

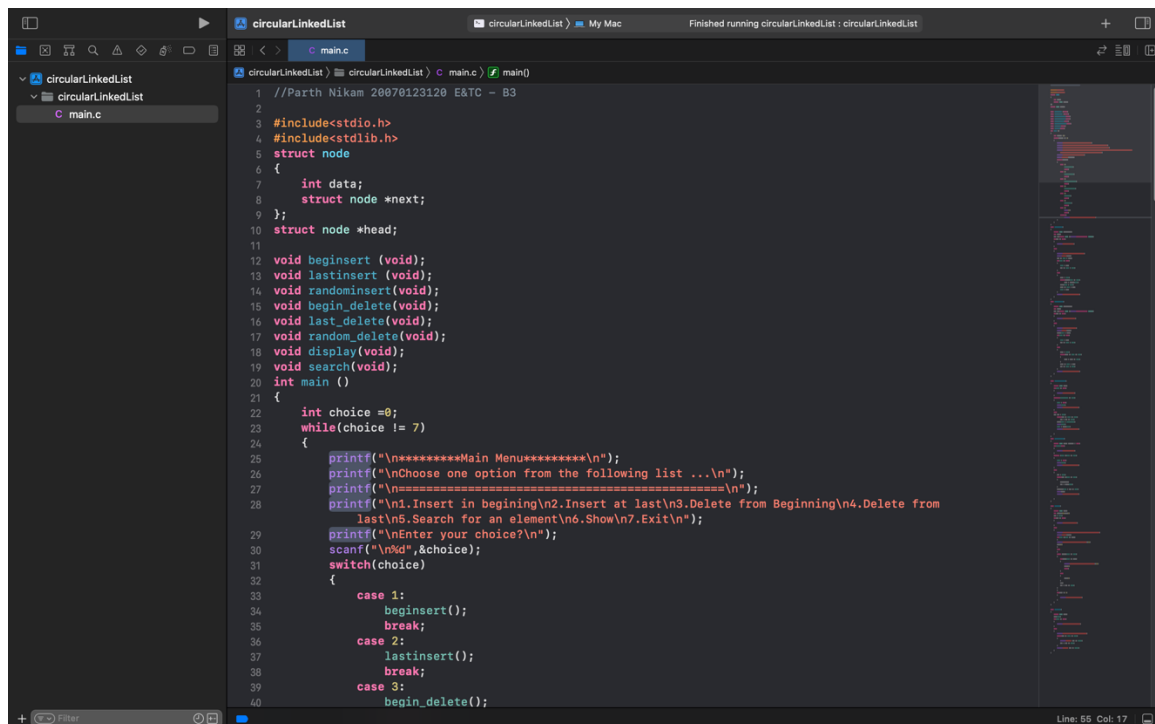
# Data Structures using C

Parth Nikam  
20070123120  
E&TC – B3

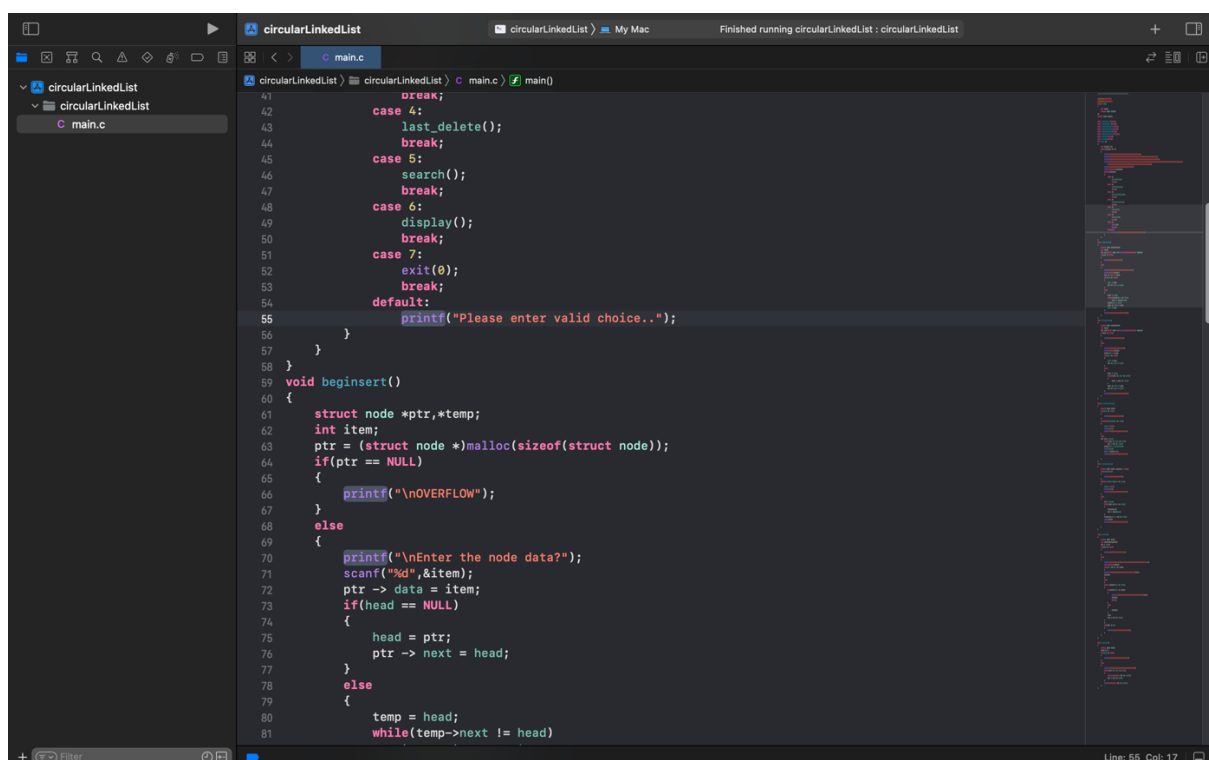
Aim: - Studying and coding on Circular Linked List.

