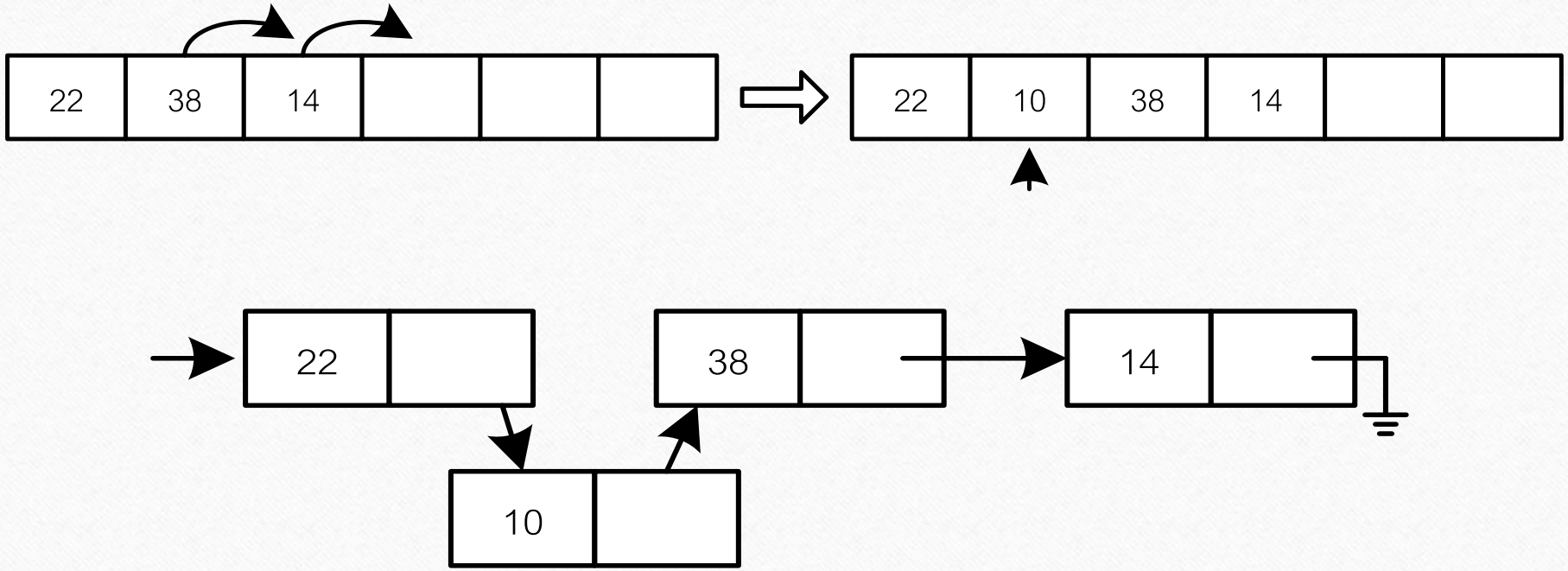
รายการโยง(Linked Lists)

รายการโยง(Linked list) เป็นการรวมกลุ่มของข้อมูลที่มีการจัดสรรโครงสร้างข้อมูลแบบพลวัตเป็นโครงสร้างข้อมูลที่ยืดหยุ่น

สมมุติว่าโปรแกรมเมอร์ต้องการเก็บข้อมูล 22 38 14



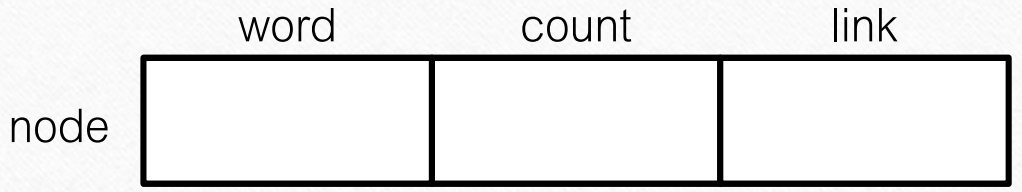
การแทรกข้อมูล



โหนด(Node)

