

ГУАП  
КАФЕДРА №43

ОТЧЁТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

К.Т.Н., доц. \_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

А.В. Туманова \_\_\_\_\_  
инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

**ОБРАБОТКА ТЕКСТОВЫХ ДАННЫХ**

по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.   Z7431  

  03.12.2018    
подпись, дата

  М.Д.Семочкин    
инициалы, фамилия

Санкт-Петербург  
2018

## 1. Цель работы

Целью работы является изучение представления строковых данных и алгоритмов их обработки.

## 2. Задание

Согласно варианту №24,

Разработать функцию, которая выполняет обработку символьной строки:

Функция дописывает одну строку в начало другой.

Примечание: строки представляются как массив символов. При реализации функции запрещается пользоваться функциями обработки строк библиотек языка C (без использования стандартной библиотеки «string.h»).

Примечание: если использовались операторы динамического выделения памяти, то следует вставить дополнительный код, обнаруживающий утечки памяти. См. раздел «Как обнаружить утечки памяти?». Все утечки памяти должны быть устранены.

## 3. Описание созданных функций

Для реализации задания нам потребуются следующие функции:

---

**Имя:** concatStrings

**Назначение:** Дописать одну строку в начало другой

**Входные данные:**

- \*string1 – указатель на первую строку
- \*string2 – указатель на вторую строку

**Выходные данные:** нет

**Побочный эффект:** первая строка перезаписывается новой строкой, которая представляет собой первую строку, в начало которой дописана вторая строка

**Тестовые данные:**

string1	string2	string1 (после выполнения функции)
антон	иванов	ивановантон

**Прототип:** void concatStrings(char \*string1, char \*string2)

## Алгоритм:

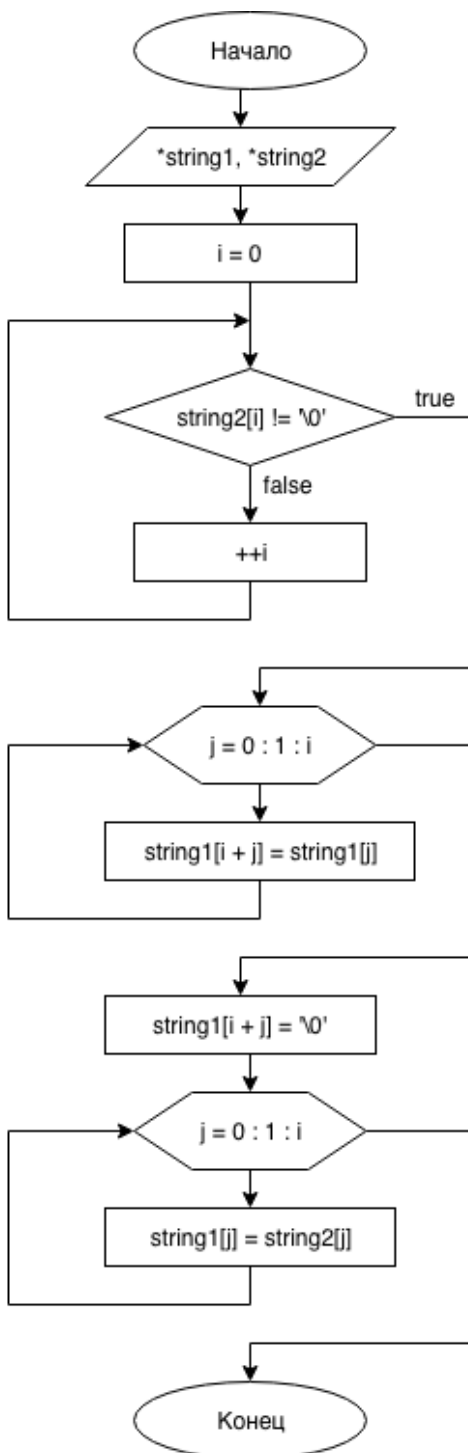
- псевдокод

подсчитать длину string2 и сохранить ее в i

записать string1 в string1 со смещением вправо на длину string2

записать string2 в начало string1

- блок-схема



#### 4. Листинг программы

```
#include <iostream>
using namespace std;

const int MAX_LENGTH = 1000;

void concatStrings(char *string1, char *string2) {
    // записывает string2 в начало string1

    int i, j;
    char tempString[MAX_LENGTH]; // вспомогательная строка
    // для сохранения string1

    // подсчитать длину string2 и сохранить ее в i
    for(i = 0; string2[i] != '\0'; ++i);

    // записать string1 в string1 со смещением вправо на длину string2
    for(j = 0; j < i; ++j) {
        string1[i + j] = string1[j];
    }
    string1[i + j] = '\0';

    // записать string2 в начало string1
    for(j = 0; j < i; ++j) {
        string1[j] = string2[j];
    }
}

int main() {

    char string1[MAX_LENGTH], string2[MAX_LENGTH];
    setlocale(LC_ALL, "russian");

    cout << "Введите первую строку: ";
    cin.getline(string1, MAX_LENGTH, '\n');
    // если ввели строку длиннее, чем выделено памяти, просто обрезаем
    // ее и чистим буфер
    if (cin.fail()) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    cout << "Введите вторую строку: ";
    cin.getline(string2, MAX_LENGTH, '\n');
    if (cin.fail()) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

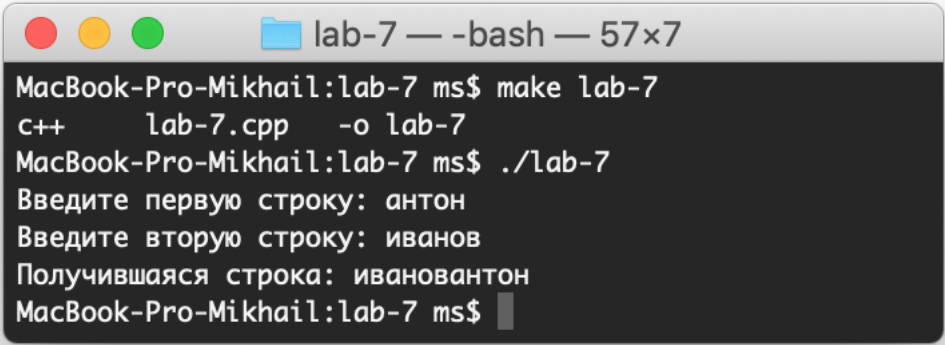
    concatStrings(string1, string2);

    cout << "Получившаяся строка: " << string1 << endl;

    return 0;
}
```

## 5. Пример выполнения программы

Ниже показан пример выполнения программы.



```
MacBook-Pro-Mikhail:lab-7 ms$ make lab-7
c++    lab-7.cpp    -o lab-7
MacBook-Pro-Mikhail:lab-7 ms$ ./lab-7
Введите первую строку: антон
Введите вторую строку: иванов
Получившаяся строка: ивановантон
MacBook-Pro-Mikhail:lab-7 ms$
```

Видно, что результаты расчётов совпадают с тестовыми данными.

## 6. Анализ результатов и выводы

К достоинствам программы можно отнести:

- Программа выполняет поставленную задачу и работает без ошибок (для корректных тестовых данных).
- Задание реализовано в виде отдельной функции.

Из недостатков можно отметить:

- Не производится проверка входных данных.
- Выделяется память для строк длиной 1000 символов, даже если на деле введенные строки окажутся короче