



EDUCBA

(<https://www.educba.com/software-development/>)

← (<https://www.educba.com/memory-allocation-in-c/>)

→ (<https://www.educba.com/volatile-in-c/>)



## Introduction to Linked List in C

As the name suggests linked list means linking lists together or we can say that a linked list is a sequence of data structures that are connected to each other via links. Linked list use pointer for its implementation in the data structure. It's a linear data structure in which data is





(<https://www.educba.com/software-development/>)

## Syntax

### Start Your Free Software Development Course

Web development, programming languages, Software testing & others

Let's have a look at the syntax of representing a linked list in your code:

```
struct node {  
    Int data ;  
    struct node *next ;  
} ;
```

In the above-linked list syntax struct is the mandatory keyword to be used because with help of structure we can create custom data structure and as it is a node so node keyword is used so we are creating a data structure and in node, we have two parts, one is integer data part while other is the pointer which we address to the next node in the list.

Basically, the linked list node has two parts:

- **A data part:** It will contain the data of the user
- **A pointer part:** It will always point to the next member of the linked list in code.

## How Linked List work in C?



Now we will discuss the working of the linked list through C code with a brief explanation. Here is the C code to demonstrate the working of the linked list:



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdlib.h>

struct node {
    int data ;
    struct node *next ;
};

struct node *start = NULL ;
void insert_begin(int) ;
void insert_end(int) ;
void traverse() ;
void delete_begin() ;
void delete_end() ;
int count = 0 ;
int main () {
    int x, data ;
    for (;;) {
        printf("1. Want to insert an element at the beginning of linked list.\n") ;
        printf("2. Want to insert an element at the end of linked list.\n") ;
        printf("3. Want to traverse linked list.\n") ;
        printf("4. Want to delete an element from beginning.\n") ;
        printf("5. Want to delete an element from end.\n") ;
        printf("6. Exit\n") ;
        scanf("%d", &x) ;
        if (x == 1) {
            printf("Enter value of element\n") ;
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
printf("Enter value of element(x) : ");
scanf("%d", &data) ;
insert_end(data) ;
}
else if (x == 3)
traverse() ;
else if (x == 4)
delete_begin() ;
else if (x == 5)
delete_end() ;
else if (x == 6)
break ;
else
printf("Please enter valid input.\n") ;
}
return 0 ;
}

void insert_begin(int i) {
struct node *t ;
t = (struct node*)malloc(sizeof(struct node)) ;
t -> data = i;
count++ ;
if (start == NULL) {
start = t ;
start->next = NULL ;
return ;
}
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
void insert_end(int i) {  
    struct node *t, *temp ;  
    t = (struct node*)malloc(sizeof(struct node));  
    t -> data = i;  
    count++ ;  
    if (start == NULL) {  
        start = t ;  
        start->next = NULL ;  
        return ;  
    }  
    temp = start ;  
    while (temp->next != NULL)  
        temp = temp->next ;  
    temp->next = t ;  
    t->next = NULL ;  
}  
  
void traverse() {  
    struct node *t ;  
    t = start ;  
    if (t == NULL) {  
        printf("Linked list is empty.\n") ;  
        return ;  
    }  
  
    printf("There are %d elements in linked list.\n", count) ;  
    while (t->next != NULL) {  
        printf("%d\n", t->data) ;
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
void delete_begin() {  
    struct node *t ;  
    int n ;  
    if (start == NULL) {  
        printf("Linked list is empty.\n") ;  
        return ;  
    }  
    n = start->data ;  
    t = start->next ;  
    free(start) ;  
    start = t ;  
    count-- ;  
    printf("%d deleted from the beginning successfully.\n", n) ;  
}  
  
void delete_end() {  
    struct node *t, *u ;  
    int n;  
    if (start == NULL) {  
        printf("Linked list is empty.\n") ;  
        return ;  
    }  
    count-- ;  
    if (start->next == NULL) {  
        n = start->data ;  
        free(start) ;  
        start = NULL ;  
    }
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
while (t->next != NULL) {  
    u = t ;  
    t = t->next ;  
}  
n = t->data ;  
u->next = NULL ;  
free(t);  
printf( "%d deleted from end successfully.\n ", n) ;  
}
```

**Output:**





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

### Explanation:

In the above code, we have created a node and we have also created the address part of a node. We have added 5 main functionality of linked that help in performing all kinds of possible operations in our code. So we have declared insertion, deletion operations at the beginning as well as the end of the linked list. One more function is declared for traversing the list linked together.

🔗 Popular Course in this category



#### C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,618 ratings)

Course Price

**\$79** ~~\$399~~







(<https://www.educba.com/software-development/>)

development/courses/c-course/?btnz=edu-blg-inline-banner1)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

In the main class, we have declared two variables of integer data type as “l” and “data”. After declaring all the functionalities in beginning we will implement an if-else loop so that we can switch between the mentioned functionalities. For all the 5 functionalities we are implementing the logic in our code through an algorithm.

For insertion in the list at the beginning, we created a node t and in its data part, we named it as x. Therefore, if the start is null then start will be put in node t data part and the address part will point to the next part which is NULL. This process will insert the element at the beginning. In the same way, we have defined the logic for insertion and deletion at the start and end of the linked list in our code.

## Conclusion

Memory utilization plays a crucial role in the complex as well as simpler codes. Therefore, with the help of Linked list memory utilization can be done easily and efficiently. Most useful when we don't have an idea about the size of the list. It can grow and shrink accordingly.

## Recommended Articles

This is a guide to Linked List in C. Here we discuss the introduction to Linked List in C, a with the syntax, how Linked List work and an example to implement. You can also go through our other related articles to learn more –





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

---

## C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edublginlinebanner3>

---

### About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

## Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

## Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

## Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

[course/?source=footer\)](#)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

