

```

1  #include <iostream>
2
3  using namespace std;
4
5  struct Node{
6      int data;
7      Node *address;
8  };
9
10
11 Node *create(int);
12 Node *insert_begining(Node*, int);
13 Node *insert_ending(Node*, int);
14 Node *delete_begining();
15 Node *delete_ending();
16 void display(Node *head);
17
18 Node *create(int item)
19 {
20     Node * npt = new Node;
21     npt -> data = item;
22     npt -> address = NULL;
23
24     return npt;
25 }
26
27
28 Node *insert_begining(Node *head, int data)
29 {
30     Node *np = create(data);
31     if(head==NULL)
32     {
33         head = np;
34         return head;
35     }
36     np->address = head;
37     head = np;
38     return head;
39 }
40
41 Node *insert_ending(Node *head, int x)
42 {
43     Node *fst = create(x);
44     Node *snd = head;
45     if(head == NULL)
46     {
47         head = fst;
48         return head;
49     }
50     while(snd->address != NULL)
51     {
52         snd = snd->address;
53     }
54     snd->address = fst;
55     return head;
56 }
57
58
59
60 Node *delete_begining(Node *head)
61 {
62     Node *fst=NULL;
63     if(head==NULL)
64     {
65         cout<<"List is Empty, so nothing to delete";
66         return head;

```

```

67     }
68     else
69     {
70         fst=head->address;
71         delete(head);
72     }
73     return fst;
74 }
75 Node *delete_ending(Node *head)
76 {
77     Node *temp=head;
78     if(head==NULL)
79     {
80         cout<<"List is Empty, so nothing to delete";
81         return head;
82     }
83     if(head->address==NULL)
84     {
85         delete head;
86         cout<<"\n";
87         return NULL;
88     }
89     while(temp->address->address!=NULL)
90     {
91         temp=temp->address;
92     }
93     temp->address=NULL;
94     return head;
95 }
96
97 void display(Node *head)
98 {
99     if(!head)
100     {
101         cout<<"\nThere are no values in the list"<<endl;
102         return;
103     }
104     Node *snd = head;
105     cout<<"\nList elements are:"<<endl;
106     while(snd!=NULL)
107     {
108         cout<<snd->data<<" ";
109         snd = snd->address;
110     }
111 }
112
113
114 int main()
115 {
116     Node *head = NULL;
117
118     //delete_beg(head);
119
120     cout<<"\nAdding values to list from begining"<<endl;
121     head = insert_begining(head,100);
122     head = insert_begining(head,200);
123     head = insert_begining(head,300);
124     head = insert_begining(head,400);
125     head = insert_begining(head,500);
126     head = insert_begining(head,600);
127     display(head);
128     cout<<"\nDeleting value from begining"<<endl;
129     head = delete_begining(head);
130     display(head);
131     cout<<"\nDeleting value from begining"<<endl;
132     head = delete_begining(head);

```

```
133     display(head);
134     cout<<"\nDeleting  value from begining"<<endl;
135     head = delete_begining(head);
136     display(head);
137
138
139
140     cout<<"\nAdding values to list from end"<<endl;
141     head = insert_ending(head,150);
142     head = insert_ending(head,250);
143     head = insert_ending(head,350);
144     head = insert_ending(head,350);
145     head = insert_ending(head,450);
146     head = insert_ending(head,550);
147     display(head);
148     cout<<"\nDeleting  value from end"<<endl;
149     head = delete_ending(head);
150     display(head);
151     cout<<"\nDeleting  value from end"<<endl;
152     head = delete_ending(head);
153     display(head);
154     cout<<"\nDeleting  value from end"<<endl;
155     head = delete_ending(head);
156     display(head);
157     return 0;
158 }
```