Data Structures using C

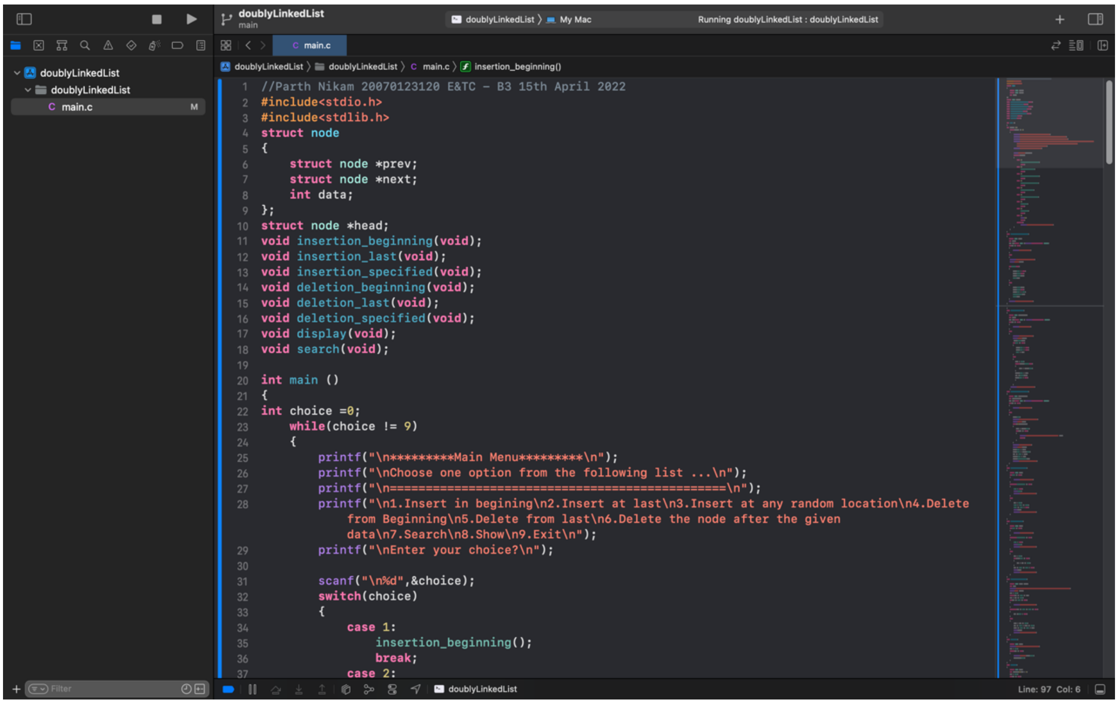
15th April 2022

Parth Nikam
20070123120
E&TC – B3

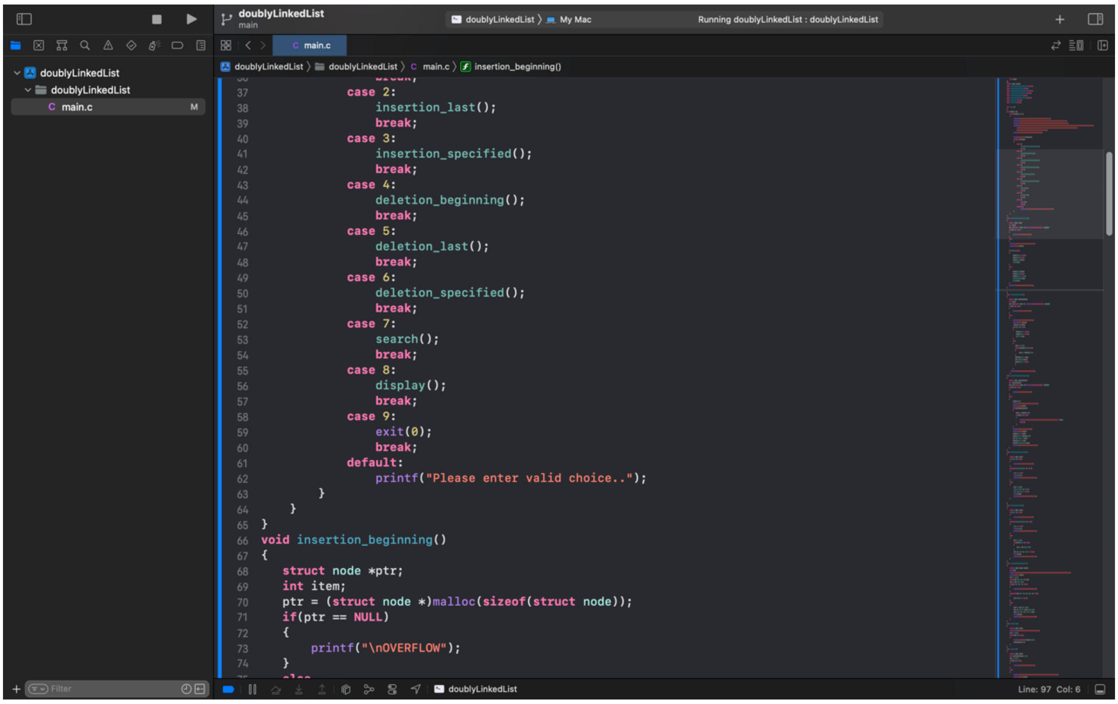
Aim: - Studying and coding on Doubly Linked List.

