

Linked List - Traversing

trav = head;

① while (trav != null)

→ trav = null

② while (trav.next != null)

→ trav = last node

while (trav.next.next != null) → trav = second last node

Node trav;

① for (trav = head; trav != null; trav = trav.next);

② for (trav = head; trav.next != null; trav = trav.next);

i = 1;

trav = head;

while (i < pos - 1) {

trav = trav.next;

i++;

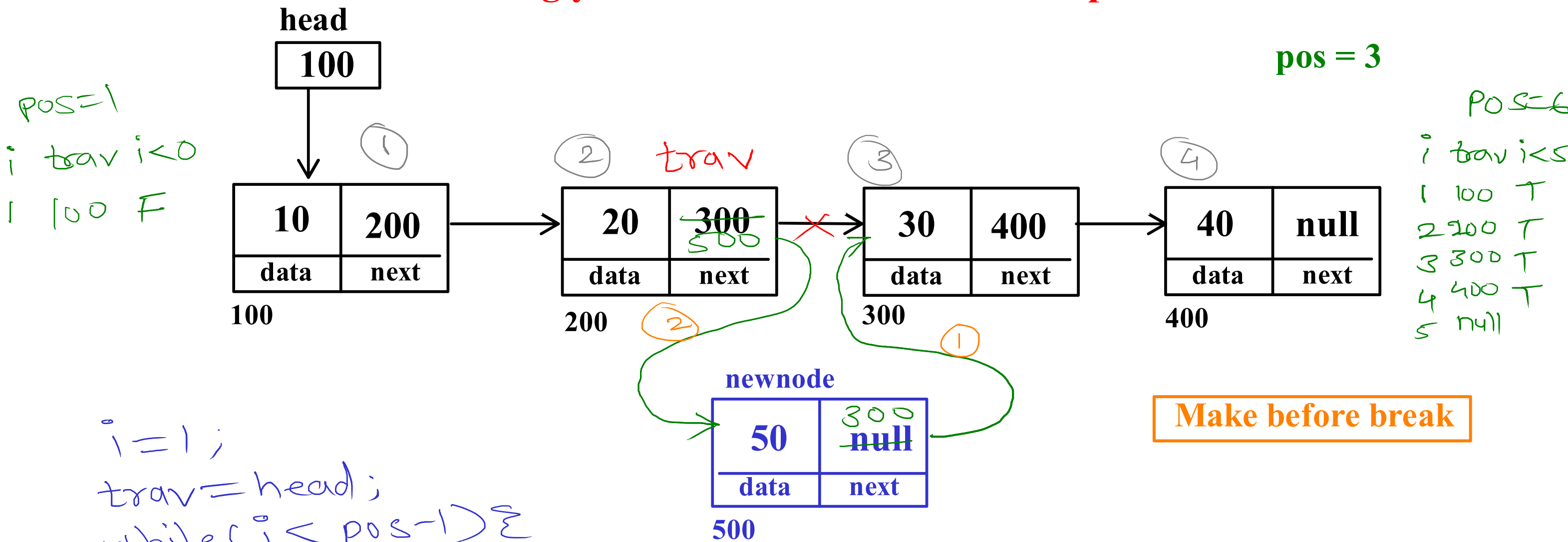
}

Node trav = head;

for (int i = 1; i < pos - 1; i++)

trav = trav.next;

Singly Linear Linked List - Add at position



```
i = 1;
trav = head;
while(i < pos - 1) {
    trav = trav->next;
    i++;
}
```

| | | | | | |
|---------|------|-------|---------|------|-------|
| pos = 3 | | | pos = 4 | | |
| i | trav | i < 2 | i | trav | i < 3 |
| 1 | 100 | T | 1 | 100 | T |
| 2 | 200 | F | 2 | 200 | T |
| | | | 3 | 300 | F |

