Aim

Implement DLL with insertion and traversal both ways.

Algorithm

Each node has `prev` and `next`. Insert nodes and traverse forward/backward.

C Code

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

struct Node{
    int data;
    struct Node* prev;
    struct Node* next;
};

struct Node* createNode(int data){
    struct Node* n=(struct Node*)malloc(sizeof(struct Node));
    n->data=data; n->prev=n->next=NULL;
    return n;
}

void printForward(struct Node* head){
    while(head){ printf("%d <-> ",head->data); head=head->next; }
    printf("NULL\n");
}
```

```
void printBackward(struct Node* tail){
    while(tail){ printf("%d <-> ",tail->data); tail=tail->prev; }
    printf("NULL\n");
}

int main(){
    struct Node* head=createNode(1);
    struct Node* n2=createNode(2);
    struct Node* n3=createNode(3);
    head->next=n2; n2->prev=head;
    n2->next=n3; n3->prev=n2;

    printf("Forward: "); printForward(head);
    printf("Backward: "); printBackward(n3);
    return 0;
}
```

Input

List: 1 ↔ 2 ↔ 3

Output

```
Forward: 1 <-> 2 <-> 3 <-> NULL
Backward: 3 <-> 2 <-> 1 <-> NULL
```

Result

Doubly linked list traversal works both directions.