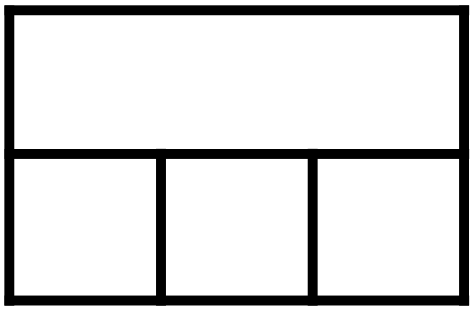
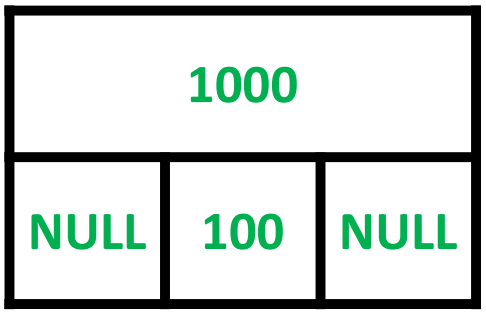
Empty Double Linked List



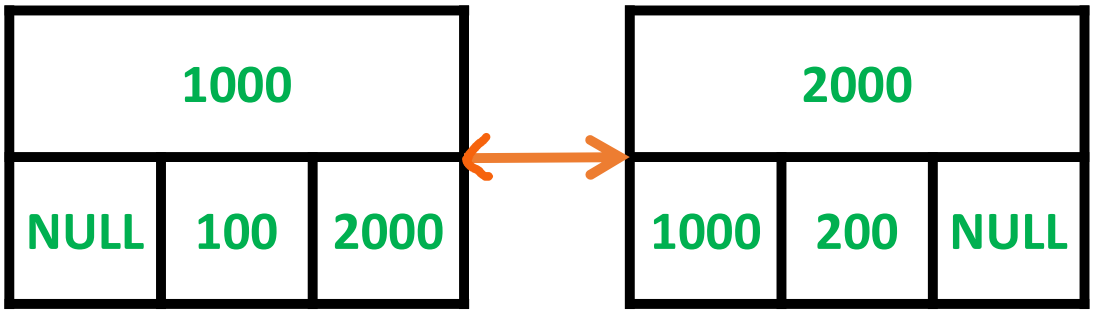
Head = NULL
Tail = NULL

Insert_At_Tail(100)



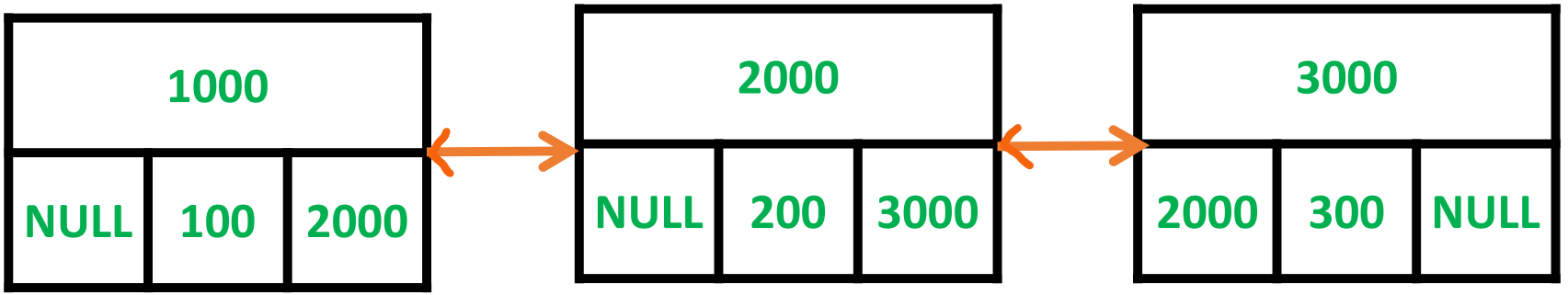
Head = 1000
Tail = 1000

Insert_At_Tail(200)



