#include <iostream>

using namespace std;

class Node {

public:

char m\_data;

Node \*next;

};

Node \*recall(Node \*head) {

if (head->next == nullptr)return head;

Node \*newHead = recall(head->next);

head->next->next = head;

head->next = nullptr;

return newHead;

}

void traverse(Node \*head) {

Node \*currentNode = head;

while (currentNode != nullptr) {

cout << currentNode->m\_data << " ";

currentNode = currentNode->next;

}

}

int main() {

Node \*a\_z = new Node[26];

for (int i = 0; i < 26; ++i) {

a\_z[i].m\_data = static\_cast<char>('A' + i);

if (i < 25) {

a\_z[i].next = &a\_z[i + 1];

}

}

a\_z[25].next = nullptr;

Node \*Head = &a\_z[0];

traverse(Head);

cout << endl;

recall(Head);

traverse(&a\_z[25]);

cout << endl;

Node \*node1 = new Node;

node1->m\_data = 'c';

Node \*node2 = new Node;

node2->m\_data = 'u';

Node \*node3 = new Node;

node3->m\_data = 'g';

node1->next = node2;

node2->next = node3;

node3->next = nullptr;

traverse(node1);

delete[]a\_z;

delete node1;

delete node2;

delete node3;

return 0;

}