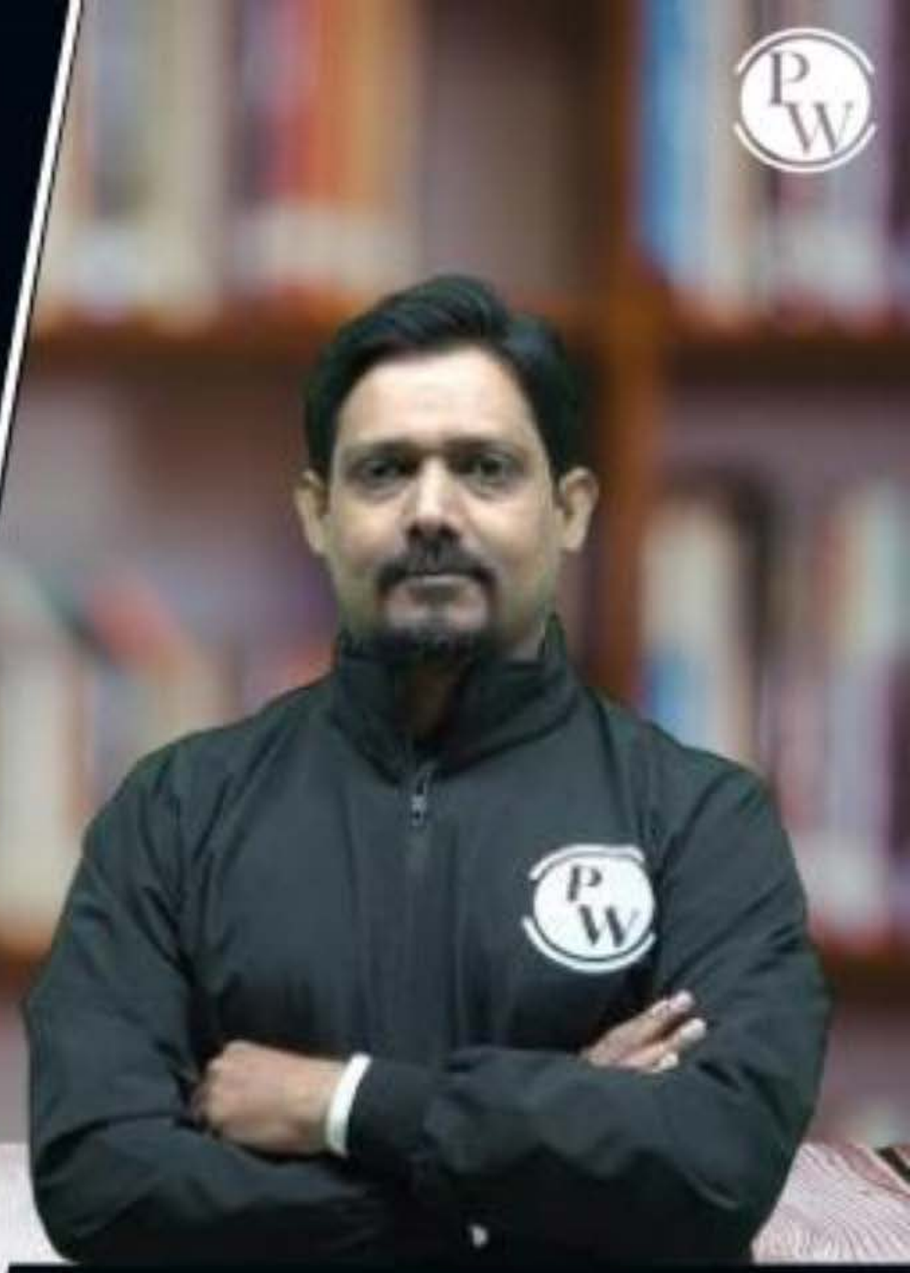
Computer Science & IT

Data Structure & Programming

Linked List

Lecture No. 03



By- Abhishek Sir

Recap of Previous Lecture

Topic	Insert begin, end
Topic	Last deletion
Topic	Recursion with Linked List
Topic	2 pointer Increement
Topic	NULL pointer dereference

