#include <iostream>

using namespace std;

struct node {

char letr;

node \* next;

node \* prev;

};

void PrintList(node \* myList) {

cout << "Here are all values:" << endl;

while (myList) {

cout << myList->letr << endl;

myList = myList->next;

}

}

void PrintListBackward(node \* myList) {

/\*

cout << "Here are all values:" << endl;

while (myList) {

cout << myList->letr << endl;

myList = myList->next;

}

\*/

}

void InsertList(char item, node \*\* head) {

node \* currPtr,

\*prevPtr=nullptr,

\*insertItem = new node;

insertItem->letr = item;

//(\*head)->prev = nullptr;

currPtr = \*head;

while (currPtr != nullptr && item > currPtr->letr)

{

prevPtr = currPtr; // advance both pointers

currPtr = currPtr->next;

}

insertItem->next = currPtr; // insert new node here

if ((\*head) == nullptr) { // new list

\*head = insertItem;

insertItem->prev = nullptr;

(\*head)->prev = nullptr;

}

else if (currPtr == nullptr)

{

insertItem->prev = prevPtr;

prevPtr->next = insertItem;

}

else

{

insertItem->prev = currPtr->prev;

currPtr->prev = insertItem;

//(\*head) = insertItem;

}

}

int main() {

node \*myList;

myList = nullptr;

PrintList(myList);

InsertList('g', &myList);

PrintList(myList);

InsertList('d', &myList);

PrintList(myList);

InsertList('r', &myList);

PrintList(myList);

InsertList('m', &myList);

PrintList(myList);

return -8;

}