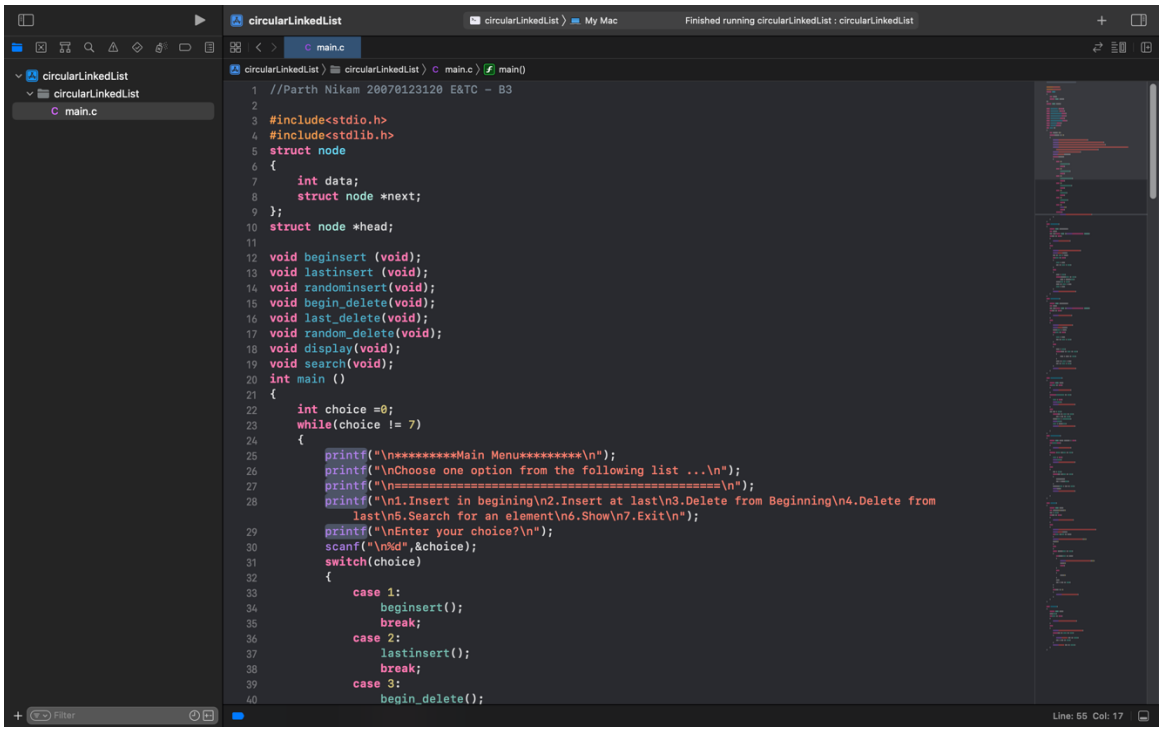
Data Structures using C

Parth Nikam
20070123120
E&TC – B3

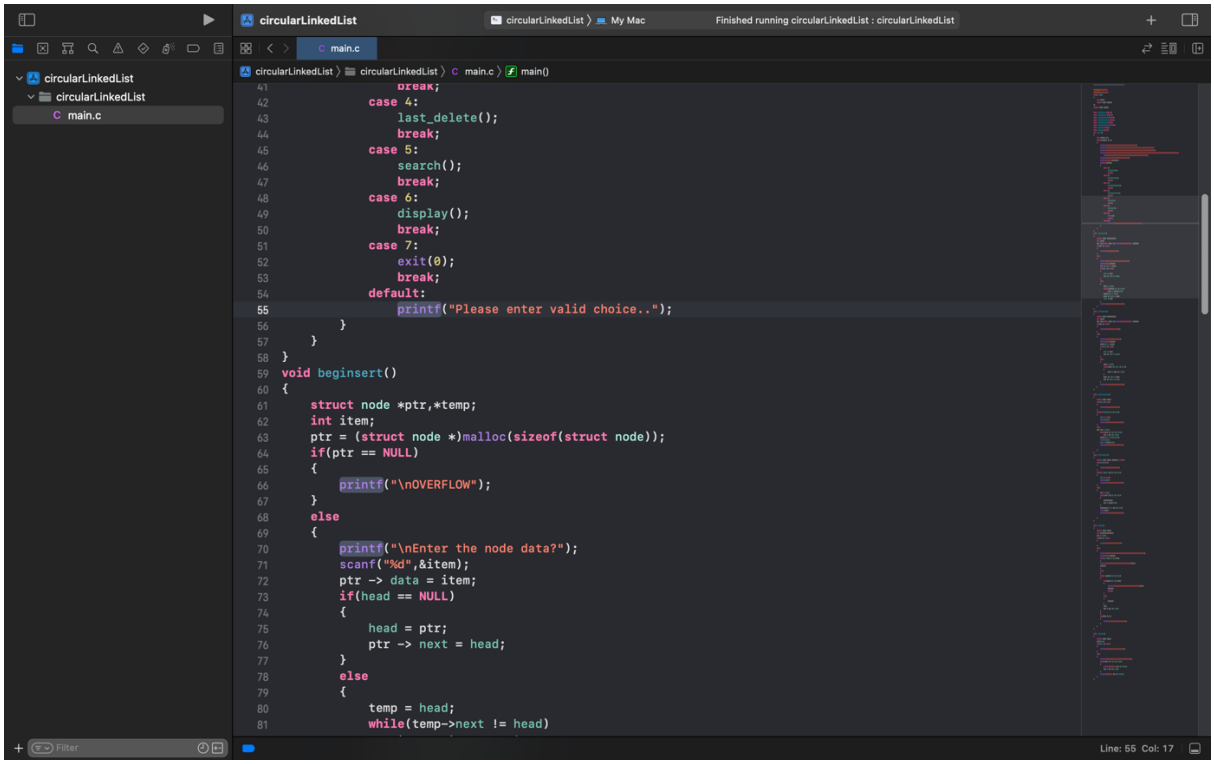
Aim: - Studying and coding on Circular Linked List.

Objective: - To perform Circular Linked List in C language

Code: -



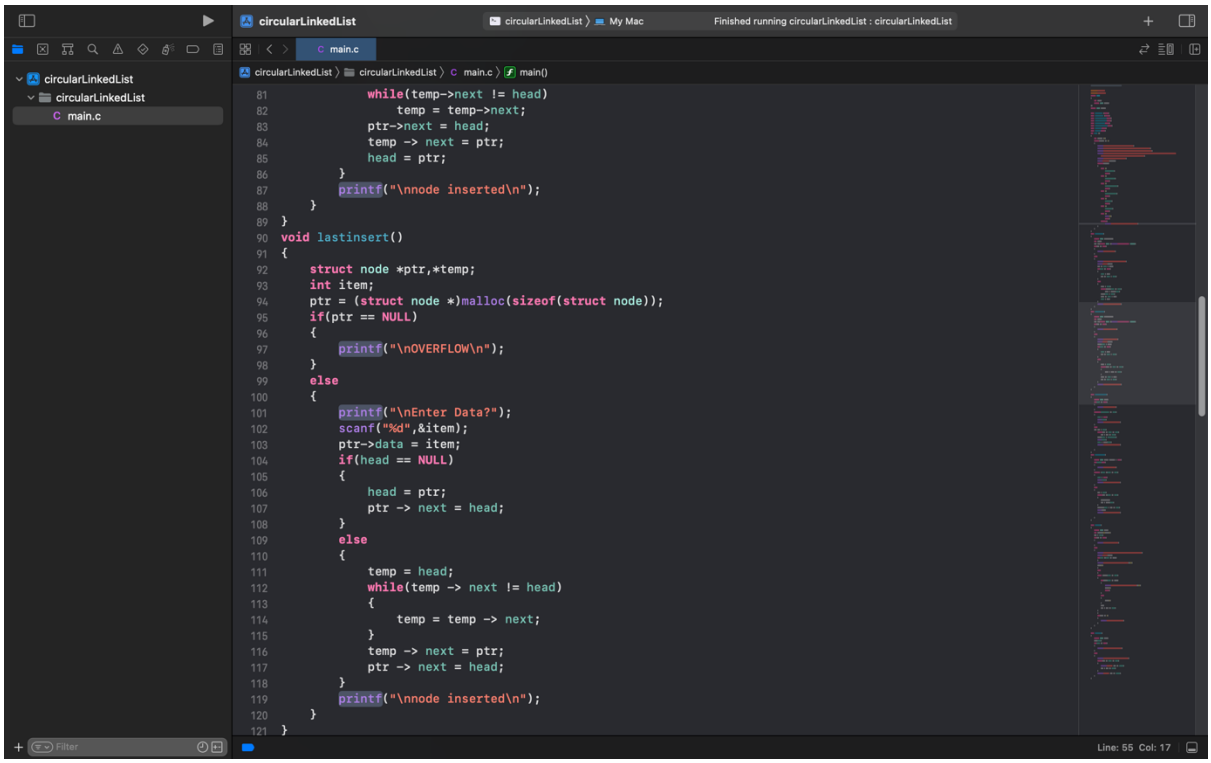
```c
//Parth Nikam 20070123120 E&TC – B3

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head;

void beginsert (void);
void lastinsert (void);
void randominsert(void);
void begin_delete(void);
void last_delete(void);
void random_delete(void);
void display(void);
void search(void);
int main ()
{
    int choice =0;
    while(choice != 7)
    {
        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.Delete from Beginning\n4.Delete from
            last\n5.Search for an element\n6.Show\n7.Exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
            case 1:
                beginsert();
                break;
            case 2:
                lastinsert();
                break;
            case 3:
                begin_delete();
```