//1. create node with given value

//2. if list is empty

//a. add newnode into head

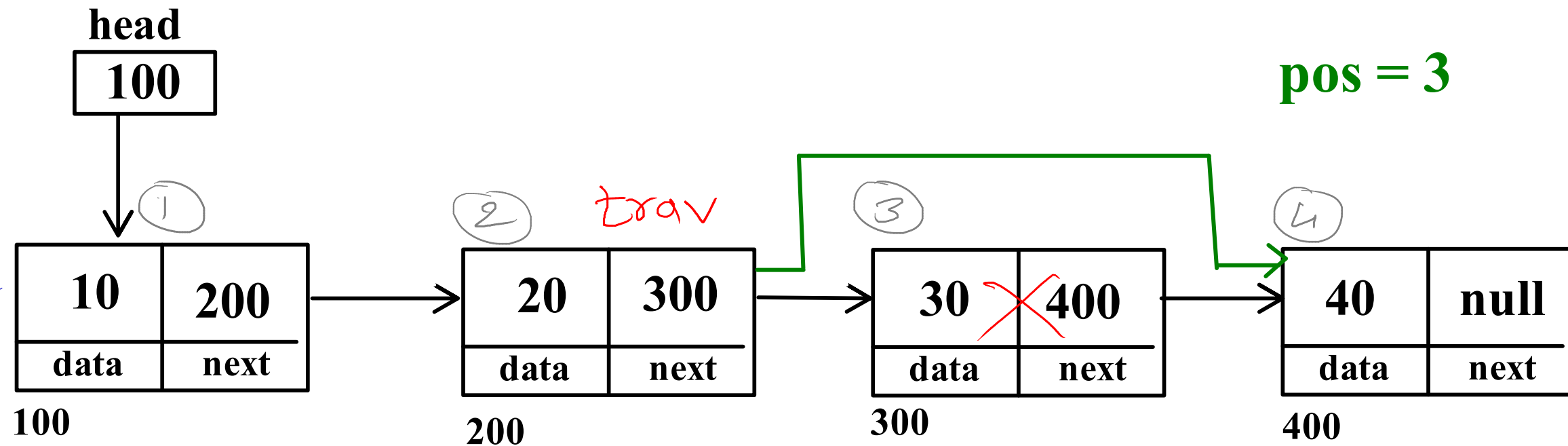
//3. if list is not empty

//a. traverse till pos - 1 node

//b. add pos node into next of newnode

//c. add newnode into next of pos-1 node

Singly Linear Linked List - Delete at position



//1. if list is empty

// do nothing

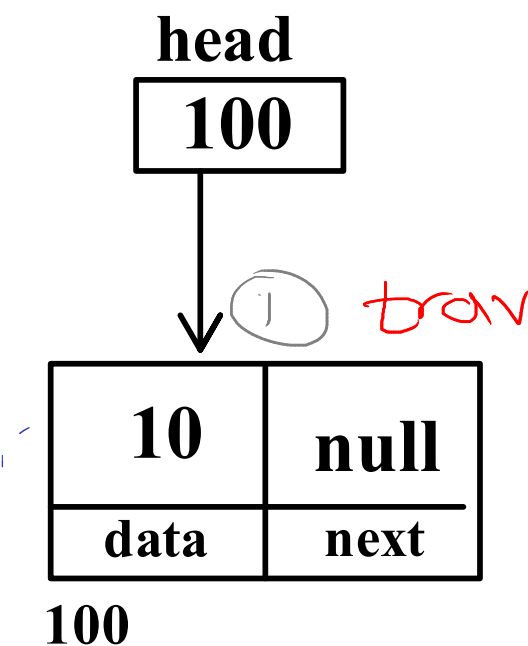
//2. if list has single node

//a. make head equal to null

//3. if list has multiple

//a. traverse till pos - 1 node

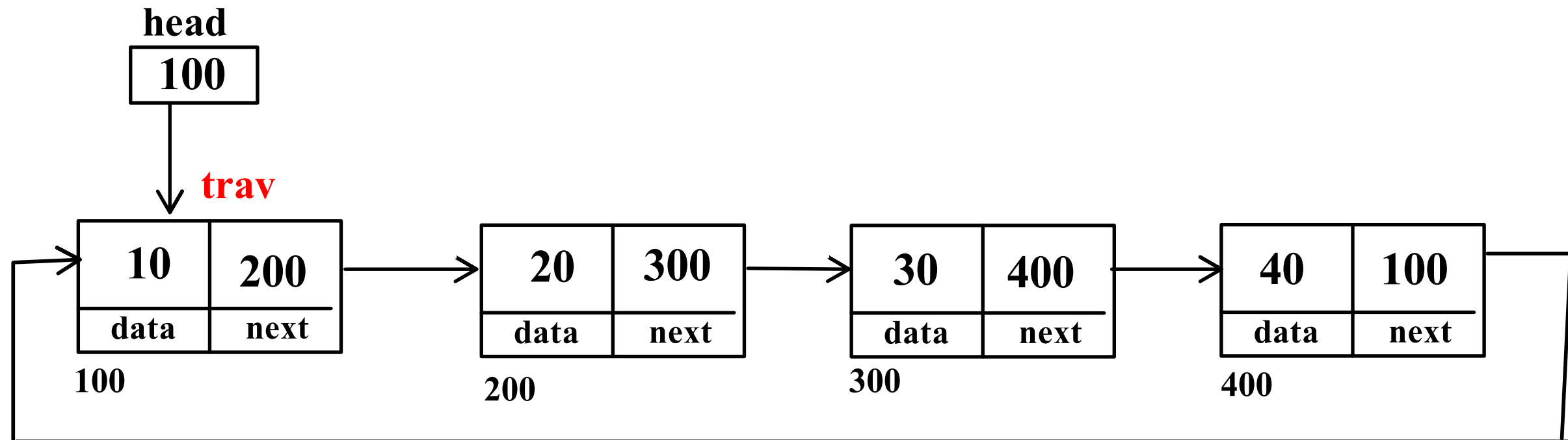
//b. add pos+1 node into next of pos-1 node



