Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы КЗИ-21-1б  
Полковников Т.Ю.

Проверил  
доцент кафедры ИТАС  
ст. преп. Яруллин Д.В.

Пермь, 2022

## Постановка задачи:

1. Сформировать однонаправленный и двунаправленный списки или стек и очередь. 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 решения задачи:

Тип информационного поля char\*. Добавить в список элемент после элемента с заданным информационным полем.

## Описание переменных:

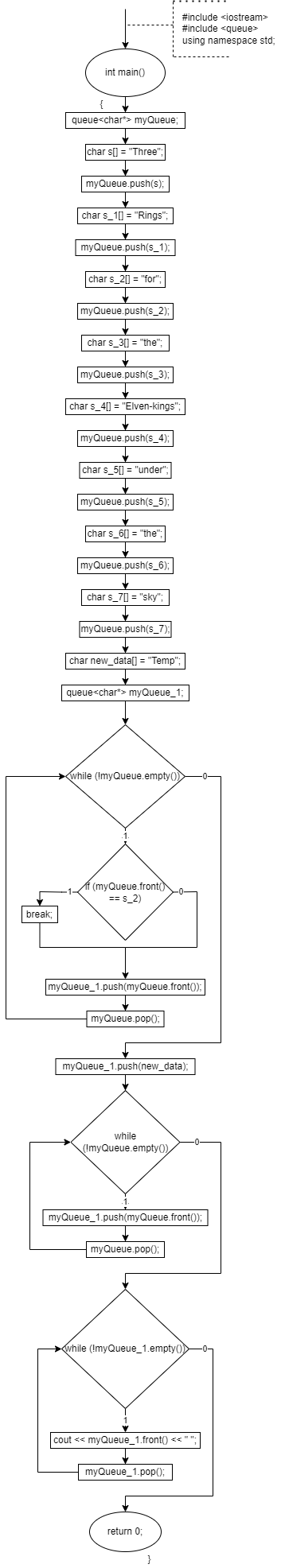
myQueue – начальная очередь

s, s\_1, s\_2, … s\_7 – строки для заполнения очереди

new\_data – информационное поле нового элемента очереди

myQueue\_1 – конечная очередь

**Блок схема**



## Код программы

#include <iostream>

#include <queue>

using namespace std;

int main() {

// создаём очередь

queue<char\*> myQueue;

// заполняем очередь

char s[] = "Three";

myQueue.push(s);

char s\_1[] = "Rings";

myQueue.push(s\_1);

char s\_2[] = "for";

myQueue.push(s\_2);

char s\_3[] = "the";

myQueue.push(s\_3);

char s\_4[] = "Elven-kings";

myQueue.push(s\_4);

char s\_5[] = "under";

myQueue.push(s\_5);

char s\_6[] = "the";

myQueue.push(s\_6);

char s\_7[] = "sky";

myQueue.push(s\_7);

char new\_data[] = "Temp";

queue<char\*> myQueue\_1;

while (!myQueue.empty()) {

if (myQueue.front() == s\_2) {

break;

}

myQueue\_1.push(myQueue.front());

myQueue.pop();

}

// вставляем новый элемент

myQueue\_1.push(new\_data);

while (!myQueue.empty()) {

myQueue\_1.push(myQueue.front());

myQueue.pop();

}

// вывод

while (!myQueue\_1.empty()) {

cout << myQueue\_1.front() << " ";

myQueue\_1.pop();

}

return 0;

}

Скриншоты результатов:



Анализ результатов:

Программа работает правильно. Выводиться очередь с новым элементом.