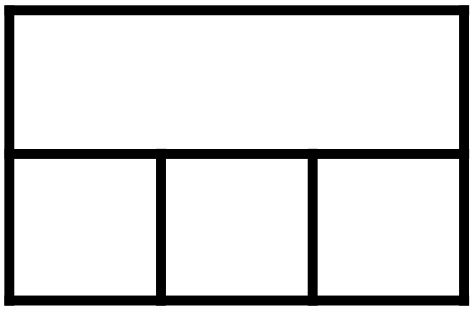
Head = 1000
Tail = 2000

Insert_At_Tail(300)



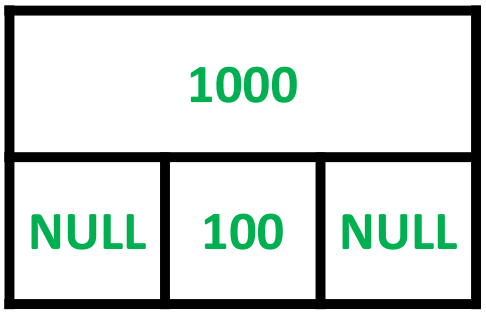
Head = 1000
Tail = 3000

Empty Double Linked List



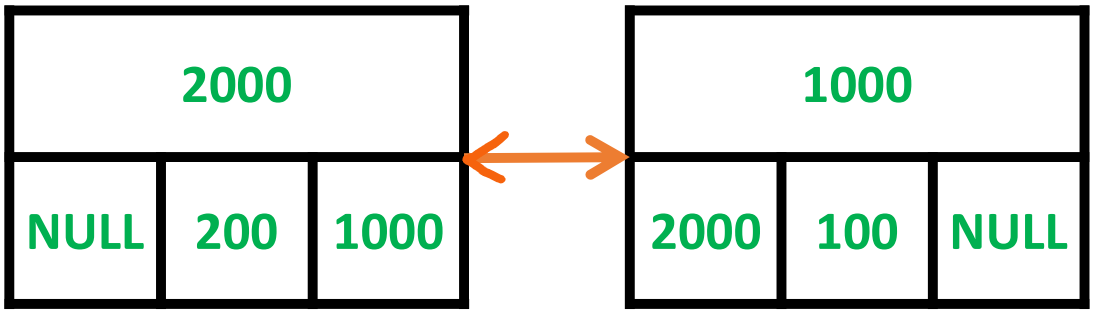
Head = NULL
Tail = NULL

Insert_At_Head(100)



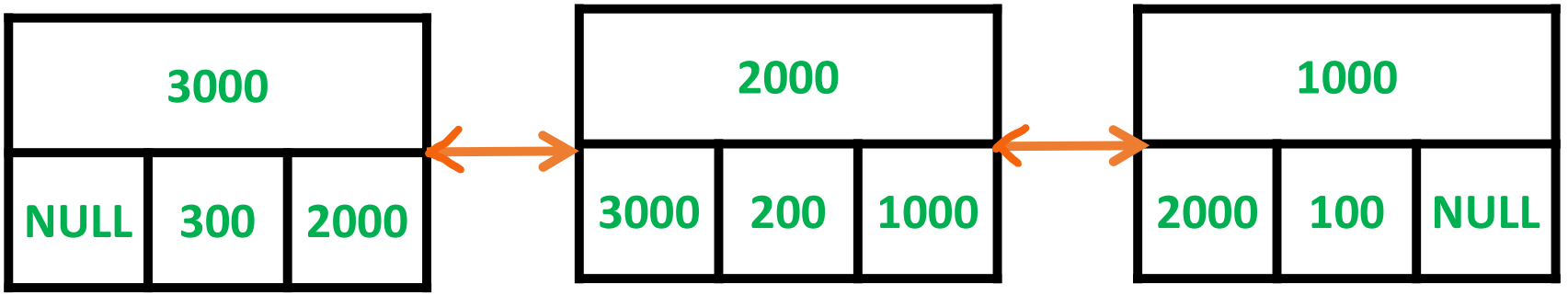
Head = 1000
Tail = 1000

Insert_At_Head(200)