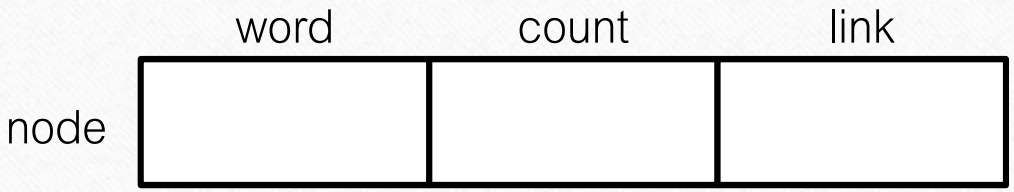
โหนด(Node)เป็นโครงสร้างข้อมูลชนิด struct ที่ประกอบด้วยเขตข้อมูลแบ่งเป็น 2 กลุ่มคือ

- Information field เป็นเขตข้อมูลที่จัดเก็บข้อมูล
- Link field เป็นเขตข้อมูลที่เก็บเลขที่อยู่ของโหนดตัวถัดไป ในกรณีที่ไม่มีข้อมูลตัวถัดไป ค่าของรายการข้อมูลนี้มีค่าเท่ากับ NULL



โครงสร้างข้อมูล

```
struct node
{
    string word;
    int count;
    node *link;
};
```



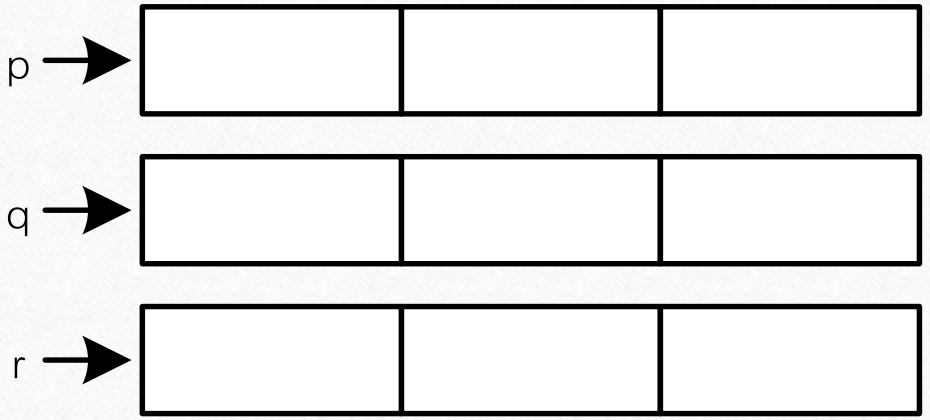
การจองเนื้อที่ว่าง

โปรแกรมเมอร์สามารถกำหนดตัวแปรชนิดตัวชี้เพื่อปฏิบัติการกับโหนดได้ดังนี้

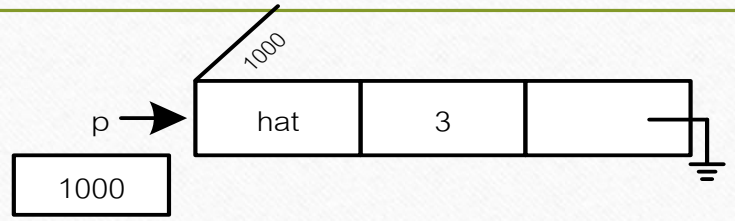
```
node *p, *q, *r;
```

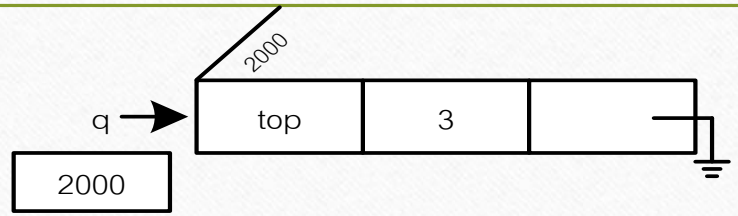
สามารถจองเนื้อที่ว่าง 1 โหนดเพื่อใช้งานได้ดังนี้

```
p = new node;
q = new node;
r = new node;
```



