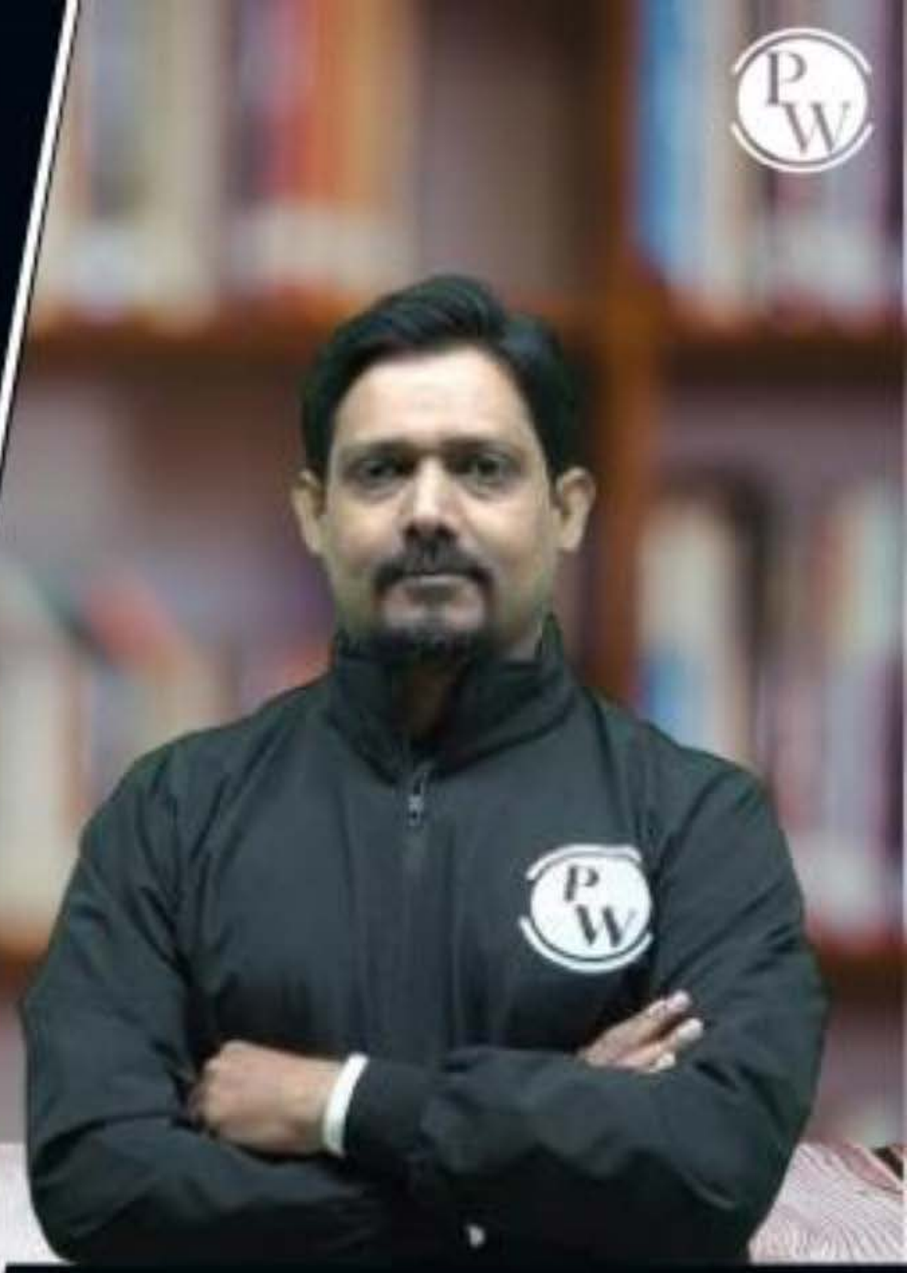
Computer Science & IT

Data Structure & Programming

Linked List

Lecture No. 01



By- Abhishek Sir

Recap of Previous Lecture

- Topic
- Topic
- Topic
- Topic
- Topic

Queue using stack

Linked list

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

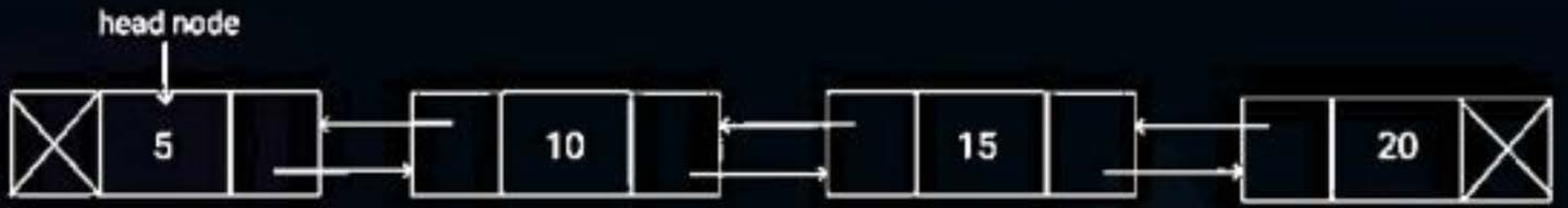
Single Linked List

Slide

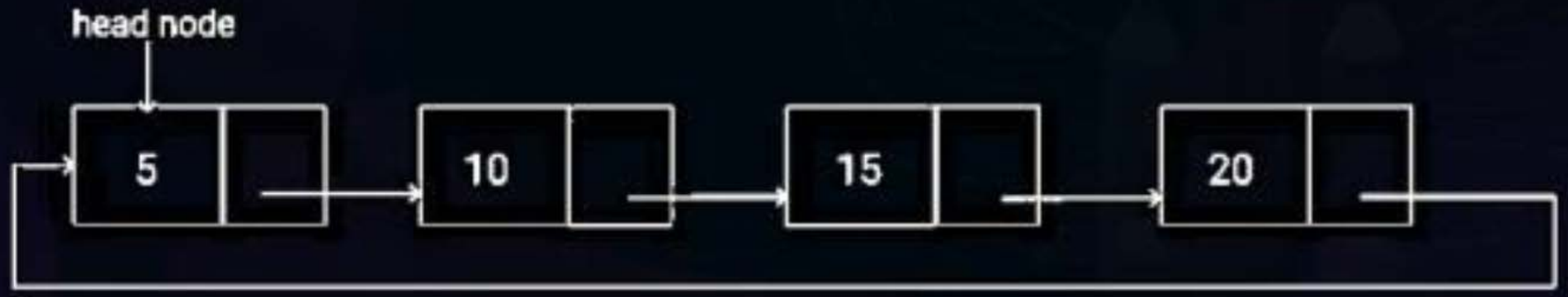
Topic : Linked List

Singly Linked List

Doubly Linked List



Circular Linked List



Topic : Single Linked List

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

```
Struct node a;
Node a;
```

Node a; ← Lazy writing

a.data;

"." Dot operator;

a.next;

Slide

Topic : Single Linked List

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

struct node

Node a;

Node * ptr;

pointer to structure

ptr = &a;

address of a assigned to ptr

Slide

Topic : Single Linked List

Using structure pointer if want to access Member of structure then `` → '' operator

```
ptr→data;
ptr→next;
```

Topic : Single Linked List

getnode function ✓

takes integer as parameter

create a Node &

return its address.

prototype

Node * getnode (int)

