S.No: 3	Exp. Name: ***Implementation of linear list using array.***	Date:2024-05-13

Aim:

Write a c program to Implement a linear list using array. (perform Insertion and deletion operations)

Source Code:

LinearList.c

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

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

void insert (int val);
void move(int val);
void print();

struct node *head=NULL;

int main()
{
   int i=1;
   while(i>0){
   printf("I - Insert\n");
   printf("D - Delete\n");
   printf("P - Print\n");
   printf("Q - Quit\n");
   int val;
   char ch;
   printf("Enter option: ");
   scanf(" %c",&ch);
   if(ch=='I' || ch== 'i')
{
      printf("Element to insert: ");
      scanf("%d",&val);
      insert(val);
   }
else if (ch=='D' || ch=='d')
   {
      printf("Element to delete: ");
      scanf("%d", &val);
      move(val);
   }
   else if(ch=='p'||ch=='P')
   {
      print();
   }
   else if(ch=='q'||ch=='Q')
   {
      break;
   }
```


ID: 92132321021

PSNA College of Engineering and Technology

```
   else
   {
      printf("Invalid option!\n");
   }
   i=i+1;
   }
   return 0;
}
void insert(int val)
{
   struct node*n1=(struct node*)malloc(sizeof(struct node));
   n1->data=val;
   n1->next=head;
   head=n1;
}
void move(int val)
{
   struct node *temp=head;
   struct node *before=NULL;
   if(temp!=NULL && temp->data==val)
   {
      head=temp->next;
      free(temp);
      return;
   }
   while(temp !=NULL){
      if (temp->data==val){
         if(before !=NULL){
            before->next=temp->next;
         }
         free(temp);
         return;
      }
      else{
         before=temp;
         temp=temp->next;
      }
   }
   printf("Element not found\n");
}
void print()
{
   struct node *temp=head;
   int size=0;
   while(temp!=NULL)
      {
         size++;
         temp=temp->next;
      }
   int arr[size];
   temp=head;

   for(int i=0;i<size;i++)
      {
      arr[i]=temp->data;
      temp=temp->next;
```

```
      }
   printf("List: ");
   for(int i=size-1;i>=0;i--)
{
         printf("%d ",arr[i]);
      }
   printf("\n");
      }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
I - Insert D
D - Delete D
P - Print D
Q - Quit D
Enter option: D
Element to delete: 14
Element not found P
I - Insert P
D - Delete P
P - Print P
Q - Quit P
Enter option: P
List: I
I - Insert I
D - Delete I
P - Print I
Q - Quit I
Enter option: I
Element to insert: 101
I - Insert I
D - Delete I
P - Print I
Q - Quit I
Enter option: I
Element to insert: 102
I - Insert I
D - Delete I
P - Print I
Q - Quit I
Enter option: I
Element to insert: 103
I - Insert p
D - Delete p
P - Print p
Q - Quit p
Enter option: p
List: 101 102 103 2

I - Insert 2
D - Delete 2
P - Print 2
Q - Quit 2
Enter option: 2
Invalid option! D
I - Insert D
D - Delete D
P - Print D
Q - Quit D
Enter option: D
Element to delete: 102
I - Insert p
D - Delete p
P - Print p
Q - Quit p
Enter option: p
List: 101 103 d
I - Insert d
D - Delete d
P - Print d
Q - Quit d
Enter option: d
Element to delete: 101
I - Insert d
D - Delete d
P - Print d
Q - Quit d
Enter option: d
Element to delete: 103
I - Insert p
D - Delete p
P - Print p
Q - Quit p
Enter option: p
List: q
I - Insert q
D - Delete q
P - Print q
Q - Quit q
Enter option: q

Test Case - 2
User Output
I - Insert 1
D - Delete 1
P - Print 1
Q - Quit 1
Enter option: 1
Invalid option! I

```
I - Insert I
D - Delete I
P - Print I
Q - Quit I
Enter option:  I
Element to insert:  15
I - Insert I
D - Delete I
P - Print I
Q - Quit I
Enter option:  I
Element to insert:  14
I - Insert i
D - Delete i
P - Print i
Q - Quit i
Enter option:  i
Element to insert:  13
I - Insert D
D - Delete D
P - Print D
Q - Quit D
Enter option:  D
Element to delete:  15
I - Insert p
D - Delete p
P - Print p
Q - Quit p
Enter option:  p
List: 14 13  Q
I - Insert Q
D - Delete Q
P - Print Q
Q - Quit Q
Enter option:  Q
```