Experiment No.7
Implement Circular Linked List ADT.
Name: Sharvari Anand Bhondekar
Roll No: 06
Date of Performance:
Date of Submission:
Marks:
Sign:

Experiment No. 7: Circular Linked List Operations

Aim: Implementation of Circular Linked List ADT

Objective:

In circular linked list last node is connected to first node. On other hand circular linked list can be used to implement traversal along web pages.

Theory:

In a circular linked list, the last node contains a pointer to the first node of the list. We can have a circular singly linked list as well as a circular doubly linked list. While traversing a circular linked list, we can begin at any node and traverse the list in any one direction, forward or backward, until we reach the same node where we started. Thus, a circular linked list has no beginning and no ending.

Inserting a New Node in a Circular Linked List

Case 1: The new node is inserted at the beginning.

Case 2: The new node is inserted at the end.

Deleting a Node from a Circular Linked List

Case 1: The first node is deleted.

Case 2: The last node is deleted.

Insertion and Deletion after or before a given node is same as singly linked list.

Algorithm

Algorithm to insert a new node at the beginning

Step 1: IF AVAIL = NULL

Write OVERFLOW

Go to Step 9 [END OF IF]

Step 2: SET NEW_NODE = AVAIL

Step 3: SET AVAIL = AVAIL→NEXT

Step 4: SET NEW_NODE-->DATA = VAL

Step 5: SET PTR=START

Repeat Step 6 while PTR NEXT != START

Step 6: SET PTR = PTR NEXT [END OF LOOP]

Step 7: SET NEW_NODE--> NEXT= START

Step 8: SET PTR-->NEXT = START

Step 9: SET START = NEW_NODE

Step 10: EXIT

Algorithm to insert a new node at the end

Step 1: IF AVAIL = NULL

Write OVERFLOW

Go to Step 11 [END OF IF]

Step 2: SET NEW_NODE = AVAIL

Step 3: SET AVAIL = AVAIL--> NEXT

Step 4: SET NEW_NODE -->DATA = VAL

Step 5: SET NEW_NODE-->NEXT = START

Step 6: SET PTR = START

Step 7: Repeat Step 8 while PTR--> NEXT != START

Step 8: SET PTR = PTR -->NEXT [END OF LOOP]

Step 9: SET PTR -->NEXT = NEW_NODE

Step 10: EXIT

Algorithm to delete the first node

Step 1: IF START = NULL

Write UNDERFLOW

Go to Step 6 [END OF IF]

Step 2: SET PTR = START

Step 3: Repeat Step 4 while PTR--> NEXT != START

Step 4: SET PTR = PTR -->NEXT [END OF LOOP]

Step 4: SET PTR→NEXT = START -->NEXT

Step 5: FREE START

Step 6: EXIT

Algorithm to delete the last node

Step 1: IF START = NULL

Write UNDERFLOW

Go to Step 7 [END OF IF]

Step 2: SET PTR = START [END OF LOOP]

Step 3: Repeat Step 4 and Step 5 while PTR -->NEXT != START

Step 4: SET PREPTR = PTR

Step 5: SET PTR = PTR -->NEXT

Step 6: SET PREPTR-->NEXT = START

Step 7: FREE PTR

Step 8: EXIT

Code:

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

struct Node {
    int data;
    struct Node* next;
};
struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
void createCircularList(struct Node** head, int n) {
    struct Node* temp = NULL;
    struct Node* tail = NULL;
    for (int i = 1; i <= n; i++) {
    struct Node* newNode = createNode(i);
```

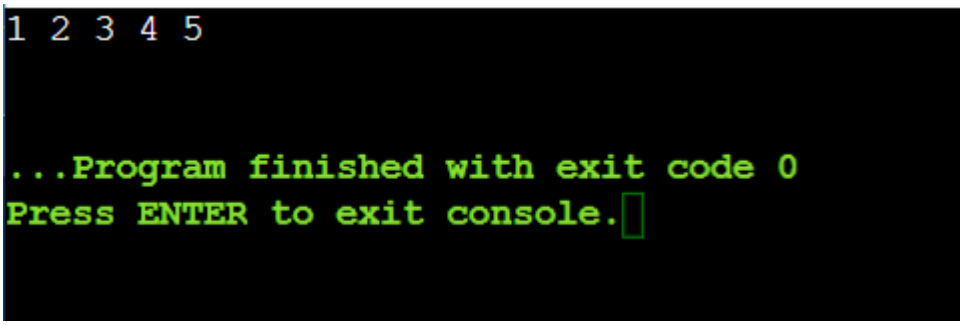
```
        if (*head == NULL) {
        *head = newNode;
        tail = newNode;
        } else {
        tail->next = newNode;
        tail = newNode;
        }
        }
        tail->next = *head;
}

void displayList(struct Node* head) {
        struct Node* temp = head;
        if (head != NULL) {
        do {
        printf("%d ", temp->data);
        temp = temp->next;
        } while (temp != head);
        }
        printf("\n");
}

int main() {
        struct Node* head = NULL;
        int n = 5;
        createCircularList(&head, n);
```

```
    displayList(head);
    return 0;
}
```

Output:



Conclusion:

Write an example of insertion and deletion in the circular linked list while traversing the web pages?

➔ Circular Linked List for Web Page Navigation:

Insertion: Inserting a new web page into the list.

You are browsing, and you open a new tab (page) in the middle of your existing browsing history.

Deletion: Deleting a web page from the list.

You close the current web page (node), removing it from your browsing history.

Example:

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

```
struct Node {
    char page[50];
    struct Node* next;
};
struct Node* createNode(char* page) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    strcpy(newNode->page, page);
    newNode->next = NULL;
    return newNode;
}
void insertPage(struct Node** head, char* newPage) {
    struct Node* newNode = createNode(newPage);
    if (*head == NULL) {
            *head = newNode;
            newNode->next = *head;
    } else {
            struct Node* temp = *head;
            while (temp->next != *head) {
            temp = temp->next;
            }
            temp->next = newNode;
```

```
        newNode->next = *head;
    }
}
void deleteCurrentPage(struct Node** head, char* currentPage) {
    if (*head == NULL) return;
    struct Node* temp = *head;
    struct Node* prev = NULL;
    while (strcmp(temp->page, currentPage) != 0) {
        if (temp->next == *head) return;
        prev = temp;
        temp = temp->next;
    }
    if (temp->next == temp) {
        *head = NULL;
    } else if (temp == *head) {
        struct Node* last = *head;
        while (last->next != *head) {
        last = last->next;
        }
        last->next = temp->next;
        *head = temp->next;
```

```
    } else {
        prev->next = temp->next;
    }
    free(temp);
}
void displayPages(struct Node* head) {
    if (head == NULL) return;
    struct Node* temp = head;
    do {
        printf("%s -> ", temp->page);
        temp = temp->next;
    } while (temp != head);
    printf("\n");
}
int main() {
    struct Node* head = NULL;

    insertPage(&head, "HomePage");
    insertPage(&head, "Page1");
    insertPage(&head, "Page2");
    insertPage(&head, "Page3");
```

```
    printf("Browsing History:\n");
    displayPages(head);
    deleteCurrentPage(&head, "Page2");
    printf("\nAfter closing 'Page2':\n");
    displayPages(head);
    insertPage(&head, "Page4");
    printf("\nAfter opening 'Page4':\n");
    displayPages(head);
    return 0;
}
```

OUTPUT:

```
Browsing History:
HomePage -> Page1 -> Page2 -> Page3 ->

After closing 'Page2':
HomePage -> Page1 -> Page3 ->

After opening 'Page4':
HomePage -> Page1 -> Page3 -> Page4 ->


...Program finished with exit code 0
Press ENTER to exit console.
```