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

practice proble

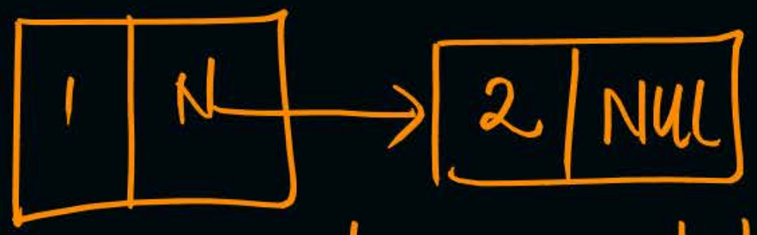
Insert x befor y

Slide

```
Node* getnode(int x) {
    Node *temp = malloc(sizeof(Node));
    temp->data = x
    temp->next = NULL
    return temp
    }
```

$temp_1$ = getnode(1)

$temp_2$ = getnode(2)



$temp_1 \rightarrow next = temp_2$

Q. Let Q denote a queue containing 3 numbers and S be an empty stack. Head (Q) returns the element at the head of the queue Q without removing it from Q. Similarly Top (S) returns the element at the top of S without removing it from S. Consider the algorithm given below.

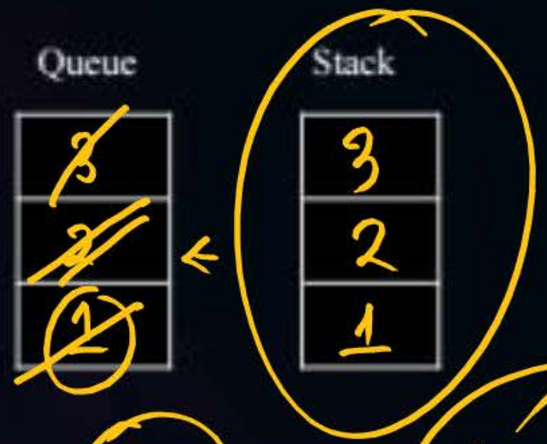
```
while Q is not Empty do
        if S is Empty OR Top (S) ≤ Head (Q) then
                X := Dequeue (Q) ;
                Push (S, x) ;
        else
                X : = Pop (S) ;
                Enqueue (Q, X) ;
        end
end
```

The maximum possible number of iterations of the while loop in the algorithm is _____ 9

(GATE_2016_2 M)

