

Data Structures using C

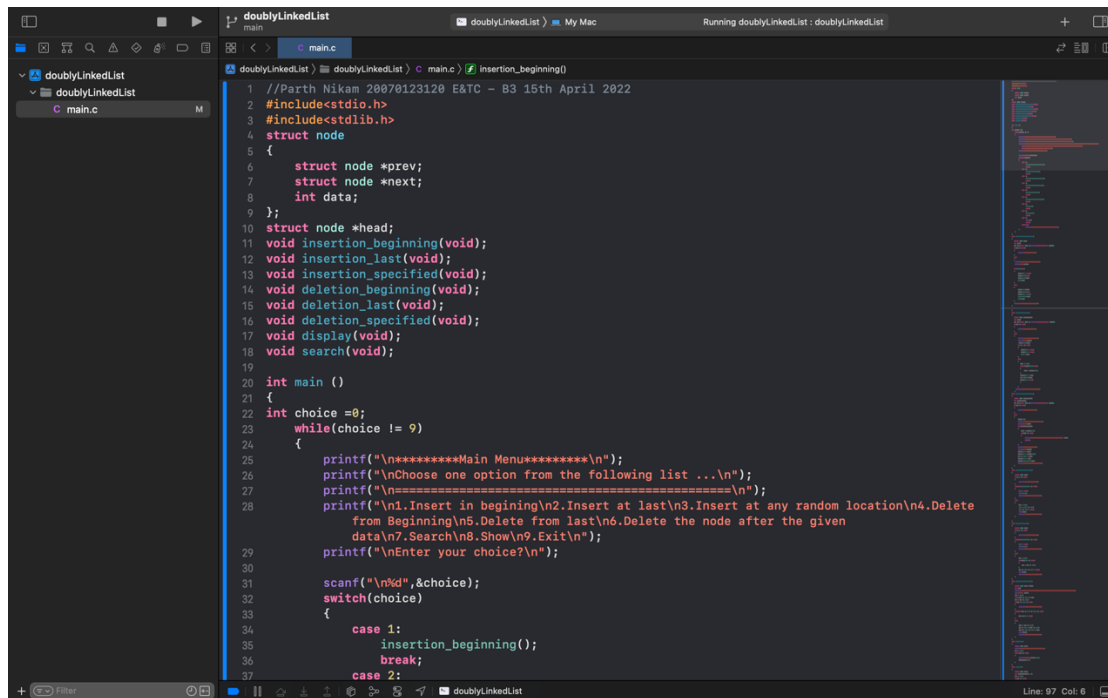
15th April 2022

Parth Nikam
20070123120
E&TC – B3

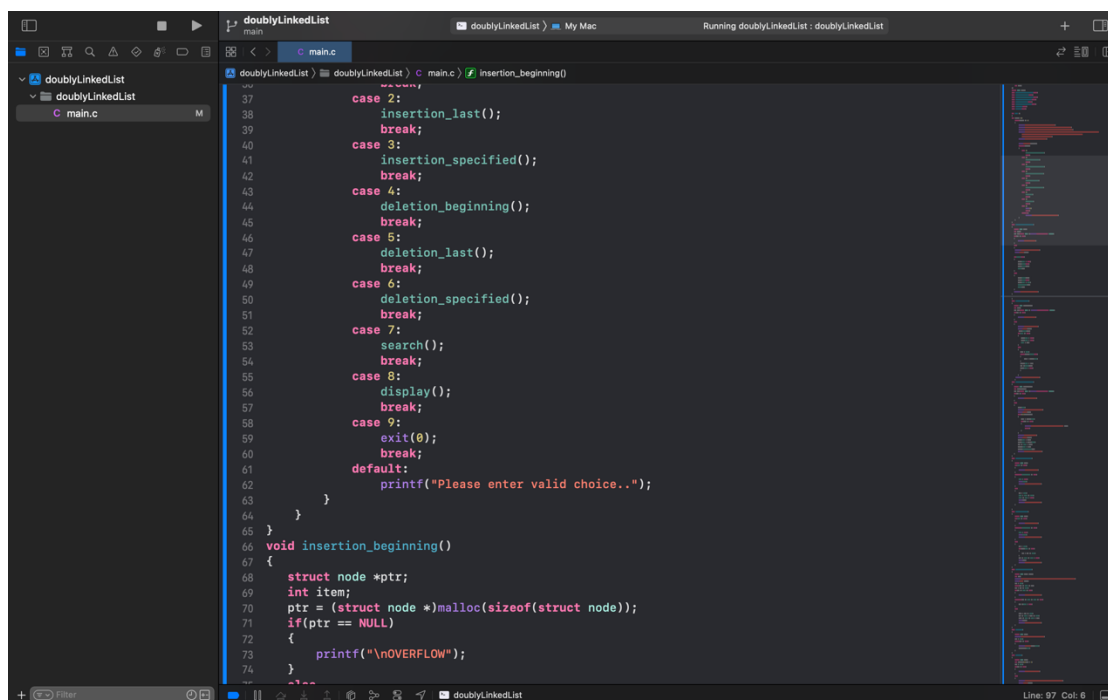
Aim: - Studying and coding on Doubly Linked List.