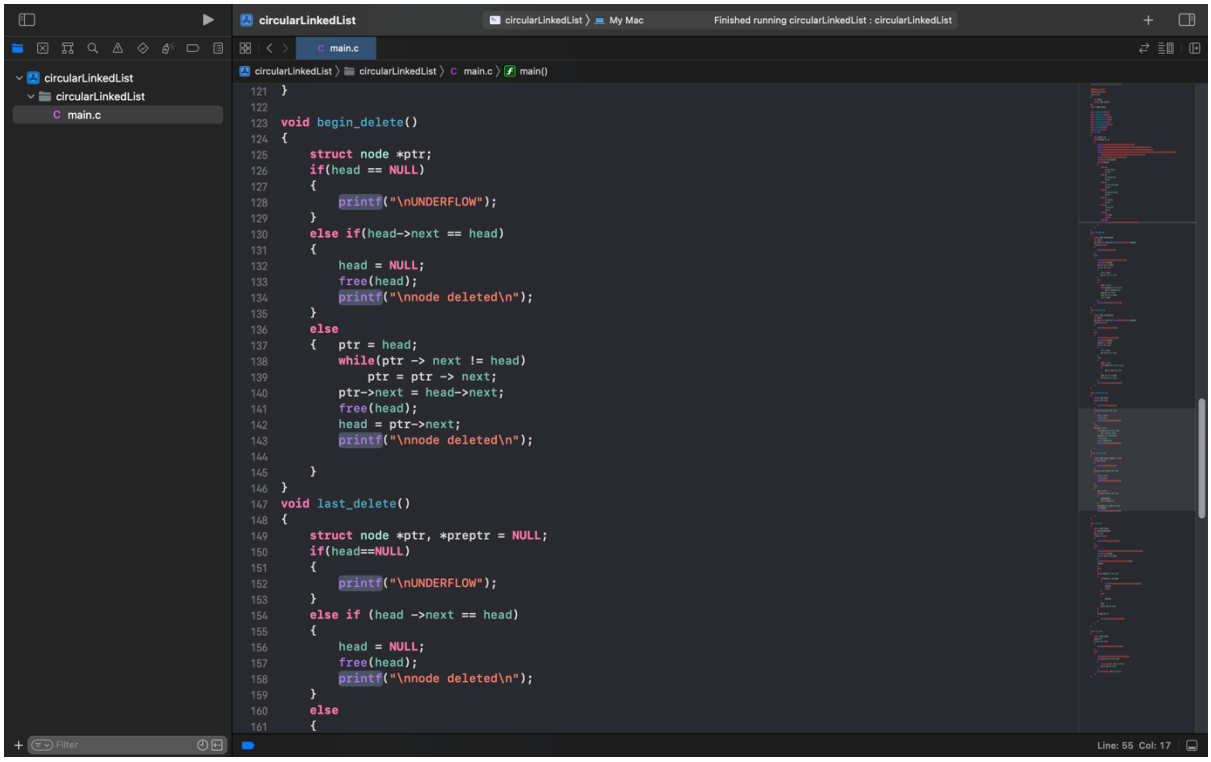
```c
                break;
            case 4:
                last_delete();
                break;
            case 5:
                search();
                break;
            case 6:
                display();
                break;
            case 7:
                exit(0);
                break;
            default:
                printf("Please enter valid choice..");
        }
    }
}
void beginsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter the node data?");
        scanf("%d",&item);
        ptr -> data = item;
        if(head == NULL)
        {
            head = ptr;
