Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №11.1**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Информационные и динамические структуры. Одно- и Двунаправленные списки

Вариант 1

Выполнил:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

1. Получить практические навыки работы с однонаправленными списками;

2) Получить практические навыки работы с двунаправленными списками;

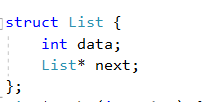
**Постановка задачи**

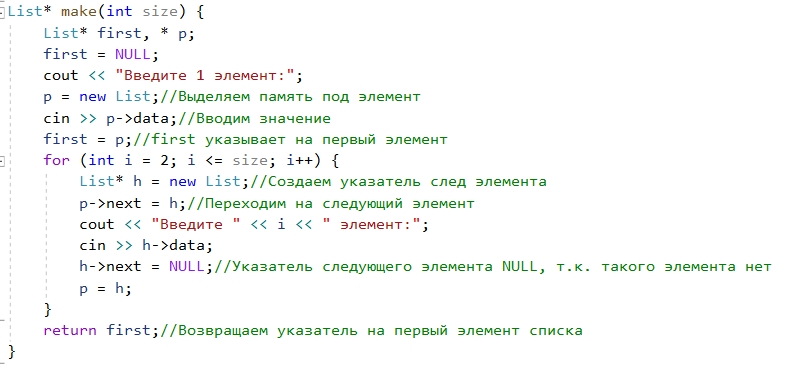
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  варианта | Однонаправленный список | Двунаправленный список | Стек | Очередь |
| 1 | Тип информационного поля int.  Удалить из списка все элементы с четными информационными полями. | Тип информационного поля char\*.  Добавить в список элемент с заданным номером. | Тип информационного поля int.  Удалить из стека все элементы с четными информационными полями. | Тип информационного поля char\*.  Добавить в очередь элемент с заданным номером. |

**Анализ задач**

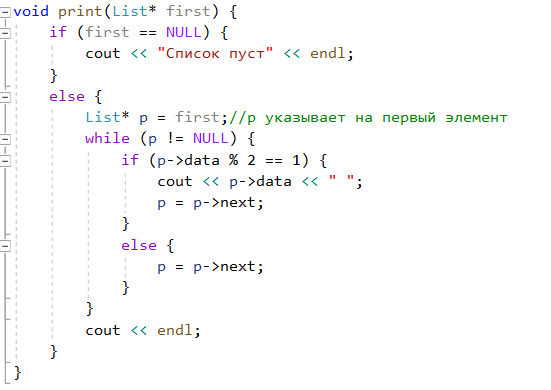
**1.** Для решения задачи необходимо:

1.1.Создать структуру переменных:



**1.2.** Разработать функцию для создания однонаправленного списка: 

**1.3.**Разработать функцию print для вывода всех нечетных элементов списка:



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для функции make используется входная переменная size целого типа данных (int)



**2.2.** Для функции print используется указатель на первый элемент

\*first :



