

Operations on Double linked list

Double linked list

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

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

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

```
printf("\n%d=%u\n",pn->data,&pn->data);

if(i==0)

{

    pn->prev=NULL;

    head=temp=ptr=pn;

}

else

{

    temp->next=pn;

    temp=pn;

    temp->prev=ptr;

    ptr=pn;

}

temp->next=NULL;

temp=head;

printf("\n starting pointer(head)=%d \n",head);

while(temp!=NULL)

{
```

```
    printf("%d<",temp->prev);

    printf("%d->",temp->data);

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

    temp=temp->next;

}

}
```

Output

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

11=7560640
enter element 2=22

22=7560664
enter element 3=33

33=7560688

starting pointer(head)=7560640
0<-11->7560664|7560640<-22->7560688|7560664<-33->0|
-----
Process exited after 25.13 seconds with return value 0
Press any key to continue . . .
Activate Windows
Go to Settings to activate Windows.
```

Insertion{Double linked list}

Case-1: Insert a node at the beginning of a double linked list

Algorithm

- ❖ **Step-1:** create a newnode(**temp**) with a value.
- ❖ **Step-2:** set **temp->prev=NULL** and **temp->next=head**
- ❖ **Step-3:** Then set **head->prev=temp**
- ❖ **Step-4:** Finally set **head=temp;**

Program

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

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

int main()
{
```

```
int len,i;

printf("Enter no.of nodes=");

scanf("%d",&len);

struct node *head,*pn,*tmp,*ptr;

for(i=0;i<len;i++)

{

    pn=(struct node*)malloc(sizeof(struct node));

    printf("\n enter element %d",(i+1));

    scanf("%d",&pn->data);

    printf("\n%d=%u\n",pn->data,&pn->data);

    if(i==0)

    {

        pn->prev=NULL;

        head=tmp=ptr=pn;

    }

    else

    {

        tmp->next=pn;

        tmp=pn;

        tmp->prev=ptr;

        ptr=pn;

    }

}
```

```
}

tmp->next=NULL;

tmp=head;

printf("\n starting pointer(head)=%d \n",head);

while(tmp!=NULL)

{

    printf("%d<-",tmp->prev);

    printf("%d->",tmp->data);

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

    tmp=tmp->next;

}

struct node *temp;

temp=(struct node*)malloc(sizeof(struct node));

printf("\n enter element %d=",(i+1));

scanf("%d",&temp->data);

printf("\n%d=%u\n",temp->data,&temp->data);

temp->prev=NULL;

temp->next=head;

head->prev=temp;

head=temp;

printf("\n starting pointer(head)=%d \n",head);

while(temp!=NULL)

{
```

```
    printf("%d<-",temp->prev);
    printf("%d->",temp->data);
    printf("%u|",temp->next);

    temp=temp->next;
}

}
```

Output

```
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\Insert at first dll.exe
Enter no.of nodes=3

enter element 1=11
11=12607208

enter element 2=22
22=12607232

enter element 3=33
33=12607256

starting pointer(head)=12607208
0<-11->12607232|12607208<-22->12607256|12607232<-33->0|
enter element 4=44

44=12607280

starting pointer(head)=12607280
0<-44->12607208|12607280<-11->12607232|12607208<-22->12607256|12607232<-33->0|
-----
Process exited after 11.1 seconds with return value 0
Press any key to continue . . .

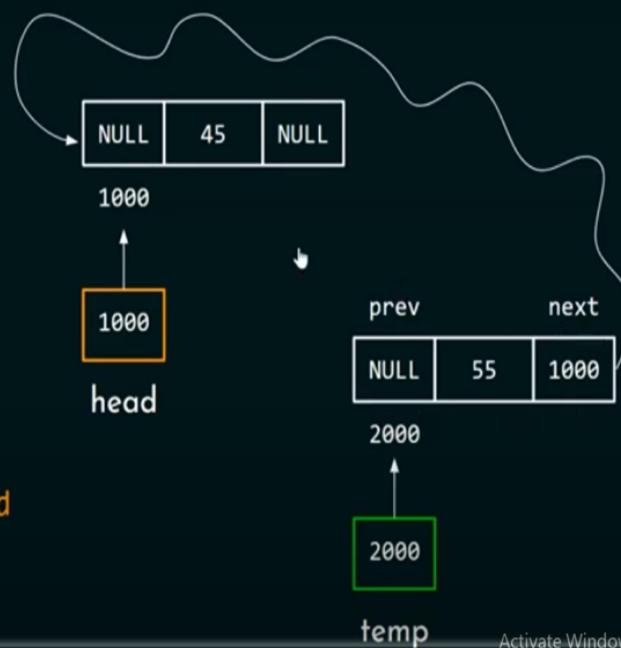
Activate Windows
Go to Settings to activate Windows.
```