Node trav = head;
 For (i = 1; i < pos - 1; i++)
 trav = trav.next;

→ trav.next = trav.next.next;

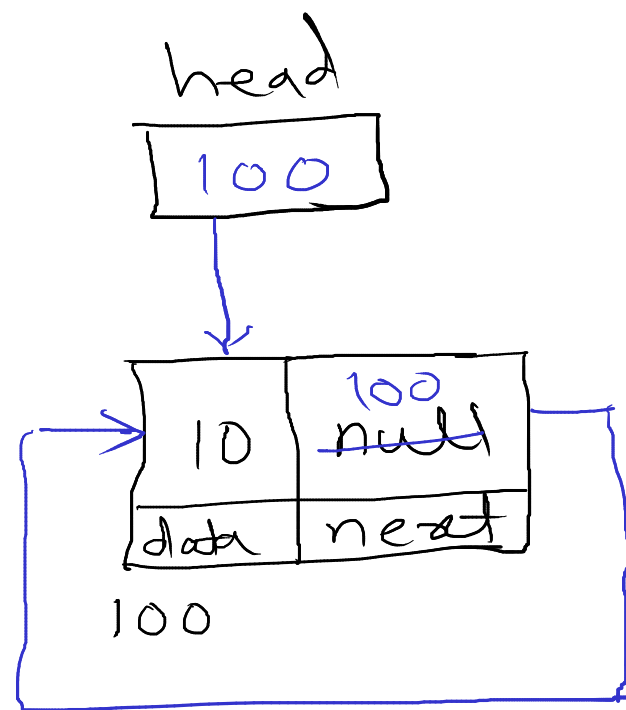
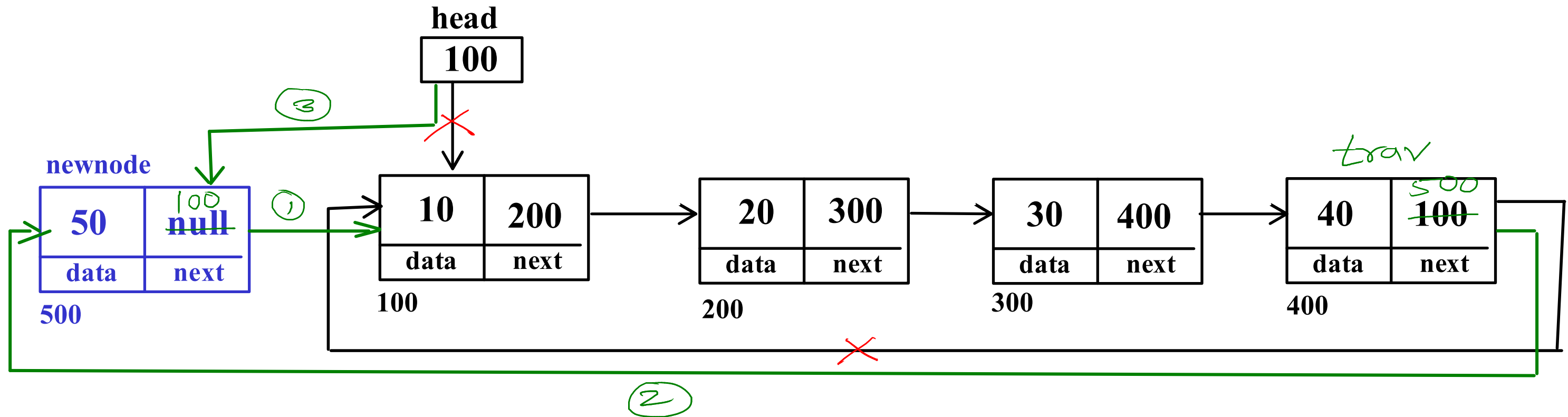
Singly Circular Linked List - Display



```
trav = head;  
do  
{  
    printf(trav->data);  
    trav = trav->next;  
} while(trav != head);
```

- //1. create trav pointer and start at head
- //2. print data of current node (trav)
- //3. go on next node
- //4. repeat step 2 and 3 till last node