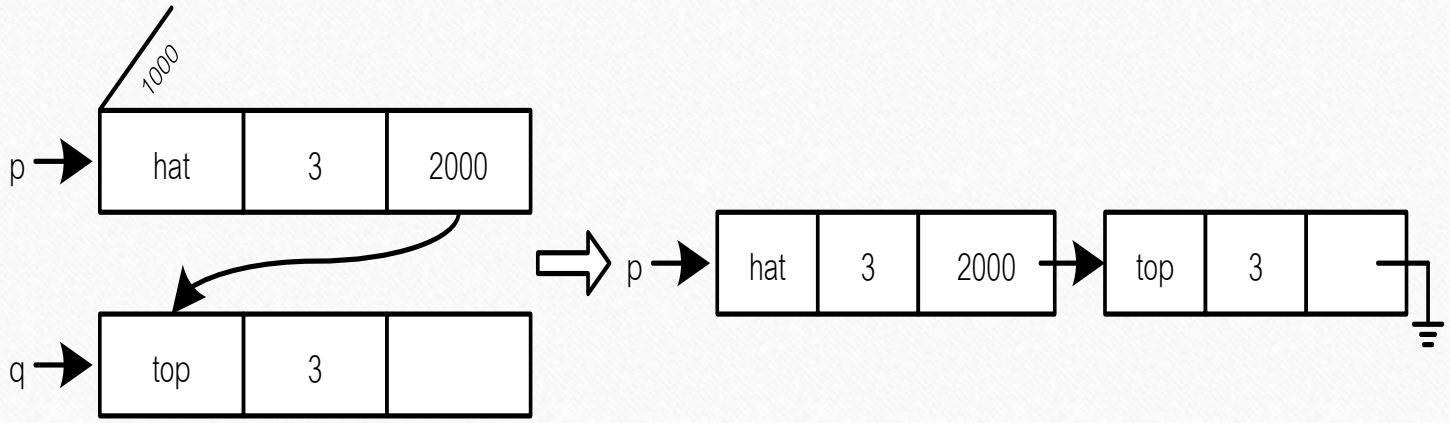
การเชื่อมโยงโหนด

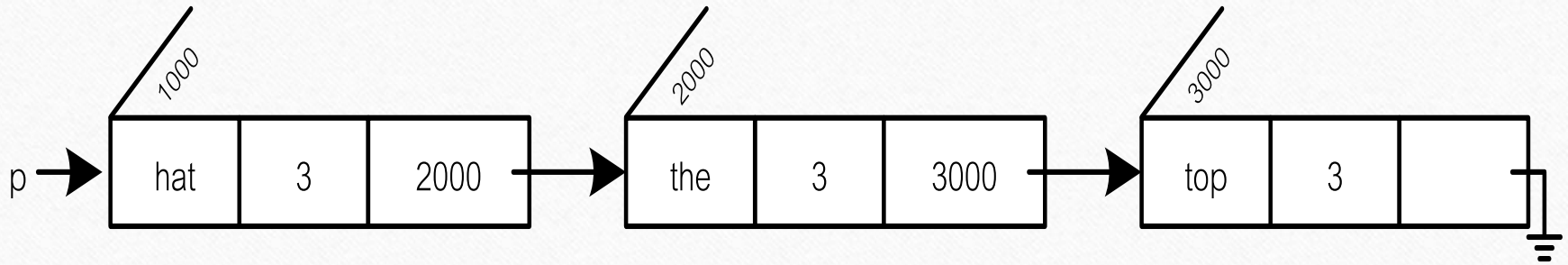




เรานำโหนดที่ตัวแปร p ชี้อยู่เชื่อมโยงกับโหนดที่ตัวแปร q ชี้ได้ดังนี้

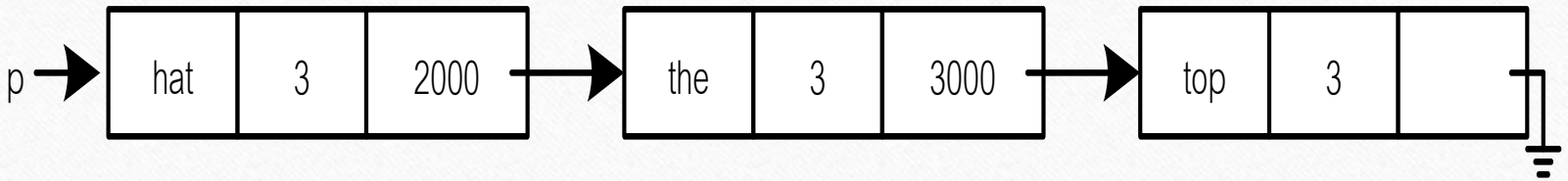
```
p -> link = q;
```



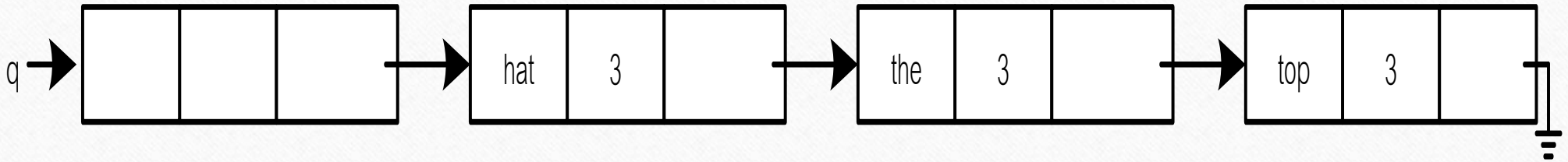


list reference	ผลลัพธ์
p->word	hat
p->link	link field of first node
p->link->word	the
p->link->link	link field of the second node
p->link->link->count	3

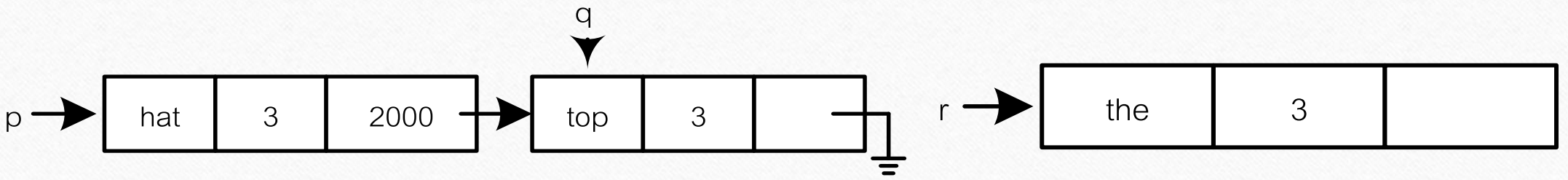
การแทรกโหนดใหม่เป็นโหนดแรก




```
q = new node;
q -> link = p;
```

