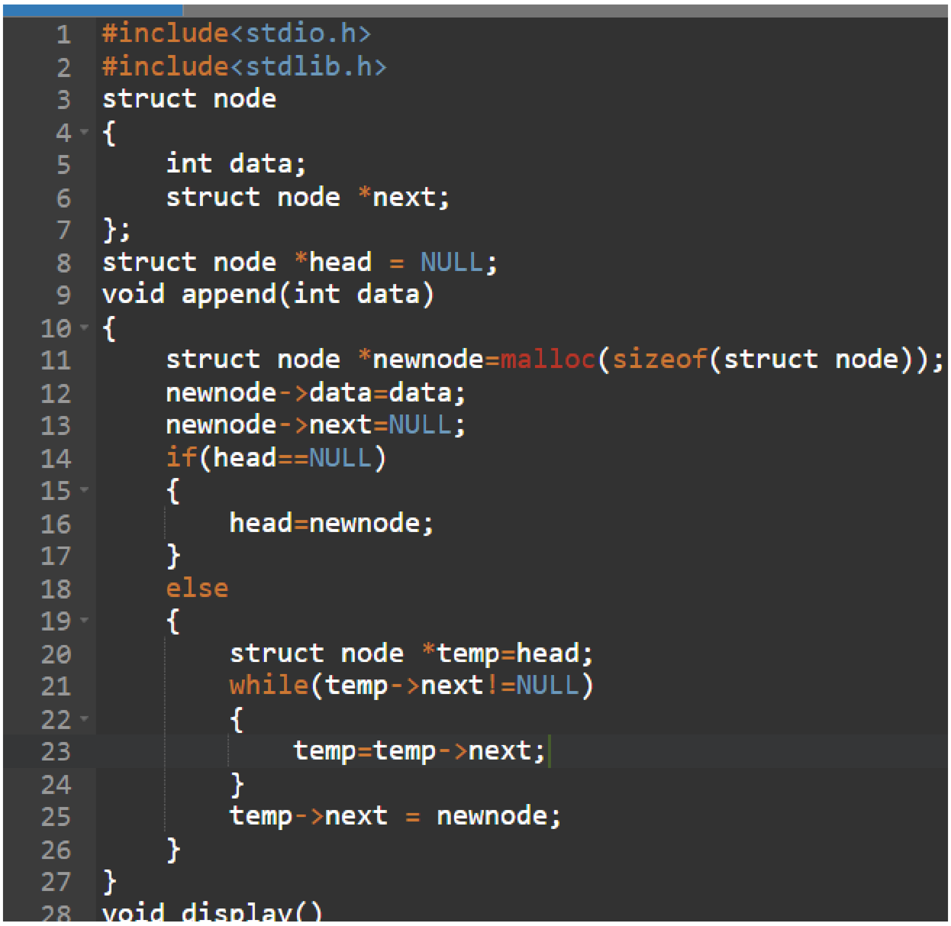
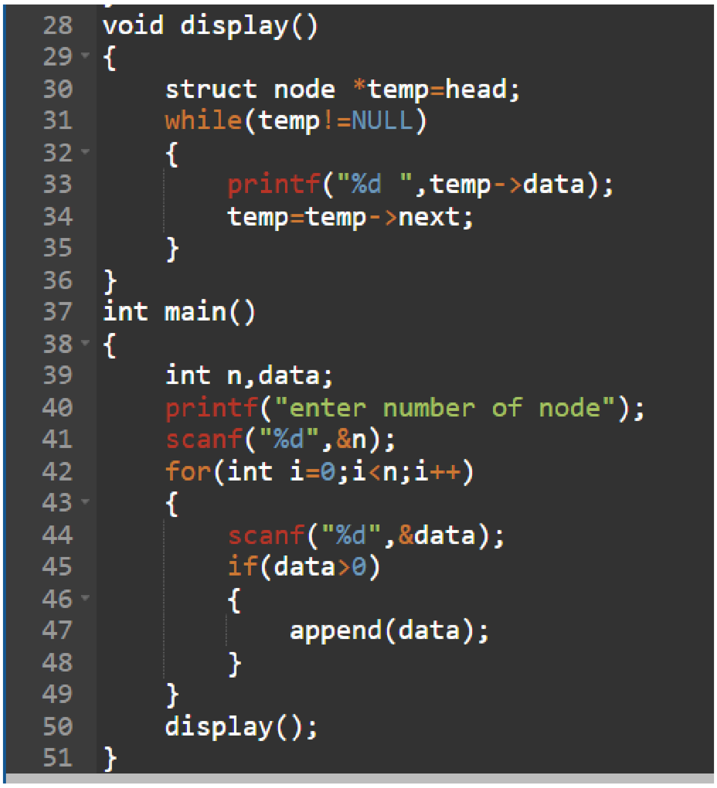
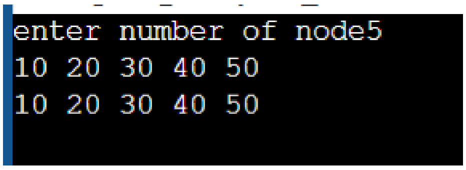
Singly linked list



```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head = NULL;
void append(int data)
{
    struct node *newnode=malloc(sizeof(struct node));
    newnode->data=data;
    newnode->next=NULL;
    if(head==NULL)
    {
        head=newnode;
    }
    else
    {
        struct node *temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next = newnode;
    }
}
void display()
```

