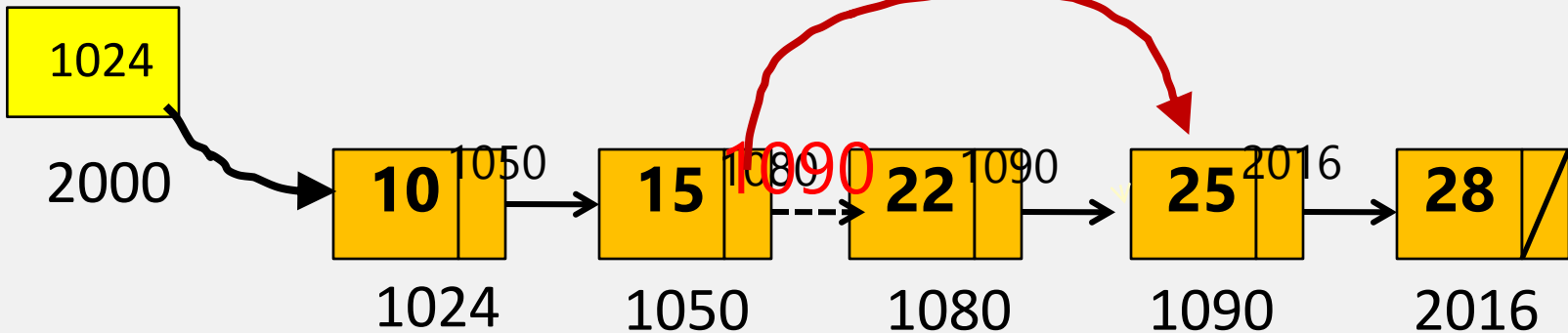




## 3. Delete

No. 66

Head



1. กรณีที่มี node เดียว `if (head->next == NULL)`
2. กรณีที่ลบ node แรก `{`  
`tmp = head;`
3. กรณีลบตั้งแต่โหนดที่ 2 เป็นต้นไป `head = NULL;`  
`delete (tmp);`  
`}`

```
if (data == head->value) {  
    head = head->next;  
}
```

```
if (data == head->value) {  
    tmp = head;  
    head = head->next;  
    delete (tmp);  
}
```



## 03603212 : Module3–List, stack, Queue 2

```
struct record *delete(struct record *head,int data)
```

```
{ struct record *node,*tmpfree;
```

```
node = head;
```

```
while (node != NULL)
```

```
{ if( data == node->next->value )
```

```
{ tmpfree = node->next;
```

```
node->next = node->next->next;
```

```
delete(tmpfree);
```

```
break;
```

```
}
```

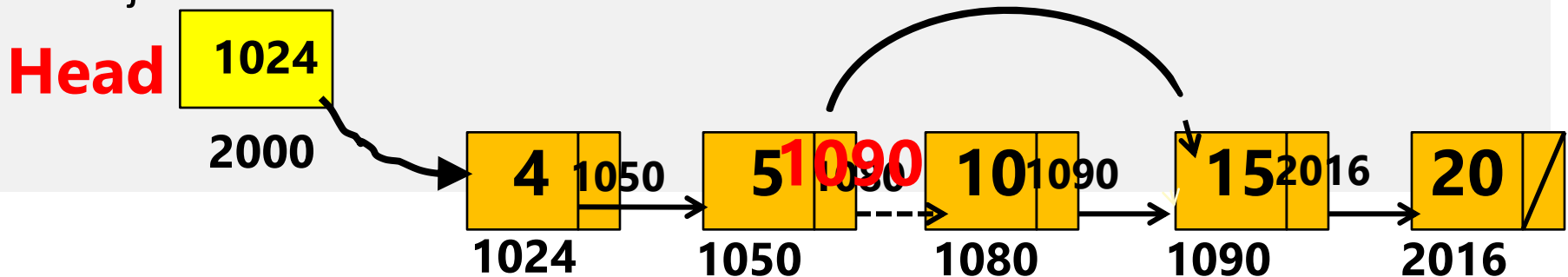
```
else
```

```
node=node->next;
```

```
} /* end while */
```

```
return node;
```

```
}
```





# 03603212 : Module3–List, stack, Queue 3

## การบ้าน

### 1. Linked list

=====

#### MENU

=====

- 1) Insert
- 2) Delete
- 3) Count
- 4) Print min to max, max to min
- 5) Print first half and second half
- 6) Find

Please choose >

ทำการ insert โดยเลือกข้อ 1

Insert : 5

List = 5

จากนั้นให้กลับไปเมนู

ให้ทดลอง insert เพิ่มทีละตัวคือ 3 1 10 8 9

จากนั้นถ้าเลือกข้อ 3 จะได้ค่า

Count = 6

เลือกข้อ 4

: 1 3 5 8 9 10

: 10 9 8 5 3 1

ถ้าเลือกข้อ 5 จะได้ค่า

First = 1 3 5

Second = 8 9 10

ทำการลบโดยเลือกข้อ 2

Delete : 9

List : 1 3 5 8 10

จากนั้นกลับไปเมนู

ลบอีกครั้ง

Delete : 7

Can't delete no 7 in list!!