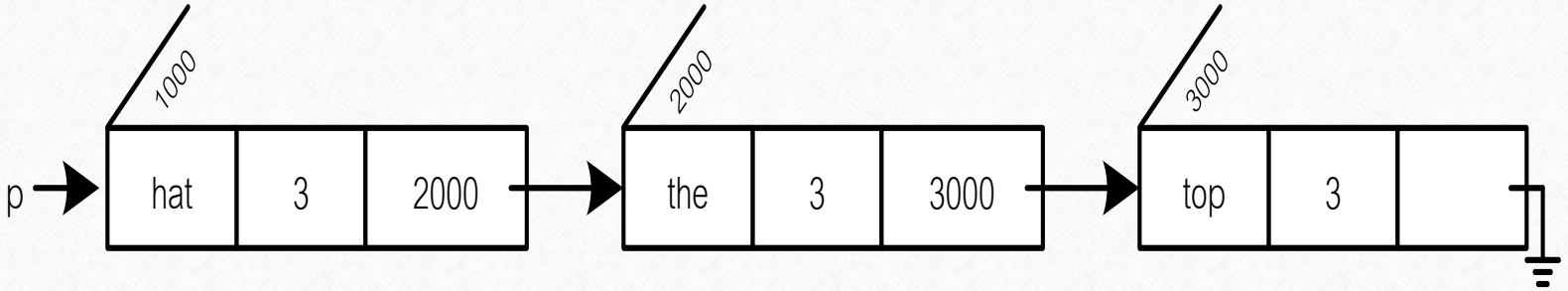


แทรกโหนดใหม่เป็นโหนดตรงกลาง

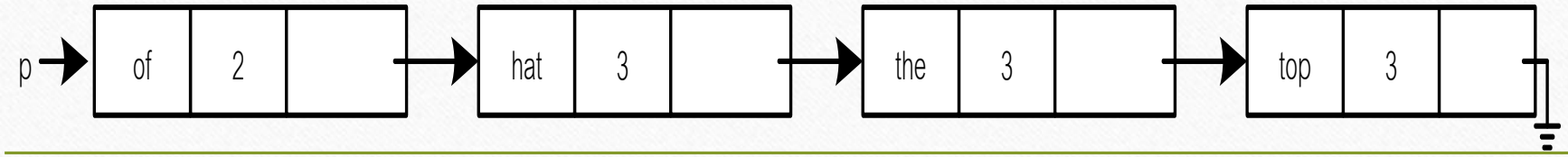


p -> link = r;

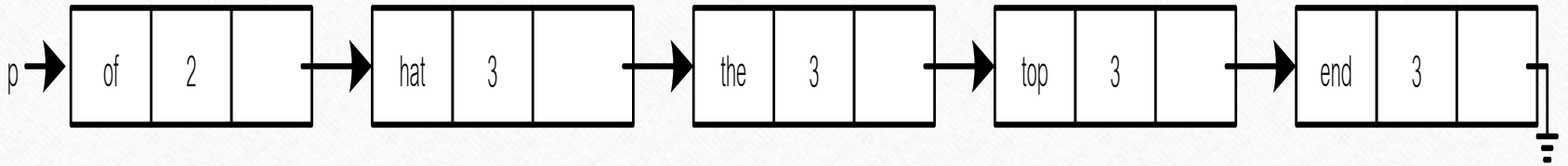
r -> link = q;