Objective: - To perform Doubly Linked List in C language

Code: -



```
1 //Parth Nikam 20070123120 E&TC - B3 15th April 2022
2 #include<stdio.h>
3 #include<stdlib.h>
4 struct node
5 {
6     struct node *prev;
7     struct node *next;
8     int data;
9 };
10 struct node *head;
11 void insertion_beginning(void);
12 void insertion_last(void);
13 void insertion_specified(void);
14 void deletion_beginning(void);
15 void deletion_last(void);
16 void deletion_specified(void);
17 void display(void);
18 void search(void);
19
20 int main ()
21 {
22     int choice =0;
23     while(choice != 9)
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.Insert at any random location\n4.Delete
        from Beginning\n5.Delete from last\n6.Delete the node after the given
        data\n7.Search\n8.Show\n9.Exit\n");
29         printf("\nEnter your choice?\n");
30
31         scanf("%d",&choice);
32         switch(choice)
33         {
34             case 1:
35                 insertion_beginning();
36                 break;
37             case 2:
```

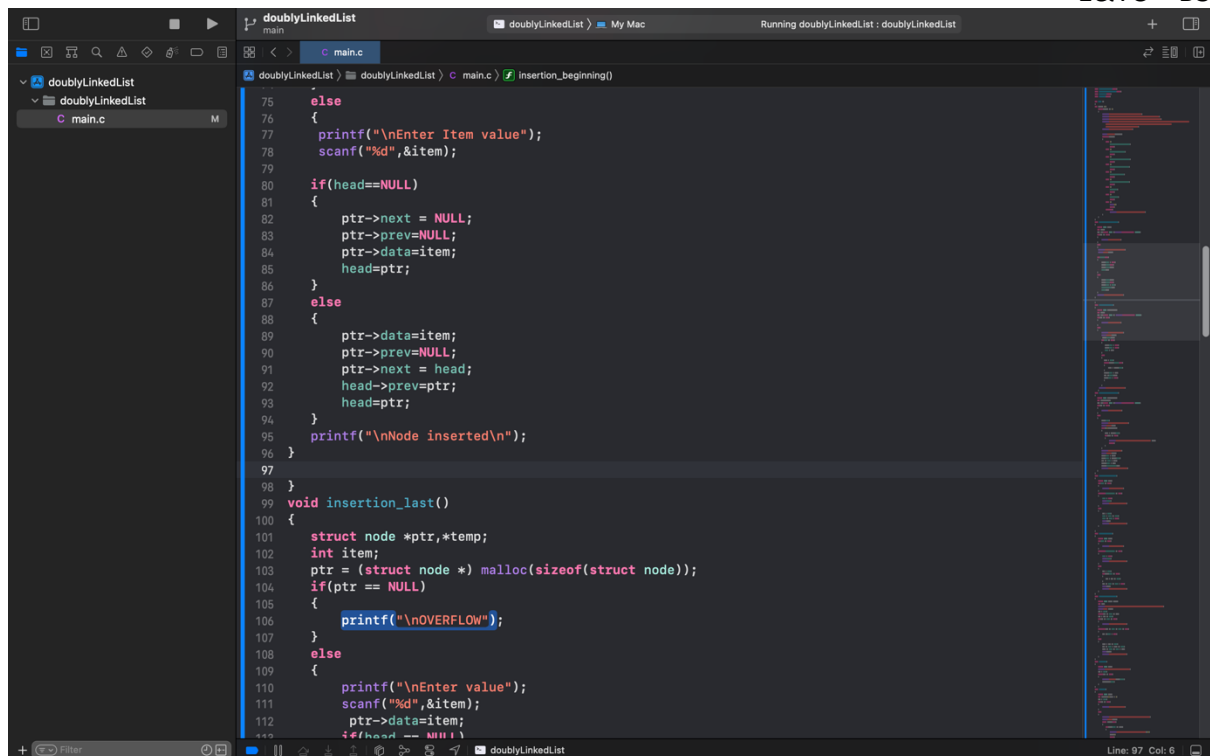


```
38                 insertion_last();
39                 break;
40             case 3:
41                 insertion_specified();
42                 break;
43             case 4:
44                 deletion_beginning();
45                 break;
46             case 5:
47                 deletion_last();
48                 break;
49             case 6:
50                 deletion_specified();
51                 break;
52             case 7:
53                 search();
54                 break;
55             case 8:
56                 display();
57                 break;
58             case 9:
59                 exit(0);
60                 break;
61             default:
62                 printf("Please enter valid choice..");
63         }
64     }
65 }
66 void insertion_beginning()
67 {
68     struct node *ptr;
69     int item;
70     ptr = (struct node *)malloc(sizeof(struct node));
71     if(ptr == NULL)
72     {
73         printf("\nOVERFLOW");
74     }
75 }
```

