

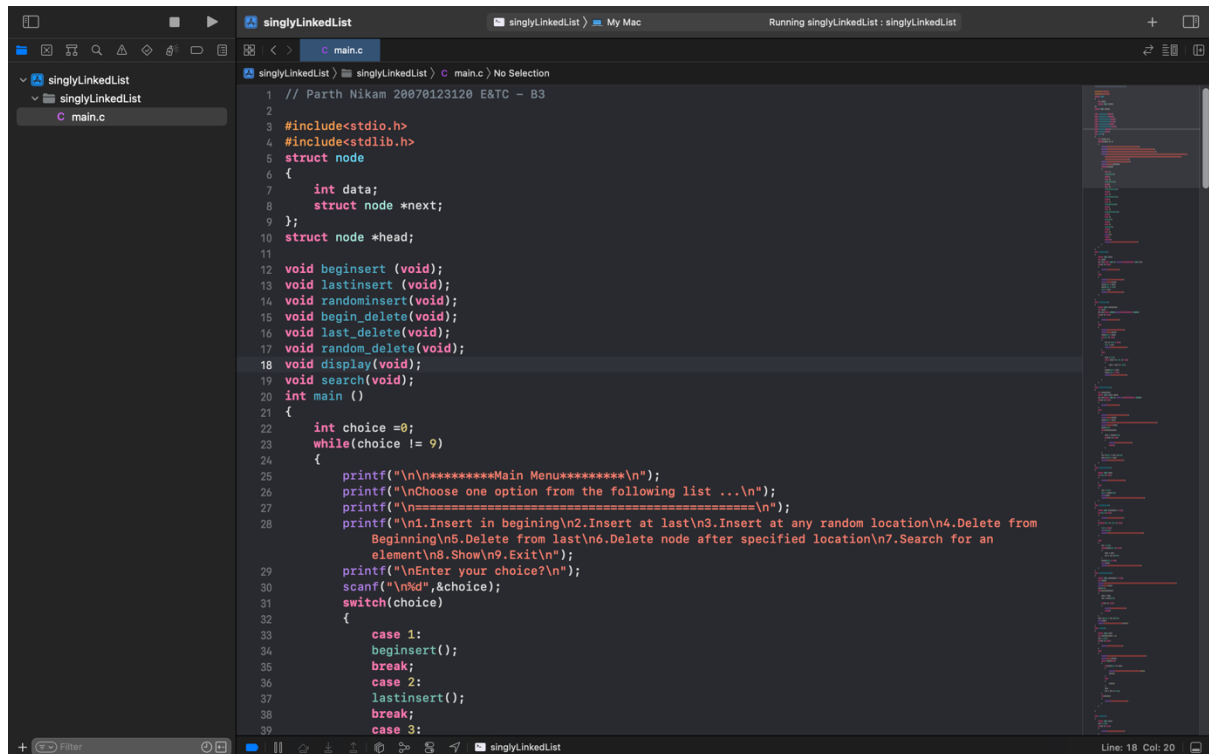
Data Structures using C

Parth Nikam
20070123120
E&TC – B3

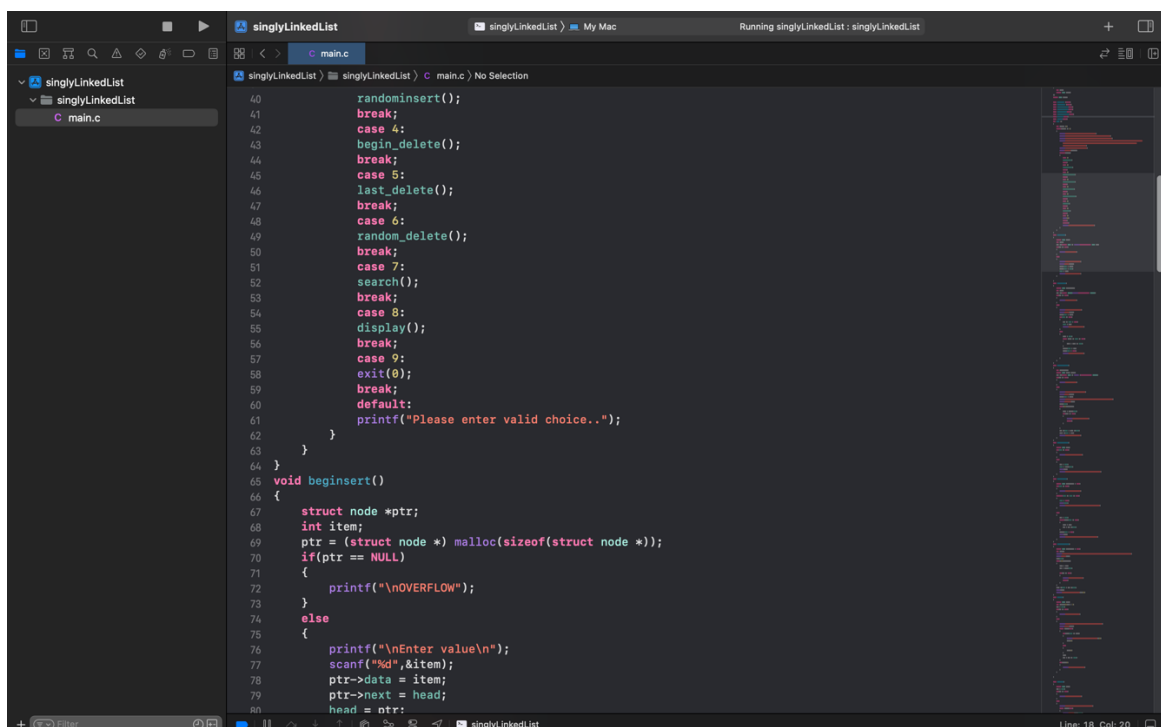
Aim: - Studying and coding on Singly Linked List.

