**DS LAB 9 - Circular Linked List**

Name : Vivian Ludrick

Branch : SE Comps A Batch C

Roll no: 9914

**CODE:**

**#include <stdio.h>**

**#include <stdlib.h>**

***// Node structure***

**typedef struct Node**

**{**

**int data;**

**struct Node \*next;**

**} Node;**

***// CLL structure***

**typedef struct**

**{**

**Node \*start;**

**} CLL;**

***// Function to create a new node***

**Node \*createNode(int *data*)**

**{**

**Node \*newNode = (Node \*)malloc(sizeof(Node));**

**newNode->data = *data*;**

**newNode->next = NULL;**

**return newNode;**

**}**

***// Function to insert a node at the end of the linked list***

**void insertAtEnd(CLL \**head*, int *data*)**

**{**

**Node \*newNode = createNode(*data*);**

**if (*head*->start == NULL)**

**{**

***head*->start = newNode;**

**newNode->next = *head*->start;**

**return;**

**}**

**Node \*temp = *head*->start;**

**while (temp->next != *head*->start)**

**{**

**temp = temp->next;**

**}**

**temp->next = newNode;**

**newNode->next = *head*->start;**

**}**

***// Function to insert a node at the beginning of the linked list***

**void insertAtBeginning(CLL \**head*, int *data*)**

**{**

**Node \*newNode = createNode(*data*);**

**if (*head*->start == NULL)**

**{**

***head*->start = newNode;**

**newNode->next = *head*->start;**

**return;**

**}**

**Node \*temp = *head*->start;**

**while (temp->next != *head*->start)**

**{**

**temp = temp->next;**

**}**

**temp->next = newNode;**

**newNode->next = *head*->start;**

***head*->start = newNode;**

**}**

***// Function to insert a node after the nth node of the linked list***

**void insertAfterNthNode(CLL \**head*, int *data*, int *n*)**

**{**

**Node \*newNode = createNode(*data*);**

**if (*head*->start == NULL)**

**{**

***head*->start = newNode;**

**newNode->next = *head*->start;**

**return;**

**}**

**Node \*temp = *head*->start;**

**if (*n* == 1)**

**{**

**Node \*headPtr = *head*->start; *// so that the while loops stops at last node***

**newNode->next = *head*->start;**

***head*->start = newNode;**

***// to update the address in the last node***

***// go to the last node***

**while (temp->next != headPtr)**

**{**

**temp = temp->next;**

**}**

**temp->next = *head*->start;**

**return;**

**}**

**for (int i = 2; i < *n* && temp->next != *head*->start; i++)**

**{**

**temp = temp->next;**

**}**

**newNode->next = temp->next;**

**temp->next = newNode;**

**}**

***// Function to delete a node from the linked list***

**void deleteNode(CLL \**head*, int *data*)**

**{**

**if (*head*->start == NULL)**

**{**

**printf("List is empty\n");**

**return;**

**}**

**Node \*temp = *head*->start;**

**Node \*prev = NULL;**

**while (temp->next != *head*->start && temp->data != *data*)**

**{**

**prev = temp;**

**temp = temp->next;**

**}**

**if (temp->data != *data*)**

**{**

**printf("Node not found\n");**

**return;**

**}**

**if (temp == *head*->start)**

**{**

***head*->start = temp->next;**

**}**

**prev->next = temp->next;**

**free(temp);**

**}**

***// Function to display the linked list***

**void display(CLL \**head*)**

**{**

**if (*head*->start == NULL)**

**{**

**printf("List is empty\n");**

**return;**

**}**

**Node \*temp = *head*->start;**

**printf("Linked list: ");**

**do**

**{**

**printf("%d ", temp->data);**

**temp = temp->next;**

**} while (temp != *head*->start);**

**printf("\n");**

**}**

***// Main function***

**int main()**

**{**

**CLL head;**

**head.start = NULL;**

**int choice, data, n;**

**while (1)**

**{**

**printf("1. Insert at end\n");**

**printf("2. Insert at beginning\n");**

**printf("3. Insert after nth node\n");**

**printf("4. Delete a node\n");**

**printf("5. Display\n");**

**printf("6. Exit\n");**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice)**

**{**

**case 1:**

**printf("Enter data: ");**

**scanf("%d", &*data*);**

**insertAtEnd(&*head*, data);**

**break;**

**case 2:**

**printf("Enter data: ");**

**scanf("%d", &*data*);**

**insertAtBeginning(&*head*, data);**

**break;**

**case 3:**

**printf("Enter data: ");**

**scanf("%d", &*data*);**

**printf("Enter position: ");**

**scanf("%d", &*n*);**

**insertAfterNthNode(&*head*, data, n);**

**break;**

**case 4:**

**printf("Enter data: ");**

**scanf("%d", &*data*);**

**deleteNode(&*head*, data);**

**break;**

**case 5:**

**display(&*head*);**

**break;**

**case 6:**

**exit(0);**

**default:**

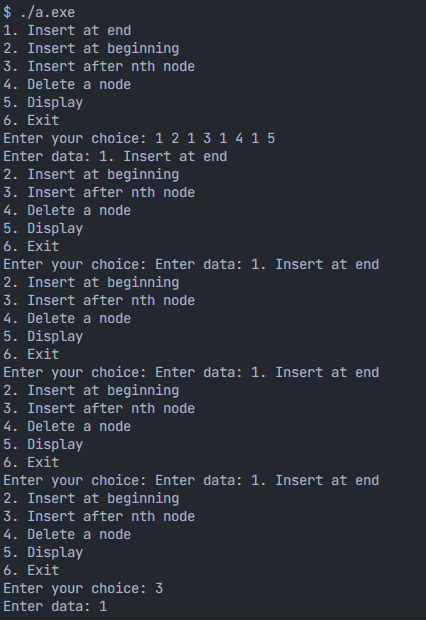
**printf("Invalid choice\n");**

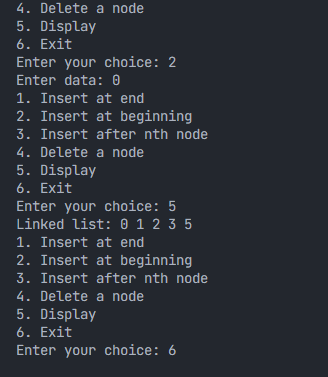
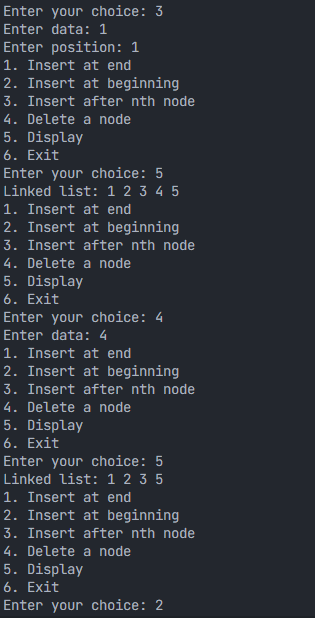
**}**

**}**

**return 0;**

**}**

**OUTPUT:**

****