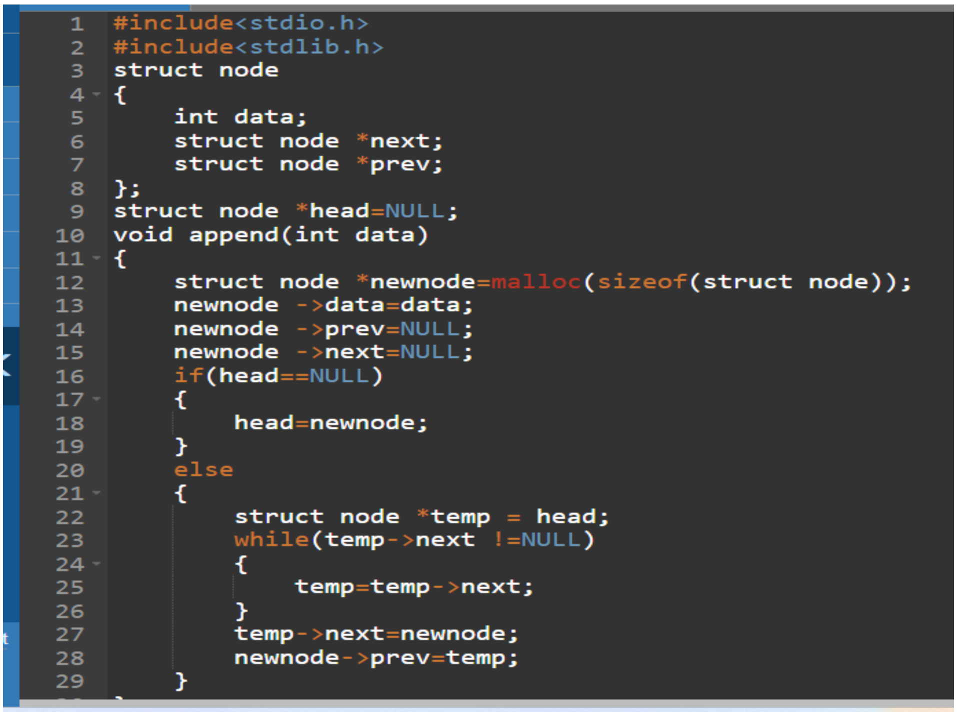


```c
void display()
{
    struct node *temp=head;
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
}
int main()
{
    int n,data;
    printf("enter number of node");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&data);
        if(data>0)
        {
            append(data);
        }
    }
    display();
}
```

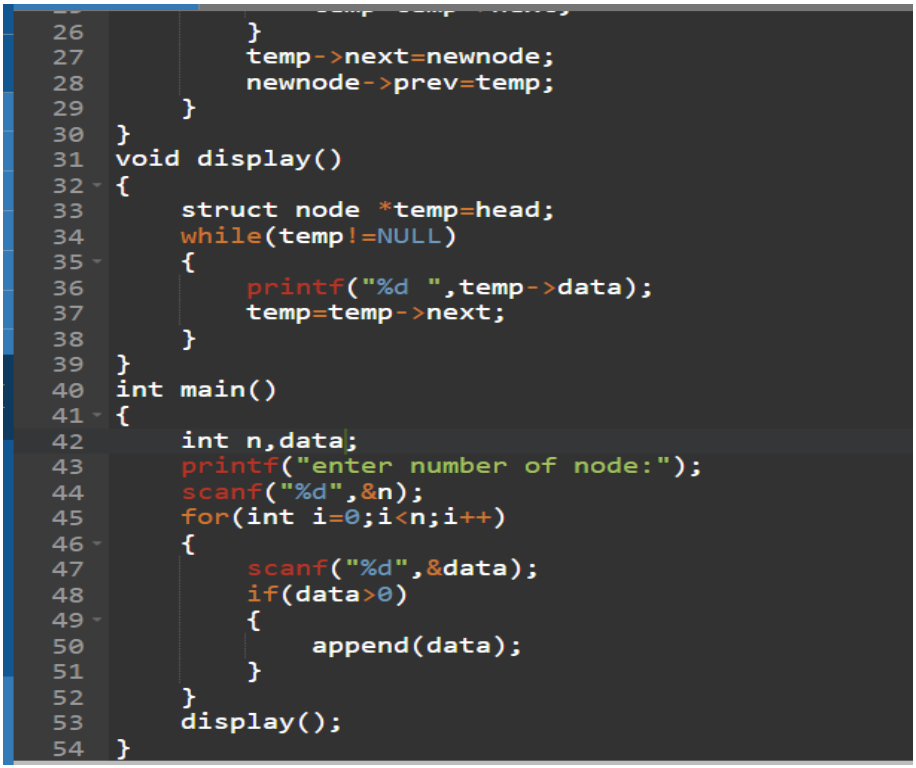


```
enter number of node5
10 20 30 40 50
10 20 30 40 50
```

