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.

Стеки»

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

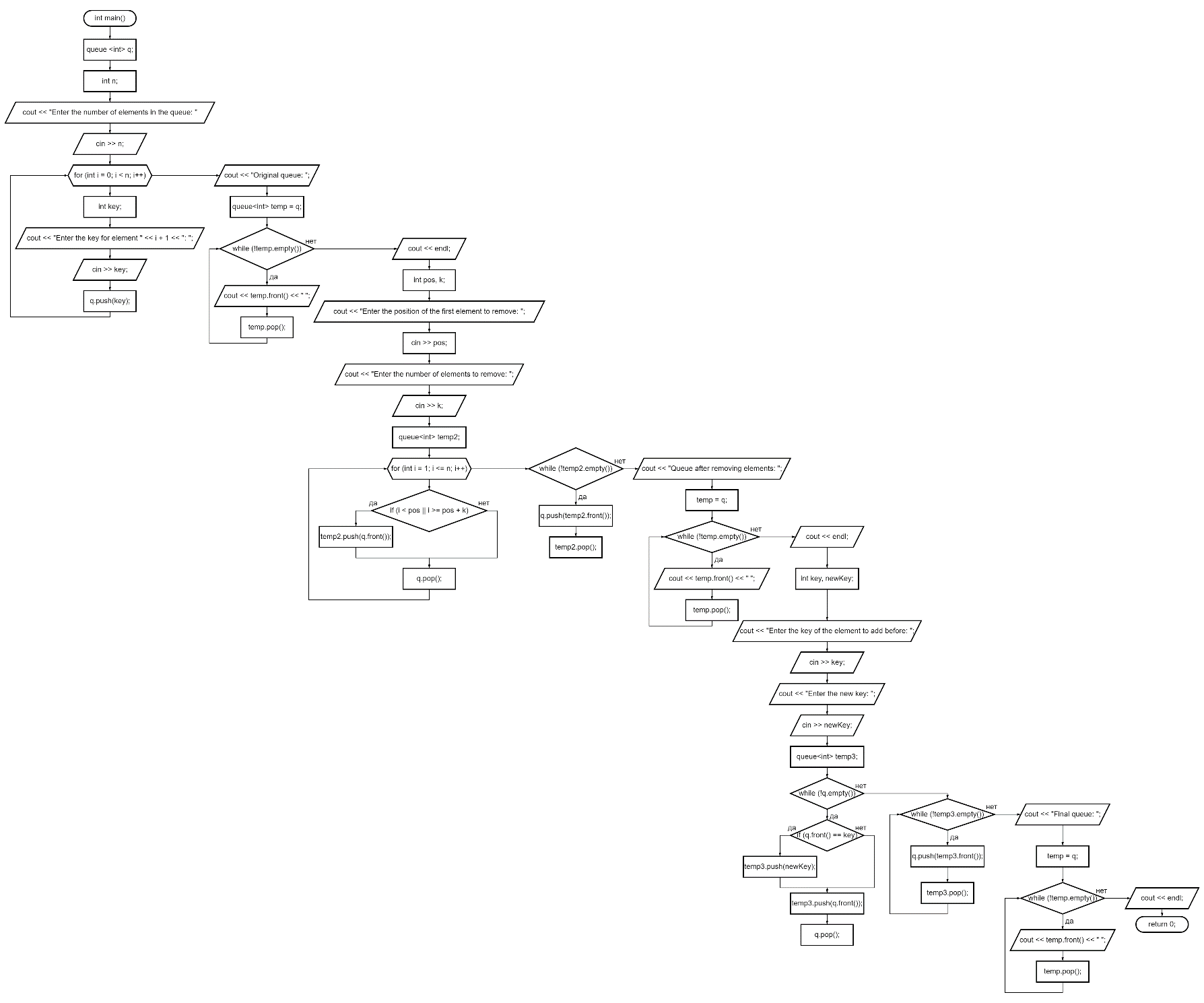
Постановка задачи

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 стека - структуры данных, которая представляет собой коллекцию элементов, где доступ к элементам осуществляется только в определенном порядке. Стек работает по принципу "последний вошел - первый вышел". Сначала мы создаем и заполняем исходный стек данными, вводимыми пользователем. Затем мы выводим исходный стек и приступаем к удалению элементов из заданного диапазона. Для этого мы создаем временный стек, в который переносим все элементы из исходного стека, кроме элементов, которые нужно удалить. Затем мы снова переносим элементы из временного стека в исходный стек, получая таким образом стек без удаленных элементов. Затем мы добавляем новый элемент перед элементом с заданным ключом. Для этого мы создаем временный стек и переносим элементы из исходного стека во временный стек, пока не найдем элемент с нужным ключом. Затем мы добавляем новый элемент во временный стек и переносим обратно элементы из временного стека в исходный стек, получая таким образом стек с добавленным элементом. В конце мы выводим окончательный результат - итоговый стек.

Блок-схема



Текст программы

#include <iostream>

#include <stack>

using namespace std;

int main() {

stack<int> st;

int n;

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

cin >> n;

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

int key;

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

cin >> key;

st.push(key);

}

cout << "Original stack: ";

stack<int> temp = st;

while (!temp.empty()) {

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

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;

stack<int> temp2;

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

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

temp2.push(st.top());

}

st.pop();

}

while (!temp2.empty()) {

st.push(temp2.top());

temp2.pop();

}

cout << "Stack after removing elements: ";

temp = st;

while (!temp.empty()) {

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

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;

stack<int> temp3;

while (!st.empty()) {

if (st.top() == key) {

temp3.push(newKey);

}

temp3.push(st.top());

st.pop();

}

while (!temp3.empty()) {

st.push(temp3.top());

temp3.pop();

}

cout << "Final stack: ";

temp = st;

while (!temp.empty()) {

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

temp.pop();

}

cout << endl;

return 0;

}