จากนั้นกลับไปเมนู

ถ้าเลือกข้อ 5

First = 1 3 //หาร2=2

Second = 5 8 10



## การบ้าน สายลับข้ามชาติ

2. ให้นิสิต input passwd ที่เข้ารหัสแล้วใส่ใน list กำหนด structure ดังนี้

struct record

```
{ char c;  
  struct record *next;  
};
```

=====

MENU

=====

- 1) Input secret code
- 2) Decode
- 3) Exit

Please choose >

สมมุติว่า รหัสคือ NOBOMB

รหัสจะทำการเลือกตัวอักษรคือออกมาก่อนคือ  
ลำดับ 1,3,5 คือ NBM จากนั้นจะเลือก  
ตัวอักษรคู่ลำดับที่ 2,4,6 ออกมาต่อท้ายอักษร  
คือคือ OOB ดังนั้น

ข้อความ : NOBOMB

เข้ารหัส : NBMOOB

ถ้าเลือกข้อ 1 ให้ใส่รหัสลับที่นิสิตได้มา

Code : NBMOOB

จากนั้นกลับไปเมนู

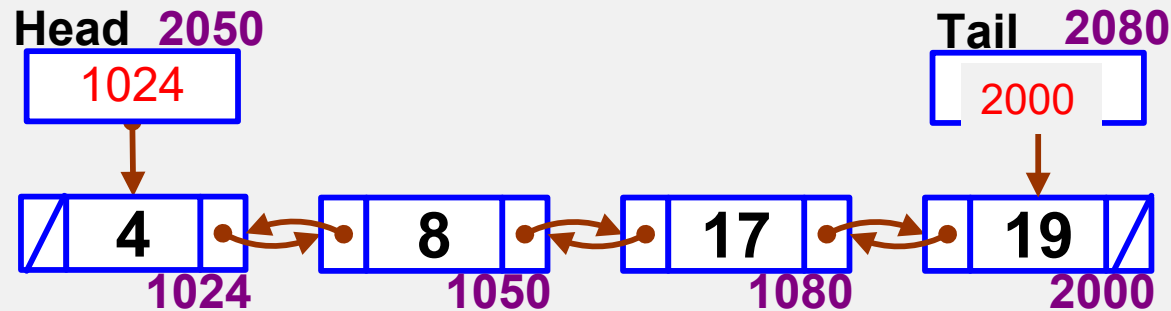
ถ้าเลือกข้อ 2 ทำการถอดรหัส

Answer : NOBOMB



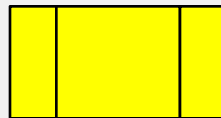
## 3.2 Doubly Linked Lists

The link list that add extra field to the data structure, containing a pointer to the previous cell.



struct record

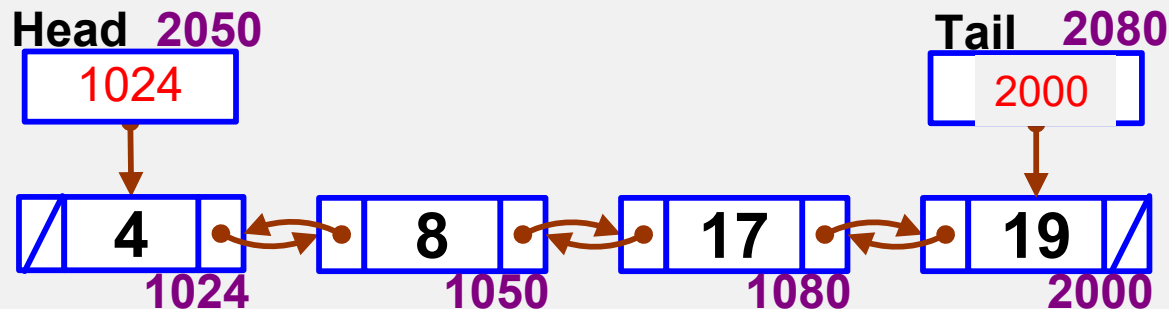
```
{  int data;  
    struct record *next;  
    struct record *prev;  
};
```





## 3.2.1 Insertion (Doubly Linklist)

1. Insert while no data in list
2. Insert first
3. Insert last
4. Insert middle





## โครงโปรแกรมในการ insert

```
if(ยังไม่มีข้อมูล ?) if (data < head->value)
```

```
{ ....  
}
```

```
else
```

```
{ สร้าง node เตรียมไว้
```

```
if ( insert ด้านหน้า ? )
```

```
{
```

```
.....
```

```
}
```

```
else
```

```
{ if( insert ด้านหลัง ? ) if ( data > tail->value)
```

```
{
```

```
....
```

```
}
```

```
else //ตรงกลาง
```

```
{ หาดำแหน่ง
```

```
.....
```

```
}
```

```
}
```

1. Insert while no data in list

2. Insert first

3. Insert last

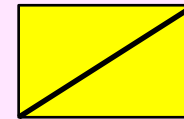
4. Insert middle



# 03603212 : Module3–List, stack, Queue 8

```
struct record
{
    int value;
    struct record *prev;
    struct record *next;
};
struct record *tail=NULL;
```

tail

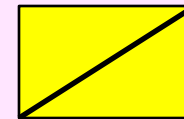


4000

```
int main()
{
    struct record *head=NULL;

    head=insert(head,8);
    hprint(head);
    tprint(tail);
}
```

head



2050

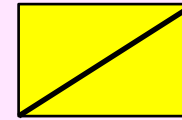




# 03603212 : Module3–List, stack, Queue 9

```
struct record
{
    int value;
    struct record *prev, *next;
};
struct record *tail=NULL;
```

