

Operations on circular linked list

Circular linked list

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node *link;
};
int main()
{
    int len,i;
    printf("Enter no.of nodes=");
    scanf("%d",&len);
    struct node *head,*ptr,*temp;
    for(i=0;i<len;i++)
    {
        ptr=(struct node*)malloc(sizeof(struct node));
        printf("enter element %d=(i+1)");
        scanf("%d",&ptr->data);
        printf("\n%d=%u\n",ptr->data,&ptr->data);

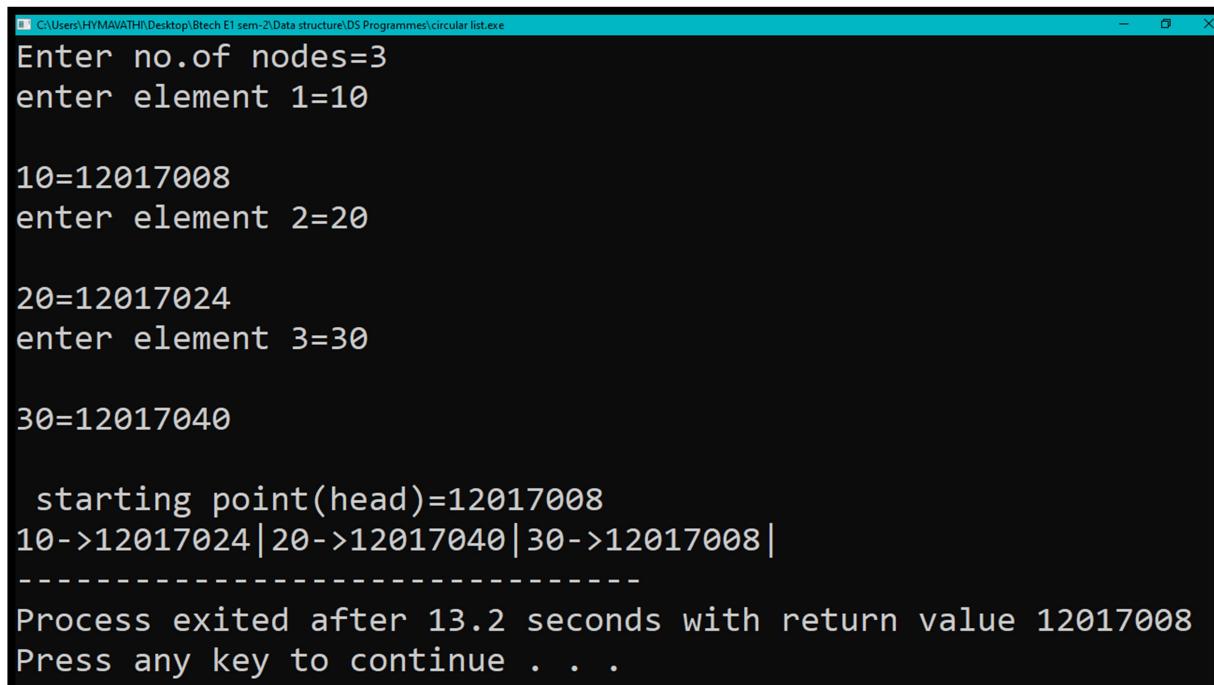
        if(i==0)
        {
            head=temp=ptr;
        }

        else
        {
            temp->link=ptr;
            temp=ptr;
        }

    }
    temp->link=head;
    ptr=head;
    printf("\n starting point(head)=%d\n",head);
```

```
do
{
    printf("%d->",ptr->data);
    printf("%u|",ptr->link);
    ptr=ptr->link;
}
while(ptr!=head);
```

Output



```
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\circular list.exe
Enter no.of nodes=3
enter element 1=10

10=12017008
enter element 2=20

20=12017024
enter element 3=30

30=12017040

starting point(head)=12017008
10->12017024|20->12017040|30->12017008|
-----
Process exited after 13.2 seconds with return value 12017008
Press any key to continue . . .
```

Insertion{circular linked list}

Case-1 : Inserting a node at the starting of a circular linked list

Algorithm:

- ❖ **Step 1** - Create a **newNode(newp)** with given value.
- ❖ **Step 2** – Create a pointer (tail) pointing to head(**tail=head**)
- ❖ **Step 3** - Check **tail->next!=head**.
- ❖ **Step 4** - If it is **True** then, set **tail= tail->next**
- ❖ **Step 5**- If it is **False** then, set **tail->link=newp** and **head=newp**

Program

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

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

int main()
{
    int len,i;
    printf("enter no. of nodes=");
    scanf("%d",&len);
    struct node *head,*temp,*ptr;
    for(i=0;i<len;i++)
```

```
{  
    ptr=(struct node*)malloc(sizeof(struct node));  
    printf("Enter element %d=", (i+1));  
    scanf("%d", &ptr->data);  
    printf("\n%d=%u\n", ptr->data, &ptr->data);  
    if(i==0)  
    {  
        head=temp=ptr;  
    }  
    else  
    {  
        temp->link=ptr;  
        temp=ptr;  
    }  
}  
temp->link=head;  
struct node *tail=head;  
do  
{  
    printf("%d->", tail->data);  
    printf("%u|", tail->link);  
    tail=tail->link;  
}while(tail!=head);  
struct node *newp=(struct node*)malloc(sizeof(struct node));
```

```
printf("\n enter new node=");  
scanf("%d",&newp->data);  
printf("\n%d=%u\n",newp->data,&newp->data);  
tail=head;  
newp->link=head;  
while(tail->link!=head)  
{  
    tail=tail->link;  
}  
tail->link=newp;  
head=newp;  
tail=head;  
printf("\n starting point(head)=%d \n",head);  
do  
{  
    printf("%d->",tail->data);  
    printf("%u|",tail->link);  
    tail=tail->link;  
}while(tail!=head);  
}
```

