

**DOPE** (/blog/)

Concatenating two linked lists in C.

🕒 May 25, 2017 📁 C (/blog/tag/c/?tag=c) LINKED LIST (/blog/tag/linked-list/?tag=linked-list) DATA STRUCTURE (/blog/tag/data-structure/?tag=data-structure) 👁 19793



Become an Author

(/blog/submit-article/)

Download Our App.



(<https://play.google.com/store/apps/details?id=com.blogsdope&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1>)

Previous:

1. Linked lists in C (Singly linked list)

(<https://www.codesdope.com/blog/article/linked-lists-in-c-singly-linked-list/>)

2. Linked list traversal using while loop and recursion

(<https://www.codesdope.com/blog/article/linked-list-traversal-using-while-loop-and-recursi/>)

Make sure that you are familiar with the concepts explained in the article(s) mentioned above before proceeding further.

We will proceed further by taking the linked list we made in the previous article.

```
#include <stdio.h>
#include <stdlib.h>

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

int main()
{
    struct node *prev, *head, *p;
    int n, i;
    printf ("number of elements:");
    scanf ("%d", &n);
    head=NULL;
    for(i=0; i<n; i++)
    {
        p=malloc(sizeof(struct node));
        scanf ("%d", &p->data);
        p->next=NULL;
        if(head==NULL)
            head=p;
        else
            prev->next=p;
        prev=p;
    }
    return 0;
}
```

Concatenating or joining two linked lists is not at all a difficult task. We just need to follow some very simple steps and the steps to join two lists (say 'a' and 'b') are as follows:

1. Traverse over the linked list 'a' until the element next to the node is not NULL.
2. If the element next to the current element is NULL (`a->next == NULL`) then change the element next to it to 'b' (`a->next = b`).

That's it. Let's code up these steps.

```
void concatenate(struct node *a, struct node *b)
{
    if (a->next == NULL)
        a->next = b;
    else
        concatenate(a->next, b);
}
```

Here, we are traversing over the article using recursion as explained in the article “Linked list traversal using while loop and recursion (<https://www.codesdope.com/blog/article/linked-list-traversal-using-while-loop-and-recursi/>)”. We are firstly checking if the next node (a->next) is NULL or not. If it is NULL, then we are just changing its value from NULL to ‘b’ (a->next = b) and if it is not then we are calling the ‘concatenate’ function again with the next element to traverse over the list.

So, the whole code is:

```
#include <stdio.h>
#include <stdlib.h>

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

display(struct node *head)
{
    if(head == NULL)
    {
        printf("NULL\n");
    }
    else
    {
        printf("%d\n", head -> data);
        display(head->next);
    }
}

void concatenate(struct node *a, struct node *b)
{
    if( a != NULL && b!= NULL )
    {
        if (a->next == NULL)
            a->next = b;
        else
            concatenate(a->next,b);
    }
    else
    {
        printf("Either a or b is NULL\n");
    }
}

int main()
{
    struct node *prev,*a, *b, *p;
    int n,i;
    printf ("number of elements in a:");
    scanf ("%d",&n);
    a=NULL;
    for(i=0;i<n;i++)
    {
        p=malloc(sizeof(struct node));
        scanf ("%d",&p->data);
        p->next=NULL;
        if(a==NULL)
            a=p;
        else
            prev->next=p;
        prev=p;
    }
    printf ("number of elements in b:");
    scanf ("%d",&n);
    b=NULL;
```

```

for(i=0;i<n;i++)
{
    p=malloc(sizeof(struct node));
    scanf("%d",&p->data);
    p->next=NULL;
    if(b==NULL)
        b=p;
    else
        prev->next=p;
    prev=p;
}
concatenate(a,b);
return 0;
}

```

Next:

1. Inserting a new node in a linked list in C

(<https://www.codesdope.com/blog/article/inserting-a-new-node-in-a-linked-list-in-c/>)

2. Deletion of a given node from a linked list in C

(<https://www.codesdope.com/blog/article/deletion-of-a-give-node-from-a-linked-list-in-c/>)

3. Array vs Linked list in C (<https://www.codesdope.com/blog/article/array-vs-linked-list-in-c/>)

Liked the post?



([https://www.facebook.com/sharer/sharer.php?](https://www.facebook.com/sharer/sharer.php?u=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/)

[u=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/](https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/))



([https://twitter.com/intent/tweet?url=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-](https://twitter.com/intent/tweet?url=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/&text=Concatenating%20two%20linked%20lists%20in%20C.%20&via=codesdope)

[c/&text=Concatenating two linked lists in C. &via=codesdope](https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/&text=Concatenating two linked lists in C. &via=codesdope))



([https://plus.google.com/share?](https://plus.google.com/share?url=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/)

[url=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/](https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/))



([https://www.linkedin.com/shareArticle?url=https://www.codesdope.com/blog/article/concatenating-two-linked-](https://www.linkedin.com/shareArticle?url=https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/&title=Concatenating two linked lists in C.)

[lists-in-c/&title=Concatenating two linked lists in C.\)](https://www.codesdope.com/blog/article/concatenating-two-linked-lists-in-c/)



([https://pinterest.com/pin/create/bookmarklet/?](https://pinterest.com/pin/create/bookmarklet/?media=https://www.codesdope.com/media/blog_images/1/2017/6/1/rings-)

[media=https://www.codesdope.com/media/blog_images/1/2017/6/1/rings-](https://www.codesdope.com/media/blog_images/1/2017/6/1/rings-)