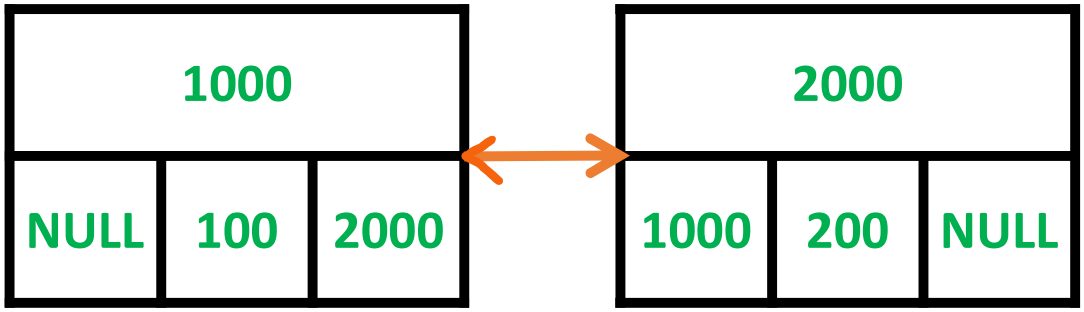
Head = 2000
Tail = 1000

Insert_At_Head(300)



Head = 3000
Tail = 1000

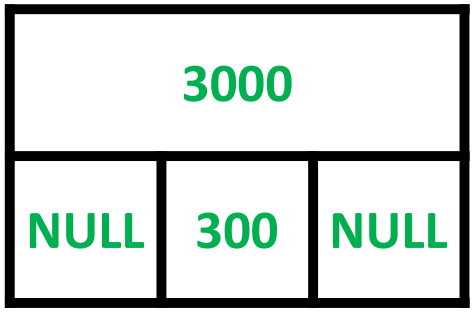
Double Linked List with 2 nodes



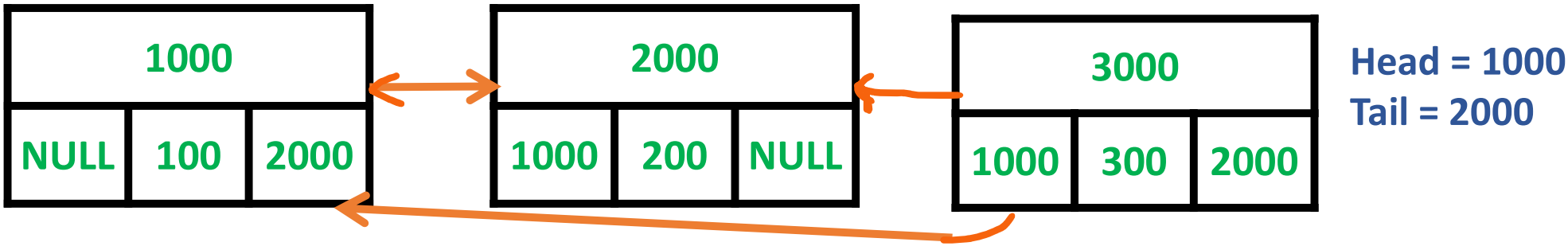
Head = 1000
Tail = 2000

Insert_At_Position(1,300)

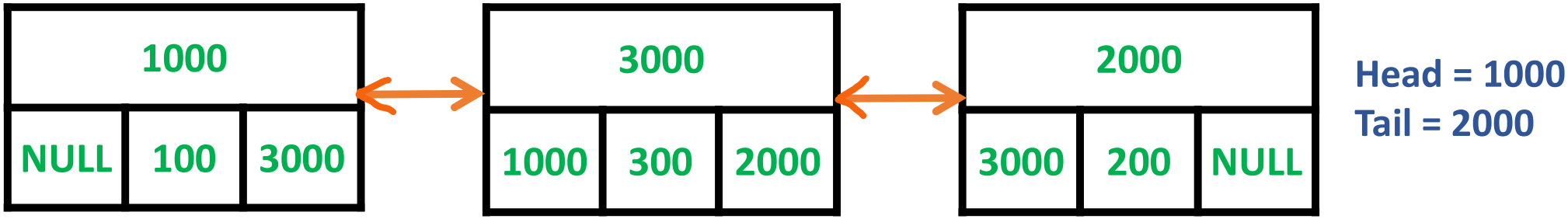
Step – 1 : Consider the new node with data as 300 at some address