Output

```
C:\Users\HYMAVATH\Desktop\Btech E1 sem-2\Data structure\DS Programmes\add at first.exe
enter no. of nodes=1
Enter element 1=34

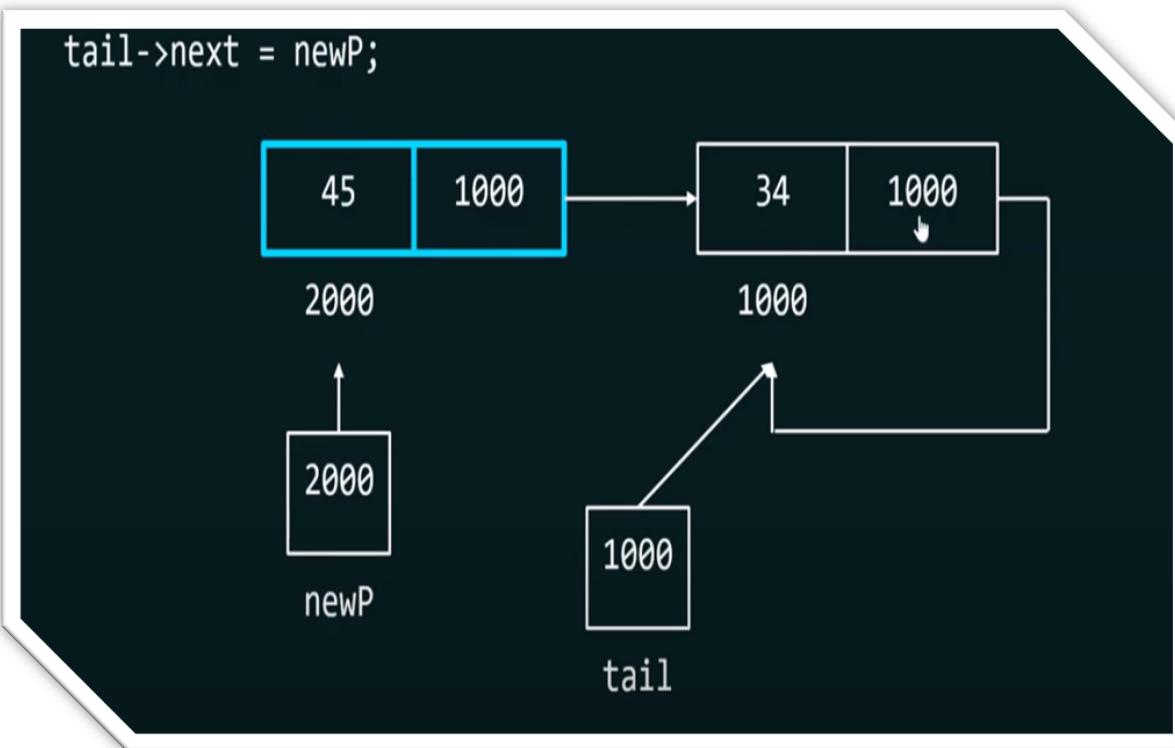
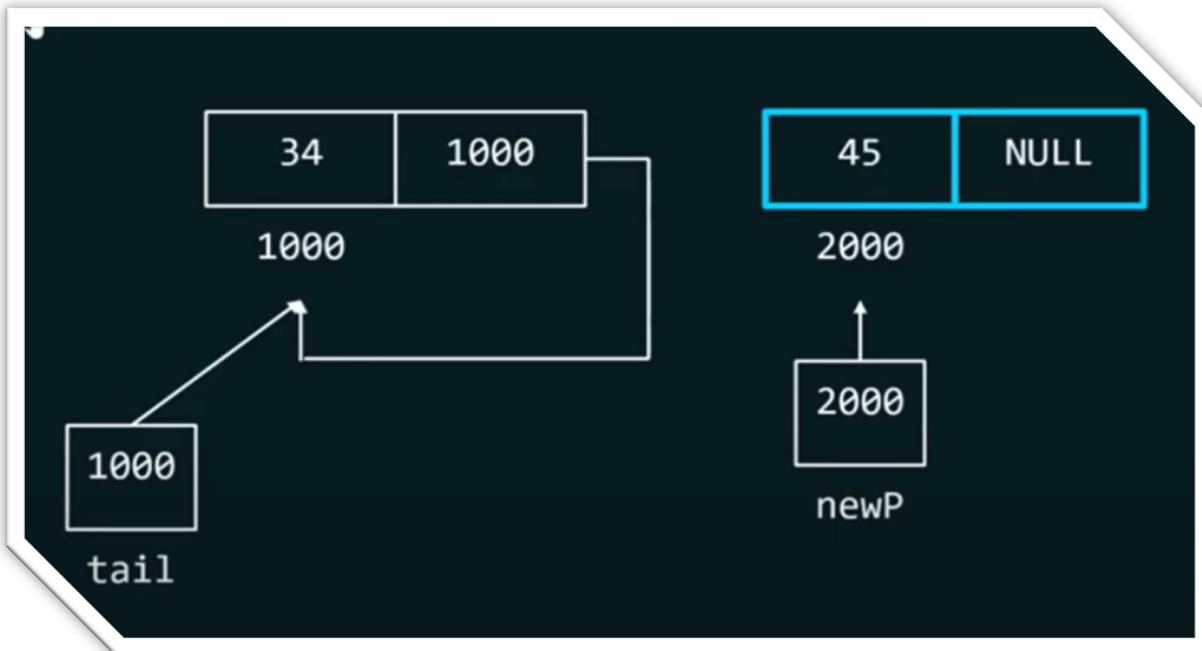
34=7429744
34->7429744|
enter new node=45

45=7429760

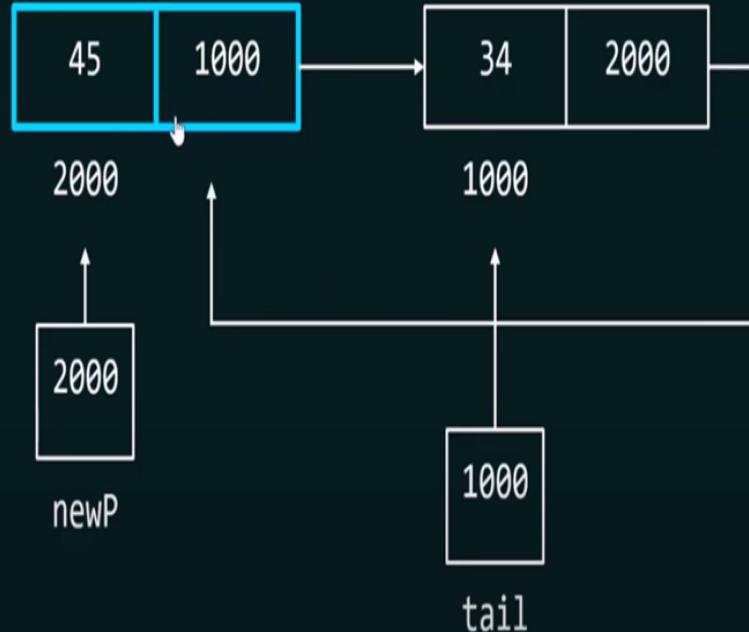
starting point(head)=7429760
45->7429744|34->7429760|
-----
Process exited after 31.83 seconds with return value 7429760
Press any key to continue . . .

Activate Windows
```

Explanation



```
tail->next = newP;
```



Case-2 : Inserting a node at the Ending of a circular linked list

Algorithm:

- ❖ **Step 1** - Create a **newNode(newp)** with given value.
- ❖ **Step 2** - Create a pointer (**tail**) pointing to **head(tail=head)**
- ❖ **Step 3** - Check **tail->next!=head**.
- ❖ **Step 4** - If it is **True** then, set **tail= tail->next**
- ❖ **Step5** - If it is **False** then, set **newp->next=tail->next** and **tail->next=newp**