Objective: - To perform Doubly Linked List in C language

Code: -


doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c
M
Filter
doublyLinkedList
Line: 97 Col: 6

```c
//Parth Nikam 20070123120 E&TC - B3 15th April 2022
#include<stdio.h>
#include<stdlib.h>
struct node
{
    struct node *prev;
    struct node *next;
    int data;
};
struct node *head;
void insertion_beginning(void);
void insertion_last(void);
void insertion_specified(void);
void deletion_beginning(void);
void deletion_last(void);
void deletion_specified(void);
void display(void);
void search(void);

int main ()
{
int choice =0;
    while(choice != 9)
    {
        printf("\n*********Main Menu*********\n");
        printf("\nChoose one option from the following list ...\n");
        printf("\n===============================================\n");
        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 the node after the given data\n7.Search\n8.Show\n9.Exit\n");
        printf("\nEnter your choice?\n");

        scanf("\n%d",&choice);
        switch(choice)
        {
            case 1:
                insertion_beginning();
                break;
            case 2:
```

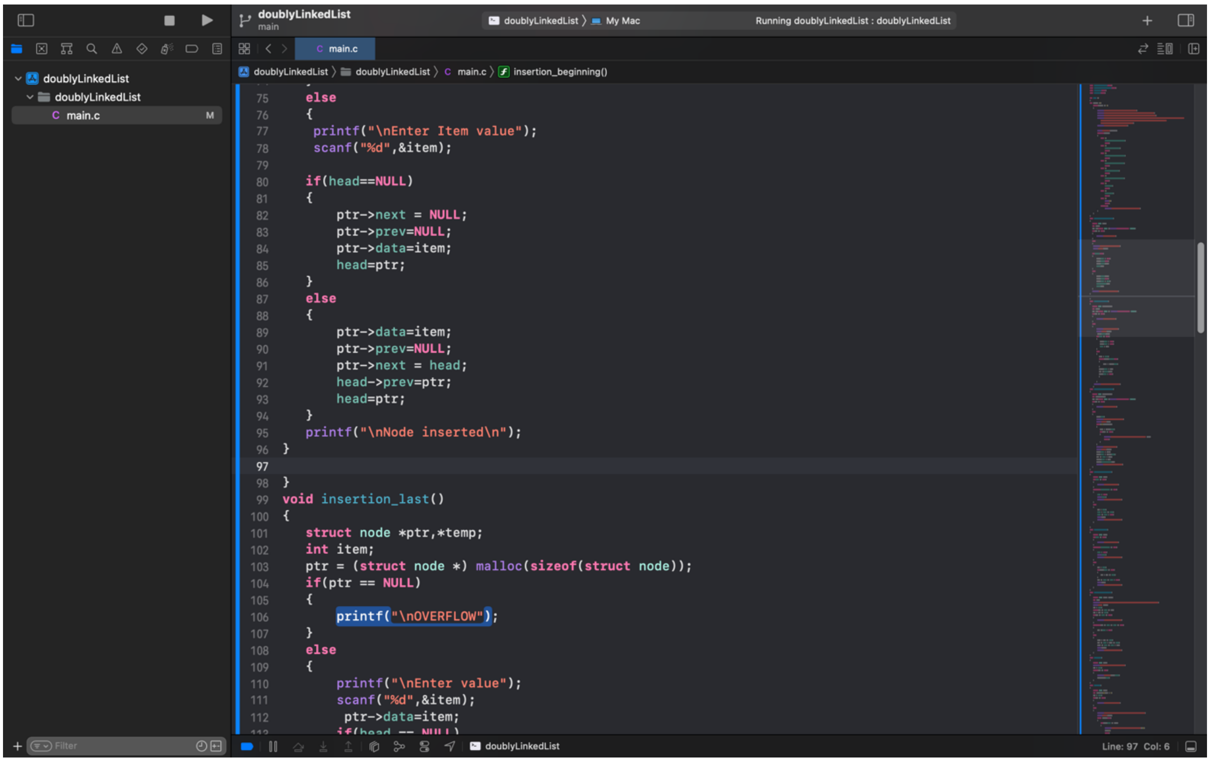

doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c
M
Filter
doublyLinkedList
Line: 97 Col: 6

```c
                break;
            case 2:
                insertion_last();
                break;
            case 3:
                insertion_specified();
                break;