Explanation



Activate Windows

`temp->next = head`



Activate Windows

```
temp->next = head  
head->prev = temp;  
head = temp;
```

Activate Windows



1000

1000

head



2000

2000

temp

Activate Windows



1000



2000

2000

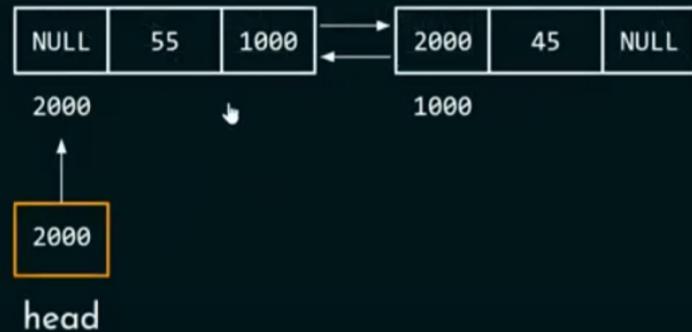
temp

head

Activate Windows

```
temp->next = head  
head->prev = temp;  
head = temp;
```

RESULTANT LIST



Activate Windows

Case-2: Insert a node at the ending of a double linked list

Algorithm

- ❖ **Step-1:** create a newnode(temp) with a value and set **temp->next=NULL**
- ❖ **Step-2:** check **tp->next=head**
- ❖ **Step-3:** If it is **True**,then set **tp = tp->next**
- ❖ **Step-4:** If it is **False**,then set **tp->next=temp** and **temp->prev=tp**

Program

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

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

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

        if(i==0)
        {
            pn->prev=NULL;
```

```
    head=tp=ptr=pn;

}

else
{
    tp->next=pn;
    tp=pn;
    tp->prev=ptr;
    ptr=pn;
}

tp->next=NULL;
tp=head;
printf("\n starting pointer(head)=%d \n",head);
while(tp!=NULL)
{
    printf("%d<-",tp->prev);
    printf("%d->",tp->data);
    printf("%u|",tp->next);
    tp=tp->next;
}
struct node *temp;
temp=(struct node*)malloc(sizeof(struct node));
```

```

printf("\n enter element %d=(i+1));
scanf("%d",&temp->data);
printf("\n%d=%u\n",temp->data,&temp->data);
temp->next=NULL;
tp=head;
while(tp->next!=NULL)
{
    tp=tp->next;
}
tp->next=temp;
temp->prev=tp;
tp=head;
printf("\n starting pointer(head)=%d \n",head);
while(tp!=NULL)
{
    printf("%d<-",tp->prev);
    printf("%d->",tp->data);
    printf("%u|",tp->next);
    tp=tp->next;
}
}

```

Output

```

C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\Insert at end dll.exe
Enter no.of nodes=3

enter element 1=11
11=6643480
enter element 2=22
22=6643504
enter element 3=33
33=6643528

starting pointer(head)=6643480
0<-11->6643504|6643480<-22->6643528|6643504<-33->0|
enter element 4=44

