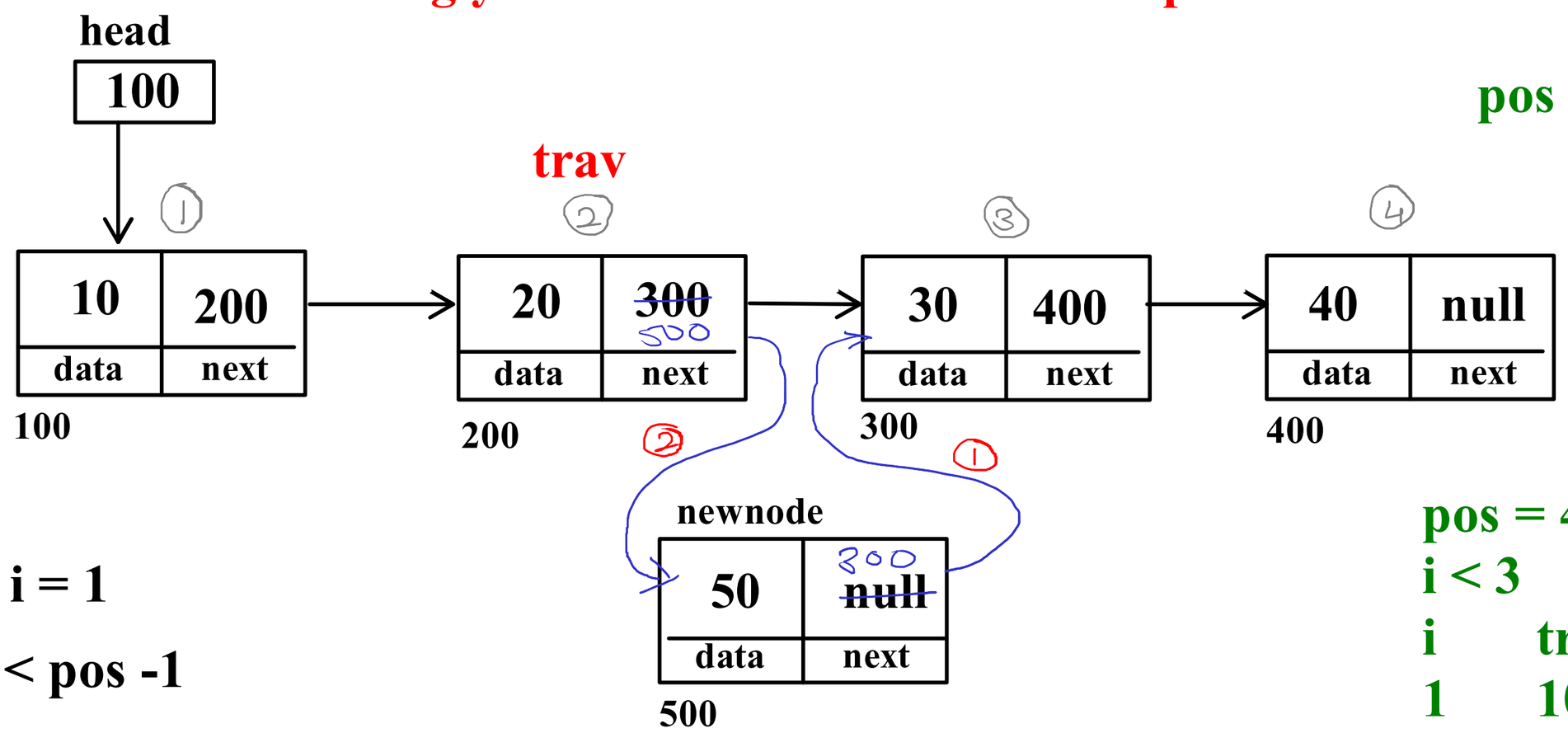
Make before break

Singly Linear Linked List - Add at position



i = 1

i < pos -1

i < 2

i	trav
1	100
2	200

pos = 4

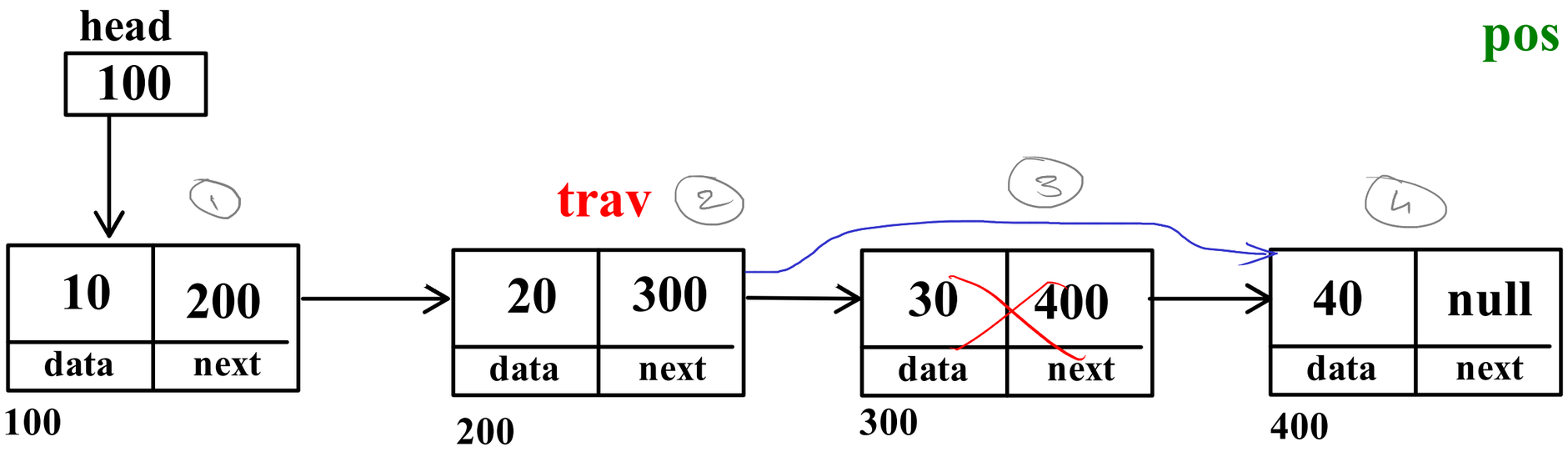
i < 3

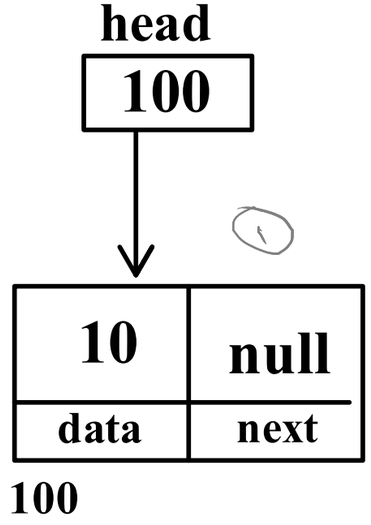
i	trav
1	100
2	200
3	300

//1. Create node with given data
//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

