

IF232

ALGORITHMS

&

DATA STRUCTURES

05
LINKED LISTS 2

DENNIS GUNAWAN

REVIEW

Linked Lists:

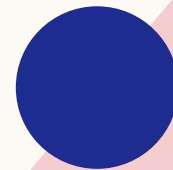
Dynamic Memory Allocation

Single Linked Lists

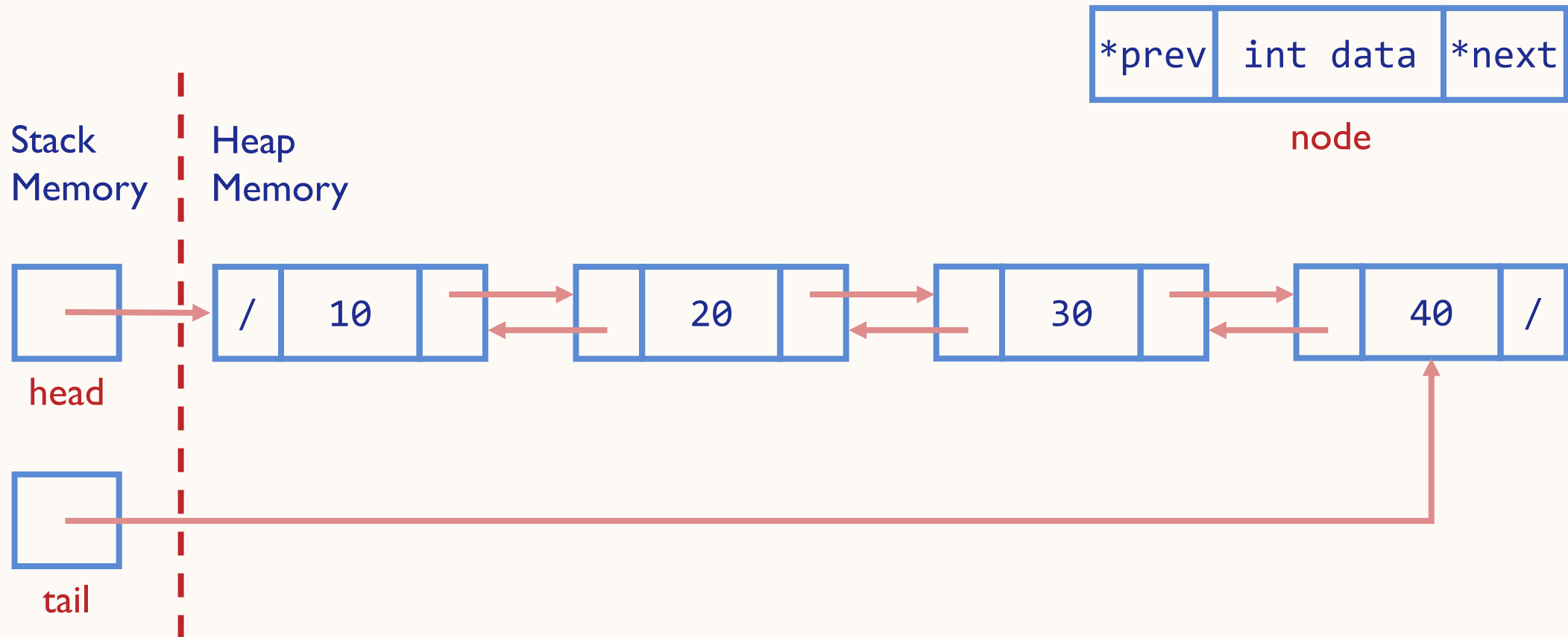
OUTLINE

Double Linked Lists

Circular Linked Lists



DOUBLE LINKED LISTS



DECLARATION

```
struct tnode{  
    int data;  
    struct tnode *prev, *next;  
};
```

```
int main()  
{  
    struct tnode *head, *tail, *node;  
    int number;  
    ...  
    head = tail = NULL;  
    ...  
}
```



struct tnode



node



head

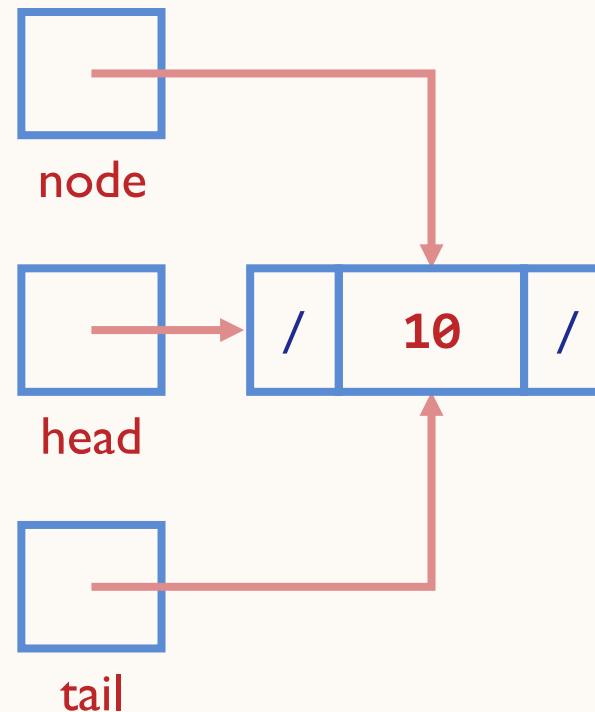


tail

INSERTION

AT THE BEGINNING OF A DOUBLY LINKED LIST

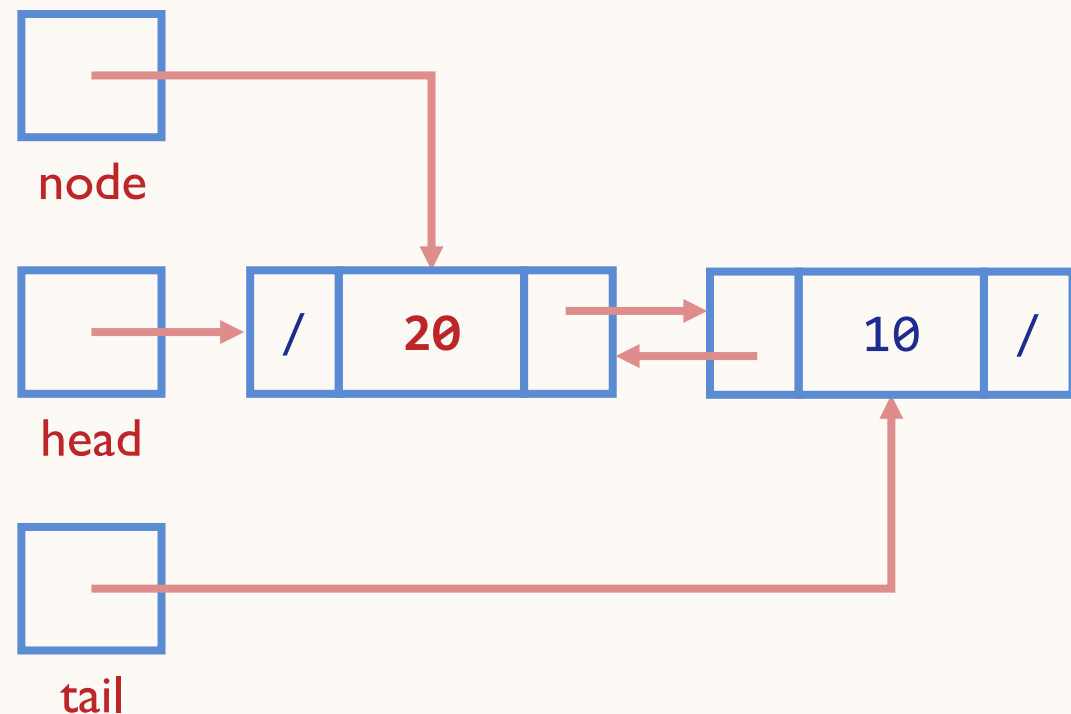
```
scanf("%d", &number); //10
node = (struct tnode *) malloc
        (sizeof(struct tnode));
node->data = number;
if(head == NULL){
    tail = node;
    node->next = NULL;
}
else{
    head->prev = node;
    node->next = head;
}
head = node;
head->prev = NULL;
```