            ptr -> next = head;
        }
        else
        {
            temp = head;
            while(temp->next != head)
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



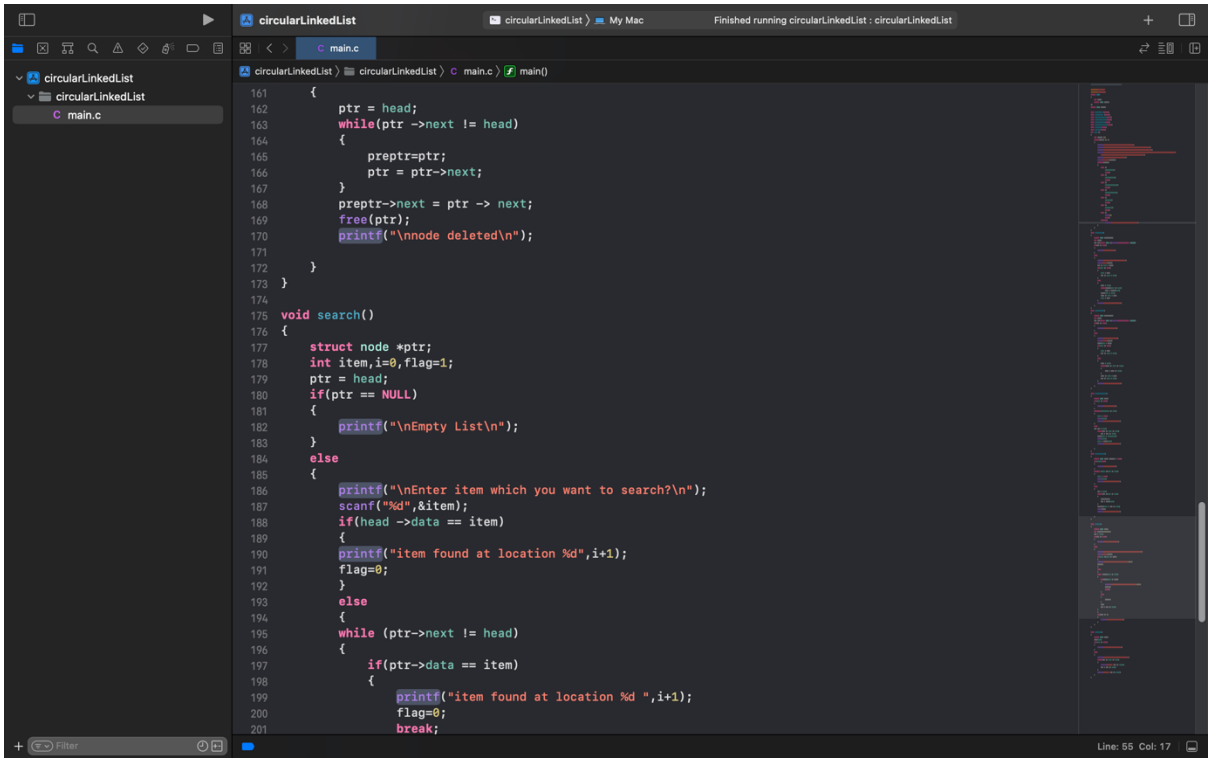
```
            while(temp->next != head)
                temp = temp->next;
            ptr->next = head;
            temp -> next = ptr;
            head = ptr;
        }
        printf("\nnode inserted\n");
    }
}
void lastinsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node *)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");
    }
}
```



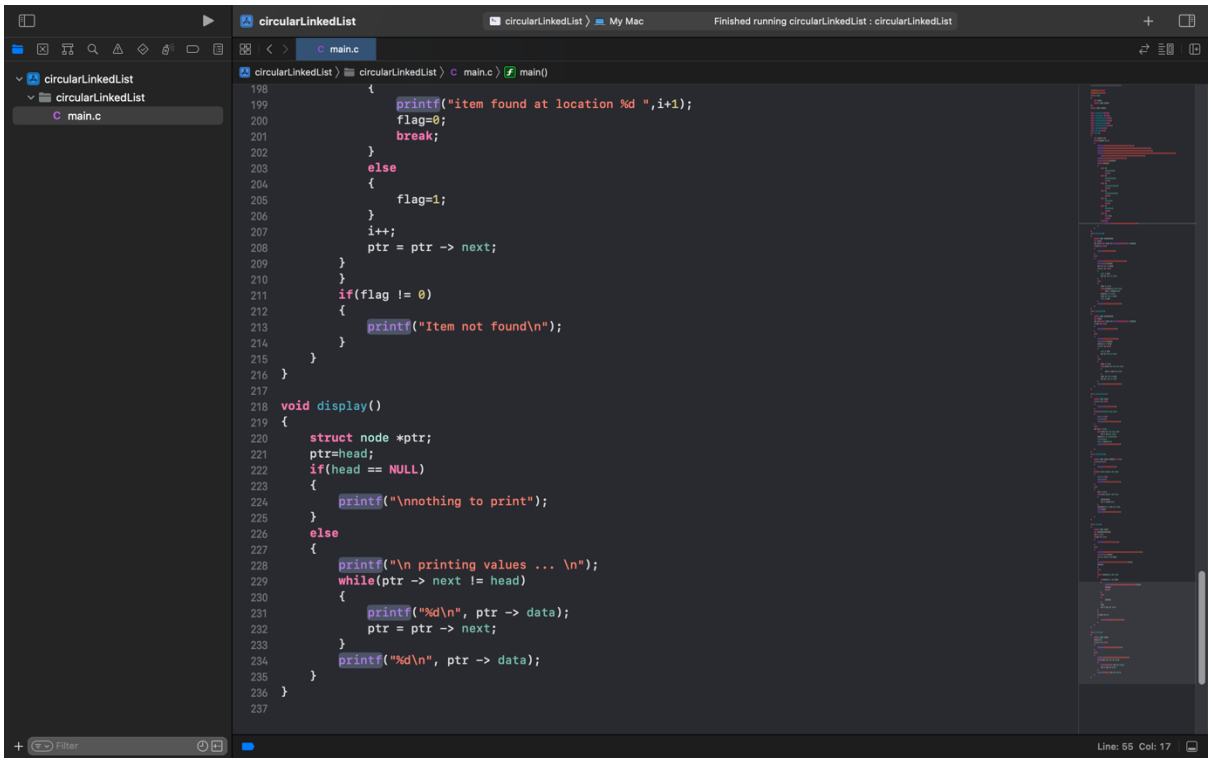
```
}