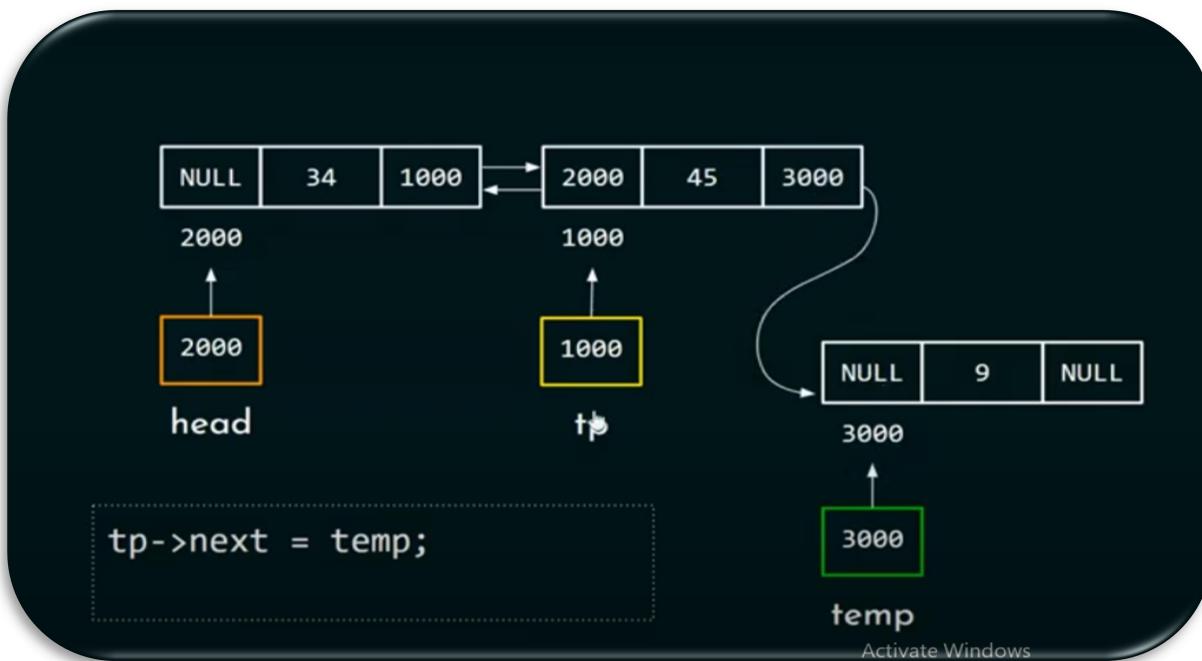
44=6643552

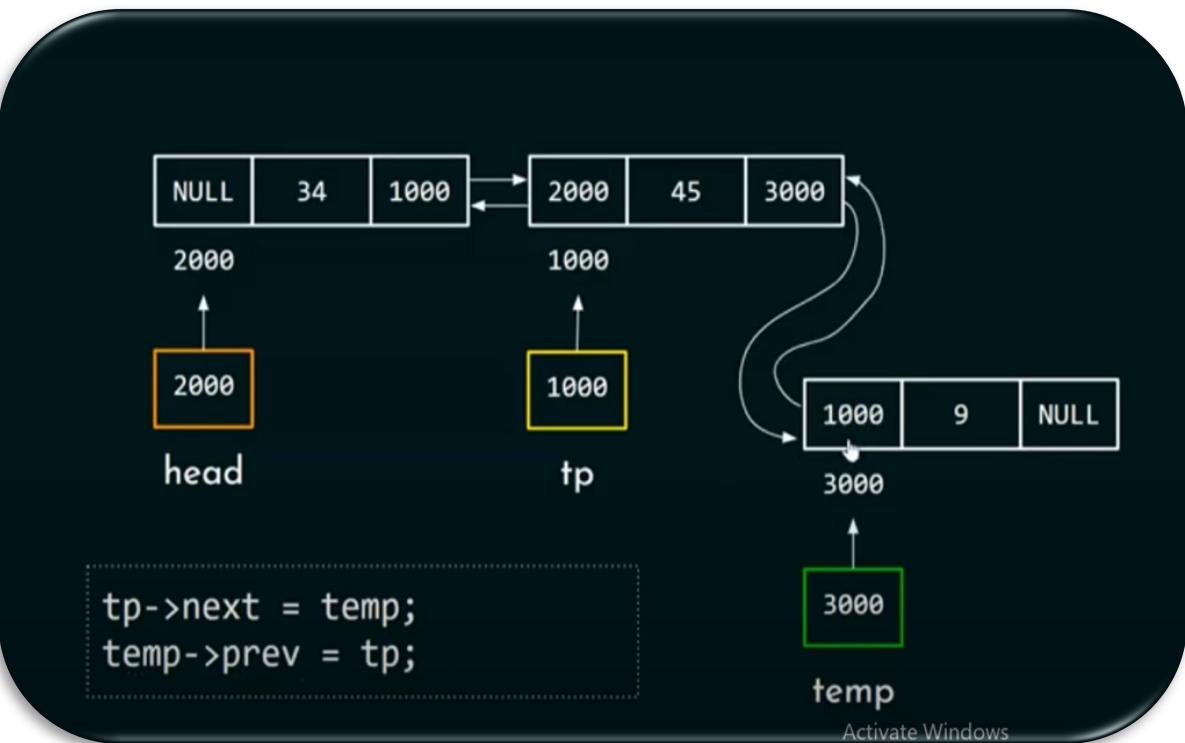
starting pointer(head)=6643480
0<-11->6643504|6643480<-22->6643528|6643504<-33->6643552|6643528<-44->0|
-----
Process exited after 11.55 seconds with return value 0
Press any key to continue . . .

```

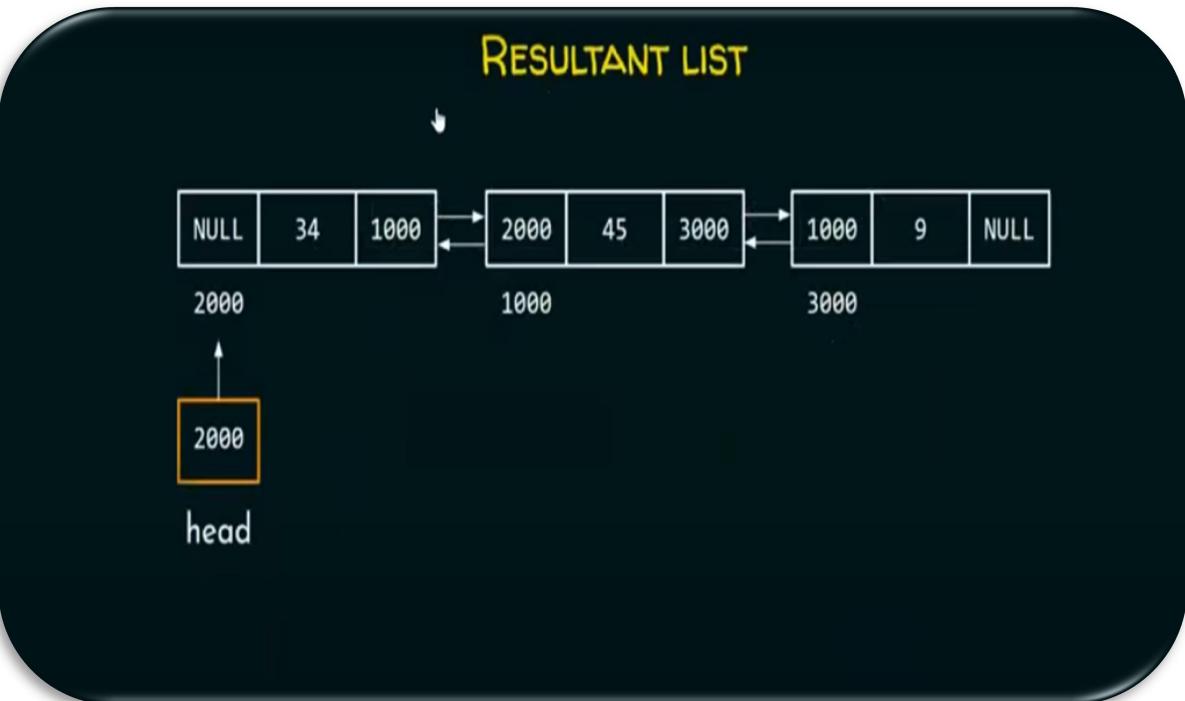
Activate Windows
Go to Settings to activate Windows.

Explanation





RESULTANT LIST



Case-2: Insert a node at the ending of a double linked list

Algorithm

- ❖ **Step-1:** create a newnode(**newp**) with a value.
- ❖ **Step-2:** set **newp->next=newp->prev=NULL**
- ❖ **Step-3:** Create two pointers pointing to head(**temp=temp2=head**)
- ❖ **Step-4:** Ask user the position(**pos**) where they want to insert.
- ❖ **Step-5:** check **pos!=2**
- ❖ **Step-6:** If it is **True**,then set **temp =temp->next** and decrement **pos--**
- ❖ **Step-7:** If it is **False**,then set
 - ✓ **temp2=temp->next;**
 - ✓ **temp->next=newp;**
 - ✓ **temp2->prev=newp;**
 - ✓ **newp->prev=temp;**
 - ✓ **newp->next=temp2;**

Program

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

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

int main()
{
    int len,i;
```

```
printf("Enter no.of nodes=");

scanf("%d",&len);

struct node *head,*pn,*tp,*ptr;

for(i=0;i<len;i++)

{

    pn=(struct node*)malloc(sizeof(struct node));

    printf("\n enter element %d=",(i+1));

    scanf("%d",&pn->data);

    printf("\n%d=%u\n",pn->data,&pn->data);

    if(i==0)

    {

        pn->prev=NULL;

        head=tp=ptr=pn;

    }

    else

    {

        tp->next=pn;

        tp=pn;

        tp->prev=ptr;

        ptr=pn;

    }

}
```

```
}

tp->next=NULL;

tp=head;

printf("\n starting pointer(head)=%d \n",head);

while(tp!=NULL)

{

    printf("%d<-",tp->prev);

    printf("%d->",tp->data);

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

    tp=tp->next;

}

struct node *newp,*temp,*temp2;

newp=(struct node*)malloc(sizeof(struct node));

printf("\n enter element %d=",(i+1));

scanf("%d",&newp->data);

printf("\n%d=%u\n",newp->data,&newp->data);

newp->next=newp->prev=NULL;

int pos;

printf("\n which position you want to insert(pos<no.of nodes)=");

scanf("%d",&pos);

temp=head;

while(pos!=2)

{

    temp=temp->next;
```

```
    pos-;

}

temp2=temp->next;
temp->next=newp;
temp2->prev=newp;
newp->prev=temp;
newp->next=temp2;

tp=head;
printf("\n starting pointer(head)=%d \n",head);
while(tp!=NULL)
{
    printf("%d<-",tp->prev);
    printf("%d->",tp->data);
    printf("%u|",tp->next);
    tp=tp->next;
}
}
```

