

```

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

struct node
{
    int info;
    struct node *link;
};

struct node *addatend(struct node *last, int data)
{
    struct node *tmp;
    if ((tmp = (struct node *)malloc(sizeof(struct node))) == NULL)
    {
        printf("Not enuf memory\n");
        exit(1);
    }
    tmp->info = data;
    if (last == NULL)
    {
        tmp->link = tmp;
        last = tmp;
        return last;
    }
    else
    {
        tmp->link = last->link;
        last->link = tmp;
        last = tmp;
        return last;
    }
}

struct node *create(struct node *last)
{
    last = NULL;
    printf("Enter the number of items: ");
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
    {
        printf("Enter the data to be inserted: ");
        int data;
        scanf("%d", &data);
        last = addatend(last, data);
    }
    return last;
}

int numnodes(struct node *last)
{
    struct node *p = last->link;
    if (last == NULL)
        return 0;
    int count = 0;
    do
    {
        count++;
        p = p->link;
    } while (p != last->link);
}

```

```

    return count;
}

struct node *deletenode(struct node *last)
{
    struct node *ptr=last, *first=last->link;
    while(ptr->link->link!=first)
    {
        struct node *tmp=ptr->link->link;
        ptr->link->link=tmp->link;
        free(tmp);
        ptr=ptr->link;
        if(ptr->link==first){
            last=ptr;
            return last;
        }
    }

    return last;
}

void display(struct node *last)
{
    struct node *p = last->link;
    if (last == NULL)
    {
        printf("The list is empty\n");
        return;
    }
    printf("The list is: ");
    do
    {
        printf("%3d", p->info);
        p = p->link;
    } while (p != last->link);
    printf("\n");
}

int main(int argc, char const *argv[])
{
    struct node *last = NULL;
    last = create(last);
    printf("The List before deletion: ");
    display(last);
    last = deletenode(last);
    printf("The list after deletion: ");
    display(last);
    return 0;
}

```