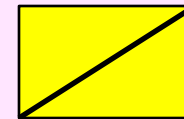
tail



4000

```
struct record *insert(struct record *head,int data)
{
    ....
    return head;
}
```

head



2050

```
int main()
{
    struct record *head=NULL;
    head=insert(head,8);
    hprint(head);
    tprint(tail);
}
```

## 1. Insert while no data in list

```
struct record *insert(struct record *head, int data)
```

```
{
    .... ประกาศตัวแปรเอง...
```

```
1. if( head == NULL)
```

```
2. {   head=new struct rec;
```

```
3.     head->value= data;
```

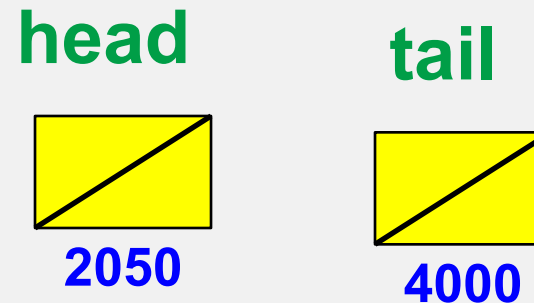
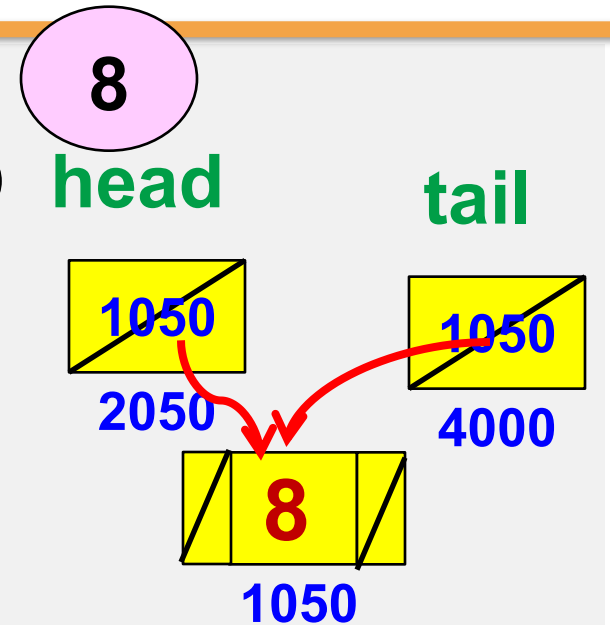
```
4.     head->next=NULL;
```

```
5.     head->prev=NULL;
```

```
6.     tail=head;
```

```
7. }
```

```
8. return head;
```

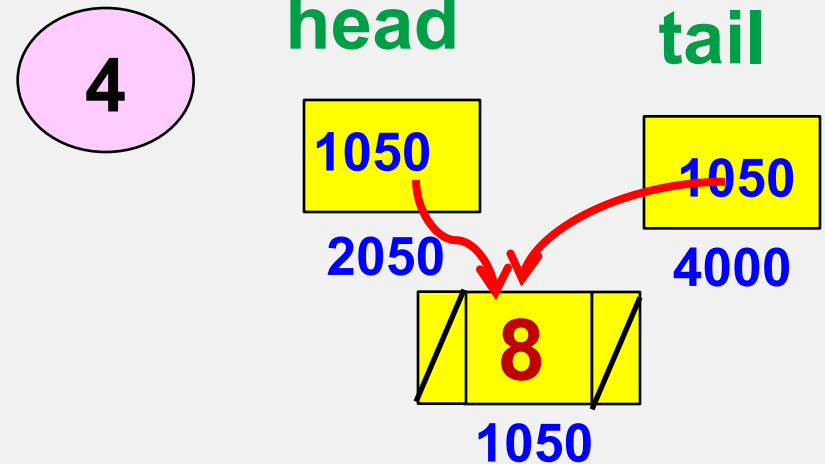


## 2. มีข้อมูลแล้วจะ Insert first

struct record \*insert(struct record \*head, int data)

{ .... ประกาศตัวแปรเอง...

1. if( head == NULL)
2. { head=new struct rec;
3. head->value= data;
4. head->next=NULL;
5. head->prev=NULL;
6. tail=head;
7. }
8. else