INSERTION

AT THE BEGINNING OF A DOUBLY LINKED LIST

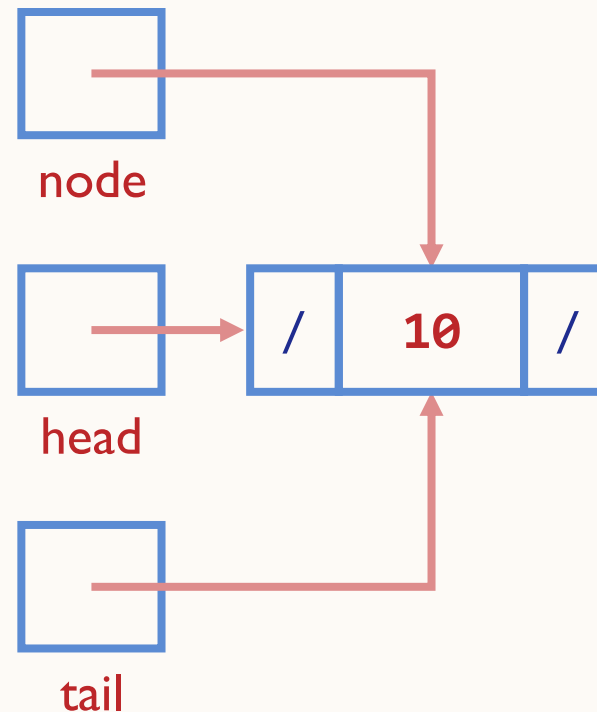
```
scanf("%d", &number); //20
node = (struct tnode *) malloc
        (sizeof(struct tnode));
node->data = number;
if(head == NULL){
    tail = node;
    node->next = NULL;
}
else{
    head->prev = node;
    node->next = head;
}
head = node;
head->prev = NULL;
```



INSERTION

AT THE END OF A DOUBLY LINKED LIST

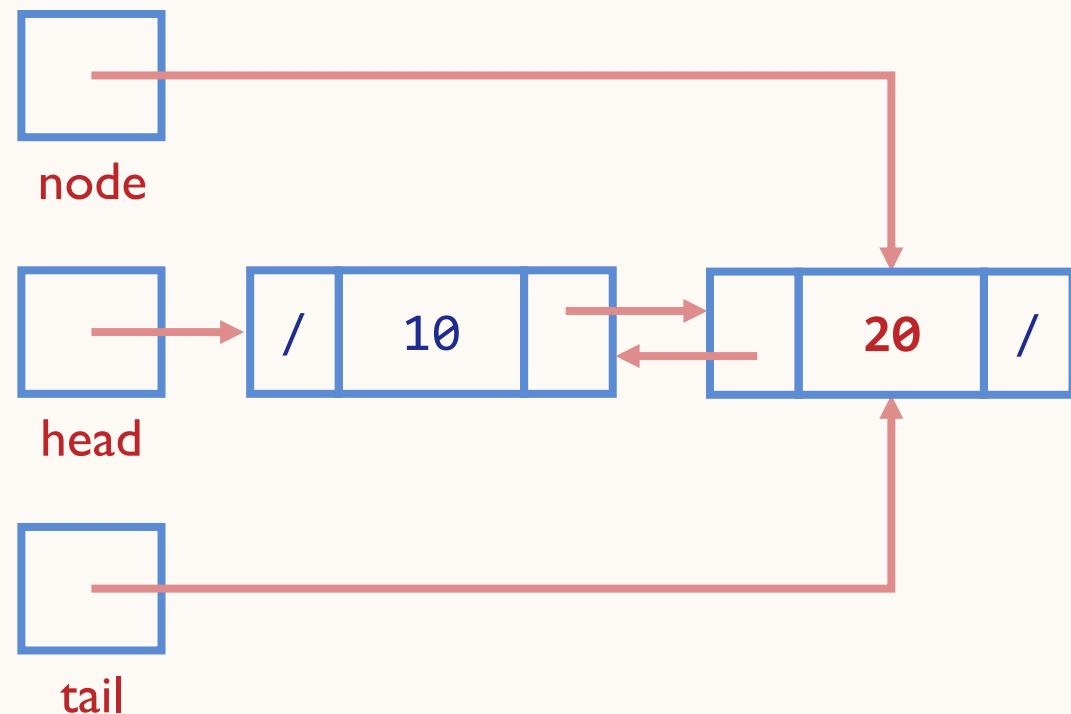
```
scanf("%d", &number); //10
node = (struct tnode *) malloc
        (sizeof(struct tnode));
node->data = number;
if(head == NULL){
    head = node;
    node->prev = NULL;
}
else{
    tail->next = node;
    node->prev = tail;
}
tail = node;
tail->next = NULL;
```