Output

```

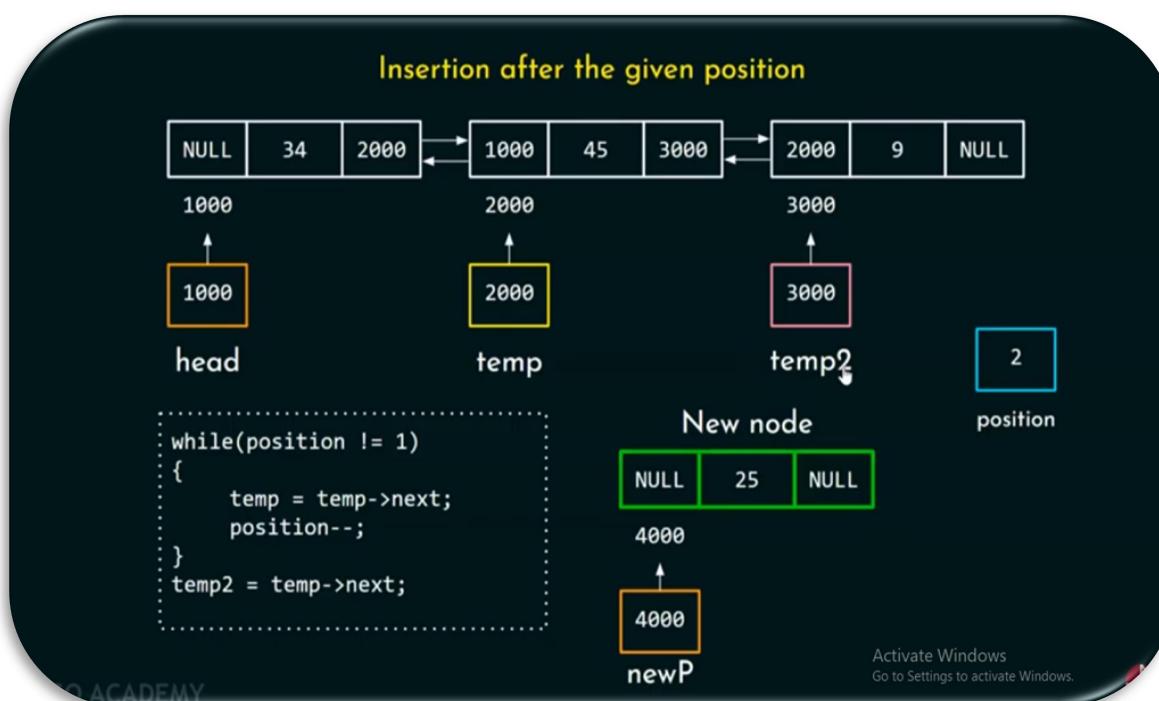
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\add at any pos dll.exe
Enter no.of nodes=3

enter element 1=11
11=7495536
enter element 2=22
22=7495560
enter element 3=33
33=7495584
starting pointer(head)=7495536
0<-11->7495560|7495536<-22->7495584|7495560<-33->0|
enter element 4=44
44=7495608
which position you want to insert(pos<no.of nodes)=2
starting pointer(head)=7495536
0<-11->7495608|7495536<-44->7495560|7495608<-22->7495584|7495560<-33->0|
-----
Process exited after 21.69 seconds with return value 0
Press any key to continue . . .

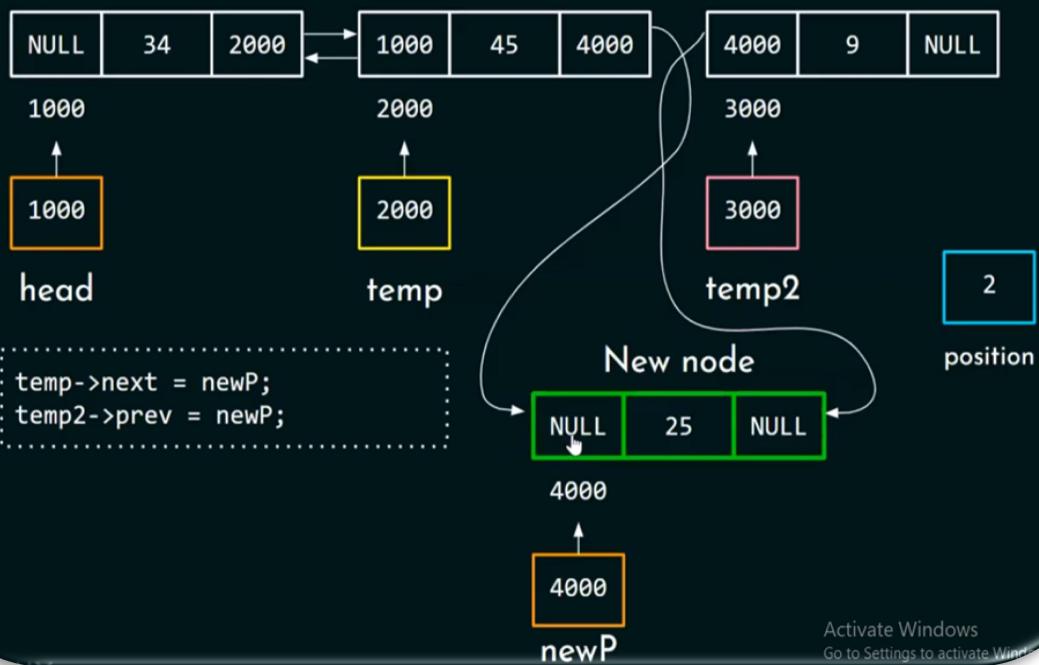
```

