#include<iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

class LinkedList {

public:

void PrintList(Node\* n) {

cout << "Printing List" << endl;

while (n!=NULL)

{

cout << n->data << endl;

n = n->next;

}

}

void PushFirst(Node\*\* head, int new\_data) {

Node\* new\_node = new Node;

new\_node->data = new\_data;

new\_node->next = \*head;

\*head = new\_node;

}

void PushLast(Node\*\* head, int new\_data) {

Node\* new\_node = new Node;

new\_node->data = new\_data;

new\_node->next = NULL;

Node\* last = \*head;

while (last->next!=NULL)

{

last = last->next;

}

last->next = new\_node;

}

void InsertAfter(Node\* prevNode, int new\_data) {

if (prevNode == NULL) {

cout << "the given node is empty" << endl;

}

Node\* new\_node = new Node;

new\_node->data = new\_data;

new\_node->next = prevNode->next;

prevNode->next = new\_node;

}

};

void main() {

Node\* head = new Node;

Node\* second = new Node;

Node\* third = new Node;

head->data = 10;

head->next = second;

second->data = 20;

second->next = third;

third->data = 30;

third->next = NULL;

LinkedList ll;

/\*ll.PrintList(head);

ll.PushLast(&head, 55);

ll.PrintList(head);\*/

/\*ll.PrintList(head);

ll.PushFirst(&head, 55);

ll.PrintList(head);\*/

ll.PrintList(head);

ll.InsertAfter(second, 55);

ll.PrintList(head);

}