Doubly linked list

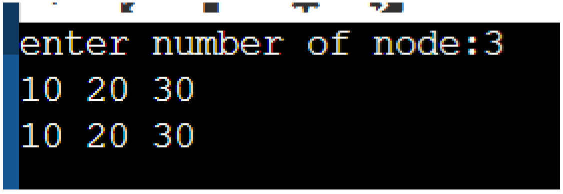


```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
	int data;
	struct node *next;
	struct node *prev;
};
struct node *head=NULL;
void append(int data)
{
	struct node *newnode=malloc(sizeof(struct node));
	newnode ->data=data;
	newnode ->prev=NULL;
	newnode ->next=NULL;
	if(head==NULL)
	{
		head=newnode;
	}
	else
	{
		struct node *temp = head;
		while(temp->next !=NULL)
		{
			temp=temp->next;
		}
		temp->next=newnode;
		newnode->prev=temp;
	}
```



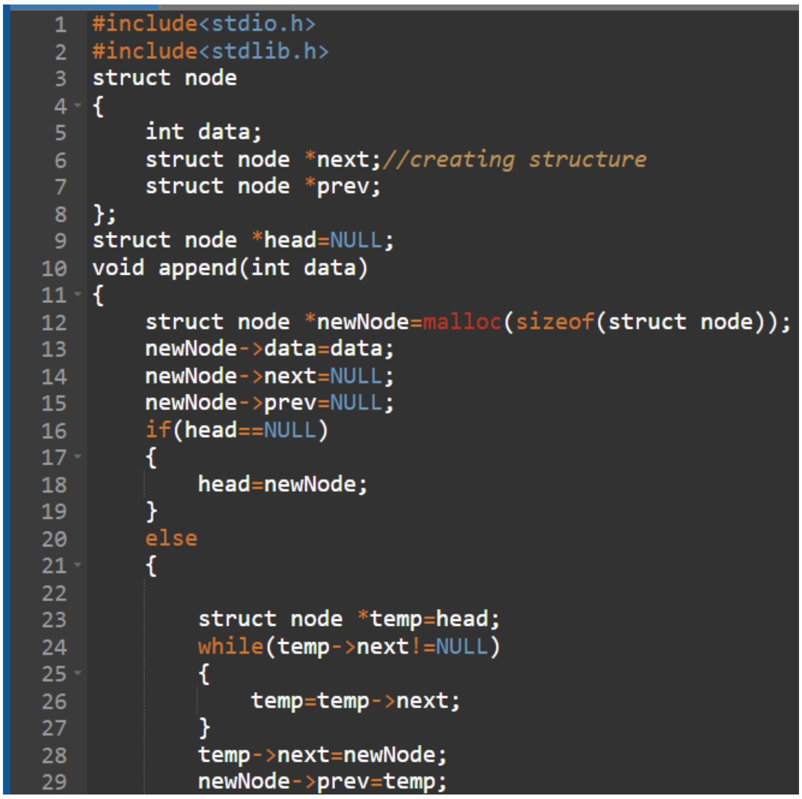
```c
		}
		temp->next=newnode;
		newnode->prev=temp;
	}
```

```c
}
void display()
{
	struct node *temp=head;
	while(temp!=NULL)
	{
		printf("%d ",temp->data);
		temp=temp->next;
	}
}
int main()
{
	int n,data;
	printf("enter number of node:");
	scanf("%d",&n);
	for(int i=0;i<n;i++)
	{
		scanf("%d",&data);
		if(data>0)
		{
			append(data);
		}
	}
	display();
}
```



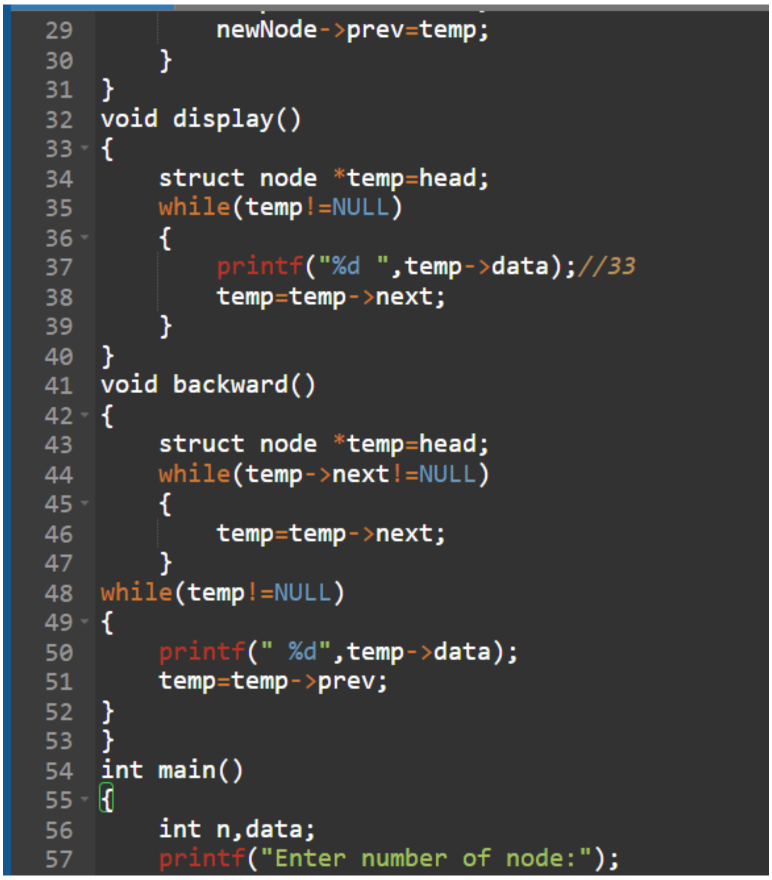
```
enter number of node:3
10 20 30
10 20 30
```