Activate Windows
Go to Settings to activate Windows.

Explanation

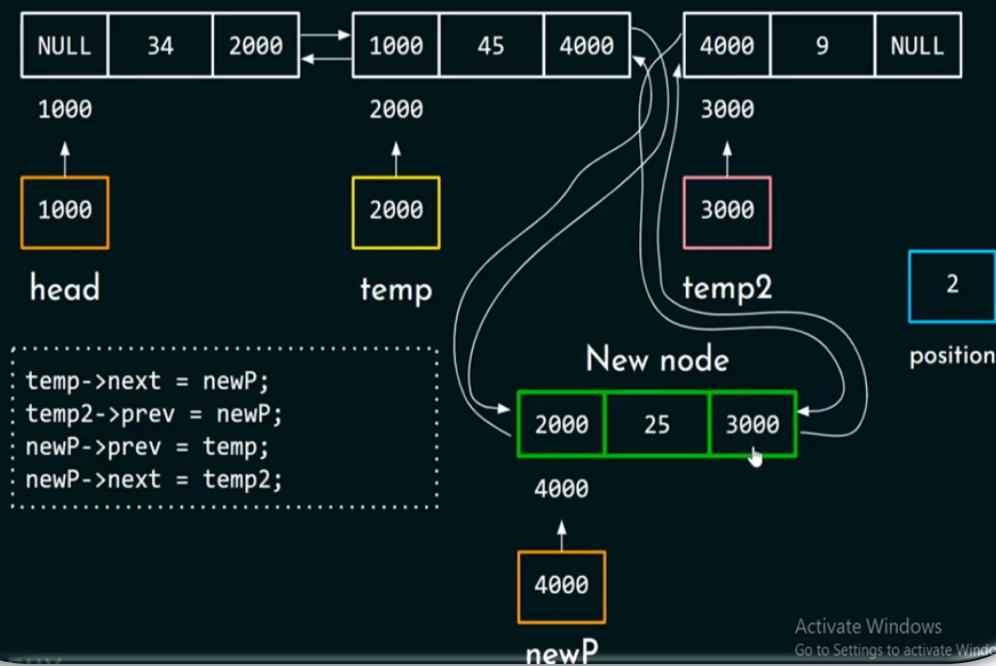


Insertion after the given position



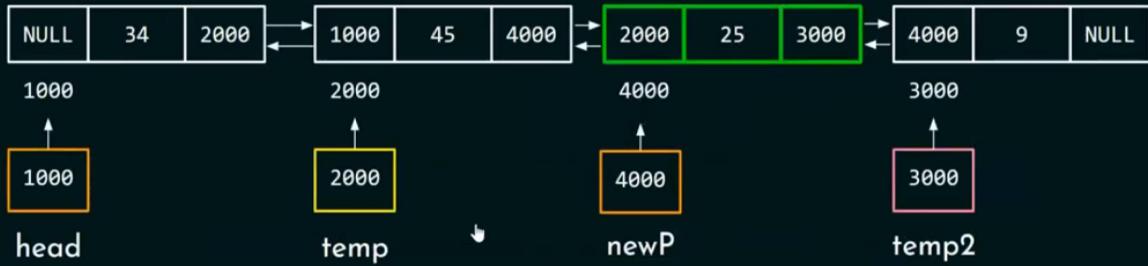
Activate Windows
Go to Settings to activate Windows