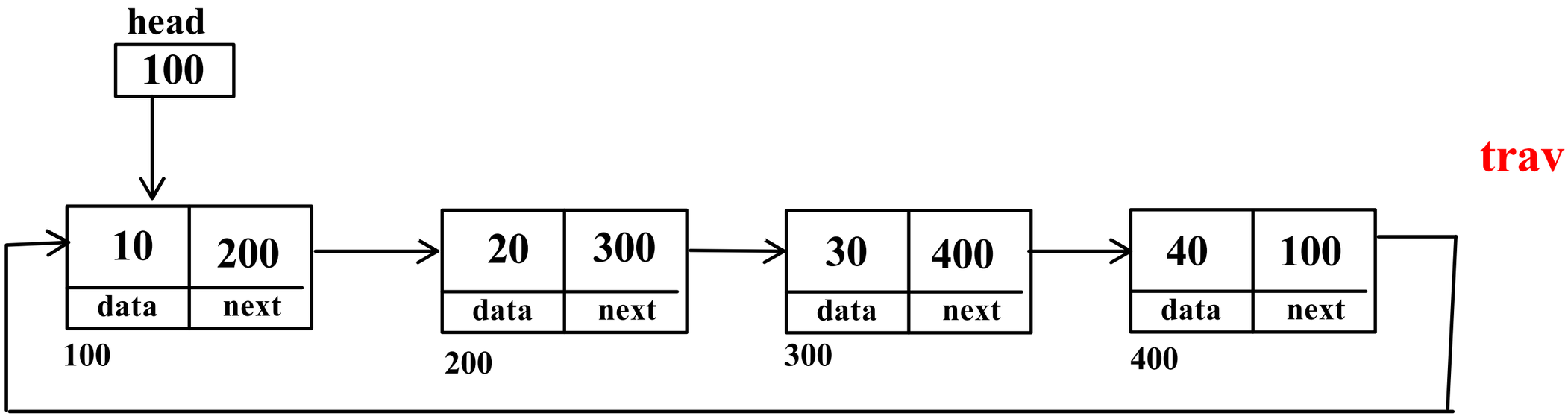
pos = 3






```
//1. if list is empty then return
//2. if list has single node
        // make head equal to null
//3. if list has multiple nodes
        //a. traverse till pos - 1 node
        //b. add pos + 1 node into next of pos - 1 node
```

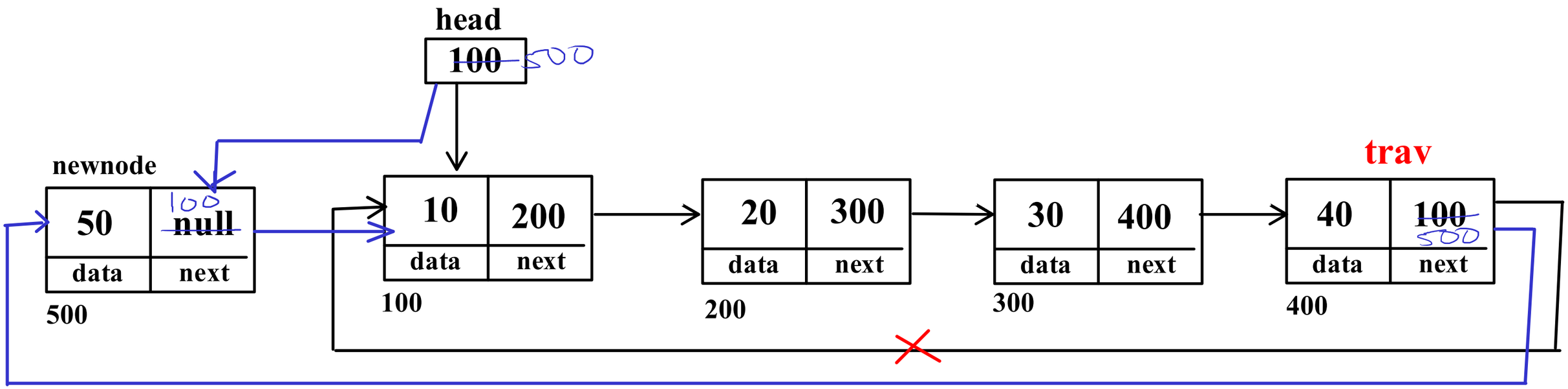
Singly Circular Linked List - Display

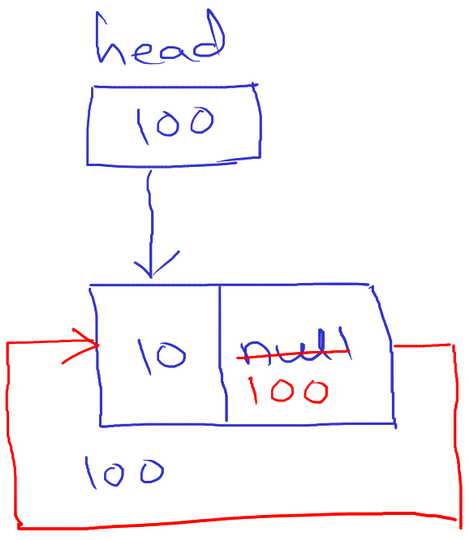


//1. create one trav referance and start it from head
//2. print data of current node(trav)
//3. go on next node(trav.next)
//4. repeat step 2 and 3 untill trav != head