            case 4:
                deletion_beginning();
                break;
            case 5:
                deletion_last();
                break;
            case 6:
                deletion_specified();
                break;
            case 7:
                search();
                break;
            case 8:
                display();
                break;
            case 9:
                exit(0);
                break;
            default:
                printf("Please enter valid choice..");
        }
    }
}
void insertion_beginning()
{
   struct node *ptr;
   int item;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
   }
   else
```

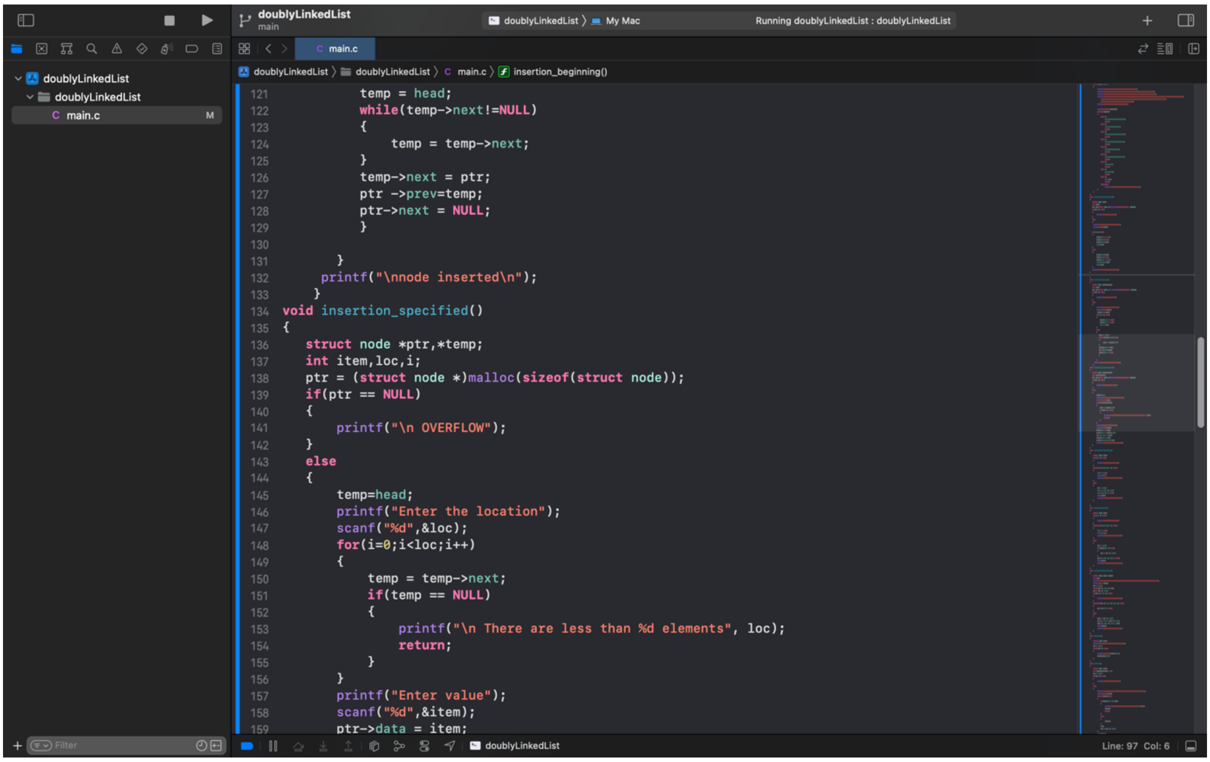
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3


doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c
M
Filter
doublyLinkedList
Line: 97 Col: 6

```c
   else
   {
    printf("\nEnter Item value");
    scanf("%d",&item);

   if(head==NULL)
   {
       ptr->next = NULL;
       ptr->prev=NULL;
       ptr->data=item;
       head=ptr;
   }
   else
   {
       ptr->data=item;
       ptr->prev=NULL;
       ptr->next = head;
       head->prev=ptr;
       head=ptr;
   }
   printf("\nNode inserted\n");
}

}
void insertion_last()
{
   struct node *ptr,*temp;
   int item;
   ptr = (struct node *) malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
   }
   else
   {
       printf("\nEnter value");
       scanf("%d",&item);
        ptr->data=item;
       if(head == NULL)
```