INSERTION

AT THE END OF A DOUBLY LINKED LIST

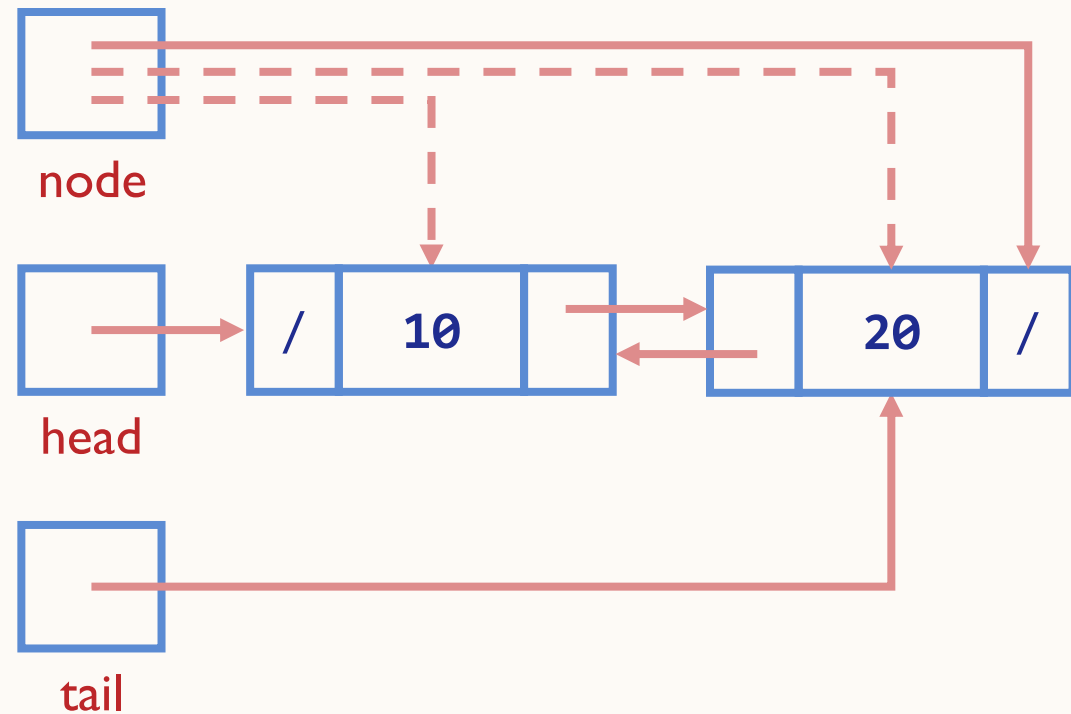
```
scanf("%d", &number); //20
node = (struct tnode *) malloc
        (sizeof(struct tnode));
node->data = number;
if(head == NULL){
    head = node;
    node->prev = NULL;
}
else{
    tail->next = node;
    node->prev = tail;
}
tail = node;
tail->next = NULL;
```



TRAVERSAL / DISPLAY

HEAD TO TAIL

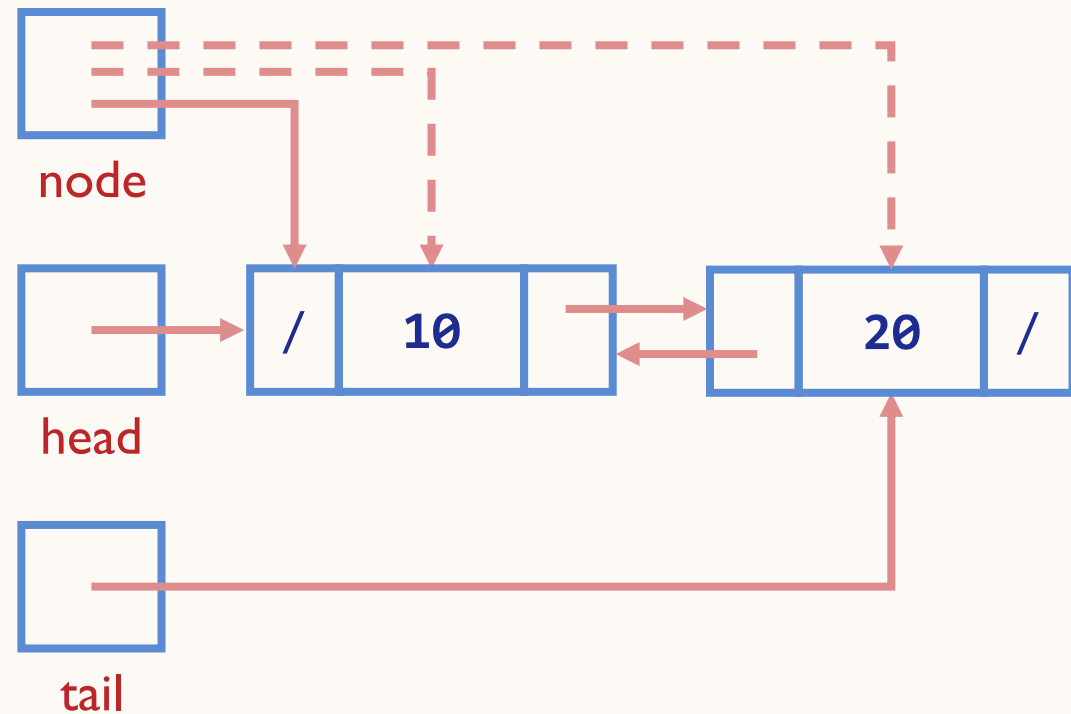
```
node = head;
while(node != NULL){
    printf("%d ", node->data);
    node = node->next;
}
```



TRAVERSAL / DISPLAY

TAIL TO HEAD

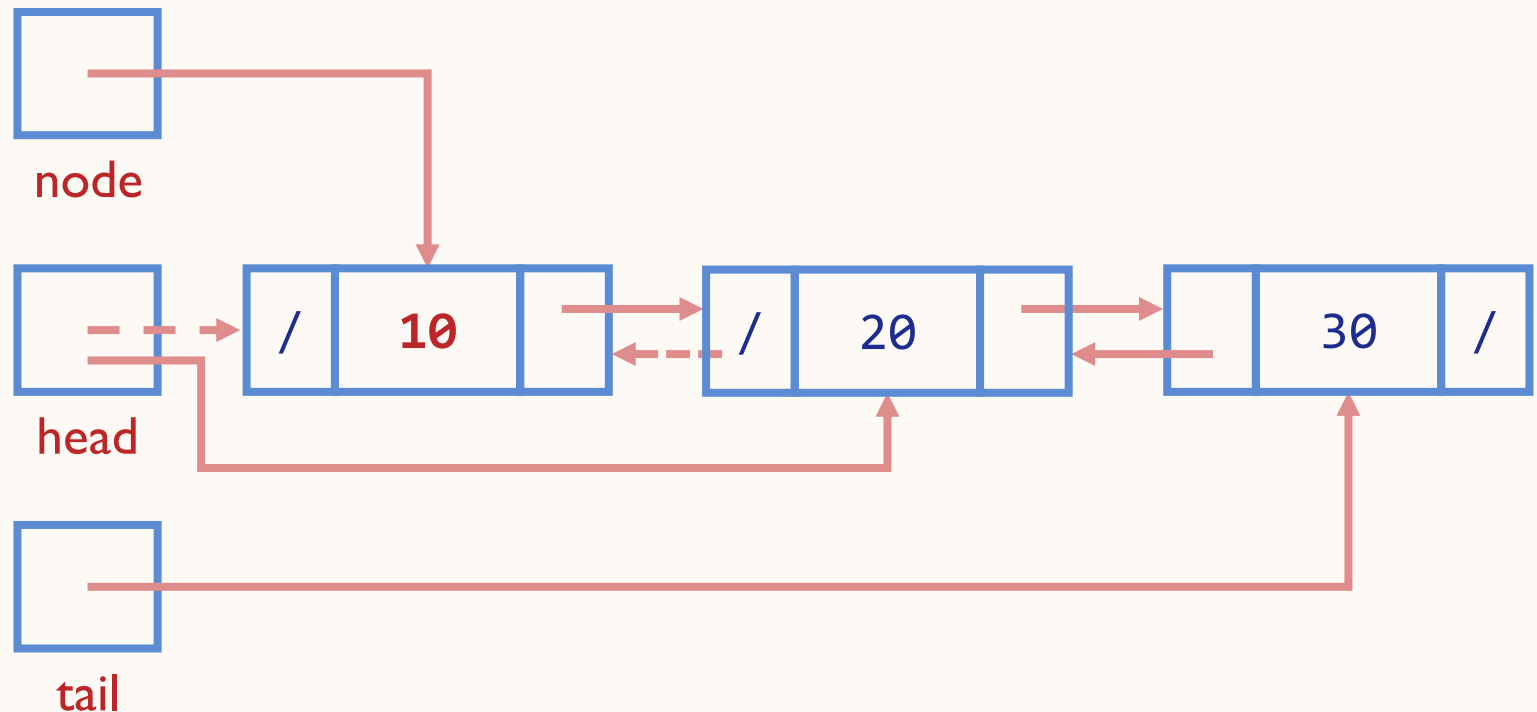
```
node = tail;
while(node != NULL){
    printf("%d ", node->data);
    node = node->prev;
}
```