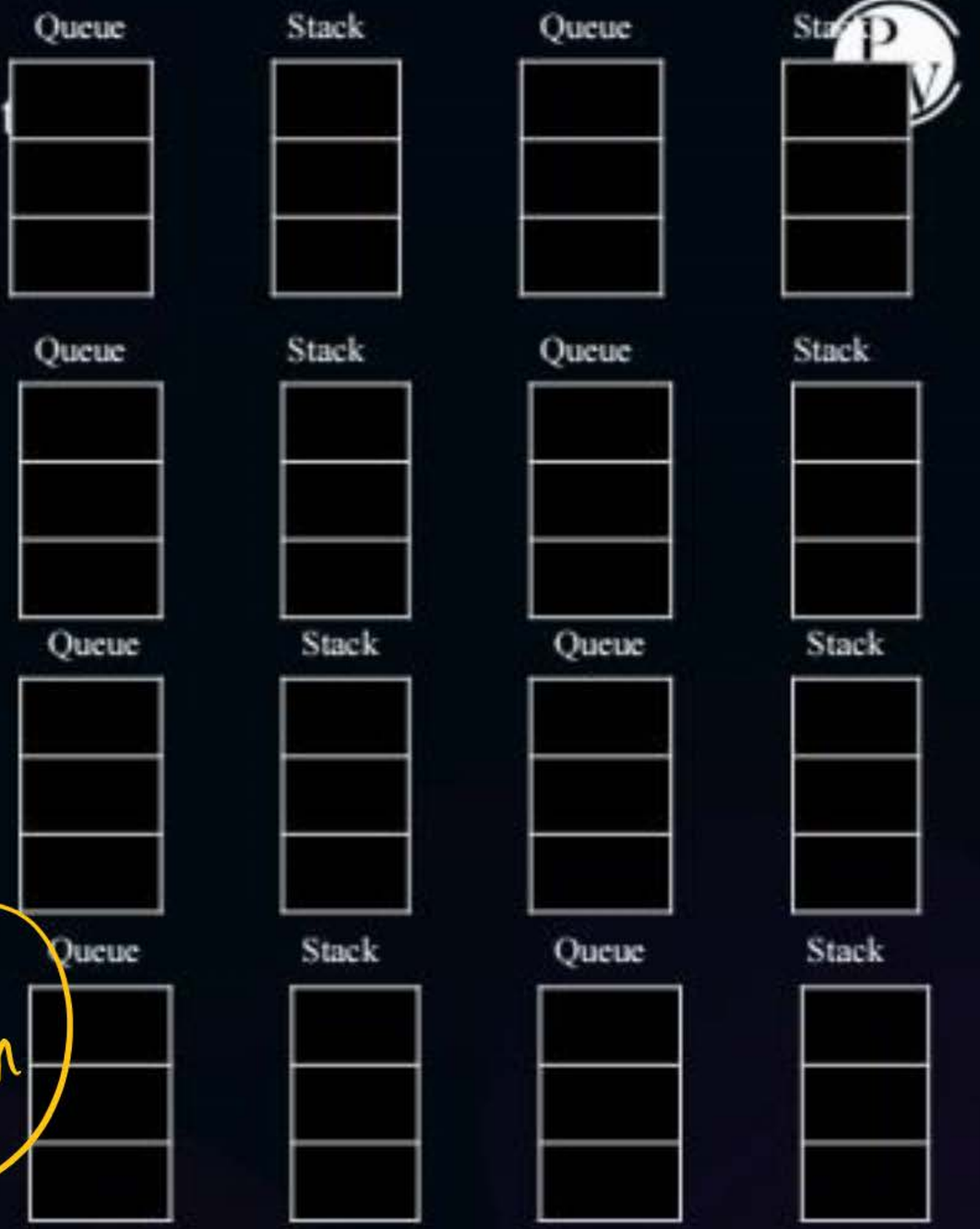
```
while Q is not Empty do
    if S is Empty OR Top (S) ≤ Head (Q)
        X := Dequeue (Q) ;
        Push (S, x) ;
    else
        X : = Pop (S) ;
        Enqueue (Q, X) ;
    end
end
```

(GATE_2016_2 M)