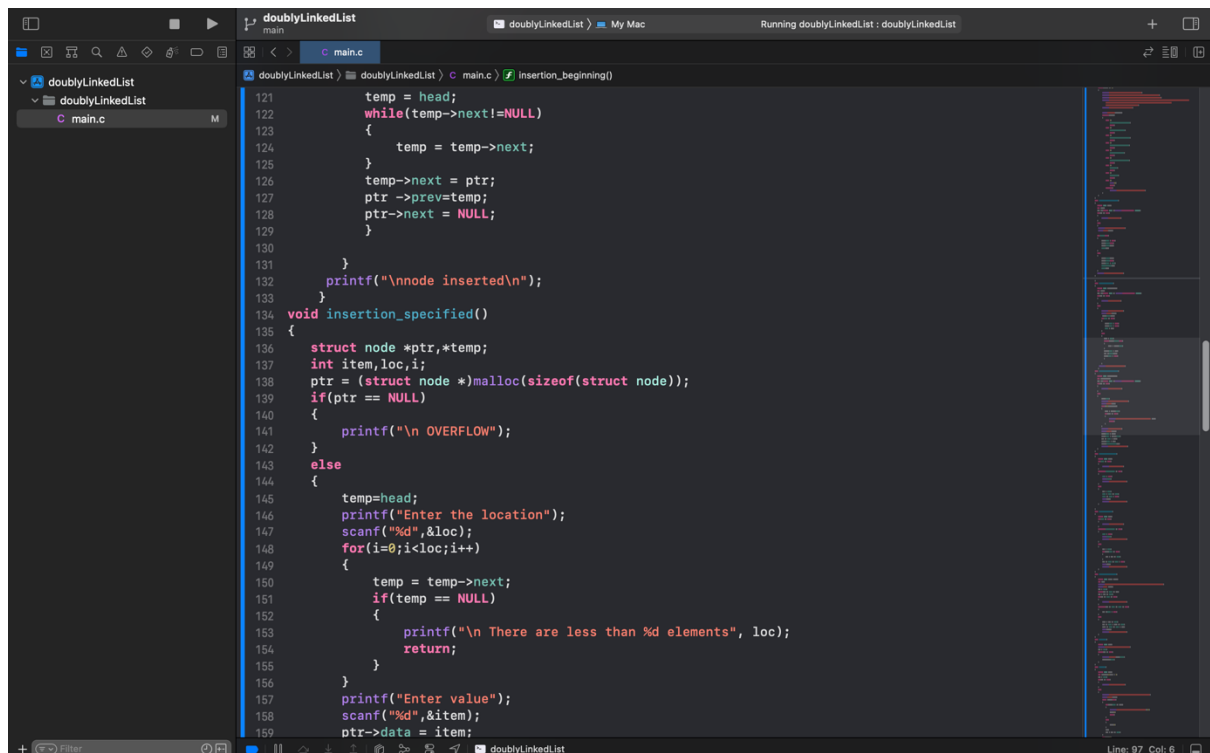
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3



```
75     else
76     {
77         printf("\nEnter Item value");
78         scanf("%d",&item);
79
80         if(head==NULL)
81         {
82             ptr->next = NULL;
83             ptr->prev=NULL;
84             ptr->data=item;
85             head=ptr;
86         }
87         else
88         {
89             ptr->data=item;
90             ptr->prev=NULL;
91             ptr->next = head;
92             head->prev=ptr;
93             head=ptr;
94         }
95         printf("\nNode inserted\n");
96     }
97 }
98
99 void insertion_last()
100 {
101     struct node *ptr,*temp;
102     int item;
103     ptr = (struct node *) malloc(sizeof(struct node));
104     if(ptr == NULL)
105     {
106         printf("\nOVERFLOW");
107     }
108     else
109     {
110         printf("\nEnter value");
111         scanf("%d",&item);
112         ptr->data=item;
113         if(head == NULL)
```



```
121         temp = head;
122         while(temp->next!=NULL)
123         {
124             temp = temp->next;
125         }
126         temp->next = ptr;
127         ptr->prev=temp;
128         ptr->next = NULL;
129     }
130 }
131 printf("\nnode inserted\n");
132 }
133 void insertion_specified()
134 {
135     struct node *ptr,*temp;
136     int item,loc,i;
137     ptr = (struct node *) malloc(sizeof(struct node));
138     if(ptr == NULL)
139     {
140         printf("\n OVERFLOW");
141     }
142     else
143     {
144         temp=head;
145         printf("Enter the location");
146         scanf("%d",&loc);
147         for(i=0;i<loc;i++)
148         {
149             temp = temp->next;
150             if(temp == NULL)
151             {
152                 printf("\n There are less than %d elements", loc);
153                 return;
154             }
155         }
156         printf("Enter value");
157         scanf("%d",&item);
158         ptr->data = item;
```