**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для обработки элементов был использован однонаправленный список

**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Ввод данных происходит благодаря функции cin:

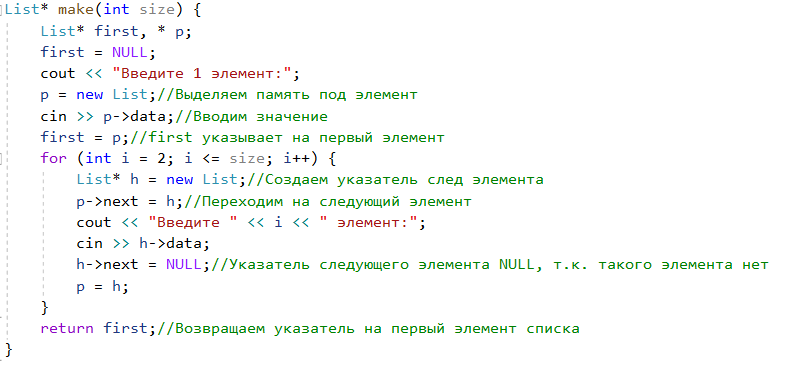


**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

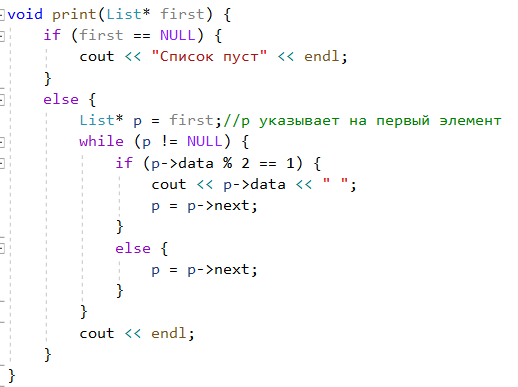


**5.**Поставленные задачи будут решены следующими действиями:

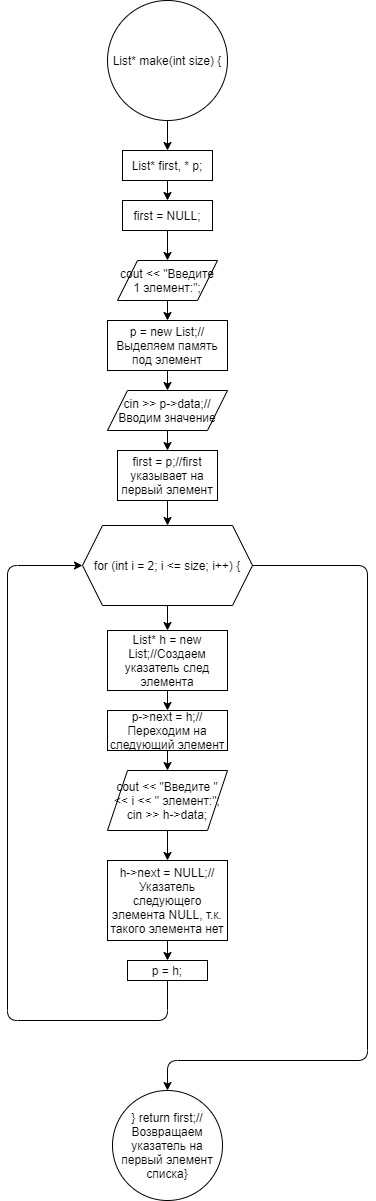
**5.1.** Для ввода данных используется цикл for, принимающий значения переменных вплоть до введенного пользователем размера, благодаря последовательному присвоению введенным элементам указатель на последующий элемент:

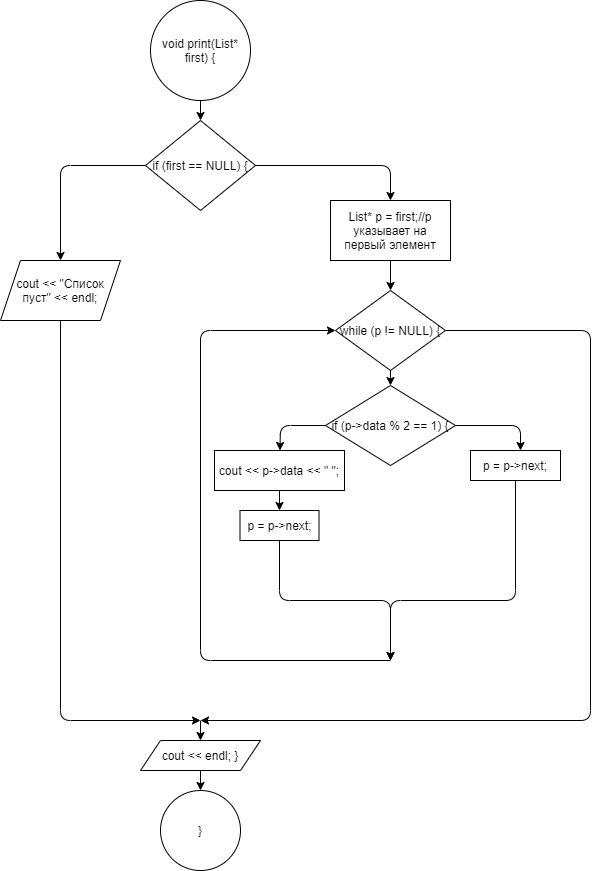


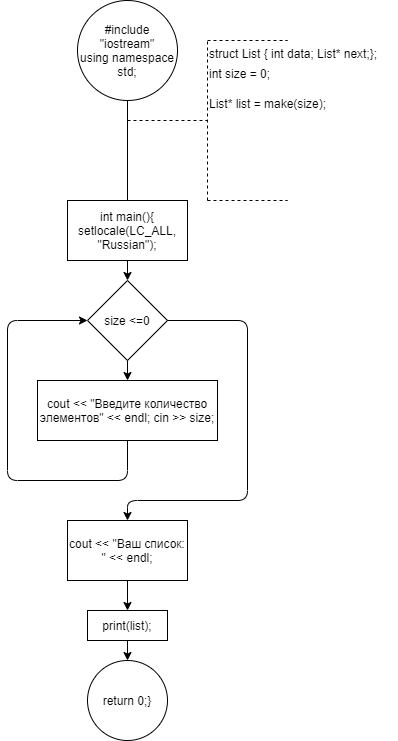
**5.2.**Вывод элементов списка реализован через функцию print:



**Блок-схема программы**







**Решение**

#include "iostream"

using namespace std;

struct List {

int data;

List\* next;

};

List\* make(int size) {

List\* first, \* p;

first = NULL;

cout << "Введите 1 элемент:";

p = new List;//Выделяем память под элемент

cin >> p->data;//Вводим значение

first = p;//first указывает на первый элемент

for (int i = 2; i <= size; i++) {

List\* h = new List;//Cоздаем указатель след элемента

p->next = h;//Переходим на следующий элемент

cout << "Введите " << i << " элемент:";

cin >> h->data;

h->next = NULL;//Указатель следующего элемента NULL, т.к. такого элемента нет

p = h;

}

return first;//Возвращаем указатель на первый элемент списка

}

void print(List\* first) {

if (first == NULL) {

cout << "Cписок пуст" << endl;

}

else {

List\* p = first;//p указывает на первый элемент

while (p != NULL) {

if (p->data % 2 == 1) {

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

p = p->next;

}

else {

p = p->next;

}

}

cout << endl;

}

}

int main(){

setlocale(LC\_ALL, "Russian");

int size = 0;

while(size <=0) {

cout << "Введите количество элементов" << endl;

cin >> size;

}

List\* list = make(size);

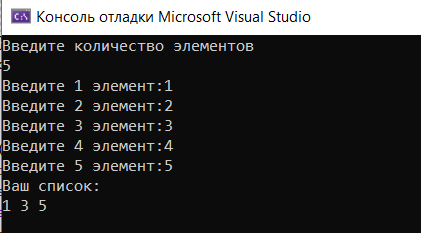
cout << "Ваш список: " << endl;

print(list);

return 0;

}

**Скриншоты результатов работы программы**

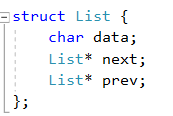


2)Двунаправленный список:

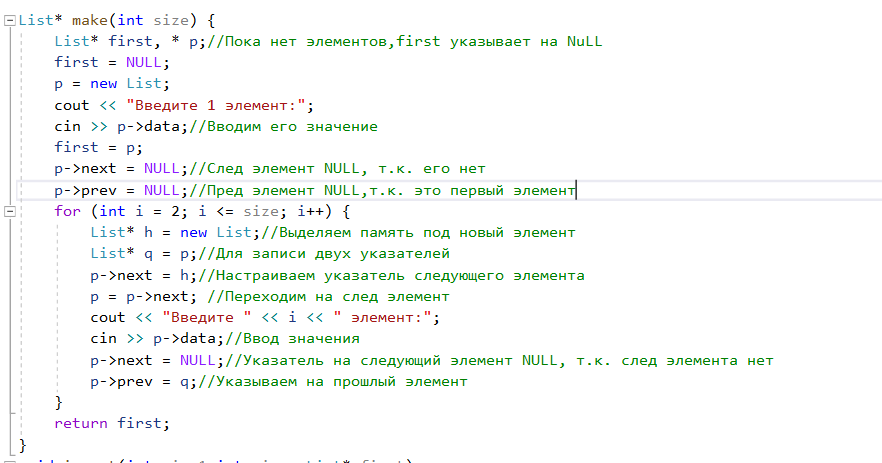
**Анализ задач**

**1.** Для решения задачи необходимо:

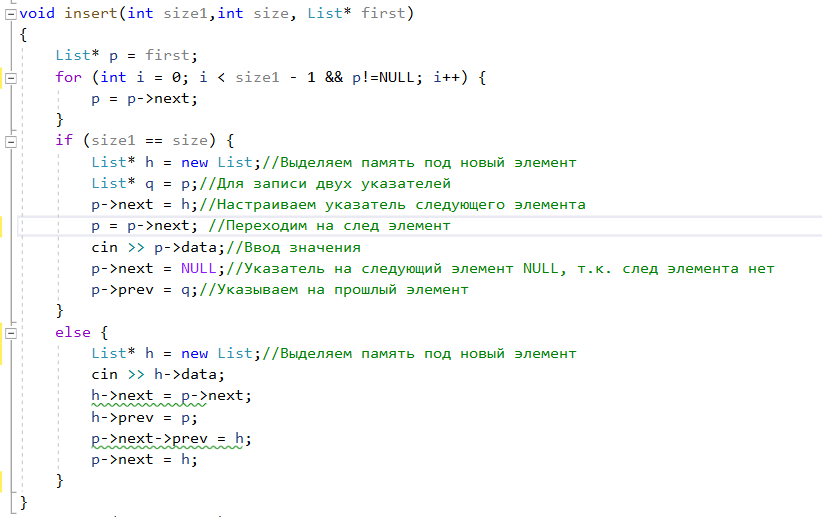
1.1.Создать структуру переменных, с указанием на последующий и предыдущий элементы:



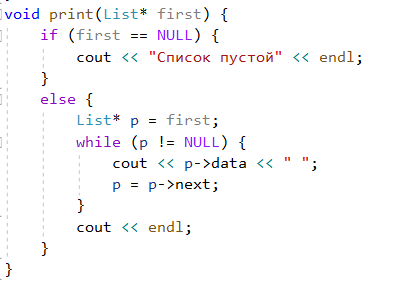
**1.2.** Разработать функцию для создания двунаправленного списка:



**1.3.**Разработать функцию insert для вставки нового элемента:



**1.4.**Разработать функцию print вывода всех элементов списка :



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для функции make используется входная переменная size целого типа данных (int)



**2.2.** Для функции print используется указатель на первый элемент

\*first :



**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобной обработки элементов было решено представить элементы в виде двунаправленного списка

**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Ввод данных происходит благодаря функции cin:

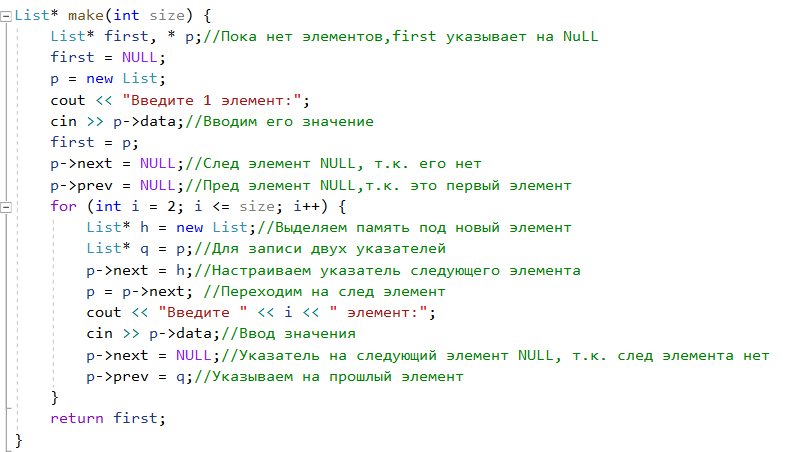


**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

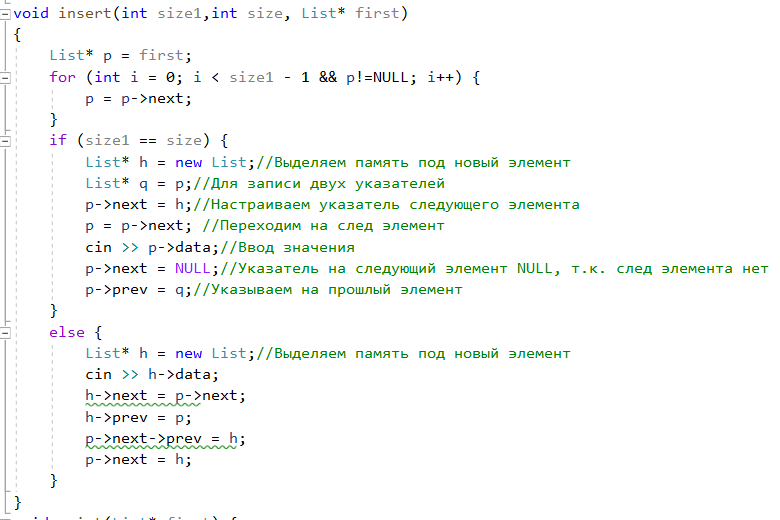


**5.**Поставленные задачи будут решены следующими действиями:

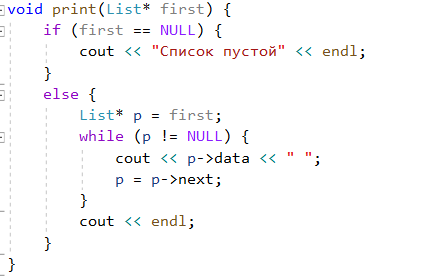
**5.1.** Создание двунаправленного списка происходит с создания первого элемента, предыдущим и последующим элементами которыми будет указание на значение NULL. Затем происходит последовательное добавление элементов к списку через цикл for, связывающий предыдущий элемент с новым и указывающий на последующий элемент.