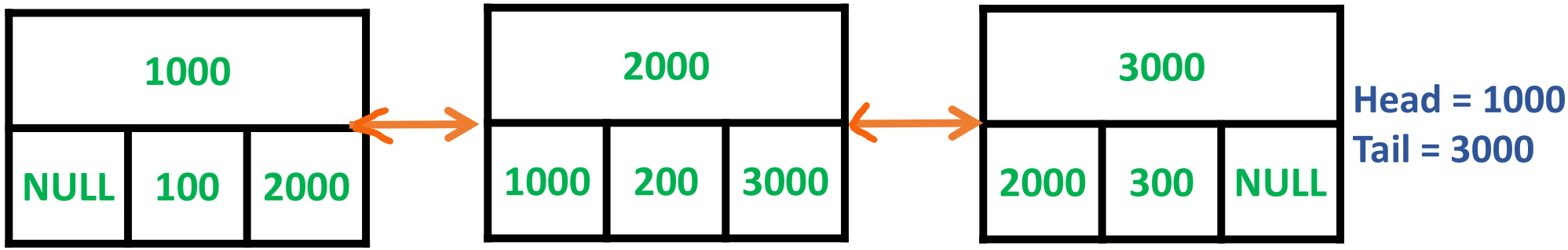
Step – 2 : Change the next node address of 300 as 2000 and prev node address as 1000



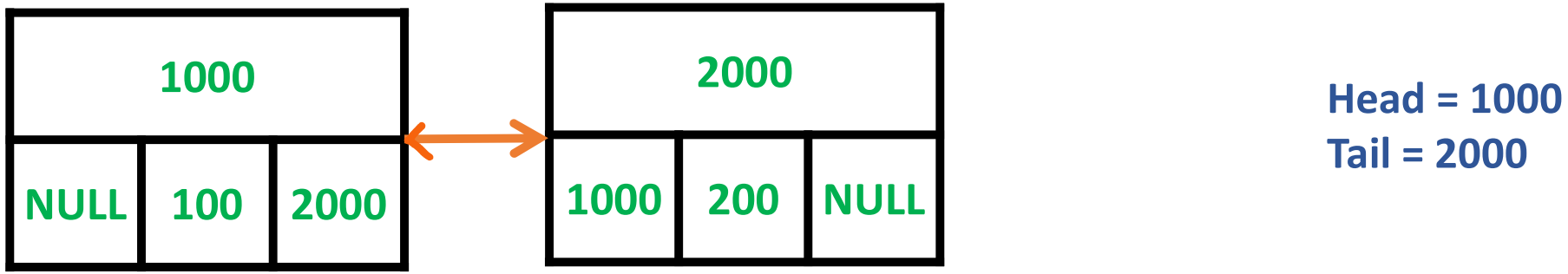
Step – 3 : Change the next node address of 100 as 3000 and prev node address of 200 as 3000



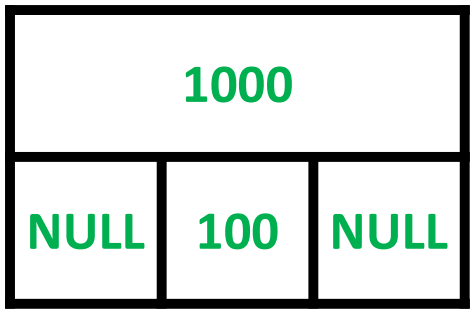
Double Linked List with three nodes



Delete_At_Tail()

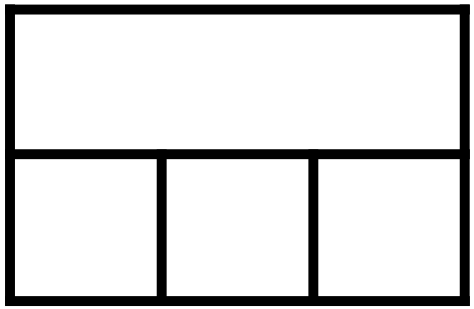


Delete_At_Tail()



