Deletion in doubly linked list at the end

Deletion of the last node in a doubly linked list needs traversing the list in order to reach the last node of the list and then make pointer adjustments at that position.

In order to delete the last node of the list, we need to follow the following steps.

- If the list is already empty then the condition head == NULL will become true and therefore the operation can not be carried on.
- If there is only one node in the list then the condition head → next == NULL become true. In this case, we just need to assign the head of the list to NULL and free head in order to completely delete the list.
- Otherwise, just traverse the list to reach the last node of the list. This will be done by using the following statements.

```
ptr = head;
    if(ptr->next != NULL)
    {
        ptr = ptr -> next;
    }
```

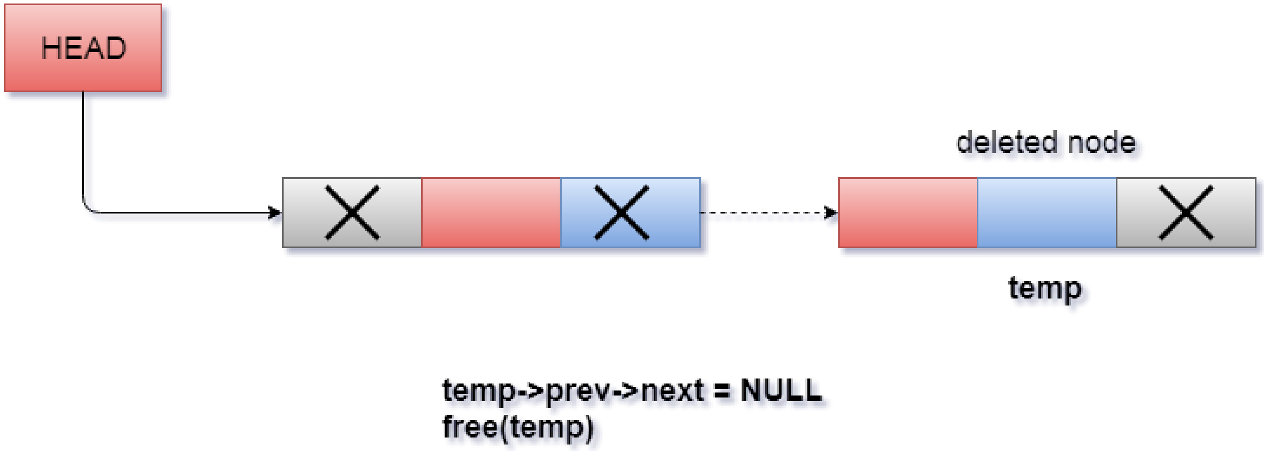
- The ptr would point to the last node of the ist at the end of the for loop. Just make the next pointer of the previous node of **ptr** to **NULL**.

```
ptr → prev → next = NULL
```

free the pointer as this the node which is to be deleted.

```
free(ptr)
```

```
Step 1: IF HEAD = NULL
    Write UNDERFLOW
    Go to Step 7
    [END OF IF]
Step 2: SET TEMP = HEAD
Step 3: REPEAT STEP 4 WHILE TEMP->NEXT != NULL
Step 4: SET TEMP = TEMP->NEXT
    [END OF LOOP]
Step 5: SET TEMP ->PREV-> NEXT = NULL
Step 6: FREE TEMP
Step 7: EXIT
```



Deletion in doubly linked list at the end

C Function

```
#include<stdio.h>
#include<stdlib.h>
void create(int);
void last_delete();
struct node
{
   int data;
   struct node *next;
   struct node *prev;
};
struct node *head;
void main ()
{
    int choice,item;
    do
    {
        printf("1.Append List\n2.Delete node from end\n3.Exit\n4.Enter your choice?");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            printf("\nEnter the item\n");
            scanf("%d",&item);
            create(item);
            break;
            case 2:
            last_delete();
            break;
            case 3:
            exit(0);
            break;
            default:
            printf("\nPlease enter valid choice\n");
        }

    }while(choice != 3);
}
void create(int item)
{

   struct node *ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
```

```
  {
     printf("\nOVERFLOW\n");
  }
  else
  {


  if(head==NULL)
  {
     ptr->next = NULL;
     ptr->prev=NULL;
     ptr->data=item;
     head=ptr;
  }
  else
  {
     ptr->data=item;
     ptr->prev=NULL;
     ptr->next = head;
     head->prev=ptr;
     head=ptr;
  }
   printf("\nNode Inserted\n");
}

}
void last_delete()
{
   struct node *ptr;
```

```
    if(head == NULL)
    {
        printf("\n UNDERFLOW\n");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nNode Deleted\n");
    }
    else
    {
        ptr = head;
        if(ptr->next != NULL)
        {
            ptr = ptr -> next;
        }
        ptr -> prev -> next = NULL;
        free(ptr);
        printf("\nNode Deleted\n");
    }
}
```

Output

```
1.Append List
2.Delete node from end
3.Exit
4.Enter your choice?1
```

```
Enter the item
12

Node Inserted
1.Append List
2.Delete node from end
3.Exit
4.Enter your choice?1

Enter the item
90

Node Inserted
1.Append List
2.Delete node from end
3.Exit
4.Enter your choice?2

Node Deleted
```

← prev next →

Please Share

Learn Latest Tutorials

Swift

Pig

Flask

C. Graphics

Automata

Testing

Preparation

⇧

Aptitude

Reasoning

Verbal A.

Interview

B.Tech / MCA

DBMS

DS

DAA

OS

C. Network

Compiler D.

COA

D. Math.

E. Hacking

Web Tech.

Cyber Sec.

C

C++

Java

.Net

Python

⇧

Control System

⇧