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

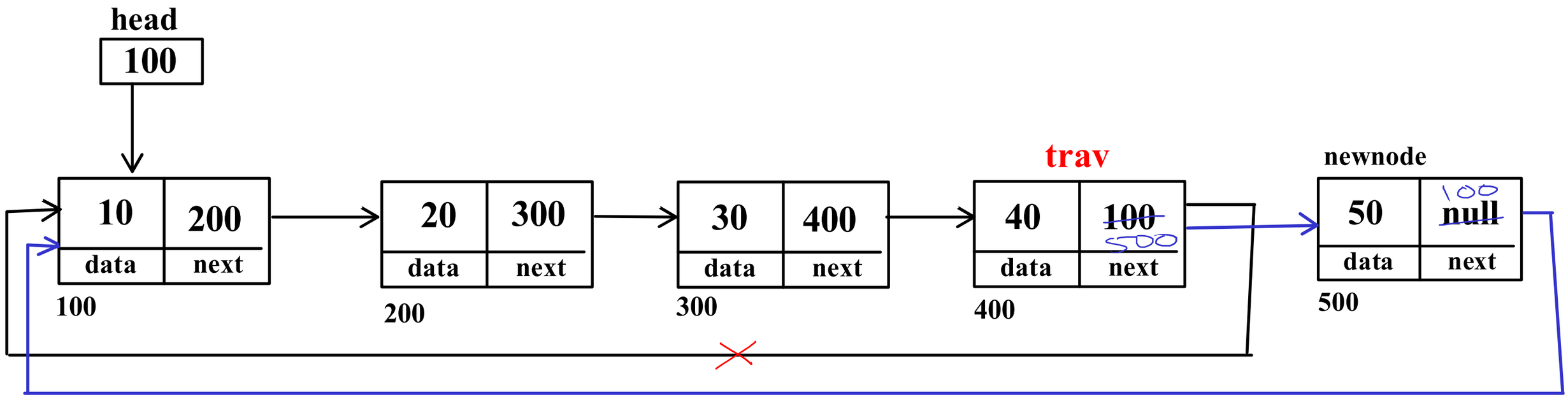
Singly Circular Linked List - Add First





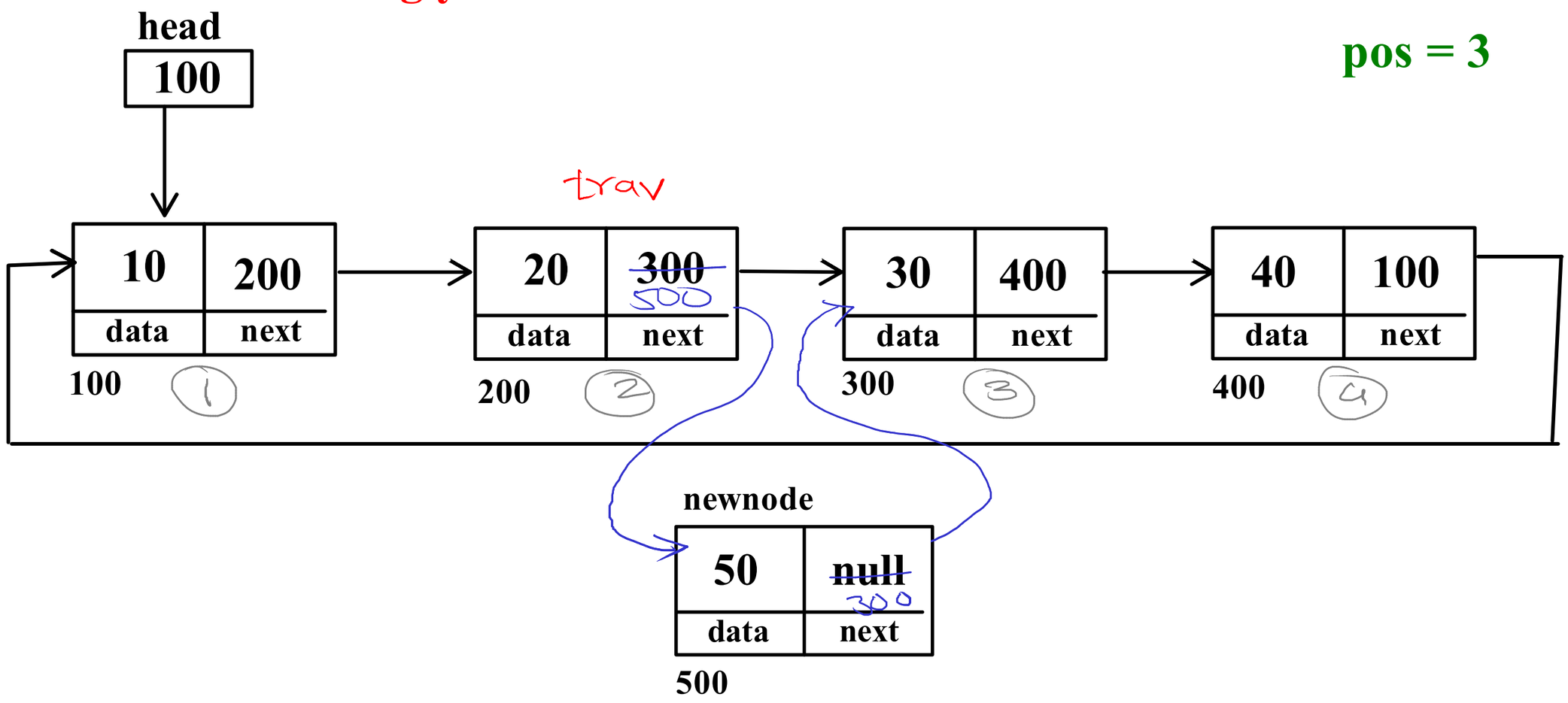

```
//1. create node with given data
//2. if list is empty
      //a. add newnode into head
      //b. make list circular
//3. if list is not empty
      //a. add first node into next of newnode
      //b. traverse till last node
      //c. add newnode into next of last node
      //d. move head on newnode
```

Singly Circular Linked List - Add last



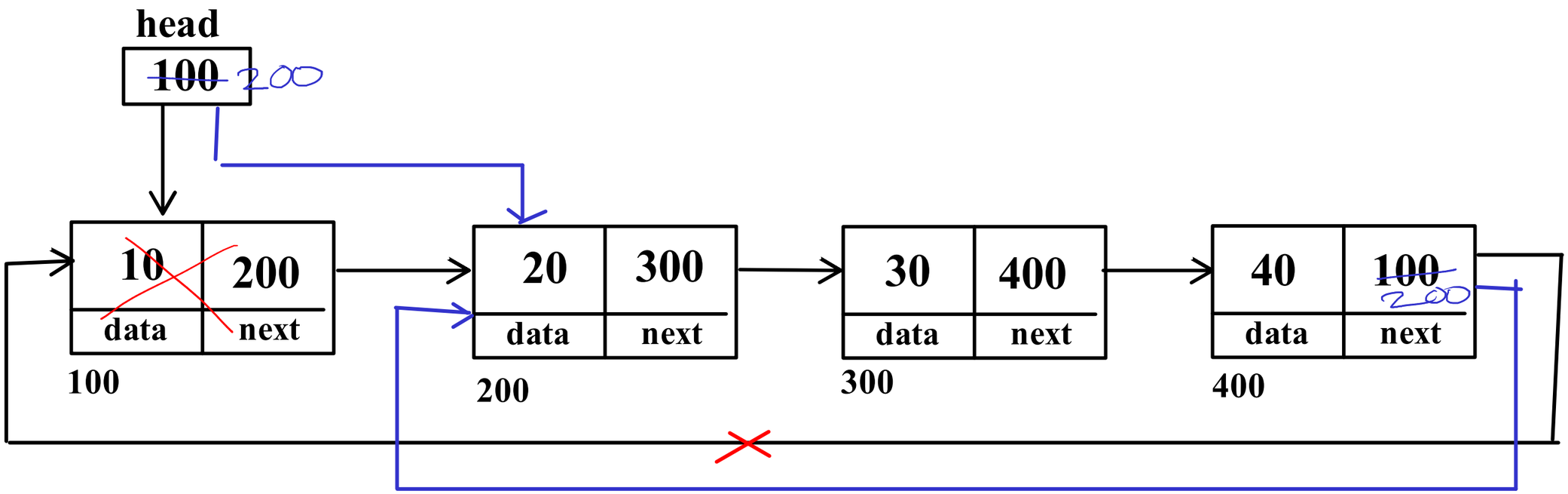

```
//1. create node with given data
//2. if list is empty
        //a. add newnode into head
        //b. make list circular
//3. if list is not empty
        //a. add first node into next of newnode
        //b. traverse till last node
        //c. add newnode into next of last node
```

