circularlinkedlist.c > main()

```c
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
    int data;
    struct node * next;
}ll;

ll *head=NULL;
ll *tail=NULL;

ll* createnode(int element){
    ll *nn=(ll*)malloc(sizeof(ll*));
    if(nn==NULL){
        printf("memory cannot be allocated");
    }
    else{
        nn->data=element;
        nn->next=NULL;
    }
    return nn;
}

void insertfront(int element){
    ll *nn=createnode(element);
    if(head==NULL){
        head=nn;
        tail=head;
        tail->next=head;
    }
    else{
        nn->next=head;
        head=nn;
        tail->next=head;
    }
}
```

int element

circularlinkedlist.c > main()

```c
void insertfromend(int element){
    ll* nn=createnode(element);
    if(head==NULL){
        head=nn;
        tail=head;
        tail->next=head;
    }
    else{
        tail->next=nn;
        tail=nn;
        nn->next=head;
    }
}
void deletefront(){
    if(head==NULL){
        printf("empty list");
    }
    else{
        ll*temp=head;
        head=head->next;
        tail->next=head;
        free(temp);
        temp=NULL;
    }
}

void deleteend(){
    if(head==NULL){
        printf("empty list");
    }
    else{
        ll*temp=head;
        ll *prev=NULL;
        while(temp!=tail){
            prev=temp;
            temp=temp->next;
```

```
        }
        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;
```

```
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");
```

```
        printf("empty list");
    }
    else{
        ll* temp=head;
        while(temp->next!=head){
            printf("%d\n",temp->data);
            temp=temp->next;
        }
        printf("%d\n",temp->data);
    }
}
void main(){
    while(1){

    printf("Enter your choice 1. insertfront 2.display 3.insertback 4.deletefront 5. deleteend 6. deleteanypos 7.insertanypos 8.exit");
    int ch;
    scanf("%d",&ch);
    if(ch==1){
        printf("Enter the element to be inserted");
        int element;
        scanf("%d",&element);
        insertfront(element);
    }
    else if(ch==2){
        display();
    }
    else if(ch==3){
        printf("Enter the element to be inserted");
        int element;
        scanf("%d",&element);
        insertfromend(element);
    }
    else if(ch==4){
        deletefront();
    }
    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);
    }
    }
}
```