lab 1 1BM18CO77
Rahul patil

```
#include <bits/stdc++.h>
#include <inttypes.h>

using namespace std;

class Node {
public:
int data;
Node* npx;
};

Node* XOR (Node *a, Node *b) {
return (Node*) ((uintptr_t) (a) ^ (uintptr_t)
(b));
}

void insertBeg(Node **head, int data) {
Node *new_node = new Node();
new_node -> data = data;

new_node -> npx = *head;

if (*head != NULL) (*head) -> npx =
```

```
XOR(new_node, (*head) -> npx);

*head = new_node;
}

void insertEnd (Node **head, int data) {
Node *new_node = new Node();
new_node -> data = data;

if (*head == NULL) {
new_node -> npx = *head;
*head = new_node;
    }

else {
Node *curr = *head;
Node *prev = NULL;
Node *next;

while XOR(prev, curr -> npx)  =!NULL) {
next = XOR (prev, curr -> npx);

prev = curr;
curr = next;
      }
```

```
new_node -> npx = curr;
curr -> npx = XOR(prev, new_node);
    }

}

Node* deleteEnd(Node* head) {
if (!head) return NULL;
Node* prev = NULL, *curr = head, *next =
XOR(prev, curr -> npx);

while (next) {
prev = curr;
curr = next;
next = XOR(prev, curr->npx);
    }

prev -> npx = XOR(prev -> npx, curr);
delete curr;
return head;
}

Node* deleteBeg(Node* head) {
if (!head) return NULL;
Node* next = XOR(head -> npx, NULL);
```

```
next->npx = XOR(head, next -> npx);
delete head;
return next;
}
void printList (Node *head) {
Node *curr = head;
Node *prev = NULL;
Node *next;

cout << "Following are the nodes of Linked List:
\n";

while (curr != NULL) {
cout << curr -> data<<" ";
next = XOR (prev, curr -> npx);

prev = curr;
curr = next;
    }
}

int main () {
Node *head = NULL;
int beg, end;
cout << "How many time you wanna insert from
```

```
beginning? \n";
cin >> beg;

while (beg--) {
int val;
cin >> val;
insertBeg(&head, val);
    }

cout << "How many time you wanna insert from
end? \n";
cin >> end;

while (end--) {
int val;
cin >> val;
insertEnd(&head, val);
    }

printList(head);
deleteBeg(head);
deleteEnd(head);

return 0;
}
```