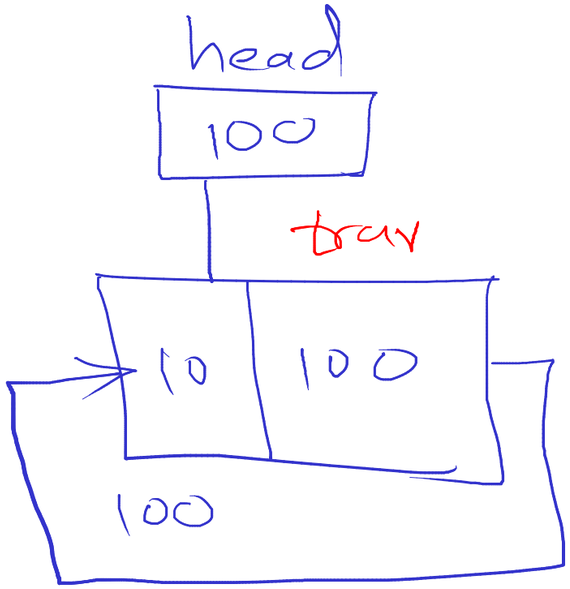
Singly Circular Linked List - Add at Position




```
//1. create node with given data
//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 + 1 node into next of newnode
        //c. add newnode into next of pos -1 node
```