DELETION

DELETING THE FIRST NODE FROM A DOUBLY LINKED LIST

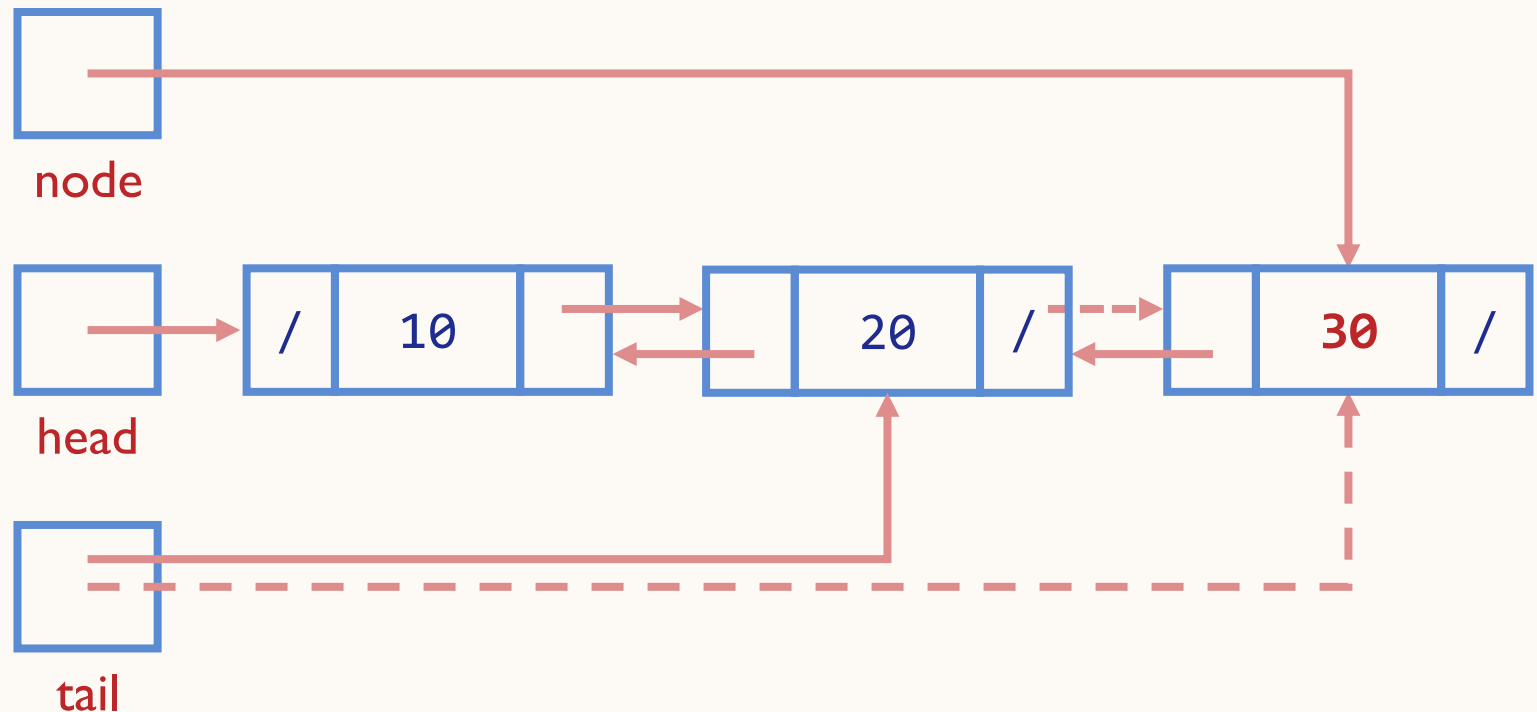
```
node = head;  
head = head->next;  
head->prev = NULL;  
free(node);
```



