

C circularlinkedlist.c > main()

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  typedef struct node
4  {
5      int data;
6      struct node * next;
7  }ll;
8
9  ll *head=NULL;
10 ll *tail=NULL;
11
12 ll* createnode(int element){
13     ll *nn=(ll*)malloc(sizeof(ll*));
14     if(nn==NULL){
15         printf("memory cannot be allocated");
16     }
17     else{
18         nn->data=element;
19         nn->next=NULL;
20     }
21     return nn;
22 }
23
24 void insertfront(int element){
25     ll *nn=createnode(element);
26     if(head==NULL){
27         head=nn;
28         tail=head;
29         tail->next=head;
30     }
31     else{
32         nn->next=head;
33         head=nn;
34         tail->next=head;
35     }
36 }
37
```

```
37
38 void insertfromend(int element){
39     ll* nn=createnode(element);
40     if(head==NULL){
41         head=nn;
42         tail=head;
43         tail->next=head;
44     }
45     else{
46         tail->next=nn;
47         tail=nn;
48         nn->next=head;
49     }
50 }
51 void deletefront(){
52     if(head==NULL){
53         printf("empty list");
54     }
55     else{
56         ll*temp=head;
57         head=head->next;
58         tail->next=head;
59         free(temp);
60         temp=NULL;
61     }
62 }
63
64 void deleteend(){
65     if(head==NULL){
66         printf("empty list");
67     }
68     else{
69         ll*temp=head;
70         ll *prev=NULL;
71         while(temp!=tail){
72             prev=temp;
73             temp=temp->next;
74         }
75     }
```

```

    }
    prev->next=head;
    tail=prev->next;
    free(temp);
    temp=NULL;
}
}

void deleteanypos(int pos){
    if(head==NULL){
        printf("empty list");
    }
    else{
        if(pos==1){
            if(head==tail){
                free(head);
                head=NULL;
                tail=NULL;
            }
            else{
                ll* temp=head;
                head=head->next;
                tail->next=head;
                free(temp);
                temp=NULL;
            }
        }
        else{
            ll* temp=head;
            ll*prev=NULL;
            for(int i=2;i<=pos;i++){
                prev=temp;
                temp=temp->next;
            }
            prev->next=temp->next;
            if(tail==temp){
                tail=prev;
            }
        }
    }
}

```

circularlinkedlist.c > main()

```
111     }
112     free(temp);
113     temp=NULL;
114 }
115 }
116 }
117
118 void insertanypos(int element,int pos){
119     ll* nn=createnode(element);
120     if(head==NULL){
121         head=nn;
122         tail=head;
123         tail->next=head;
124     }
125     else{
126         if(pos==1){
127             nn->next=head;
128             head=nn;
129             tail->next=head;
130         }
131         else {
132             ll*temp=head;
133             for(int i=2;i<=pos-1;i++){
134                 temp=temp->next;
135             }
136             nn->next=temp->next;
137             temp->next=nn;
138             if(temp==tail){
139                 tail=temp->next;
140                 tail->next=head;
141             }
142         }
143     }
144 }
145 void display(){
146     if(head==NULL){
147         printf("empty list");
```

```

7         printf("empty list");
8     }
9     else{
10         ll* temp=head;
11         while(temp->next!=head){
12             printf("%d\n",temp->data);
13             temp=temp->next;
14         }
15         printf("%d\n",temp->data);
16     }
17 }
18 void main(){
19     while(1){
20
21         printf("\nEnter your choice 1. insertfront 2.display 3.insertback 4.deletefront 5. deleteend 6. deleteanypos 7.insertanypos 8.exit");
22         int ch;
23         scanf("%d",&ch);
24         if(ch==1){
25             printf("Enter the element to be inserted");
26             int element;
27             scanf("%d",&element);
28             insertfront(element);
29         }
30         else if(ch==2){
31             display();
32         }
33         else if(ch==3){
34             printf("Enter the element to be inserted");
35             int element;
36             scanf("%d",&element);
37             insertfromend(element);
38         }
39         else if(ch==4){
40             deletefront();
41         }
42         else if(ch==5){

```

```
}  
else if(ch==5){  
    deleteend();  
}  
else if(ch==6){  
    printf("Enter the position of deletion");  
    int pos;  
    scanf("%d",&pos);  
    deleteanypos(pos);  
}  
else if(ch==7){  
    printf("Enter the element to insert");  
    int element;  
    scanf("%d",&element);  
    printf("Enter the position of insertion");  
    int pos;  
    scanf("%d",&pos);  
    insertanypos(element,pos);  
}  
else{  
    exit(0);  
}  
}
```