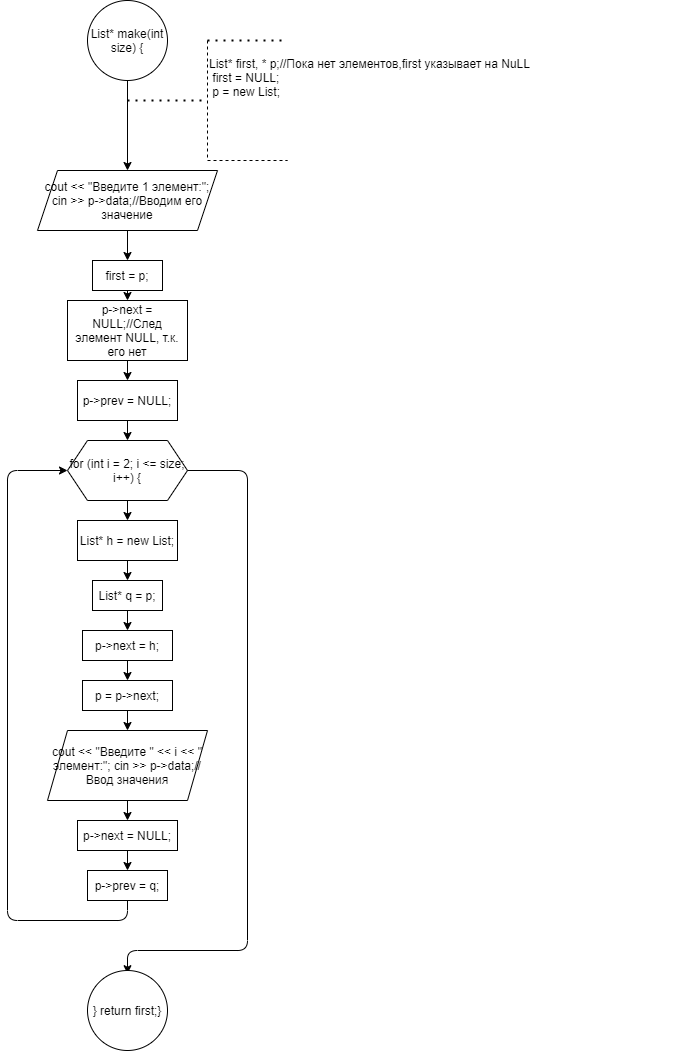
**5.2.**Вставка элементов реализована через функцию insert, перебирающую все элементы списка, пока не дойдет до нужного пользователем положения. Затем происходит выделение памяти под новый элемент и ее связка с предыдущим элементом списка и последующим (вставка нового элемента):