Program

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

```
#include<stdlib.h>

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

int main()
{
    int len,i;
    printf("enter no. of nodes=");
    scanf("%d",&len);
    struct node *head,*temp,*ptr;
    for(i=0;i<len;i++)
    {
        ptr=(struct node*)malloc(sizeof(struct node));
        printf("Enter element %d=%d,(i+1));
        scanf("%d",&ptr->data);
        printf("\n%d=%d\n",ptr->data,&ptr->data);
        if(i==0)
        {
            head=temp=ptr;
        }
        else
        {
            temp->next=ptr;
            temp=ptr;
        }
    }
}
```

```
        }

    temp->next=head;
    struct node *tail=head;
    do
    {
        printf("%d->",tail->data);
        printf("%u|",tail->next);
        tail=tail->next;
    }while(tail!=head);

    struct node *newp=(struct node*)malloc(sizeof(struct node));
    printf("\n enter new node=");
    scanf("%d",&newp->data);
    tail=head;
    while(tail->next!=head)
    {
        tail=tail->next;
    }
    newp->next=tail->next;
    tail->next=newp;
    tail=head;
    printf("\n starting point(head)=%d \n",head);
    do
    {
        printf("%d->",tail->data);
```

```
    printf("%u|",tail->next);

    tail=tail->next;

}while(tail!=head);

}
```

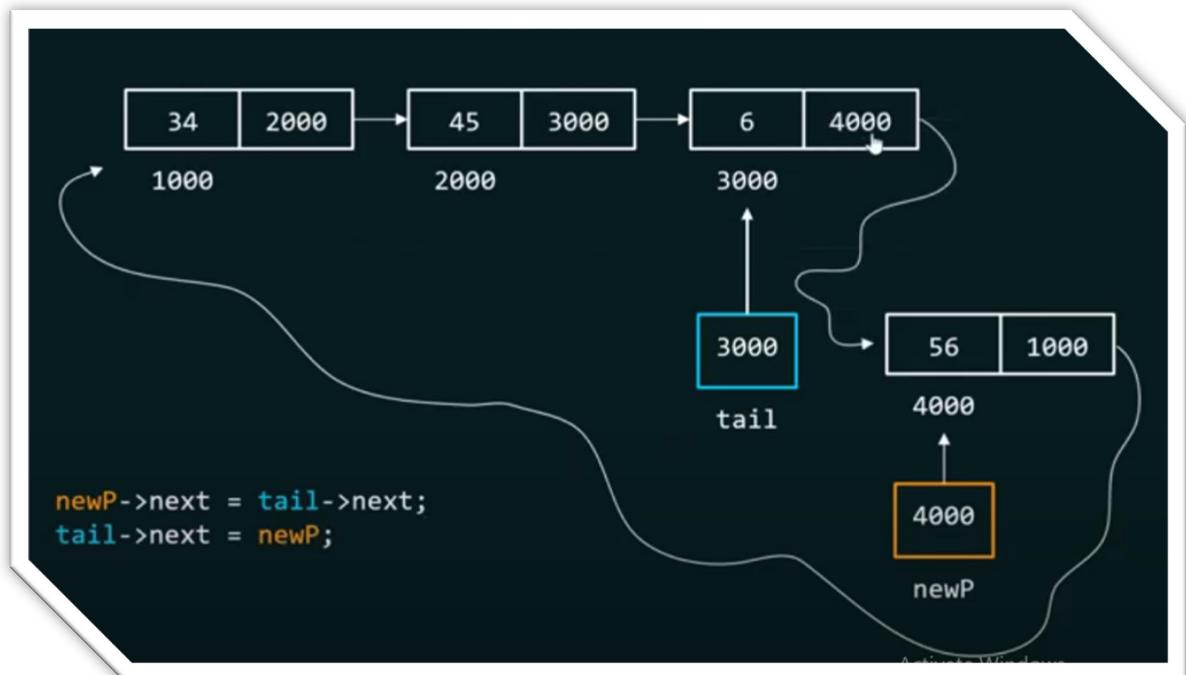
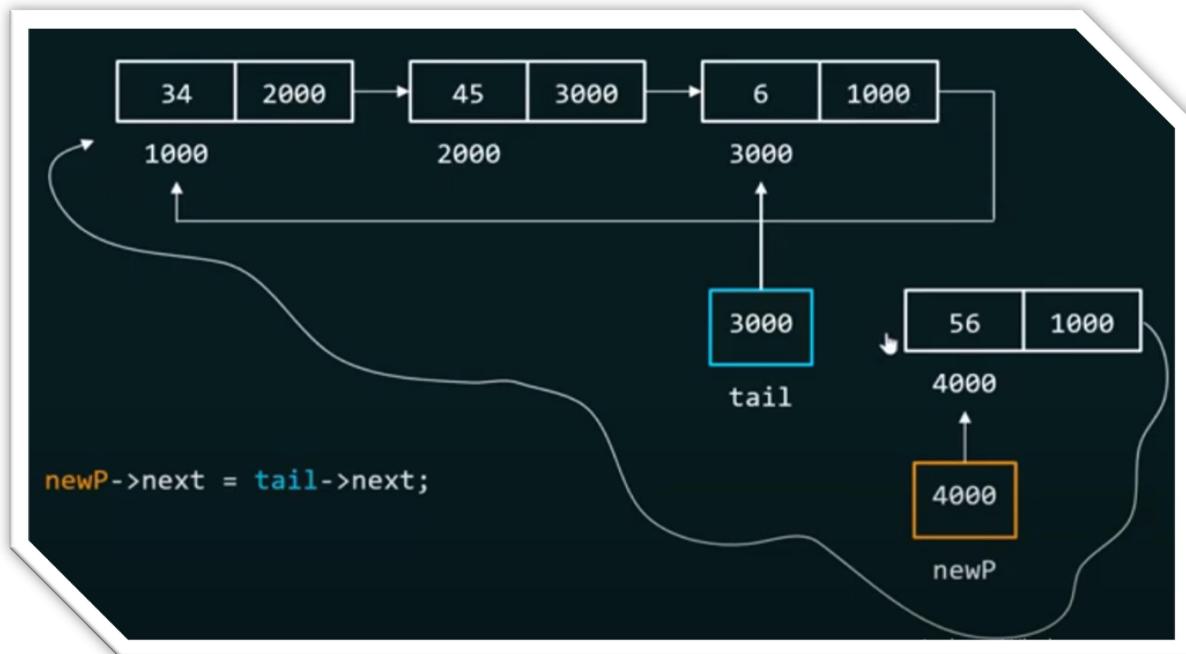
Output

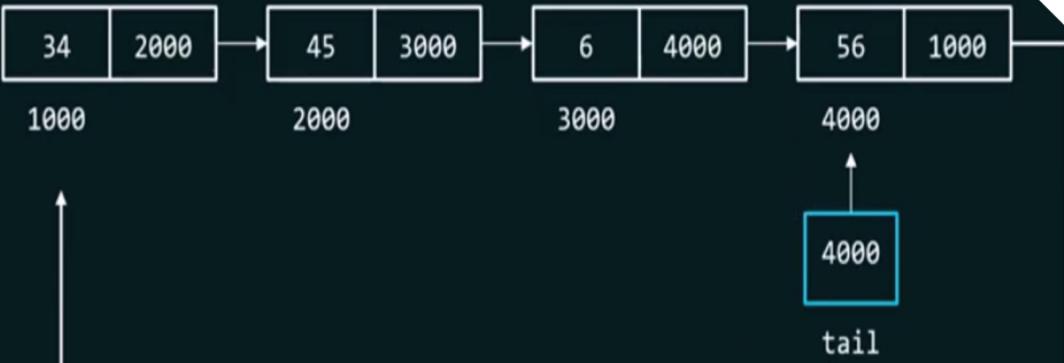
```
C:\Users\HMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\add node at end.exe
enter no. of nodes=3
Enter element 1=11
11=11164720
Enter element 2=22
22=11164736
Enter element 3=33
33=11164752
11->11164736|22->11164752|33->11164720|
enter new node=44

starting point(head)=11164720
11->11164736|22->11164752|33->11164768|44->11164720|
-----
Process exited after 48.02 seconds with return value 11164720
Press any key to continue . . .
```

Activate Windows
Go to Settings to activate Windows.