Objective: - To perform Singly Linked List in C language

Code: -



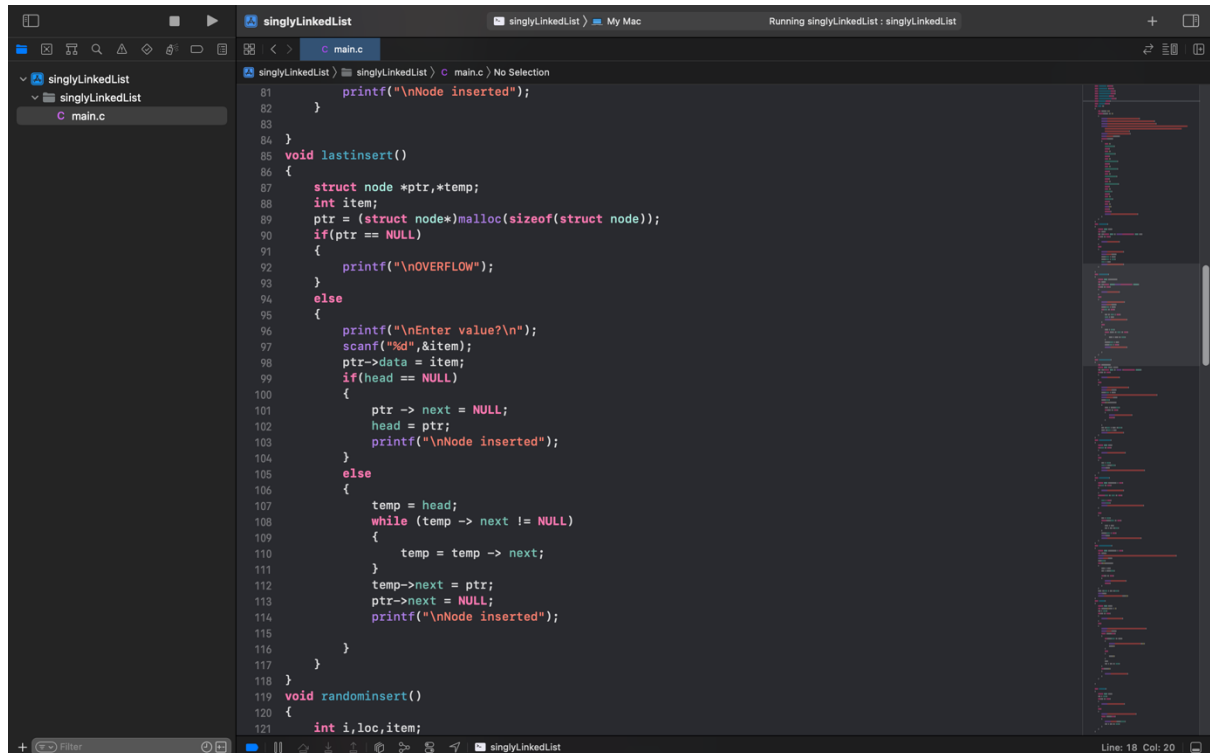
```
1 // Parth Nikam 20070123120 E&TC - B3
2
3 #include<stdio.h>
4 #include<stdlib.h>
5 struct node
6 {
7     int data;
8     struct node *next;
9 };
10 struct node *head;
11
12 void begininsert (void);
13 void lastinsert (void);
14 void randominsert (void);
15 void begin_delete (void);
16 void last_delete (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 != 9)
24     {
25         printf("\n\n*****Main Menu*****\n");
26         printf("\nChoose one option from the following list ...\n");
27         printf("\n*****\n");
28         printf("\n1.Insert in begining\n2.Insert at last\n3.Insert at any random location\n4.Delete from
                Beginning\n5.Delete from last\n6.Delete node after specified location\n7.Search for an
                element\n8.Show\n9.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:
```



