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 11.2**

Дисциплина: Основы алгоритмизации и программирования

Вариант 15

Выполнила:

Студентка группы ИВТ-22-2б

Кучугова Яна Александровна

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2023**

**Постановка задачи**

Тип информационного поля int. Удалить из списка последний элемент с четным информационным полем.

1. Сформировать двунаправленный список. Тип информационного поля указан в варианте.
2. Распечатать полученную структуру.
3. Выполнить обработку структуры в соответствии с заданием.
4. Распечатать полученный результат.
5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

1. Создается структура двунаправленного списка:

struct DoubleList

{

int data;

DoubleList\* next;

DoubleList\* prev;

};

1. Создаются функции для заполнения списка:

void double\_list()

{

int numbersCount = -1;

int k, j = -1;

while (numbersCount <= 0)

{

cout << "Введите количество элементов списка: ";

cin >> numbersCount;

}

cout << "Введите элементы списка:" << endl;

for (int i = 0; i < numbersCount; i++)

{

cin >> k;

Add(k, i);

if (k % 2 == 0)

{

j = i + 1;

}

}

if (j == -1)

{

cout << "Четных элементов нет";

}

else

{

Delete(j);

cout << "Измененный список: ";

Print();

}

}

void Add(int value, int position)

{

DoubleList\* node = new DoubleList;

node->data = value;

if (head == NULL)

{

node->next = node;

node->prev = node;

head = node;

}

else

{

DoubleList\* p = head;

for (int i = position; i > 0; i--)

p = p->next;

p->prev->next = node;

node->prev = p->prev;

node->next = p;

p->prev = node;

}

}

1. Создаём функцию для удаления из списка последнего элемента с чётным информационным полем:

int Delete(int position)

{

if (head == NULL)

{

cout << "\nСписок пуст\n\n";

return 0;

}

if (head == head->next)

{

delete head;

head = NULL;

}

else

{

DoubleList\* a = head;

for (int i = position; i > 1; i--)

{

a = a->next;

}

if (a == head)

{

head = a->next;

}

a->prev->next = a->next;

a->next->prev = a->prev;

delete a;

}

}

1. Для вывода создается функция print:

void Print()

{

DoubleList\* a = head;

do

{

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

a = a->next;

} while (a != head);

cout << "\n\n";

}

1. В главной функции запрашивается размерность списка и вызывается List\* list = add(size);

для ввода элементов списка.

Далее вызывается функция

print(list);

и выводится изначальный список.

Затем вызывается функция

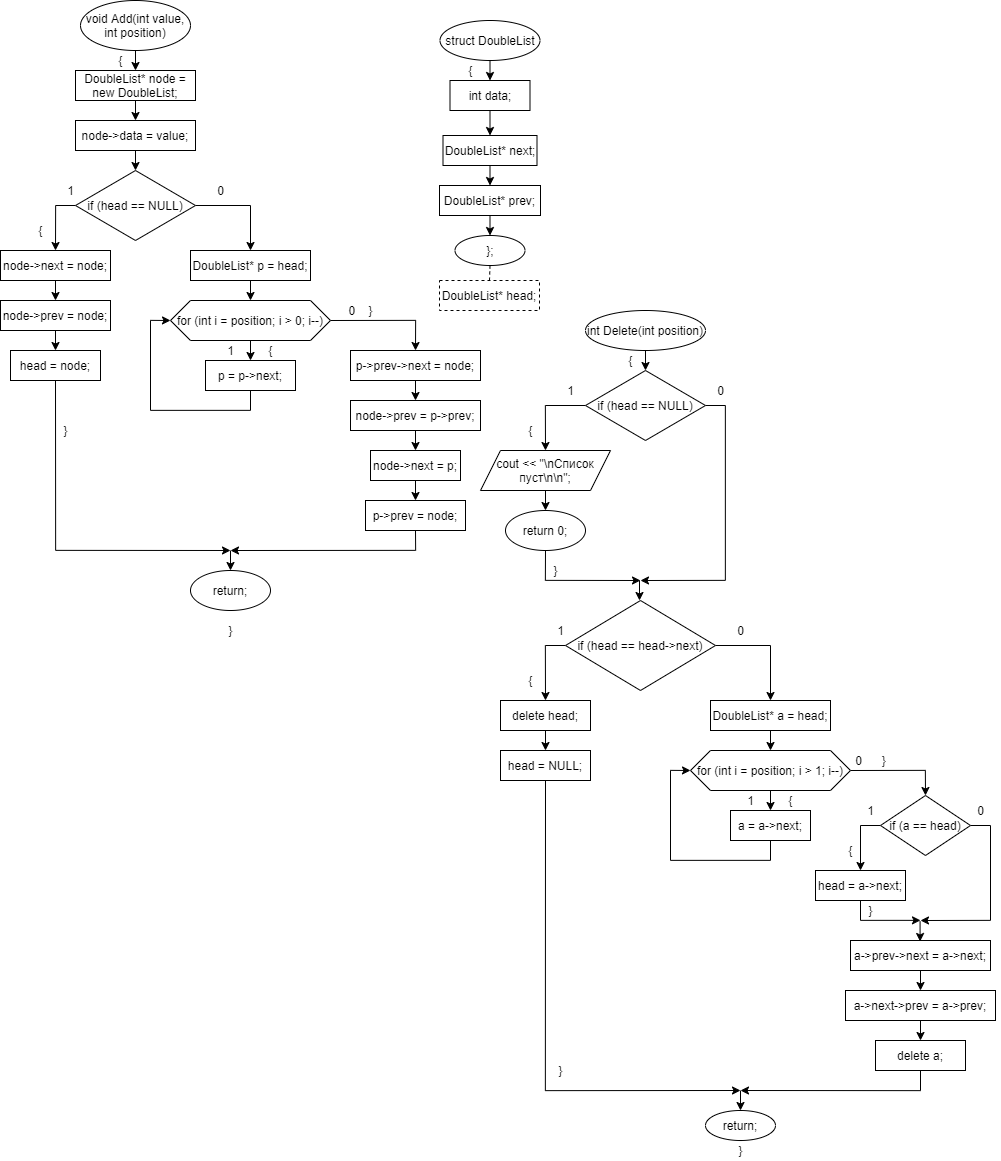
add(list, size);

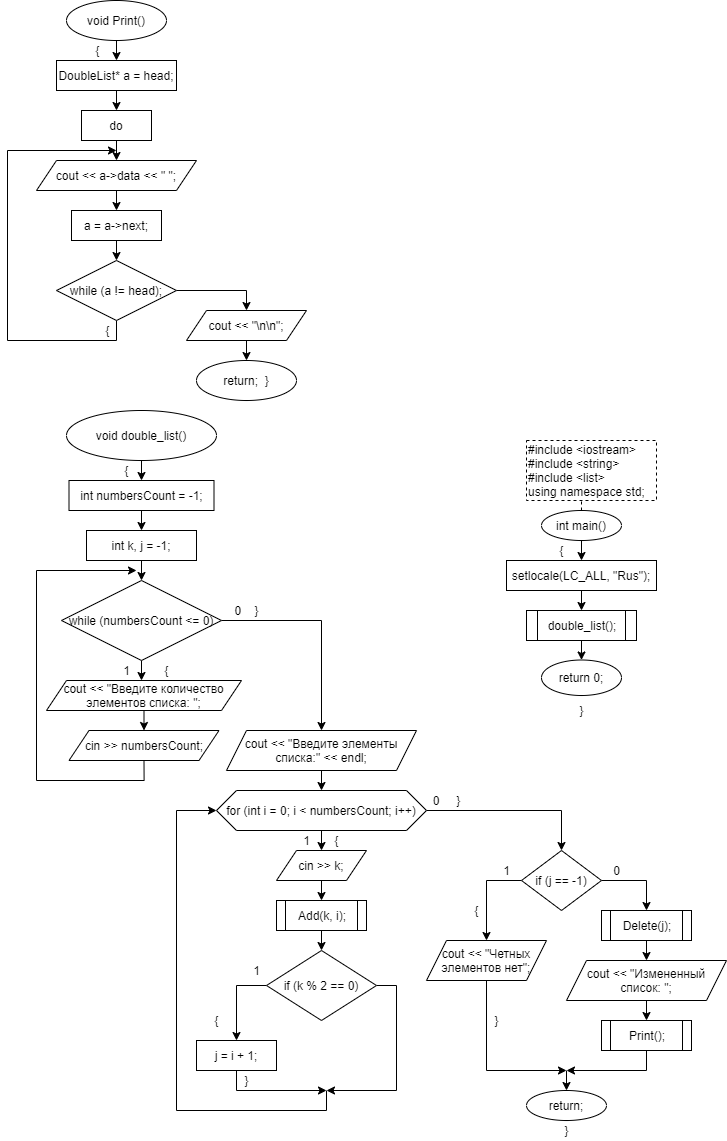
для добавления новых элементов в список под нечетными номерами.

Выводится измененный список вызовом функции

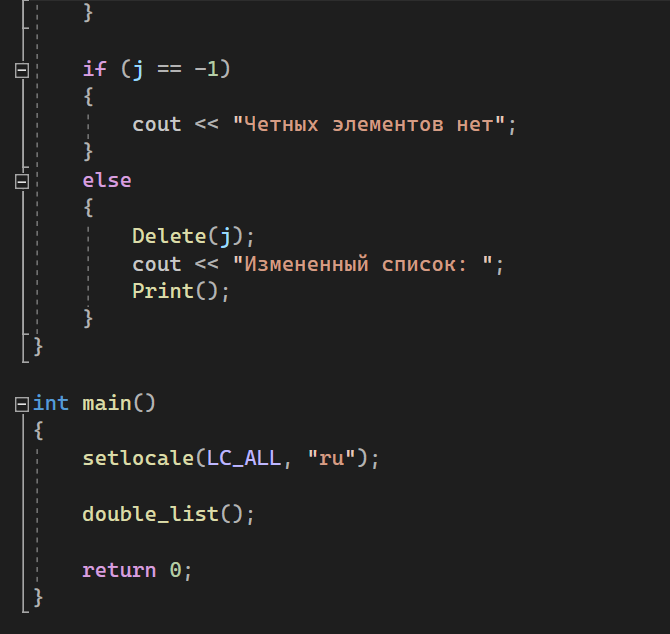
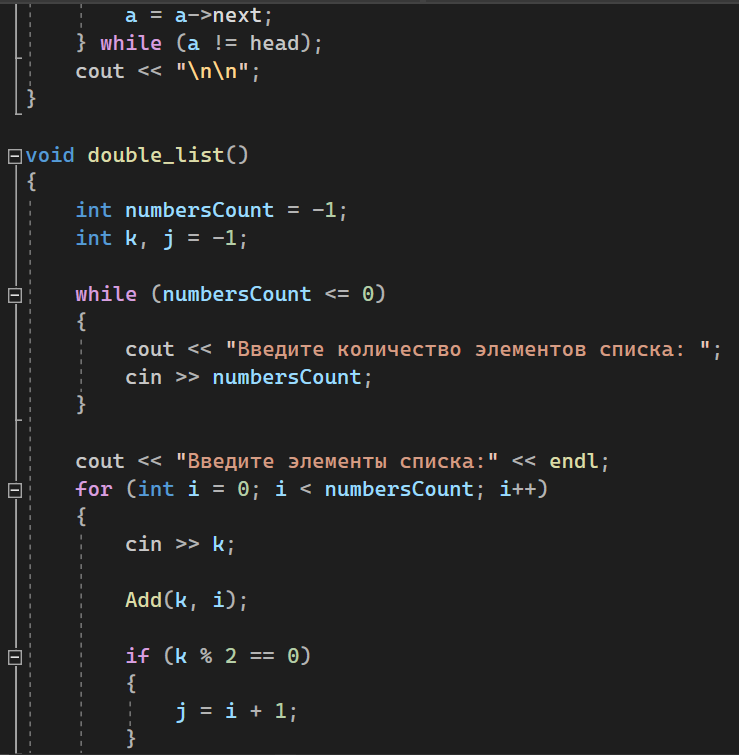
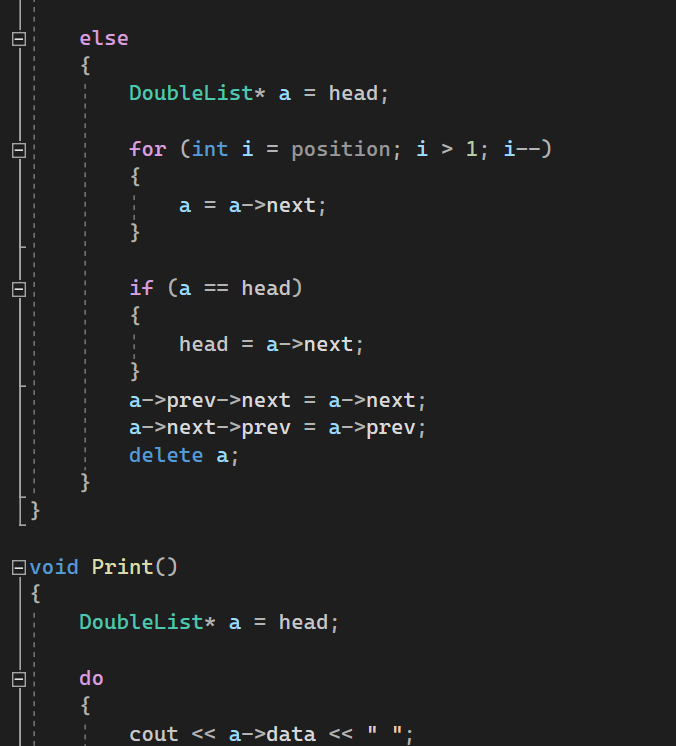
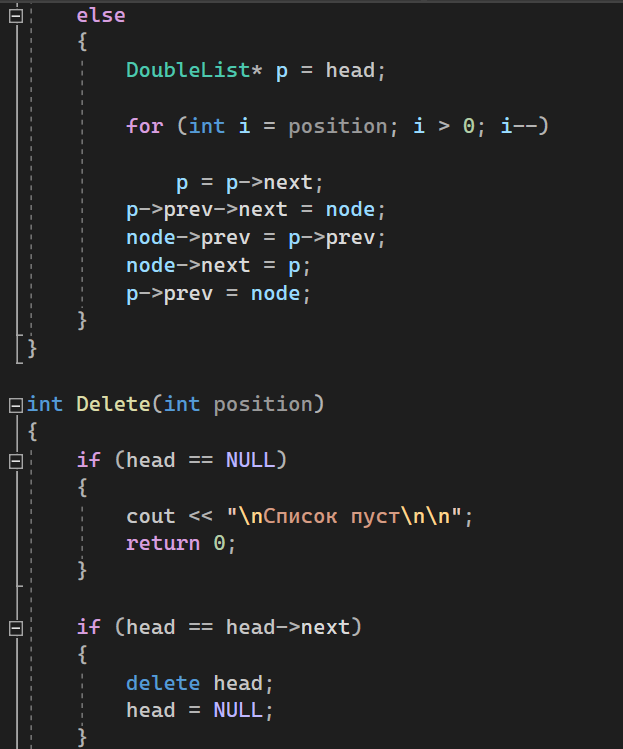
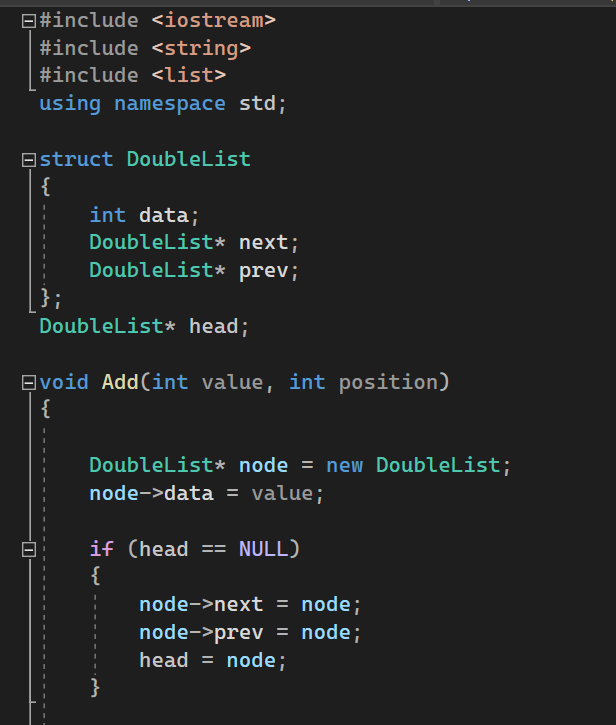
print(list);

**Блок-схема**

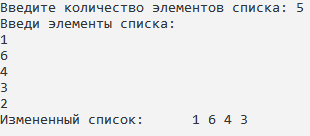


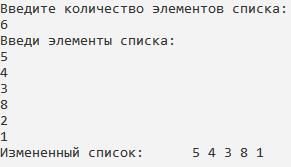


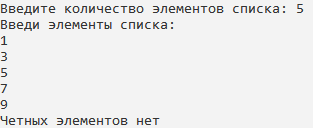
**Исходный код**



**Скриншоты программы**

****

****

****

**Анализ (объяснение скриншотов)**

Программа запрашивает у пользователя количество элементов списка и все элементы списка. Далее программа выводит измененный список, в котором удаляет последний четный элемент.