Head = 1000
Tail = 1000

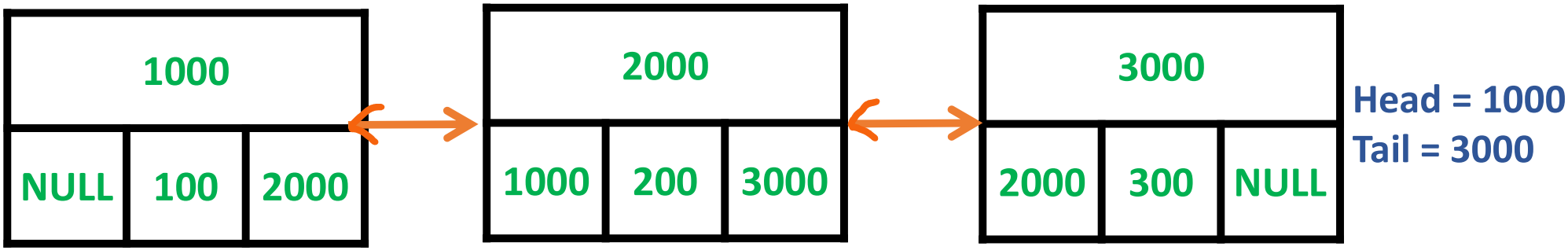
Delete_At_Tail()



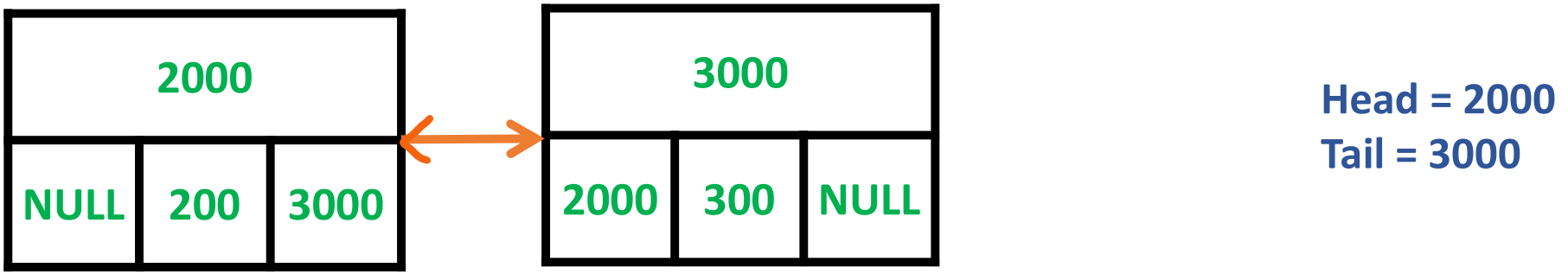
Head = NULL
Tail = NULL

Prints no nodes. Since Linked List is empty

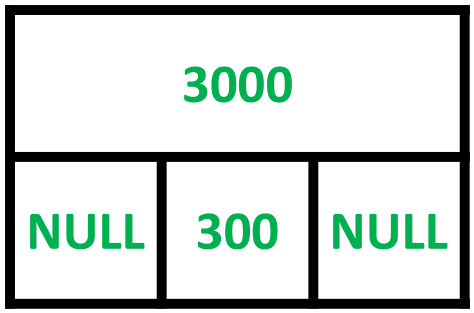
Double Linked List with three nodes



Delete_At_Head()

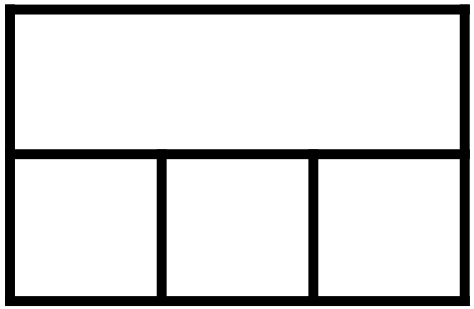


Delete_At_Head()



Head = 3000
Tail = 3000

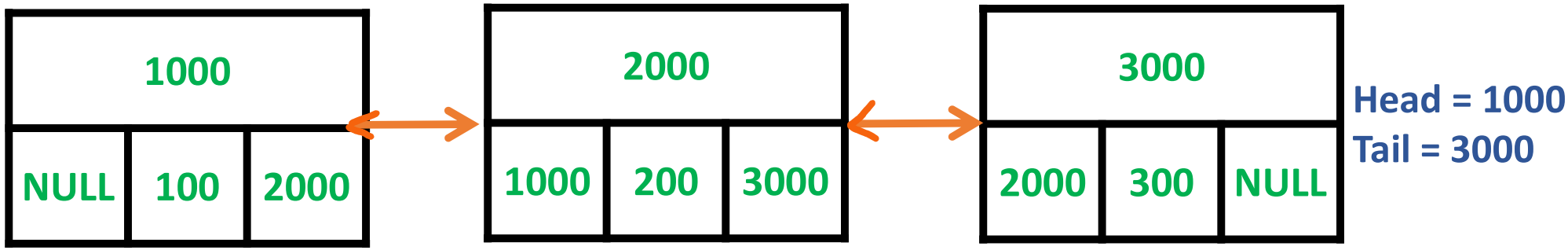
Delete_At_Head()