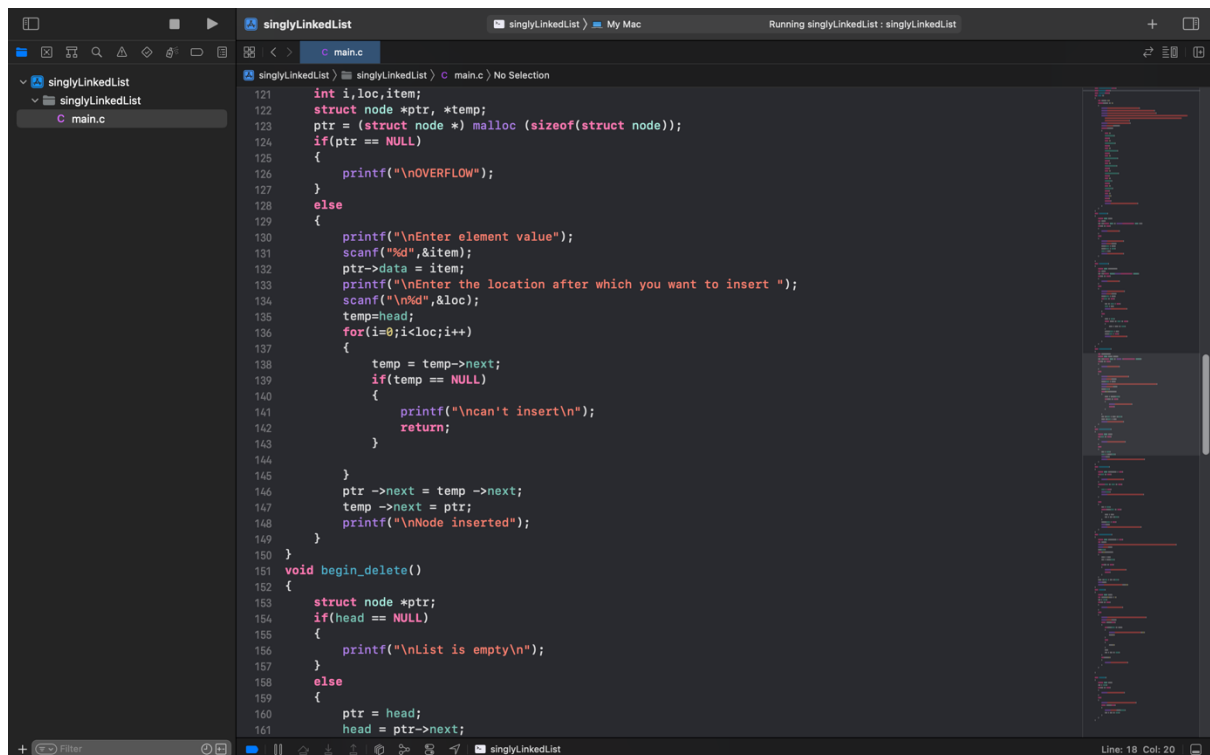
```
40         randominsert();
41         break;
42         case 4:
43             begin_delete();
44             break;
45         case 5:
46             last_delete();
47             break;
48         case 6:
49             random_delete();
50             break;
51         case 7:
52             search();
53             break;
54         case 8:
55             display();
56             break;
57         case 9:
58             exit(0);
59             break;
60         default:
61             printf("Please enter valid choice..");
62     }
63 }
64
65 void begininsert()
66 {
67     struct node *ptr;
68     int item;
69     ptr = (struct node *) malloc(sizeof(struct node *));
70     if(ptr == NULL)
71     {
72         printf("\nOVERFLOW");
73     }
74     else
75     {
76         printf("\nEnter value\n");
77         scanf("%d",&item);
78         ptr->data = item;
79         ptr->next = head;
80         head = ptr;
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



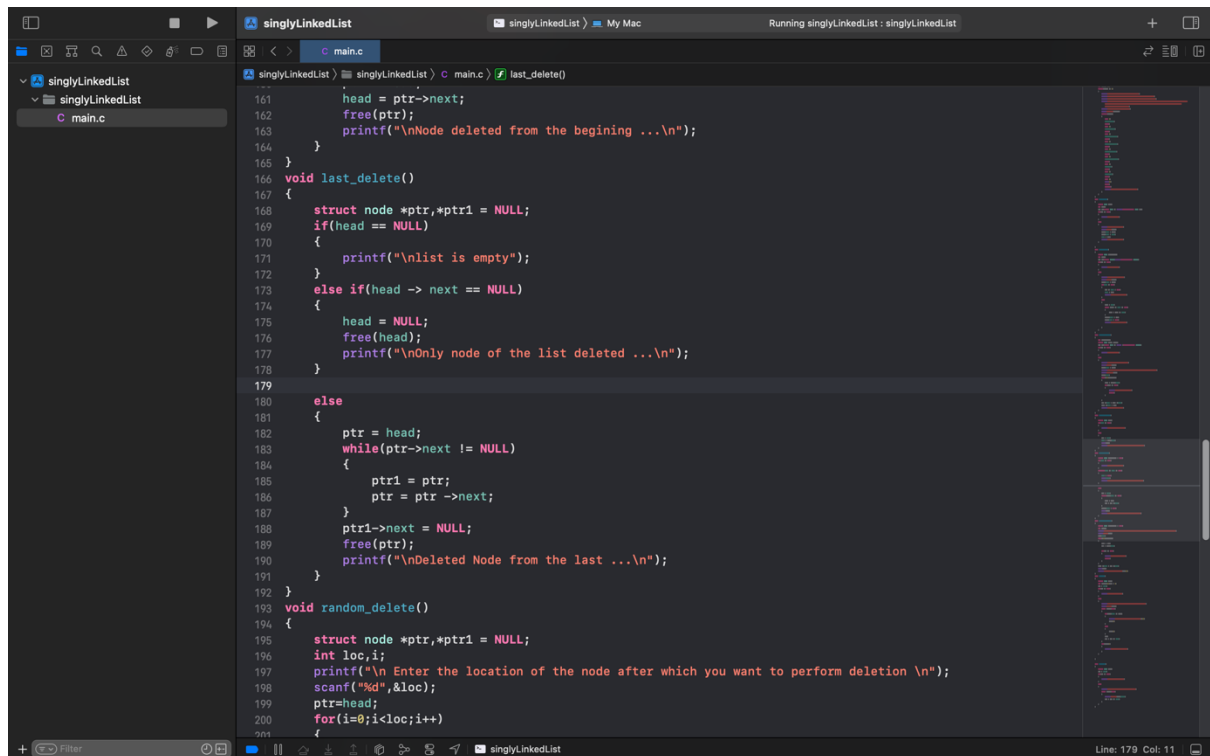
```
81     printf("\nNode inserted");
82 }
83 }
84 }
85 void lastinsert()
86 {
87     struct node *ptr,*temp;
88     int item;
89     ptr = (struct node*)malloc(sizeof(struct node));
90     if(ptr == NULL)
91     {
92         printf("\nOVERFLOW");
93     }
94     else
95     {
96         printf("\nEnter value?\n");
97         scanf("%d",&item);
98         ptr->data = item;
99         if(head == NULL)
100         {
101             ptr -> next = NULL;
102             head = ptr;
103             printf("\nNode inserted");
104         }
105         else
106         {
107             temp = head;
108             while (temp -> next != NULL)
109             {
110                 temp = temp -> next;
111             }
112             temp->next = ptr;
113             ptr->next = NULL;
114             printf("\nNode inserted");
115         }
116     }
117 }
118 }
119 void randominsert()
120 {
121     int i,loc,item;
```