Printing in reverse

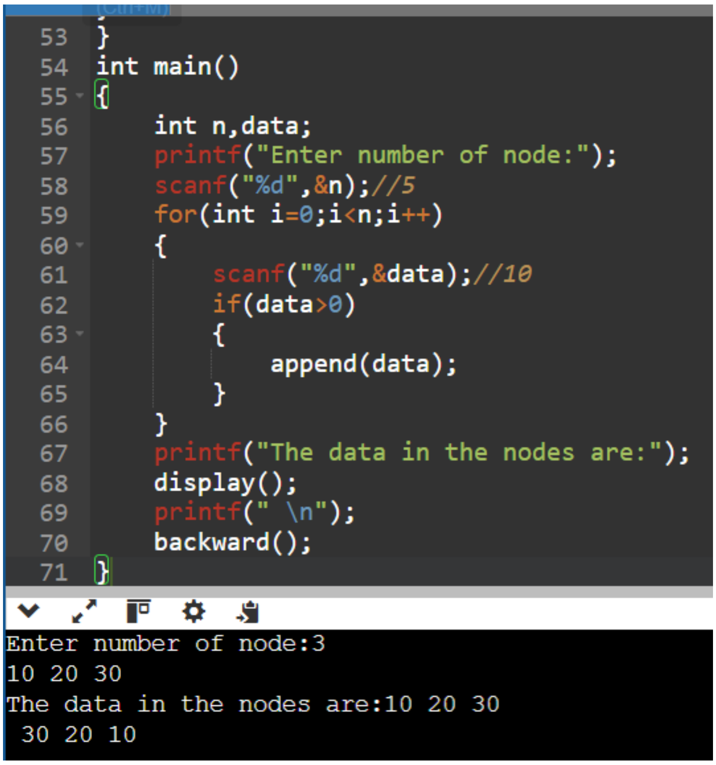


```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;//creating structure
    struct node *prev;
};
struct node *head=NULL;
void append(int data)
{
    struct node *newNode=malloc(sizeof(struct node));
    newNode->data=data;
    newNode->next=NULL;
    newNode->prev=NULL;
    if(head==NULL)
    {
        head=newNode;
    }
    else
    {

        struct node *temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newNode;
        newNode->prev=temp;
```



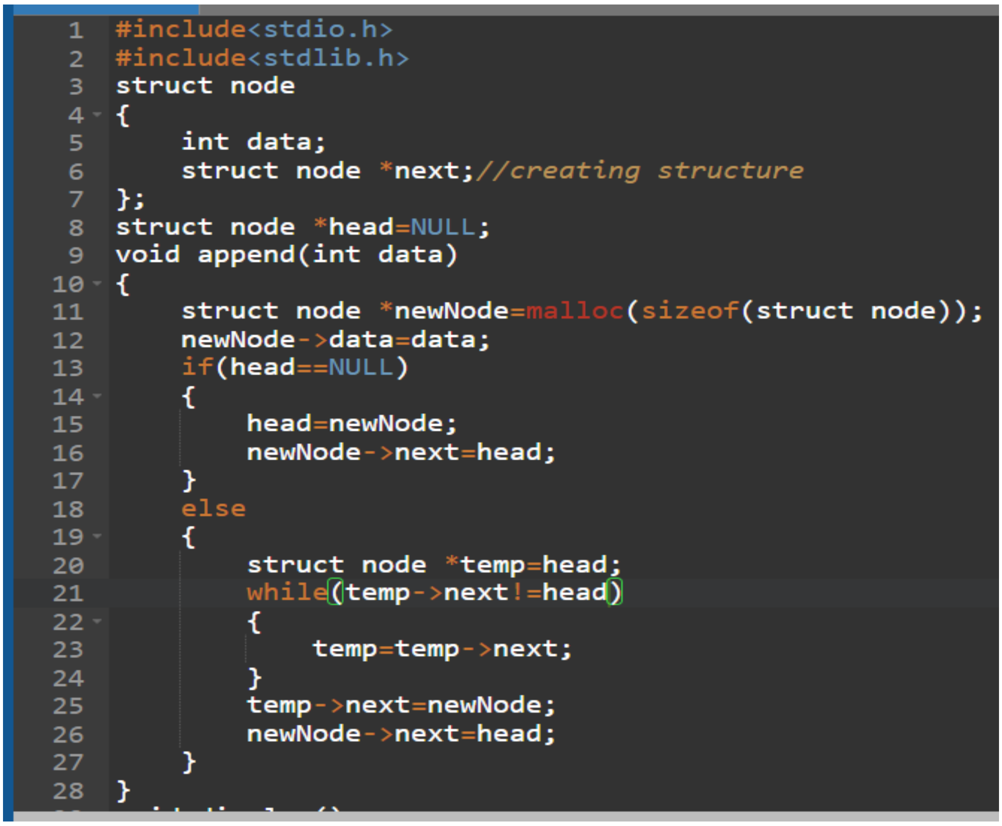
```c
        newNode->prev=temp;
    }
}
void display()
{
    struct node *temp=head;
    while(temp!=NULL)
    {
        printf("%d ",temp->data);//33
        temp=temp->next;
    }
}
void backward()
{
    struct node *temp=head;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
while(temp!=NULL)
{
    printf(" %d",temp->data);
    temp=temp->prev;
}
}
int main()
{
    int n,data;
    printf("Enter number of node:");
```