Explanation





```

newP->next = tail->next;
tail->next = newP; ↓
tail = tail->next;

```

Case-3 : Inserting a node at the Middle of a circular linked list

Algorithm

- ❖ **Step 1** - Create a **newNode(newp)** with given value.
- ❖ **Step 2** – Create two pointer (p,tail) pointing to head(**p=tail=head**)
- ❖ **Step 3** – Ask the user which position(**pos**) they want to add the node.
- ❖ **Step 4** - Check **pos!=2**.
- ❖ **Step 5** - If it is **True** then, set **p=tail** and **tail=tail->next**
- ❖ **Step 6** - Then **pos--**
- ❖ **Step 7** - If it is **False** then, set **newp->next=p->next** and **p->next=newp**

Program

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

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

int main()
{
    int len,i;
    printf("Enter no.of nodes=");
    scanf("%d",&len);
    struct node *head,*pn,*tail;
    for(i=0;i<len;i++)
    {
        pn=(struct node*)malloc(sizeof(struct node));
        printf("enter element %d=%d,(i+1));
        scanf("%d",&pn->data);
        printf("\n%d=%d\n",pn->data,&pn->data);

        if(i==0)
        {
            head=tail=pn;
        }
        else
```

```
{  
    tail->next=pn;  
    tail=pn;  
}  
  
}  
tail->next=head;  
tail=head;  
do  
{  
    printf("%d->",tail->data);  
    printf("%u|",tail->next);  
    tail=tail->next;  
}while(tail!=head);  
  
struct node *newp;  
newp=(struct node*)malloc(sizeof(struct node));  
printf("\nenter a element=");  
scanf("%d",&newp->data);  
printf("\n%d=%u\n",newp->data,&newp->data);  
int pos;  
printf("\nwhich position you want to insert node=");  
scanf("%d",&pos);  
struct node *p;  
p=tail=head;
```

```
while(pos!=1)
{
    p=tail;
    tail=tail->next;
    pos--;
}

newp->next=p->next;
p->next=newp;
tail=head;
printf("starting point(head)=%d\n",tail);
do
{
    printf("%d->",tail->data);
    printf("%u|",tail->next);
    tail=tail->next;
}while(tail!=head);

}
```

Output

```

C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\add at middle.exe
Enter no.of nodes=4
enter element 1=11

11=1334440
enter element 2=22

22=1334456
enter element 3=33

33=1334472
enter element 4=44

44=1334488
11->1334456|22->1334472|33->1334488|44->1334440|
enter a element=55

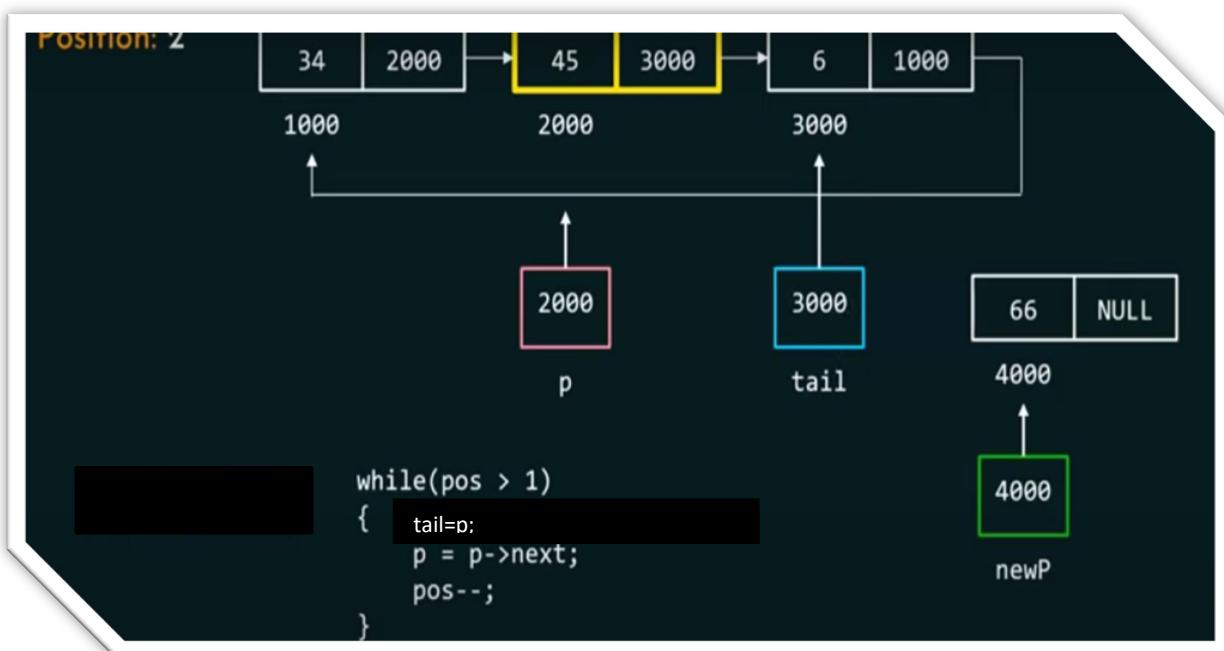
55=1334504

which position you wanrt to insert node=3
starting point(head)=1334440
11->1334456|22->1334504|55->1334472|33->1334488|44->1334440|
-----
Process exited after 14.84 seconds with return value 1334440
Press any key to continue . .