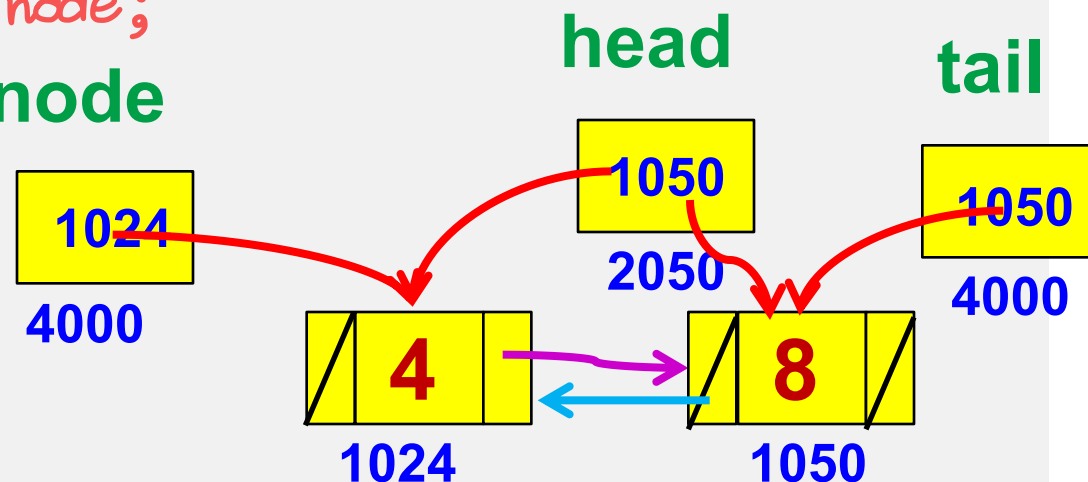
## 2. มีข้อมูลแล้วจะ Insert first (ต่อ)

```

8. else      node → value = data;
9. {   node = สร้าง node เก็บข้อมูลใหม่
10.   if (data <= head->value ) //แทรกหน้า
      {
        node->next = head;
        head->prev = node;
        head = node;
        node->prev = NULL;
      }

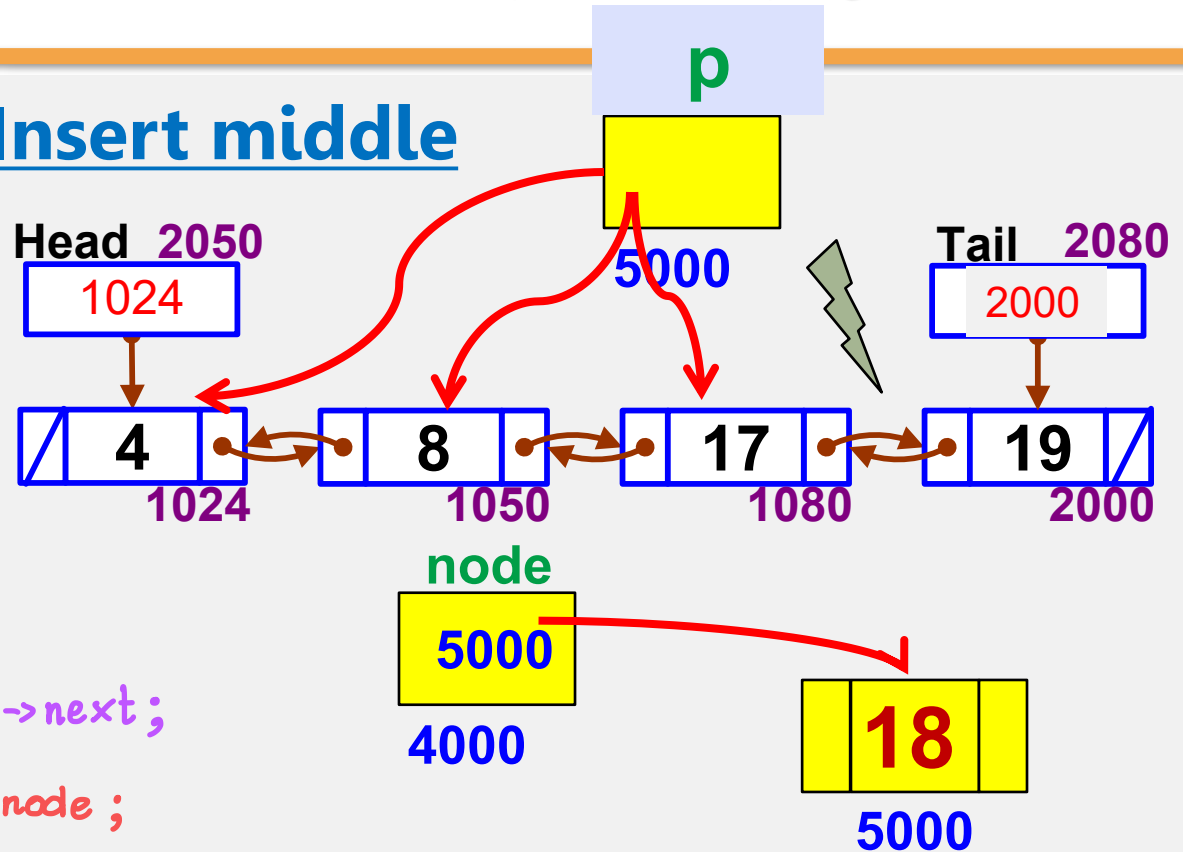
```