Singly Circular Linked List - Del First





//1. if list has single node

//a. make head equal to null

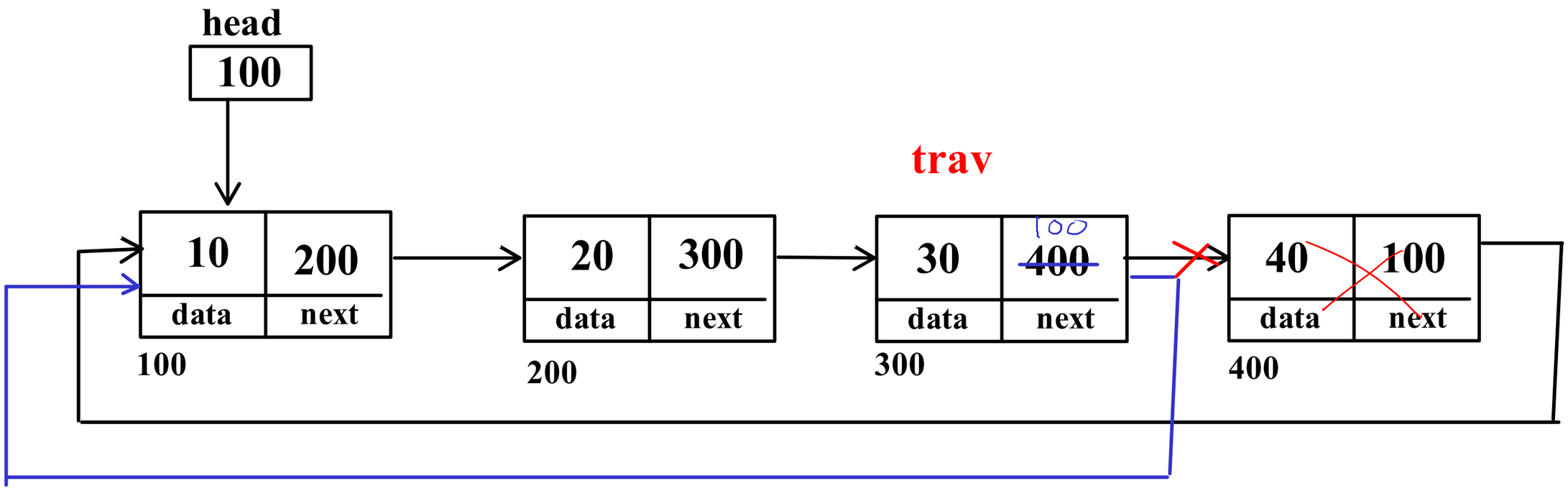
//2. if list has multiple nodes

//a. traverse till last node

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

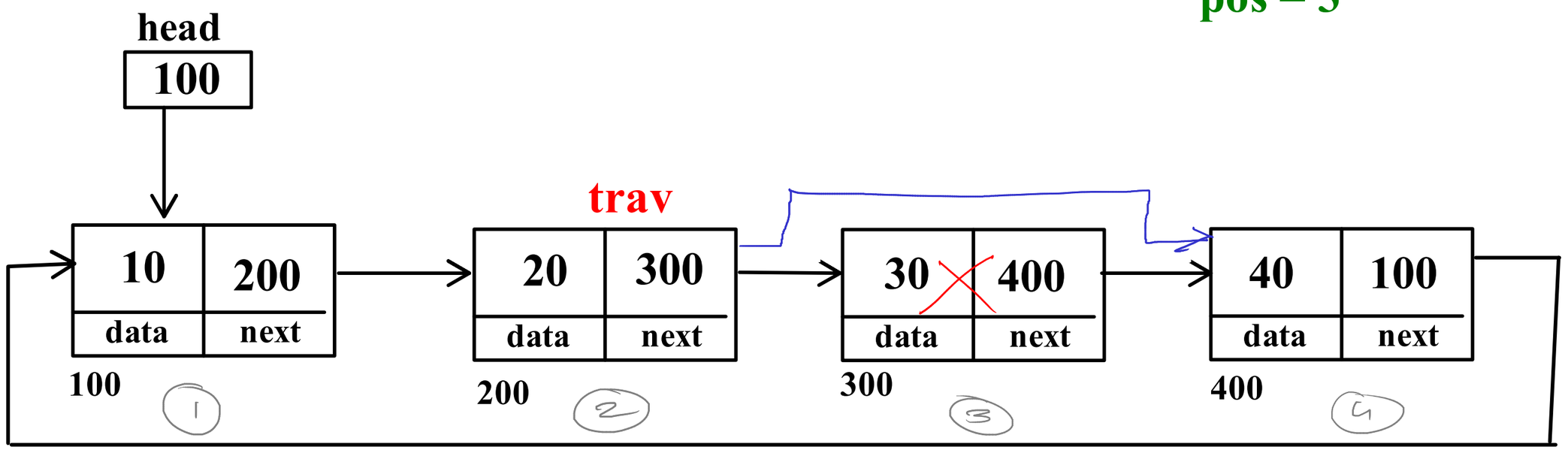
//c. move head on second node

Singly Circular Linked List - Del Last



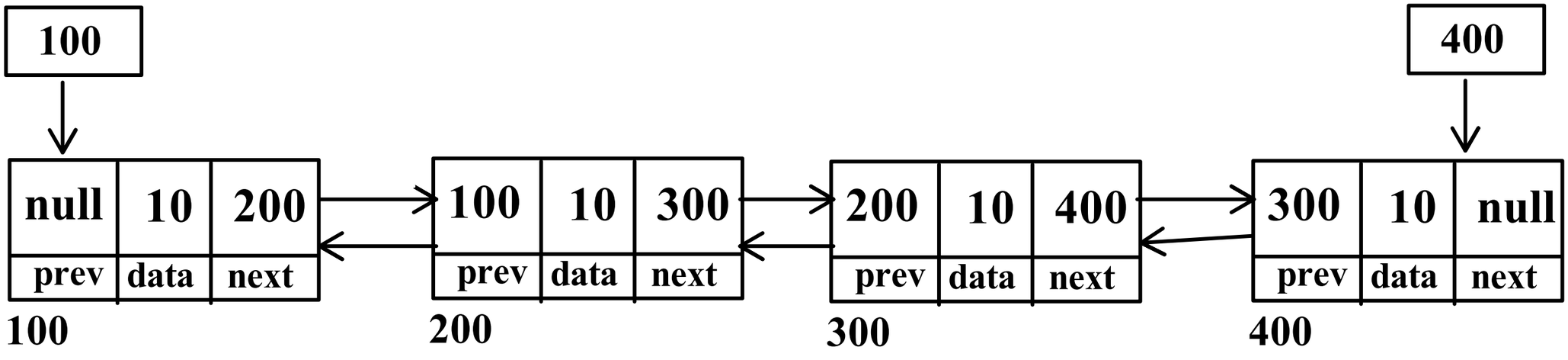

```
//1. if list has single node
        // make head equal to null
//2. if list has multiple nodes
        //a. traverse till second last node
        //b. add head into next of second last node
```

Singly Circular Linked List - Del pos




```
//1. if list has single node
        // make head equal to null
//2. if list has multiple nodes
        //a. traverse till pos-1 node
        //b. add pos+1 node into next of pos-1 node
```

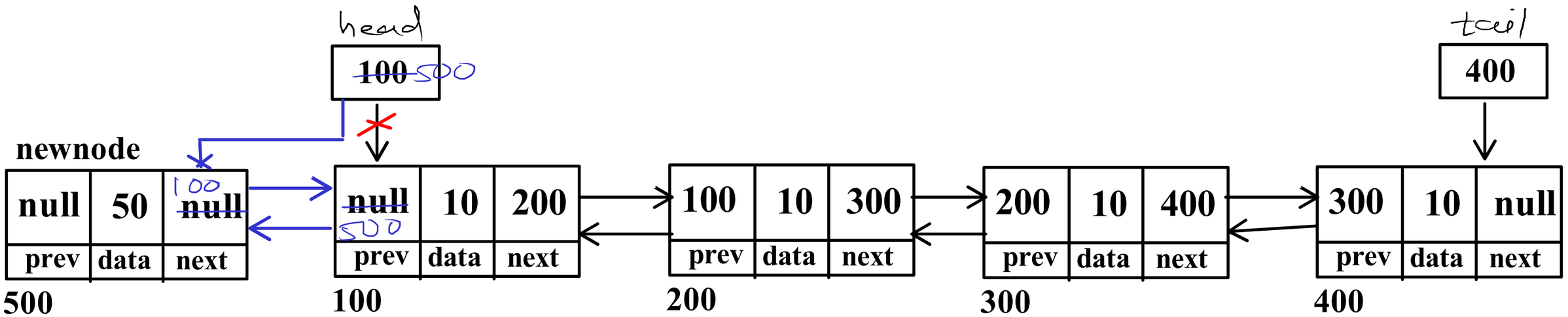
Doubly Linear Linked List - Display




```
// forward displat
//1. create trav and start at first node
//2. print data of current node
//3. go on next node
//4. repeat step 2 and 3 untill trav != null
```

```
// reverse displat
//1. create trav and start at last node
//2. print data of current node
//3. go on prev node
//4. repeat step 2 and 3 untill trav != null
```

