Insertion in singly linked list after specified Node

- In order to insert an element after the specified number of nodes into the linked list, we need to skip the desired number of elements in the list to move the pointer at the position after which the node will be inserted. This will be done by using the following statements.

```
emp=head;
        for(i=0;i<loc;i++)
        {
            temp = temp->next;
            if(temp == NULL)
            {
                return;
            }
        }
```

- Allocate the space for the new node and add the item to the data part of it. This will be done by using the following statements.

```
ptr = (struct node *) malloc (sizeof(struct node));
        ptr->data = item;
```

- Now, we just need to make a few more link adjustments and our node at will be inserted at the specified position. Since, at the end of the loop, the loop pointer temp would be pointing to the node after which the new node will be inserted. Therefore, the next part of the new node ptr must contain the address of the next part of the temp (since, ptr will be in between temp and the next of the temp). This will be done by using the following statements.

ptr→ next = temp → next

now, we just need to make the next part of the temp, point to the new node ptr. This will insert the new node ptr, at the specified position.

temp ->next = ptr;

Algorithm

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

  WRITE OVERFLOW
    GOTO STEP 12
    END OF IF

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

- **STEP 3:** NEW_NODE → DATA = VAL

- **STEP 4:** SET TEMP = HEAD

- **STEP 5:** SET I = 0

- **STEP 6:** REPEAT STEP 5 AND 6 UNTIL I

- **STEP 7:** TEMP = TEMP → NEXT

- **STEP 8:** IF TEMP = NULL

```
      WRITE "DESIRED NODE NOT PRESENT"
        GOTO STEP 12
      END OF IF
    END OF LOOP
```

- ○ **STEP 9:** PTR → NEXT = TEMP → NEXT

- ○ **STEP 10:** TEMP → NEXT = PTR

- ○ **STEP 11:** SET PTR = NEW_NODE

- ○ **STEP 12:** EXIT
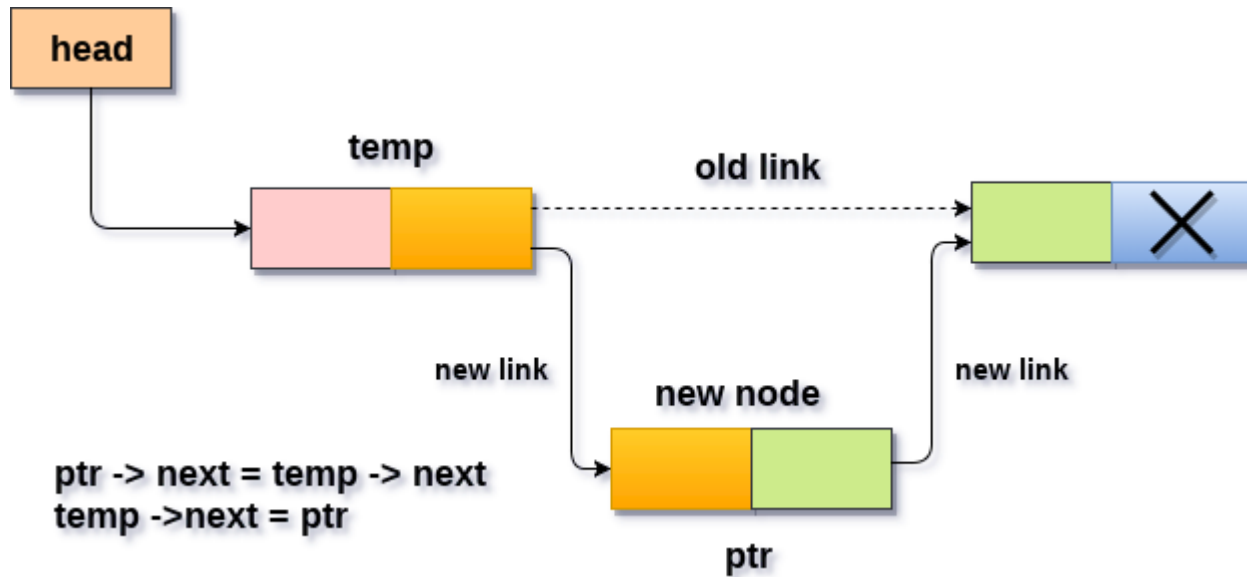
C Function

```c
#include<stdio.h>
#include<stdlib.h>
void randominsert(int);
void create(int);
struct node
{
    int data;
    struct node *next;
};
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)
```

```c
    {
        create(item);
    }
    else
    {
        randominsert(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\n");
    }
    else
    {
        ptr->data = item;
        ptr->next = head;
        head = ptr;
        printf("\nNode inserted\n");
    }
}
void randominsert(int item)
    {
```

```c
    struct node *ptr = (struct node *) malloc (sizeof(struct node));
    struct node *temp;
    int i,loc;
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {

        printf("Enter the location");
        scanf("%d",&loc);
        ptr->data = item;
        temp=head;
        for(i=0;i<loc;i++)
        {
            temp = temp->next;
            if(temp == NULL)
            {
                printf("\ncan't insert\n");
                return;
            }

        }
        ptr ->next = temp ->next;
        temp ->next = ptr;
        printf("\nNode inserted");
    }
```

```
    }
```

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.

Web Tech.

Cyber Sec.

C

C++

Java

.Net

Python

Programs

Control S.