Objective: - To perform Circular Linked List in C language

Code: -



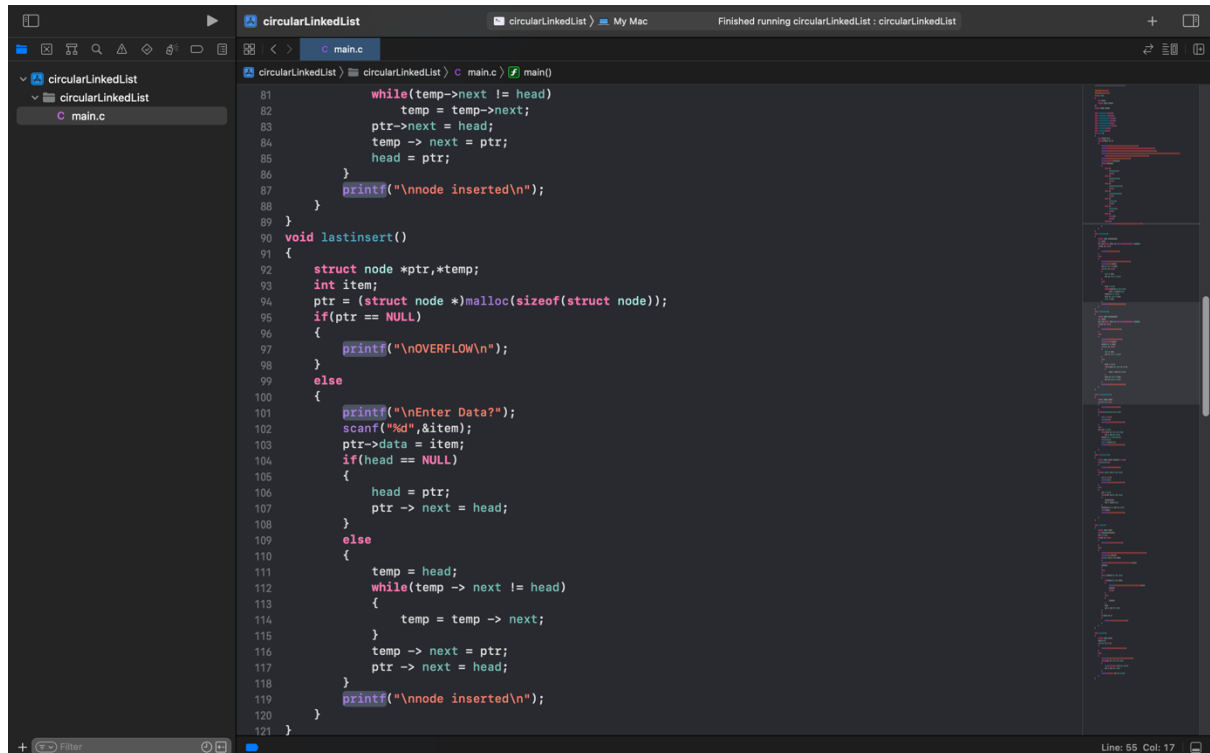
```
1 //Parth Nikam 20070123120 E&TC - B3
2
3 #include<stdio.h>
4 #include<stdlib.h>
5 struct node
6 {
7     int data;
8     struct node *next;
9 };
10 struct node *head;
11
12 void begininsert (void);
13 void lastinsert (void);
14 void randominsert(void);
15 void begin_delete(void);
16 void last_delete(void);
17 void random_delete(void);
18 void display(void);
19 void search(void);
20 int main ()
21 {
22     int choice =0;
23     while(choice != 7)
24     {
25         printf("\n*****Main Menu*****\n");
26         printf("\nChoose one option from the following list ...\n");
27         printf("\n*****\n");
28         printf("\n1.Insert in beginning\n2.Insert at last\n3.Delete from Beginning\n4.Delete from
                last\n5.Search for an element\n6.Show\n7.Exit\n");
29         printf("\nEnter your choice?\n");
30         scanf("%d",&choice);
31         switch(choice)
32         {
33             case 1:
34                 begininsert();
35                 break;
36             case 2:
37                 lastinsert();
38                 break;
39             case 3:
40                 begin_delete();
```



```
41         break;
42         case 4:
43             last_delete();
44             break;
45         case 5:
46             search();
47             break;
48         case 6:
49             display();
50             break;
51         case 7:
52             exit(0);
53             break;
54         default:
55             printf("Please enter valid choice..");
56     }
57 }
58 }
59 void begininsert()
60 {
61     struct node *ptr,*temp;
62     int item;
63     ptr = (struct node *)malloc(sizeof(struct node));
64     if(ptr == NULL)
65     {
66         printf("\nOVERFLOW");
67     }
68     else
69     {
70         printf("\nEnter the node data?");
71         scanf("%d",&item);
72         ptr->data = item;
73         if(head == NULL)
74         {
75             head = ptr;
76             ptr->next = head;
77         }
78         else
79         {
80             temp = head;
81             while(temp->next != head)
```