```
121     int i,loc,item;
122     struct node *ptr, *temp;
123     ptr = (struct node *) malloc (sizeof(struct node));
124     if(ptr == NULL)
125     {
126         printf("\nOVERFLOW");
127     }
128     else
129     {
130         printf("\nEnter element value");
131         scanf("%d",&item);
132         ptr->data = item;
133         printf("\nEnter the location after which you want to insert ");
134         scanf("%d",&loc);
135         temp=head;
136         for(i=0;i<loc;i++)
137         {
138             temp = temp->next;
139             if(temp == NULL)
140             {
141                 printf("\ncan't insert\n");
142                 return;
143             }
144         }
145         ptr ->next = temp ->next;
146         temp ->next = ptr;
147         printf("\nNode inserted");
148     }
149 }
150 }
151 void begin_delete()
152 {
153     struct node *ptr;
154     if(head == NULL)
155     {
156         printf("\nList is empty\n");
157     }
158     else
159     {
160         ptr = head;
161         head = ptr->next;
```

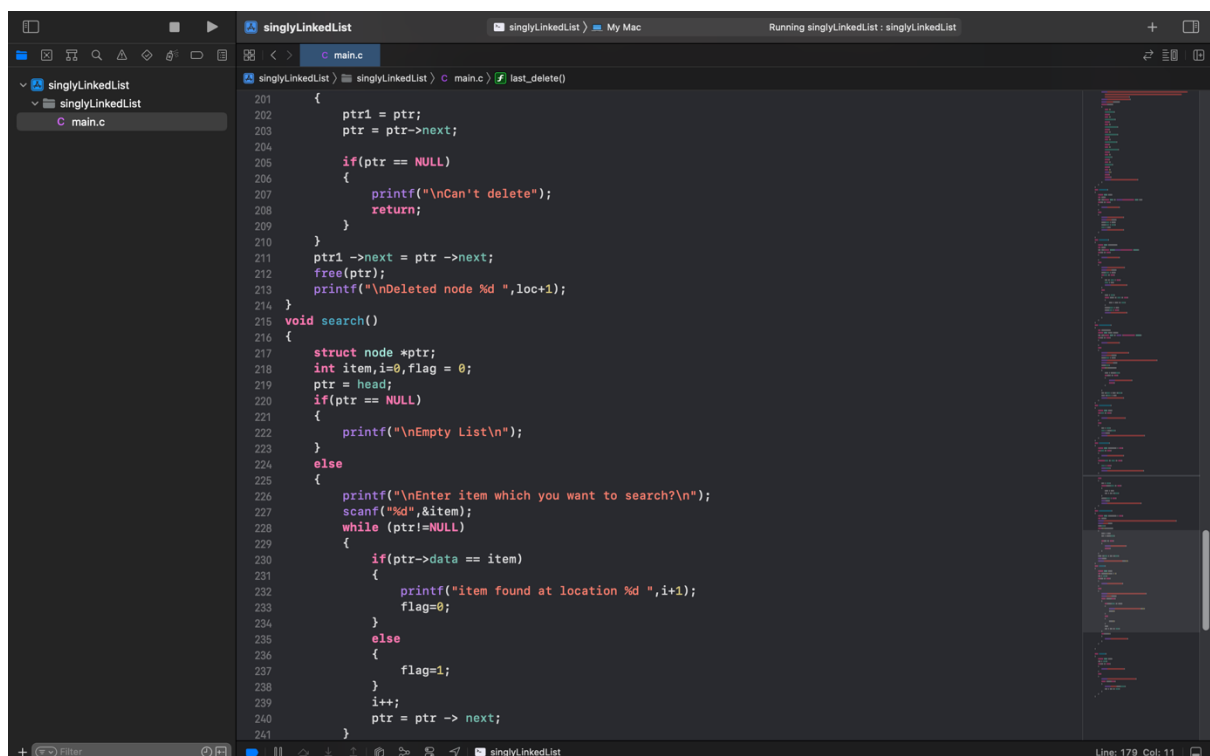
Data Structures using C

Parth Nikam
20070123120
E&TC – B3



