# DSA Assignment: 7

**Exp 7:** Implementation of Circular Singly Linked List

Shashwat Tripathi

D10A   Roll No: 60

**AIM:** In this experiment, we will implement circular singly linked list.

**CODE:**

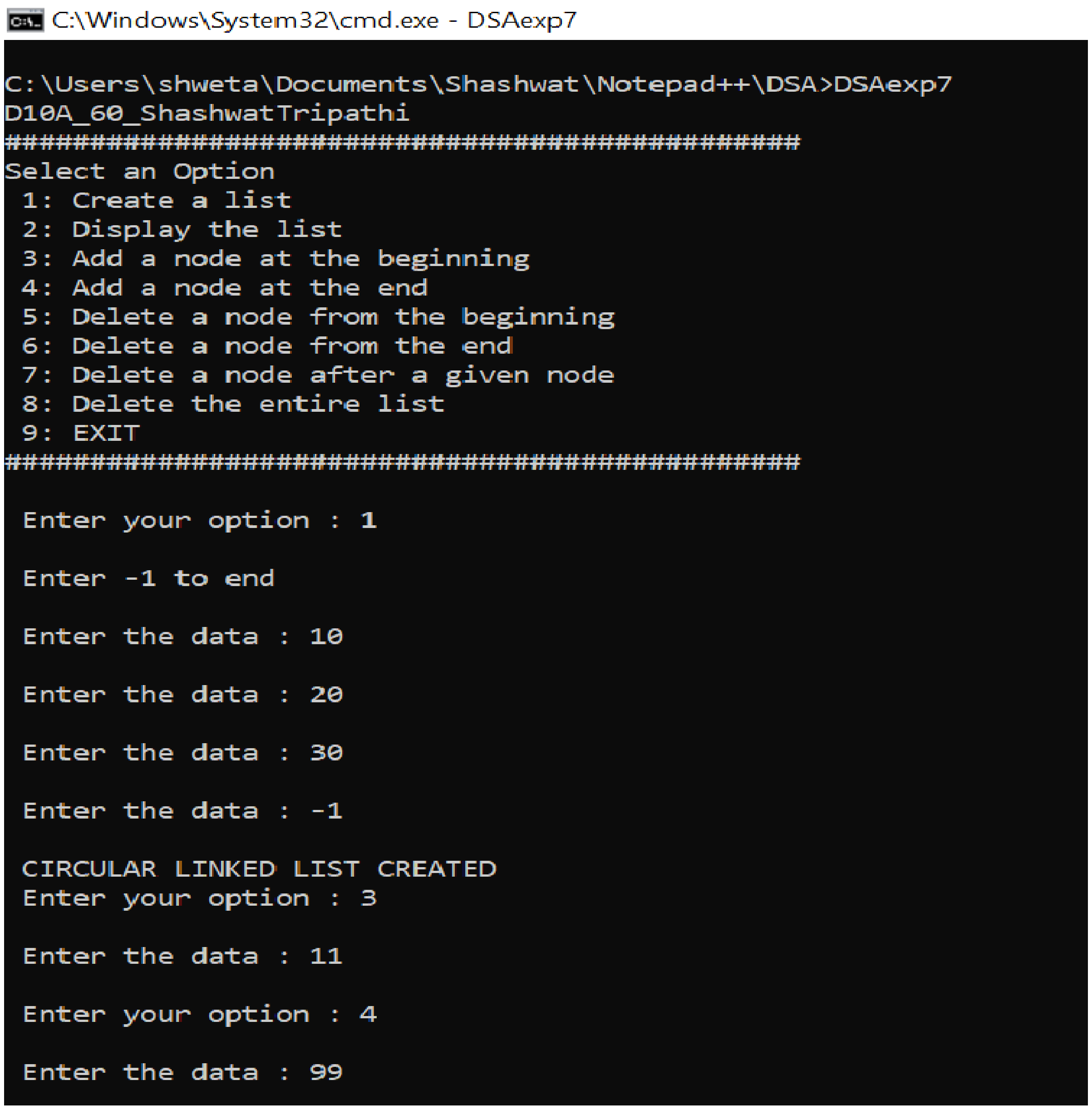
```
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node *next;
};
struct node *start = NULL;
struct node *create_cll(struct node *);
struct node *display(struct node *);
struct node *insert_beg(struct node *);
struct node *insert_end(struct node *);
struct node *delete_beg(struct node *);
struct node *delete_end(struct node *);
struct node *delete_after(struct node *);
struct node *delete_list(struct node *);

int main()
{

    int option;
    printf("D10A_60_ShashwatTripathi");
    printf("\n################################################\n");
    printf("Select an Option");
    printf("\n 1: Create a list");
    printf("\n 2: Display the list");
    printf("\n 3: Add a node at the beginning");
    printf("\n 4: Add a node at the end");
    printf("\n 5: Delete a node from the beginning");
    printf("\n 6: Delete a node from the end");
    printf("\n 7: Delete a node after a given node");
    printf("\n 8: Delete the entire list");
    printf("\n 9: EXIT");
    printf("\n################################################\n");

    do
    {
        printf("\n Enter your option : ");
        scanf("%d", &option);
        switch (option)
        {
        case 1:
            start = create_cll(start);
            printf("\n CIRCULAR LINKED LIST CREATED");
            break;
```

```c
            case 2:
                start = display(start);
                break;
            case 3:
                start = insert_beg(start);
                break;
            case 4:
                start = insert_end(start);
                break;
            case 5:
                start = delete_beg(start);
                break;
            case 6:
                start = delete_end(start);
                break;
            case 7:
                start = delete_after(start);
                break;
            case 8:
                start = delete_list(start);
                printf("\n CIRCULAR LINKED LIST DELETED");
                break;
            }
        } while (option != 9);
        return 0;
}

struct node *create_cll(struct node *start)
{
    struct node *new_node, *ptr;
    int num;
    printf("\n Enter -1 to end\n");
    printf("\n Enter the data : ");
    scanf("%d", &num);
    while (num != -1)
    {
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = num;
        if (start == NULL)
        {
            new_node->next = new_node;
            start = new_node;
        }
        else
        {
            ptr = start;
            while (ptr->next != start)
                ptr = ptr->next;
            ptr->next = new_node;
            new_node->next = start;
        }
        printf("\n Enter the data : ");
        scanf("%d", &num);
    }
    return start;
}
```

```c
struct node *display(struct node *start)
{
    struct node *ptr;
    ptr = start;
    while (ptr->next != start)
    {
        printf("\t %d", ptr->data);
        ptr = ptr->next;
    }
    printf("\t %d", ptr->data);
    return start;
}

struct node *insert_beg(struct node *start)
{
    struct node *new_node, *ptr;
    int num;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    new_node = (struct node *)malloc(sizeof(struct node));
    new_node->data = num;
    ptr = start;
    while (ptr->next != start)
        ptr = ptr->next;
    ptr->next = new_node;
    new_node->next = start;
    start = new_node;
    return start;
}

struct node *insert_end(struct node *start)
{
    struct node *ptr, *new_node;
    int num;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    new_node = (struct node *)malloc(sizeof(struct node));
    new_node->data = num;
    ptr = start;
    while (ptr->next != start)
        ptr = ptr->next;
    ptr->next = new_node;
    new_node->next = start;
    return start;
}

struct node *delete_beg(struct node *start)
{
    struct node *ptr;
    ptr = start;
    while (ptr->next != start)
        ptr = ptr->next;
    ptr->next = start->next;
    free(start);
    start = ptr->next;
    return start;
}
```

