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 11.4**

Дисциплина: Основы алгоритмизации и программирования

Выполнил:

Студентка группы ИВТ-22-2б

Мифтахов Марат Ринатович

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2023**

**Постановка задачи**

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

**Анализ задачи**

1. Создаем очередь:

queue<int> q;

1. В главной функции запрашиваем у пользователя количество элементов очереди и сами элементы.
2. Далее, чтобы заполнить очередь используем метод *q.push(el);*

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

{

cin >> el;

q.push(el);

}

1. Создаем функцию для вывода очереди:

void Print(queue<int> q)

{

int s = q.size();

cout << "Очередь: " << endl;

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

{

cout << q.front() << " ";

q.pop();

}

cout << endl;

}

1. Создаем функцию для удаления последнего четного элемента очереди:

void Delete(queue<int> q)

{

queue<int> q2;

int last = -1;

int s = q.size();

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

{

if (q.back() % 2 == 0)

{

last = i;

}

if (last >= 0)

{

for (int j = 0; i < s - last; i++)

{

q2.push(q.back());

}

for (int k = s - last + 1; k < s; k++)

{

q2.push(q.back());

}

}

else

{

cout << "В очереди нет четных элементов." << endl;

break;

}

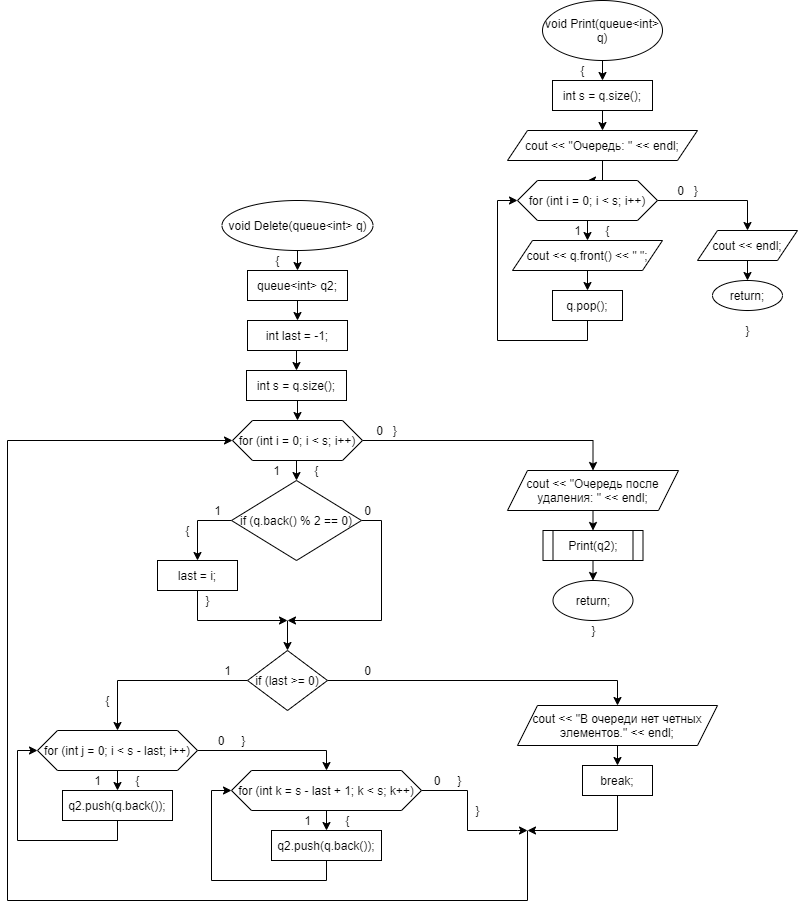
}

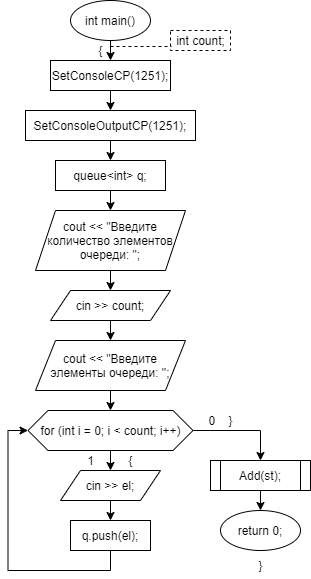
cout << "Очередь после удаления: " << endl;

Print(q2);

}

**Блок-схема**



**Исходный код**

#include<iostream>

#include<Windows.h>

#include<queue>

using namespace std;

int el;

void Print(queue<int> q)

{

int s = q.size();

cout << "Очередь: " << endl;

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

{

cout << q.front() << " ";

q.pop();

}

cout << endl;

}

void Delete(queue<int> q)

{

queue<int> q2;

int last = -1;

int s = q.size();

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

{

if (q.back() % 2 == 0)

{

last = i;

}

if (last >= 0)

{

for (int j = 0; i < s - last; i++)

{

q2.push(q.back());

}

for (int k = s - last + 1; k < s; k++)

{

q2.push(q.back());

}

}

else

{

cout << "В очереди нет четных элементов." << endl;

break;

}

}

cout << "Очередь после удаления: " << endl;

Print(q2);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

queue<int> q;

int count;

cout << "Введите количество элементов очереди: ";

cin >> count;

cout << "Введите элементы очереди: ";

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

{

cin >> el;

q.push(el);

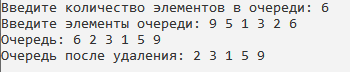
}

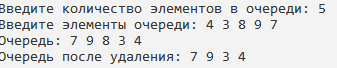
Delete(q);

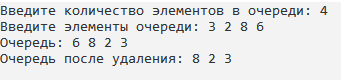
return 0;

}

**Скриншоты программы**

****

****

****

**Анализ (объяснение скриншотов)**

Программа запрашивает у пользователя количество элементов очереди и все элементы очереди, затем выводит очередь. Далее программа удаляет последний элемент с четным информационным полем и выводит получившуюся очередь.