void begin_delete()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\nUNDERFLOW");
    }
    else if(head->next == head)
    {
        head = NULL;
        free(head);
        printf("\nnode deleted\n");
    }
    else
    {   ptr = head;
        while(ptr -> next != head)
            ptr = ptr -> next;
        ptr->next = head->next;
        free(head);
        head = ptr->next;
        printf("\nnode deleted\n");

    }
}
void last_delete()
{
    struct node *ptr, *preptr = NULL;
    if(head==NULL)
    {
        printf("\nUNDERFLOW");
    }
    else if (head ->next == head)
    {
        head = NULL;
        free(head);
        printf("\nnode deleted\n");
    }
    else
    {
```

Data Structures using C

Parth Nikam
20070123120
E&TC – B3



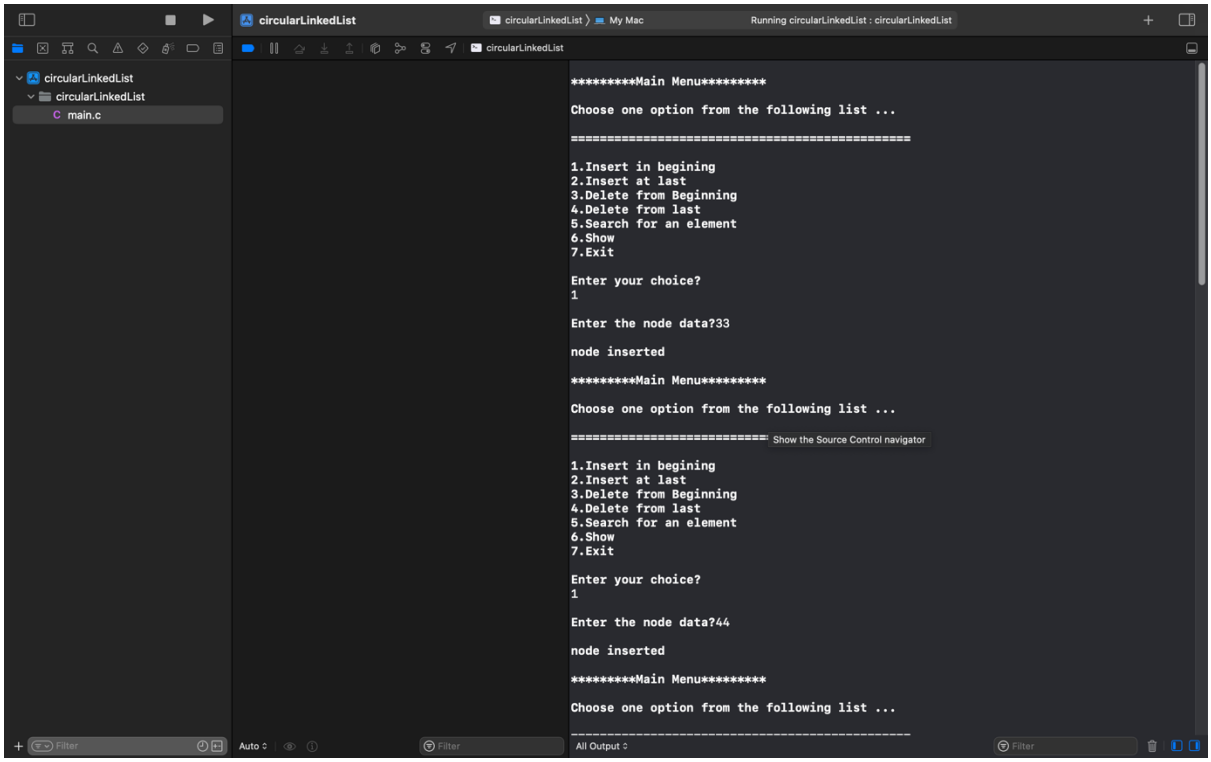
```
    {
        ptr = head;
        while(ptr ->next != head)
        {
            preptr=ptr;
            ptr = ptr->next;
        }
        preptr->next = ptr -> next;
        free(ptr);
        printf("\nnode deleted\n");