3

Minimum

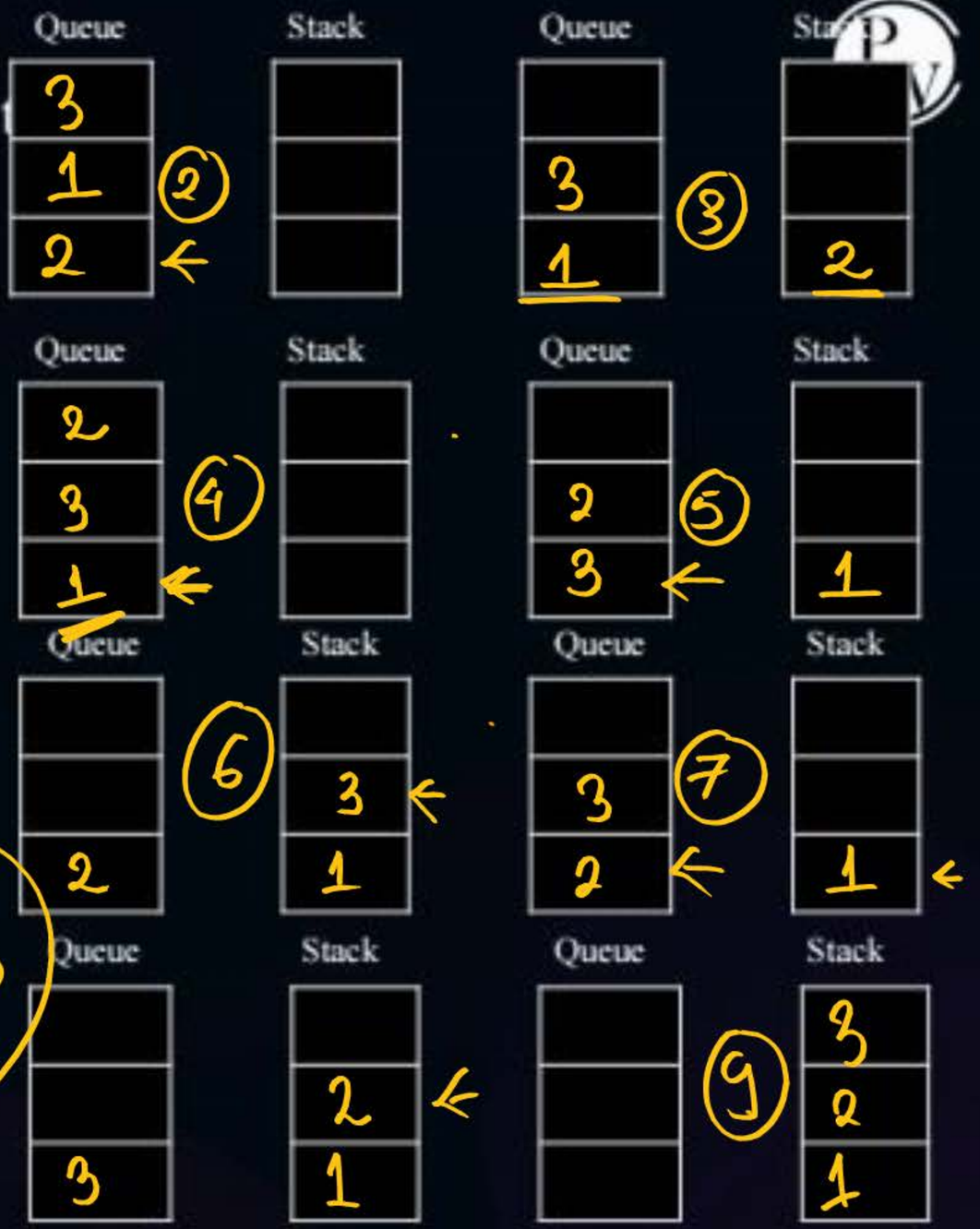


```
while Q is not Empty do
    if S is Empty OR Top (S) ≤ Head (Q) {
        X := Dequeue (Q) ;
        Push (S, x) ;
    }
    else
        X : = Pop (S) ;
        Enqueue (Q, X) ;
    end
end
```

HW

Anaysis

(GATE_2016_2 M)



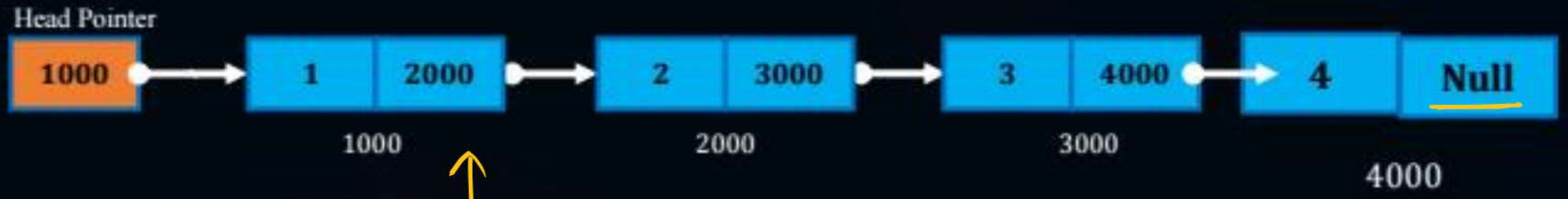
n^2

Head

$3+3 \neq 9$

$16^2 : 256$

Topic : Single Linked List



Node* p;

p = NULL; (Not Error)

p → data; (NULL pointer dereference)

