#include <iostream>

using namespace std;

class Node {

public:

int data;

Node \*next;

Node(int da = 0, Node \*p = NULL) {

this->data = da;

this->next = p;

}

};

class List {

private:

Node \*head, \*tail;

int position;

public:

List() { head = tail = NULL; };

~List() {

delete head;

delete tail;

};

void print();

void Insert(int da = 0);

void Delete(int da = 0);

void Search(int da = 0);

int getValueAt(int position);

void setValueAt(int position, int da);

};

int List::getValueAt(int position) {

Node \*p = head;

if (p == NULL) {

cout << "List is Empty!" << endl;

}

else {

int posi = 0;

while (p != NULL && posi != position) {

posi++;

p = p->next;

}

if (p == NULL) {

cout << "There is no value of this position in this List!" << endl;

}

else {

cout << "value is " << p->data << endl;

}

}

return p->data;

}

void List::setValueAt(int position, int da) {

Node \*p = head;

if (p == NULL) {

cout << "List is Empty!" << endl;

}

else {

int posi = 0;

while (p != NULL && posi != position) {

posi++;

p = p->next;

}

if (p == NULL) {

cout << "There is No Position in this List!" << endl;

}

else {

p->data = da;

cout << "Value Updated!" << endl;

}

}

}

void List::Search(int da) {

Node \*p = head;

if (p == NULL) {

cout << "List is Empty!" << endl;

return;

}

int count = 0;

while (p != NULL && p->data != da) {

p = p->next;

count++;

}

if (p != NULL)

cout << "value position " << count << endl;

else

cout << "Value not found" << endl;

}

void List::Delete(int da) {

Node \*p = head, \*q = head;

if (p == NULL) {

cout << "List is Empty!" << endl;

return;

}

while (p != NULL && p->data != da) {

q = p;

p = p->next;

}

q->next = p->next;

cout << "Deletion finished!" << endl;

}

void List::Insert(int da) {

if (head == NULL) {

head = tail = new Node(da);

head->next = NULL;

tail->next = NULL;

}

else {

Node \*p = new Node(da);

tail->next = p;

tail = p;

tail->next = NULL;

}

}

void List::print() {

Node \*p = head;

while (p != NULL) {

cout << p->data << " \a";

p = p->next;

}

cout << endl;

}

int main()

{

List l1;

l1.Insert(1);

l1.Insert(2);

l1.Insert(3);

l1.Insert(4);

l1.Insert(5);

l1.Insert(6);

l1.Insert(7);

l1.print();

l1.Search(10);

l1.Delete(6);

l1.print();

l1.getValueAt(3);

l1.setValueAt(3, 9);

l1.print();

return 0;

}