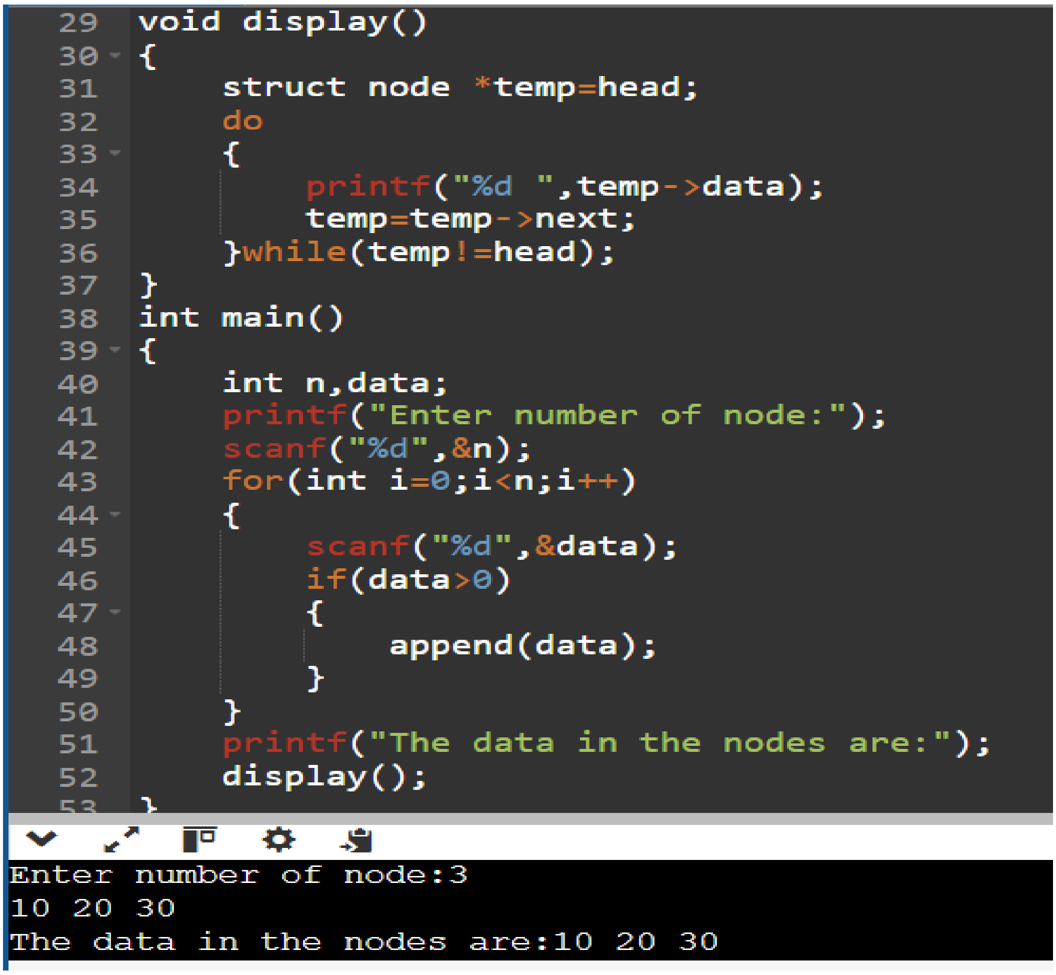
```c
}
int main()
{
    int n,data;
    printf("Enter number of node:");
    scanf("%d",&n);//5
    for(int i=0;i<n;i++)
    {
        scanf("%d",&data);//10
        if(data>0)
        {
            append(data);
        }
    }
    printf("The data in the nodes are:");
    display();
    printf(" \n");
    backward();
}
```

```
Enter number of node:3
10 20 30
The data in the nodes are:10 20 30
 30 20 10
```