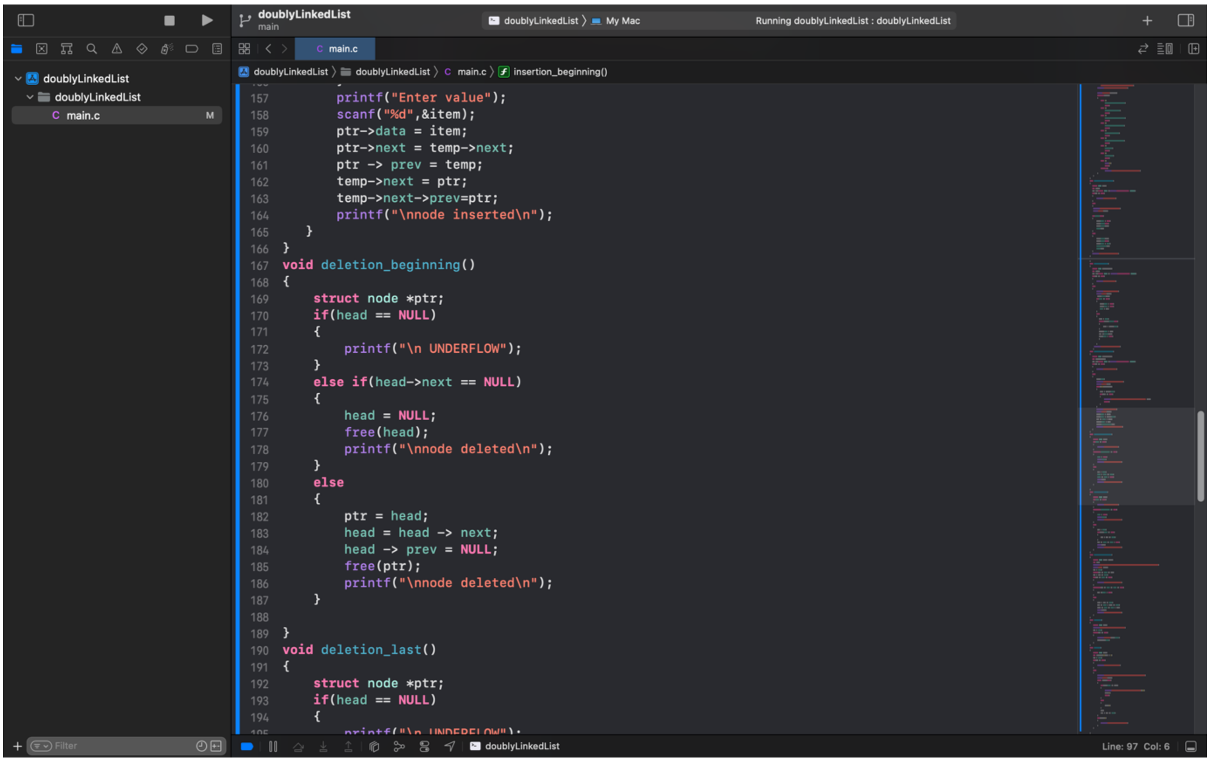

doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c
M
Filter
doublyLinkedList
Line: 97 Col: 6

```c
          temp = head;
          while(temp->next!=NULL)
          {
              temp = temp->next;
          }
          temp->next = ptr;
          ptr ->prev=temp;
          ptr->next = NULL;
          }

       }
     printf("\nnode inserted\n");
    }
void insertion_specified()
{
   struct node *ptr,*temp;
   int item,loc,i;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\n OVERFLOW");
   }
   else
   {
       temp=head;
       printf("Enter the location");
       scanf("%d",&loc);
       for(i=0;i<loc;i++)
       {
           temp = temp->next;
           if(temp == NULL)
           {
               printf("\n There are less than %d elements", loc);
               return;
           }
       }
       printf("Enter value");
       scanf("%d",&item);
       ptr->data = item;
```