4



### 3. มีข้อมูลแล้วจะ Insert middle

18



else

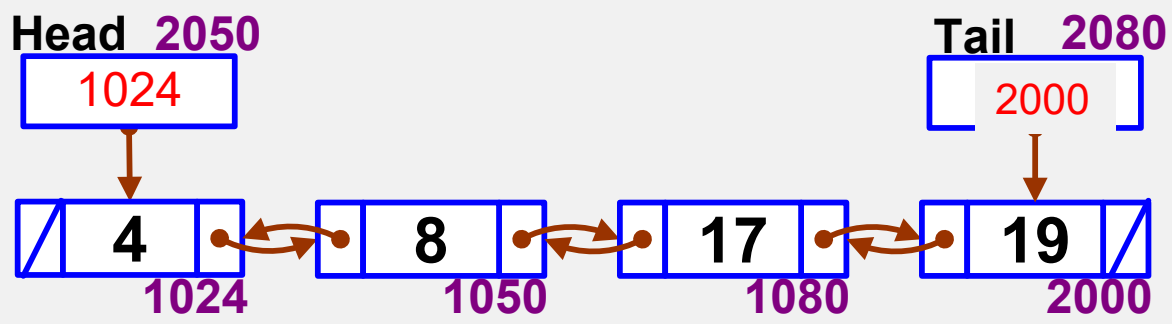
`node -> next = p -> next;`

`p -> next -> prev = node;`

`node -> prev = p;`

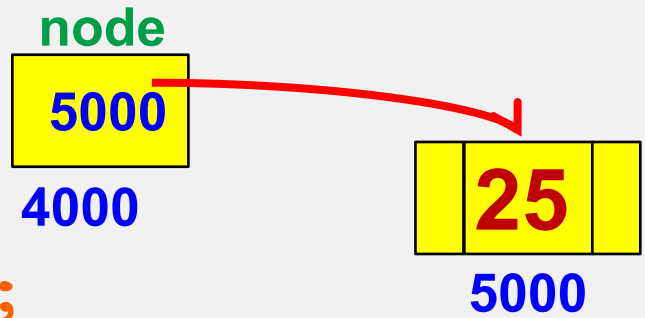
`p -> next = node;`

4. มีข้อมูลแล้วจะ Insert last



25

สร้าง node และใส่ค่า  
if(data > tail->value)



```

node -> prev = p;
p->next  = node;
node->next = NULL;
tail = node;

```

## การบ้าน

### 3. Doubly Linked list

=====

#### MENU

=====

- 1) Insert
- 2) Print min to max and max to min
- 3) Exit

Please choose >

ทำการ insert โดยเลือกข้อ 1

Insert : 5

List = 5

จากนั้นให้กลับไปเมนู

ให้ทดลอง insert เพิ่มทีละตัวคือ 3 1 10 8 9

ถ้าเลือกข้อ 2

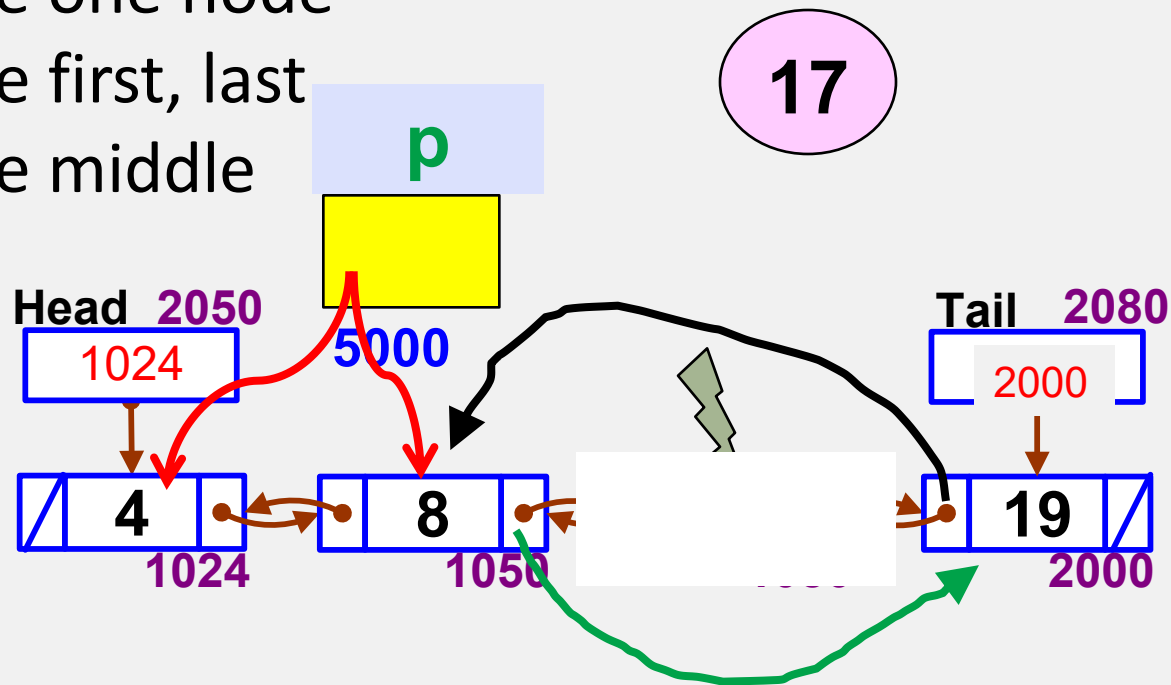
Min to max : 1 3 5 8 9 10

Max to min : 10 9 8 5 3 1

```
struct record *tmp,  
tmp = tail ;  
while ( tmp != NULL) {  
    cout << p->value << " ";  
    p = p->prev ;  
}
```