DELETION

DELETING THE LAST NODE FROM A DOUBLY LINKED LIST

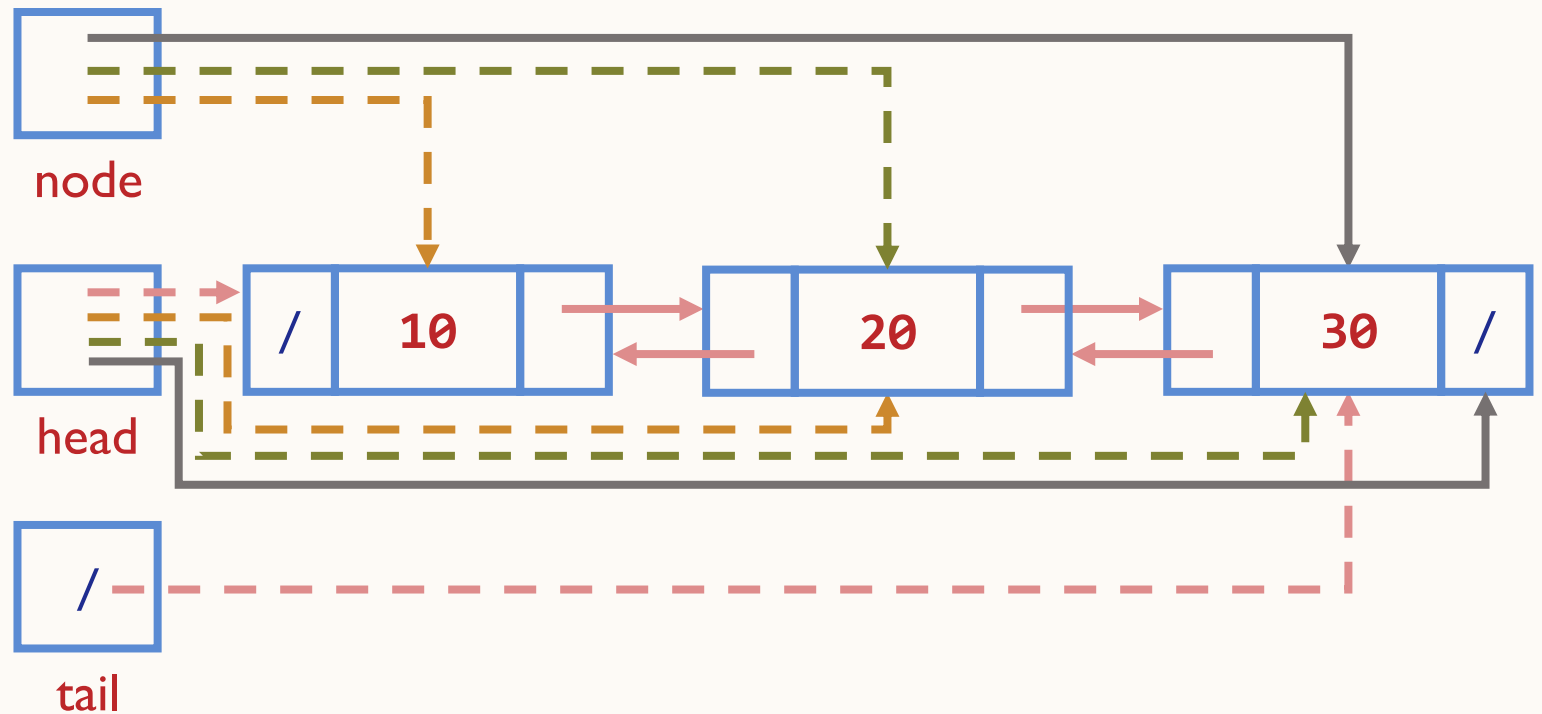
```
node = tail;  
tail = tail->prev;  
tail->next = NULL;  
free(node);
```



DELETION

DELETING THE ENTIRE LIST (HEAD TO TAIL)

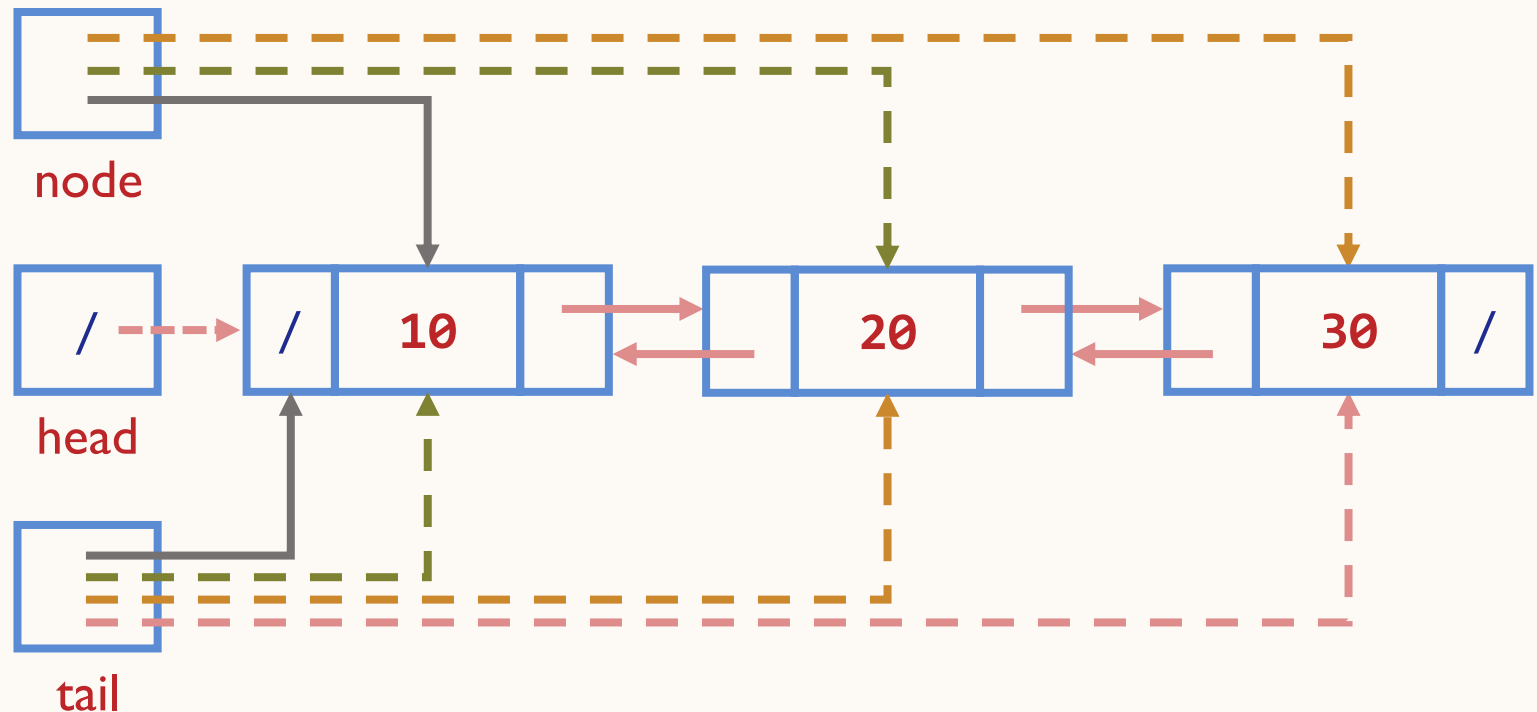
```
while(head != NULL){  
    node = head;  
    head = head->next;  
    free(node);  
}  
tail = NULL;
```