Circular linked list



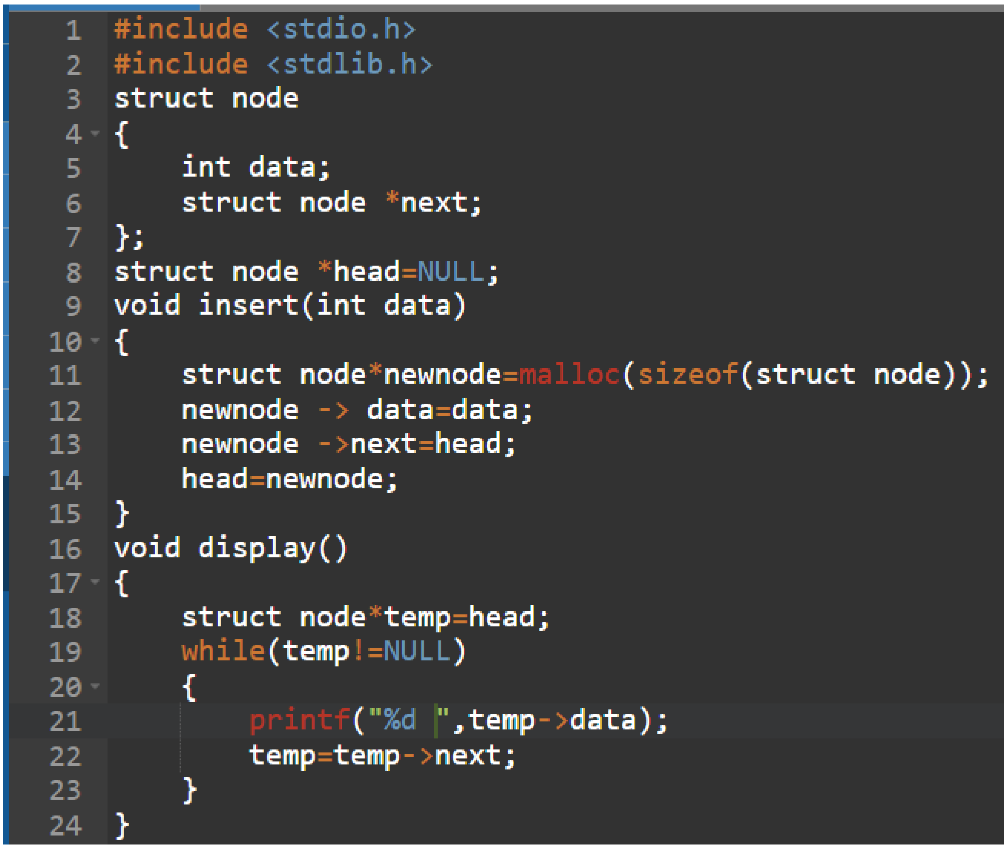
```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
	int data;
	struct node *next;//creating structure
};
struct node *head=NULL;
void append(int data)
{
	struct node *newNode=malloc(sizeof(struct node));
	newNode->data=data;
	if(head==NULL)
	{
		head=newNode;
		newNode->next=head;
	}
	else
	{
		struct node *temp=head;
		while(temp->next!=head)
		{
			temp=temp->next;
		}
		temp->next=newNode;
		newNode->next=head;
	}
}
```



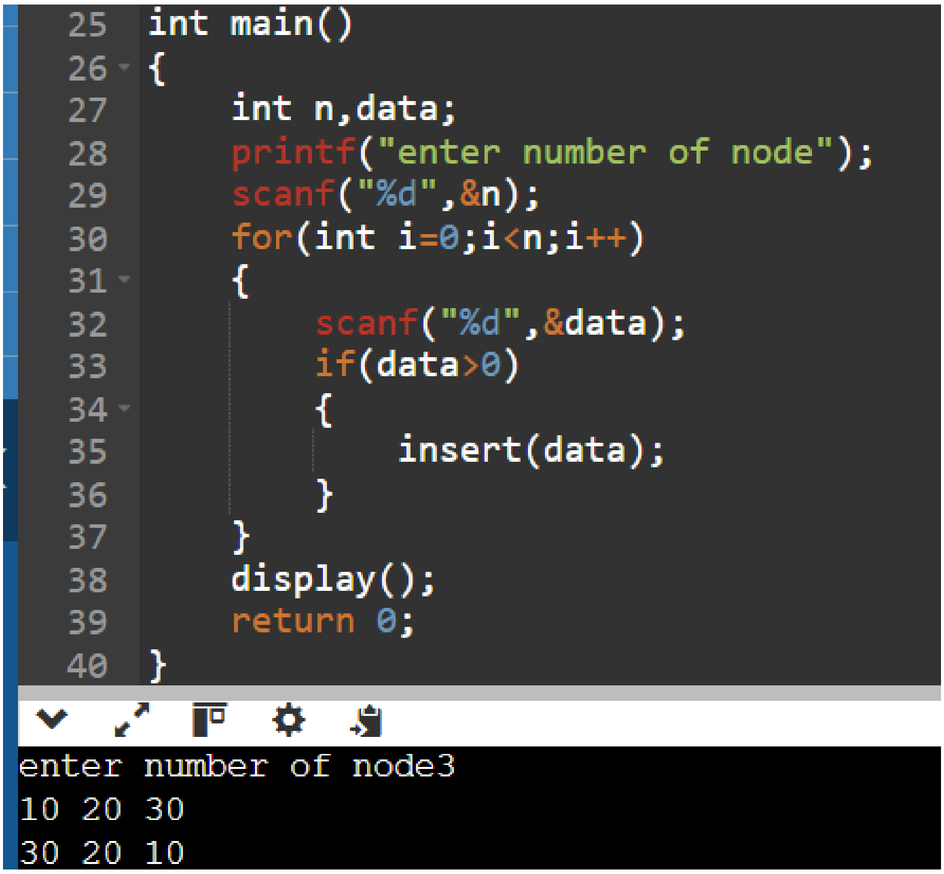
```c
void display()
{
	struct node *temp=head;
	do
	{
		printf("%d ",temp->data);
		temp=temp->next;
	}while(temp!=head);
}
int main()
{
	int n,data;
	printf("Enter number of node:");
	scanf("%d",&n);
	for(int i=0;i<n;i++)
	{
		scanf("%d",&data);
		if(data>0)
		{
			append(data);
		}
	}
	printf("The data in the nodes are:");
	display();
}
```

```
Enter number of node:3
10 20 30
The data in the nodes are:10 20 30
```