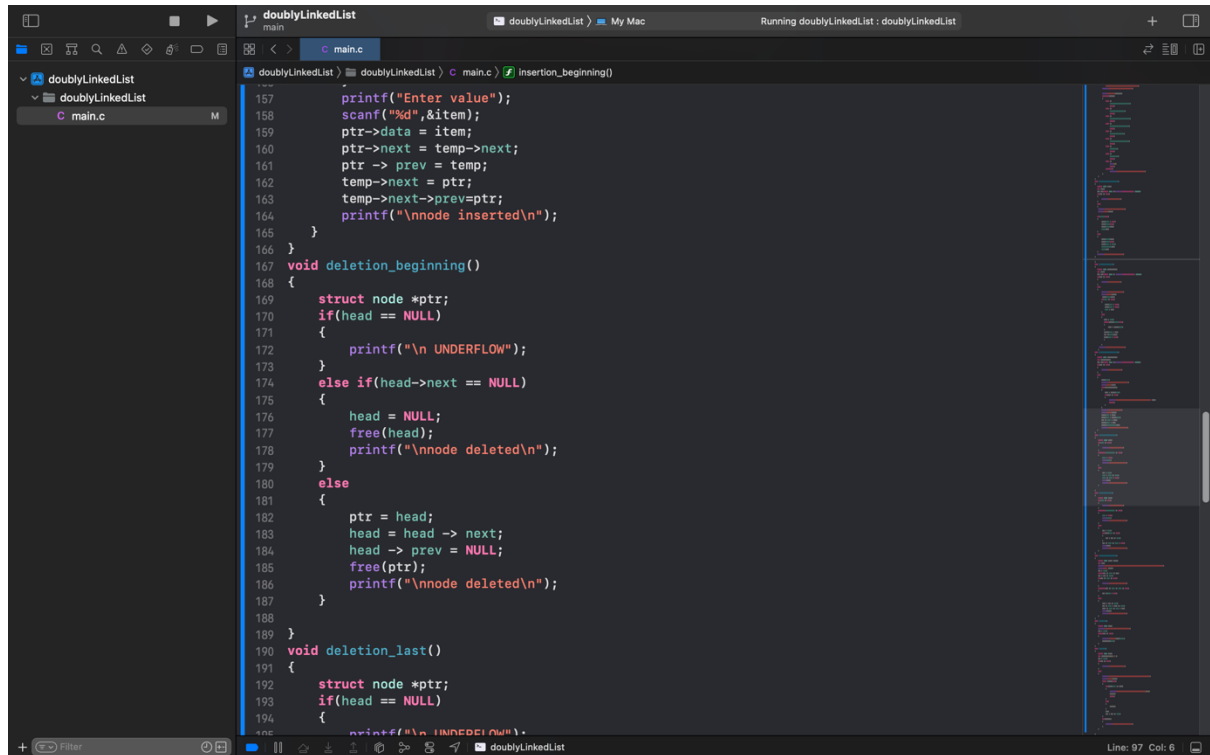
Data Structures using C

15th April 2022

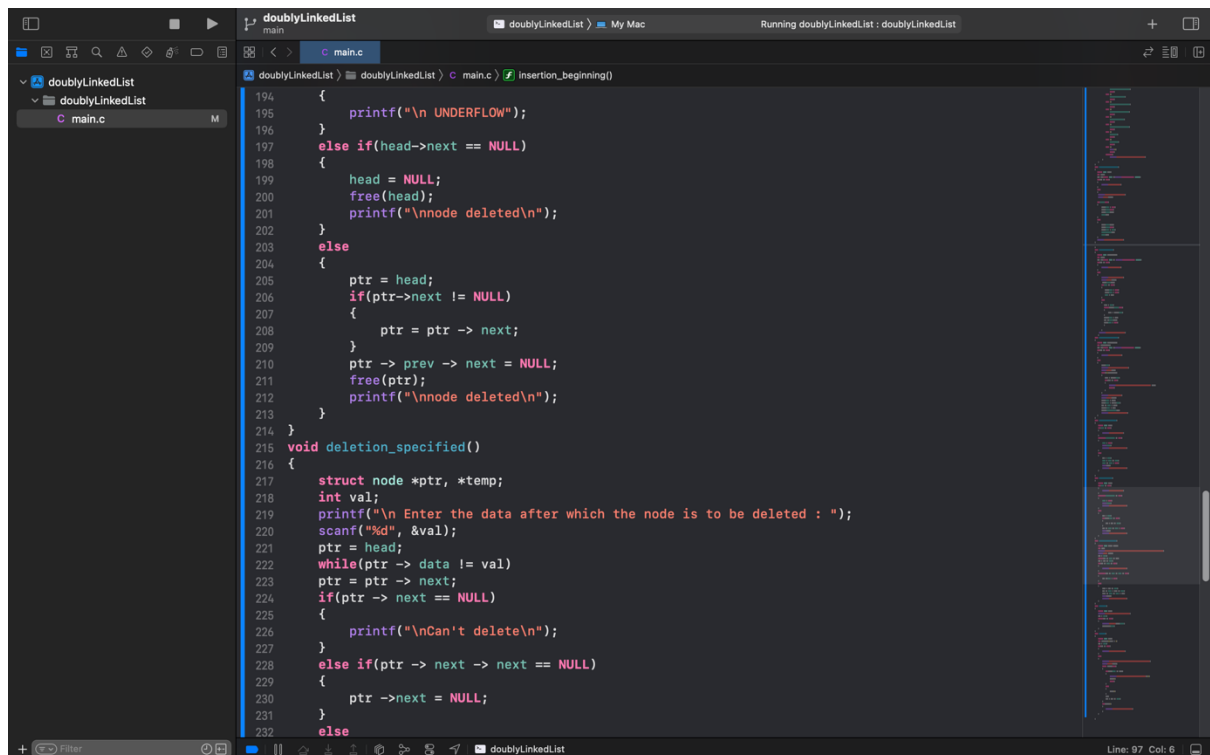
Parth Nikam

20070123120

E&TC – B3



```
157     printf("Enter value");
158     scanf("%d",&item);
159     ptr->data = item;
160     ptr->next = temp->next;
161     ptr -> prev = temp;
162     temp->next = ptr;
163     temp->next->prev=ptr;
164     printf("\nnode inserted\n");
165 }
166 }
167 void deletion_beginning()
168 {
169     struct node *ptr;
170     if(head == NULL)
171     {
172         printf("\n UNDERFLOW");
173     }
174     else if(head->next == NULL)
175     {
176         head = NULL;
177         free(head);
178         printf("\nnode deleted\n");
179     }
180     else
181     {
182         ptr = head;
183         head = head -> next;
184         head -> prev = NULL;
185         free(ptr);
186         printf("\nnode deleted\n");
187     }
188 }
189 }
190 void deletion_last()
191 {
192     struct node *ptr;
193     if(head == NULL)
194     {
```

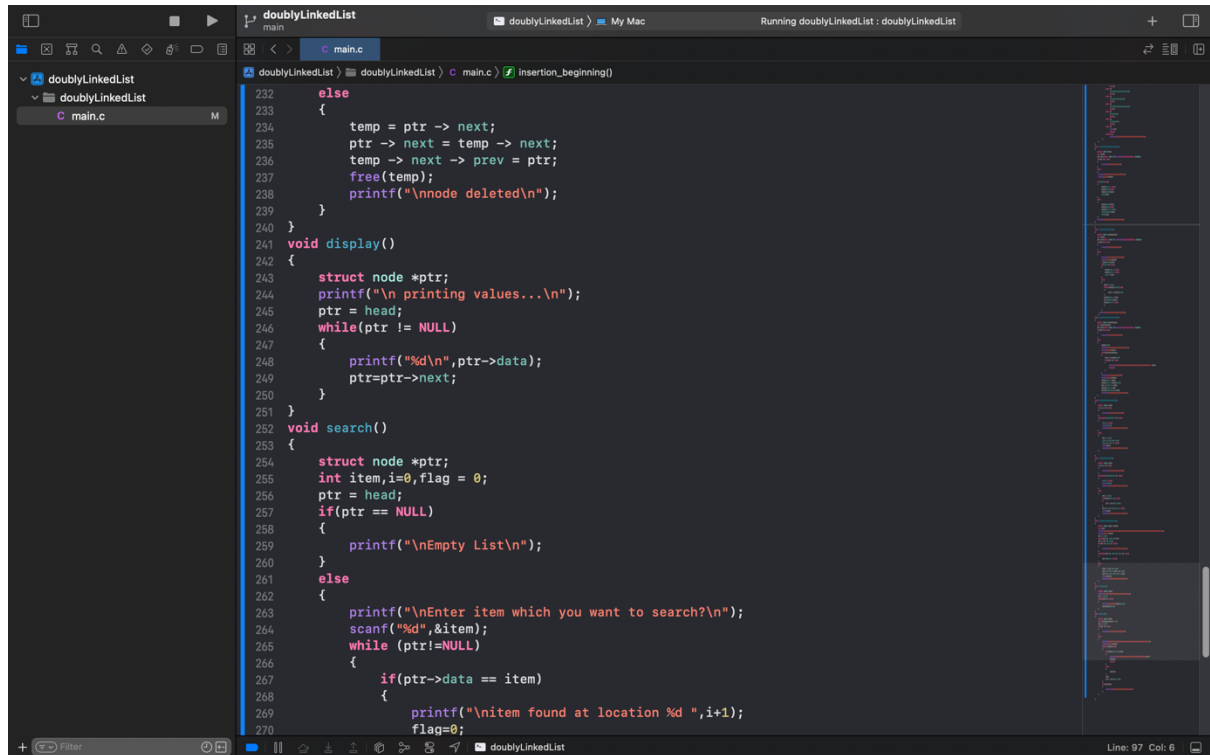


```
194     {
195         printf("\n UNDERFLOW");
196     }
197     else if(head->next == NULL)
198     {
199         head = NULL;
200         free(head);
201         printf("\nnode deleted\n");
202     }
203     else
204     {
205         ptr = head;
206         if(ptr->next != NULL)
207         {
208             ptr = ptr -> next;
209         }
210         ptr -> prev -> next = NULL;
211         free(ptr);
212         printf("\nnode deleted\n");
213     }
214 }
215 void deletion_specified()
216 {
217     struct node *ptr, *temp;
218     int val;
219     printf("\n Enter the data after which the node is to be deleted : ");
220     scanf("%d", &val);
221     ptr = head;
222     while(ptr -> data != val)
223     ptr = ptr -> next;
224     if(ptr -> next == NULL)
225     {
226         printf("\nCan't delete\n");
227     }
228     else if(ptr -> next -> next == NULL)
229     {
230         ptr ->next = NULL;
231     }
232     else
```