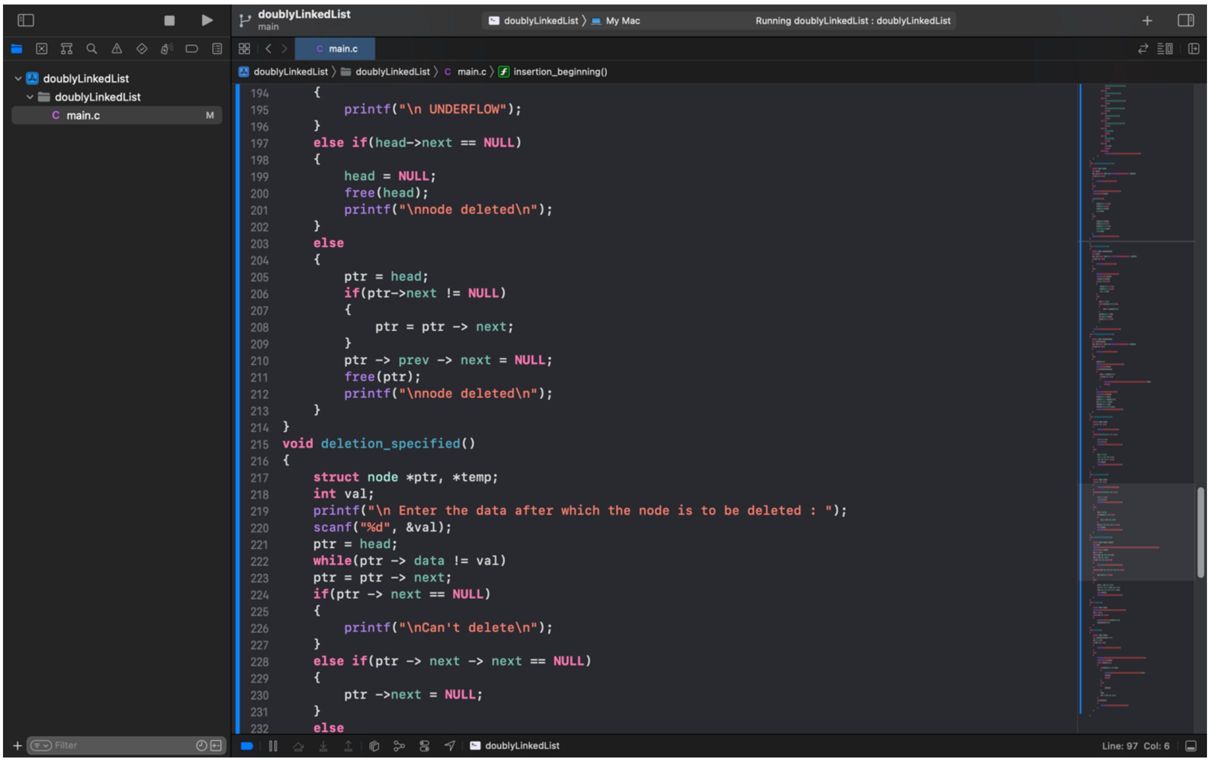
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3


doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c M
Filter
doublyLinkedList
Line: 97 Col: 6

```
       printf("Enter value");
       scanf("%d",&item);
       ptr->data = item;
       ptr->next = temp->next;
       ptr -> prev = temp;
       temp->next = ptr;
       temp->next->prev=ptr;
       printf("\nnode inserted\n");
   }
}
void deletion_beginning()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\n UNDERFLOW");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nnode deleted\n");
    }
    else
    {
        ptr = head;
        head = head -> next;
        head -> prev = NULL;
        free(ptr);
        printf("\nnode deleted\n");
    }

}
void deletion_last()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\n UNDERFLOW");
```

