**Objective -**

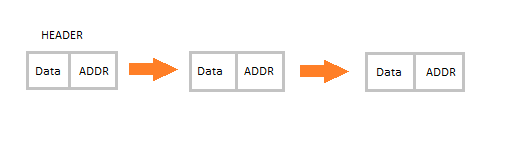
Write a program to concate two given linked list .

Description-

Linked List is a very commonly used linear data structure which consists of group of **nodes** in a sequence.

Each node holds its own **data** and the **address of the next node** hence forming a chain like structure.

Linked Lists are used to create trees and graphs.



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:

### Algorithm for concatenation

Let us assume that the two linked lists are referenced by **head1** and **head2** respectively.

1. If the first [linked list](https://www.thecrazyprogrammer.com/2013/12/c-program-to-create-binary-tree-using-recursion.html) is empty then return **head2**.

2. If the second linked list is empty then return **head1**.

3. Store the address of the starting node of the first linked  
list in a pointer variable, say **p**.

4. Move the **p** to  
the last node of the linked list through simple linked list traversal  
technique.

5. Store the address of the first node of the second linked  
list in the next field of the node pointed by **p**. Return **head1.**

Program –

#include<stdio.h> //header file

#include<stdlib.h>

struct node //structure definition

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*);

struct node \*concat( struct node \*start1,struct node \*start2);

struct node \*addatbeg(struct node \*start, int data);

struct node \*addatend(struct node \*start,int data);

void display(struct node \*start); //function declaration

main()

{

Clrscr();

struct node \*start1=NULL,\*start2=NULL;

start1=create\_list(start1);

start2=create\_list(start2);

printf("First list is : ");

display(start1); //function calling

printf("Second list is : ");

display(start2);

start1=concat(start1, start2);

printf("Concatenated list is : ");

display(start1);

}

struct node \*concat( struct node \*start1,struct node \*start2)

{

struct node \*ptr;

if(start1==NULL)

{

start1=start2;

return start1;

}

if(start2==NULL)

return start1;

ptr=start1;

while(ptr->link!=NULL) //while loop

ptr=ptr->link;

ptr->link=start2;

return start1;

}

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++) //for loop

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}

void display(struct node \*start) //function definition

{

struct node \*p;

if(start==NULL)

{

printf("List is empty\n");

return;

}

p=start;

while(p!=NULL)

{

printf("%d ", p->info);

p=p->link;

}

printf("\n");

}

struct node \*addatbeg(struct node \*start,int data)

{

struct node \*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

tmp->link=start;

start=tmp;

return start;

}

struct node \*addatend(struct node \*start, int data)

{

struct node \*p,\*tmp;

tmp= (struct node \*)malloc(sizeof(struct node));

tmp->info=data;

p=start;

while(p->link!=NULL)

p=p->link;

p->link=tmp;

tmp->link=NULL;

return start;

}

Output:-