Insertion after the given position



Activate Windows
Go to Settings to activate Windows

Insertion after the given position



Deletion{Double linked list}

Case-1: Delete first node of a double linked list

Algorithm

- ❖ **Step-1:** Move **head**=**head->next**
- ❖ **Step-2:** Then **free(head->prev)**
- ❖ **Step-3:** Finally set **head->prev=NULL**

Program

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

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

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

        if(i==0)
        {
            pn->prev=NULL;
```

```
    head=tmp=ptr=pn;

}

else
{
    tmp->next=pn;
    tmp=pn;
    tmp->prev=ptr;
    ptr=pn;
}

}

tmp->next=NULL;
tmp=head;
printf("\n starting pointer(head)=%d \n",head);
while(tmp!=NULL)
{
    printf("%d<-",tmp->prev);
    printf("%d->",tmp->data);
    printf("%u|",tmp->next);
    tmp=tmp->next;
}

head=head->next;
free(head->prev);
```

```

head->prev=NULL;

tmp=head;

printf("\n starting pointer(head)=%d \n",head);

while(tmp!=NULL)

{

    printf("%d<-",tmp->prev);

    printf("%d->",tmp->data);

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

    tmp=tmp->next;

}

}

```

Output

```

C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\remove first node dll.exe
Enter no.of nodes=3

enter element 1=11
11=155248

enter element 2=22
22=155272

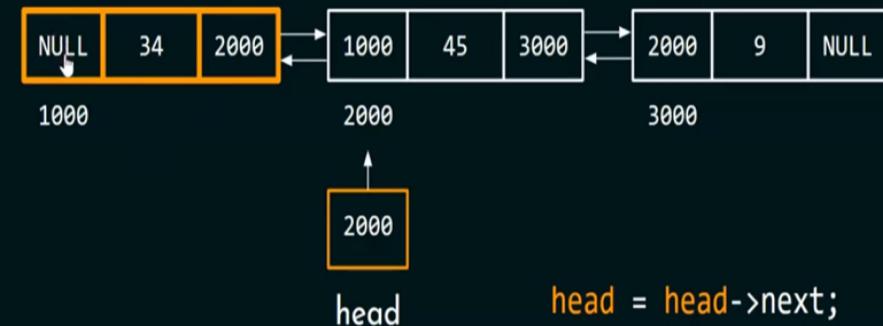
enter element 3=33
33=155296

starting pointer(head)=155248
0<-11->155272|155248<-22->155296|155272<-33->0|
starting pointer(head)=155272
0<-22->155296|155272<-33->0|
-----
Process exited after 6.996 seconds with return value 0
Press any key to continue . . .

```

Activate Windows
Go to Settings to activate Windows.

Explanation





Case-2: Delete last node of a double linked list

Algorithm

- ❖ **Step-1:** Create two pointers pointing to head(**temp=temp2=head**)
- ❖ **Step-2:** check **temp->next!=NULL**
- ❖ **Step-3:** If it is **True**,then set **temp =temp->next**
- ❖ **Step-4:** If it is **False**,then set
 - ✓ **temp2=temp->prev;**
 - ✓ **temp2->next=NULL;**
 - ✓ **free(temp);**
 - ✓ **temp=NULL;**

Program

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

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

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

        if(i==0)
        {
            pn->prev=NULL;
```

```
head=tp=ptr=pn;  
}  
  
else  
{  
    tp->next=pn;  
    tp=pn;  
    tp->prev=ptr;  
    ptr=pn;  
}  
  
}  
tp->next=NULL;  
tp=head;  
printf("\n starting pointer(head)=%d \n",head);  
while(tp!=NULL)  
{  
    printf("%d<-",tp->prev);  
    printf("%d->",tp->data);  
    printf("%u|",tp->next);  
    tp=tp->next;  
}  
struct node *temp,*temp2;  
temp=temp2=head;
```

```
while(temp->next!=NULL)
{
    temp=temp->next;
}

temp2=temp->prev;
temp2->next=NULL;
free(temp);
temp=NULL;

tp=head;
printf("\n starting pointer(head)=%d \n",head);
while(tp!=NULL)
{
    printf("%d<-",tp->prev);
    printf("%d->",tp->data);
    printf("%u|",tp->next);

    tp=tp->next;
}
}
```

