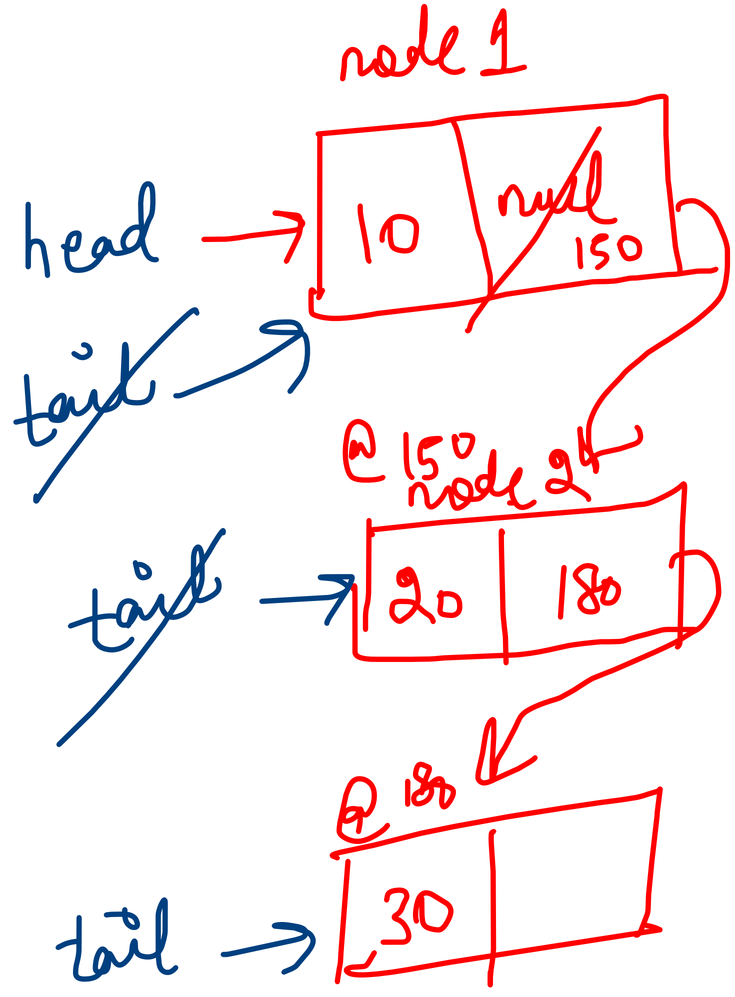
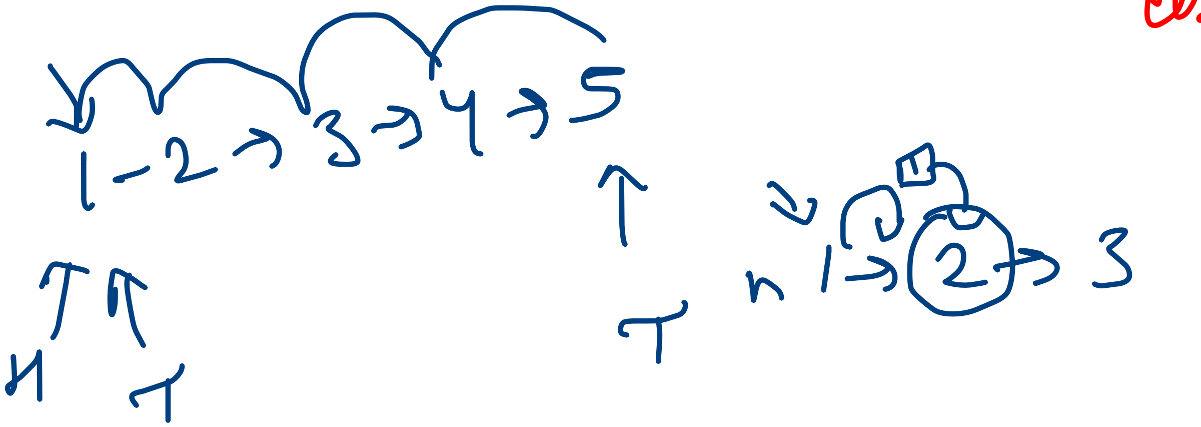
10 → 20

