G. H. RAISONI COLLEGE OF ENGG., NAGPUR
(An Autonomous Institute under UGC Act 1956)
Department of Artificial Intelligence

Date: 13/08/2020

Practical Subject: Data Structures and Algorithms
Session: 2020-21

Student Details:

Roll Number	58
Name	Shivam Tawari
Semester	3
Section	A
Branch	Artificial Intelligence

Practical Details: Practical Number- 5

Practical Aim	Design, develop and implement a program in C to implement doubly linked list where each node consists of integers. The program should support following functions: a) Create a doubly linked list b) Insert a new node c) Delete a node if it is found, otherwise display appropriate message d) Display the nodes of doubly linked list
Theory	***Linked List:*** A linked list is a linear data structure where each element is a separate object. Linked list elements are not stored at contiguous location; the elements are linked using pointers. Each node of a list is made up of two items - the data and a reference to the next node. The last node has a reference to null. The entry point into a linked list is called the head of the list. ***Doubly Linked List:*** Doubly linked list is a type of linked list in which each node apart from storing its data has two links. The first link points to the previous node in the list and the second link points to the next node in the list. The first node of the list has its previous link pointing to NULL similarly the last node of the list has its next node pointing to NULL.

	Insertion: Adds an element at the beginning of the list. The new node is always added before the head of the given Linked List. And newly added node becomes the new head of the Linked List. ***Delete:*** Deletes an element using the given key. ***Display:*** Displays the complete list: Traversal in forward direction.
Procedure	1. START 2. Ask user for list size 3. Insert elements in Linked List from head 4. Display Linked List 5. Ask user for deleting an element 6. Find and delete the node 7. Display new Linked List 8. STOP
Algorithm	Step 1: START Step 2: Ask user for list size n and initialize i = 0 Step 3: Start with an empty list; point head to NULL Step 4: Enter an element temp and pass it to insert function Step 5: Allocate and put data in node Step 6: Make next of new node as head and previous as NULL Step 7: Change prev of head node to new node Step 8: Move the head to point to the new node Step 9: Increment i by 1 Step 10: While i < size of list n, go to Step 4 else go to Step 11 Step 11: Display the list Step 12: Ask user for deleting an element temp Step 13: Find previous node of the node to be deleted Step 14: Change the next of previous node Step 15: Free memory for the node to be deleted Step 16: Display New Linked List Step 17: STOP

Program

main.c

Run

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

struct Node {
    int data;
    struct Node* next;
    struct Node* prev;
};

void insert(struct Node** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));

    new_node -> data = new_data;

    new_node -> next = (*head_ref);
    new_node -> prev = NULL;

    if((*head_ref) != NULL)
        (*head_ref) -> prev = new_node;

    (*head_ref) = new_node;
}

void deleteNode(struct Node **head_ref, int key)
{
```

```
    struct Node* temp = *head_ref, *prev;
    if (temp != NULL && temp->data == key)
    {
        *head_ref = temp->next;
        free(temp);
        return;
    }
    while (temp != NULL && temp->data != key)
    {
        prev = temp;
        temp = temp->next;
    }
    if (temp == NULL) return;
    prev->next = temp->next;

    free(temp);
}

void printList(struct Node* node)
{
    struct Node* last;
    while(node != NULL) {
        printf("%d ", node -> data);
        last = node;
        node = node -> next;
    }
}

```

```c
int main(int argc, char const *argv[]) {
    struct Node* head = NULL;
    int temp, n, i;

    printf("\n Name: Shivam Tawari");
    printf("\nSection: A \t Semester: 3");
    printf("\nRoll Number: 58");

    printf("\nEnter number of elements to be inserted: ");
    scanf("%d", &n);

    for(i=0; i<n; i++)  {
        printf(" Enter element %d: ", i+1);
        scanf("%d", &temp);
        insert(&head, temp);
    }

    printf("\n Created Doubly Linked List: ");
    printList(head);

    printf("\n Enter element to be deleted: ");
    scanf("%d", &temp);
    deleteNode(&head, temp);

    printf("\n Doubly Linked List after deletion of %d: ", temp);
    printList(head);

    return 0;
}
```

Output

Output Clear

```
gcc -o /tmp/kWOAOxTSBV.o /tmp/kWOAOxTSBV.c -lm
/tmp/kWOAOxTSBV.o
Name: Shivam Tawari
Section: A   Semester: 3
Roll Number: 58
Enter number of elements to be inserted: 5
Enter element 1: 2
Enter element 2: 3
Enter element 3: 7
Enter element 4: 5
Enter element 5: 3
Created Doubly Linked List: 3 5 7 3 2
 Enter element to be deleted: 5
 Doubly Linked List after deletion of 5: 3 7 3 2
```

Conclusion	Hence, successfully designed and developed a program to create a Doubly Linked List and performed basic operations – insertion, delete and display on it.