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 21

Выполнил:

Студент группы РИС-20-1б

Ознобихин Елисей Андреевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

1) Получить практические навыки работы с однонаправленными списками;

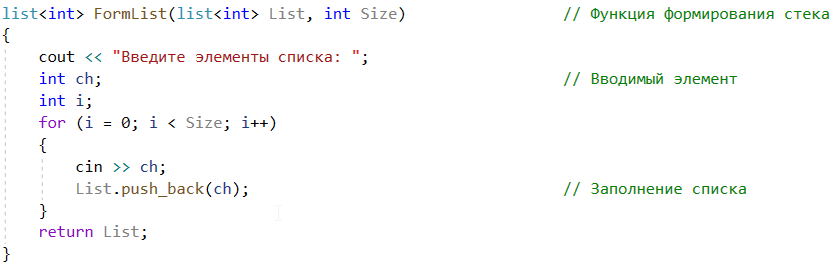
2) Получить практические навыки работы с двунаправленными списками;

**Постановка задачи**

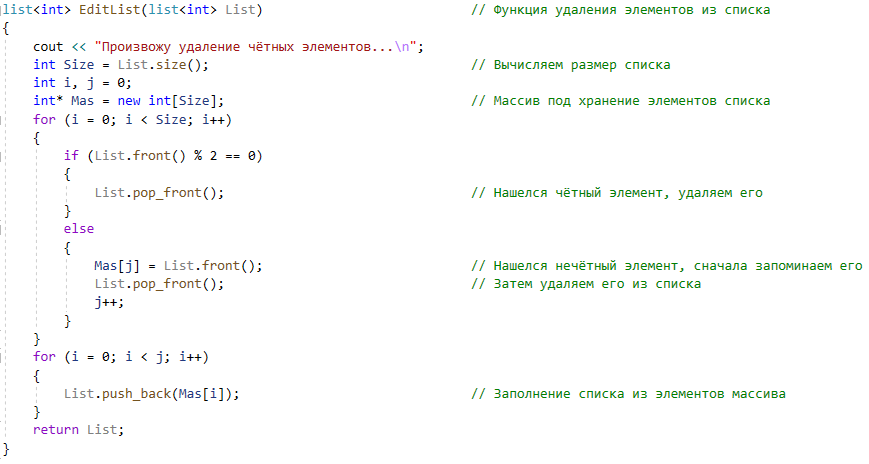
1. Тип информационного поля char\*. Добавить в список элемент после элемента с заданным информационным полем.
2. Тип информационного поля int. Удалить из списка все элементы с четными информационными полями.

**STL-библиотека**

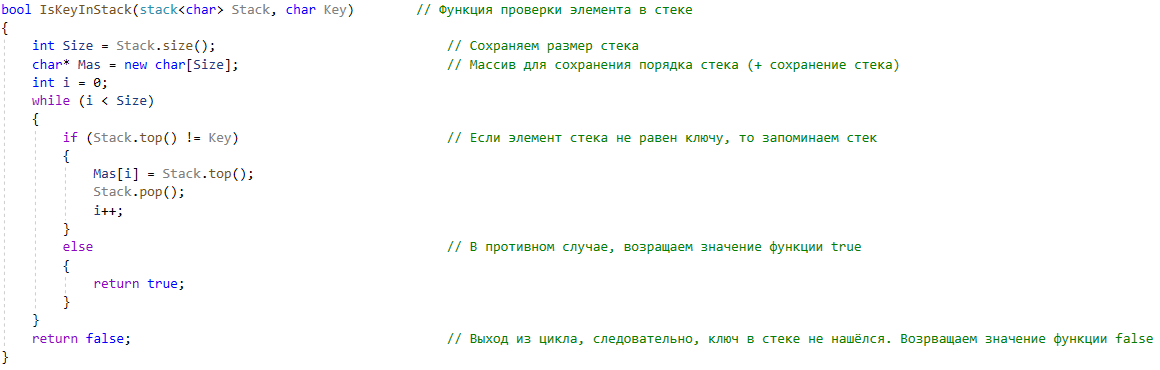
1. Для решения задачи необходимо:
   1. Разработать функцию для формирования списка через stl-библиотеки.



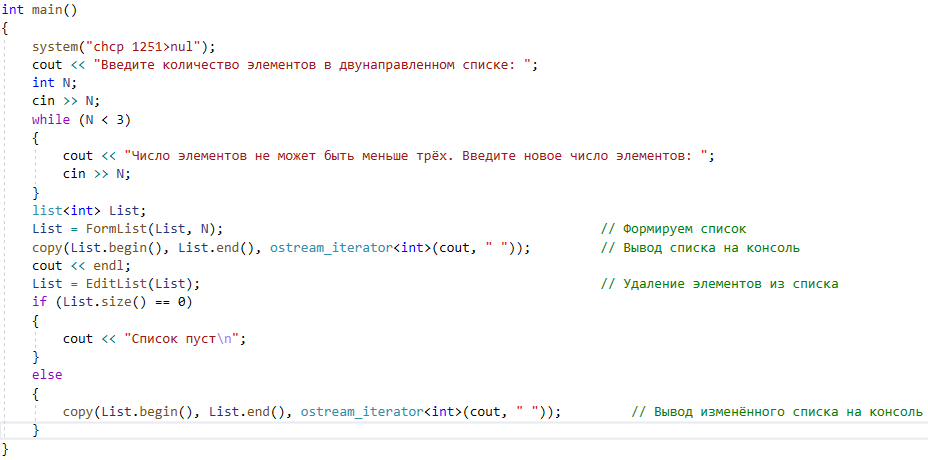
* 1. Разработать функцию для удаления элементов с чётными информационными полями из списка.



* 1. Разработать функцию для нахождения ключа в стеке.



* 1. Реализовать применение этих функций в программе.



1. В ходе работы были использованы типы данных:
   1. Для функции FormList использовались следующие аргументы:
   2. Тип list<int>: двунаправленный список.
   3. Тип int: размер списка.

Сама функция имеет тип list<int> и возвращает список.



* 1. Для функции EditList использовались следующие аргументы:
     + 1. Тип list<int>: двунаправленный список.