## 3.2.2 Delete (Doubly Linklist)

1. Delete while no data in list
2. Delete one node
3. Delete first, last
4. Delete middle



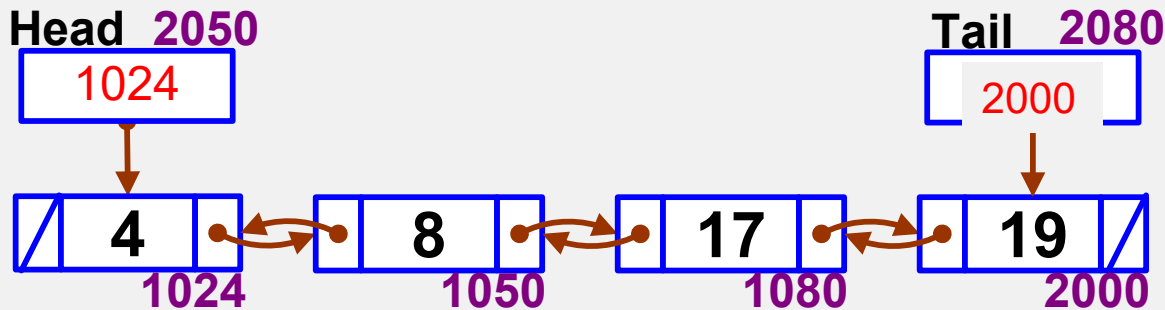


## โครงโปรแกรมในการ delete

```
if(มี node เดียว ?)
{
    ....
}
else
{
    if ( ลบโหนดแรก ? )
    {
        .....
    } else if(ลบโหนดท้าย? )
    {
        ....
    }
    else //ลบตรงกลาง
    {
        หาตำแหน่ง
        .....
    }
}
```

## 1. Delete first

4



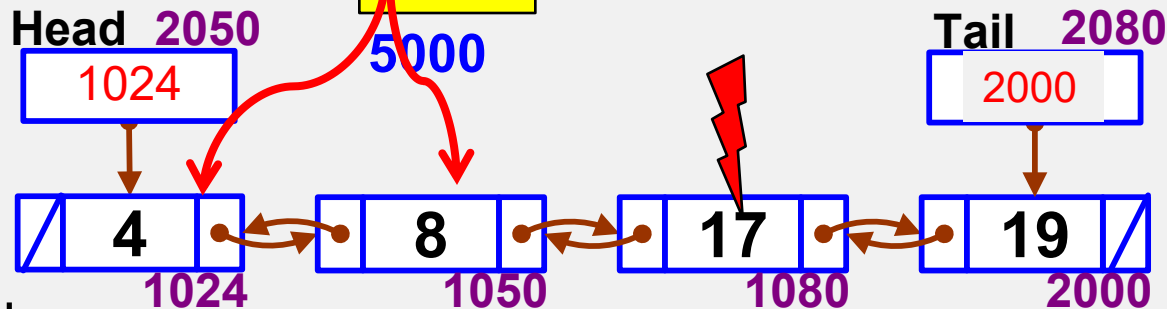
```
if( data == head->value)
{
    tmp=head;
    head=head->next;
    head->prev=NULL;
    delete(tmp);
}
```

## 2. Delete middle ส่วน delete last ให้ทำเอง

17

p

$p \rightarrow next \rightarrow next$



```

p = head;
while(p->next != NULL)
{
    if( data == tail->value){
        temp = tail;
        tail = tail->prev;
        delete (tmp);
        tail->next = NULL;
    }
}

```

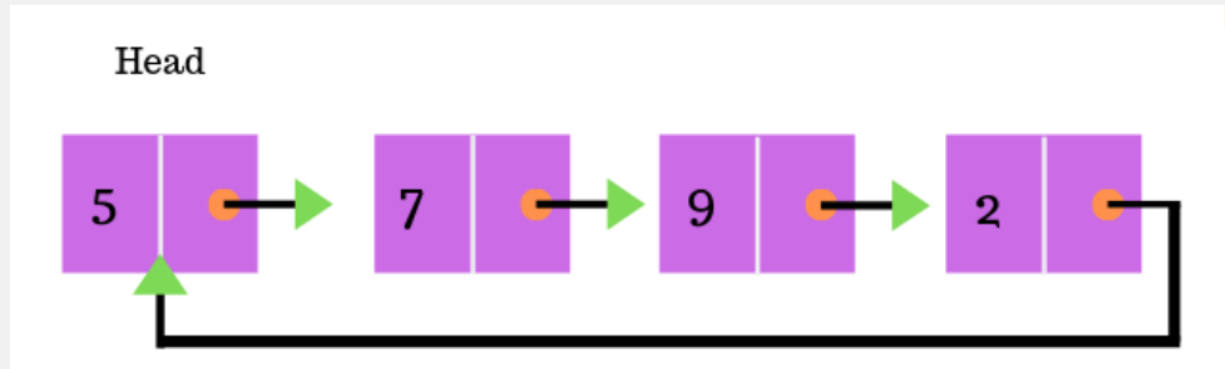
```

tmp = p -> next ;
p->next->next->prev = p ;
p->next = p->next->next ;
delete (tmp);

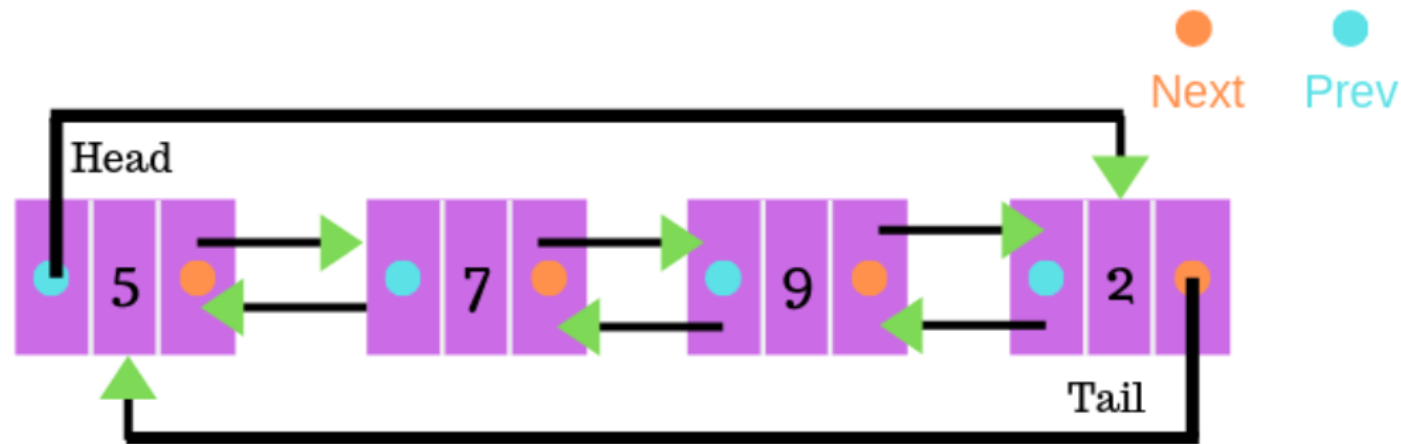
```