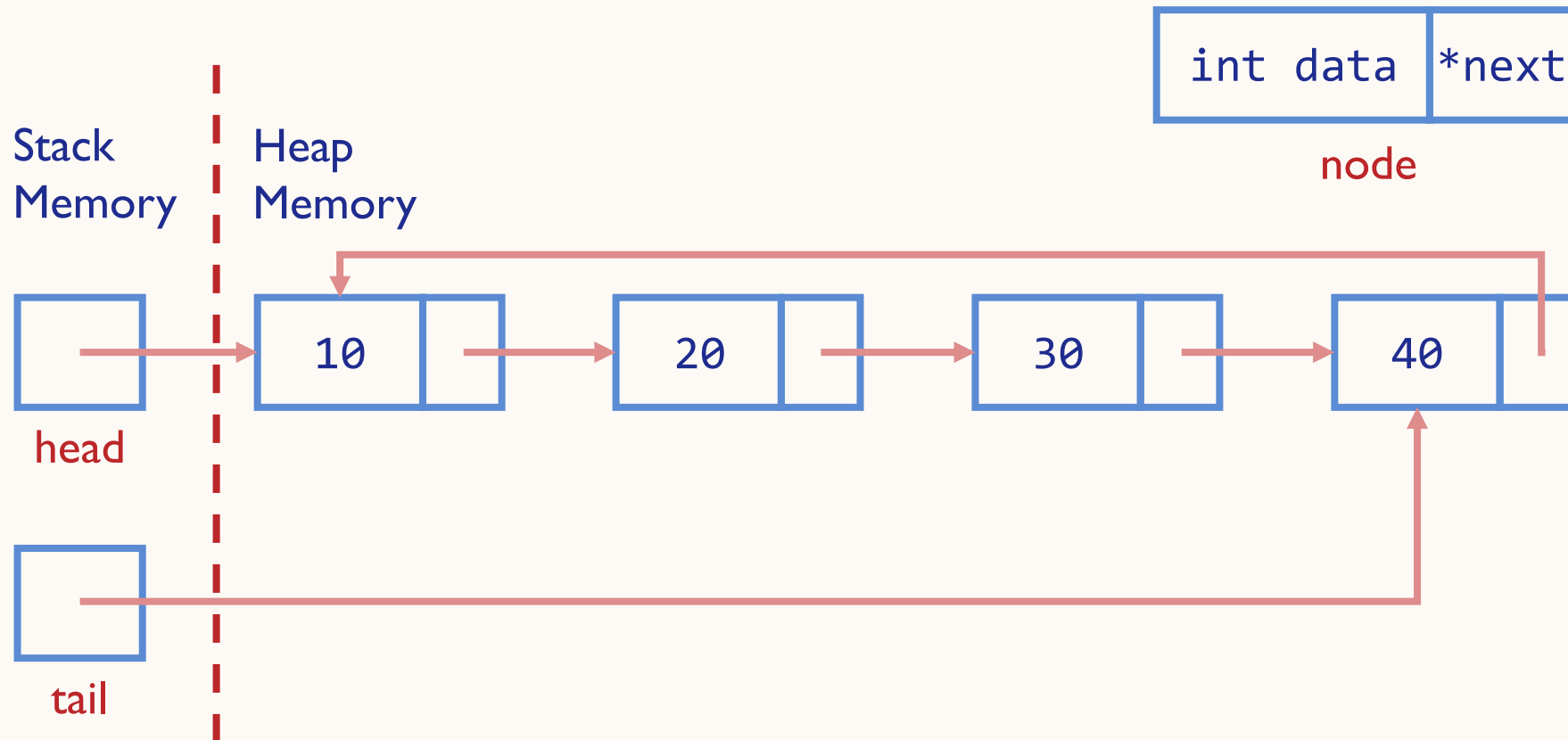
DELETION

DELETING THE ENTIRE LIST (TAIL TO HEAD)

```
while(tail != NULL){  
    node = tail;  
    tail = tail->prev;  
    free(node);  
}  
head = NULL;
```



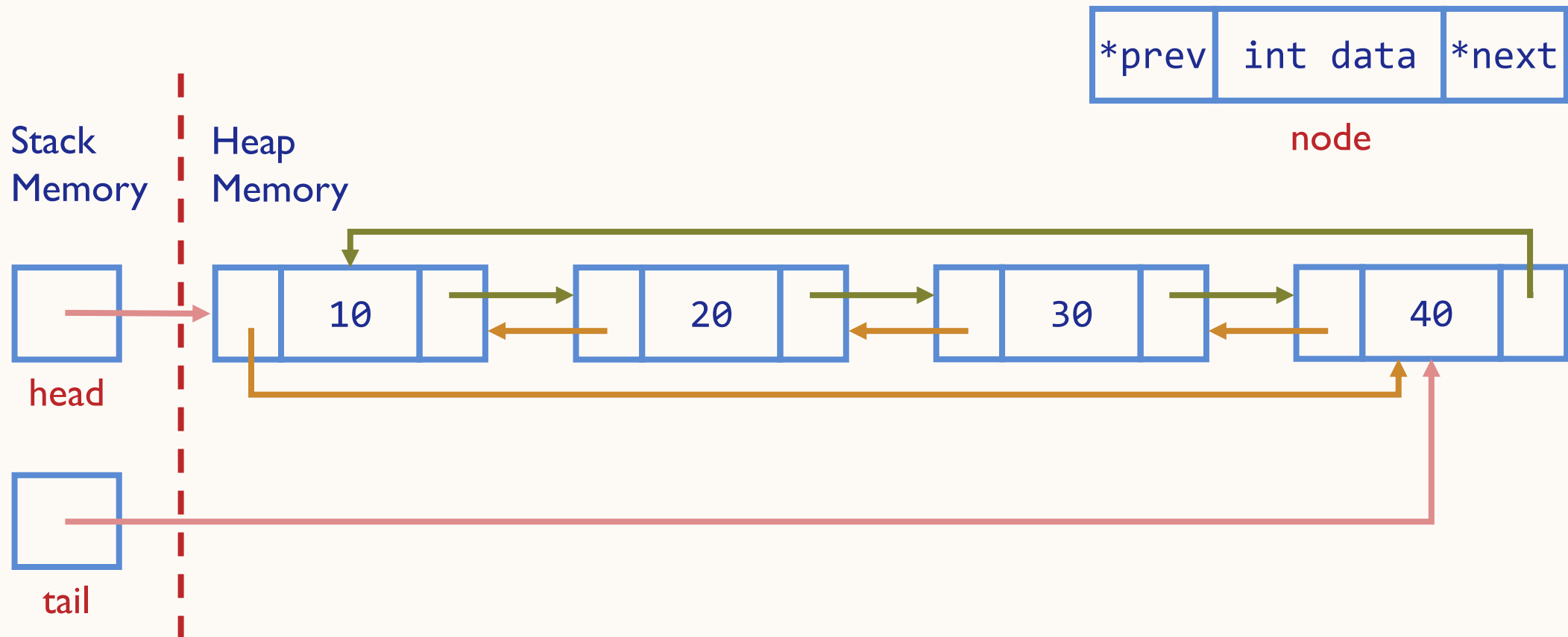
CIRCULAR SINGLY LINKED LISTS



CIRCULAR SINGLY LINKED LISTS

- How to insert a node at the beginning of a circular singly linked list?
- How to insert a node at the end of a circular singly linked list?
- How to delete the first node from a circular singly linked list?
- How to delete the last node from a circular singly linked list?
- How to delete the entire list?

