

P-22 : Delete the last element in circular Linked list.

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

```
#include<conio.h>
```

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

```
void delete_last();
```

```
void insert_end();
```

```
void display();
```

```
struct node {
```

```
int data;
```

```
struct node *next;
```

```
}*start=NULL;
```

```
void main() {
```

```
int ch; clrscr();
```

```
while(1)
```

```
{
```

```
printf("\n ***CIRCULAR LINKLIST MENU***");  
printf("\n\n1.insert_end\n2. delete last \n 3.Display\n4.exit");  
printf("\n\n enter your choice ");  
scanf("%d",&ch);  
  
    switch(ch)  
{  
case 1:insert_end();  
break;  
case 2:delete_last();  
break;  
case 3:display();  
break;  
    case 4: exit(0);  
break;  
default:printf("\nwrong coice!");  
break;  
} }
```

```
getch();}
```

```
void insert_end() {
```

```
int val;
```

```
struct node *new_node,*ptr;
```

```
new_node=(struct node*)(malloc(sizeof(struct node)));
```

```
printf("Enter an element:");
```

```
scanf("%d",&val);
```

```
new_node->data=val;
```

```
if(start==NULL)    //If list is empty
```

```
{
```

```
    start=new_node;
```

```
}
```

```
else
```

```
{
```

```
    ptr=start;
```

```
    while(ptr->next!=start)
```

```
{
```

```
    ptr=ptr->next;
```

```
}
```

```
    ptr->next=new_node;
```

```
}
```

```
new_node->next=start;
```

```
}
```

```
void display()
```

```
{
```

```
    struct node *ptr;
```

```
    ptr=start;
```

```
    while(ptr->next!=start)
```

```
{
```

```
        printf("\nelement is %d",ptr->data);
```

```
        ptr=ptr->next;
```

```
}
```

```
    printf("\nelement is %d",ptr->data);
```

```
}
```

```
void delete_last()
```

```
{
struct node *ptr, *preptr;
    if(start==NULL)
    {
        printf("\nUNDERFLOW\n");
    }
    else if (start ->next == start)
    {
        start= NULL;
        free(start);
        printf("\nNode Deleted\n");
    }
    else
    {
        ptr = start;
        while(ptr ->next != start)
        {
            preptr=ptr;
            ptr = ptr->next;
        }
    }
}
```

```
}  
preptr->next = ptr -> next;  
free(ptr);  
printf("\nNode Deleted\n");  
}  
}
```