Singly Circular Linked List - Add First



//1. create node with given value

//2. if list is empty

//a. add newnode into head

//b. make list circular

//3. if list is not empty

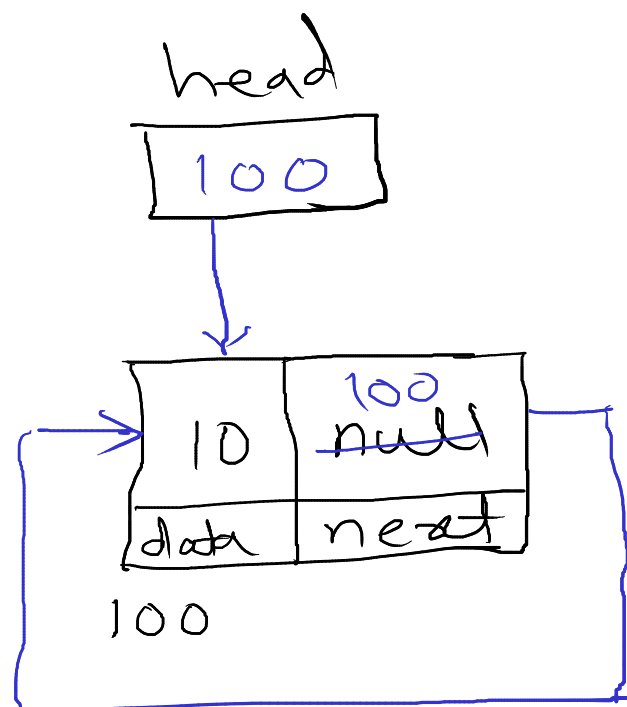
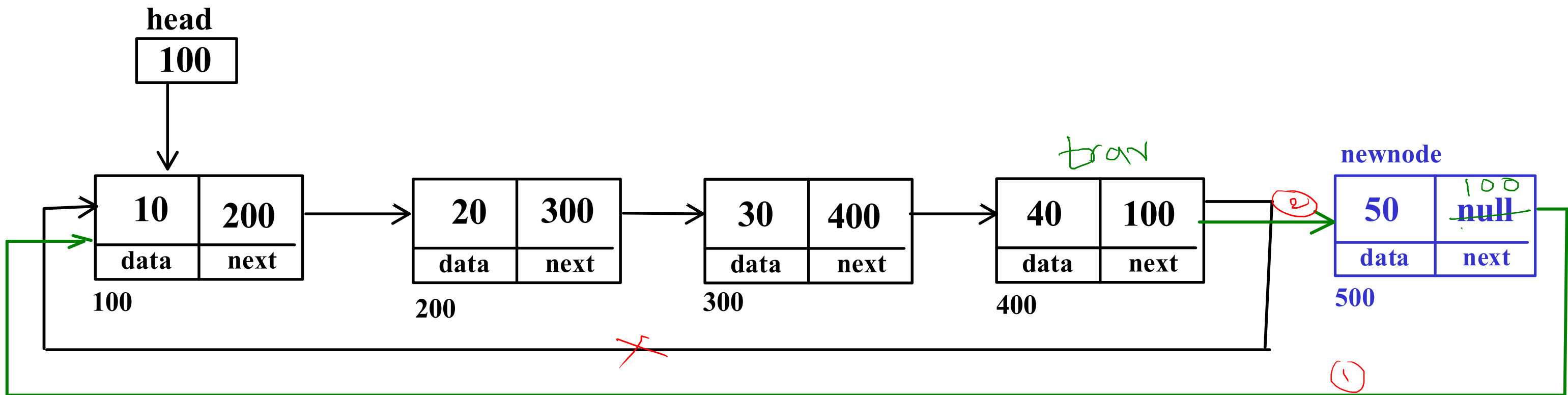
//a. add first node into next of newnode

