```cpp
/*-----Doubly Connected Linklist------*/

#include<iostream>
using namespace std;

struct node
{
    int data;

    node *next;
    node *prev;
};

class dLinkList
{
    private:
        node *head;
        node *tail;
    public:
        dLinkList()
        {
            head = NULL;
            tail = NULL;
        }
        void addFront(int n)
        {
            node *temp = new node;
            temp -> data = n;
            temp -> prev = NULL;
            temp -> next = NULL;

            if(head == NULL)
            {
                head = tail = temp;
            }
            else
            {
                temp -> next = head;
                head -> prev = temp;
                head = temp;
            }

        }

        void addBack(int n)
        {
```

```
47:             node *temp = new node;
48:             temp -> data = n;
49:             temp -> prev = NULL;
50:             temp -> next = NULL;
51:
52:             if(tail == NULL)
53:             {
54:                 head = tail = temp;
55:             }
56:             else
57:             {
58:                 temp -> prev = tail;
59:                 tail -> next = temp;
60:                 tail = temp;
61:             }
62:         }
63:
64:         void showList()
65:         {
66:             node *temp = new node;
67:             temp = head;
68:             while(temp != NULL)
69:             {
70:                 cout << temp -> data << " ";
71:                 temp = temp -> next;
72:             }
73:
74:         }
75: };
76:
77: int main()
78: {
79:     dLinkList list1;
80:     list1.addFront(2);
81:     list1.addFront(1);
82:     list1.addBack(3);
83:     list1.addBack(4);
84:
85:     list1.showList();
86:
87:     return 0;
88: }
```