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

typedef struct cll {
        int data;
        struct cll * link;
}cll;

cll * head = NULL;
int count = 0;

cll * createnode() {
        cll * newnode = (cll*)malloc(sizeof(cll));
        printf("Enter the data element : ");
        scanf("%d",&newnode->data);
        newnode->link = NULL;
        count++;
        return newnode;
}

void insert() {
        if(head == NULL) {
                printf("Creating the very first node.\n");
                head = createnode();
                head->link = head;
        }
        else {
                printf("Enter the posistion at which you want to
insert the node (between 0 and %d) : ",count);
                int pos;
                scanf("%d",&pos);

                if(pos <= count && pos>-1) {
                        cll * trav = head;
                        if(pos == 0) {
                                for(int i = 0 ; i < count-1 ; i++ )
                                        trav = trav->link;
                                cll * temp = createnode();
                                trav->link = temp;
                                temp->link = head;
                                head = temp;
                        }
                        else {
                                cll * temp = createnode();
                                for(int i = 0 ; i < pos-1 ; i++ )
                                        trav = trav->link;
                                temp->link = trav->link;
                                trav->link = temp;
                        }
                }
                else
                        printf("Invalid input.\n");
        }
}

void delete() {
```

```
        if(head == NULL )
                printf("Circular Linked List is empty.\n");
        else {
                printf("Enter the posistion at which you want to
delete the node (between 0 and %d) : " , count-1);
                int pos;
                scanf("%d",&pos);

                if(pos < count && pos>-1) {

                        cll * trav = head;
                        if(pos==0) {
                                for(int i = 0 ; i<count-1 ; i++)
                                        trav = trav->link;
                                printf("Element %d deleted.\n",head-
>data);
                                trav->link = head->link;
                                head = head->link;
                        }
                        else {
                                for(int i = 0 ; i < pos-1 ; i++ )
                                        trav = trav->link;
                                cll * temp = trav->link;
                                printf("Element %d deleted.\n",temp-
>data);
                                trav->link = temp->link;
                                temp->link = NULL;
                                temp = NULL;
                        }
                        count--;
                        if(count == 0)
                                head = NULL;
                }
                else
                        printf("Invalid Input.\n");
        }
}

void display() {
        if(head == NULL)
                printf("Circular Linked List is empty.\n");
        else {
                printf("The elements of Circular Linked List are : ");
                printf("%d ",head->data);
                cll * strt = head->link;
                while(strt != head) {
                        printf("%d ", strt->data);
                        strt= strt->link;
                }
                printf("\n");
        }
}

void main() {
        int choice , flag = 1 ;
        printf("Enter the operation to be performed.\n");
        while(flag) {
```

```
                printf("1.Insert \t 2.Delete \t 3.Display \t 4.Exit  :
");
                scanf("%d" , &choice);
                switch(choice) {
                        case 1 : insert();
                                 break;

                        case 2 : delete();
                                 break;

                        case 3 : display();
                                 break;

                        case 4 : flag = 0 ;
                                 break;

                        default : printf("Invalid input.\n");
                                  break;
                }
        }
}
```