    }
}

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

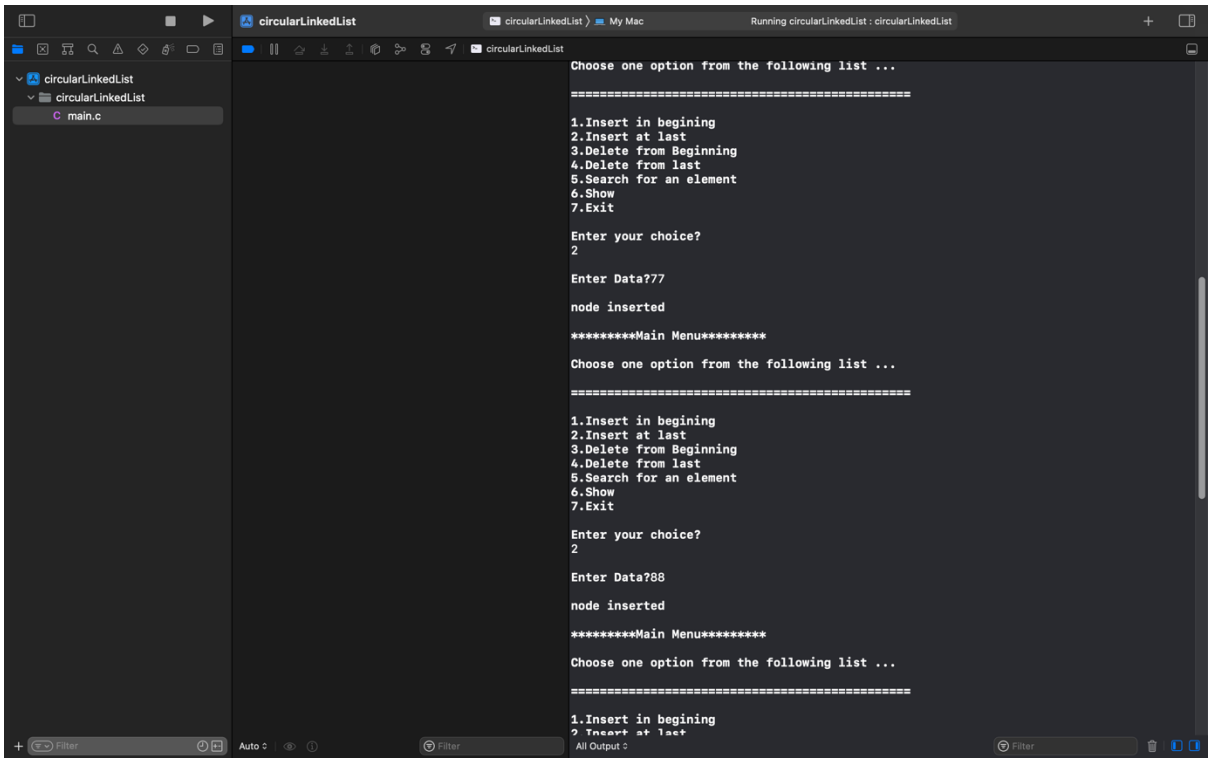


```
                {
                    printf("item found at location %d ",i+1);
                    flag=0;
                    break;
                }
                else
                {
                    flag=1;
                }
                i++;
                ptr = ptr -> next;
            }
            }
            if(flag != 0)
            {
                printf("Item not found\n");
            }
        }
}

void display()
{
    struct node *ptr;
    ptr=head;
    if(head == NULL)
    {
        printf("\nnothing to print");
    }
    else
    {
        printf("\n printing values ... \n");
        while(ptr -> next != head)
        {
            printf("%d\n", ptr -> data);
            ptr = ptr -> next;
        }
        printf("%d\n", ptr -> data);
    }
}
```