Doubly Linear Linked List - Add first



//1. create node with data

//2. if list is empty

//a. add newnode into head and tail

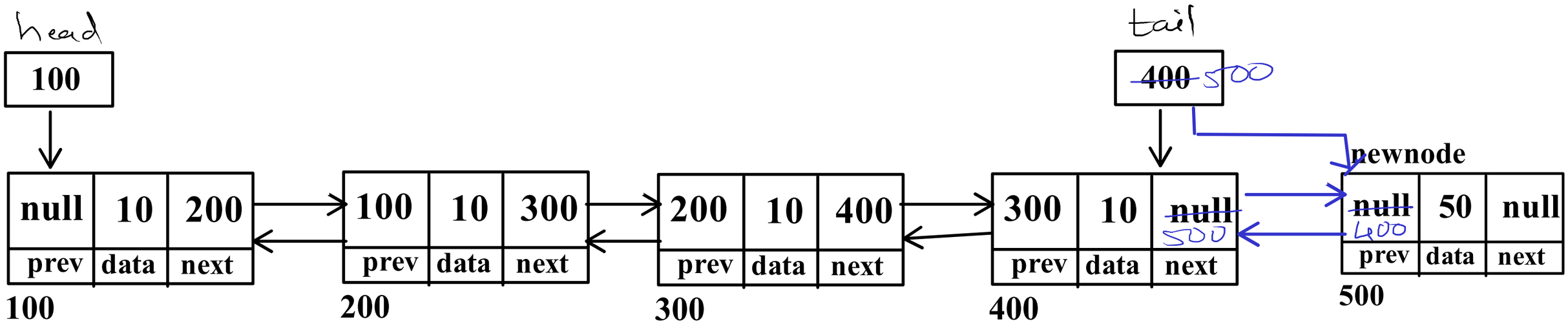
//3. if list is not empty

//a. add first node into next of newnode

//b. add newnode into prev of first node

//c. move head on newnode

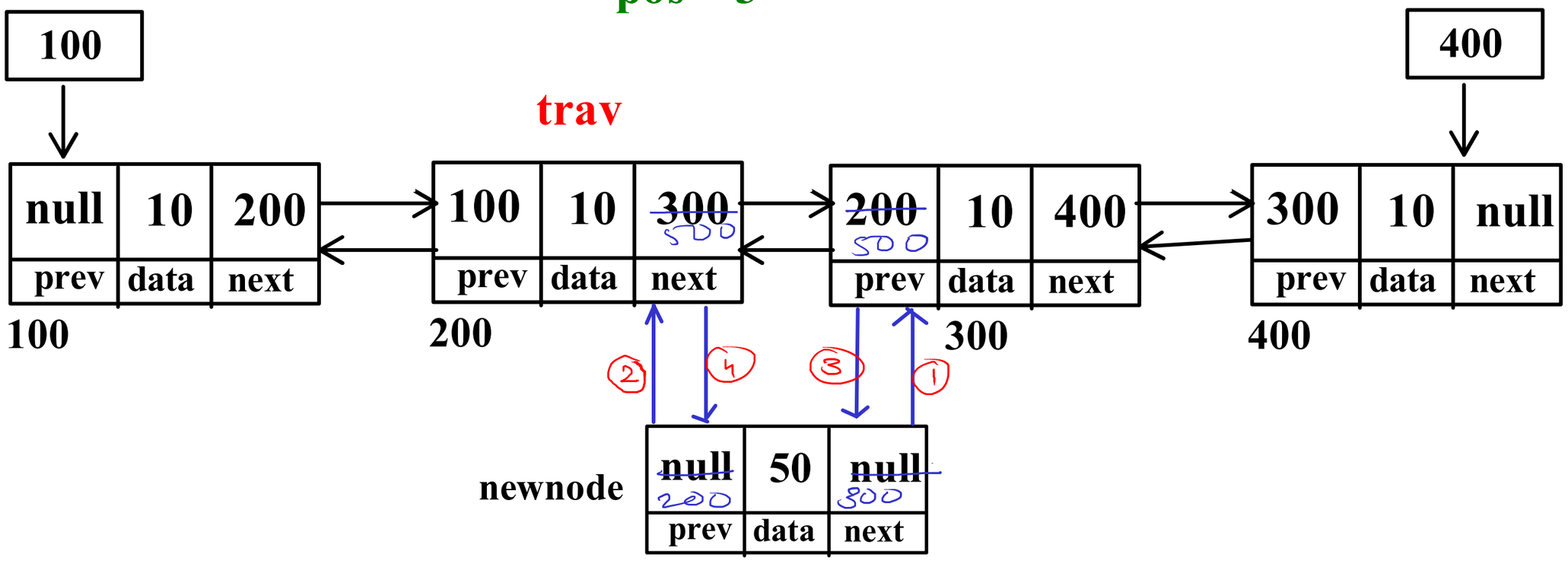
Doubly Linear Linked List - Add Last




```
//1. create node with data
//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
```

Doubly Linear Linked List - Add at Position

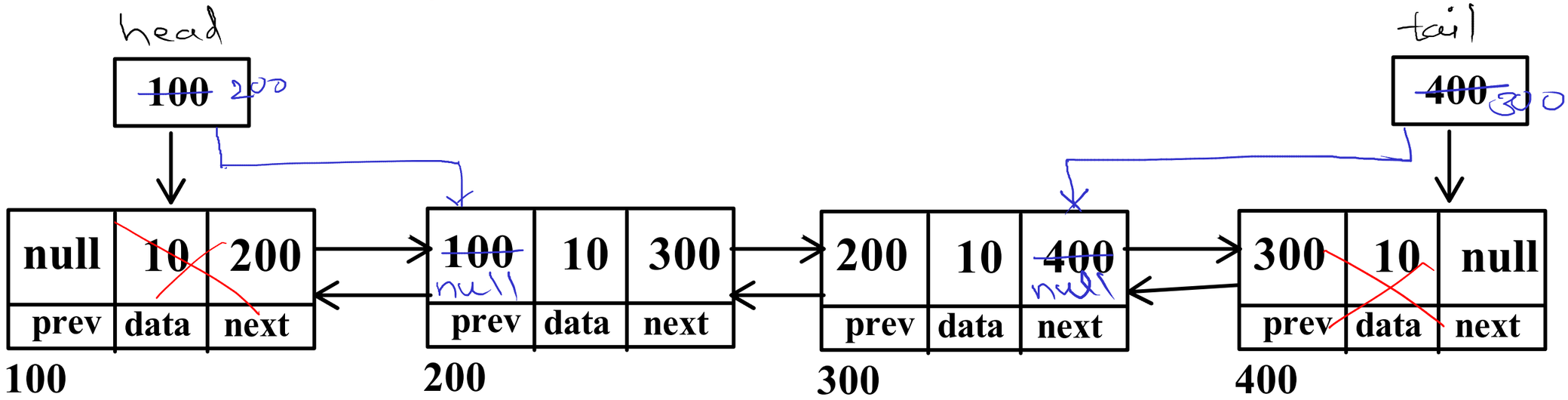
pos = 3



500

```
//1. create node with data
//2. if list is empty
        //a. add newnode into head and tail
//3. if list is not empty
        //traverse till pos-1 node
        //a. add pos node into next of newnode
        //b. add pos -1 node into prev of newnode
        //c. add newnode into prev of pos node
        //d. add newnode into next of pos-1 node
```