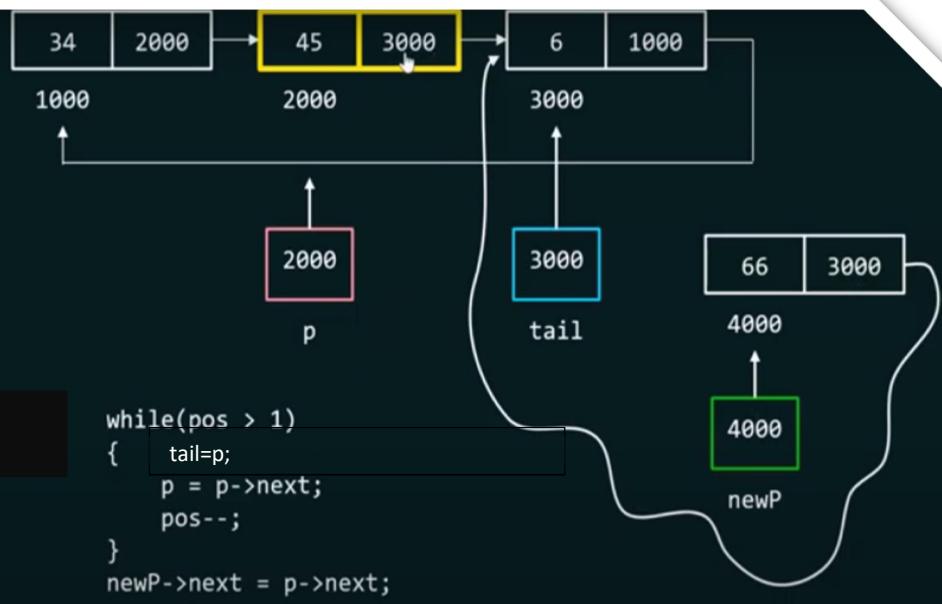
```

Activate Windows
Go to Settings to activate Windows.

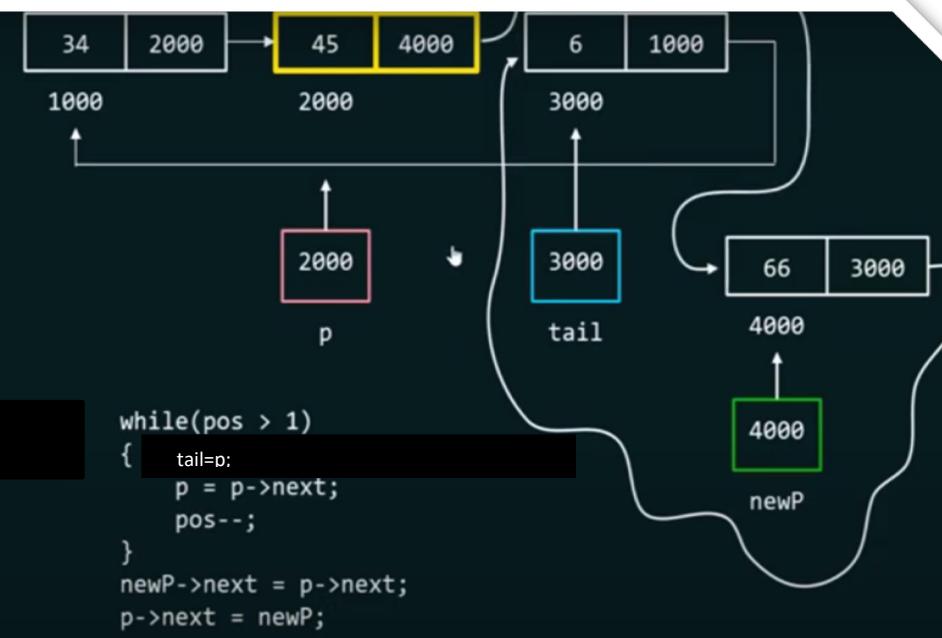
Explanation



Position: 2



Position: 2



Deletion{circular linked list}

Case-1 : Deletion the first node of a circular linked list

Algorithm:

- ❖ **Step 1** - Create two pointers (temp,tail) pointing to head(**temp=tail=head**)
- ❖ **Step 2** - Check **tail->next!=head**.
- ❖ **Step 3** - If it is **True** then, set **tail= tail->next**
- ❖ **Step 4**- If it is **False** then, set **tail->next=temp->next** and **head=temp->next**
- ❖ **Step 5**- Then **free temp**

Program

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

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

int main()
{
    int len,i;
    printf("enter no. of nodes=");
    scanf("%d",&len);
    struct node *head,*t,*ptr;
```

```
for(i=0;i<len;i++)
{
    ptr=(struct node*)malloc(sizeof(struct node));
    printf("Enter element %d=",i+1);
    scanf("%d",&ptr->data);
    printf("\n%d=%u\n",ptr->data,&ptr->data);
    if(i==0)
    {
        head=t=ptr;
    }
    else
    {
        t->next=ptr;
        t=ptr;
    }
}
t->next=head;
t=head;
do
{
    printf("%d->",t->data);
    printf("%u|",t->next);
    t=t->next;
}while(t!=head);
```

```
struct node *temp,*tail;  
temp=tail=head;  
while(tail->next!=head)  
{  
    tail=tail->next;  
}  
tail->next=temp->next;  
head=temp->next;  
free(temp);  
tail=head;  
printf("\n starting pointer(head)=%d \n",tail);  
do  
{  
    printf("%d->",tail->data);  
    printf("%u|",tail->next);  
    tail=tail->next;  
}while(tail!=head);  
}
```