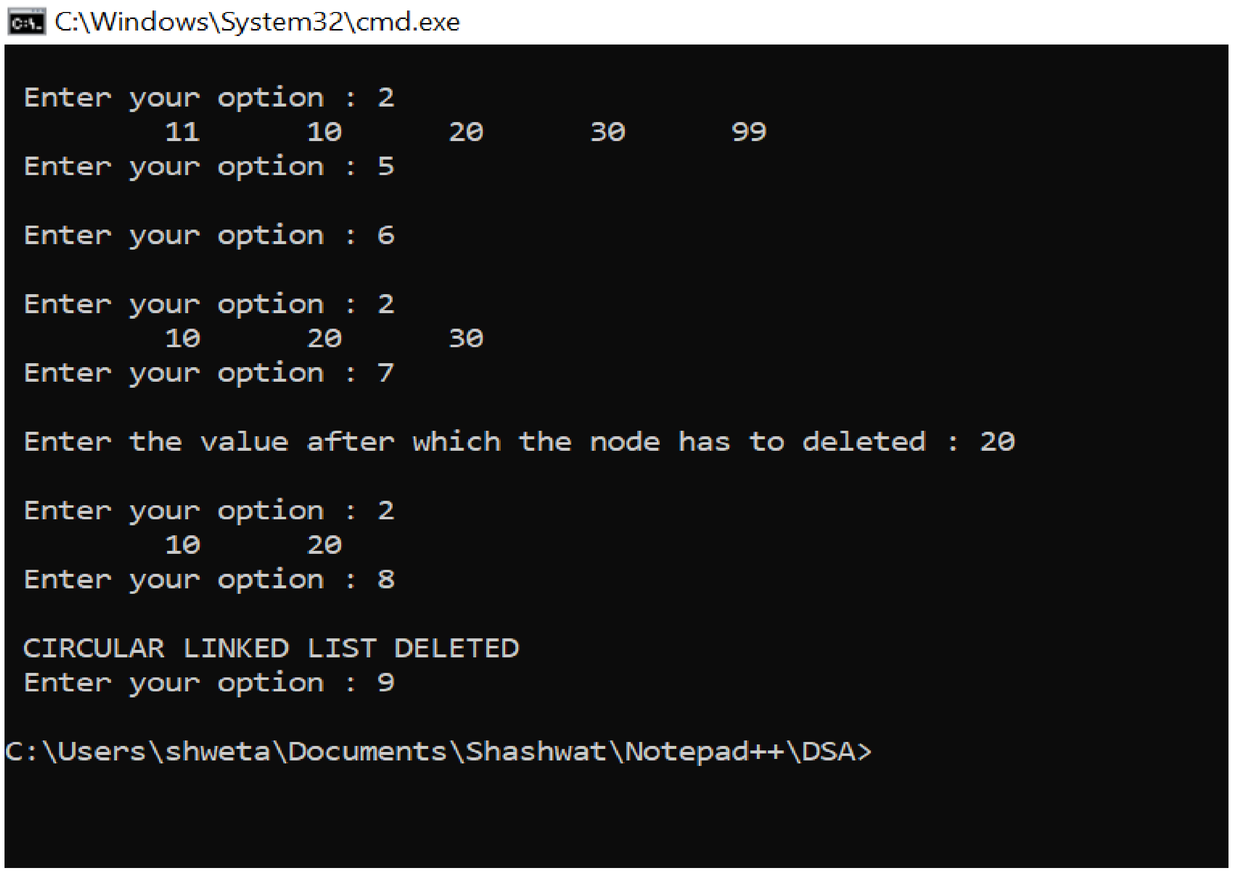
```
struct node *delete_end(struct node *start)
{
    struct node *ptr, *preptr;
    ptr = start;
    while (ptr->next != start)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = ptr->next;
    free(ptr);
    return start;
}

struct node *delete_after(struct node *start)
{
    struct node *ptr, *preptr;
    int val;
    printf("\n Enter the value after which the node has to deleted : ");
    scanf("%d", &val);
    ptr = start;
    preptr = ptr;
    while (preptr->data != val)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = ptr->next;
    if (ptr == start)
        start = preptr->next;
    free(ptr);
    return start;
}

struct node *delete_list(struct node *start)
{
    struct node *ptr;
    ptr = start;
    while (ptr->next != start)
        start = delete_end(start);
    free(start);
    return start;
}
```

