

## Table of Content

- ✓ [Linked List Home](#)
- ✓ [Introduction to linked list](#)
- ✓ [Linked list advantages](#)
- ✓ [Linked list disadvantages](#)
- ✓ [Linked list Vs array](#)
- ✓ [Singly linked list](#)
- ✓ [Doubly linked list](#)
- ✓ [Circular linked list](#)
- ✓ [Singly linked list node structure](#)

## LL Operations

- ✓ [Create singly linked list](#)
- ✓ [Traversing through LL](#)

# Insert node at Last Position : Singly Linked List

[← Previous Page](#)

[Next Page →](#)

## Insert node at Last / End Position in Singly Linked List

### Inserting node at start in the SLL (Steps):

1. **Create New Node**
2. Fill Data into **"Data Field"**
3. Make it's **"Pointer"** or **"Next Field"** as **NULL**
4. Node is to be inserted at Last Position so we need to traverse **SLL upto Last Node.**
5. Make link between **last node and newnode**

```
void insert_at_end()
{
    struct node *new_node,*current;

    new_node=(struct node *)malloc(sizeof(struct node));

    if(new_node == NULL)
```

✓ Display Linked List from First to

Last

✓ Linked list terms

✓ Insert node at First Position

✓ **Insert node at Last Position**

✓ Insert node at middle position

✓ Delete Node from First Postion

✓ Search Perticular Element

✓ Counting number of Nodes in

Singly Linked List

✓ C Program to Create Singly

Linked List .using Node Structure

```
printf("\nFailed to Allocate Memory");
```

```
printf("\nEnter the data : ");  
scanf("%d",&new_node->data);  
new_node->next=NULL;
```

```
if(start==NULL)  
{  
    start=new_node;  
    current=new_node;  
}  
else  
{  
    temp = start;  
    while(temp->next!=NULL)  
    {  
        temp = temp->next;  
    }  
    temp->next = new_node;  
}  
}
```

**Diagram :**

**Attention :**

1. If starting node is not available then **"Start = NULL"** then following part is executed

```
if (start==NULL)
{
    start=new_node;
```

```
current=new_node;  
}
```

2. If we have previously created First or starting node then **“else part”** will be executed to insert node at start
3. Traverse Upto Last Node., So that **temp** can keep track of Last node

```
else  
{  
    temp = start;  
    while(temp->next!=NULL)  
    {  
        temp = temp->next;  
    }  
}
```

4. Make **Link between Newly Created node and Last node ( temp )**

```
temp->next = new_node;
```

**To pass Node Variable to Function Write it as -**

```
void insert_at_end(struct node *temp)
```

[↩ Previous Page](#)

[Next Page ➞](#)

Google Custom Search