Doubly Linear Linked List - Delete First and Last

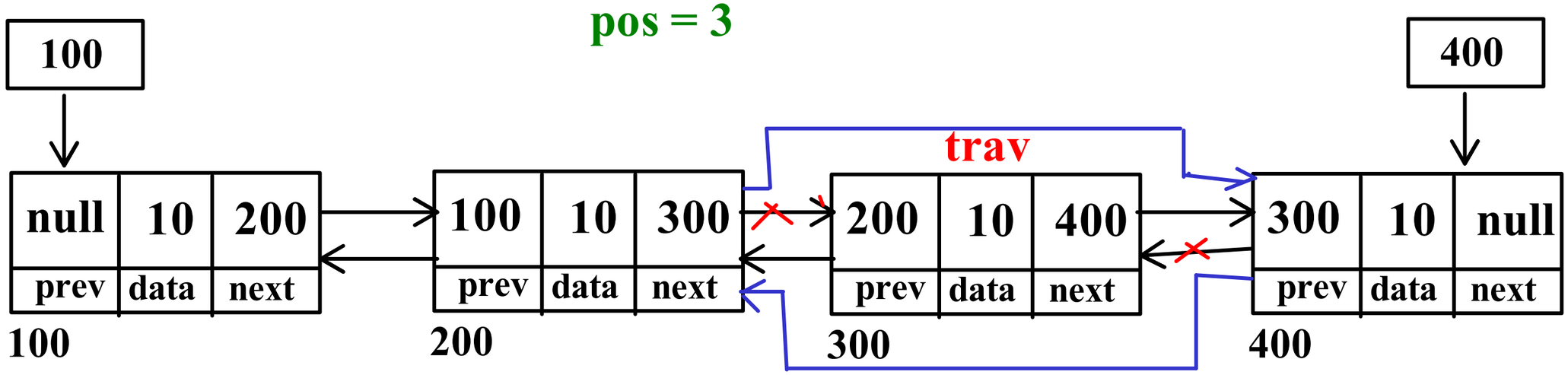


//1. if list has single
//a make head = tail = null
//2. if list has multiple nodes
//a. move head on second node
//b. add null into prev of second node

//1. if list has single
//a make head = tail = null
//2. if list has multiple nodes
//a. move tail on second last node
//b. add null into next of second last node

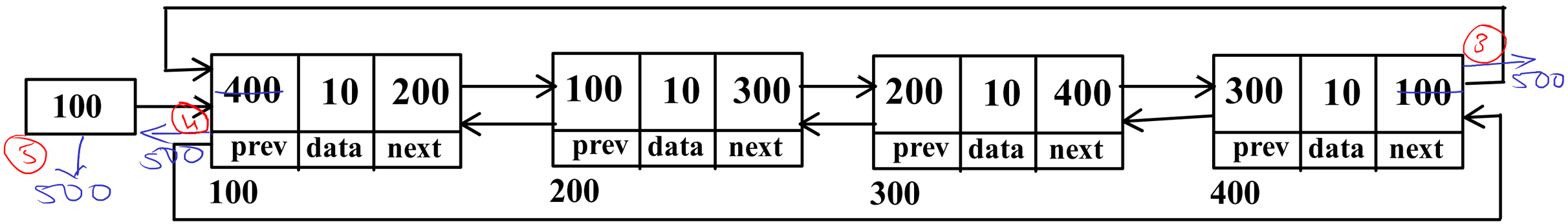
Doubly Linear Linked List - Delete Position

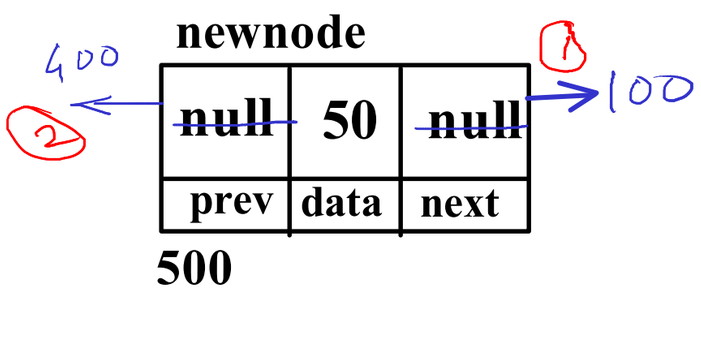
pos = 3




```
//1. if list has single node
        // make head = tail = null;
//2. if list has multiple nodes
        //a. traverse till pos node
        //b. add pos-1 node into prev of pos+1 node
        //c. add pos+1 node into next of pos-1 node
```

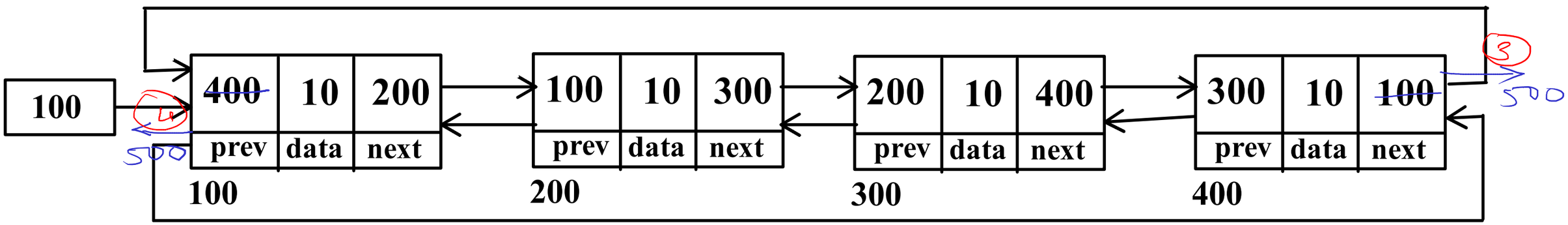
Doubly Circular Linked List - Add First

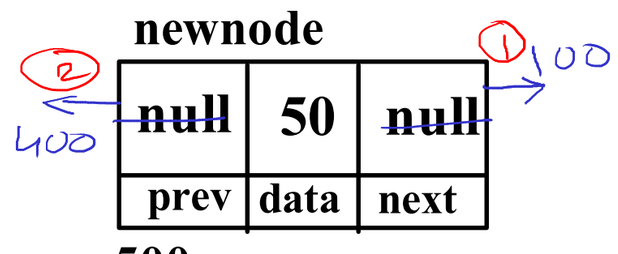





```
//1. create node
//2. if list is empty
        //a. add newnode into head
        //b. make it circular
//3. if list is not empty
        //a. add first node into next of newnode
        //b. add last node into prev of newnode
        //c. add newnode into next of last node
        //d. add newnode into prev of first node
        //e. add nenwode into head
```

