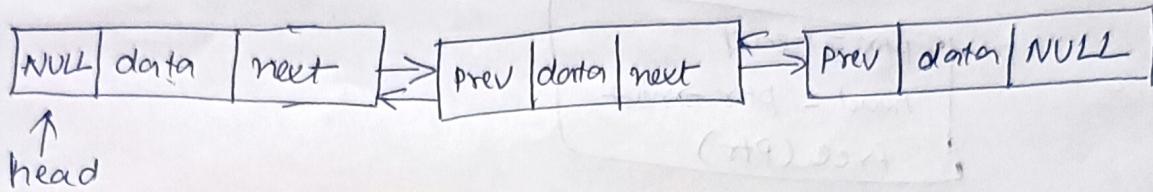
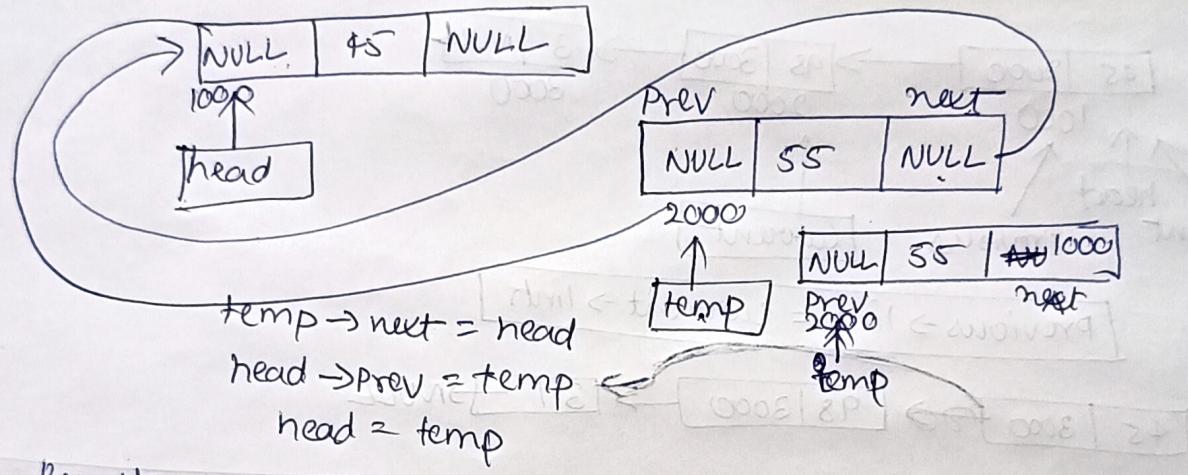


* Doubly linked list :-

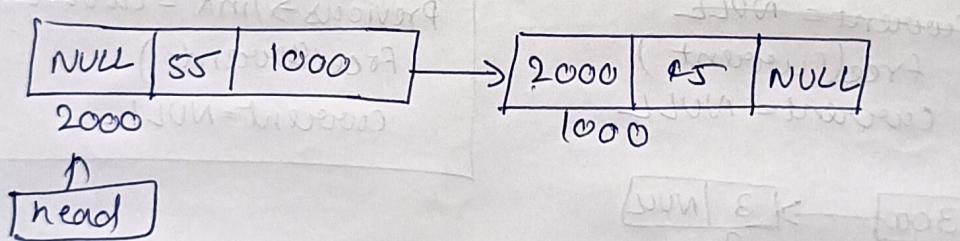
A doubly linkedlist is different from a singly linked-list in a way that each node has an extra pointer that points to the previous node, together with the next pointer and data similar to singly linked list.