Data Structures using C

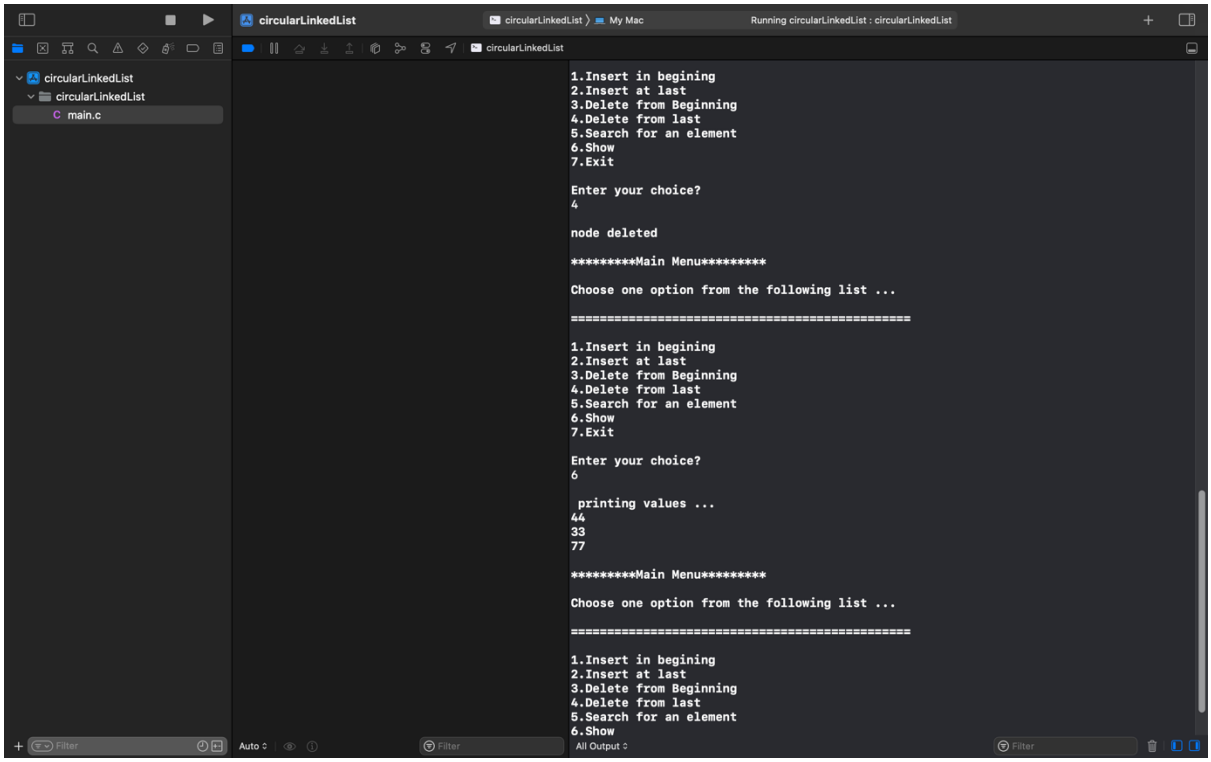
Parth Nikam
20070123120
E&TC – B3

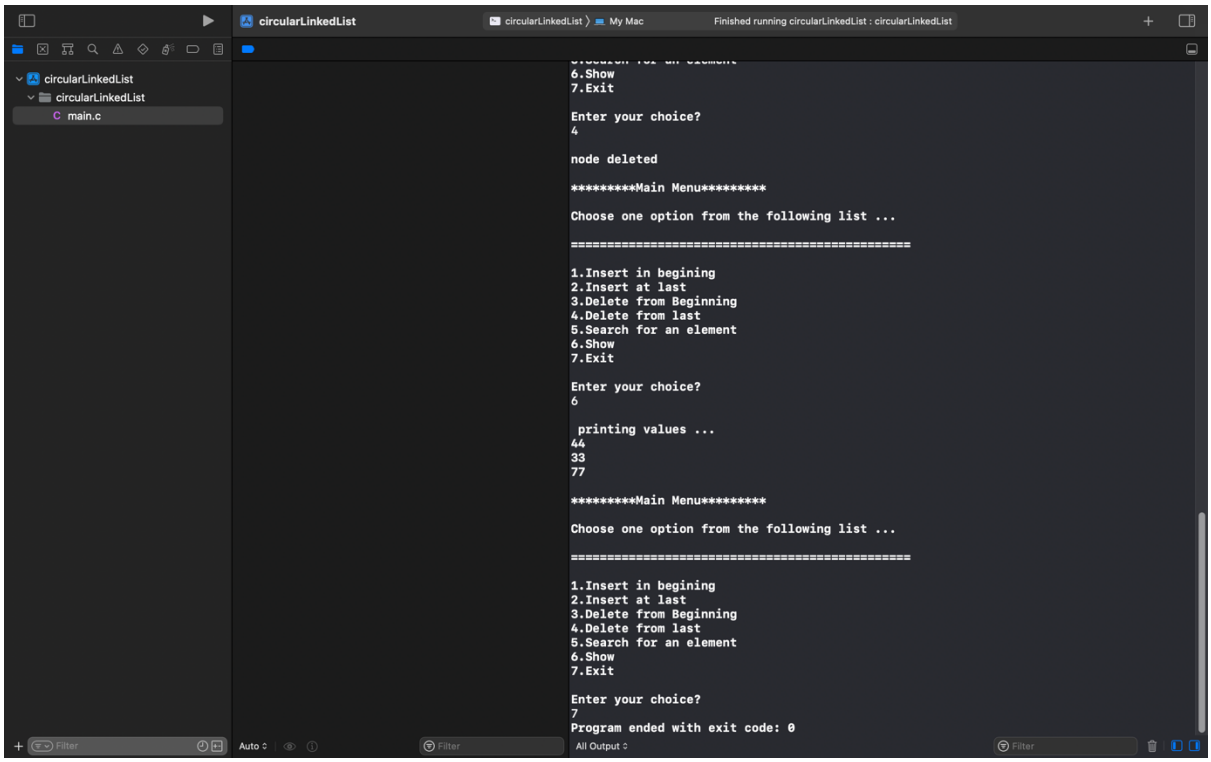




Data Structures using C

Parth Nikam
20070123120
E&TC – B3





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