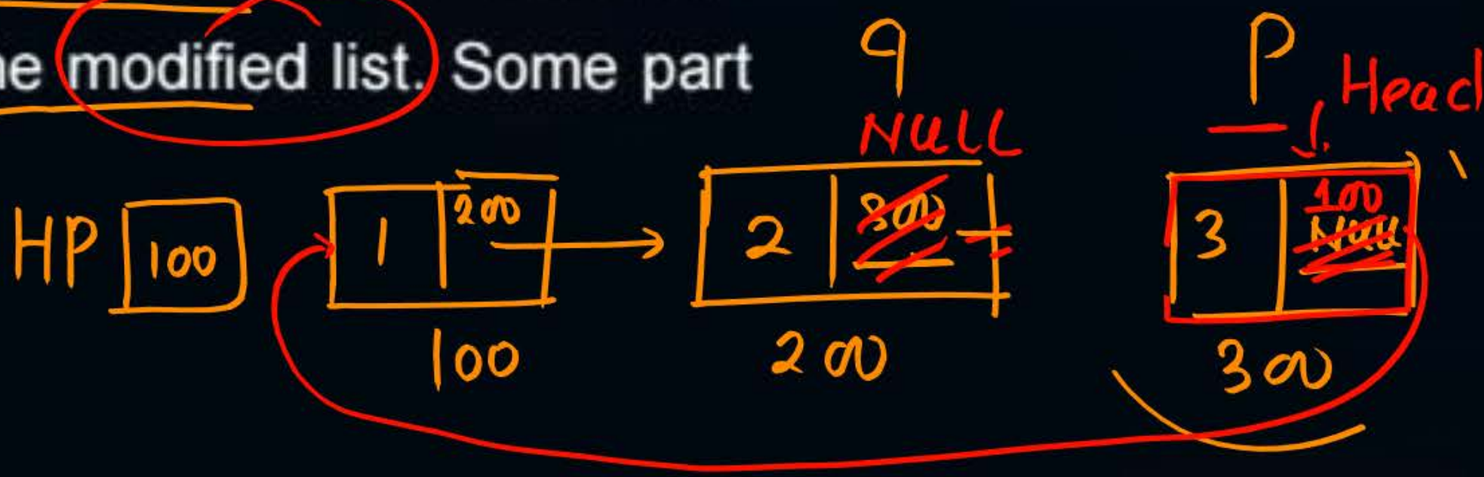
Two pointer increment

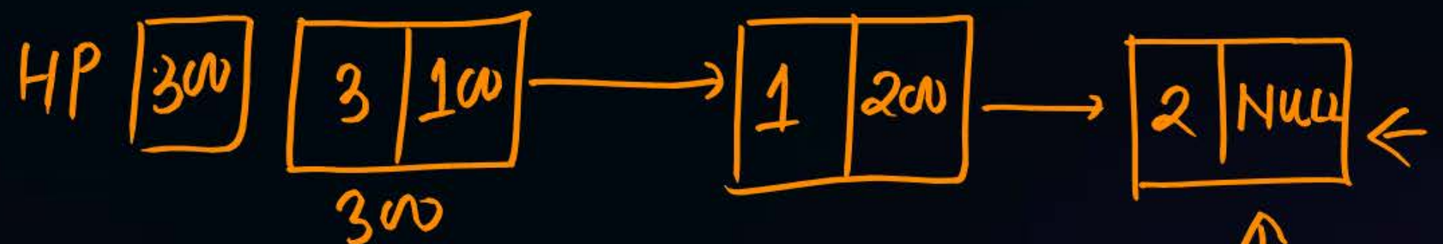
```
while(q->next != NULL){
    r = q;
    q = q->next
}
```

Slide

The following C function takes a ~~simply~~ Single-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.

```
typedef struct node {
        int value;
        struct node *next;
}Node;
```






```
q->next : NULL;
p->next : HP
Head : p;
```

```
e *move to front (Node *head) {
  Node *p, *q;
  if ((head == NULL) || (head -> next == NULL))
      return head;
  q= NULL; p=head;
  while (p-> next! = NULL) {
        q = p;
        p = p -> next;
  }
  _____
  return head;
```

empty list

only one Node

← Two pointer iteration

Head = 3w

pressure

Choose the correct alternative to replace the blank line.

(A) `q = NULL; p - > next = head; head = p;`

(B) `q -> next = NULL; head = p; p -> next = head;`

(C) `Head = p; p -> next = q;q -> next = NULL;`

(D) `q -> next = NULL; p -> next = head; head = p;`

Topic : Question

Consider the following function

```
int dosmothing ( struct node * q ){
int c = q -> data ;
if (q==null) return 0;
if (q -> link == NULL) return c;
else
c = c + dosmothing (q = q -> link) ;
return c ;
}
```

The above function run on the linked list that contain 1, 2, 3, 4, 5, 6, 7, 8. What will the content of the linked lists & the return value?