Address

Slide

Topic : Single Linked List

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

Data	Next

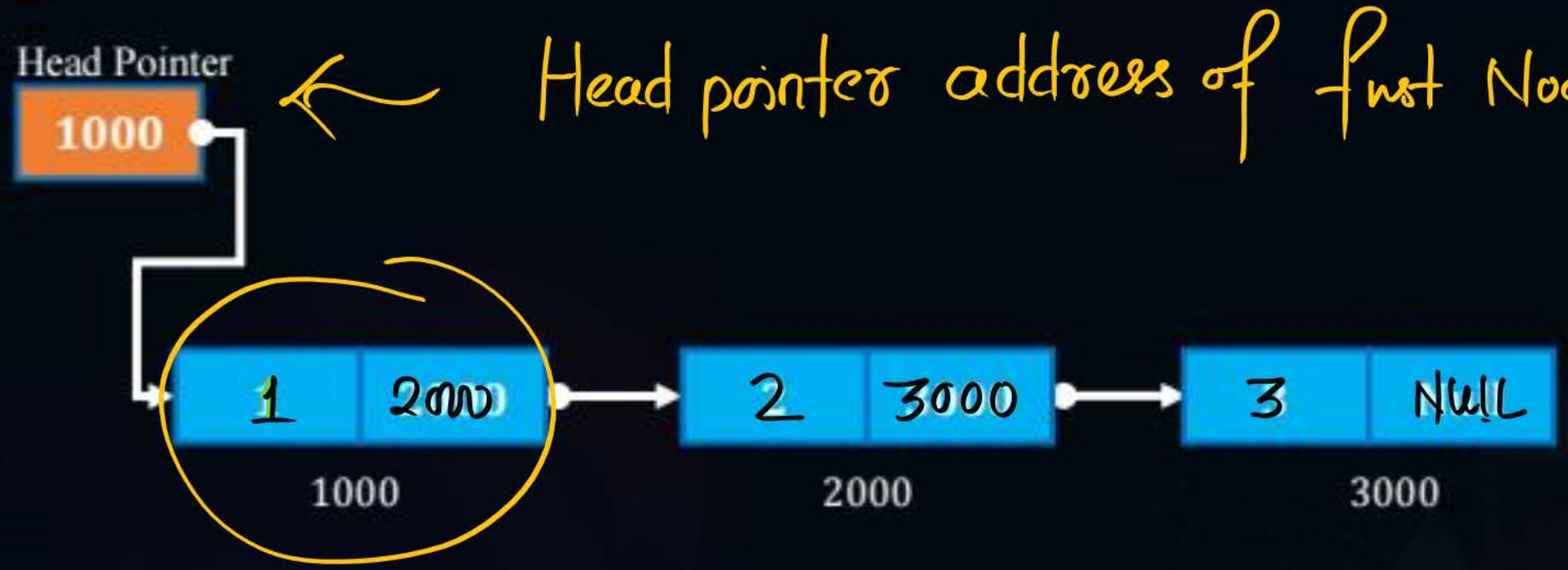
1000

1	NULL

100

Slide

Topic : Single Linked List



Slide

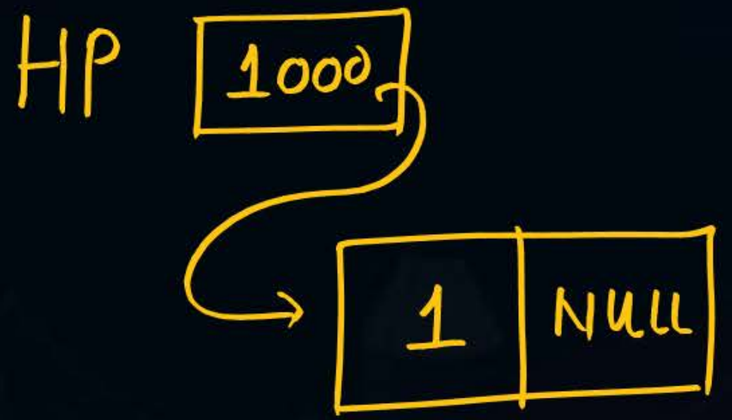
Topic : Single Linked List

Struct node a
↑ global variable

```
Node* HP;   ← global variable

int main() {

    HP = getnode(1);
```



Slide

Topic : Single Linked List

Build123()

Build 123 creates a linked list 1→2→3 ✓ and return address of first Node.

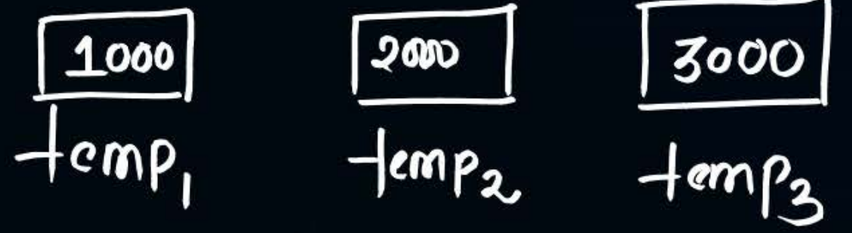
function prototype

Node * Build123( )

No parameter

Topic : Single Linked List


```
Node * Build123() {
    Node *temp1, *temp2, *temp3;
    temp1 = getnode(1);
    temp2 = getnode(2);
    temp3 = getnode(3);
    temp1 -> next = temp2;
    temp2 -> next = temp3
    return temp1;
}
```



