**COURSE CODE:** DJS22ITL302 **DATE:5/10/2023**

**COURSE NAME:** Data Structure Laboratory **CLASS: I1-Batch1**

**NAME:** Anish Sharma

**Experiment No. 5**

**CO/LO: CO1**

**Aim: Implement Doubly LinkedList**

#### **Theory:**

Doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence. Therefore, in a doubly linked list, a node consists of three parts: node data, pointer to the next node in sequence (next pointer) , pointer to the previous node (previous pointer).

**Program:**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct node{**

**int data;**

**struct node\* next;**

**struct node\* prev;**

**};**

**struct node\*temp,\*newnode;**

**struct node\*head=NULL;**

**void insertAtEnd(){**

**int x;**

**newnode=(struct node\*)malloc(sizeof(struct node));**

**printf("\nEnter a data of created node:\n");**

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

**newnode->data=x;**

**newnode->next=NULL;**

**newnode->prev=NULL;**

**if(head==NULL){**

**head=temp=newnode;**

**}**

**else{**

**temp->next=newnode;**

**newnode->prev=temp;**

**temp=newnode;**

**}**

**}**

**void insertAtStart(){**

**int x;**

**newnode=(struct node\*)malloc(sizeof(struct node));**

**printf("\nEnter a data of created node:\nqaws");**

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

**newnode->data=x;**

**newnode->prev=NULL;**

**newnode->next=head;**

**head=newnode;**

**}**

**void insertAtPos(){**

**int pos;**

**printf("\nEnter a posiion:\n");**

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

**temp=head;**

**int i;**

**for(i=1;i<pos;i++){**

**temp=temp->next;**

**}**

**newnode=(struct node\*)malloc(sizeof(struct node));**

**int x;**

**printf("\nEnter a data of created node:\n");**

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

**newnode->data=x;**

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

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

**newnode->next=temp;**

**temp->prev=newnode;**

**}**

**void delAtEnd(){**

**temp=head;**

**while(temp->next!=NULL){**

**temp=temp->next;**

**}**

**printf("\nDeleted data:%d\n",temp->data);**

**temp->prev->next=0;**

**temp=temp->prev;**

**}**

**void delAtPos(){**

**int pos;**

**printf("\nEnter a posiion:\n");**

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

**temp=head;**

**int i;**

**for(i=1;i<pos;i++){**

**temp=temp->next;**

**}**

**printf("\nDelete element:%d",temp->data);**

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

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

**}**

**void delAtStart(){**

**printf("Delete first node:%d",head->data);**

**head=head->next;**

**head->prev=0;**

**}**

**void display(){**

**temp=head;**

**while(temp->next!=0){**

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

**temp=temp->next;**

**}**

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

**}**

**int main()**

**{**

**int choice;**

**while(1){**

**printf("\nInsertion at End- 1\nInsertion at starting - 2\nDisplaying data - 3\nDelete at End - 4\nDelete at Start - 5\nInsert At any position -6\nDelete at any position-7\nExit -0\n");**

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

**if(choice==1){**

**insertAtEnd();**

**}**

**else if(choice==2){**

**insertAtStart();**

**}**

**else if(choice==3){**

**display();**

**}**

**else if(choice==4){**

**delAtEnd();**

**}**

**else if(choice==5){**

**delAtStart();**

**}**

**else if(choice==6){**

**insertAtPos();**

**}**

**else if(choice==7){**

**delAtPos();**

**}**

**else if(choice==0){**

**break;**

**}**

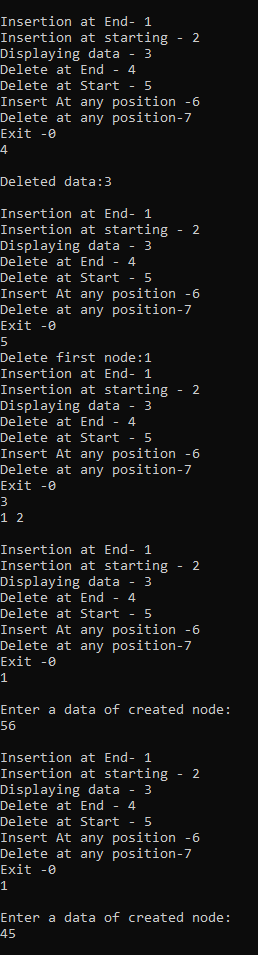
**}**

**return 0;**

**}**

**Output screenshots:**

#### 



#### **Conclusion:**

I have understood the concept of Doubly Linkedlist with insertion, deletion and displaying element.

**REFERENCES:**

No references(mam’s notes)