Сама функция имеет тип list<int> и возвращает список.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_jgaHWcz4OC.png

1. Для решения задачи данные были представлены в следующем виде:
   1. Для осуществления работы данные были записаны ручным вводом с консоли в виде структур: однонаправленные и двунаправленные списки, стек и очередь.
2. Для операций ввода и вывода данных использовались следующие операторы и функции:
   1. Для вывода информации на консоль использовался оператор cout.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_VuFYbpFQVe.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_wRxP2lZFTk.png

* 1. Для ввода данных через консоль использовался оператор cin.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_skQj1fzU4u.png

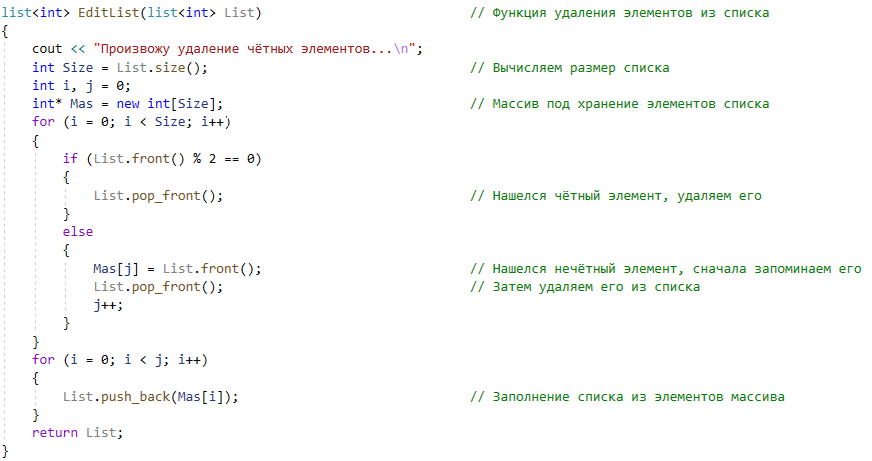
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_d5CIhDC2W5.png

1. Поставленные задачи решены следующими действиями:
   1. Все действия со списком/ со стеком/ с очередью реализованы через написанные вручную функции.

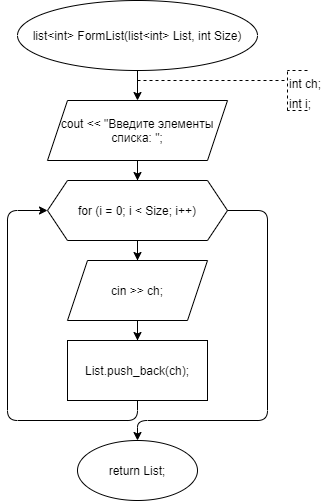


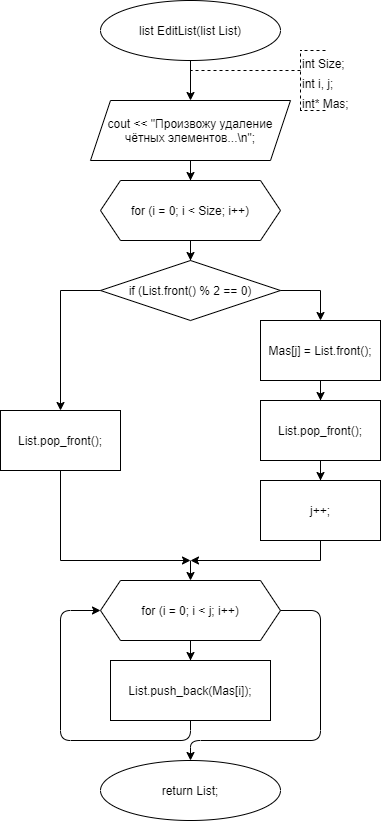
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_jgaHWcz4OC.png

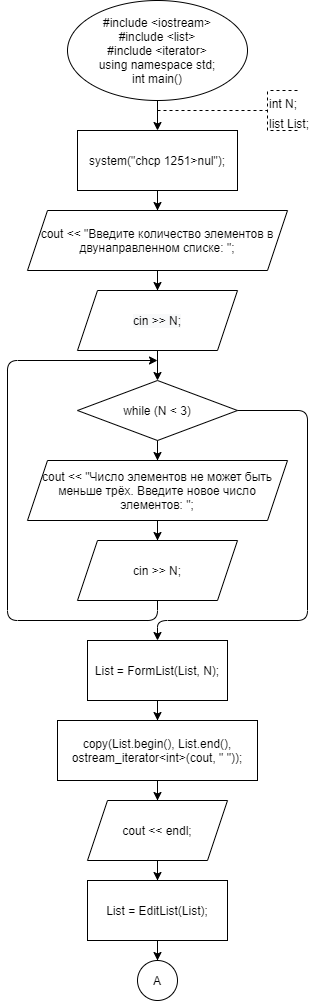
* 1. Иллюзия удаления элементов из списка реализовано с помощью цикла while, условного оператора if и дополнительного массива элементов. В массив записываются нечётные элементы и происходит их удаление из очереди, чётные же удаляются из очереди и не заносятся в массив.

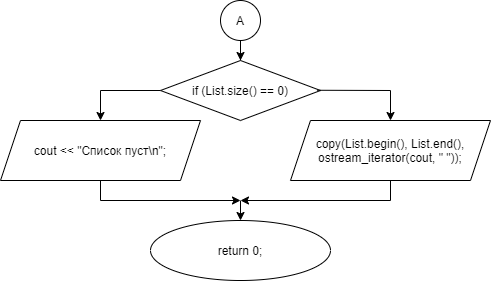


**Блок-схема программы**









**Реализация задачи на языке С++**

#include <iostream>

#include <list>

#include <iterator>

using namespace std;

list<int> FormList(list<int> List, int Size)

{

cout << "Введите элементы списка: ";

int ch;

int i;

for (i = 0; i < Size; i++)

{

cin >> ch;

List.push\_back(ch);

}

return List;

}

list<int> EditList(list<int> List)

{

cout << "Произвожу удаление чётных элементов...\n";

int Size = List.size();

int i, j = 0;

int\* Mas = new int[Size];

for (i = 0; i < Size; i++)

{

if (List.front() % 2 == 0)

{

List.pop\_front();

}

else

{

Mas[j] = List.front();

List.pop\_front();

j++;

}

}

for (i = 0; i < j; i++)

{

List.push\_back(Mas[i]);

return List;

}

int main()

{

system("chcp 1251>nul");

cout << "Введите количество элементов в двунаправленном списке: ";

int N;

cin >> N;

while (N < 3)

{

cout << "Число элементов не может быть меньше трёх. Введите новое число элементов: ";

cin >> N;

}

list<int> List;

List = FormList(List, N);

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

cout << endl;

List = EditList(List);

if (List.size() == 0)

{

cout << "Список пуст\n";

}

else

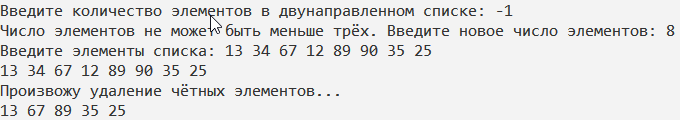
{

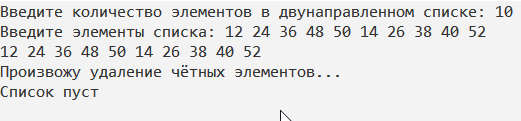
copy(List.begin(), List.end(), ostream\_iterator<int>(cout, " "));

}

}

**Скриншоты результатов работы программы**

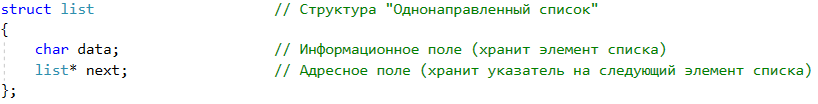
****

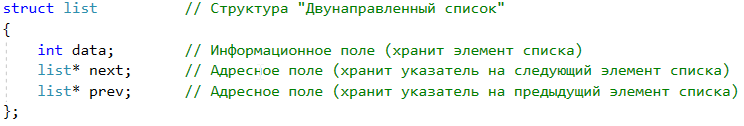
****

**Cтруктура**

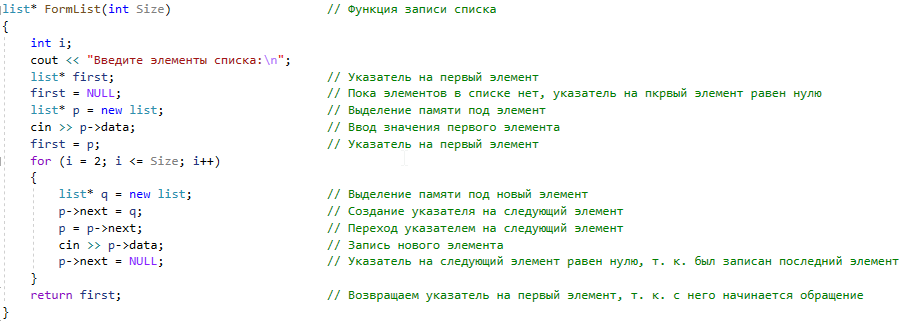
**Анализ задачи**

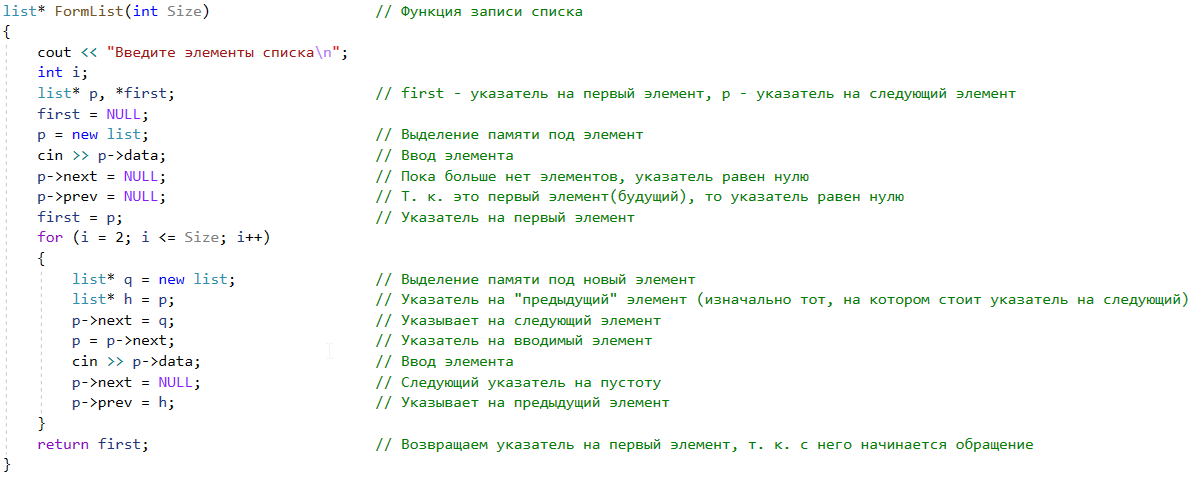
1. Для решения задачи необходимо:
   1. Зарезервировать память под структуру, содержащую информационное поле и адресное поле-указатель на следующий/предыдущий элемент.



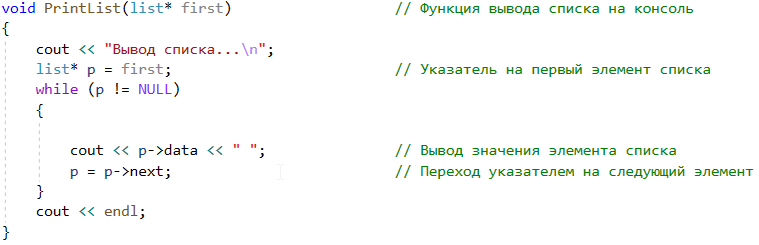


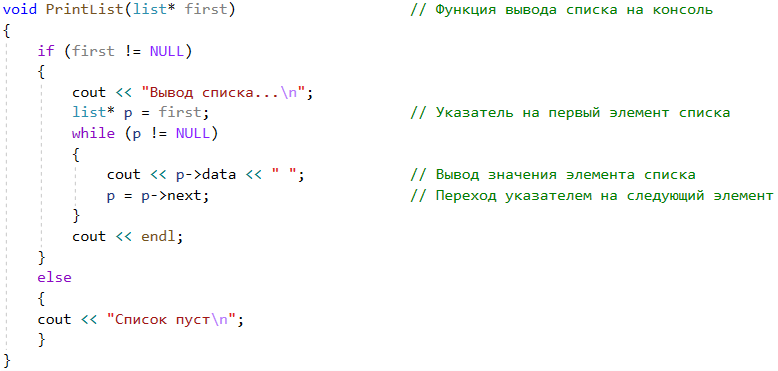
* 1. Разработать функцию для формирования однонаправленного списка/ двунаправленного списка через структуру.



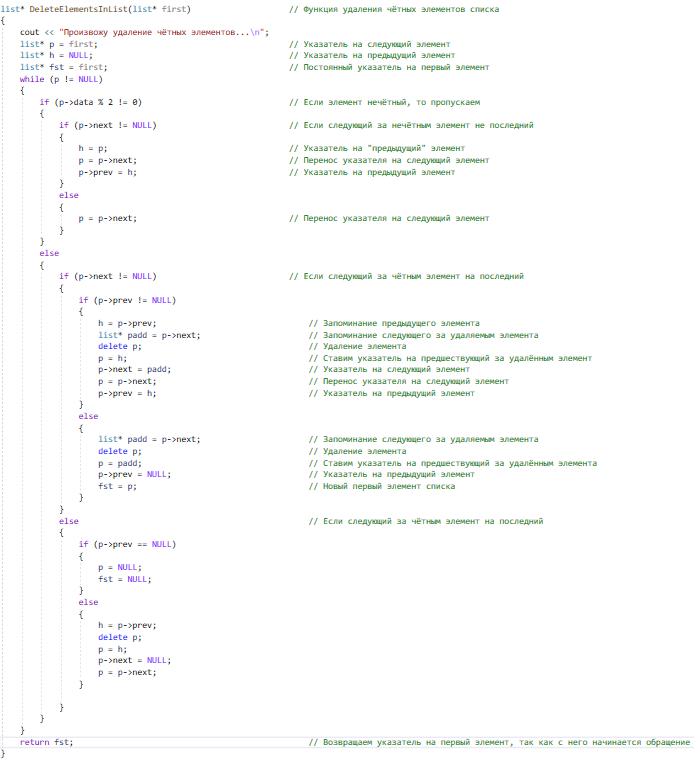


* 1. Разработать функцию для вывода информационного поля однонаправленного списка/ двунаправленного списка на экран через консоль.

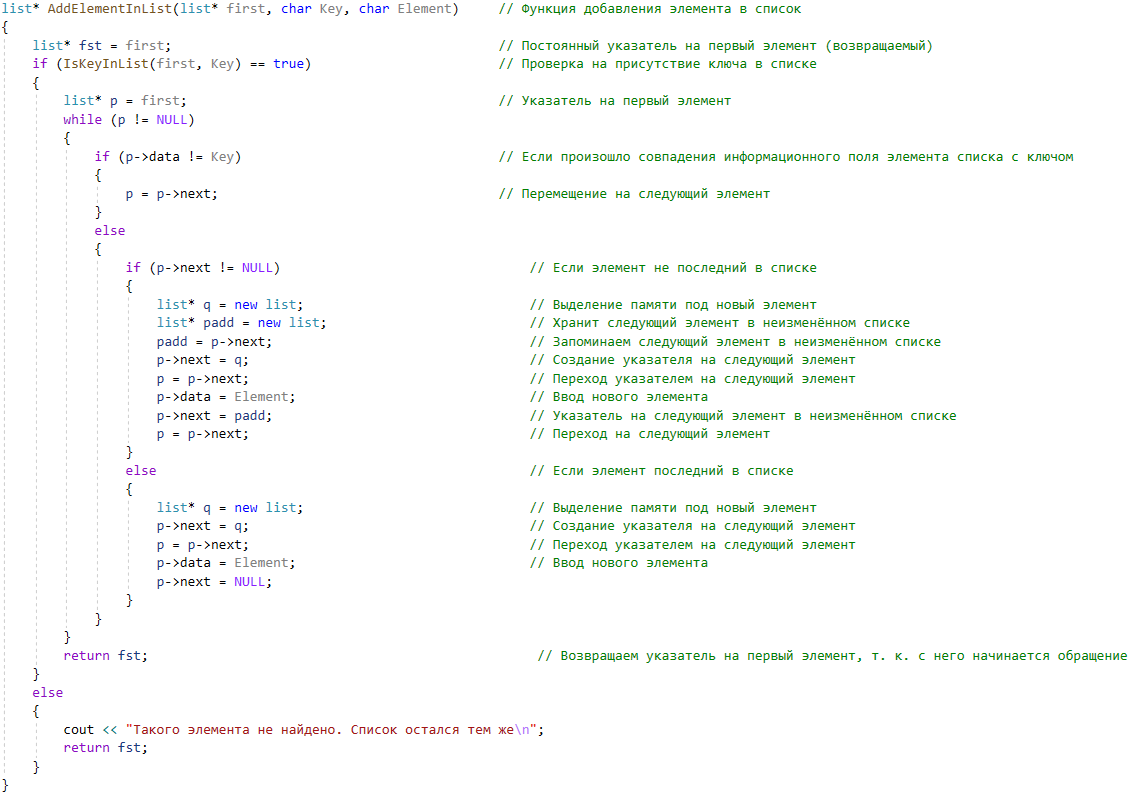




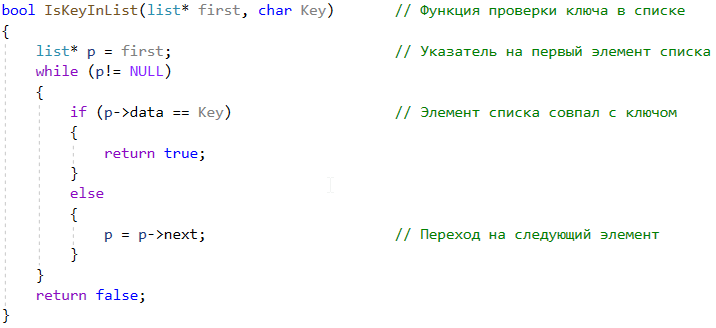
* 1. Разработать функцию для удаления элементов с чётными информационными полями из двунаправленного списка.



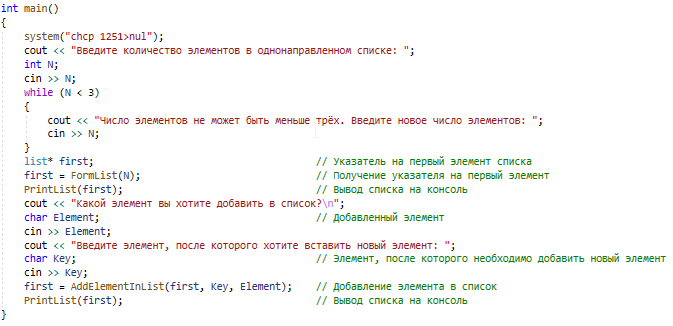
* 1. Разработать функцию для добавления нового элемента после определённого в однонаправленный список.

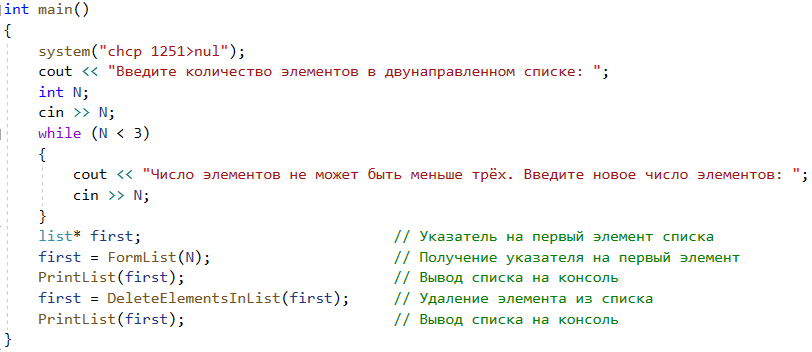


* 1. Разработать функцию для нахождения ключа в однонаправленном списке.



* 1. Реализовать применение этих функций в программе.





1. В ходе работы были использованы типы данных:
   1. Для функции FormList/ FormList использовались следующие аргументы:
   2. Тип int: Размер очереди

Сама функция имеет тип указателя на list/ list/ stack/ queue и возвращает указатель на начало списка/ списка/ очереди/ вершину стека.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_KrW5aRKjeh.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_KrW5aRKjeh.png

(стек)

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_qUUbea2vQp.png

* 1. Для функции PrintList/ PrintList/ PrintStack/ PrintQueue использовались следующие аргументы:
  2. Указатель на list/ list: Указатель на первый элемент однонаправленного списка/ двунаправленного списка.

Сама функция имеет тип void, поскольку при работе функции ничего не нужно возвращать.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_m8SLNeeUQI.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_m8SLNeeUQI.png

* 1. Для функции DeleteElementsInList использовались следующие аргументы:
     + 1. Указатель на queue: Указатель на начало двунаправленного списка.

Сама функция имеет тип указателя на list и возвращает указатель на первый элемент двунаправленного списка

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_IBuiCW1T6f.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_vmSHdtQARa.png

* 1. Для функции AddElementInList использовались следующие аргументы:
  2. Указатель на list: указатель на первый элемент списка.
  3. Тип char: ключ поиска элемента в списке.
  4. Тип char: добавленный элемент в список.

Сама функция имеет тип указателя на list и возвращает указатель на первый элемент однонаправленного списка.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_q7iffeucxQ.png

* 1. Для функции IsKeyInList использовались следующие аргументы:
  2. Указатель на list: указатель на первый элемент списка.
  3. Тип char: ключ поиска элемента в списке.

Сама функция имеет тип bool и возвращает значение истина, если ключ нашелся в списке, или ложь, если ключа в списке нет.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_0PqHbJBAsY.png

1. Для решения задачи данные были представлены в следующем виде:
   1. Для осуществления работы данные были записаны ручным вводом с консоли в виде структур: однонаправленные и двунаправленные списки, стек и очередь.
2. Для операций ввода и вывода данных использовались следующие операторы и функции:
   1. Для вывода информации на консоль использовался оператор cout.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_NZuMkD0GWK.png

* 1. Для ввода данных через консоль использовался оператор cin.



1. Поставленные задачи решены следующими действиями:
   1. Все действия со списком/ со списком/ со стеком/ с очередью реализованы через написанные вручную функции.

Однонаправленный список

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_IBuiCW1T6f.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_0PqHbJBAsY.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_m8SLNeeUQI.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_KrW5aRKjeh.png

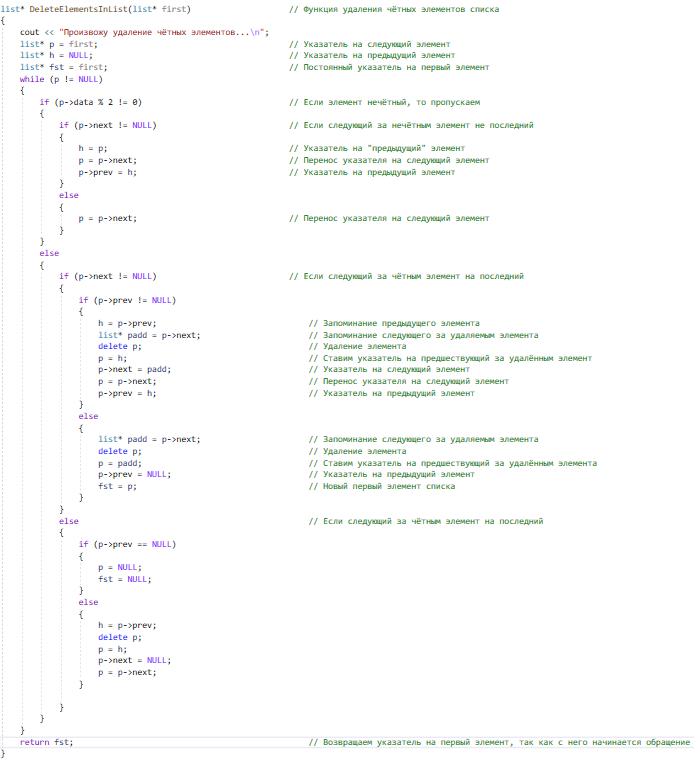
Двунаправленный список

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_q7iffeucxQ.png

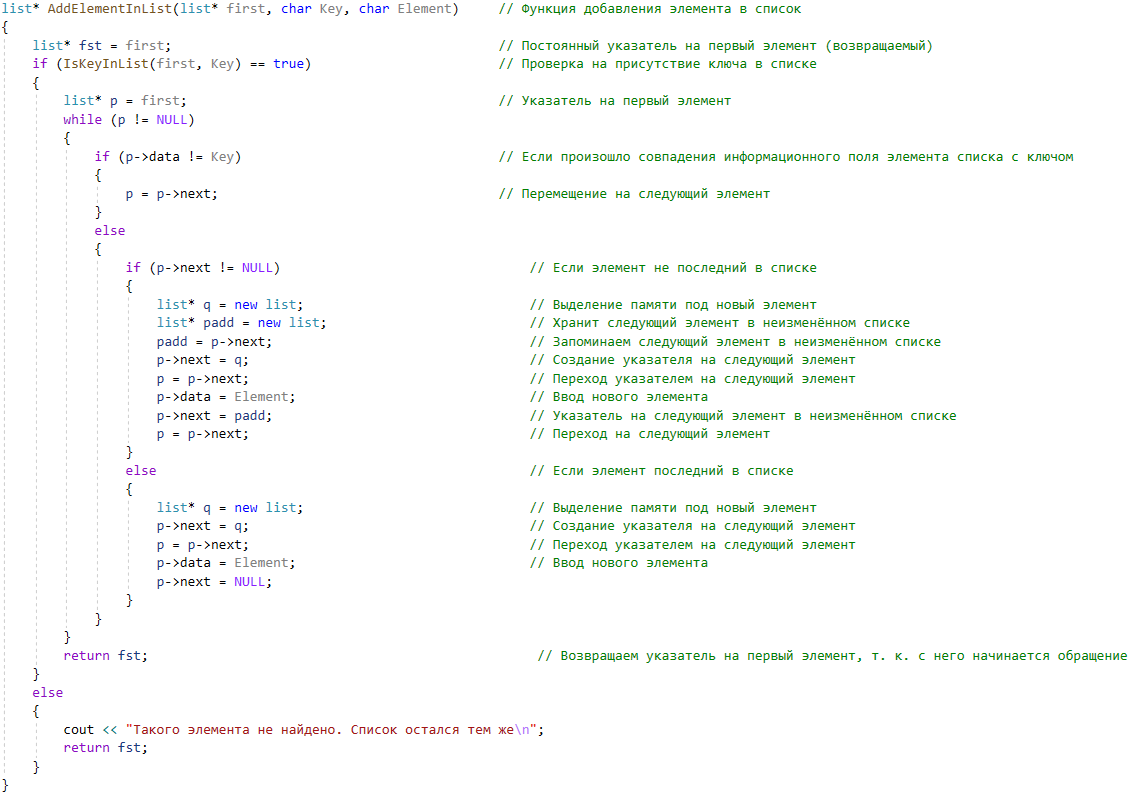
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_m8SLNeeUQI.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_KrW5aRKjeh.png

* 1. Удаление элементов из двунаправленного списка реализовано с помощью цикла while. Каждый элемент списка проверяется условным оператором на делимость на 2, и в случае истины удаляет элемент из списка. В противном случае элемент списка игнорируется и проверяется следующий за ним элемент. Цикл while продолжается до тех пор, пока не прошлись по всей длине списка.

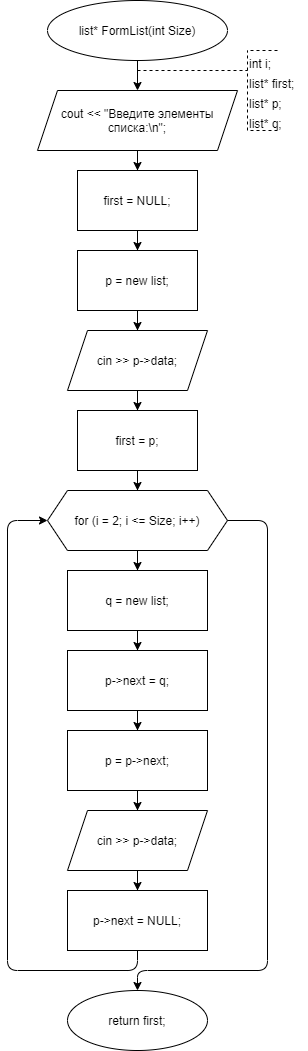


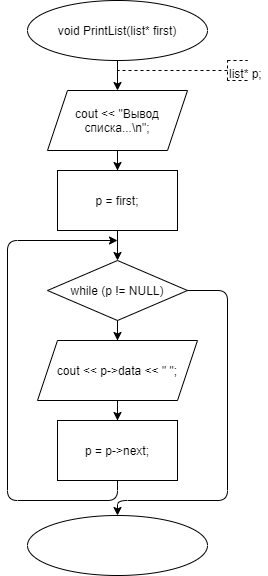
* 1. Добавление элемента после определённого элемента в однонаправленном списке реализовано с помощью цикла while и вспомогательных функций поиска ключа во множестве.

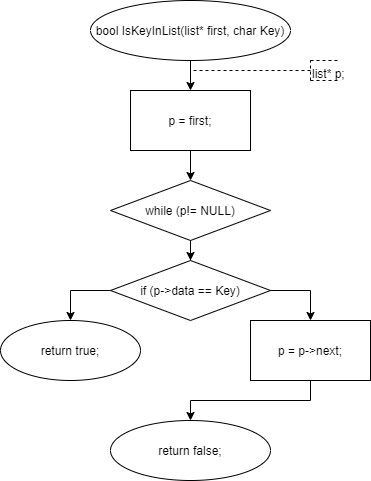


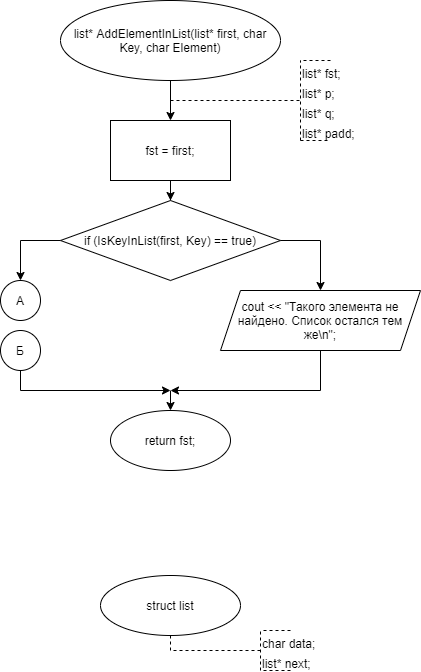
**Блок-схема программы**

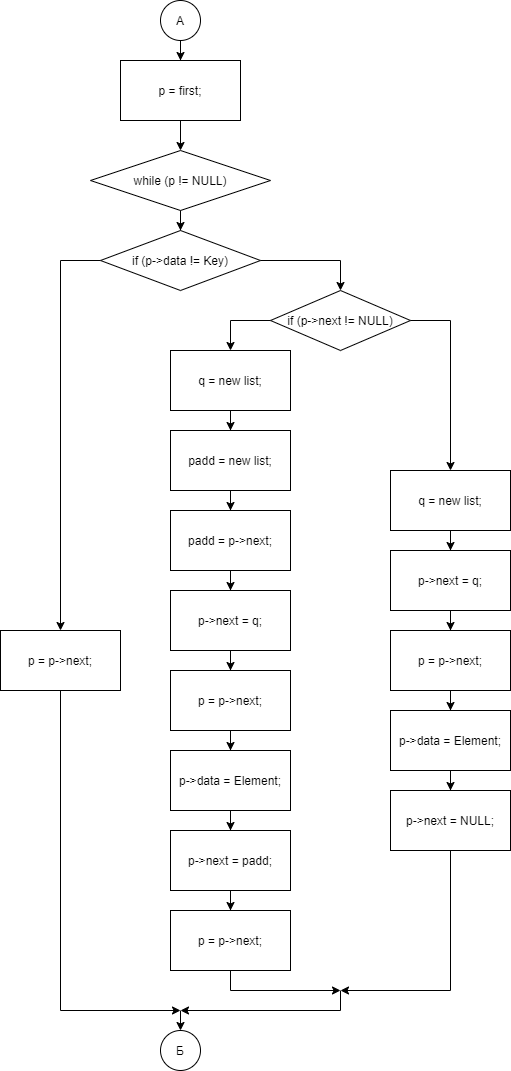
Однонаправленный список

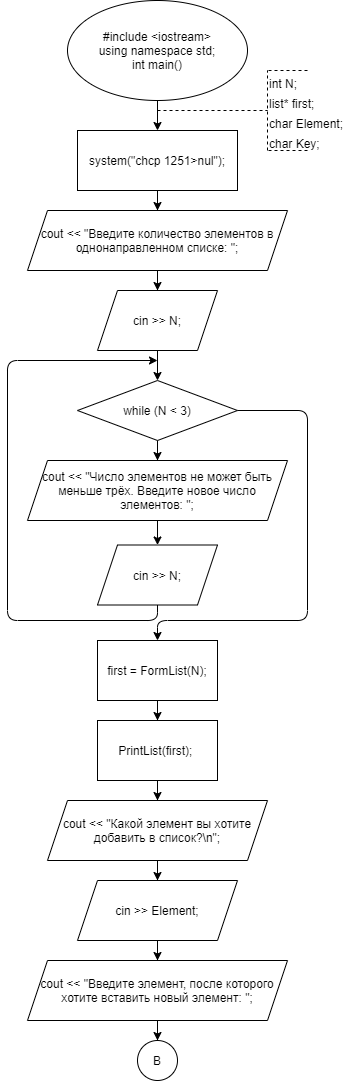


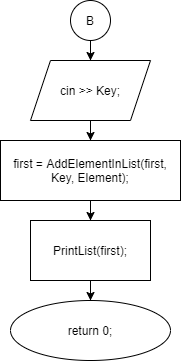




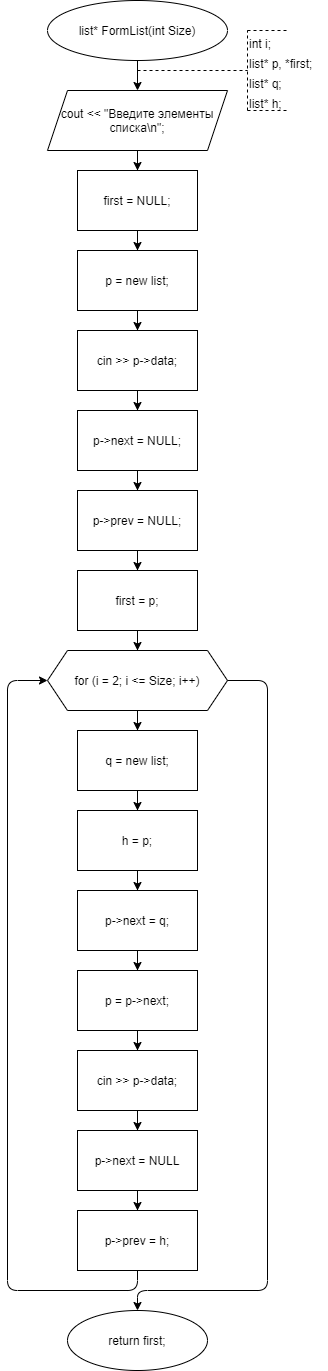


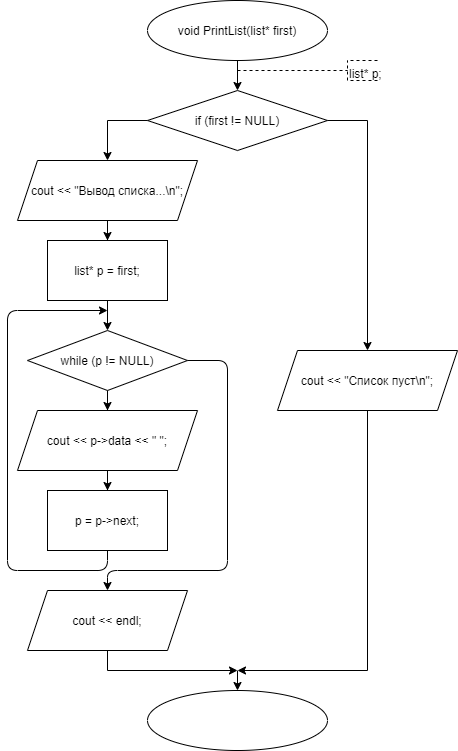


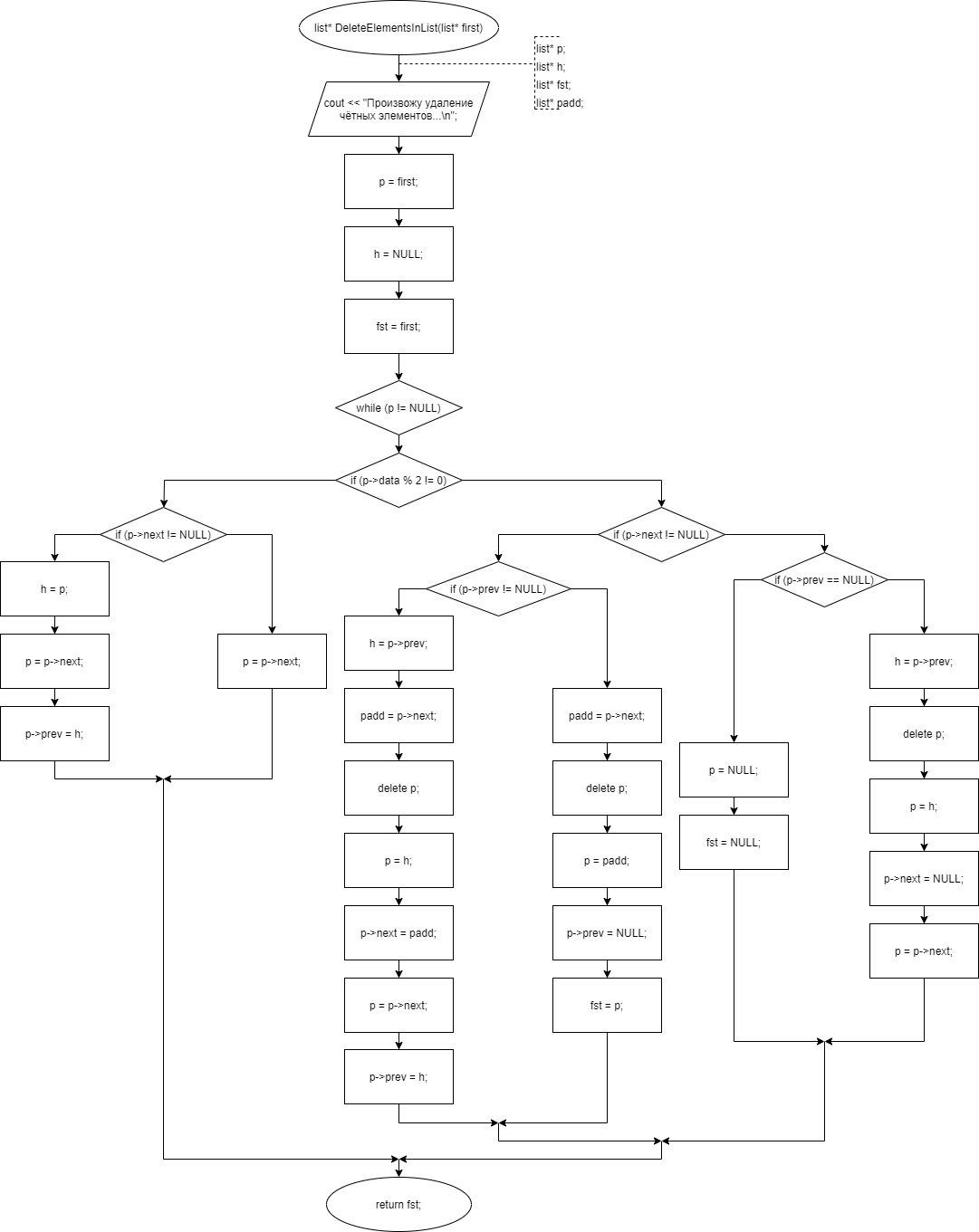


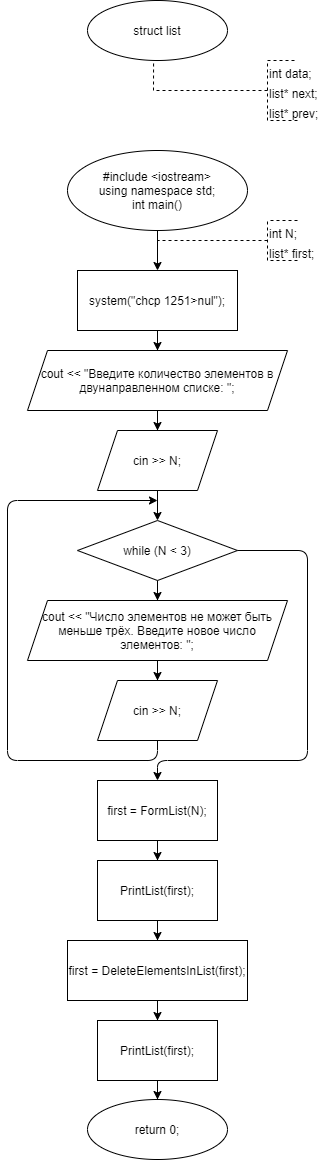


Двунаправленный список









**Реализация задачи на языке С++**

Однонаправленный список

#include <iostream>

using namespace std;

struct list

{

char data;

list\* next;

};

list\* FormList(int Size)

{

int i;

cout << "Введите элементы списка:\n";

list\* first;

first = NULL;

list\* p = new list;

cin >> p->data;

first = p;

for (i = 2; i <= Size; i++)

{

list\* q = new list;

p->next = q;

p = p->next;

cin >> p->data;

p->next = NULL;

}

return first;

}

void PrintList(list\* first)

{

cout << "Вывод списка...\n";

list\* p = first;

while (p != NULL)

{

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

p = p->next;

}

cout << endl;

}

bool IsKeyInList(list\* first, char Key)

{

list\* p = first;

while (p!= NULL)

{

if (p->data == Key)

{

return true;

}

else

{

p = p->next;

}

}

return false;

}

list\* AddElementInList(list\* first, char Key, char Element)

{

list\* fst = first;

if (IsKeyInList(first, Key) == true)

{

list\* p = first;

while (p != NULL)

{

if (p->data != Key)

{

p = p->next;

}

else

{

if (p->next != NULL)

{

list\* q = new list;

list\* padd = new list;

padd = p->next;

p->next = q;

p = p->next;

p->data = Element;

p->next = padd;

p = p->next;

}

else

{

list\* q = new list;

p->next = q;

p = p->next;

p->data = Element;

p->next = NULL;

}

}

}

return fst;

}

else

{

cout << "Такого элемента не найдено. Список остался тем же\n";

return fst;

}

}

int main()

{

system("chcp 1251>nul");

cout << "Введите количество элементов в однонаправленном списке: ";

int N;

cin >> N;

while (N < 3)

{

cout << "Число элементов не может быть меньше трёх. Введите новое число элементов: ";

cin >> N;

}

list\* first;

first = FormList(N);

PrintList(first);

cout << "Какой элемент вы хотите добавить в список?\n";

char Element;

cin >> Element;

cout << "Введите элемент, после которого хотите вставить новый элемент: ";

char Key;

cin >> Key;

first = AddElementInList(first, Key, Element);

PrintList(first);

}

Двунаправленный список

#include <iostream>

using namespace std;

struct list

{

int data;

list\* next;

list\* prev;

};

list\* FormList(int Size)

{

cout << "Введите элементы списка\n";

int i;

list\* p, \*first;

first = NULL;

p = new list;

cin >> p->data;

p->next = NULL;

p->prev = NULL;

first = p;

for (i = 2; i <= Size; i++)

{

list\* q = new list;

list\* h = p;

p->next = q;

p = p->next;

cin >> p->data;

p->next = NULL;

p->prev = h;

}

return first;

}

void PrintList(list\* first)

{

if (first != NULL)

{

cout << "Вывод списка...\n";

list\* p = first;

while (p != NULL)

{

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

p = p->next;

}

cout << endl;

}

else

{

cout << "Список пуст\n";

}

}

list\* DeleteElementsInList(list\* first)

{

cout << "Произвожу удаление чётных элементов...\n";

list\* p = first;

list\* h = NULL;

list\* fst = first;

while (p != NULL)

{

if (p->data % 2 != 0)

{

if (p->next != NULL)

{

h = p;

p = p->next;

p->prev = h;

}

else

{

p = p->next;

}

}

else

{

if (p->next != NULL)

{

if (p->prev != NULL)

{

h = p->prev;

list\* padd = p->next;

delete p;

p = h;

p->next = padd;

p = p->next;

p->prev = h;

}

else

{

list\* padd = p->next;

delete p;

p = padd;

p->prev = NULL;

fst = p;

}

}

else

{

if (p->prev == NULL)

{

p = NULL;

fst = NULL;

}

else

{

h = p->prev;

delete p;

p = h;

p->next = NULL;

p = p->next;

}

}

}

}

return fst;

}

int main()

{

system("chcp 1251>nul");

cout << "Введите количество элементов в двунаправленном списке: ";

int N;

cin >> N;

while (N < 3)

{

cout << "Число элементов не может быть меньше трёх. Введите новое число элементов: ";

cin >> N;

}

list\* first;

first = FormList(N);

PrintList(first);

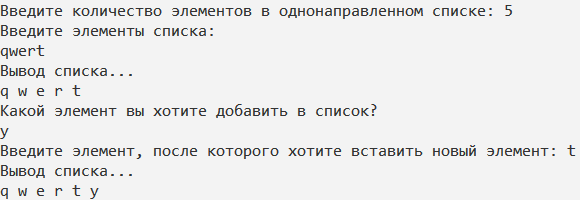
first = DeleteElementsInList(first);

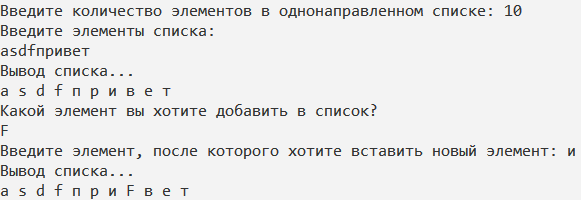
PrintList(first);

}

**Скриншоты результатов работы программы**

Однонаправленный список





Двунаправленный список

