ANAGHA ACHARYA

1BM19BT005

LAB – 8: DLL

WAP Implement doubly link list with primitive operations a) Create a doubly linked list. b) Insert a new node to the left of the node. c) Delete the node based on a specific value d) Display the contents of the list

#include<stdio.h>

#include<stdlib.h>

struct node

{

struct node \*prev;

struct node \*next;

int data;

};

struct node \*head;

void insertion\_beginning();

void deletion\_specified();

void display();

void main ()

{

int choice =0;

while(choice != 9)

{

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

printf("\n1.Insert to left of the node\n2.Delete from any specified location\n3.Display list\n4.Exit\n");

printf("\nEnter your choice: \n");

scanf("\n%d",&choice);

switch(choice)

{

case 1:

insertion\_beginning();

break;

case 2:

deletion\_specified();

break;

case 3:

display();

break;

case 4:

exit(0);

break;

default:

printf("\n Invalid choice. Enter again!");

}

}

}

void insertion\_beginning()

{

struct node \*ptr;

int item;

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

if(ptr == NULL)

{

printf("\n OVERFLOW!\n");

}

else

{

printf("\nEnter value to be inserted: \n");

scanf("%d",&item);

if(head==NULL)

{

ptr->next = NULL;

ptr->prev=NULL;

ptr->data=item;

head=ptr;

}

else

{

ptr->data=item;

ptr->prev=NULL;

ptr->next = head;

head->prev=ptr;

head=ptr;

}

printf("\nNode inserted succesfully\n");

}

}

void deletion\_specified()

{

struct node \*ptr, \*temp;

int val;

printf("\nEnter the data after which the node is to be deleted: \n");

scanf("%d", &val);

ptr = head;

while(ptr -> data != val)

ptr = ptr -> next;

if(ptr -> next == NULL)

{

printf("\nCan't be deleted!\n");

}

else if(ptr -> next -> next == NULL)

{

ptr ->next = NULL;

printf("\nNode deletion successful\n");

}

else

{

temp = ptr -> next;

ptr -> next = temp -> next;

temp -> next -> prev = ptr;

free(temp);

printf("\nNode deletion successful\n");

}

}

void display()

{

struct node \*ptr;

printf("\nThe values in the doubly linked list are:\n");

ptr = head;

while(ptr != NULL)

{

printf("%d\t",ptr->data);

ptr=ptr->next;

}

}

OUTPUT:







