Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №11.4**

Дисциплина: основы алгоритмизации и программирования

Тема: Очередь, как динамическая структура данных в С++

Вариант 24

Выполнил работу

студент группы РИС-20-1б

Коваленко Н. А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

Цель – написать программу для создания, редактирования и вывода очереди, как динамической структуры данных

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать логику очереди
* Продумать, как будет создана очередь в C++
* Создать очередь в C++
* Написать алгоритм, который будет редактировать исходную очередь

Постановка задачи

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

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание динамического массива, который будет являться очередью

char\* a1 = new char[numbersCount];

* Реализовать верный ввод данных в массив, чтобы он подчинялся логике очереди

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

{

cin >> a1[i];

if (a1[i] % 2 == 0) k = i;

}

* Создать новый динамический массив, в котором будет реализован новая очередь
* Написать алгоритм, который будет редактировать очередь

1. Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами и контроля размера очереди

int numbersCount = -1;

int k, j=-1;

Указатели на области памяти, которые хранят новые элементы очереди

char\* a2 = new char[numbersCount];

1. Ввод данных осуществляется посредством функции

cin >> numbersCount;

Вывод данных осуществляется посредством функции

cout << "Измененный стек: \t";

1. Для решения задачи будут использованы циклы. Например, цикл, который создает некоторое количество новых элементов в очереди и заносит в них данные:

for (int i = k + 1; i < numbersCount; i++)

a2[i - 1] = a1[i];

Для защиты программы от случайных ошибок были реализованы проверки входных данных:

Проверка размерности массива (размер больше 0):

int numbersCount = -1;

while (numbersCount <= 0)

{

cout << "Введи количество элементов в списке: ";

cin >> numbersCount;

}

Код

#include <iostream>

#include <string>

#include <list>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int numbersCount = -1, j = 0,k;

while (numbersCount <= 0)

{

cout << "Введи количество элементов в очереди: ";

cin >> numbersCount;

}

char\* a1 = new char[numbersCount];

cout << "Вводи элементы очереди: ";

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

{

cin >> a1[i];

if (a1[i] % 2 == 0) k = i;

}

cout << "Очередь: ";

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

cout << i << ")" << a1[i] << " ";

char\* a2 = new char[numbersCount-1];

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

a2[i] = a1[i];

for (int i = k + 1; i < numbersCount; i++)

a2[i - 1] = a1[i];

