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

typedef struct Node {
    int value;
    struct Node* prev;
    struct Node* next;
}node;

node* head = NULL;

void add_beg(int value) {
    node* ptr = (node*) malloc(sizeof(node));
    ptr->value = value;
    ptr->prev = NULL;
    ptr->next = head;
    if(head!=NULL) {
        head->prev = ptr;
    }
    head = ptr;
}

void add_key(int value, int key) {
    node* tmp = head;
    while(tmp!=NULL) {
        if(tmp->value == key) {
            break;
        }
        tmp = tmp->next;
    }
    if(tmp==NULL) {
        printf("\nNo Match\n");
        return;
    }
    if(tmp==head) {
        add_beg(value);
        return;
    }
```

```
38        }
39        node* ptr = (node*) malloc(sizeof(node));
40        ptr->value = value;
41        ptr->prev = tmp->prev;
42        ptr->next = tmp;
43        (tmp->prev)->next = ptr;
44        tmp->prev = ptr;
45    }
46  void del_key(int key) {
47        if(head == NULL) {
48            printf("\nList is Empty\n");
49            return;
50        }
51        node* tmp = head;
52        while(tmp != NULL) {
53            if(tmp -> value == key) {
54                break;
55            }
56            tmp = tmp->next;
57        }
58        if(tmp==head) {
59            if(head->next==NULL)
60            {
61                free(head);
62                head=NULL;
63                return;
64            }
65            head = head->next;
66            free(head->prev);
67            head->prev = NULL;
68            return;
69        }
70        if(tmp==NULL) {
71            printf("\nNo Match\n");
72            return;
73        }
74        if(tmp->next==NULL) {
75            tmp->prev->next = NULL;
```

```
76          free(tmp);
77          return;
78      }
79      tmp->next->prev = tmp->prev;
80      tmp->prev->next = tmp->next;
81      free(tmp);
82  }
83  void display() {
84      if(head == NULL) {
85          printf("\nList is Empty\n");
86          return;
87      }
88      node* tmp = head;
89      printf("\nLinked list contains : ");
90      while(tmp!= NULL) {
91          printf("%d ", tmp->value);
92          tmp = tmp->next;
93      }
94      printf("\n");
95  }
96  void main() {
97      int choice, value, key;
98      while(1) {
99          printf("Enter 1 to add at beginning\n");
100         printf("Enter 2 to add at left of a node\n");
101         printf("Enter 3 to delete a node\n");
102         printf("Enter 4 to display\n");
103         printf("Enter -1 to quit\n");
104         printf("Enter your choice : ");
105         scanf("%d", &choice);
106         if(choice==-1)
107             break;
108         switch(choice) {
109             case 1:
110                 printf("\nEnter value to insert : ");
111                 scanf("%d", &value);
112                 add_beg(value);
```

```
    while(1) {
        printf("Enter 1 to add at beginning\n");
        printf("Enter 2 to add at left of a node\n");
        printf("Enter 3 to delete a node\n");
        printf("Enter 4 to display\n");
        printf("Enter -1 to quit\n");
        printf("Enter your choice : ");
        scanf("%d", &choice);
        if(choice==-1)
            break;
        switch(choice) {
            case 1:
                printf("\nEnter value to insert : ");
                scanf("%d", &value);
                add_beg(value);
                break;
            case 2:
                printf("\nEnter value to insert : ");
                scanf("%d", &value);
                printf("\nEnter value of key node : ");
                scanf("%d", &key);
                add_key(value, key);
                break;
            case 3:
                printf("\nEnter value of node to be deleted : ");
                scanf("%d", &key);
                del_key(key);
                break;
            case 4:
                display();
                break;
            default:
                printf("\n\nWrong Input\n\n");
        }
    }
    printf("\n\n-----DONE-----\n\n");
}
```

```
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 1

Enter value to insert : 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 1

Enter value to insert : 5
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 2

Enter value to insert : 6

Enter value of key node : 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 4

Linked list contains : 5 6 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
```

```
Enter -1 to quit
Enter your choice : 2

Enter value to insert : 6

Enter value of key node : 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 4

Linked list contains : 5 6 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 3

Enter value of node to be deleted : 6
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice : 4

Linked list contains : 5 4
Enter 1 to add at beginning
Enter 2 to add at left of a node
Enter 3 to delete a node
Enter 4 to display
Enter -1 to quit
Enter your choice :
```