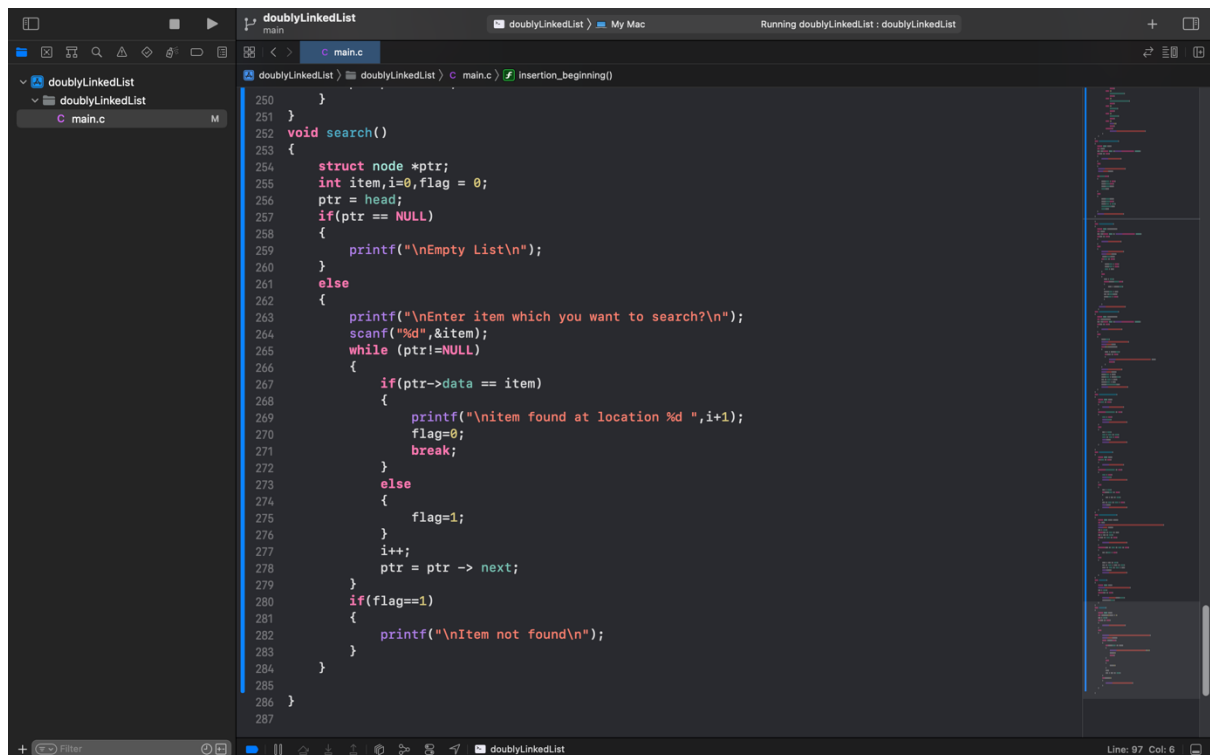
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3



```
232     else
233     {
234         temp = ptr -> next;
235         ptr -> next = temp -> next;
236         temp -> next -> prev = ptr;
237         free(temp);
238         printf("\nnode deleted\n");
239     }
240 }
241 void display()
242 {
243     struct node *ptr;
244     printf("\n printing values...\n");
245     ptr = head;
246     while(ptr != NULL)
247     {
248         printf("%d\n",ptr->data);
249         ptr=ptr->next;
250     }
251 }
252 void search()
253 {
254     struct node *ptr;
255     int item,i=0,flag = 0;
256     ptr = head;
257     if(ptr == NULL)
258     {
259         printf("\nEmpty List\n");
260     }
261     else
262     {
263         printf("\nEnter item which you want to search?\n");
264         scanf("%d",&item);
265         while (ptr!=NULL)
266         {
267             if(ptr->data == item)
268             {
269                 printf("\nitem found at location %d ",i+1);