//b. traverse till last node

//c. add newnode into next of last node

//d. move head on newnode

Singly Circular Linked List - Add last



//1. create node with given value

//2. if list is empty

//a. add newnode into head

//b. make list circular

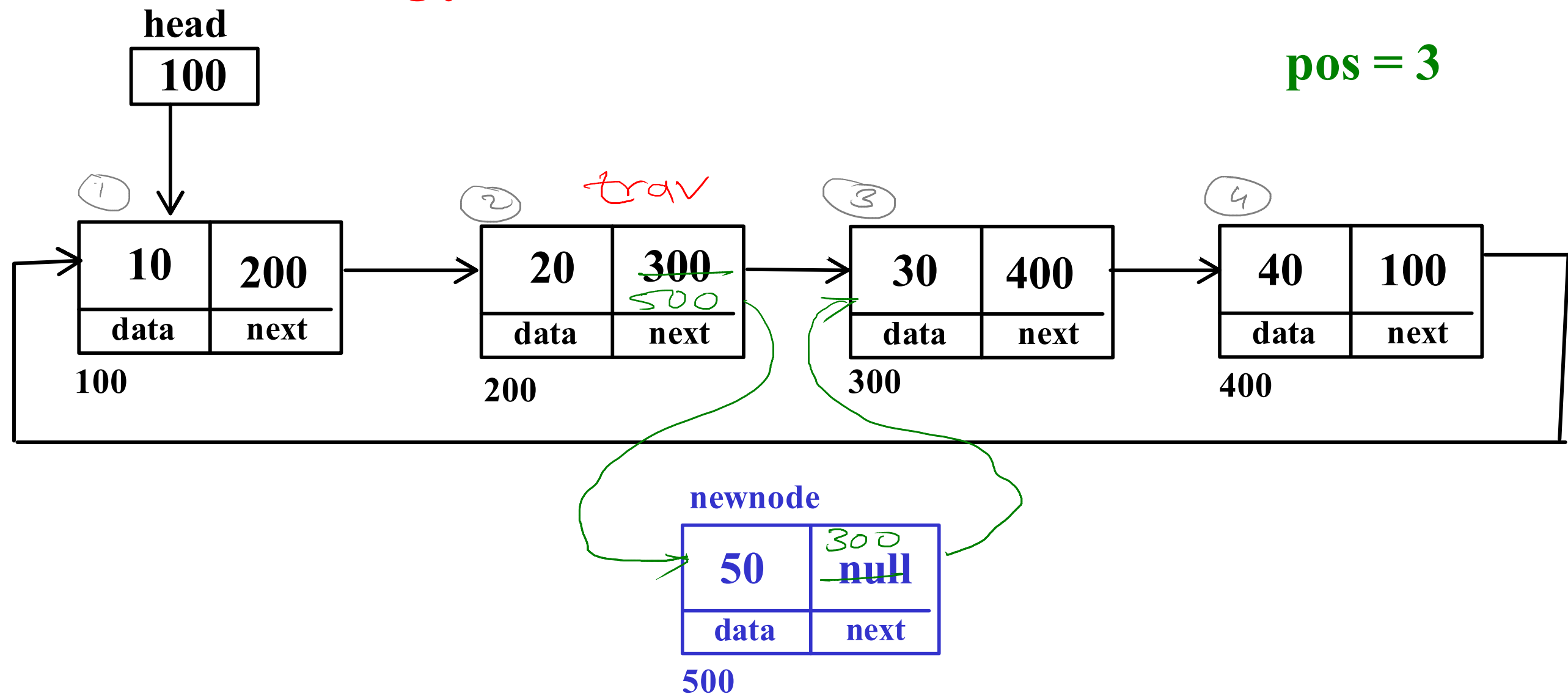
//3. if list is not empty

//a. add first node into next of newnode

//b. traverse till last node

//c. add newnode into next of last node

Singly Circular Linked List - Add at Position



//1. create node with given value

//2. if list is empty

//a. add newnode into head

//b. make list circular

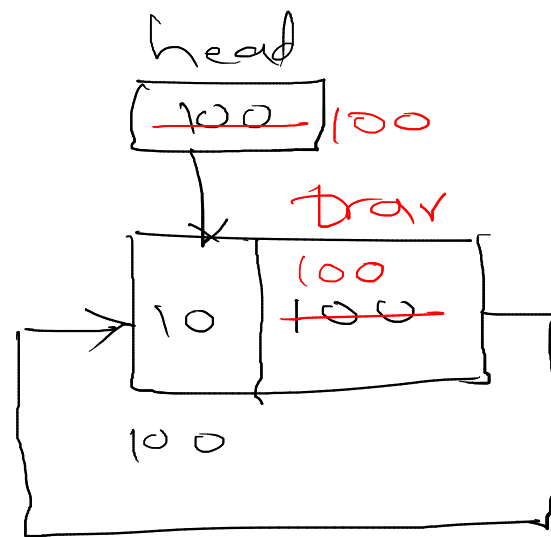
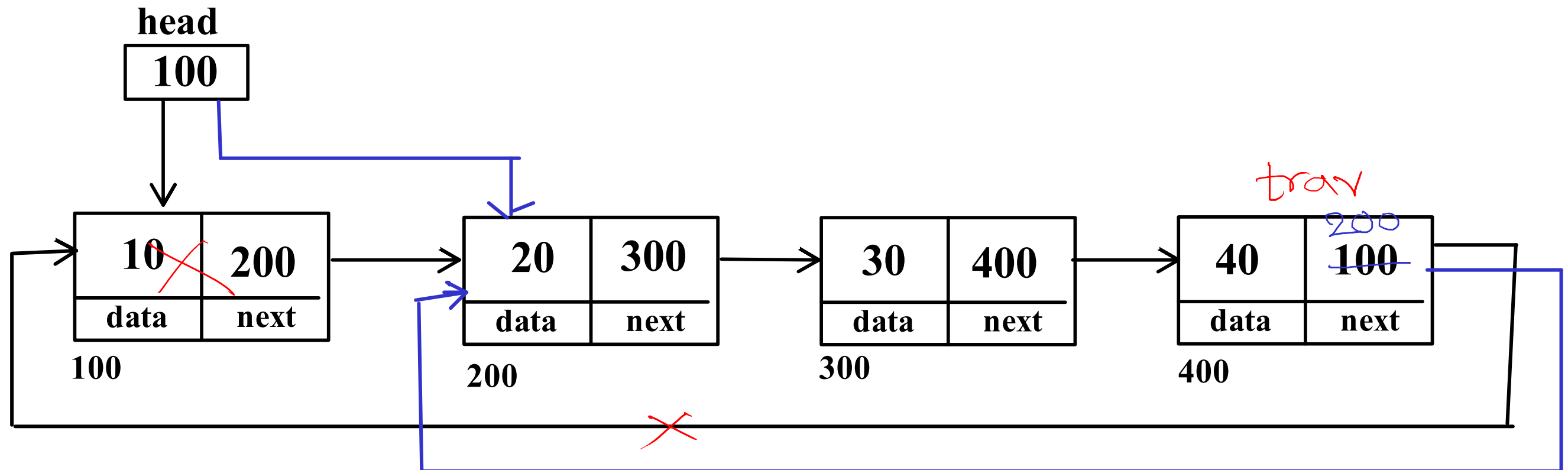
//3. if list is not empty