```
singlyLinkedList
├── singlyLinkedList
│   └── C main.c
└── C main.c

singlyLinkedList > singlyLinkedList > C main.c > last_delete()
161     head = ptr->next;
162     free(ptr);
163     printf("\nNode deleted from the begining ...\n");
164 }
165 }
166 void last_delete()
167 {
168     struct node *ptr,*ptr1 = NULL;
169     if(head == NULL)
170     {
171         printf("\nlist is empty");
172     }
173     else if(head->next == NULL)
174     {
175         head = NULL;
176         free(head);
177         printf("\nOnly node of the list deleted ...\n");
178     }
179     else
180     {
181         ptr = head;
182         while(ptr->next != NULL)
183         {
184             ptr1 = ptr;
185             ptr = ptr->next;
186         }
187         ptr1->next = NULL;
188         free(ptr);
189         printf("\nDeleted Node from the last ...\n");
190     }
191 }
192 }
193 void random_delete()
194 {
195     struct node *ptr,*ptr1 = NULL;
196     int loc,i;
197     printf("\n Enter the location of the node after which you want to perform deletion \n");
198     scanf("%d",&loc);
199     ptr=head;
200     for(i=0;i<loc;i++)
201     {
```

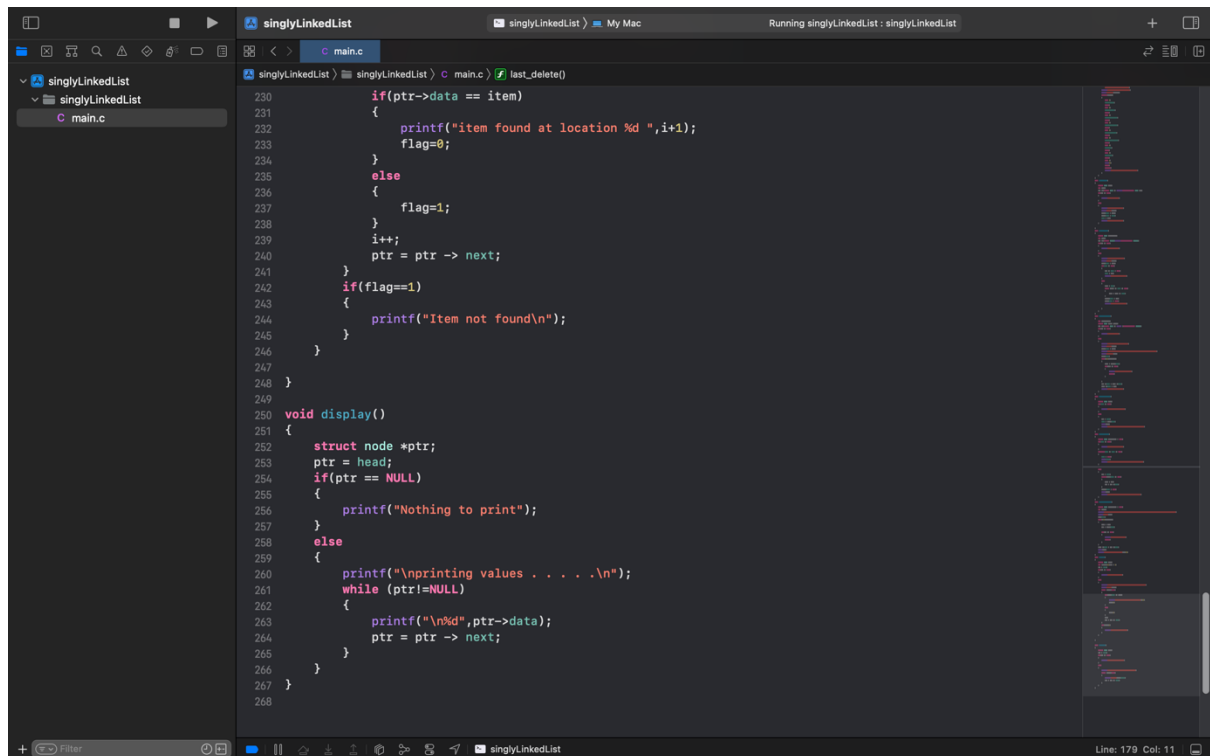