แทรกเป็นโหนดสุดท้าย



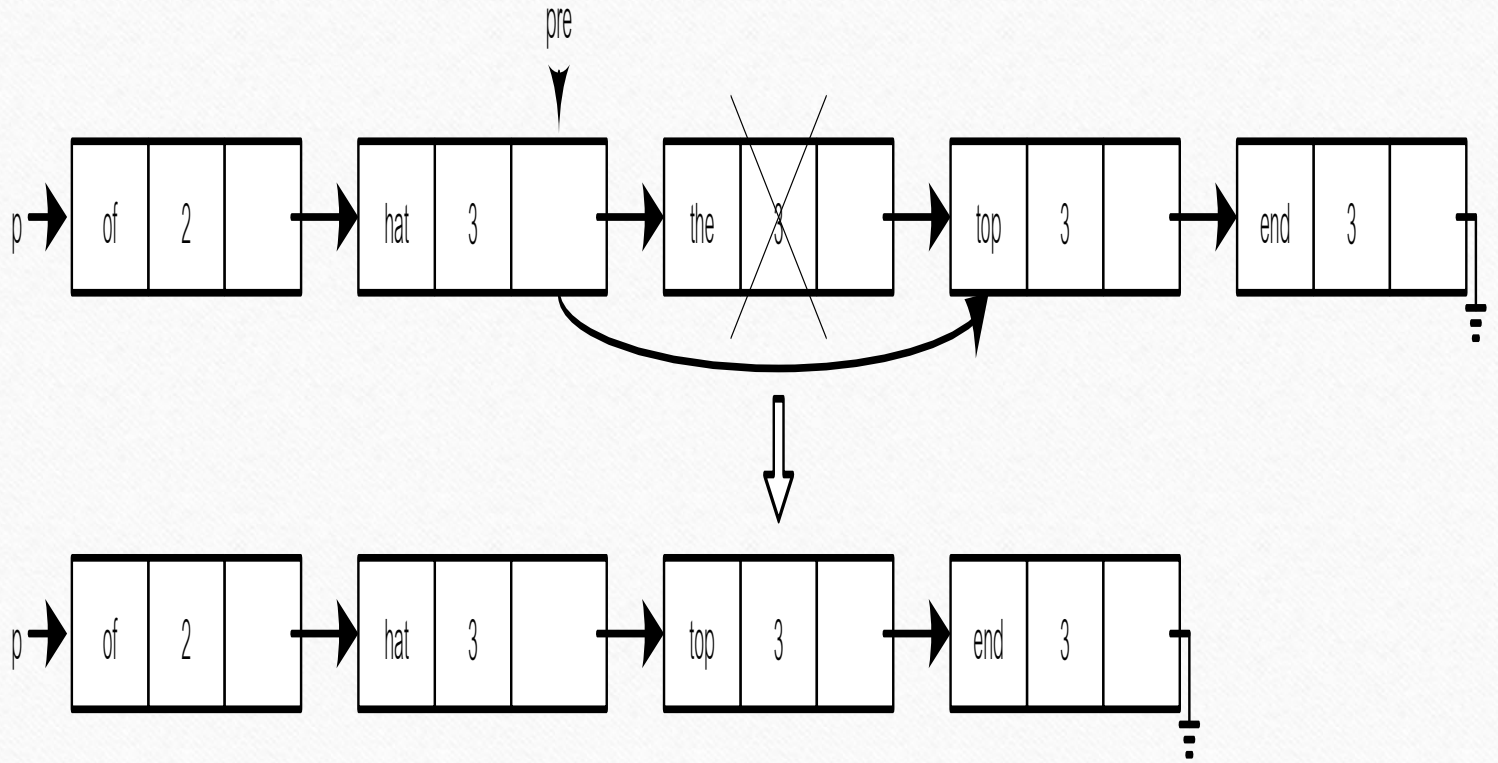

```
q = new node;
q -> word = "end";
q -> count = 3;
q -> link = NULL;
```




```
r = p;
while (r -> link != NULL)   //จะตรวจสอบจนกว่า r -> link = NULL
      r = r -> link;        //เลื่อนไปชี้ที่โหนดถัดไป
r -> link = q;              //เชื่อมกับ q เป็นโหนดสุดท้าย
```