//a. traverse till pos - 1 node

//b. add pos node into next of newnode

//c. add newnode into next of pos-1 node

Singly Circular Linked List - Del First



//1. if list is empty

// do nothing

//2. if list has single node

//a. make head = null

//2. if list multiple nodes

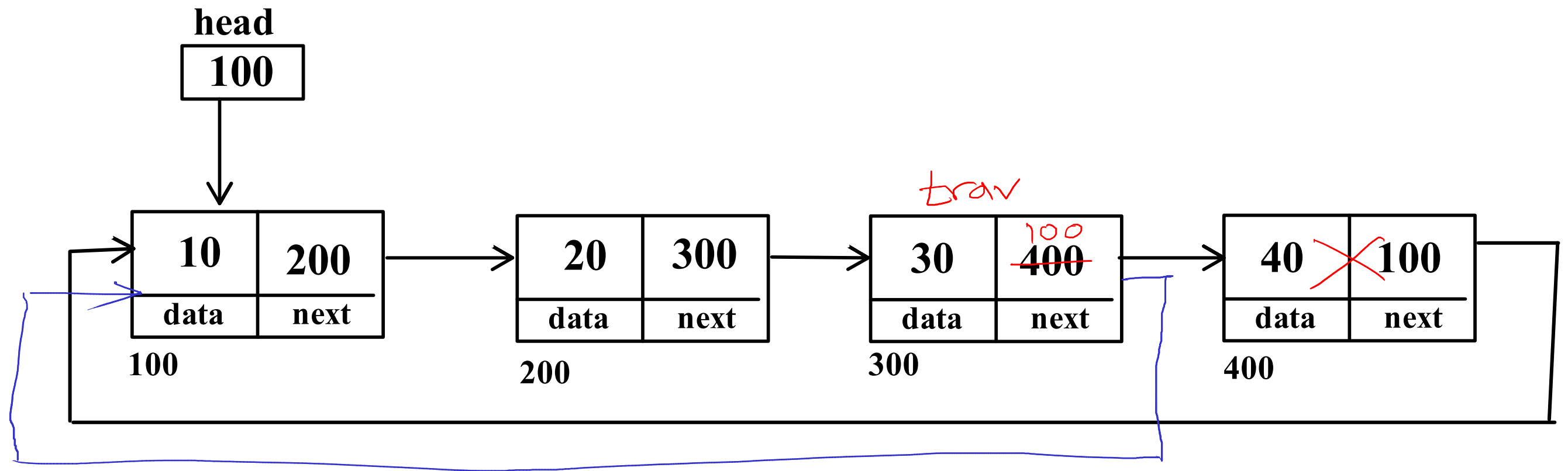
//a. traverse till last node

//b. add second node into next of last node

//c. move head on second node

```
trav = head;  
while (trav.next != head) {  
    trav = trav.next;  
}  
trav.next = head.next;  
head = head.next;
```