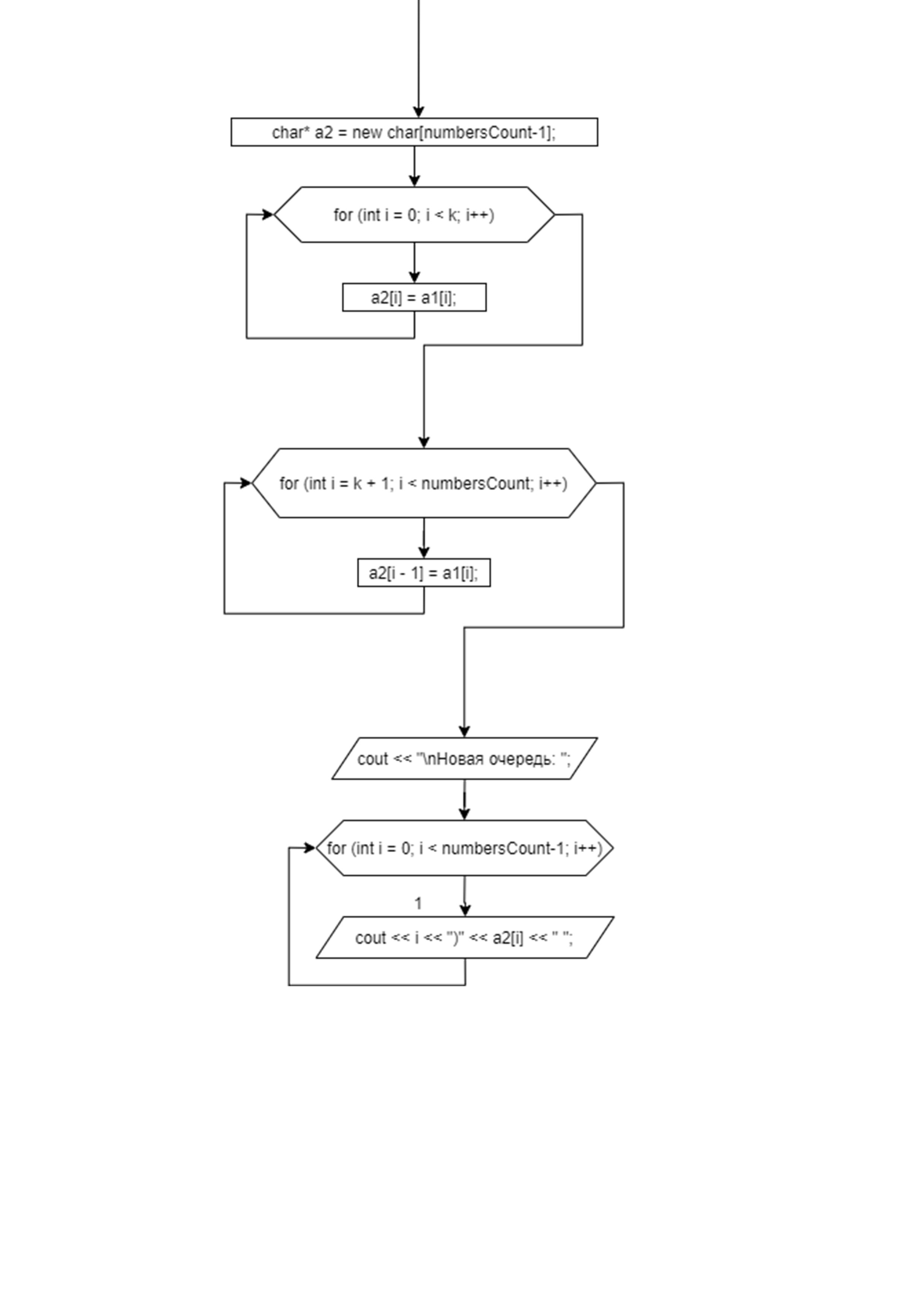
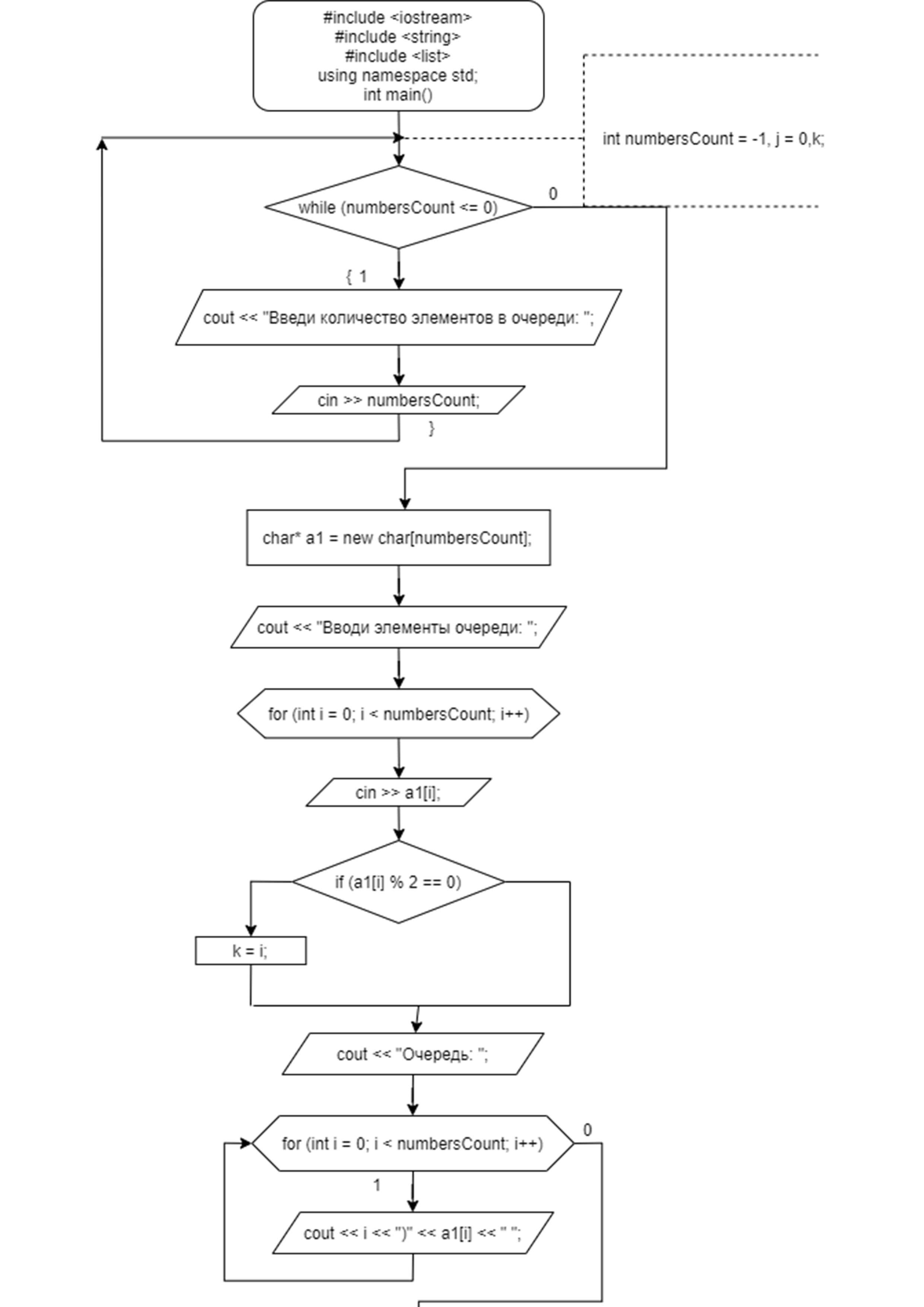
cout << "\nНовая очередь: ";

for (int i = 0; i < numbersCount-1; i++)

cout << i << ")" << a2[i] << " ";

}

Блок-схема:



Работа кода

