# Insertion in doubly linked list after Specified node

In order to insert a node after the specified node in the list, we need to skip the required number of nodes in order to reach the mentioned node and then make the pointer adjustments as required.

Use the following steps for this purpose.

- Allocate the memory for the new node. Use the following statements for this.

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

- Traverse the list by using the pointer **temp** to skip the required number of nodes in order to reach the specified node.

```
temp=head;
  for(i=0;i<loc;i++)
  {
     temp = temp->next;
     if(temp == NULL) // the temp will be //null if the list doesn't last long //up to mentioned location
     {
          return;
     }
  }
```

- The temp would point to the specified node at the end of the **for** loop. The new node needs to be inserted after this node therefore we need to make a fer pointer adjustments here. Make the next pointer of **ptr** point to the next node of temp.

```
ptr → next = temp → next;
```

make the **prev** of the new node ptr point to temp.

```
ptr → prev = temp;
```

make the **next** pointer of temp point to the new node ptr.

```
temp → next = ptr;
```

make the **previous** pointer of the next node of temp point to the new node.

```
temp → next → prev = ptr;
```

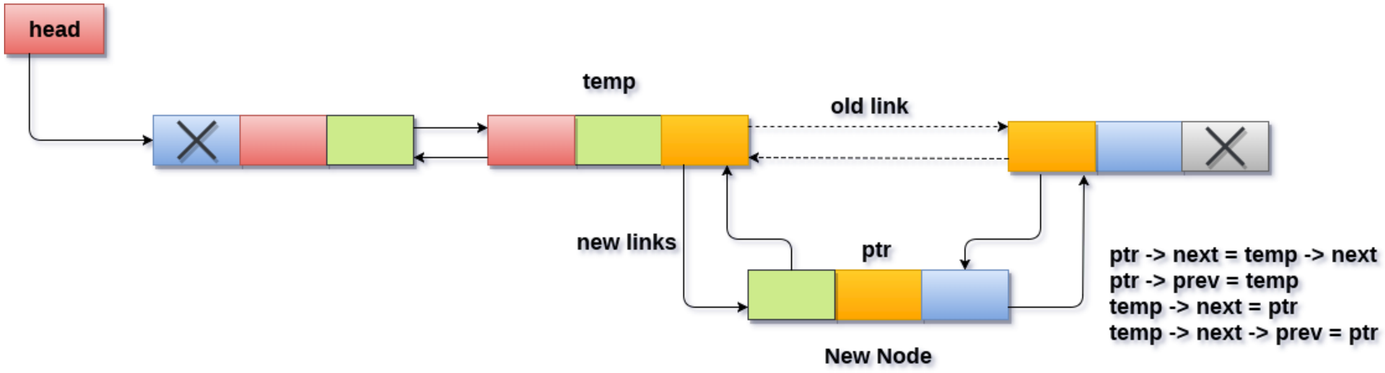
# Algorithm

```
Step 1: IF PTR = NULL

          Write OVERFLOW
          Go to Step 15
        [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 TEMP = START

Step 6: SET I = 0
```

- **Step 7:** REPEAT 8 to 10 until I
- **Step 8:** SET TEMP = TEMP -> NEXT
- **STEP 9:** IF TEMP = NULL
- **STEP 10:** WRITE "LESS THAN DESIRED NO. OF ELEMENTS"

```
      GOTO STEP 15
      [END OF IF]
    [END OF LOOP]
```

- **Step 11:** SET NEW_NODE -> NEXT = TEMP -> NEXT
- **Step 12:** SET NEW_NODE -> PREV = TEMP
- **Step 13 :** SET TEMP -> NEXT = NEW_NODE
- **Step 14:** SET TEMP -> NEXT -> PREV = NEW_NODE
- **Step 15:** EXIT



**Insertion into doubly linked list after specified node**

# C Function

```
#include<stdio.h>
#include<stdlib.h>
void insert_specified(int);
void create(int);
struct node
{
    int data;
    struct node *next;
    struct node *prev;
};
struct node *head;
void main ()
{
    int choice,item,loc;
    do
    {
        printf("\nEnter the item which you want to insert?\n");
        scanf("%d",&item);
        if(head == NULL)
        {
            create(item);
        }
        else
        {
            insert_specified(item);
        }
        printf("\nPress 0 to insert more ?\n");
        scanf("%d",&choice);
    }while(choice == 0);
}
void create(int item)
    {
   struct node *ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
   }
   else
   {


   if(head==NULL)
   {
       ptr->next = NULL;
       ptr->prev=NULL;
       ptr->data=item;
       head=ptr;
   }
   else
   {
       ptr->data=item;printf("\nPress 0 to insert more ?\n");
       ptr->prev=NULL;
       ptr->next = head;
       head->prev=ptr;
       head=ptr;
   }
    printf("\nNode Inserted\n");
}

}
void insert_specified(int item)
{

   struct node *ptr = (struct node *)malloc(sizeof(struct node));
   struct node *temp;
   int i, loc;
   if(ptr == NULL)
   {
       printf("\n OVERFLOW");
   }
   else
   {
       printf("\nEnter the location\n");
       scanf("%d",&loc);
       temp=head;
       for(i=0;i<loc;i++)
       {
           temp = temp->next;
           if(temp == NULL)
           {
               printf("\ncan't insert\n");
               return;
           }
       }
       ptr->data = item;
       ptr->next = temp->next;
       ptr -> prev = temp;
       temp->next = ptr;
       temp->next->prev=ptr;
       printf("Node Inserted\n");
   }
}
```

**Output**

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

Node Inserted

Press 0 to insert more ?
0

Enter the item which you want to insert?
90
```

```
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.