Stack implementation, reversing , inserting node at beginning



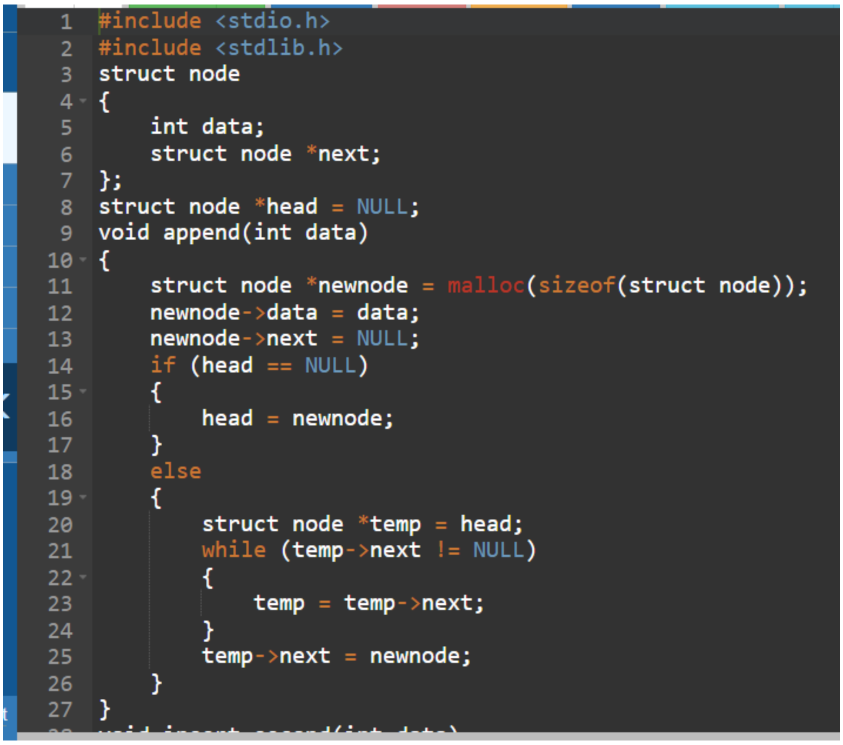
```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;
void insert(int data)
{
    struct node*newnode=malloc(sizeof(struct node));
    newnode -> data=data;
    newnode ->next=head;
    head=newnode;
}
void display()
{
    struct node*temp=head;
    while(temp!=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }
}
```