```
singlyLinkedList
├── singlyLinkedList
│   └── C main.c
└── C main.c

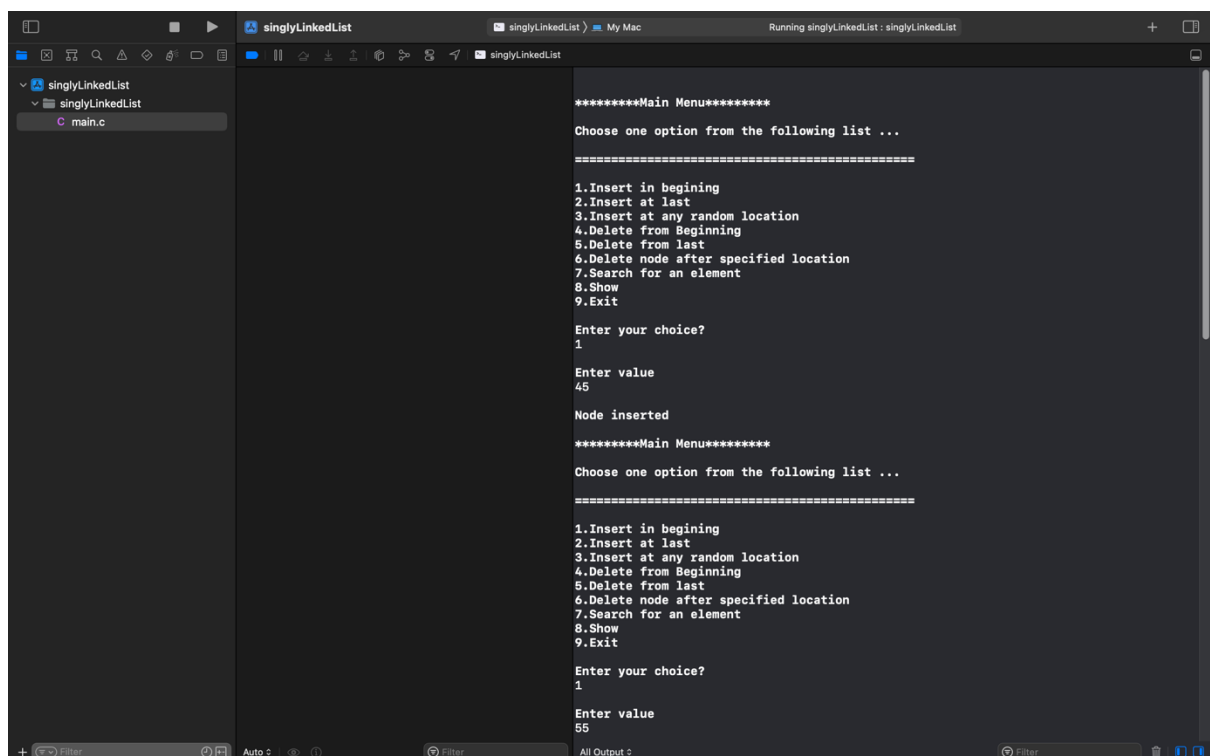
singlyLinkedList > singlyLinkedList > C main.c > last_delete()
201 {
202     ptr1 = ptr;
203     ptr = ptr->next;
204
205     if(ptr == NULL)
206     {
207         printf("\nCan't delete");
208         return;
209     }
210
211     ptr1->next = ptr->next;
212     free(ptr);
213     printf("\nDeleted node %d ",loc+1);
214 }
215 void search()
216 {
217     struct node *ptr;
218     int item,i=0,flag = 0;
219     ptr = head;
220     if(ptr == NULL)
221     {
222         printf("\nEmpty List\n");
223     }
224     else
225     {
226         printf("\nEnter item which you want to search?\n");
227         scanf("%d",&item);
228         while (ptr!=NULL)
229         {
230             if(ptr->data == item)
231             {
232                 printf("item found at location %d ",i+1);
233                 flag=0;
234             }