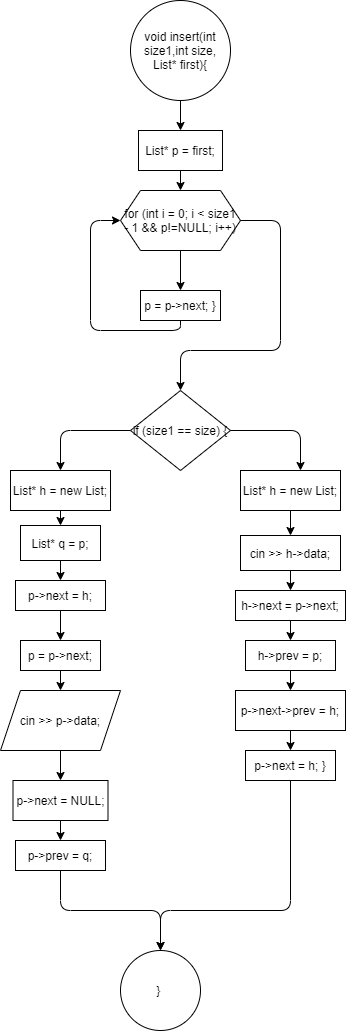


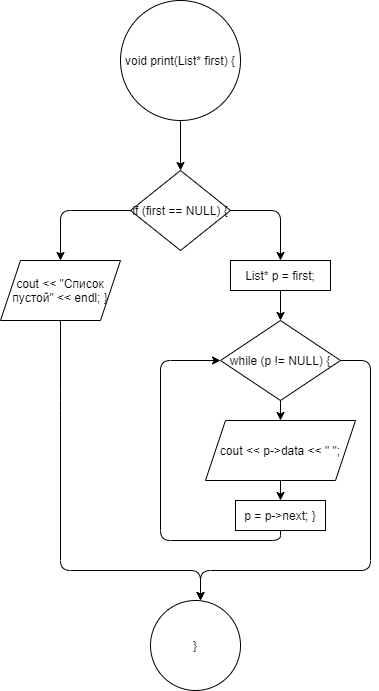
**5.3.** Вывод всех элементов списка происходит через функцию print, основанный на цикле while и выводящий элементы до тех пор, пока не встретит последний элемент со значением NULL:

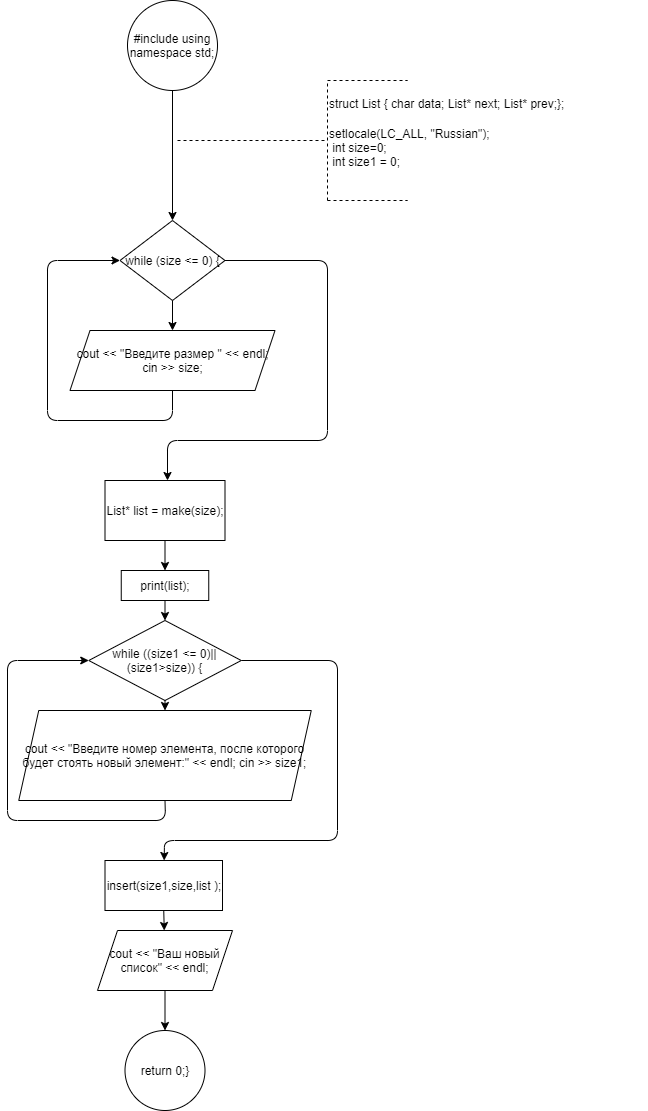


**Блок-схема программы**









**Решение**

#include <iostream>

using namespace std;

struct List {

char data;

List\* next;

List\* prev;

};

List\* make(int size) {

List\* first, \* p;//Пока нет элементов,first указывает на NuLL

first = NULL;

p = new List;

cout << "Введите 1 элемент:";

cin >> p->data;//Вводим его значение

first = p;

p->next = NULL;//Cлед элемент NULL, т.к. его нет

p->prev = NULL;//Пред элемент NULL,т.к. это первый элемент

for (int i = 2; i <= size; i++) {

List\* h = new List;//Выделяем память под новый элемент

List\* q = p;//Для записи двух указателей

p->next = h;//Настраиваем указатель следующего элемента

p = p->next; //Переходим на след элемент

cout << "Введите " << i << " элемент:";

cin >> p->data;//Ввод значения

p->next = NULL;//Указатель на следующий элемент NULL, т.к. след элемента нет

p->prev = q;//Указываем на прошлый элемент

}

return first;

}

void insert(int size1,int size, List\* first)

{

List\* p = first;

for (int i = 0; i < size1 - 1 && p!=NULL; i++) {

p = p->next;