**OUTPUT:**



C:\Windows\System32\cmd.exe - DSAexp7

```
C:\Users\shweta\Documents\Shashwat\Notepad++\DSA>DSAexp7
D10A_60_ShashwatTripathi
###############################################
Select an Option
 1: Create a list
 2: Display the list
 3: Add a node at the beginning
 4: Add a node at the end
 5: Delete a node from the beginning
 6: Delete a node from the end
 7: Delete a node after a given node
 8: Delete the entire list
 9: EXIT
###############################################

 Enter your option : 1

 Enter -1 to end

 Enter the data : 10

 Enter the data : 20

 Enter the data : 30

 Enter the data : -1

 CIRCULAR LINKED LIST CREATED
 Enter your option : 3

 Enter the data : 11

 Enter your option : 4

 Enter the data : 99
```



C:\Windows\System32\cmd.exe

```
 Enter your option : 2
        11      10      20      30      99
 Enter your option : 5

 Enter your option : 6

 Enter your option : 2
        10      20      30
 Enter your option : 7

 Enter the value after which the node has to deleted : 20

 Enter your option : 2
        10      20
 Enter your option : 8

 CIRCULAR LINKED LIST DELETED
 Enter your option : 9

C:\Users\shweta\Documents\Shashwat\Notepad++\DSA>
```