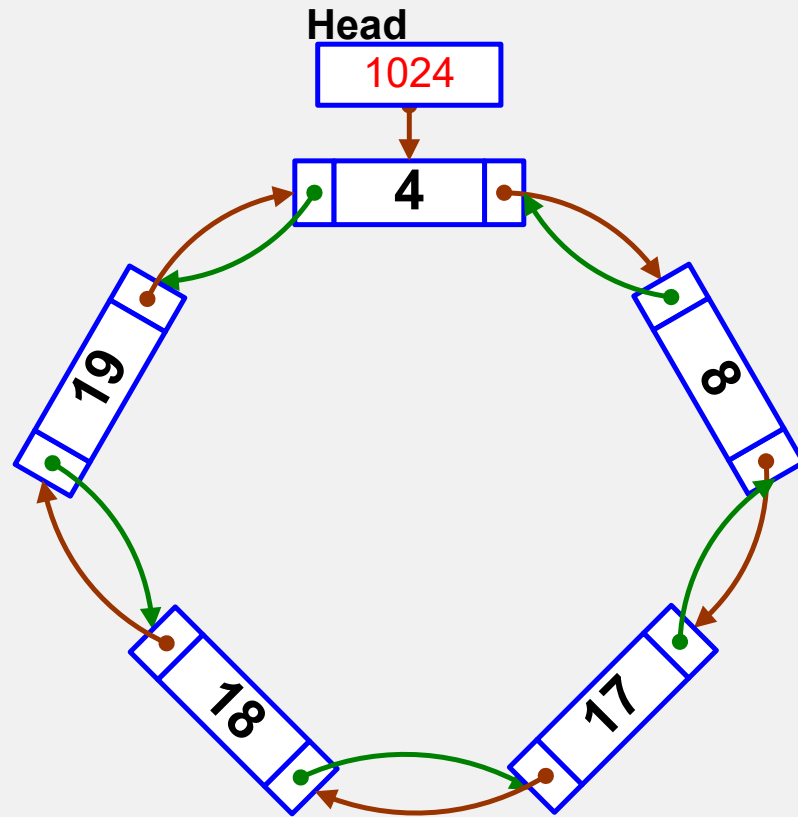
### 3.3 Circularly linked list and circular doubly linked lists

**Circular linked list** คือ linked list ชนิดหนึ่งที่แตกต่างกันจากการโยงปกติ ตรงที่โหนดสุดท้ายของรายการไม่ได้ชี้ไปยังค่า NULL แต่จะชี้กลับไปยังโหนดแรกของ list ทำให้เกิดลักษณะเป็นวงกลม



**Circularly doubly linked lists** : A popular convention is to have the last cell keep pointer back to the first. This can be done with or without header (If the header present, the last cell point to it.)





### การบ้าน

4. เก็บเลขจำนวนเต็มขนาดไม่จำกัด ให้นี้สื่รับข้อมูลเป็นเลขจำนวนเต็มขนาดไม่จำกัด (ตัวเลขนี้ใหญ่มากจนไม่สามารถเก็บลงในตัวแปร int, float, long double ได้