235             else
236             {
237                 flag=1;
238             }
239             i++;
240             ptr = ptr->next;
241         }
242     }
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



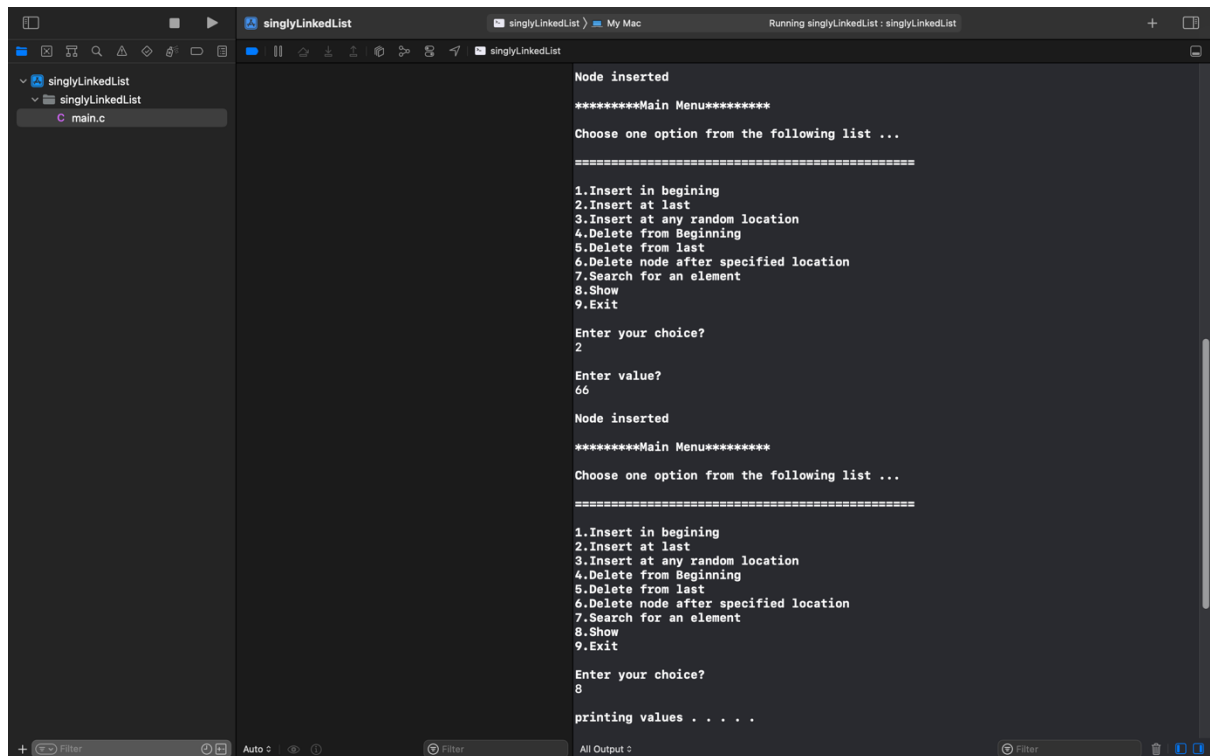
```
230     if(ptr->data == item)
231     {
232         printf("item found at location %d ",i+1);
233         flag=0;
234     }
235     else
236     {
237         flag=1;
238     }
239     i++;
240     ptr = ptr -> next;
241 }
242 if(flag==1)
243 {
244     printf("Item not found\n");
245 }
246 }
247 }
248 }
249
250 void display()
251 {
252     struct node *ptr;
253     ptr = head;
254     if(ptr == NULL)
255     {
256         printf("Nothing to print");
257     }
258     else
259     {
260         printf("\nprinting values . . . . \n");
261         while (ptr!=NULL)
262         {
263             printf("\n%d",ptr->data);
264             ptr = ptr -> next;
265         }
266     }
267 }
268 }
```