270                 flag=0;
271             }
272         }
273     }
274 }
```

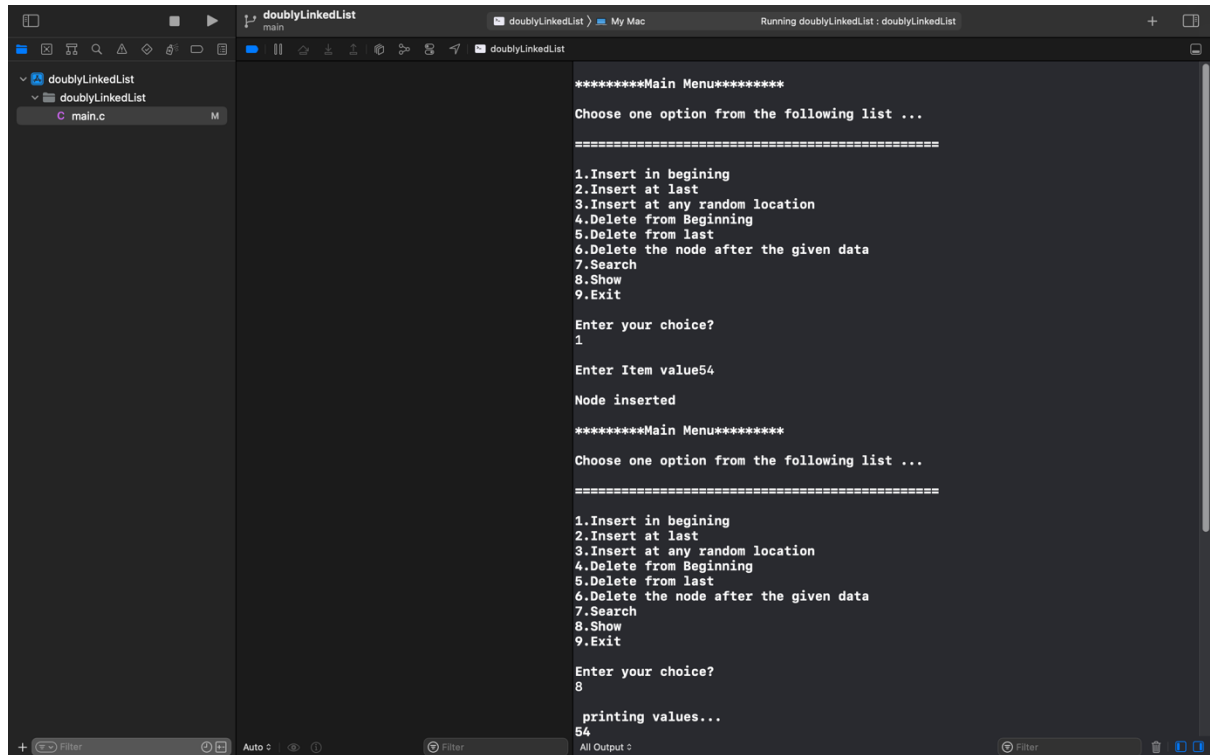


```
250     }
251 }
252 void search()
253 {
254     struct node *ptr;
255     int item,i=0,flag = 0;
256     ptr = head;
257     if(ptr == NULL)
258     {
259         printf("\nEmpty List\n");
260     }
261     else
262     {
263         printf("\nEnter item which you want to search?\n");
264         scanf("%d",&item);
265         while (ptr!=NULL)
266         {
267             if(ptr->data == item)
268             {
269                 printf("\nitem found at location %d ",i+1);
270                 flag=0;
271                 break;
272             }
273             else
274             {
275                 flag=1;
276             }
277             i++;
278             ptr = ptr -> next;
279         }
280         if(flag==1)
281         {
282             printf("\nItem not found\n");
283         }
284     }
285 }
286 }
```