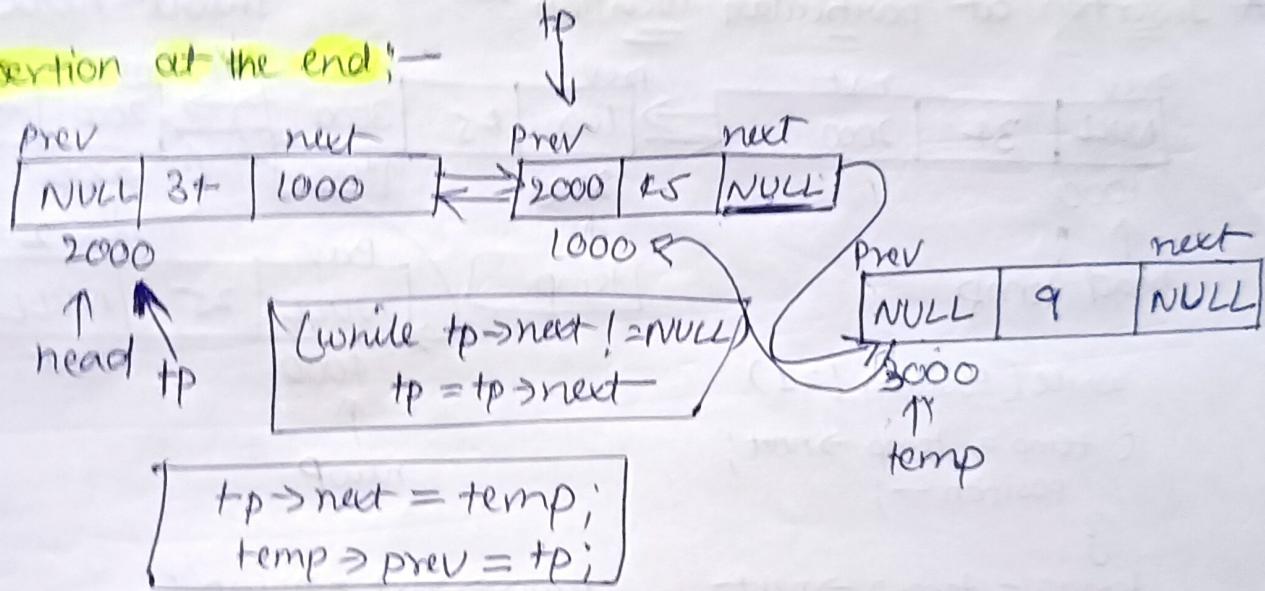
1) Insertion at the beginning :-



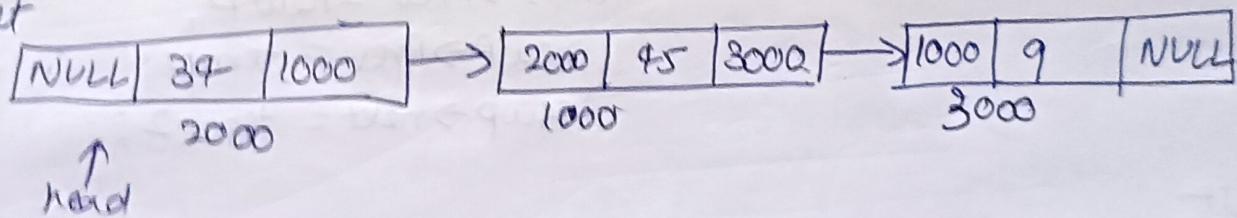
Result



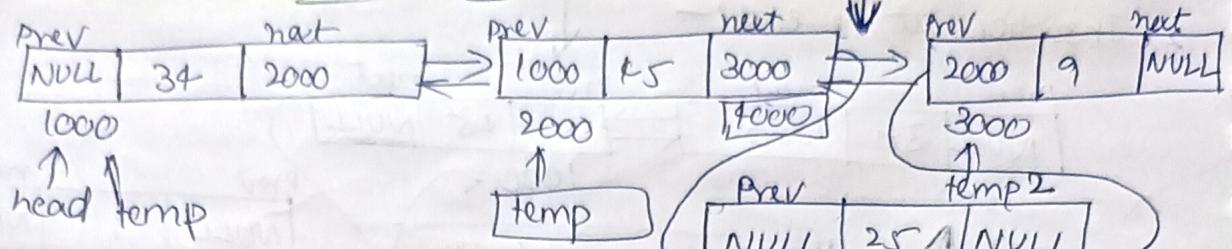
2) Insertion at the end:-



result



* Insertion at particular location :-



while (Position != 1)