Output

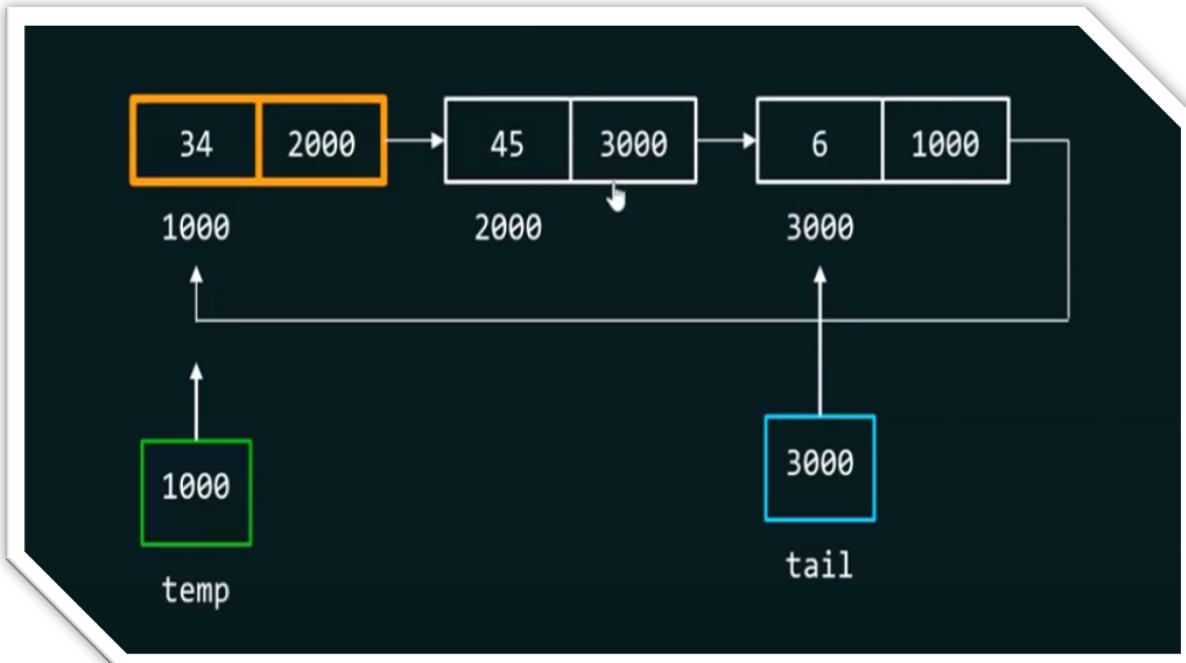
```
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\delete first node cli.exe
enter no. of nodes=3
Enter element 1=11

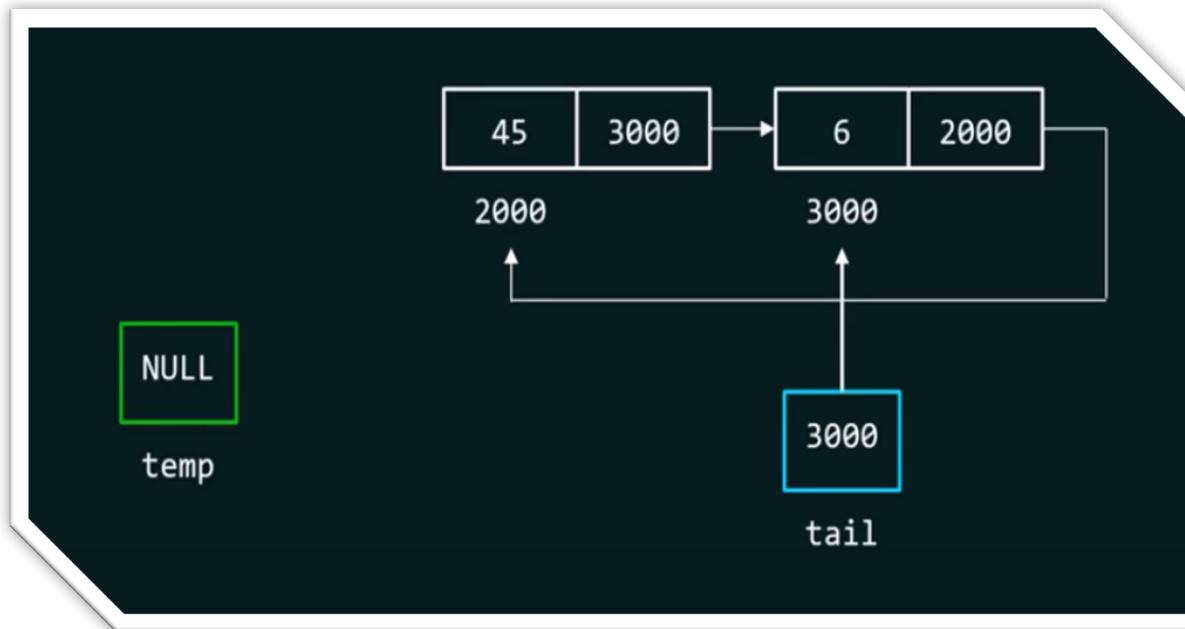
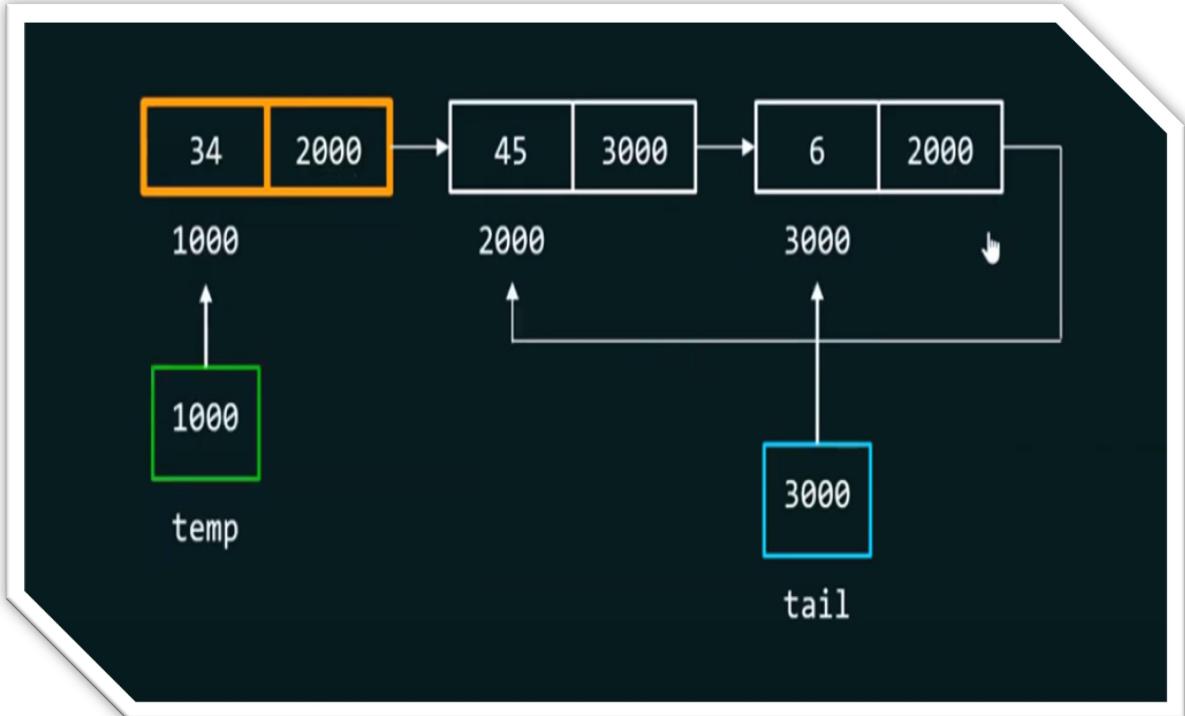
11=12541496
Enter element 2=22

22=12541512
Enter element 3=33

33=12541528
11->12541512|22->12541528|33->12541496|
starting pointer(head)=12541512
22->12541528|33->12541512|
-----
Process exited after 9.533 seconds with return value 12541512
Press any key to continue . . .
```

Explanation





Case-2 : Deletion the last node of a circular linked list

Algorithm:

- ❖ **Step 1** - Create two pointers(temp,tail) pointing to head(**temp=tail=head**)
- ❖ **Step 2** - Check **tail->next!=head**.
- ❖ **Step 3** - If it is **True** then, set **temp=tail** and **tail= tail->next**
- ❖ **Step 4** - If it is **False** then, set **temp->next=tail->next**
- ❖ **Step 5** - Then **free tail**

Program

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

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

int main()
{
    int len,i;
    printf("enter no. of nodes=");
    scanf("%d",&len);
    struct node *head,*t,*ptr;
    for(i=0;i<len;i++)
```

```
{  
    ptr=(struct node*)malloc(sizeof(struct node));  
    printf("Enter element %d=",i+1);  
    scanf("%d",&ptr->data);  
    printf("\n%d=%u\n",ptr->data,&ptr->data);  
    if(i==0)  
    {  
        head=t=ptr;  
    }  
    else  
    {  
        t->next=ptr;  
        t=ptr;  
    }  
}  
t->next=head;  
t=head;  
do  
{  
    printf("%d->",t->data);  
    printf("%u|",t->next);  
    t=t->next;  
}while(t!=head);
```

```
struct node *temp,*tail;  
  
temp=tail=head;  
  
while(tail->next!=head)  
{  
    temp=tail;  
  
    tail=tail->next;  
  
