main.c

Run

Output

Clear

```c
#192210211
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
struct node {
   int data;
   struct node *next;
};
struct node *head = NULL;
struct node *current = NULL;
void printList(){
   struct node *p = head;
   printf("\n[");
   while(p != NULL) {
      printf(" %d ",p->data);
      p = p->next;
   }
   printf("]");
}
void insertatbegin(int data){
   struct node *lk = (struct node*) malloc(sizeof(struct node));
   lk->data = data;
   lk->next = head;
   head = lk;
}
void insertatend(int data){
   struct node *lk = (struct node*) malloc(sizeof(struct node));
   lk->data = data;
   struct node *linkedlist = head;
   while(linkedlist->next != NULL)
```

main.c

Run

```
      linkedlist = linkedlist->next;
   linkedlist->next = lk;
}
void insertafternode(struct node *list, int data){
   struct node *lk = (struct node*) malloc(sizeof(struct node));
   lk->data = data;
   lk->next = list->next;
   list->next = lk;
}
void deleteatbegin(){
   head = head->next;
}
void deleteatend(){
   struct node *linkedlist = head;
   while (linkedlist->next->next != NULL)
      linkedlist = linkedlist->next;
   linkedlist->next = NULL;
}
void deletenode(int key){
   struct node *temp = head, *prev;
   if (temp != NULL && temp->data == key) {
      head = temp->next;
      return;
   }

   // Find the key to be deleted
   while (temp != NULL && temp->data != key) {
      prev = temp;
      temp = temp->next;
   }
```

Output

Clear

```
/tmp/jS5ujL9wmC.o
Linked List:
[ 50  22  12  33  30  44 ]
Linked List after deletion:
[ 22  33  30 ]
Updated Linked List:
[ 16  4  22  33  30 ]
Element is found
```

```
60    }
61
62    // If the key is not present
63    if (temp == NULL) return;
64
65    // Remove the node
66    prev->next = temp->next;
67  }
68  int searchlist(int key){
69    struct node *temp = head;
70    while(temp != NULL) {
71      if (temp->data == key) {
72         return 1;
73       }
74       temp=temp->next;
75    }
76    return 0;
77  }
78  void main(){
79    int k=0;
80    insertatbegin(12);
81    insertatbegin(22);
82    insertatend(30);
83    insertatend(44);
84    insertatbegin(50);
85    insertafternode(head->next->next, 33);
86    printf("Linked List: ");
87    printList();
88    deleteatbegin();
89    deleteatend();
```

```
/tmp/jS5ujL9wmC.o
Linked List:
[ 50  22  12  33  30  44 ]
Linked List after deletion:
[ 22  33  30 ]
Updated Linked List:
[ 16  4  22  33  30 ]
Element is found
```

main.c

Run

```
    }
    temp=temp->next;
  }
  return 0;
}
void main(){
  int k=0;
  insertatbegin(12);
  insertatbegin(22);
  insertatend(30);
  insertatend(44);
  insertatbegin(50);
  insertafternode(head->next->next, 33);
  printf("Linked List: ");
  printList();
  deleteatbegin();
  deleteatend();
  deletenode(12);
  printf("\nLinked List after deletion: ");
  printList();
  insertatbegin(4);
  insertatbegin(16);
  printf("\nUpdated Linked List: ");
  printList();
  k = searchlist(16);
  if (k == 1)
    printf("\nElement is found");
  else
    printf("\nElement is not present in the list");
}
```

Output

Clear

```
/tmp/jS5ujL9wmC.o
Linked List:
[ 50  22  12  33  30  44 ]
Linked List after deletion:
[ 22  33  30 ]
Updated Linked List:
[ 16  4  22  33  30 ]
Element is found
```