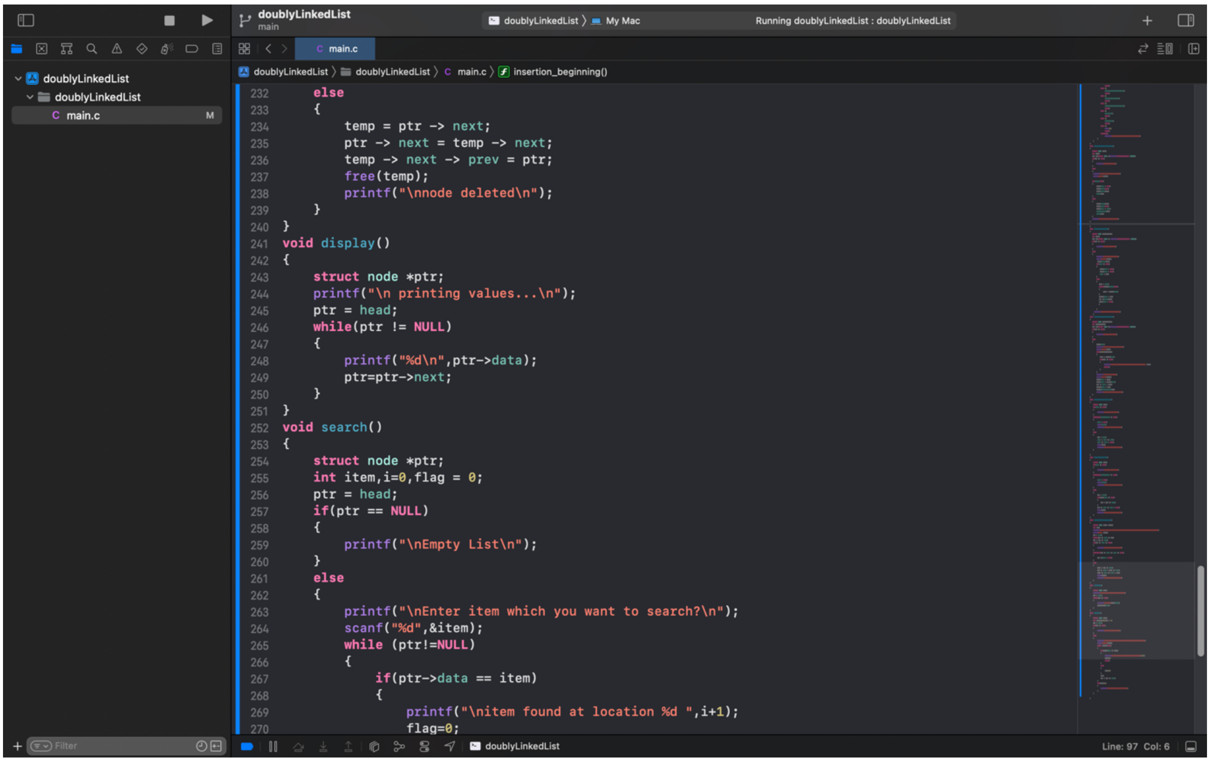

doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
doublyLinkedList
doublyLinkedList
main.c M
Filter
doublyLinkedList
Line: 97 Col: 6

```
    {
        printf("\n UNDERFLOW");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nnode deleted\n");
    }
    else
    {
        ptr = head;
        if(ptr->next != NULL)
        {
            ptr = ptr -> next;
        }
        ptr -> prev -> next = NULL;
        free(ptr);
        printf("\nnode deleted\n");
    }
}
void deletion_specified()
{
    struct node *ptr, *temp;
    int val;
    printf("\n Enter the data after which the node is to be deleted : ");
    scanf("%d", &val);
    ptr = head;
    while(ptr -> data != val)
    ptr = ptr -> next;
    if(ptr -> next == NULL)
    {
        printf("\nCan't delete\n");
    }
    else if(ptr -> next -> next == NULL)
    {
        ptr ->next = NULL;
    }
    else
```

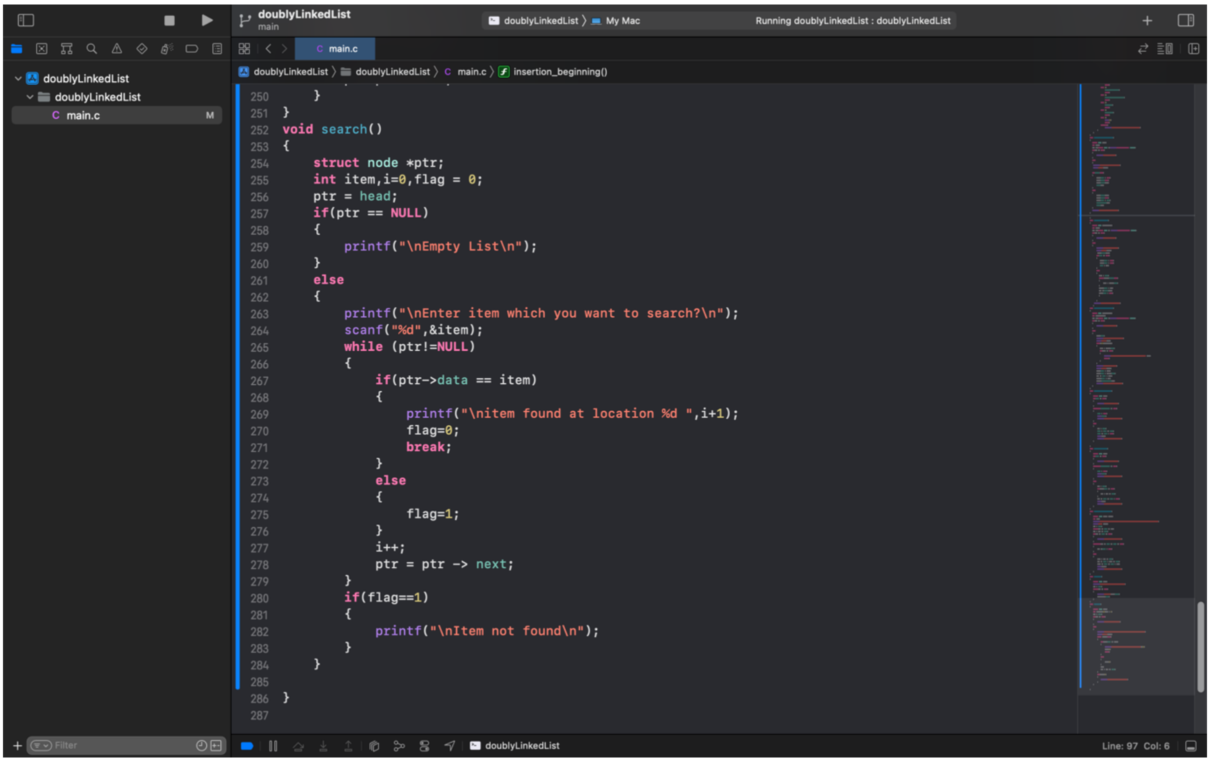
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3


doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList
doublyLinkedList
main.c
M
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
Filter
doublyLinkedList
Line: 97 Col: 6

```
    else
    {
        temp = ptr -> next;
        ptr -> next = temp -> next;
        temp -> next -> prev = ptr;
        free(temp);
        printf("\nnode deleted\n");
    }
}
void display()
{
    struct node *ptr;
    printf("\n printing values...\n");
    ptr = head;
    while(ptr != NULL)
    {
        printf("%d\n",ptr->data);
        ptr=ptr->next;
    }
}
void search()
{
    struct node *ptr;
    int item,i=0,flag = 0;
    ptr = head;
    if(ptr == NULL)
    {
        printf("\nEmpty List\n");
    }
    else
    {
        printf("\nEnter item which you want to search?\n");
        scanf("%d",&item);
        while (ptr!=NULL)
        {
            if(ptr->data == item)
            {
                printf("\nitem found at location %d ",i+1);
                flag=0;
```

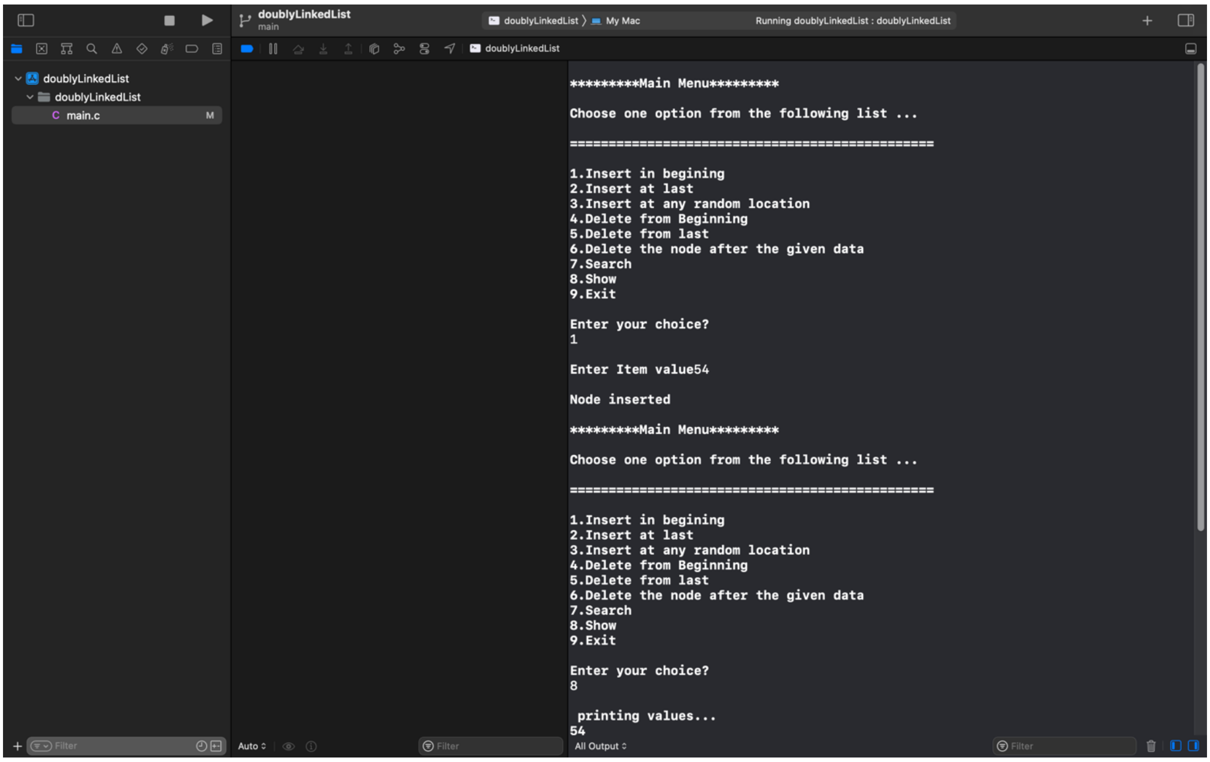

doublyLinkedList
main
doublyLinkedList 〉 My Mac
Running doublyLinkedList : doublyLinkedList
main.c
doublyLinkedList
doublyLinkedList
main.c
M
doublyLinkedList 〉 doublyLinkedList 〉 main.c 〉 insertion_beginning()
Filter