Output

```
C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\Data structure\DS Programmes\remove last node dll.exe
Enter no.of nodes=3

enter element 1=11

11=6971056

enter element 2=22

22=6971080

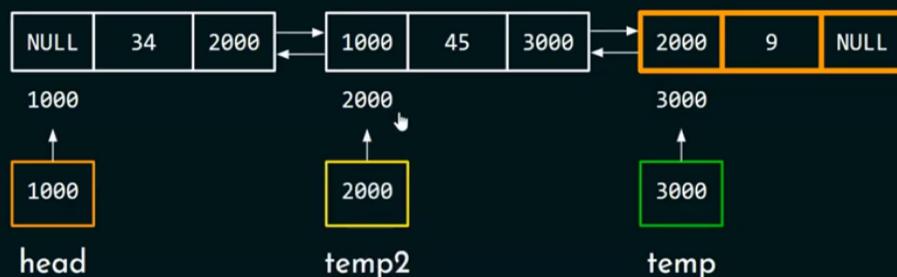
enter element 3=33

33=6971104

starting pointer(head)=6971056
0<-11->6971080|6971056<-22->6971104|6971080<-33->0|
starting pointer(head)=6971056
0<-11->6971080|6971056<-22->0|
-----
Process exited after 13.23 seconds with return value 0
Press any key to continue . . .
```

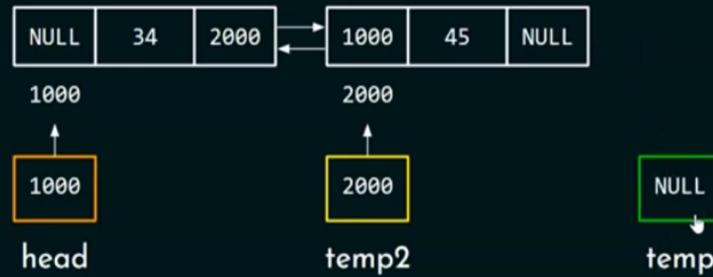
Activate Windows
Go to Settings to activate Windows.

Explanation



```
while(temp->next != NULL)
{
    temp = temp->next;
}
temp2 = temp->prev;
```

Activate Windows
Go to Settings to activate Windows.



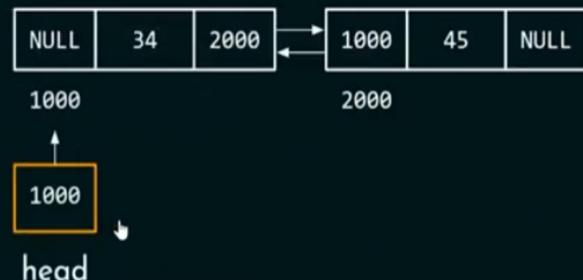
```

while(temp->next != NULL)
{
    temp = temp->next;
}
temp2 = temp->prev;
temp2->next = NULL;
free(temp);
temp = NULL;

```

Activate Windows
Go to Settings to activate Windows.

RESULTANT LINKED LIST



Activate Windows
Go to Settings to activate Windows.

Case-3: Delete middle node of a double linked list

Algorithm

- ❖ **Step-1:** Create two pointers pointing to head(**temp=temp2=head**)
- ❖ **Step-2:** Ask user the position(**pos**) where they want to insert.
- ❖ **Step-3:** check **pos!=2**
- ❖ **Step-4:** If it is **True**,then set **temp =temp->next** and decrement **pos--**
- ❖ **Step-5:** If it is **False**,then set
 - ✓ **temp2=temp->prev;**
 - ✓ **temp2->next=temp->next;**
 - ✓ **temp->next->prev=temp2;**
 - ✓ **free(temp);**
 - ✓ **temp=NULL;**

Program

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

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

int main()
{
    int len,i;
```

```
printf("Enter no.of nodes=");

scanf("%d",&len);

struct node *head,*pn,*tp,*ptr;

for(i=0;i<len;i++)

{

    pn=(struct node*)malloc(sizeof(struct node));

    printf("\n enter element %d=",(i+1));

    scanf("%d",&pn->data);

    printf("\n%d=%u\n",pn->data,&pn->data);

    if(i==0)

    {

        pn->prev=NULL;

        head=tp=ptr=pn;

    }

    else

    {

        tp->next=pn;

        tp=pn;

        tp->prev=ptr;

        ptr=pn;

    }

}
```

```
}

tp->next=NULL;

tp=head;

printf("\n starting pointer(head)=%d \n",head);

while(tp!=NULL)

{

    printf("%d<-",tp->prev);

    printf("%d->",tp->data);

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

    tp=tp->next;

}

struct node *temp,*temp2;

int pos;

printf("\n which position you want to delete=");

scanf("%d",&pos);

temp=temp2=head;

while(pos>1)

{

    temp=temp->next;

    pos--;

}

temp2=temp->prev;

temp2->next=temp->next;

temp->next->prev=temp2;
```

```

        free(temp);

        temp=NULL;

        tp=head;

        printf("\n starting pointer(head)=%d \n",head);

        while(tp!=NULL)

        {

            printf("%d<-",tp->prev);

            printf("%d->",tp->data);