{ temp = temp → next;
position --;

}

temp2 = temp2 → next

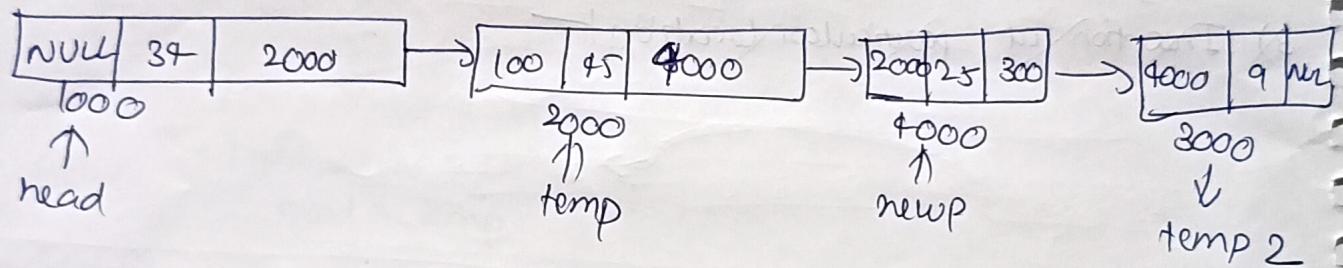
temp → next = newP ;

temp2 → prev = newP ;

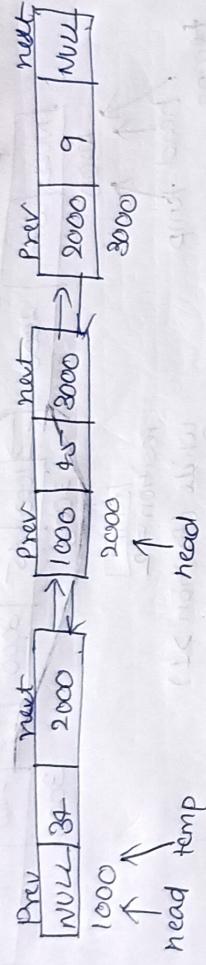
newP → prev = temp ;

newP → next = temp2 ;

Result



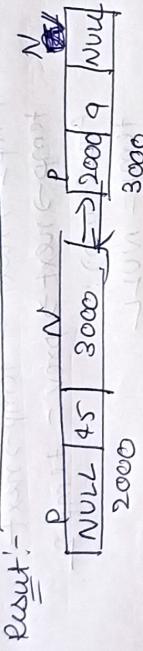
Doubly Linked List - Deleting the first node



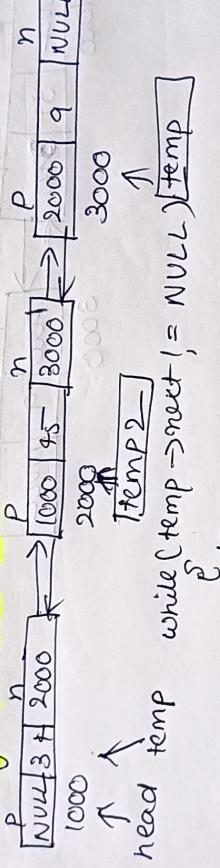
```

head = head->next;
free(temp);
head->prev = NULL;

```



* Deleting the last node



```

temp = temp->next;
while (temp->next != NULL) temp = temp->next;

```

```

temp = temp->next;

```

```

temp2 = temp->prev;

```

```

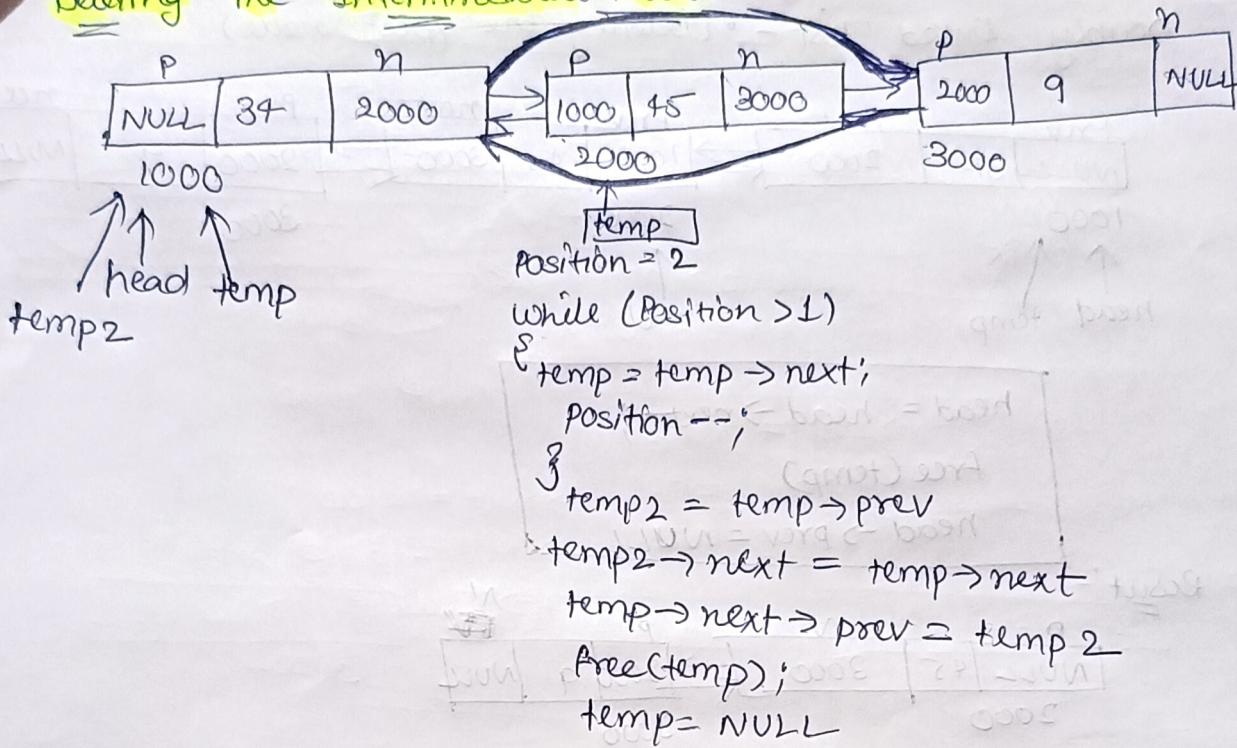
temp2->next = NULL;
free(temp);
temp = NULL;

```

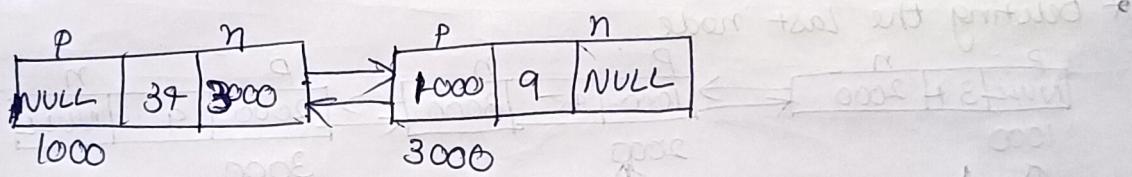
Result:-



Deleting the Intermediate Node



Result



$qnext = tnext < qnext \rightarrow$

$t + tnext < qnext = qnext$

$tnext < qnext = qnext$

$qnext = tnext < qnext$

$(qnext) == tnext$

$qnext = tnext$