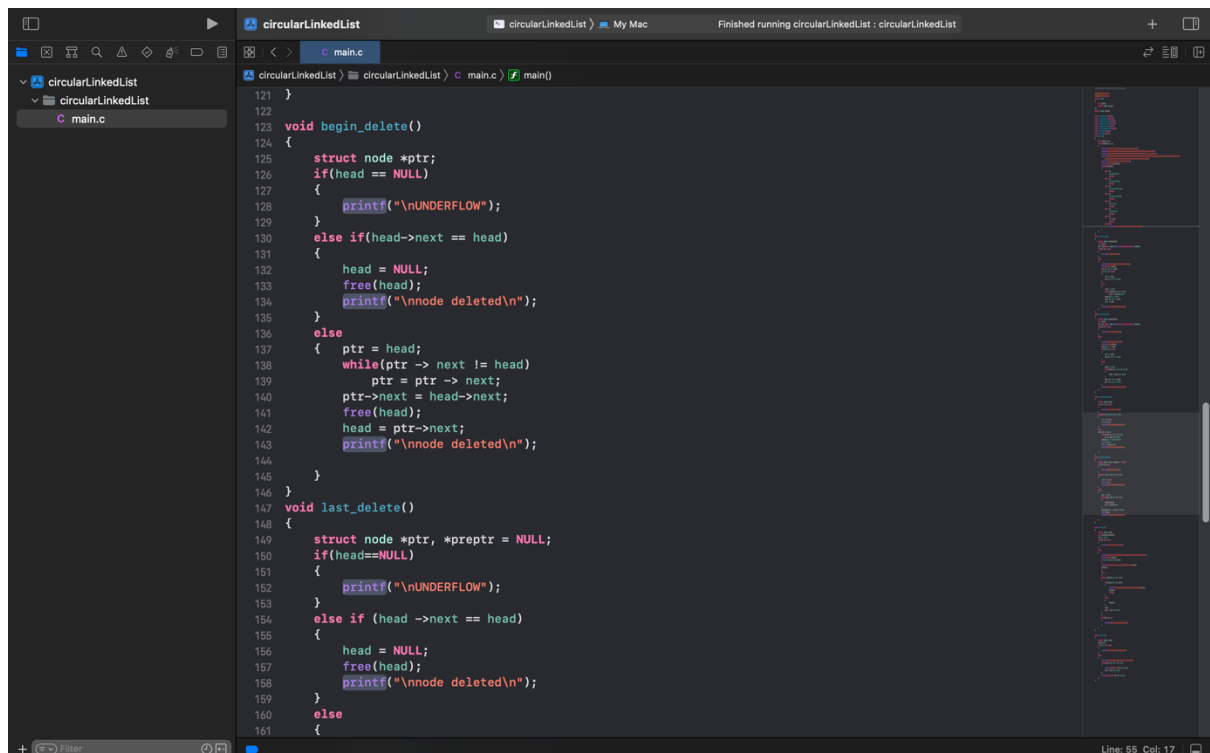
# Data Structures using C

Parth Nikam  
20070123120  
E&TC – B3



```
117         ptr -> next = head;
118     }
119     printf("\nnode inserted\n");
120 }
121 }

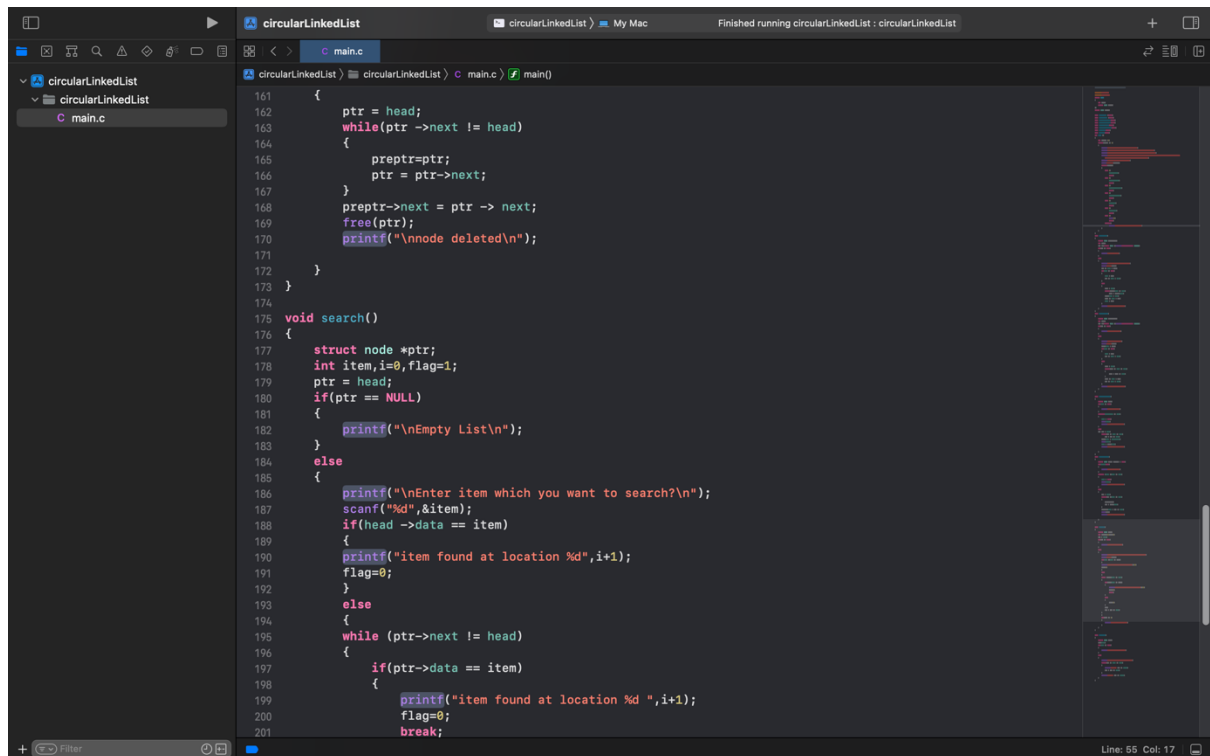
void lastinsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW\n");
    }
    else
    {
        printf("\nEnter Data?");
        scanf("%d",&item);
        ptr->data = item;
        if(head == NULL)
        {
            head = ptr;
            ptr -> next = head;
        }
        else
        {
            temp = head;
            while(temp -> next != head)
            {
                temp = temp -> next;
            }
            temp -> next = ptr;
            ptr -> next = head;
        }
        printf("\nnode inserted\n");
    }
}
```



