

7/01/2021

DSC LAB EXAM

Deeven Prakash Paramaj
IBM19IS048
3rd Sem
Batch - 3

Doubly Linked List

- i) Insert before given value
- ii) Delete the middle node

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

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

typedef struct node Node;

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

Node* InsertAtHead(Node *head, int data)
{
    Node *new_node = newNode(data);
    if(head == NULL)
        return new_node;
    head->prev = new_node;
    new_node->next = head;
}
```

①

DParamaj

```
    head = new_node;
    return head;
}
```

```
Node* InsertBeforeKey(Node* head, int key, int data)
```

```
{  
    Node* curr = head;  
    while (curr != NULL && curr->data != key)  
        curr = curr->next;
```

```
    if (curr == NULL)
```

```
        printf("The element %d is not present in the list\n", key);  
    }  
    return head;
```

```
{  
    if (curr == head)
```

```
        head = InsertHead(head, data);
```

```
    }  
}
```

```
    Node* new_node = newNode(data);
```

```
    Node* prev = curr->prev;
```

```
    new_node->next = curr;
```

```
    new_node->prev = prev;
```

```
    curr->prev = new_node;
```

```
    prev->next = new_node;
```

```
    return head;
```

```
}
```

```
Node* DeleteMiddleNode(Node* head)
```

```
{
```

```
    if (head == NULL)
```

```
        return NULL;
```

```
    Node* slow = head, * fast = head;
```

(2)

DParamaj

```
while (slow != NULL && fast != NULL && fast->next != NULL)
```

```
{  
    slow = slow->next;
```

```
    fast = fast->next->next;
```

```
}  
if (slow == head) // when only one node is present
```

```
{  
    Node *temp = head;
```

```
    head = NULL;
```

```
    free(temp);
```

```
}  
else if (slow->next == NULL) // when 2 nodes are present
```

```
{  
    Node *temp = slow;
```

```
    slow = slow->prev;
```

```
    slow->next = NULL;
```

```
    free(temp);
```

```
}  
else
```

```
{  
    Node *prev = slow->prev;
```

```
    Node *next = slow->next;
```

```
    prev->next = next;
```

```
    next->prev = prev;
```

```
    free(slow);
```

```
}  
return head;
```

```
}
```

```
void Print(Node *head)
```

```
{  
    if (head == NULL)
```

③

DParamaj

```

printf("The list is empty \n");
return;
}

printf("The elements in the list are : ");
Node *n = head;
while (n != NULL)
{
    printf("%d, ", n->data);
    n = n->next;
}
printf("\n");

void freeList(Node *head)
{
    Node *n = head, *temp;
    while (n != NULL)
    {
        temp = n;
        n = n->next;
        free(temp);
    }
}

int main()
{
    int opt;
    Node *head = NULL;
    do
    {
        printf("In Menu : \n");
        printf("1. Insert At head \n");
        printf("2. Insert Before a given key value \n");
        printf("3. Delete middle node \n");
        printf("4. Exit \n");
        printf("Enter option : ");
        scanf("%d", &opt);
    }

```

```
if (opt == 1)
{
    int data;
    printf("Enter a value for insertion : ");
    scanf("%d", &data);
    head = InsertAtHead(head, data);
}

else if (opt == 2)
{
    int key, data;
    printf("Enter a key value and a value to be inserted : ");
    scanf("%d %d", &key, &data);
    head = InsertBeforeKey(head, key, data);
}

else if (opt == 3)
{
    head = DeleteMiddleNode(head);
}

Print(head);

} while (opt != 4);
freeList(head);
return 0;
}
```

The screenshot shows a code editor window with the following details:

- Title Bar:** The title bar displays "doublyLL.c" and the path "~./DSC_Lab/labTest".
- File Menu:** An "Open" menu is visible on the left.
- Save Button:** A "Save" button is located in the top right corner.
- Code Area:** The main area contains C code for a doubly linked list. The code includes a node structure, a newNode function to allocate memory, and two insertion functions: InsertAtHead and InsertBeforeKey.
- Code Content:**

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct node
5 {
6     int data;
7     struct node *prev;
8     struct node *next;
9 };
10 typedef struct node Node;
11
12 Node* newNode(int data)
13 {
14     Node *n = (Node*)malloc(sizeof(Node));
15     n->data = data;
16     n->prev = NULL;
17     n->next = NULL;
18     return n;
19 }
20
21 Node* InsertAtHead(Node *head, int data)
22 {
23     Node *new_node = newNode(data);
24     if(head == NULL)
25         return new_node;
26     head->prev = new_node;
27     new_node->next = head;
28     head = new_node;
29     return head;
30 }
31
32 Node* InsertBeforeKey(Node *head, int key, int data)
33 {
34     Node *curr = head;
35     while(curr != NULL && curr->data != key)
36         curr = curr->next;
37     if(curr == NULL)
38     {
39         printf("The element %d is not present in the list\n", key);
40         return head;
41     }
42     if(curr == head)
43     {
44         head = InsertAtHead(head, data);
45         return head;
46     }
47     Node *new_node = newNode(data);
```
- Status Bar:** The bottom status bar shows "C" with a dropdown arrow, "Tab Width: 4", "Ln 1, Col 1", and "INS".