}  
  
temp->next=tail->next;  
  
free(tail);  
  
temp=head;  
  
printf("\n starting pointer(head)=%d \n",temp);  
  
do  
{  
    printf("%d->",temp->data);  
  
    printf("%u|",temp->next);  
  
    temp=temp->next;  
  
}while(temp!=head);  
}
```

Output

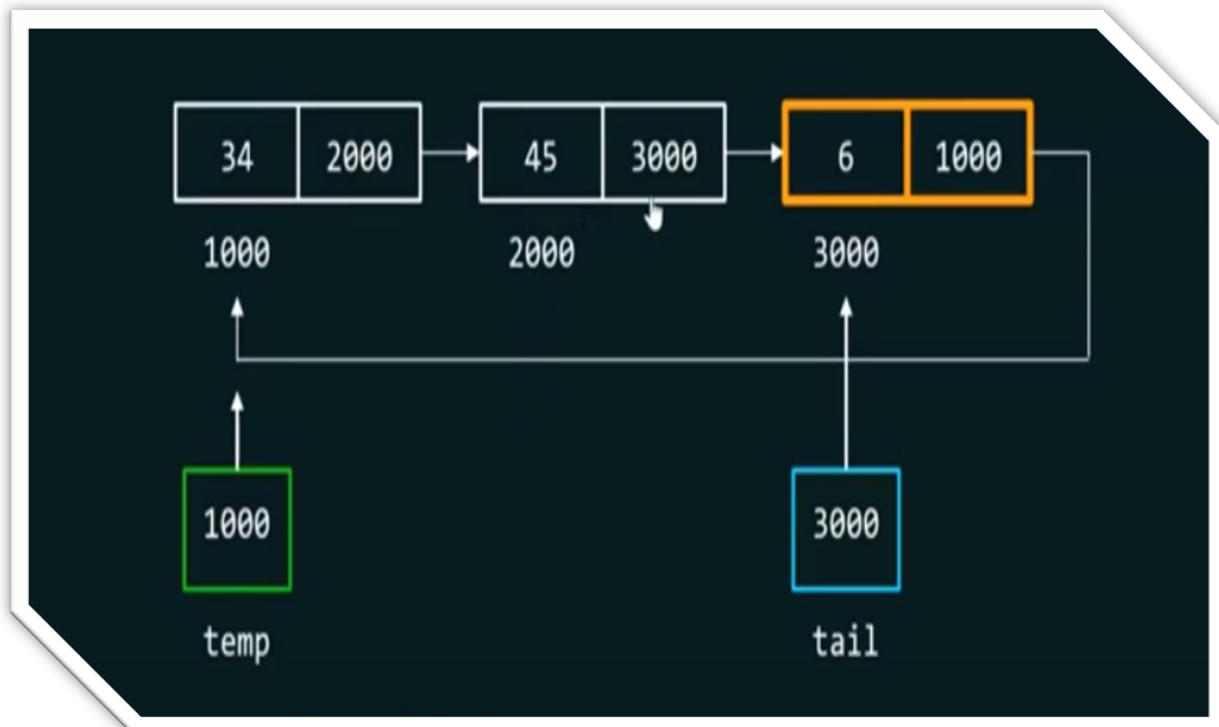
```
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\delete last node cli.exe
enter no. of nodes=3
Enter element 1=11

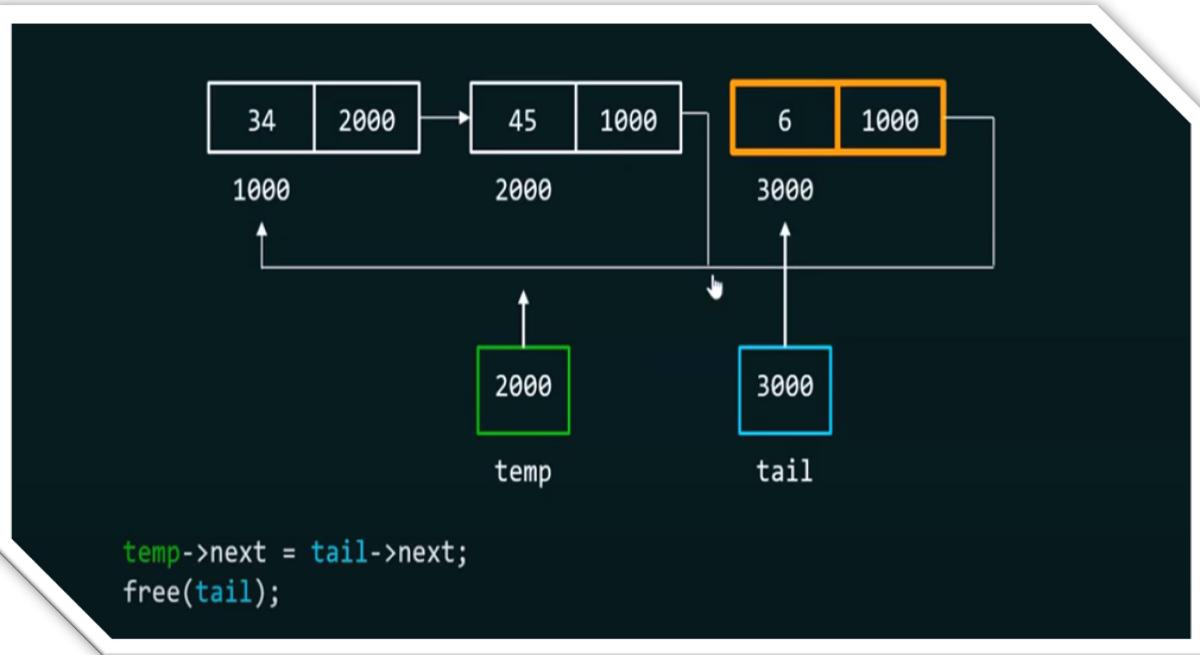
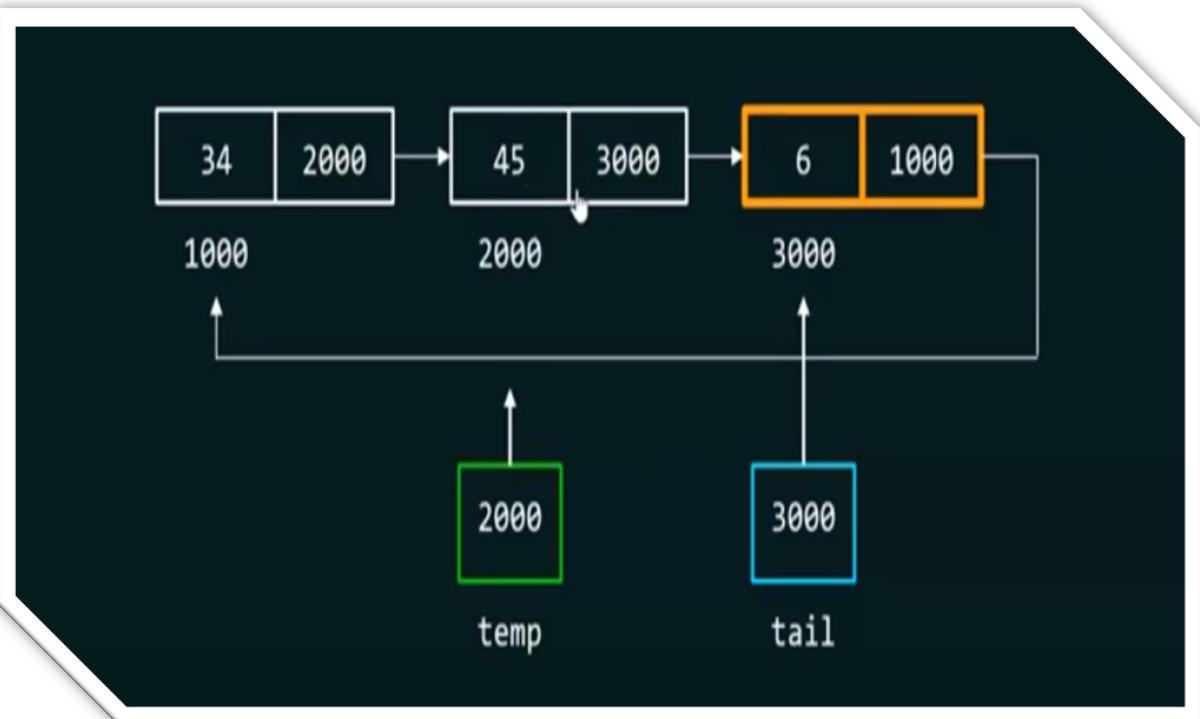
11=10378792
Enter element 2=22

