

```

#include <iostream>
using namespace std;

struct el {
    int item;
    el* next;
};

//предварительное объявление класса итератора
class listIterator;

class list {
private:
    el* head;
    el* tail;
public:
    list ();
    ~list ();
    void operator+=(int x);
    listIterator begin() const;
    listIterator end() const;
};

class listIterator {
private:
    const list *collection;
    el *cur;
public:
    listIterator (const list *s, el *e):
        collection(s), cur(e){}
    const int &operator *() {
        return cur -> item;
    }
    listIterator operator++(); // префиксный ++
    bool operator == (const listIterator &ri) const;
    bool operator != (const listIterator &ri) const;
};

listIterator listIterator:: operator++() {
    cur = cur->next;
    return *this;
}
bool listIterator:: operator==(const listIterator &ri) const{
    return ((collection == ri.collection) && (cur == ri.cur));
}
bool listIterator:: operator!=(const listIterator &ri) const {
    return ! (*this==ri);
}
list::list () {

```

```

        head=nullptr;
        tail=nullptr;
    }
list::~~list () {
    el *q;
    while (head!=nullptr){
        q=head;
        head=head->next;
        delete q;
    }
    tail=nullptr;
}
listIterator list::begin() const {
    listIterator iter(this,head);
    return iter;
}
listIterator list::end() const {
    listIterator iter(this,nullptr);
    return iter;
}

void list::operator+=(int x){
    el* q=new el;
    q->item=x;
    q->next=nullptr;
    if (head!=nullptr) {
        tail->next=q;
        tail=q;
    }
    else {
        head=q;
        tail=q;
    }
}

int main() {
list s1;
int n,a,i;
cout<<"count=\n";
cin>>n;
cout<<"elements=\n";
cout<<"1>\n";
for( i=0;i<n; i++) {
    cin>>a;
    s1+=a;
}
listIterator it = s1.begin();
cout<<" list \n";

```

```
while (it!= s1.end()){
    cout << *it<<" ";
    ++it;
}
cout<<endl;
cout<<" list \n";

for (it=s1.begin(); it!=s1.end(); ++it) cout << *it<<" ";

return 0;
}
```