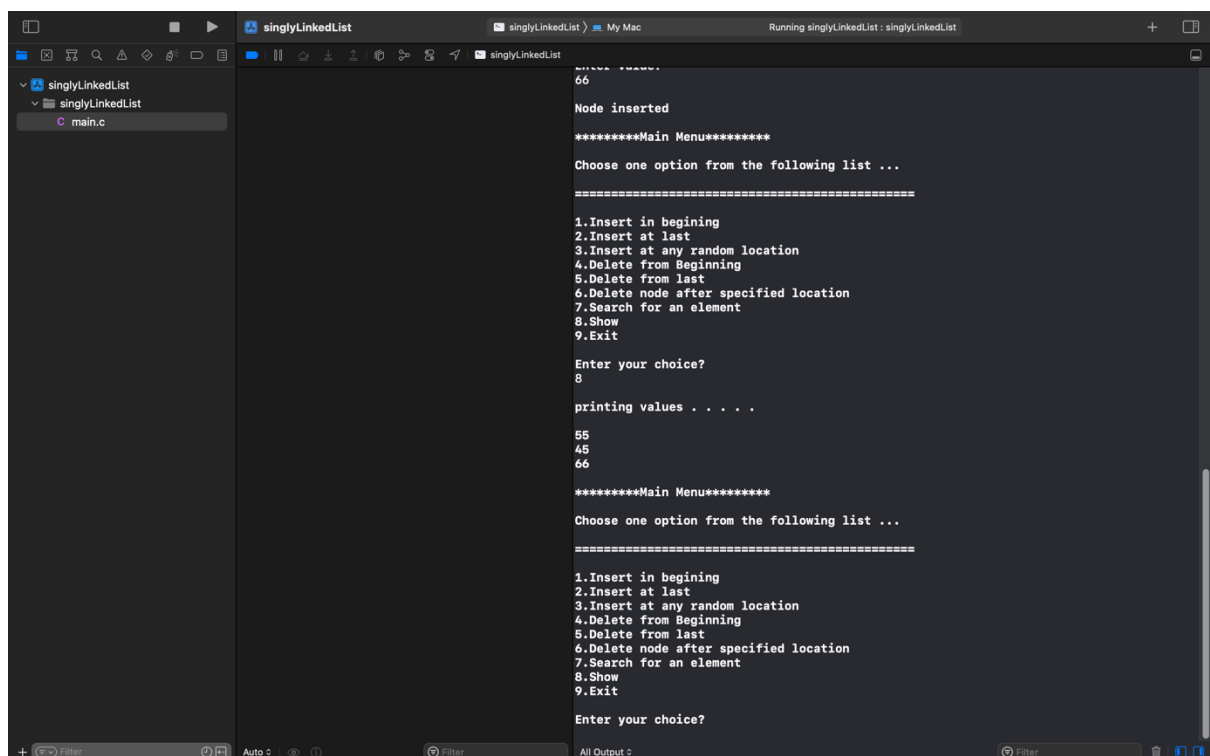
```
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
1
Enter value
45
Node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
1
Enter value
55
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



```
singlyLinkedList
Node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
2
Enter value?
66
Node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
8
printing values . . . .
```



```
singlyLinkedList
Node inserted
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
8
printing values . . . .
55
45
66
*****Main Menu*****
Choose one option from the following list ...
=====
1.Insert in begining
2.Insert at last
3.Insert at any random location
4.Delete from Beginning
5.Delete from last
6.Delete node after specified location
7.Search for an element
8.Show
9.Exit
Enter your choice?
```

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