#include<iostream>

using namespace std;

class node {

public:

int value;

node\*next\_node;

void setvalue(int value){

this->value = value;

}

void setnextnode(node \* next\_node){

this->next\_node = next\_node;

}

int getvalue(){

return value;

}

node \* getnextnode(){

return next\_node;

}

};

class linklist{

node \* header\_node;

node \* current\_node;

node \* new\_node;

node \* temp;

int size;

public:

linklist(){

header\_node = 0;

current\_node = 0;

new\_node = 0;

}

void insert(int n){

new\_node = new node();

new\_node->setvalue(n);

new\_node->setnextnode(0);

if(header\_node != 0){

current\_node->setnextnode(new\_node);

current\_node = new\_node;

}

else {

header\_node = new\_node;

current\_node = new\_node;

}

size++;

}

printList(){

node \* temp = header\_node;

if (header\_node == 0) {

cout << "List empty" << endl;

}

while (temp != 0) {

cout << temp->value << " ";

temp = temp->next\_node;

}

}

update\_value(int old, int new) {

int pos = 0;

if(header\_node==0) {

cout<<"Linked List not initialized";

}

current\_node = header\_node;

while(current\_node->next\_node!=0) {

if(current->value == old) {

current->value = new;

printf("\n%d found at position %d, replaced with %d\n", old, pos, new);

}

current\_node = current\_node->next\_node;

pos++;

}

printf("%d does not exist in the list\n", old);

}

};

int main(){

linklist glist;

glist.insert(10);

glist.insert(20);

glist.insert(30);

glist.insert(40);

glist.insert(50);

glist.insert(60);

glist.insert(70);

glist.insert(80);

glist.insert(90);

glist.insert(100);

glist.printList();

glist.update\_data(60);

return 0;

}