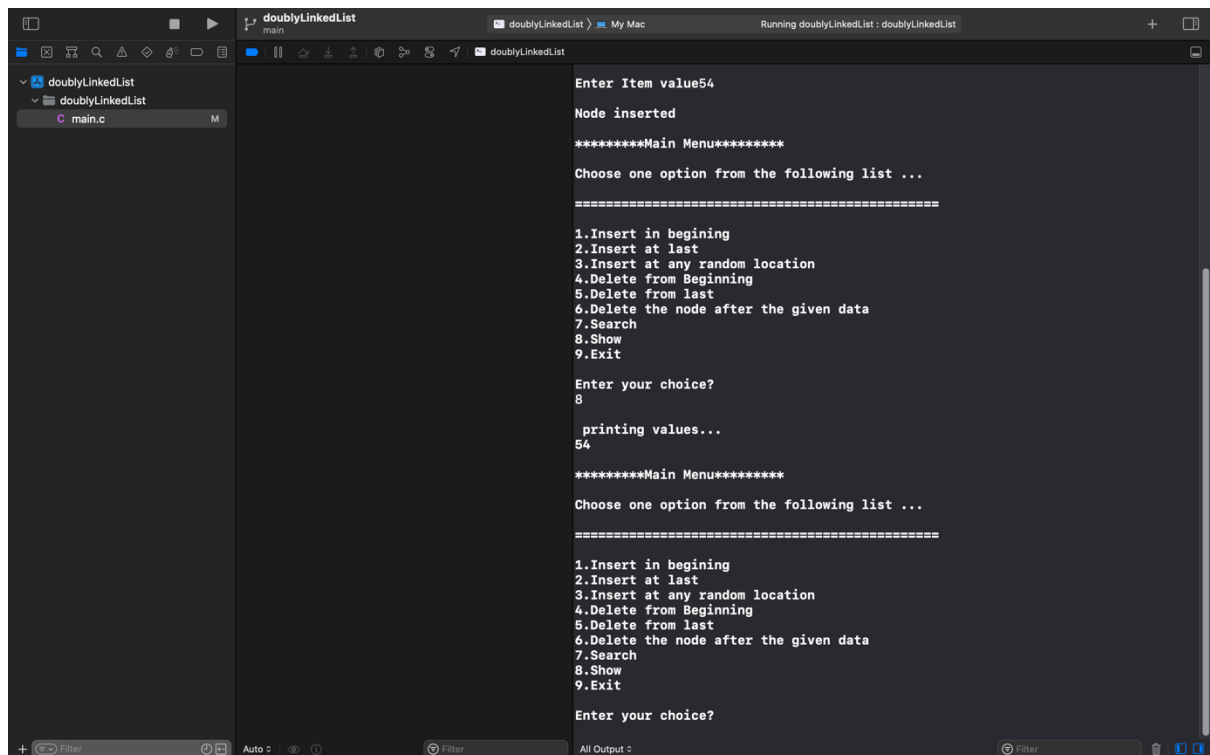
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3



```
*****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 the node after the given data
7.Search
8.Show
9.Exit
Enter your choice?
1
Enter Item value54
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 the node after the given data
7.Search
8.Show
9.Exit
Enter your choice?
8
printing values...
54
```



```
Enter Item value54
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 the node after the given data
7.Search
8.Show
9.Exit
Enter your choice?
8
printing values...
54
*****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 the node after the given data
7.Search
8.Show
9.Exit
Enter your choice?
```

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