9) **Doubly Linked List Program**

```
#include <stdio.h>
#include< stdlib.h>
struct node {
int data;
struct node *next;
struct node *prev;
};
struct node *head=NULL;
void insert_beg() {
struct node *new_node;
new_node=(struct node*) malloc (size of (struct node));
printf("Enter the item : \n");
scanf("%d", &new_node ->data);
new_node -> next = NULL;
new_node -> prev= NULL;
if (head==NULL)
head= new_node;
else {
new_node -> next =head;
head -> prev = new_node;
head= new_node;
}
}
void insert_end() {
struct node ~~*new_node~~ *new_node, *temp;
new_node =(struct node*) malloc (size of (struct node));
printf ("Enter the item \n");
scanf ("%d", &new_node -> data);
new_node -> next = NULL;
new_node -> prev =NULL;
if (head == NULL)
head = new_node;
else {
temp = head;
while (temp -> next != NULL)
temp = temp -> next;
temp -> next = new_node;
new_node -> prev = temp;
}
}
void insert_bef () {
struct node *new_node, *ptr;
int num, val;
printf ("Enter the data: ");
scanf ("%d", &num);
printf ("Enter the value before which the data has to inserted :")
scanf ("%d", &val);
new_node = (struct node *) malloc (size of (struct node));
new_node -> data = num;
ptr = head;
while (ptr -> data != val) {
ptr = ptr -> next;
if (ptr == NULL)
printf ("Element is not in the list !!! "); return;
}
newnode -> next = ptr;
new_node -> prev = ptr -> prev;
ptr -> prev -> next = new_node;
ptr -> prev = new_node;
}
void insert_after () {
```

```
int listele ;
struct node *new_node , *temp ;
printf ("Enter the element in the list \n");
scanf ("%d" ,&listele);
new_node = (struct node*) malloc(size of(struct node)) ;
printf ("Enter the new node data \n");
scanf ("%d", &new_node -> data);
newnode -> next = NULL ;
new_node -> prev = NULL ;
if (head = = NULL)
printf ("Empty list \n"); return ;
temp = head ;
while (temp -> data ! = listele ) {
temp = temp -> next ;
if ( temp = =NULL)
printf ("Element is not in the list"); return ;
}
new_node -> next = temp -> next ;
temp -> next = new_node ;
new_node -> prev = temp ;
new_node -> next -> prev = new_node ;
}
void del () {
struct node *temp ;
int ele ;
if (head = =NULL)
printf ("Empty list \n"); return ;
printf ("Enter the element to be deleted \n");
scanf ("%d" , &ele);
temp = head ;
while (temp -> data ! =ele) {
temp = temp -> next ;
```

```
if (temp = =NULL)
printf ("Element is not in the list !!!\n"); break;
}
if (temp = =head)
head = head -> next ;
else if (temp -> next = =NULL) {
temp = temp -> prev ;
temp -> next = NULL ;
} else {
temp -> prev -> next = temp -> next ;
temp -> next -> prev = temp -> prev ;
}
}
void display() {
struct node *temp ;
temp = head ;
printf ("<--Contents of Doubly Linked list -->\n");
while (temp != NULL) {
~~printf ("\t %d"~~ printf ("\t %d", temp -> data);
temp = temp -> next ;
}
printf ("\n <--------------------> \n");
}
int main () {
int choice ;
while(1){
printf ("\n <------MENU------->\n") ;
printf ("\n 1. Insert at the Beginning .\n");
printf ("2. Insert at the End .\n");
printf ("3. Insert before a given node . \n");
printf ("4. Insert after a given node . \n");
printf ("5. Delete a node . \n");
printf ("6. DISPLAY \n");
printf ("7. Exit \n");
printf ("\nEnter your choice : \n");
scanf ("%d"& choice );
switch (choice) {
case 1 : insert_beg (); break;
case 2 : insert_end (); break;
case 3 : insert_bef (); break;
case 4: insert_aftr (); break;
case 5: del () ; break;
case 6 : display (); break;
case 7: exit(0);
}
}
}
```