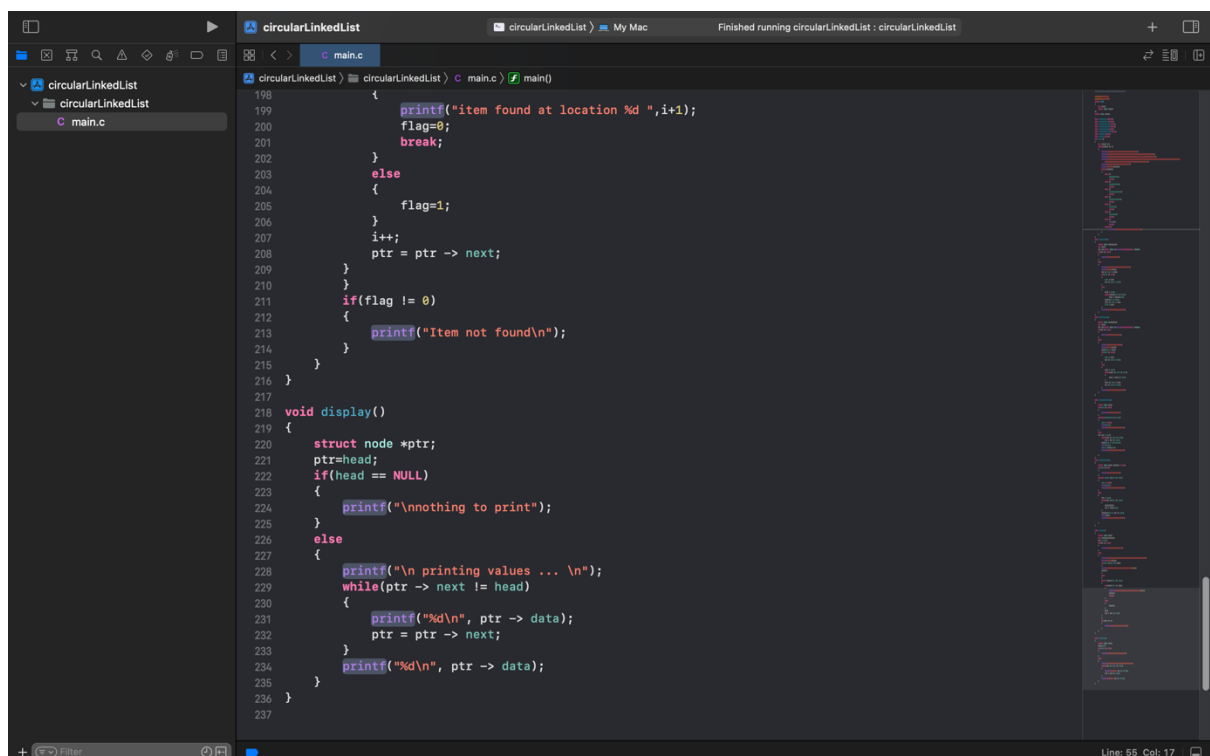
```
121 }
122 }
123 void begin_delete()
124 {
125     struct node *ptr;
126     if(head == NULL)
127     {
128         printf("\nUNDERFLOW");
129     }
130     else if(head->next == head)
131     {
132         head = NULL;
133         free(head);
134         printf("\nnode deleted\n");
135     }
136     else
137     {
138         ptr = head;
139         while(ptr -> next != head)
140             ptr = ptr -> next;
141         ptr->next = head->next;
142         free(head);
143         head = ptr->next;
144         printf("\nnode deleted\n");
145     }
146 }
147 void last_delete()
148 {
149     struct node *ptr, *preptr = NULL;
150     if(head==NULL)
151     {
152         printf("\nUNDERFLOW");
153     }
154     else if (head ->next == head)
155     {
156         head = NULL;
157         free(head);
158         printf("\nnode deleted\n");
159     }
160     else
161     {
```

# Data Structures using C

Parth Nikam  
20070123120  
E&TC – B3



```
161 {
162     ptr = head;
163     while(ptr->next != head)
164     {
165         preptr=ptr;
166         ptr = ptr->next;
167     }
168     preptr->next = ptr->next;
169     free(ptr);
170     printf("\nnode deleted\n");
171 }
172 }
173 }
174
175 void search()
176 {
177     struct node *ptr;
178     int item,i=0,flag=1;
179     ptr = head;
180     if(ptr == NULL)
181     {
182         printf("\nEmpty List\n");
183     }
184     else
185     {
186         printf("\nEnter item which you want to search?\n");
187         scanf("%d",&item);
188         if(head->data == item)
189         {
190             printf("item found at location %d",i+1);
191             flag=0;
192         }
193         else
194         {
195             while (ptr->next != head)
196             {
197                 if(ptr->data == item)
198                 {
199                     printf("item found at location %d ",i+1);
200                     flag=0;
201                     break;
202                 }
203                 ptr = ptr->next;
204                 i++;
205             }
206             if(flag != 0)
207             {
208                 printf("Item not found\n");
209             }
210         }
211     }
212 }
213
214 void display()
215 {
216     struct node *ptr;
217     ptr=head;
218     if(head == NULL)
219     {
220         printf("\nnothing to print");
221     }
222     else
223     {
224         printf("\n printing values ... \n");
225         while(ptr->next != head)
226         {
227             printf("%d\n", ptr->data);
228             ptr = ptr->next;
229         }
230         printf("%d\n", ptr->data);
231     }
232 }
233 }
```



```
198 {
199     printf("item found at location %d ",i+1);
200     flag=0;
201     break;
202 }
203 else
204 {
205     flag=1;
206 }
207 i++;
208 ptr = ptr->next;
209 }
210 }
211 if(flag != 0)
212 {
213     printf("Item not found\n");
214 }
215 }
216 }
217
218 void display()
219 {
220     struct node *ptr;
221     ptr=head;
222     if(head == NULL)
223     {
224         printf("\nnothing to print");
225     }
226     else
227     {
228         printf("\n printing values ... \n");
229         while(ptr->next != head)
230         {
231             printf("%d\n", ptr->data);
232             ptr = ptr->next;
233         }
234         printf("%d\n", ptr->data);
235     }
236 }
237 }
```

# Data Structures using C

Parth Nikam  
20070123120  
E&TC – B3

```
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice?
1
Enter the node data?33
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice?
1
Enter the node data?44
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
```

```
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice?
2
Enter Data?77
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice?
2
Enter Data?88
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
Enter your choice?
2
Enter Data?123
node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
```

# Data Structures using C

Parth Nikam  
20070123120  
E&TC – B3

```
1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice?
4

node deleted

*****Main Menu*****

Choose one option from the following list ...

=====

1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice?
6

printing values ...
44
33
77

*****Main Menu*****

Choose one option from the following list ...

=====

1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit
```

```
6.Show
7.Exit

Enter your choice?
4

node deleted

*****Main Menu*****

Choose one option from the following list ...

=====

1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice?
6

printing values ...
44
33
77

*****Main Menu*****

Choose one option from the following list ...

=====

1.Insert in beginning
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice?
7

Program ended with exit code: 0
```

Result: - Circular Linked List is performed and studies successfully.