ให้รับเลขและพิมพ์ข้อมูลออกทางจอภาพ

Input : 123456789012345678901234567890

Output : 123456789012345678901234567890

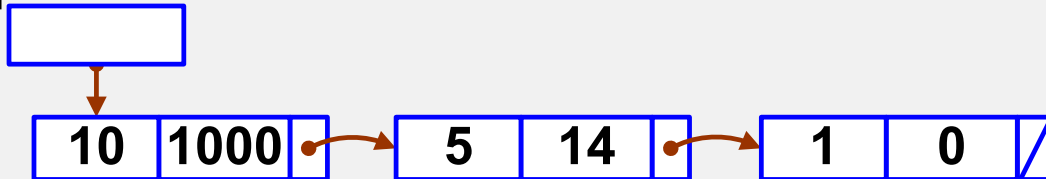
## 3.4 Examples

### 3.4.1 The polynomial ADT

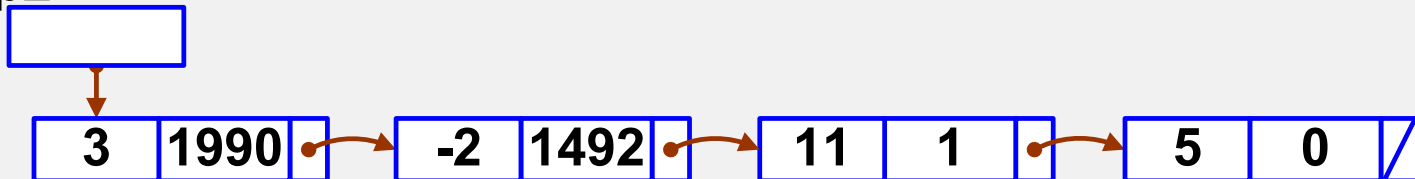
$$p1(x) = 10x^{1000} + 5x^{14} + 1$$

$$p2(x) = 3x^{1990} - 2x^{1492} + 11x + 5$$

p1



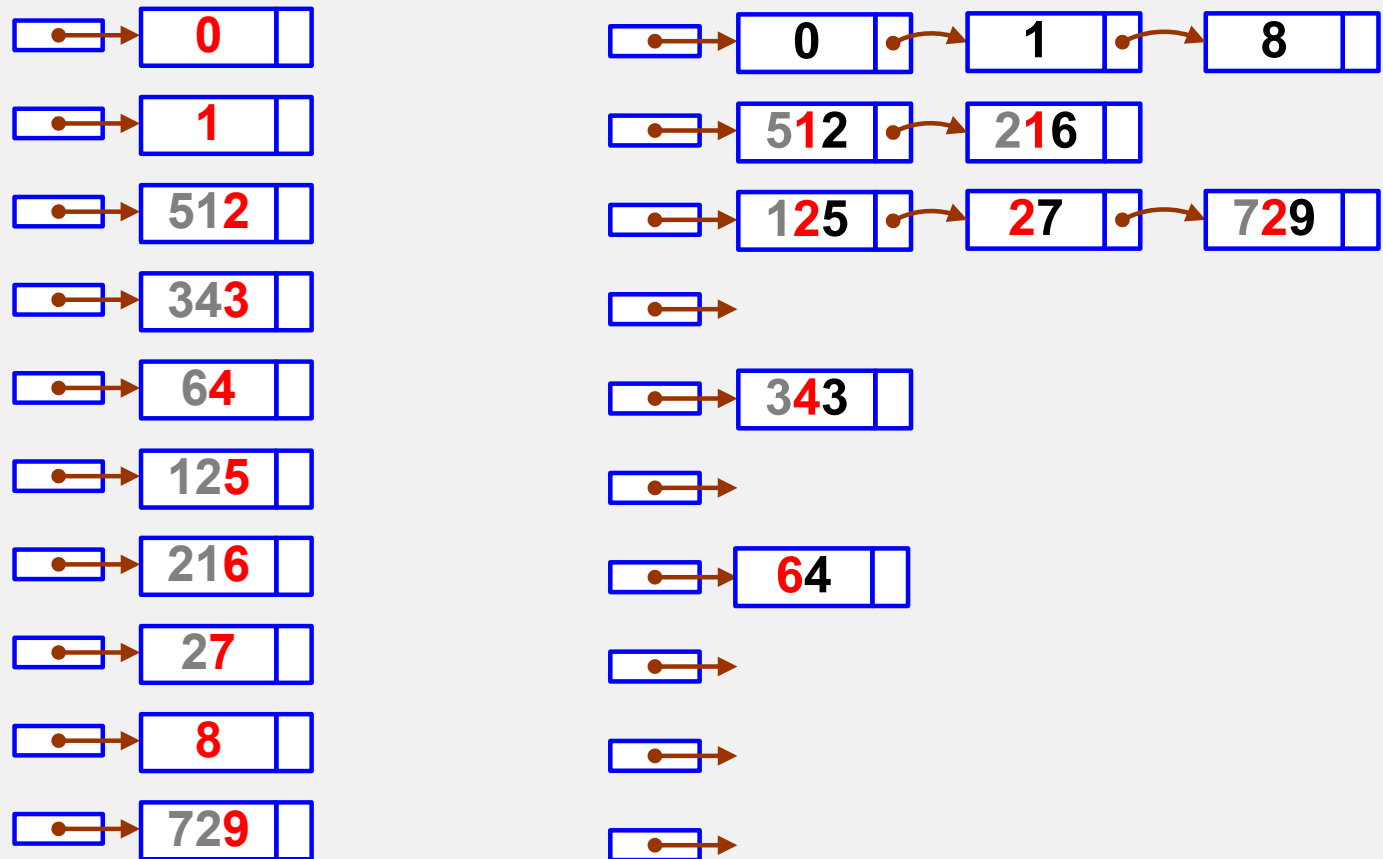
p2





## 3.4.2 Radix Sort

Input 64, 8, 216, 512, 27, 729, 0, 1, 343, 125



### 3.4.3 Multilists