Slide

Topic : Single Linked List

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

Node *HP; ✓
```

```
int main() {
  HP = Build123();
  print();
  printf("%d", count());
}
```

3

Slide

Topic : Single Linked List

Head pointer address of first Node of list

q = q→next , q = 2000; , q = 2000→next

q = 3000

1 , 2 , 3

q = 3000→next;

q = NULL;

Slide

Topic : Single Linked List

void print()

prints element in

Linked List

Traversal depends upon

Size of Linked List Hence

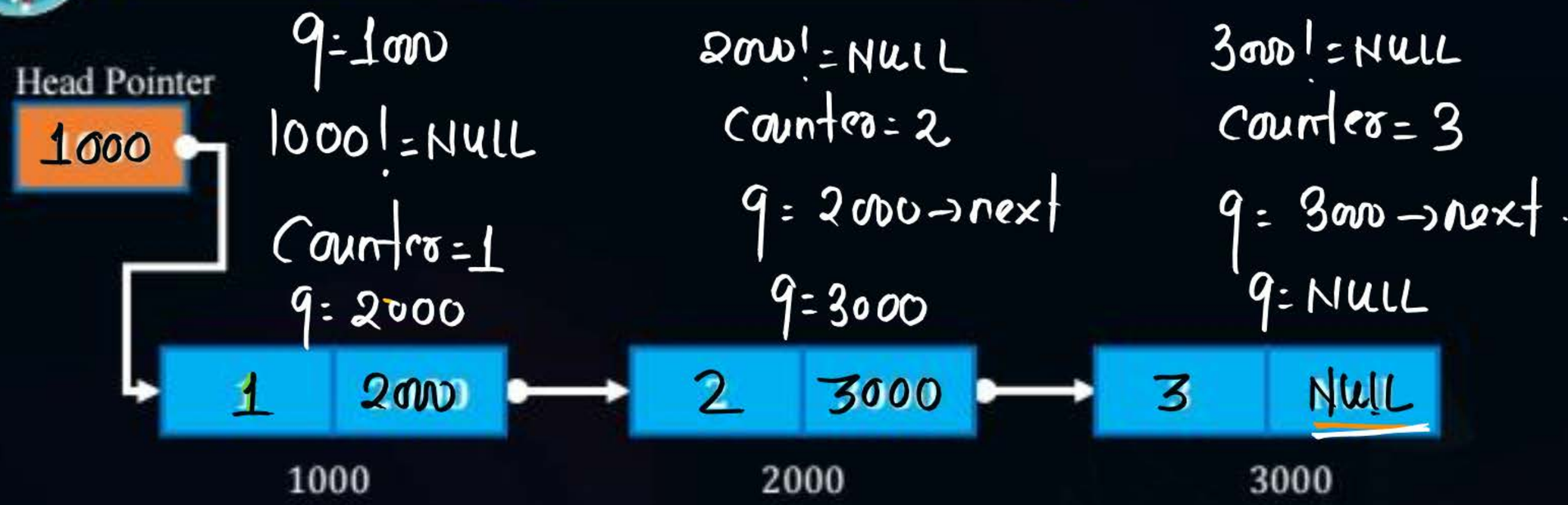
≈ n iteration for every case

```
void print() {
  Node *q = HP;
  while(q != NULL) {
    printf("%d", q->data);
    q = q->next;
  }
}
```

① q = 1000
② q = 2000
③ q = 3000;

Slide

Topic : Single Linked List



Slide

Topic : Single Linked List

Count function

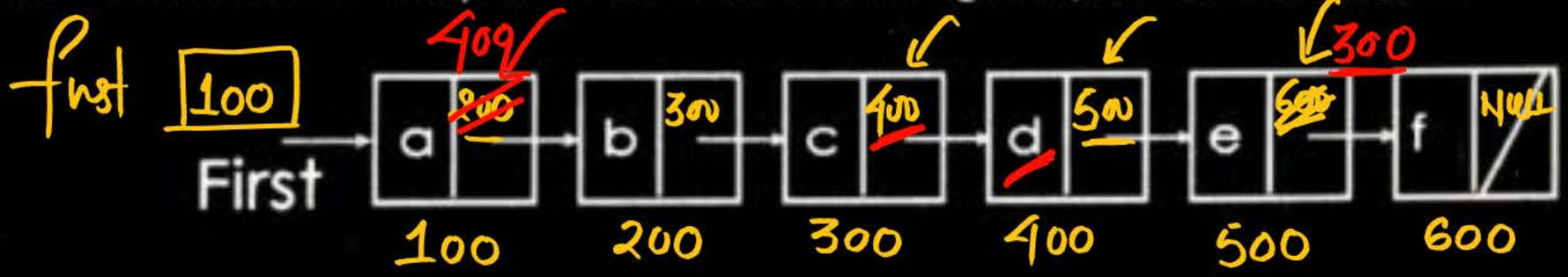
Count No. of Node in List and return its value.

```
int count ( ) {
    Node*q = HP;
    int counter = 0;
    while(q != NULL){
        counter = counter+1;
        q = q -> next;
    }
    return counter;
}
```

Slide

Topic : Question

02. What is the output after the following sequence of steps?



P = 300

```
struct node *P;
(i)   P = first → link → link;
(ii)  P → link → link → link = P;
(iii) first →link = P → link;
(iv)  printf ("%c", first → link → link →link → link → data);
```

(a) a (b) b (c) c (d) d

[P]

Slide

Topic : Question

first 100

What would be the output after the following sequence of steps?

```
struct node *p;
p = first → link → link → link;
p → link → link = p;
first → link → link = p → link;
printf ("%c", first → link → link → link → data);
```

p = 400

(a) a (b) b (c) c (d) d

[d]

Slide

2 mins Summary

Topic — Linked

Topic — getnode, Build123

Topic — Count print

Topic — practice proble

Topic

Slide

THANK - YOU