}

if (size1 == size) {

List\* h = new List;//Выделяем память под новый элемент

List\* q = p;//Для записи двух указателей

p->next = h;//Настраиваем указатель следующего элемента

p = p->next; //Переходим на след элемент

cin >> p->data;//Ввод значения

p->next = NULL;//Указатель на следующий элемент NULL, т.к. след элемента нет

p->prev = q;//Указываем на прошлый элемент

}

else {

List\* h = new List;//Выделяем память под новый элемент

cin >> h->data;

h->next = p->next;

h->prev = p;

p->next->prev = h;

p->next = h;

}

}

void print(List\* first) {

if (first == NULL) {

cout << "Список пустой" << endl;

}

else {

List\* p = first;

while (p != NULL) {

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

p = p->next;

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int size=0;

int size1 = 0;

while (size <= 0) {

cout << "Введите размер " << endl;

cin >> size;

}

List\* list = make(size);

print(list);

while ((size1 <= 0)||(size1>size)) {

cout << "Введите номер элемента, после которого будет стоять новый элемент:" << endl;

cin >> size1;

}

insert(size1,size,list );

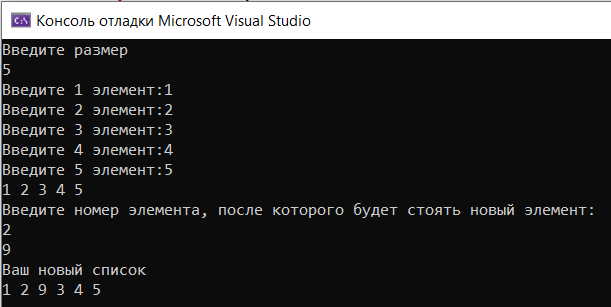
cout << "Ваш новый список" << endl;

print(list);

return 0;

}

**Скриншоты результатов работы программы**



3)Двунаправленный список (List):

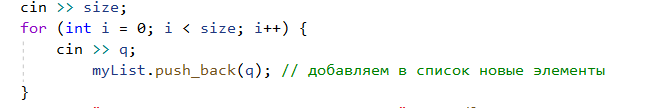
**Анализ задач**

**1.** Для решения задачи необходимо:

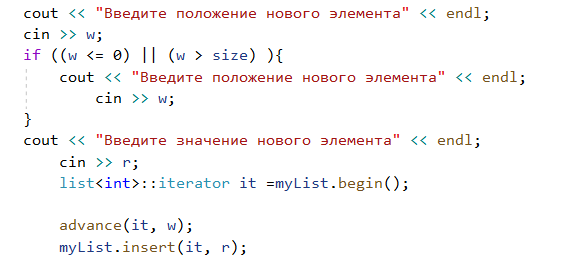
1.1.Объявить пустой список:



**1.2.** Поочередно вставить элементы в список:



**1.3.**Добраться до нужного элемента списка и вставить после него новый элемент:



**1.4.**Вывести на экранвсе элементы списка:



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Для функции int main использовались переменные типа данных int:



**3.**Для решения задачи данные были представлены в следующем виде:

**3.1.** Для удобной обработки элементов было решено представить элементы в виде двунаправленного списка

**4.**Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.**Ввод данных происходит благодаря функции cin:

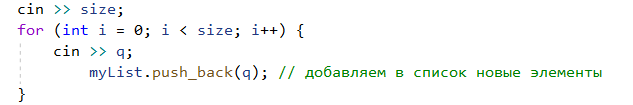


**4.2.** Вывод данных на консоль реализован с помощью оператора cout и copy.

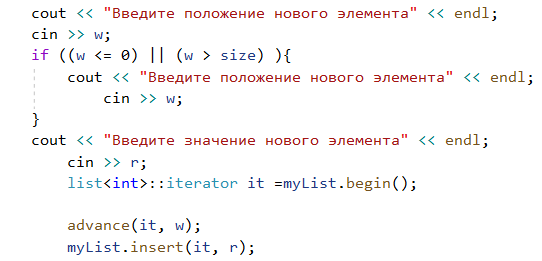


**5.**Поставленные задачи будут решены следующими действиями:

**5.1.** Создание двунаправленного списка начинается с его объявления, с последующим добавлением элементов в конец списка через функцию push\_back:



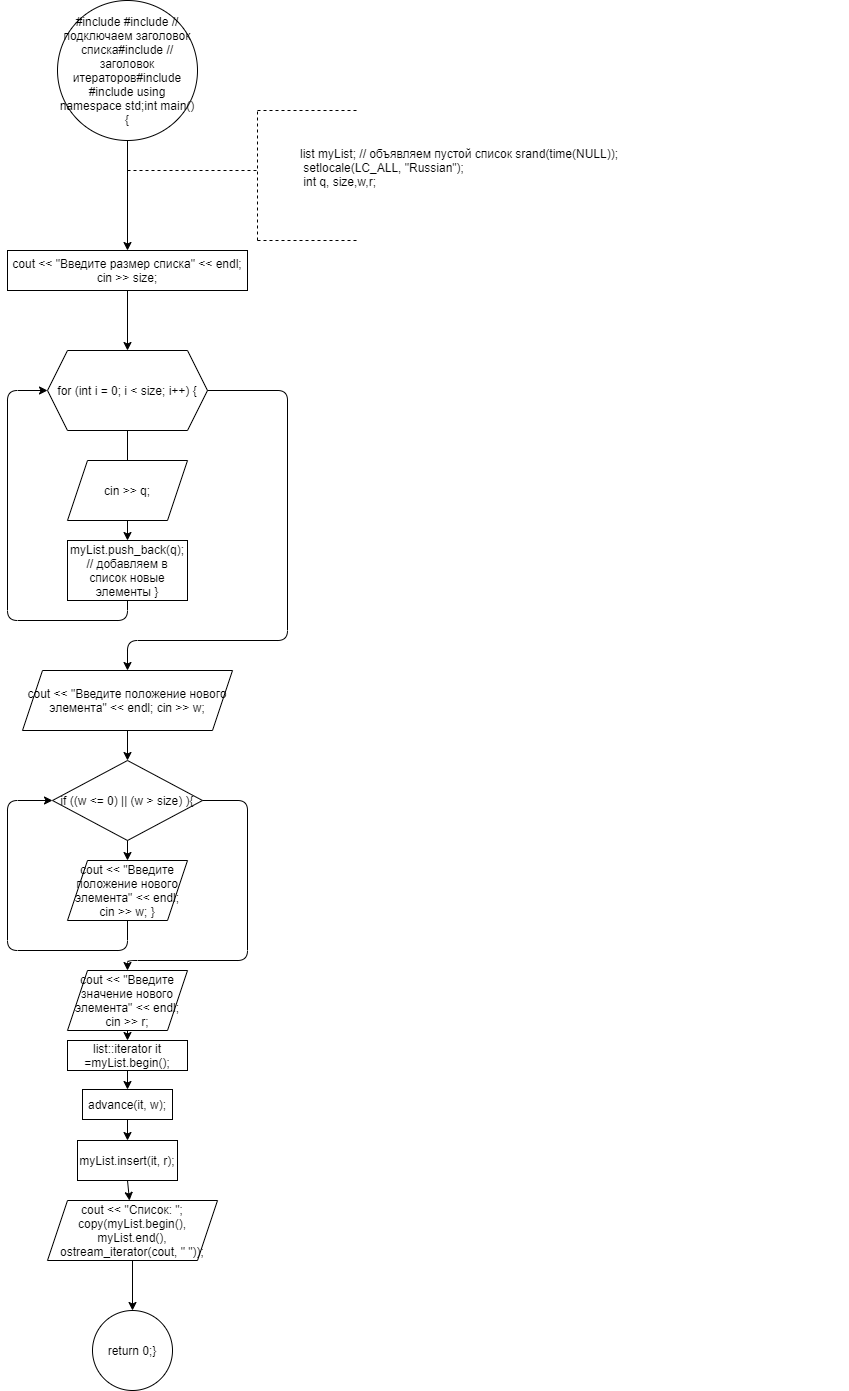
**5.2.**Вставка элементов реализована через функцию insert,принимающий положение нового элемента и его значение.



**5.3.** Вывод всех элементов списка происходит через функцию copy, принимающий начало и конец выводимых элементов:



**Блок-схема программы**



**Решение**

#include <iostream>

#include <list> // подключаем заголовок списка

#include <iterator> // заголовок итераторов

#include <ctime>

#include <cstdlib>

using namespace std;

int main()

{

list<int> myList; // объявляем пустой список

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

int q, size,w,r;

cout << "Введите размер списка" << endl;

cin >> size;

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

cin >> q;

myList.push\_back(q); // добавляем в список новые элементы

}

cout << "Введите положение нового элемента" << endl;

cin >> w;

if ((w <= 0) || (w > size) ){

cout << "Введите положение нового элемента" << endl;

cin >> w;

}

cout << "Введите значение нового элемента" << endl;

cin >> r;

list<int>::iterator it =myList.begin();

advance(it, w);

myList.insert(it, r);

cout << "Список: ";

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

return 0;

}

**Скриншоты результатов работы программы**