Head = NULL
Tail = NULL

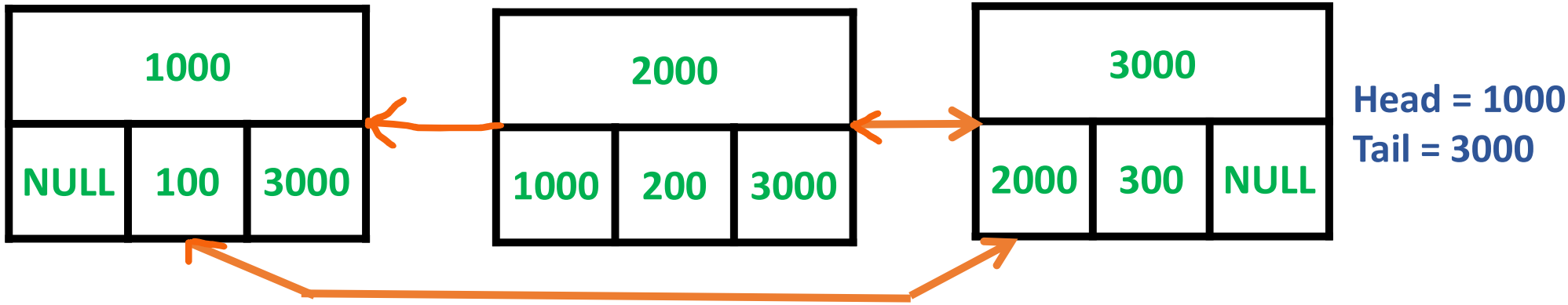
Prints no nodes. Since Linked List is empty

Double Linked List with three nodes

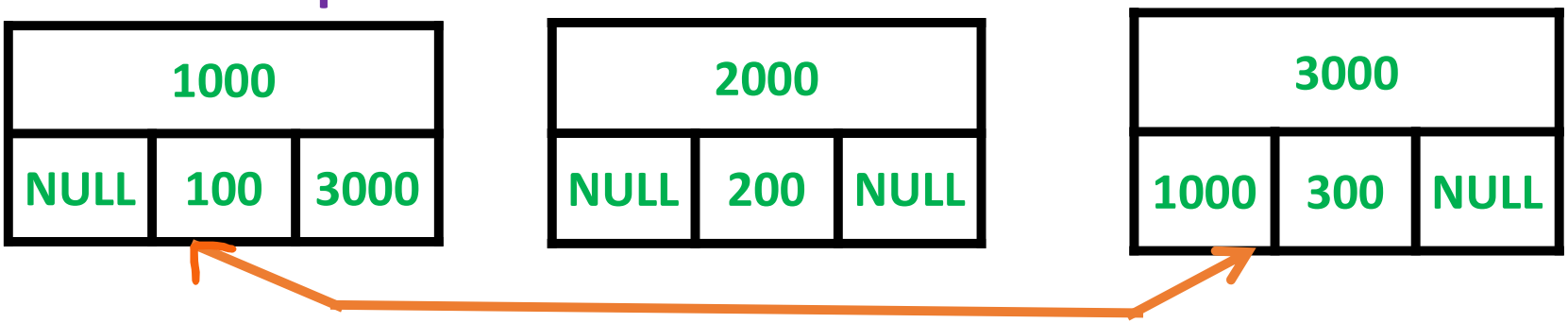


Delete_At_Position(1)

Step – 1 : Change the next node address of 0^{th} $[1-1]$ node as 3000



Step – 2 : Change the prev and next node address of 1^{st} $[1-1]$ node as NULL and prev node address of 2nd node as 1000



Step – 3 : Remove unwanted node and free its memory.