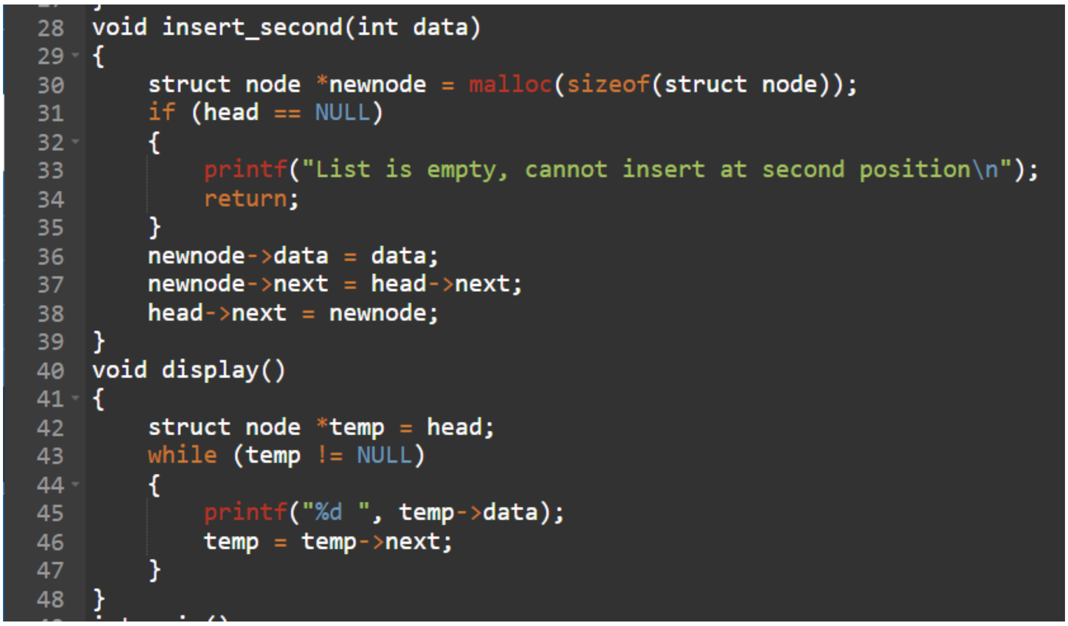
```c
int main()
{
    int n,data;
    printf("enter number of node");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&data);
        if(data>0)
        {
            insert(data);
        }
    }
    display();
    return 0;
}
```

```
enter number of node3
10 20 30
30 20 10
```

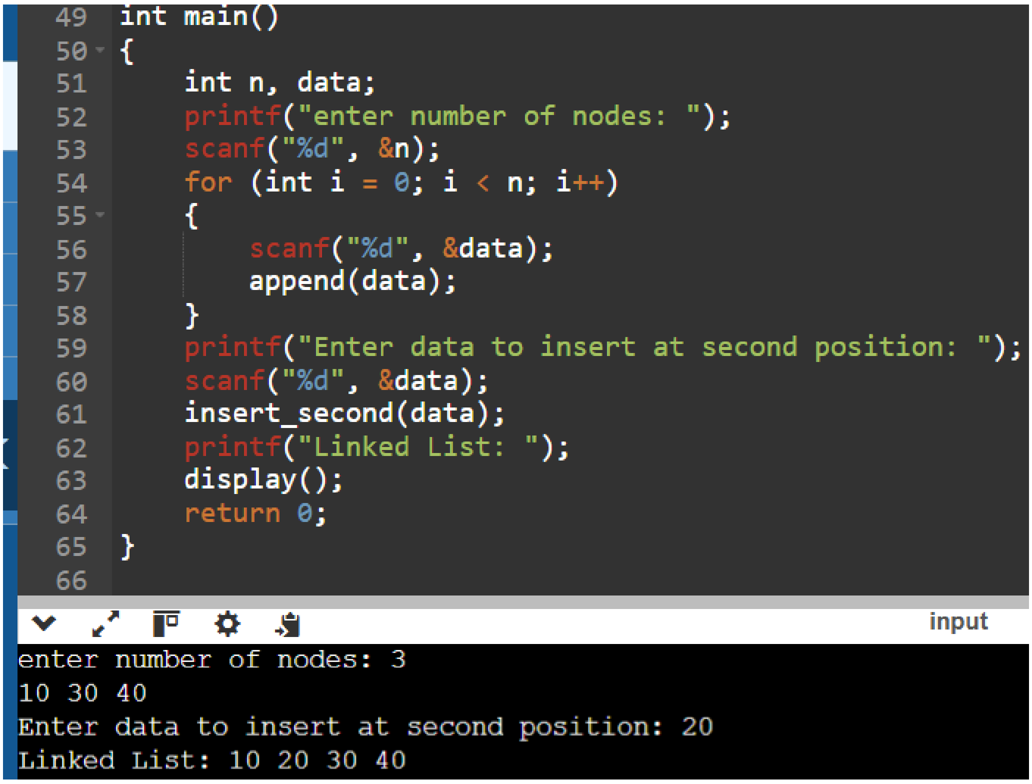
Inserting element at second postion



```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head = NULL;
void append(int data)
{
    struct node *newnode = malloc(sizeof(struct node));
    newnode->data = data;
    newnode->next = NULL;
    if (head == NULL)
    {
        head = newnode;
    }
    else
    {
        struct node *temp = head;
        while (temp->next != NULL)
        {
            temp = temp->next;
        }
        temp->next = newnode;
    }
}
```



```c
void insert_second(int data)
{
    struct node *newnode = malloc(sizeof(struct node));
    if (head == NULL)
    {
        printf("List is empty, cannot insert at second position\n");
        return;
    }
    newnode->data = data;
    newnode->next = head->next;
    head->next = newnode;
}
void display()
{
    struct node *temp = head;
    while (temp != NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}
```



```c
int main()
{
    int n, data;
    printf("enter number of nodes: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &data);
        append(data);
    }
    printf("Enter data to insert at second position: ");
    scanf("%d", &data);
    insert_second(data);
    printf("Linked List: ");
    display();
    return 0;
}

```

```
enter number of nodes: 3
10 30 40
Enter data to insert at second position: 20
Linked List: 10 20 30 40
```