Doubly linked list :→

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

struct node {
    int data;
    struct node *prev;
    struct node *next;
};
struct node *head;
void create() {
    struct node *newnode, *temp;
    int choice = 2;
    head = NULL;
    while (choice != 0) {
    newnode = (struct node *)malloc(sizeof(struct node));
    printf("Enter the data: ");
    scanf("%d", &newnode->data);
    newnode->prev = NULL;
    newnode->next = NULL;
    if (head == NULL) {
        temp = head = newnode;
    }
    else {
        temp->next = newnode;
        newnode->prev = temp;
        temp = newnode;
    }
    printf("Do you want to continue ?(1|0):");
    scanf("%d", &choice);
}
}
```

```
void display() {
    struct node *temp;
    temp = head;
    if (temp == NULL) {
        printf("List is empty\n");
    } else {
        printf("Elements of the list:");
        while (temp != NULL) {
            printf("%d", temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}
void deletespecific() {
    struct node *temp, *prevnode, *nextnode;
    int value;
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    printf("Enter the data to delete:");
    scanf("%d", &value);
    temp = head;
    while (temp != NULL && temp->data != value)
    {
        temp = temp->next;
    }
    if (temp == NULL) {
        printf("Value not found in the list\n");
        return;
    }
```

```
prev_node = temp -> prev;
next_node = temp -> next;
if (prev_node != NULL) {
        prev_node -> next = next_node;
}
else {
        head = next_node;
}
if (next_node != NULL) {
        next_node -> prev = prev_node;
}
free (temp);
}
void insertleft (int item) {
struct node *temp, *newnode;
  temp = head;
newnode = (struct node *) malloc (sizeof (struct node));
newnode -> data = item;
newnode -> prev = 0;
newnode -> next = 0;
int value;
printf ("enter the left the insert ?");
scanf ("%d", &value);
while (temp -> data != value) {
        temp = temp -> next;
}
temp2 = temp -> prev;
```

```
if (temp1 == 0) {
    newnode -> next = temp;
    temp1 = newnode;
    head = newnode;
}
else {
    newnode -> prev = temp2;
    newnode -> next = temp;
    temp2 -> next = newnode;
    temp -> prev = newnode;
}
}
int main() {
    create();
    display();
    deletespecific();
    display();
    insertleft(5);
    display();
    return 0;
}
```

Output

Enter data: 1

Do you want to continue(1/0): 1

Enter data: 2

Do you want to continue(1/0): 0

Elements in the list: 1 2

Enter the data to delete: 2

Elements of the list : 2
which node value left to insert ?: 2
Elements of the list : 5 2