Singly Circular Linked List - Del Last



//1. if list is empty

// return

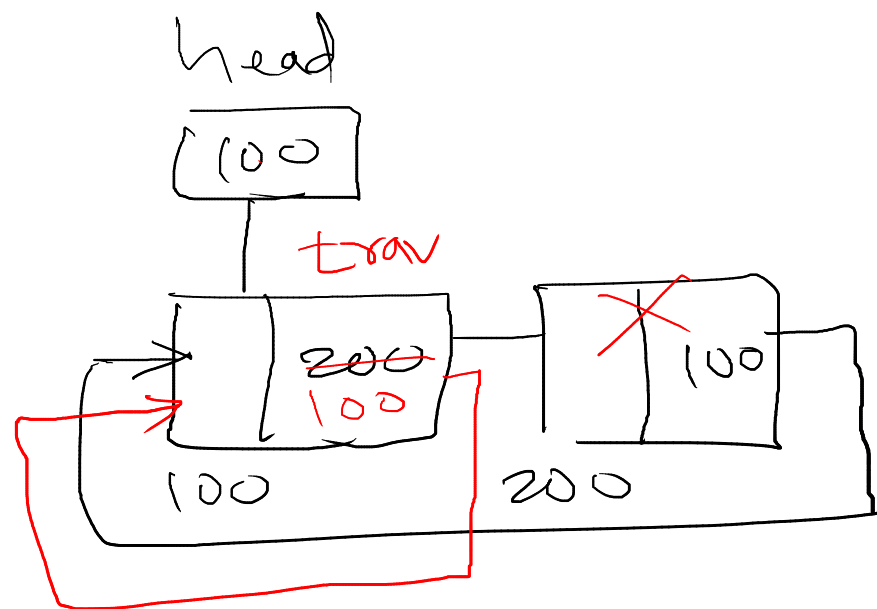
//2. if list has single node

//a. make head = null

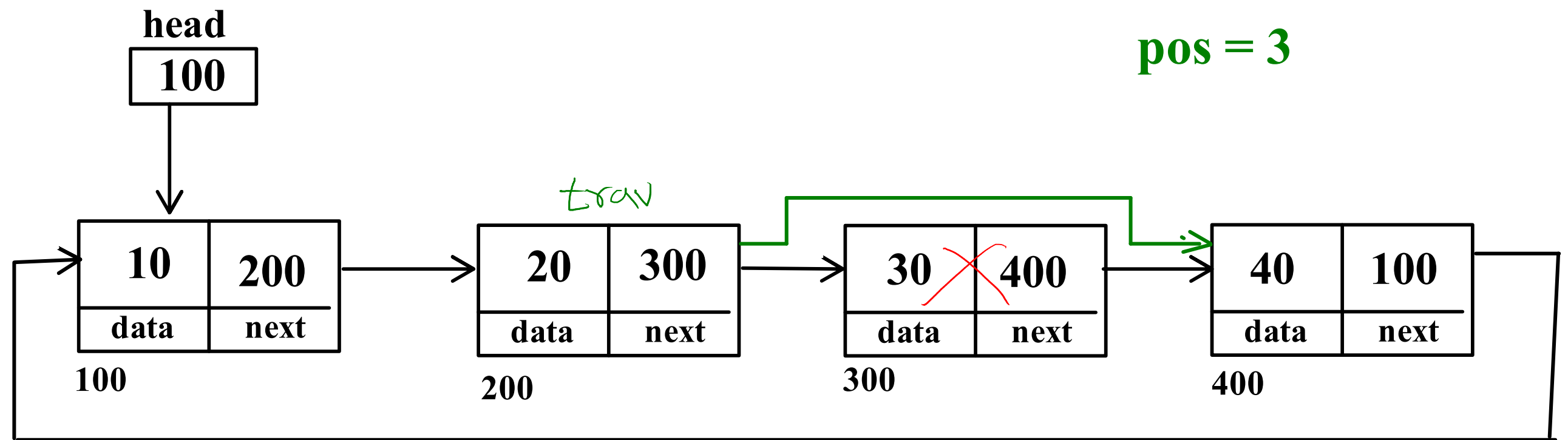
//3. if list has multiple nodes

//a. traverse till second last node

//b. add first node into next of second last node



Singly Circular Linked List - Del pos



//1. if list is empty

// do nothing

//2. if list has single node

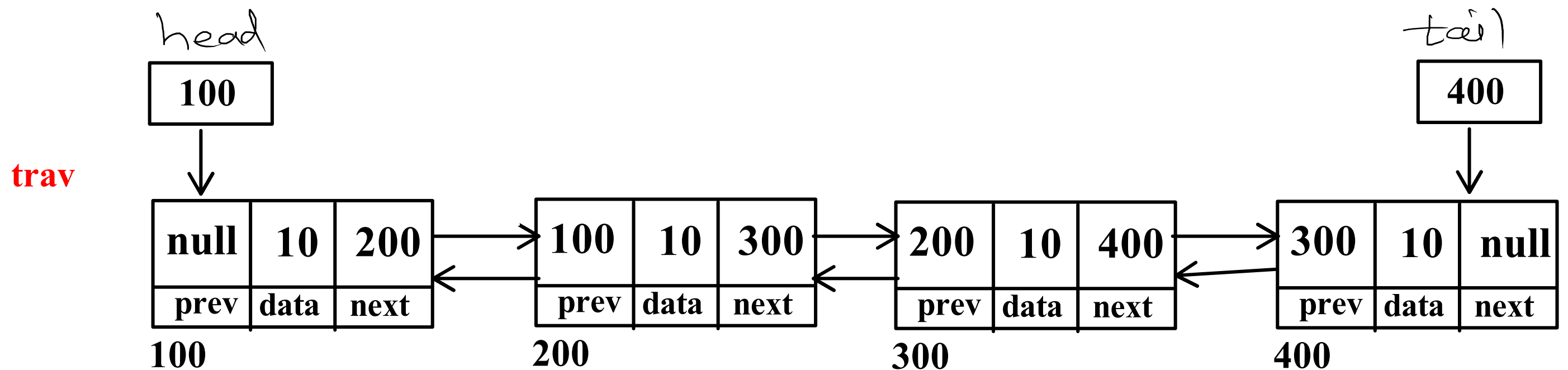
//a. make head = null

//3. if list has multiple nodes

//a. traverse till pos -1 node

//b. add pos + 1 node into next of pos - 1 node

Doubly Linear Linked List - Display



// forward display

//1. create trav reference and start at head

//2. print data of current node

//3. go on next node

//4. repeat step 2 and 3 till last node

// backward display

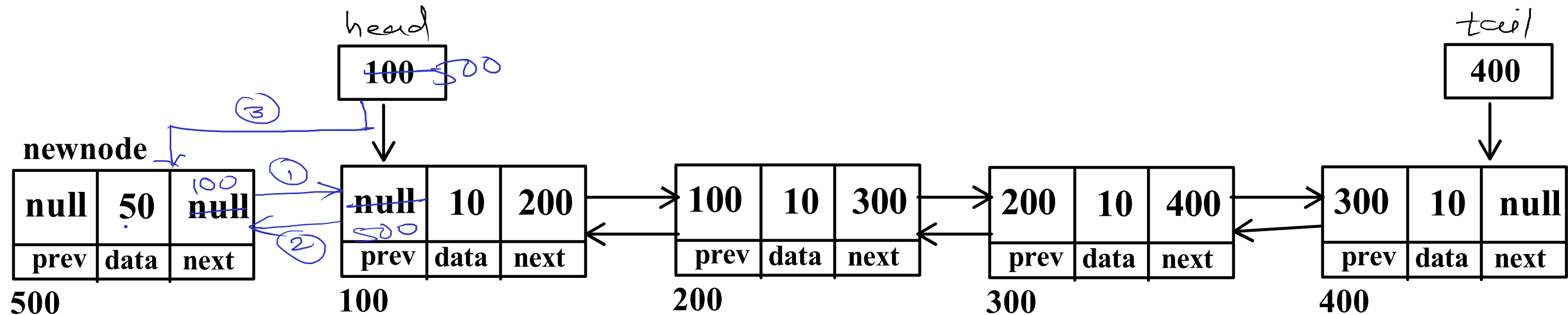
//1. create trav reference and start at tail

//2. print data of current node

//3. go on prev node

//4. repeat step 2 and 3 till first node

Doubly Linear Linked List - Add first



//1. create node with given value

//2. if list is empty

 //a. add newnode into head and tail

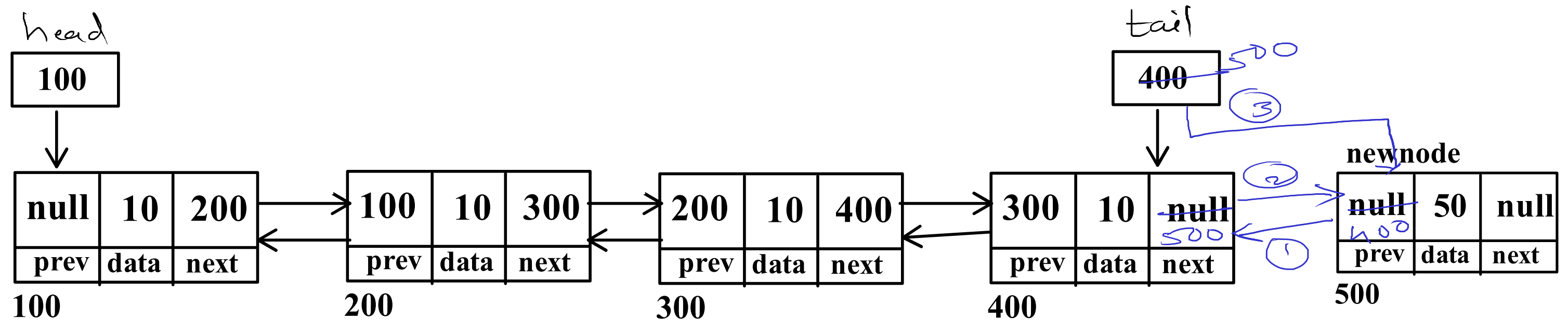
//3. if list is not empty

 //a. add first node into next of newnode

 //b. add newnode into prev of first node

 //c. move head on newnode

Doubly Linear Linked List - Add Last



//1. create node with given value

//2. if list is empty

//a. add newnode into head and tail

//3. if list is not empty

//a. add last node into prev of newnode

//b. add newnode into next of last node

//c. move tail on newnode