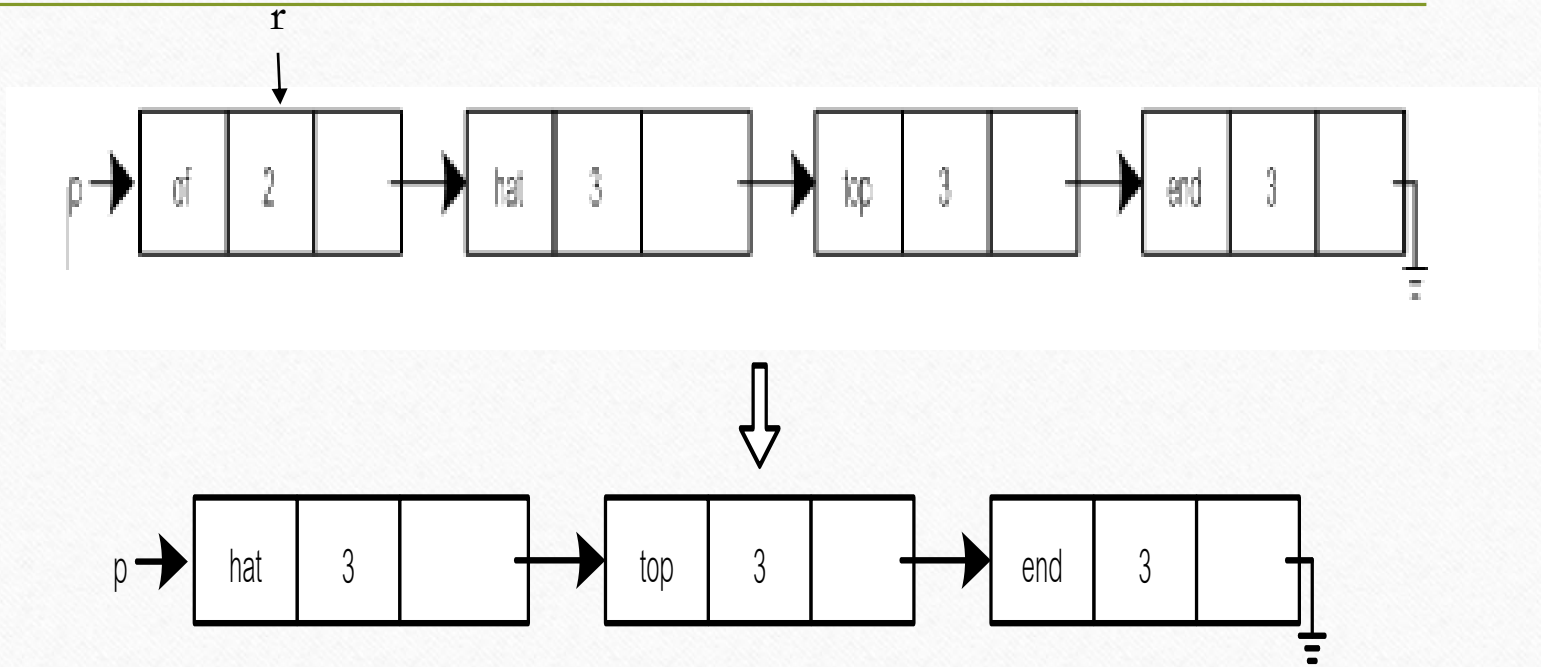
การลบโหนดข้อมูล

```
pre = NULL;
r = p;
while (r -> word != "the")
{
        pre = r;
        r = r -> link;
}
pre -> link = r -> link;
delete r;
```

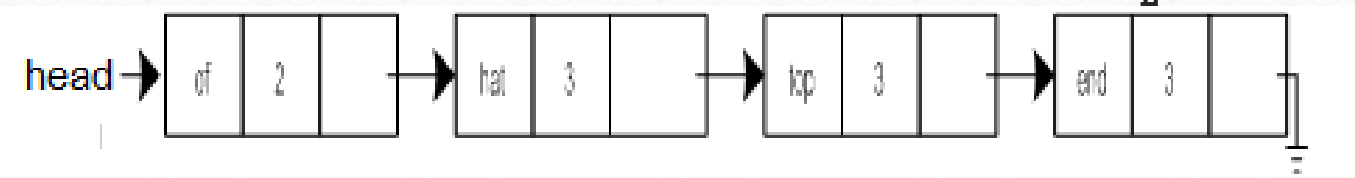


การลบโหนดข้อมูลเป็นโหนดแรก

```
if (pre = = NULL)
{
        p = p -> link;
        delete r;
}
```