CIRCULAR DOUBLY LINKED LISTS



CIRCULAR DOUBLY LINKED LISTS

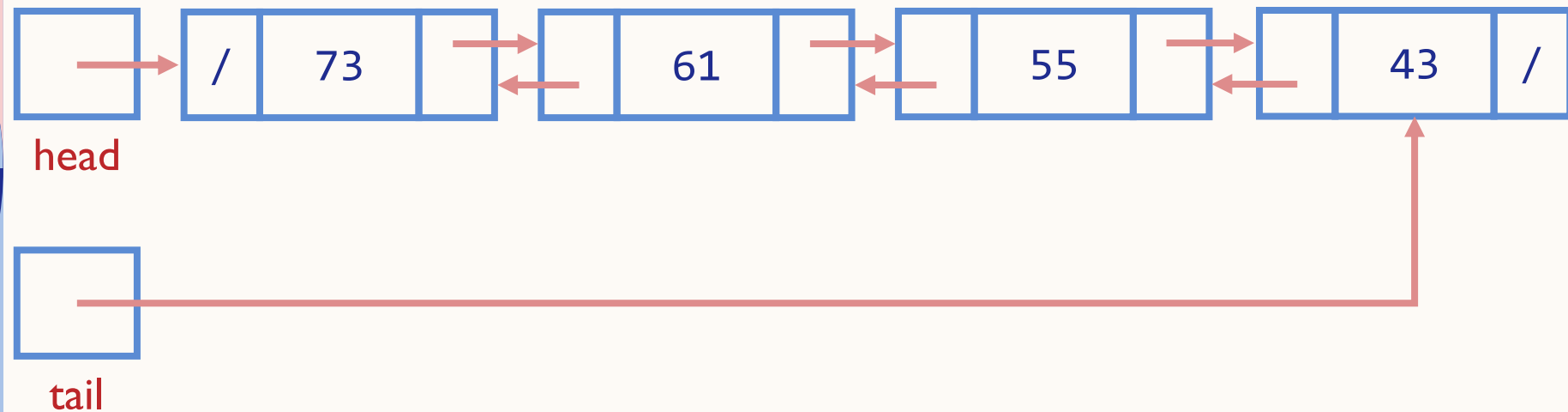
- How to insert a node at the beginning of a circular doubly linked list?
- How to insert a node at the end of a circular doubly linked list?
- How to delete the first node from a circular doubly linked list?
- How to delete the last node from a circular doubly linked list?
- How to delete the entire list?



PRACTICE

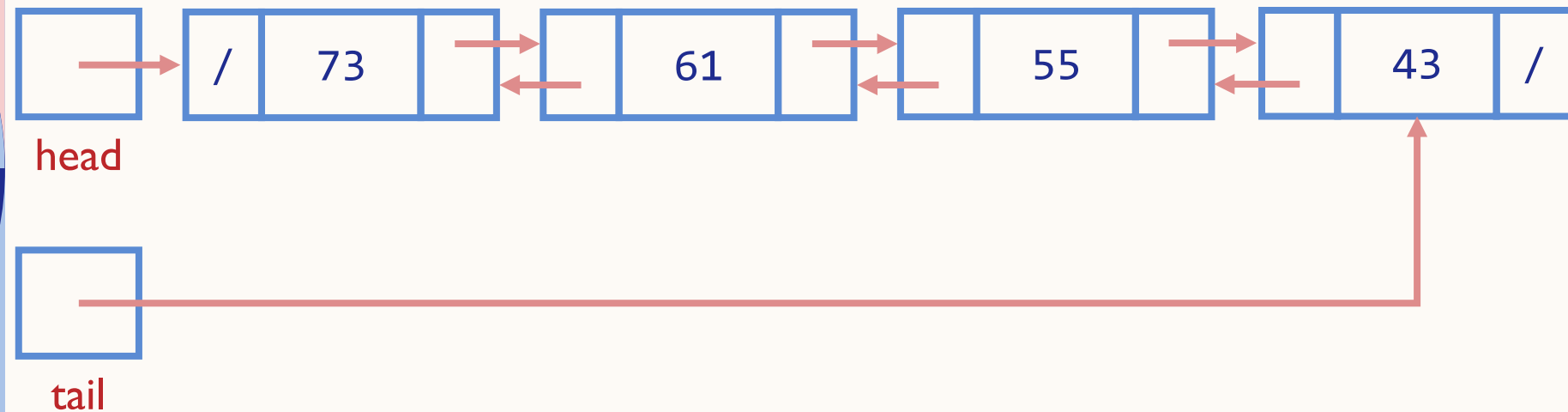
EXERCISES

1. Write a program that adds 10 to the values stored in the nodes of a doubly linked list.



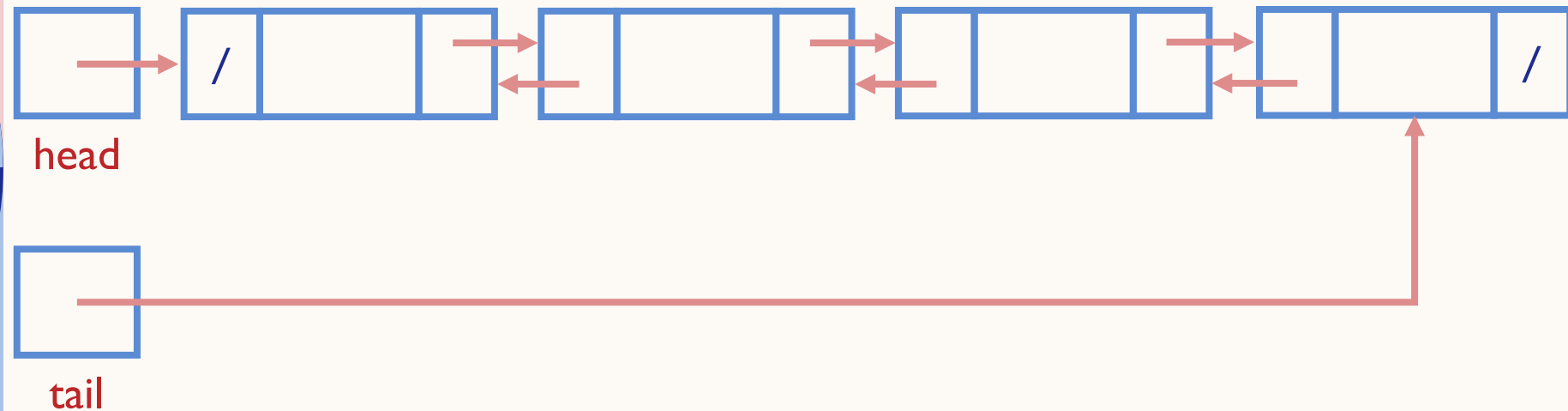
EXERCISES

2. Write a program to interchange the value of the first element with the last element, second element with second last element, so on and so forth of a doubly linked list



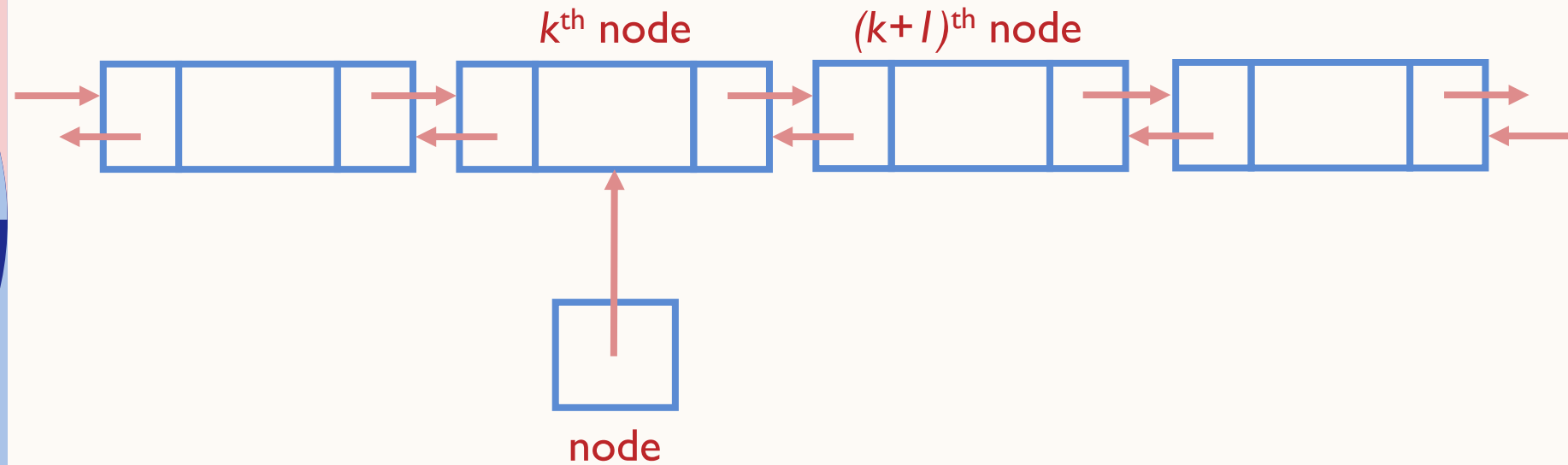
EXERCISES

3. Write a program to delete the first element of a doubly linked list. Add this node as the last node of the list.
4. Write a program to move the third node of a doubly linked list to the top of the list.



EXERCISES

5. Write a program to interchange the k^{th} and the $(k+1)^{\text{th}}$ node of a doubly linked list.



REFERENCES

- Deitel, P. and Harvey Deitel (2022), C How to Program (9th Edition), Pearson Education.
- Thareja, R. (2014), Data Structures Using C (2nd Edition), India: Oxford University Press.

NEXT

Stacks:

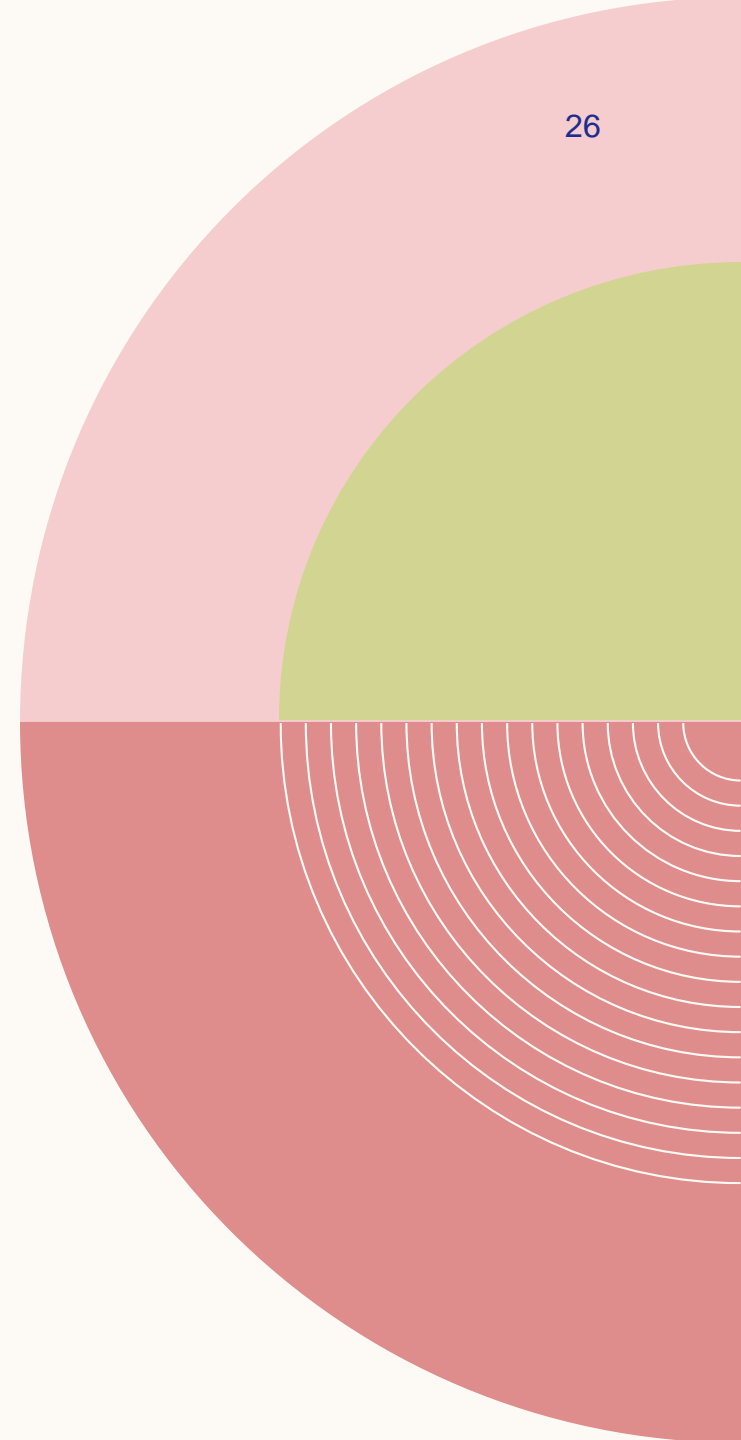
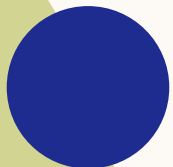
Array Representation of Stacks

Operations on a Stack

Linked Representation of Stacks

Operations on a Linked Stack

Applications of Stacks



VISION

To become an **outstanding** undergraduate Computer Science program that produces **international-minded** graduates who are **competent** in software engineering and have **entrepreneurial spirit** and **noble character**.

MISSION

1. To conduct studies with the best technology and curriculum, supported by professional lecturer
2. To conduct research in Informatics to promote science and technology
3. To deliver science-and-technology-based society services to implement science and technology

Without hard work,
nothing grows but weeds.



if INFORMATIKA
UMN

Have patience.

All things are difficult before they become easy.