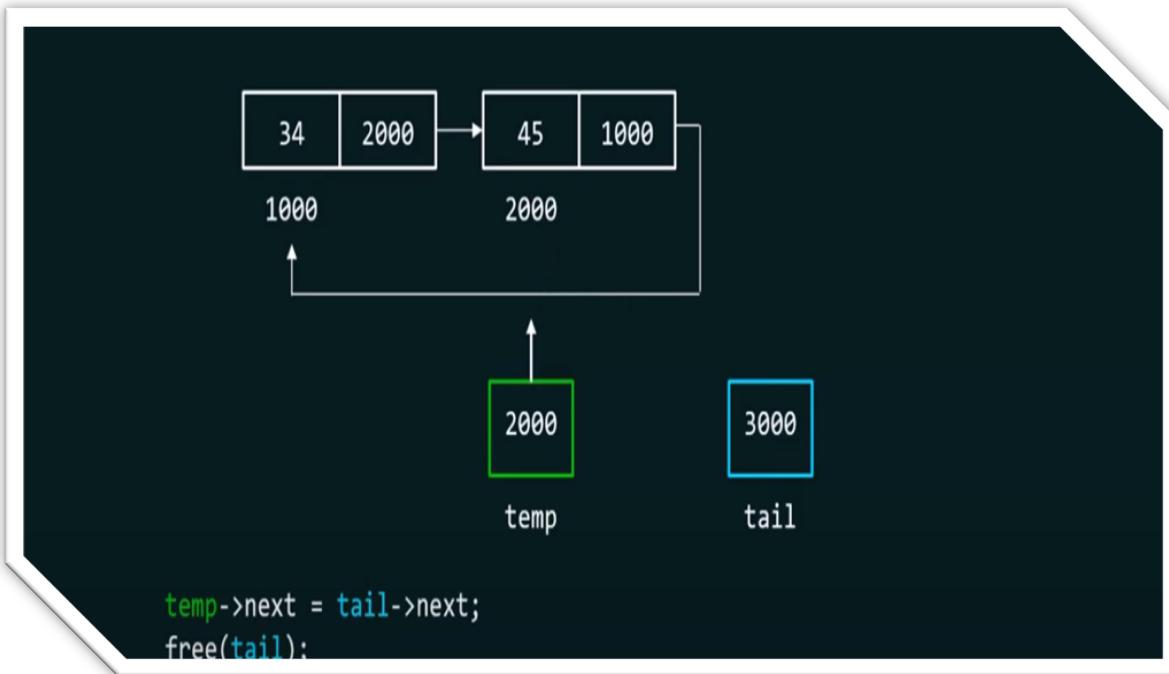
22=10378808
Enter element 3=33

33=10378824
11->10378808|22->10378824|33->10378792|
starting pointer(head)=10378792
11->10378808|22->10378792|
-----
Process exited after 16.36 seconds with return value 10378792
Press any key to continue . . .
```

Explanation







Case-3 : Deletion the middle node of a circular linked list

Algorithm:

- ❖ **Step 1** - Create two pointers (**temp,temp2**) pointing to head(**temp=temp2=head**)
- ❖ **Step 2** - Ask the user which position(**p**) they want to add the node.
- ❖ **Step 3** - Check **p!=1**.
- ❖ **Step 4** - If it is True then, set **temp=temp2** and **temp2=temp2->link**
- ❖ **Step 5** - Then decrement **p--**
- ❖ **Step 6** - If it is False then, set **temp->link=temp2->link**
- ❖ **Step 7** - Then **free(temp2)**

Program

```

#include<stdio.h>
#include<stdlib.h>
struct node{

```

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

int main()
{
    int len,i;
    printf("enter no. of nodes=");
    scanf("%d",&len);
    struct node *head,*t,*ptr;
    for(i=0;i<len;i++)
    {
        ptr=(struct node*)malloc(sizeof(struct node));
        printf("Enter element %d=%d,(i+1));
        scanf("%d",&ptr->data);
        printf("\n%d=%d\n",ptr->data,&ptr->data);
        if(i==0)
        {
            head=t=ptr;
        }
        else
        {
            t->next=ptr;
            t=ptr;
        }
    }
}
```

```
t->next=head;  
t=head;  
do  
{  
    printf("%d->",t->data);  
    printf("%u|",t->next);  
    t=t->next;  
}while(t!=head);
```

```
int p;  
printf("\n enter the position you want to delete=");  
scanf("%d",&p);  
struct node *temp,*temp2;  
temp=temp2=head;  
while(p!=1)  
{  
    temp=temp2;  
    temp2=temp2->next;  
    p--;  
}  
temp->next=temp2->next;  
free(temp2);  
temp=head;  
printf("\n starting pointer(head)=%d \n",temp);  
do
```

```
{  
    printf("%d->",temp->data);  
    printf("%u|",temp->next);  
    temp=temp->next;  
}while(temp!=head);  
}
```

Output

```
C:\Users\HYMAVATH\Desktop\Btech E1 sem-2\Data structure\DS Programmes\delete middle node cll.exe  
enter no. of nodes=4  
Enter element 1=11  
  
11=220768  
Enter element 2=22  
  
22=220784  
Enter element 3=33  
  
33=220800  
Enter element 4=44  
  
44=220816  
11->220784|22->220800|33->220816|44->220768|  
enter the position you want to delete=3  
  
starting pointer(head)=220768  
11->220784|22->220816|44->220768|  
-----  
Process exited after 12.93 seconds with return value 220768  
Press any key to continue . . .
```

Explanation