doublyLinkedList
Line: 97 Col: 6

```
    }
}
void search()
{
    struct node *ptr;
    int item,i=0,flag = 0;
    ptr = head;
    if(ptr == NULL)
    {
        printf("\nEmpty List\n");
    }
    else
    {
        printf("\nEnter item which you want to search?\n");
        scanf("%d",&item);
        while (ptr!=NULL)
        {
            if(ptr->data == item)
            {
                printf("\nitem found at location %d ",i+1);
                flag=0;
                break;
            }
            else
            {
                flag=1;
            }
            i++;
            ptr = ptr -> next;
        }
        if(flag==1)
        {
            printf("\nItem not found\n");
        }
    }

}

```

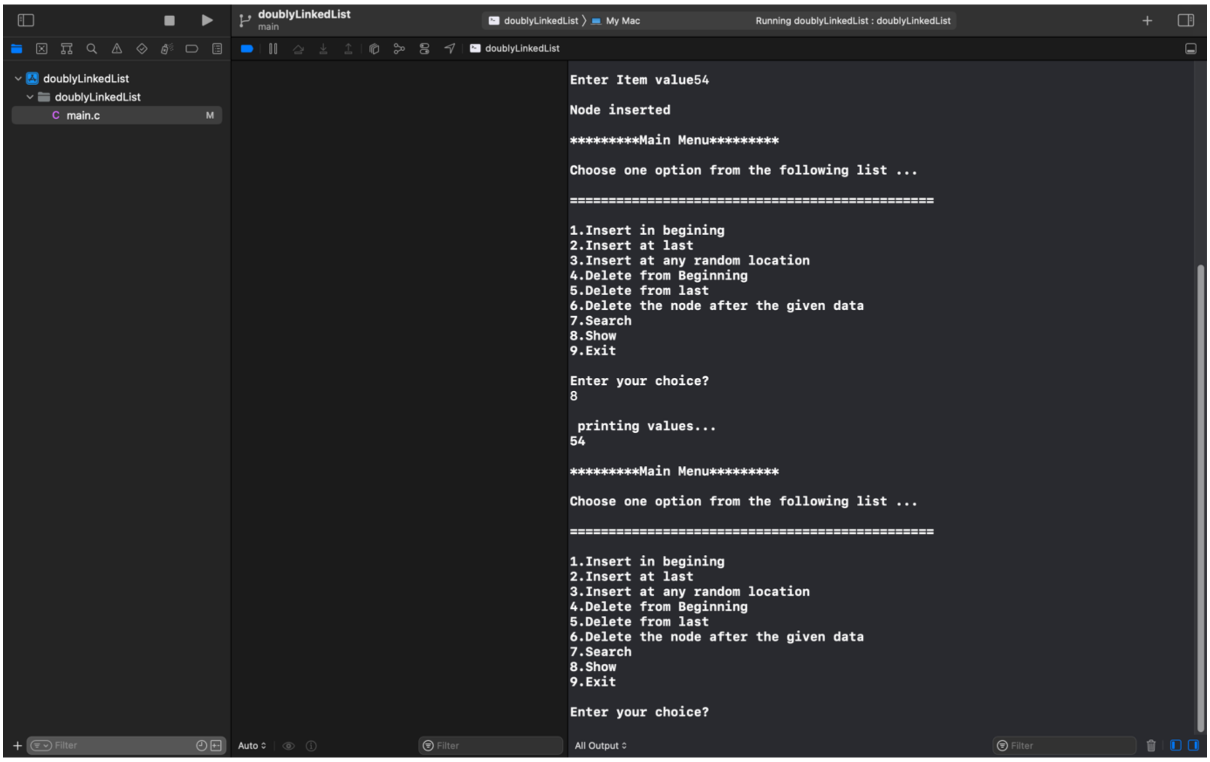
Data Structures using C

15th April 2022

Parth Nikam
20070123120
E&TC – B3



```
*********Main Menu*********

Choose one option from the following list ...

===============================================

1.Insert in begining
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 begining
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 begining
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 begining
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.