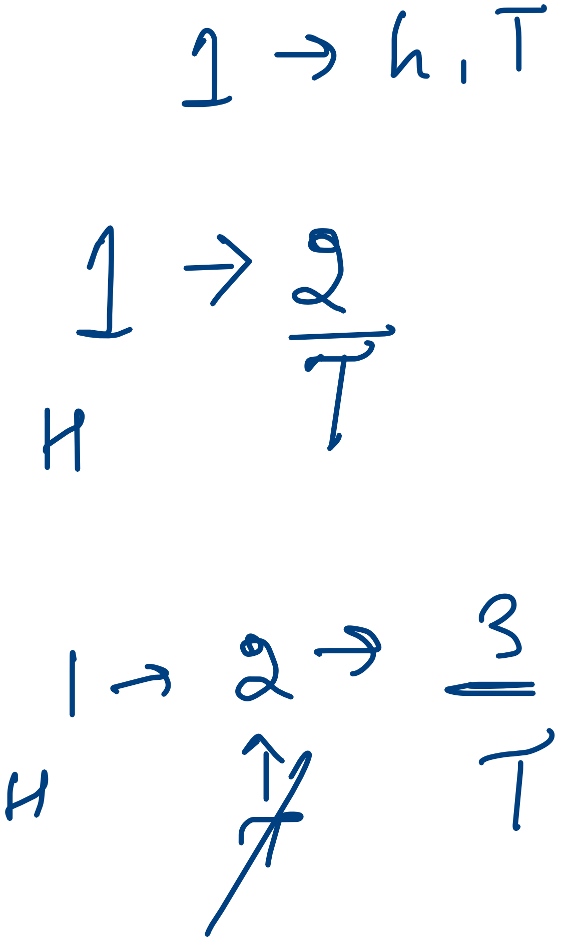




-1 → null

10 - head

```
head = null
tail = null
data = S.nextInt(); → 10
while (data != -1)
newNode(data)
if(head == null)
    head = newNode;
    tail = newNode;
else {
    tail.next = newNode;
    tail = newNode;
}
    data = S.next();
}
    return head.
```



tail = newNode

first take i/P → first node

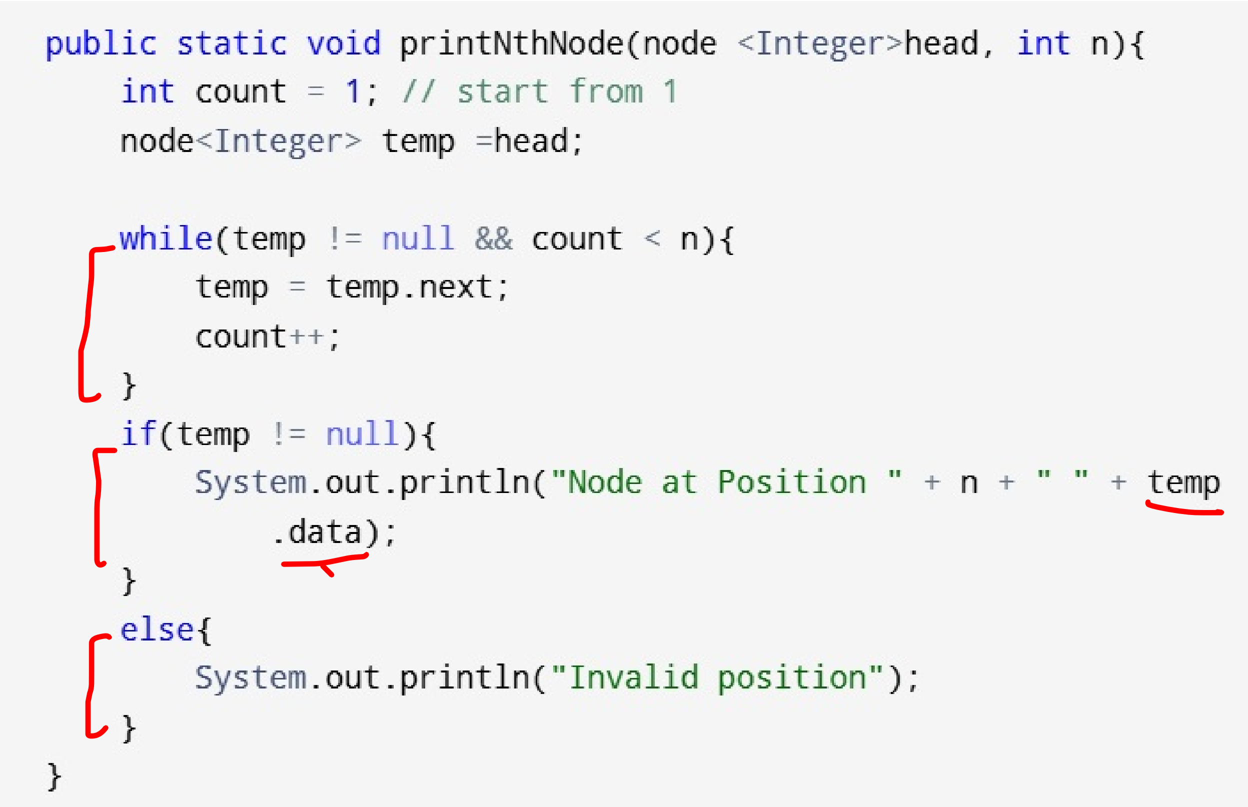
Second take i/P within loop → remaining nodes.

h
1 → 2 → 3 → 4 → 5 → null

3 → 3 o/p

Count = ~~1~~ ~~2~~ 3

h
1 → 2 → 3 → 4 → null
n=2



```
public static void printNthNode(node <Integer>head, int n){
    int count = 1; // start from 1
    node<Integer> temp =head;

    while(temp != null && count < n){
        temp = temp.next;
        count++;
    }
    if(temp != null){
        System.out.println("Node at Position " + n + " " + temp
            .data);
    }
    else{
        System.out.println("Invalid position");
    }
}
```

Count = 1

temp = 1

temp != null && 1 < 2) T

temp = 2 | o != null T

count = 2   2 < 2 F

temp != null

temp.data → 2 → o/p

TC → O(n)   SC → O(1)

parameter a, b → function define

Argument x, y → function call

sum(a, b)

return a+b

sum(x, y)

void

return