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

typedef struct dll {
    int data;
    struct dll * left ;
    struct dll * right;
}dll;

dll * head = NULL;
int count = 0;

dll * createnode() {
    dll * newnode = (dll *)malloc(sizeof(dll));
    printf("Enter the data element : ");
    scanf("%d" , &newnode->data);
    newnode->left = NULL;
    newnode->right = NULL;
    count++;
    return newnode;
}

void insert() {
    if (head == NULL) {
        printf("Creating the very first node.\n");
        head = createnode();
    }
    else {
        printf("Enter the posistion at which you want to insert
(between 0 and %d) : " , count);
                int pos;
                scanf("%d", &pos);
                if(pos <= count && pos>-1) {
            dll * temp = createnode();
            if(pos == 0) {
                temp->right = head;
                head->left = temp;
                head = temp;
            }
            else {
                dll * trav = head;
                                for(int i = 0 ; i < pos-1 ; i++ )
                                        trav = trav->right;
                temp->left = trav;
                temp->right = trav->right;
                trav->right = temp;
            }
        }
        else
            printf("Invalid Input.\n");
    }
}

void delete() {
    if(head == NULL)
        printf("Doubly Linked List is empty.\n");
    else {
        printf("Enter the posistion at which you want to delete
(between 0 and %d) : " , count-1);
                int pos;
                scanf("%d", &pos);
        if(pos < count && pos>-1) {
            if(pos == 0) {
                printf("Element %d deleted.\n" , head->data);
                head = head->right;
            }
            else {
                dll * trav = head;
                for(int i = 0 ; i < pos ; i++ )
                                        trav = trav->right;
                printf("Element %d deleted.\n" , trav->data);
                dll * back = trav->left;
                dll * front = trav->right;
                back->right = front;
                if (front != NULL)
                    front->left = back;
                trav = NULL;
            }
        }
        else
            printf("Invalid input.\n");
    }
}

void display() {
        if(head != NULL) {
                dll * strt = head;
                printf("Elements of list are : ");
                while(strt != NULL) {
                        printf("%d " , strt->data);
                        strt = strt->right;
                }
                printf("\n");
        }
        else
                printf("Doubly Linked List is empty.\n");
}

void main() {
    printf("-----Doubly Linked List-----\n");
        printf("We follow zero based index.\n");
        printf("Enter the operation to be performed.\n");
        int ch , flag  =1;
        while(flag) {
                printf("1.Insert \t 2.Display \t 3.Delete \t 4.Exit :
");
                scanf("%d" , &ch);
                switch(ch) {
                        case 1 : insert();
                                         break;
                        case 2 : display();
                                         break;
                        case 3 : delete();
```

```
                                         break;
                        case 4 : flag = 0;
                                         break;
                        default : printf("Invalid Input.\n");
                                          break;
                }
        }
}
```