(A) 1, 2, 3, 4, 5, 6, 7, 8 return value 28

(B) 2, 1, 4, 3, 6, 5, 8, 7 return value 36

(C) 1, 2, 3, 4, 5, 6, 7, 8 return value 36

(D) 2, 1, 4, 3, 6, 5, 8, 7 return value 28

c = 1 + DS(2ω)

2 + DS(3ω)

3 + DS(4ω)

4 + DS(5ω)

5 + DS(6ω)

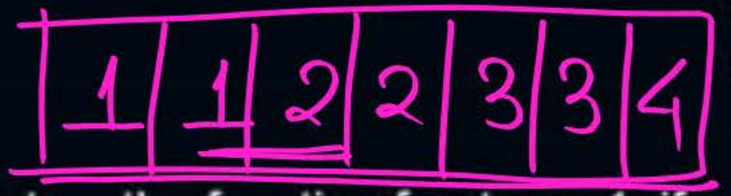
6 + DS(7ω)

7 + DS(8ω)

1+2+3+4+...+8

$= \frac{8 \times 9}{2} = 36$

Topic Question



Consider the function defined below.

```
struct item {
        int data;
        struct item * next;
};
int f(struct item *p) {
    return ((p == NULL) || (p->next == NULL) ||
        ((p->data <= p ->next -> data) &&
         f(p->next)));
}
```

For a given linked list p , the function f returns 1 if and only if

A the list is empty or has exactly one element

B the elements in the list are sorted in non-decreasing order of data value

C the elements in the list are sorted in non-increasing order of data value

D . not all elements in the list have the same data value

DM

order Increasing

Decreasing

1 <= 2

Slide

Topic : Single Linked List

Let's understand the problem!

Given a singly linked list, write a program to find the middle node of the linked list. If the number of nodes is even, we need to return the second middle node.

Example 1: Input: 5->4->3->2->1, Output: 3

Explanation: Here the number of nodes is 5, so there is one middle node which is 3.

Example 2: Input: 6->5->4->3->2->1, Output: 3

Explanation: The number of nodes is 6, where the first middle node is 4 and the second middle node is 3. So we need to return the pointer to the node 3.

Topic : Single Linked List

9

Insertx aftery . Insert a Node with data 5 after 2

* traversal required

Slide

Topic : Single Linked List

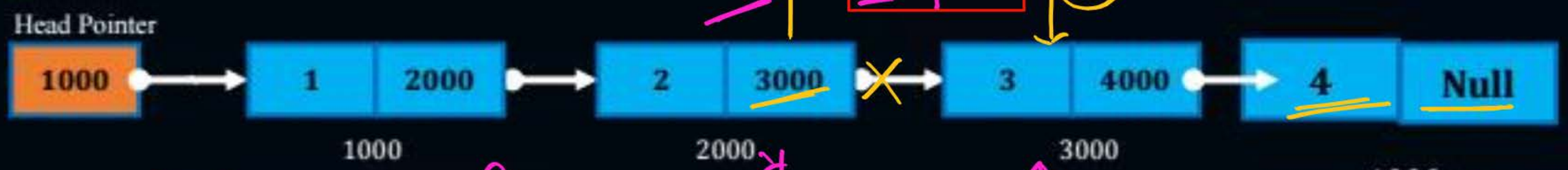
Insert x after y . Insert a Node with data 5 after 2

* traversal required
* We need to reach Node that contain y
* New Node required

```
q → next = p
p → next = r
```

Slide

Topic : Single Linked List



Insert x after y . Insert a Node with data after 2

```
p->next = q->next
q->next = p;
```

order is Imp

* traversal required
* We need to reach Node that contain y
* New Node required

Slide

Topic : Single Linked List

```
void InsertxofterY(int x, int y){
    Node* p = getnode(x);
    Node* q = HP, *r = NULL;
    if(q==NULL){
        free(p);
        return;
    }
    while(q->data!=y && q->next!=NULL)
        q = q->next;
    if(q->data==y){
        p->next = q->next;
        q->next = p;
    }
    else{
        printf("y Not present")
        free(p)
    }
}
```

Last Node

Single Node

Slide

2 mins Summary

Topic

Topic

Topic

Topic

Topic

proachic prooblem

Inseotxoftexy

Slide

THANK - YOU