Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

ОТЧЁТ

«Лабораторная работа № 11. Информационные динамические структуры.

Очереди»

Выполнил

Студент группы РИС-22-2Б

Хохряков А.С.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь 2023

# Содержание отчета

1. Постановка задачи.
2. Анализ задачи.
3. Блок-схема
4. Текст программы

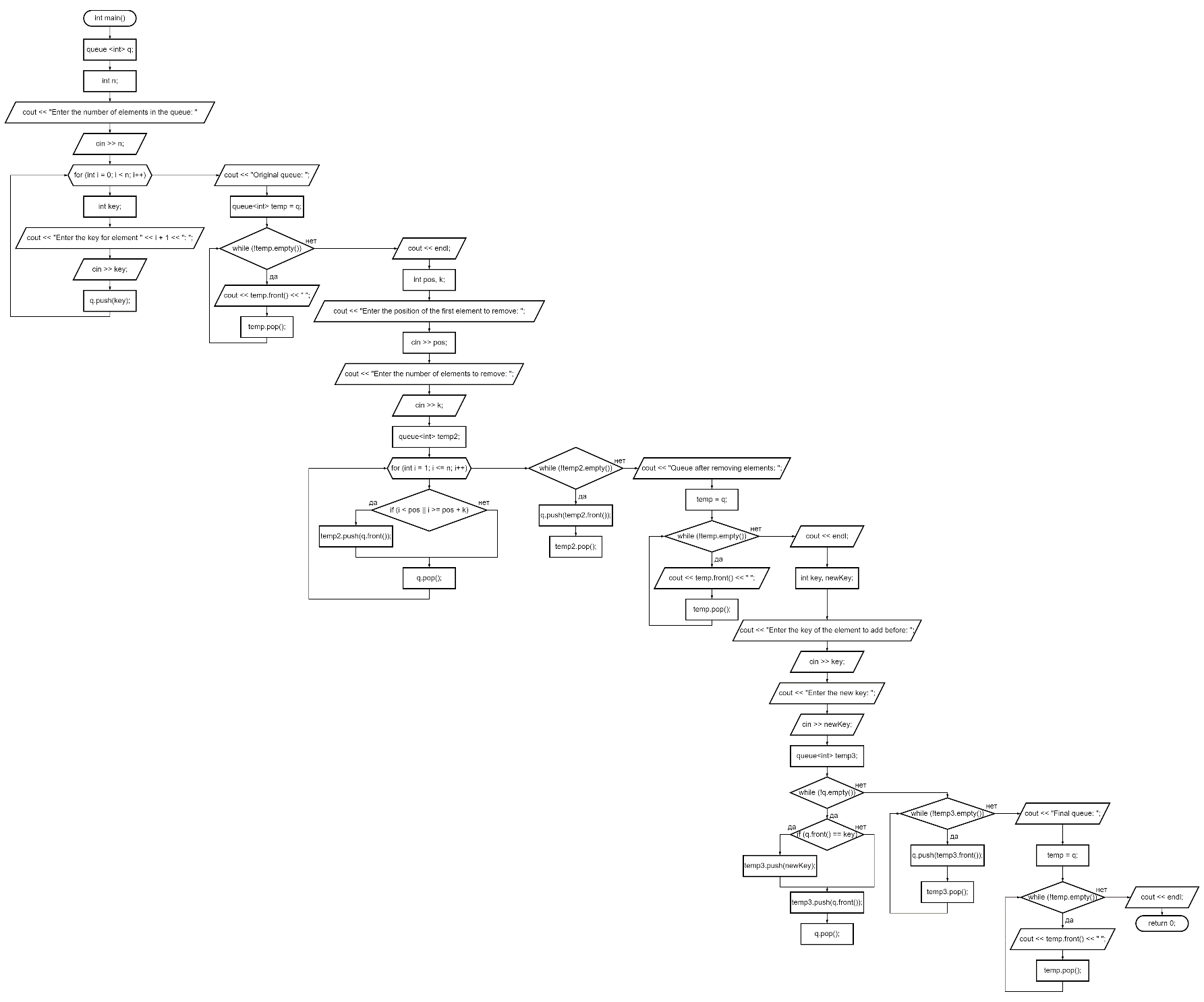
Постановка задачи

Записи в линейном списке содержат ключевое поле типа int. Сформировать однонаправленный список. Удалить из него К элементов, начиная с заданного номера, добавить элемент перед элементом с заданным ключом;

Анализ задачи

Задача решается с помощью использования очереди - структуры данных, которая обладает свойством "First In First Out" (FIFO), что означает, что первый элемент, который был добавлен в очередь, будет первым элементом, который будет удален из нее. Первоначально мы заполняем очередь данными, вводимыми пользователем. Затем выводим исходную очередь и приступаем к удалению элементов из заданного диапазона. Для этого создаем временную очередь, в которую мы будем помещать все элементы из исходной очереди, кроме тех, которые нужно удалить. После того, как мы прошлись по всей исходной очереди и удалили ненужные элементы, мы переносим все элементы из временной очереди обратно в исходную очередь. Затем мы добавляем новый элемент перед элементом с заданным ключом, для этого проходимся по всей исходной очереди, добавляя элементы во временную очередь. Как только мы находим элемент с нужным ключом, добавляем новый элемент во временную очередь перед этим элементом. После этого мы переносим все элементы из временной очереди обратно в исходную очередь. В конце мы выводим окончательный результат - итоговую очередь.

Блок-схема



Текст программы

#include <iostream>

#include <queue>

using namespace std;

int main() {

queue<int> q;

int n;

cout << "Enter the number of elements in the queue: ";

cin >> n;

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

int key;

cout << "Enter the key for element " << i + 1 << ": ";

cin >> key;

q.push(key);

}

cout << "Original queue: ";

queue<int> temp = q;

while (!temp.empty()) {

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

temp.pop();

}

cout << endl;

int pos, k;

cout << "Enter the position of the first element to remove: ";

cin >> pos;

cout << "Enter the number of elements to remove: ";

cin >> k;

queue<int> temp2;

for (int i = 1; i <= n; i++) {

if (i < pos || i >= pos + k) {

temp2.push(q.front());

}

q.pop();

}

while (!temp2.empty()) {

q.push(temp2.front());

temp2.pop();

}

cout << "Queue after removing elements: ";

temp = q;

while (!temp.empty()) {

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

temp.pop();

}

cout << endl;

int key, newKey;

cout << "Enter the key of the element to add before: ";

cin >> key;

cout << "Enter the new key: ";

cin >> newKey;

queue<int> temp3;

while (!q.empty()) {

if (q.front() == key) {

temp3.push(newKey);

}

temp3.push(q.front());

q.pop();

}

while (!temp3.empty()) {

q.push(temp3.front());

temp3.pop();

}

cout << "Final queue: ";

temp = q;

while (!temp.empty()) {

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

temp.pop();

}

cout << endl;

return 0;

}