Open ▾ doublyLL.c
~/DSC_Lab/labTest Save ⌂ ⌁ ⌒

```
32 Node* InsertBeforeKey(Node *head, int key, int data)
33 {
34     Node *curr = head;
35     while(curr != NULL && curr->data != key)
36         curr = curr->next;
37     if(curr == NULL)
38     {
39         printf("The element %d is not present in the list\n", key);
40         return head;
41     }
42     if(curr == head)
43     {
44         head = InsertAtHead(head, data);
45         return head;
46     }
47     Node *new_node = newNode(data);
48     Node *prev = curr->prev;
49     new_node->next = curr;
50     new_node->prev = prev;
51     curr->prev = new_node;
52     prev->next = new_node;
53     return head;
54 }
55
56 Node* DeleteMiddleNode(Node *head)
57 {
58     if(head == NULL)
59         return NULL;
60     Node *slow = head, *fast = head;
61     while(slow != NULL && fast != NULL && fast->next != NULL)
62     {
63         slow = slow->next;
64         fast = fast->next->next;
65     }
66     if(slow == head)//for one node
67     {
68         Node *temp = head;
69         head = NULL;
70         free(temp);
71     }
72     else if(slow->next == NULL)//for 2 nodes
73     {
74         Node *temp = slow;
75         slow = slow->prev;
76         slow->next = NULL;
77         free(temp);
78     }
}
```

C ▾ Tab Width: 4 ▾ Ln 78, Col 1 ▾ INS

Open ▾ doublyLL.c
~/DSC_Lab/labTest Save

```
72     else if(slow->next == NULL)//for 2 nodes
73     {
74         Node *temp = slow;
75         slow = slow->prev;
76         slow->next = NULL;
77         free(temp);
78     }
79     else
80     {
81         Node *prev = slow->prev;
82         Node *next = slow->next;
83         prev->next = next;
84         next->prev = prev;
85         free(slow);
86     }
87     return head;
88 }
89
90 void Print(Node *head)
91 {
92     if(head == NULL)
93     {
94         printf("The list is empty\n");
95         return;
96     }
97     printf("The elements in the list are : ");
98     Node *n = head;
99     while(n != NULL)
100    {
101        printf("%d ", n->data);
102        n = n->next;
103    }
104    printf("\n");
105 }
106
107 void freeList(Node *head)
108 {
109     Node *n = head, *temp;
110     while(n != NULL)
111     {
112         temp = n;
113         n = n->next;
114         free(temp);
115     }
116 }
117
118 int main()
```

C ▾ Tab Width: 4 ▾ Ln 118, Col 1 ▾ INS

Open ▾ doublyLL.c
~/DSC_Lab/labTest Save ⌂ ⌁ ⌒

```
104     printf("\n");
105 }
106
107 void freeList(Node *head)
108 {
109     Node *n = head, *temp;
110     while(n != NULL)
111     {
112         temp = n;
113         n = n->next;
114         free(temp);
115     }
116 }
117
118 int main()
119 {
120     int opt;
121     Node *head = NULL;
122     do{
123         printf("\nMenu : \n");
124         printf("1. Insert At Head\n");
125         printf("2. Insert Before a given key value\n");
126         printf("3. Delete the middle node\n");
127         printf("4. Exit\n");
128         printf("Enter option : ");
129         scanf("%d", &opt);
130         if(opt == 1)
131         {
132             int data;
133             printf("Enter a value for insertion : ");
134             scanf("%d", &data);
135             head = InsertAtHead(head, data);
136         }
137         else if(opt == 2)
138         {
139             int data, key;
140             printf("Enter a key value and a value for insertion : ");
141             scanf("%d%d", &key, &data);
142             head = InsertBeforeKey(head, key, data);
143         }
144         else if(opt == 3)
145             head = DeleteMiddleNode(head);
146             Print(head);
147     }while(opt != 4);
148     freeList(head);
149     return 0;
150 }
```

Bracket match found on line: 119 C Tab Width: 4 Ln 150, Col 2 INS

```
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest

deven@deven-VirtualBox:~/DSC_Lab/labTest$ gcc doublyLL.c
deven@deven-VirtualBox:~/DSC_Lab/labTest$ ./a.out

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 1
Enter a value for insertion : 4
The elements in the list are : 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 1
Enter a value for insertion : 7
The elements in the list are : 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 1
Enter a value for insertion : 3
The elements in the list are : 3 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 2
Enter a key value and a value for insertion : 7 9
The elements in the list are : 3 9 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 2
Enter a key value and a value for insertion : 1 0
```

```
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest

Enter a value for insertion : 3
The elements in the list are : 3 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 2
Enter a key value and a value for insertion : 7 9
The elements in the list are : 3 9 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 2
Enter a key value and a value for insertion : 1 0
The element 1 is not present in the list
The elements in the list are : 3 9 7 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 3
The elements in the list are : 3 9 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 3
The elements in the list are : 3 4

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Delete the middle node
4. Exit
Enter option : 4
The elements in the list are : 3 4
deven@deven-VirtualBox:~/DSC_Lab/labTest$
```

```
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest$ gcc modification.c
deven@deven-VirtualBox: ~/DSC_Lab/labTest$ ./a.out

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 5
The elements in the list are : 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 2
The elements in the list are : 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 7
The elements in the list are : 7 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 2
Enter a key value and a value for insertion : 2 1
The elements in the list are : 7 1 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 2
Enter a key value and a value for insertion : 9 0
The element 9 is not present in the list
The elements in the list are : 7 1 2 5

Menu :
1. Insert At Head
```

```
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest
deven@deven-VirtualBox: ~/DSC_Lab/labTest

2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 5
The elements in the list are : 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 2
The elements in the list are : 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 1
Enter a value for insertion : 7
The elements in the list are : 7 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 2
Enter a key value and a value for insertion : 2 1
The elements in the list are : 7 1 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 2
Enter a key value and a value for insertion : 9 0
The element 9 is not present in the list
The elements in the list are : 7 1 2 5

Menu :
1. Insert At Head
2. Insert Before a given key value
3. Exit
Enter option : 3
The elements in the list are : 7 1 2 5
deven@deven-VirtualBox:~/DSC_Lab/labTest$
```