Insertion in doubly linked list at the end

In order to insert a node in doubly linked list at the end, we must make sure whether the list is empty or it contains any element. Use the following steps in order to insert the node in doubly linked list at the end.

- Allocate the memory for the new node. Make the pointer **ptr** point to the new node being inserted.

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

- Check whether the list is empty or not. The list is empty if the condition **head == NULL** holds. In that case, the node will be inserted as the only node of the list and therefore the prev and the next pointer of the node will point to NULL and the head pointer will point to this node.

```
ptr->next = NULL;
   ptr->prev=NULL;
   ptr->data=item;
   head=ptr;
```

- In the second scenario, the condition head == NULL become false. The new node will be inserted as the last node of the list. For this purpose, we have to traverse the whole list

in order to reach the last node of the list. Initialize the pointer **temp** to head and traverse the list by using this pointer.

```
Temp = head;
while (temp != NULL)
{
temp = temp → next;
   }
```

the pointer temp point to the last node at the end of this while loop. Now, we just need to make a few pointer adjustments to insert the new node ptr to the list. First, make the next pointer of temp point to the new node being inserted i.e. ptr.

```
temp→next =ptr;
```

make the previous pointer of the node ptr point to the existing last node of the list i.e. temp.

```
ptr → prev = temp;
```

make the next pointer of the node ptr point to the null as it will be the new last node of the list.

```
ptr → next = NULL
```

Algorithm

- **Step 1:** IF PTR = NULL

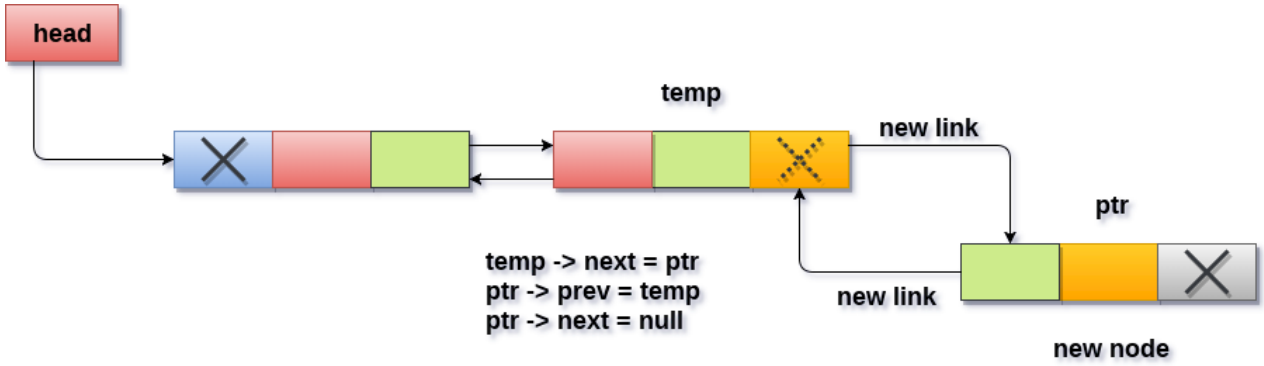
  Write OVERFLOW
  Go to Step 11
  [END OF IF]

- **Step 2:** SET NEW_NODE = PTR

- **Step 3:** SET PTR = PTR -> NEXT
- **Step 4:** SET NEW_NODE -> DATA = VAL
- **Step 5:** SET NEW_NODE -> NEXT = NULL
- **Step 6:** SET TEMP = START
- **Step 7:** Repeat Step 8 while TEMP -> NEXT != NULL
- **Step 8:** SET TEMP = TEMP -> NEXT

  [END OF LOOP]

- **Step 9:** SET TEMP -> NEXT = NEW_NODE
- **Step 10C:** SET NEW_NODE -> PREV = TEMP
- **Step 11:** EXIT



Insertion into doubly linked list at the end

C Program

```
#include<stdio.h>
#include<stdlib.h>
void insertlast(int);
struct node
{
   int data;
   struct node *next;
   struct node *prev;
};
struct node *head;
void main ()
{
   int choice,item;
   do
   {
      printf("\nEnter the item which you want to insert?\n");
      scanf("%d",&item);
      insertlast(item);
      printf("\nPress 0 to insert more ?\n");
```

```
        scanf("%d",&choice);
    }while(choice == 0);
}
void insertlast(int item)
{

   struct node *ptr = (struct node *) malloc(sizeof(struct node));
   struct node *temp;
   if(ptr == NULL)
   {
      printf("\nOVERFLOW");

   }
   else
   {

       ptr->data=item;
      if(head == NULL)
      {
         ptr->next = NULL;
         ptr->prev = NULL;
         head = ptr;
      }
      else
      {
        temp = head;
        while(temp->next!=NULL)
        {
           temp = temp->next;
```

```
        }
        temp->next = ptr;
        ptr ->prev=temp;
        ptr->next = NULL;
      }
 printf("\nNode Inserted\n");

   }
 }
```

Output

```
Enter the item which you want to insert?
12

Node Inserted

Press 0 to insert more ?
2
```

← 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

Programs

Control S.