Doubly Circular Linked List - Add Last

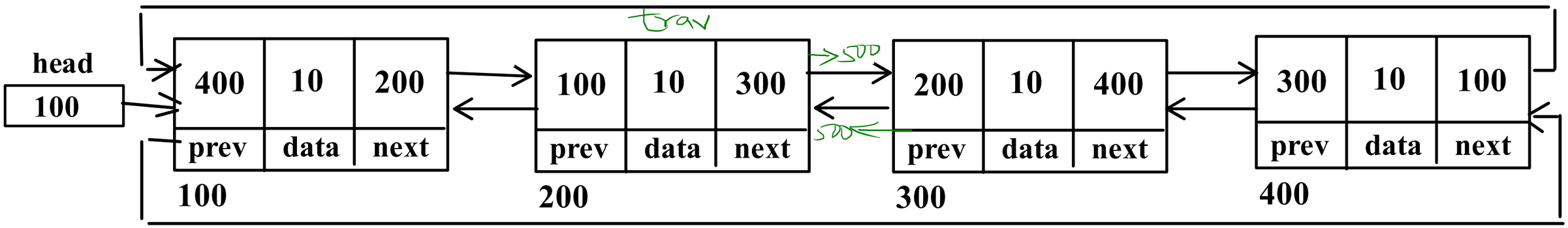


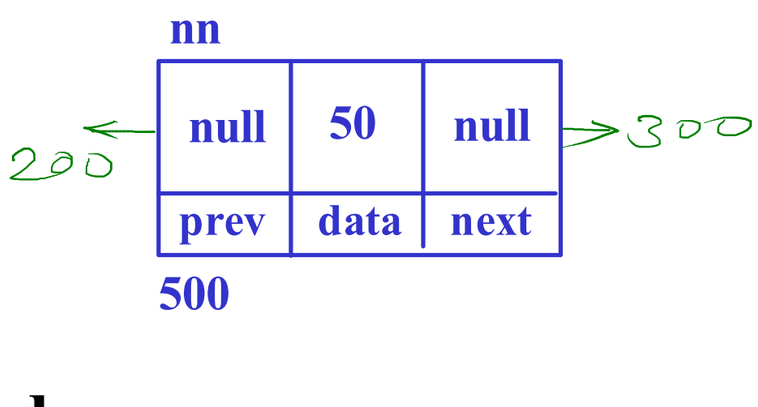



```
//1. create node
//2. if list is empty
        //a. add newnode into head
        //b. make it circular
//3. if list is not empty
        //a. add first node into next of newnode
        //b. add  last node into prev of newnode
        //c. add newnode into next of last node
        //d. add newnode into prev of first node
```

Doubly Circular Linked List - Add pos

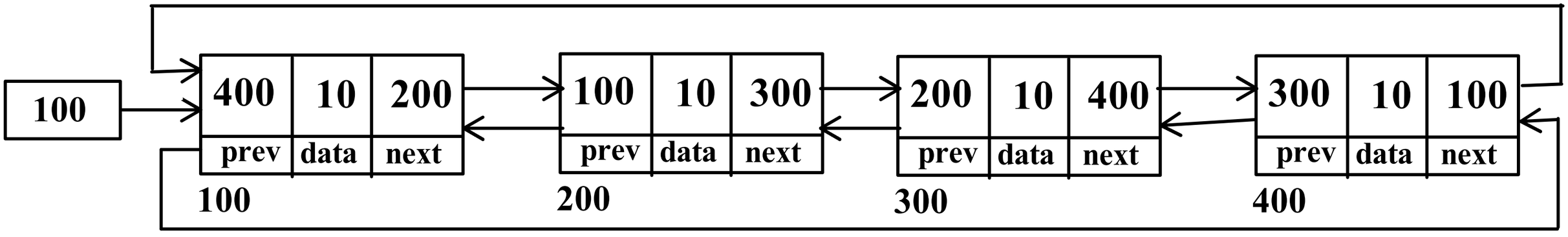
pos=3






```
//1. create node
//2. if list is empty
        // add nn into head
        // make it circular
//3. if list is not empty
        // traverse till pos -1 node
        //a. add pos node into next of nn
        //b. add pos-1 node into prev of nn
        //c. add nn into next of pos-1 node
        //d. add nn into prev of pos node
```

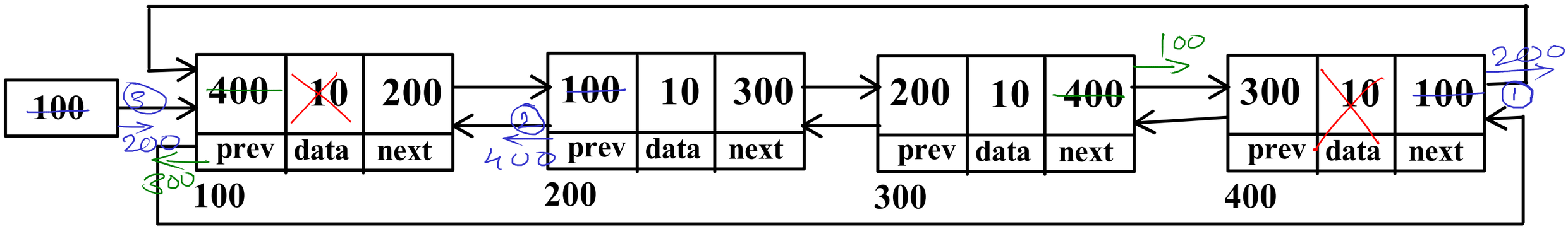
Doubly Circular Linked List - Display



// Forward display
//1. create trav and start at first node
//2. print current node
//3. go on next node
//4. repeat step 2 and 3 till last node

// Reverse display
//1. create trav and start at last node
//2. print current node
//3. go on prev node
//4. repeat step 2 and 3 till first node

Doubly Circular Linked List - Delete First and Last



Delete First

//1. if list has single node

//make head = null

//2. if list has multiple nodes

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

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

//c. move head on second node

Delete Last

//1. if list has single node

//make head = null

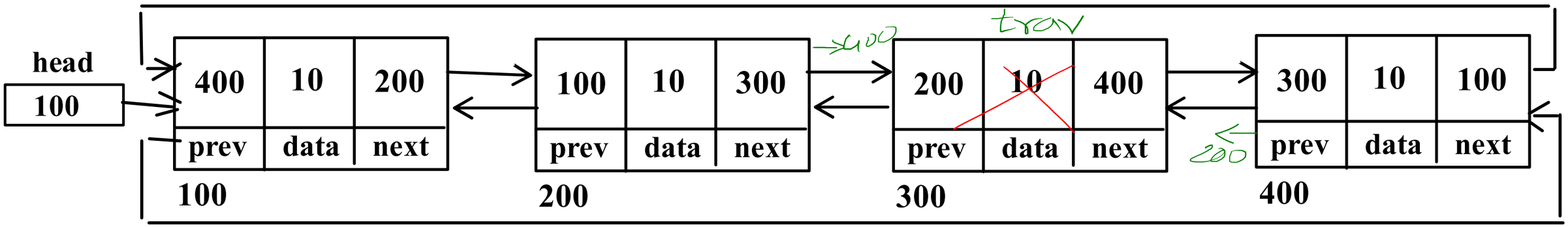
//2. if list has multiple nodes

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

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

Doubly Circular Linked List - Del Pos

pos=3




```
//1. if list is empty
        // do nothing
//2. if list has single node
        // make head  = null
//3. if list has multiple nodes
        //a. traverse till pos node
        //b. add pos + 1 (400) into next of pos - 1 node(200)
        //c. add pos - 1(200) into prev of pos + 1(400)
```