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

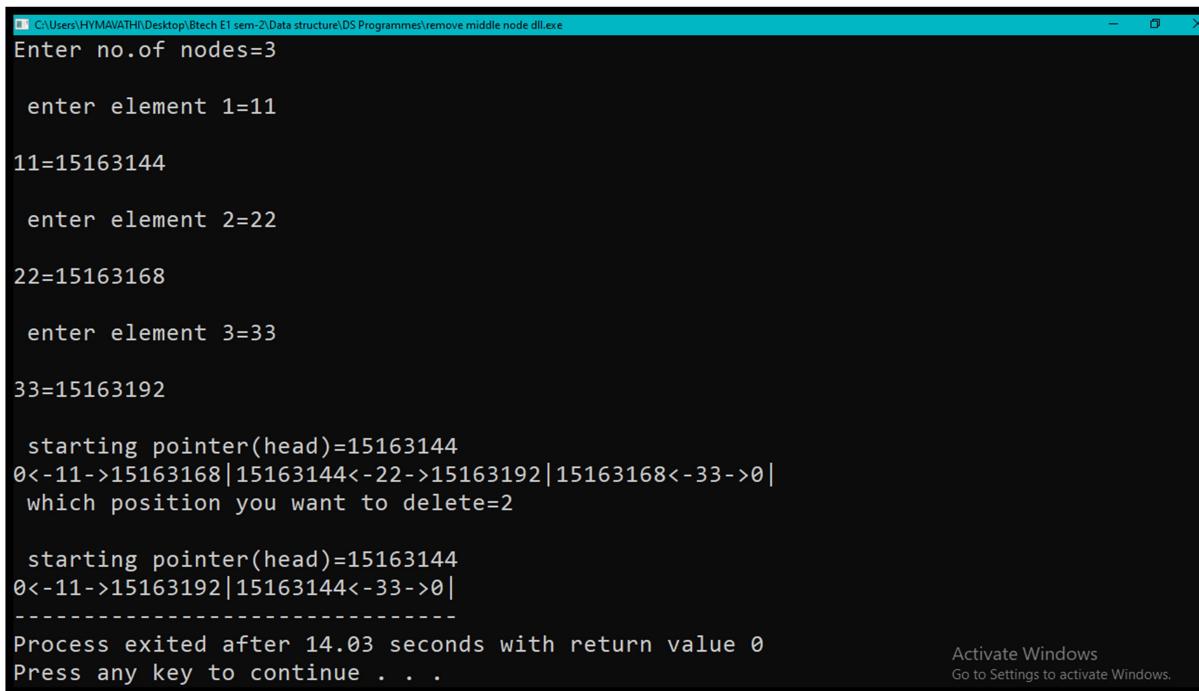
            tp=tp->next;

        }

    }

```

Output



C:\Users\HYMAVATHI\Desktop\Btech E1 sem-2\DS Programmes\remove middle node dll.exe

Enter no.of nodes=3

enter element 1=11

11=15163144

enter element 2=22

22=15163168

enter element 3=33

33=15163192

starting pointer(head)=15163144
0<-11->15163168|15163144<-22->15163192|15163168<-33->0|
which position you want to delete=2

starting pointer(head)=15163144
0<-11->15163192|15163144<-33->0|

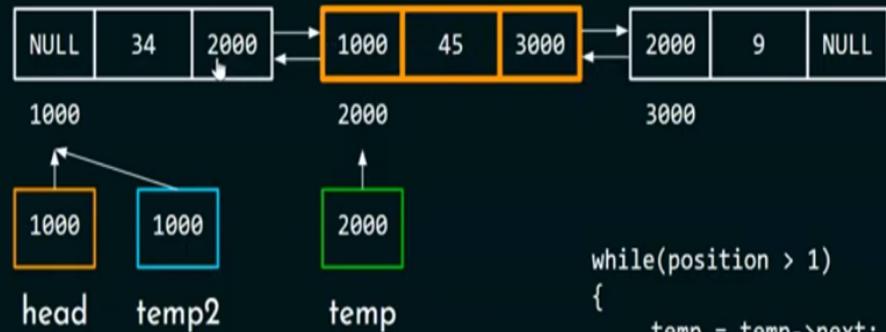
Process exited after 14.03 seconds with return value 0

Press any key to continue . . .

Activate Windows
Go to Settings to activate Windows.

Explanation

position = 2



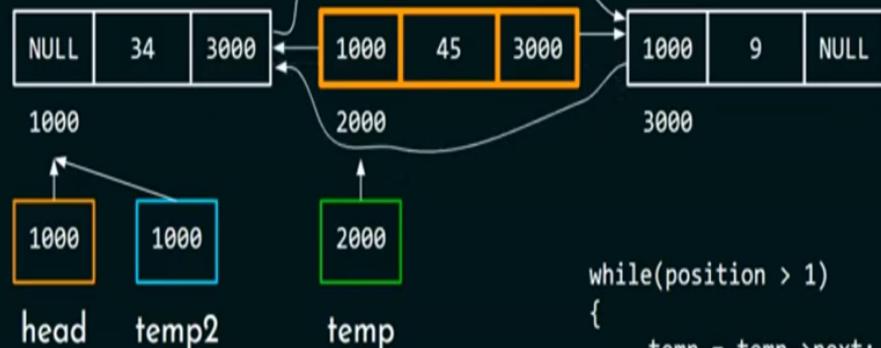
```
while(position > 1)
{
    temp = temp->next;
    position--;
}
temp2 = temp->prev;
```

position = 2



```
while(position > 1)
{
    temp = temp->next;
    position--;
}
temp2 = temp->prev;
temp2->next = temp->next;
```

position = 2



```
while(position > 1)
{
    temp = temp->next;
    position--;
}
temp2 = temp->prev;
temp2->next = temp->next;
temp->next->prev = temp2;
```

position = 2



```
while(position > 1)
{
    temp = temp->next;
    position--;
}
temp2 = temp->prev;
temp2->next = temp->next;
temp->next->prev = temp2;
free(temp);
temp = NULL;
```

Activate Windows
Go to Settings to activate Windows.

RESULTANT LINKED LIST