การพิมพ์ข้อมูล




```
void printlist  (listnode *head)  //IN:pointer to list to be printed
{
      while (head != NULL)
            {
                  //no prior value of head was NULL.
                  cout  << head->word << " " << head-link<<endl;
                  head = head->link;     //advance to next list node.
            }
}    //end printlist
```

โปรแกรมเรียงลำดับคะแนนสอบ

- ฟังก์ชัน input ทำหน้าที่รับข้อมูลเป็นคะแนนสอบ(score) จากแป้นพิมพ์ การทำงานของฟังก์ชันจะนำคะแนนสอบที่รับมาเก็บในโหนดข้อมูลและเชื่อมในรายการโยงต่อท้ายจากโหนดสุดท้าย กำหนดให้ตัวแปรชื่อ first ชี้อยู่ที่ข้อมูลโหนดแรก
- ฟังก์ชัน print ทำหน้าที่แสดงข้อมูลที่เก็บในรายการโยงเริ่มจากข้อมูลโหนดแรกเป็นลำดับจนถึงโหนดสุดท้าย โดยเขียนในลักษณะของฟังก์ชันเรียกซ้ำ
- ฟังก์ชัน sort ทำหน้าที่ดึงข้อมูลในรายการโยงที่ตัวแปร first ชี้อยู่ทีละตัวเพื่อนำมาสร้างเป็นรายการโยงใหม่ที่มีการเรียงลำดับข้อมูลจากน้อยไปมาก กำหนดให้ตัวแปรsecond ชี้ที่รายการโยงใหม่

กำหนดโครงสร้างลิสต์เชื่อมโยง

```
struct node{
    int score;
    node* link;
};
```

```
void input(node*& first)
{
        char ch; node *p,*q;
        do
        {
                p =new node;
                cout<<"score=?";cin>>p->score;p->link=NULL;
                if(first==NULL)
                        first=p;
                else{
                        q=first;
                        while(q->link!=NULL)
                                q=q->link;
                        q->link =p;
                }
                cout<<"Run again y/n?";cin>>ch;
        }while(ch=='y');
}
```

```
void print(node *first)
{
    if(first!=NULL)
    {
        cout<<first->score<<" ";
        print(first->link);
    }
}
```

```
void sort(node *first,node*& newlist)
{
        node *p,*prev,*current;
        while(first != NULL)
        {
                p = new node;
                p->score=first->score;p->link=NULL;
                if(newlist==NULL)
                        newlist=p;
                else
                {
                        prev=NULL;current=newlist;
                        while(current!=NULL && p->score > current->score)       {
                                prev=current;
                                current=current->link;
                        }
                        if(prev==NULL)                  {
                                p->link=current;
                                newlist=p;
                        }
                        else                    {
                                prev->link=p;
                                p->link = current;
                        }
                }
                first=first->link;
        }
}
```

```
int main()
{
    node *first,*second;
    first=NULL;
    input(first);print(first);
    second=NULL;
    sort(first,second);
    cout<<"Sort data"<<endl;
    print(second);
    return 0;
}
```

แบบฝึกหัด

- จงเขียนโปรแกรมรับคะแนนของนักศึกษาจำนวน n คนเก็บในโครงสร้างลิสต์เชื่อมโยงทางเดียว
- จงหาคะแนนสูงสุด คะแนนต่ำสุด และคะแนนเฉลี่ย

การบ้าน

- จงเขียนโปรแกรมรับข้อมูลหนังสือประกอบด้วยรหัสหนังสือ ชื่อหนังสือ และผู้แต่งหนังสือจำนวน n เล่มทางแป้นพิมพ์เก็บในโครงสร้างลิสต์เชื่อมโยงทางเดียว
- จงลบหนังสือโดยป้อนชื่อผู้แต่งหนังสือ ที่ผู้แต่งหนังสือคนนี้เขียนซึ่งเก็บในลิสต์เชื่อมโยง และพิมพ์หนังสือก่อนลบและหลังลบออกทางจอภาพ