#include<bits/stdc++.h>

using namespace std;

int size = 0;

class node{

private:

int data;

node \*next;

public:

void addlast(int value);

void print();

void removefirst();

int getvalue(int key);

int getat(int key);

int getfirst();

int getlast();

void addFirst(int value);

void addat(int key, int value);

void removelast();

void removeat(int key);

};

node \*head = NULL;

node \*tail = NULL;

void node::addlast(int value){

node \*temp = new node();

temp->data = value;

temp->next = NULL;

if(head==NULL){

head = temp;

tail = temp;

}

else{

tail->next = temp;

tail = temp;

}

size++;

}

void node::addFirst(int value){

node\* temp = new node();

temp->data = value;

temp->next = head;

head = temp;

}

void node::print(){

node \*temp = head;

while(temp!=NULL){

cout<<temp->data<<" ";

temp = temp->next;

}

cout<<endl;

}

void node::removefirst(){

if(head==NULL){

cout<<"Invalid arguments"<<endl;

}

else{

node\* temp = head;

head = head->next;

delete temp;

}

}

int node::getvalue(int key){

node\* temp = head;

int idx = 1;

while(temp->data!=key){

temp = temp->next;

idx++;

}

return idx;

}

int node::getfirst(){

if(head==NULL){

cout<<"List is empty"<<endl;

return -1;

}

else{

node\* temp = head;

return temp->data;

}

}

int node::getat(int key){

node\* temp = head;

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

temp = temp->next;

}

return temp->data;

}

int node::getlast(){

if(head==NULL){

cout<<"List is empty"<<endl;

return -1;

}

else{

node\* temp = head;

while(temp->next!=NULL){

temp = temp->next;

}

return temp->data;

}

}

void node::addat(int key, int value){

node\* temp = head;

for(int i=0;i<key-1;i++){

temp = temp->next;

}

node\* temp1 = new node();

temp1->data = value;

temp1->next = temp->next;

temp->next = temp1;

}

void node::removelast(){

node\* temp = head;

node\* prev;

while(temp->next!=NULL){

prev = temp;

temp = temp->next;

}

prev->next = NULL;

delete temp;

}

void node::removeat(int key){

node\* temp = head;

node\* prev;

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

prev = temp;

temp = temp->next;

}

prev->next = temp->next;

delete temp;

}

//-----------------------------------main function-----------------------------------------------

int main(){

head = NULL;

string s;

getline(cin,s);

node list;

// if(s=="display"){

// list.print();

// }

while(s!="quit"){

if(s=="display"){

list.print();

}

else if(s=="size"){

cout<<size<<endl;

}

else if(s=="removeFirst"){

list.removefirst();

}

else if(s=="getFirst"){

cout<<list.getfirst()<<endl;

}

else if(s=="getLast"){

cout<<list.getlast()<<endl;

}

else if(s=="removeLast"){

list.removelast();

}

else{

int pos = s.find(" ");

string operation = s.substr(0,pos);

string value = s.substr(pos+1);

int num = stoi(value);

if(operation=="addLast"){

list.addlast(num);

}

else if(operation=="getAt"){

int res = list.getat(num);

cout<<res<<endl;

}

else if(operation=="addFirst"){

list.addFirst(num);

}

else if(operation=="removeAt"){

list.removeat(num);

}

}

getline(cin,s);

if(s=="quit"){

break;

}

}

}