

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 №1

## ВЫЧИСЛЕНИЕ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ

по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.   Z7431  

  17.11.2018    
подпись, дата

  М.Д.Семочкин    
инициалы, фамилия

Санкт-Петербург  
2018

## 1. Цель работы

Целью работы является вычисление сложных математических выражений, а также отладка программы для поиска ошибок.

## 2. Задание

Согласно варианту №14,  
Написать программу для расчёта двух выражений:

$$z_1 = \frac{\cos \alpha + \sin \alpha}{\cos \alpha - \sin \alpha}$$
$$z_2 = \operatorname{tg} 2\alpha + \sec 2\alpha$$

Предварительно подготовить тестовые примеры по второй формуле с помощью калькулятора (результат вычисления по первой формуле должен совпадать со второй). Значение параметров тригонометрических функций должны задаваться пользователем в градусах.

## 3. Описание созданных функций

Для реализации задания нам потребуются следующие функции:

---

**Имя:** degToRad

**Назначение:** перевести введенное пользователем значение  $\alpha$  из градусов в радианы

**Входные данные:**

- angleInDegrees – угол в градусах

**Выходные данные:**

- вычисленное значение угла в радианах, соответствующего углу angleInDegrees в градусах

**Побочный эффект:** отсутствует.

**Тестовые данные:**

angleInDegrees	результат
35	0.61086
10	0.17453

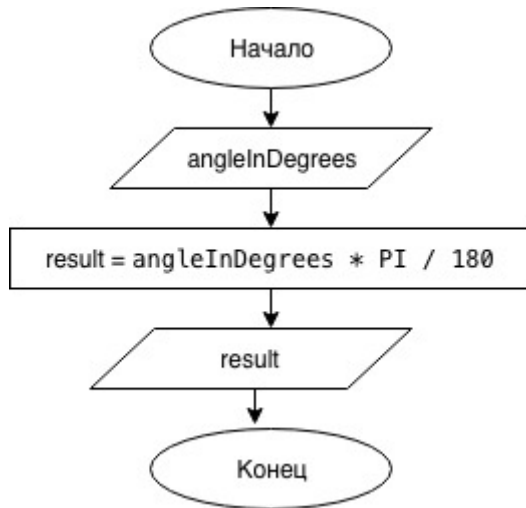
**Прототип:** double degToRad(const double angleInDegrees)

**Алгоритм:**

- псевдокод

вернуть  $\text{angleInDegrees} \times \pi / 180$

- блок-схема



**Имя:** sec

**Назначение:** вычислить значение секанса

**Входные данные:**

- angle – угол в радианах

**Выходные данные:**

- вычисленное значение секанса для данного угла

**Побочный эффект:** отсутствует.

**Тестовые данные:**

angle	результат
0.6	1.21162
0.17	1.01462

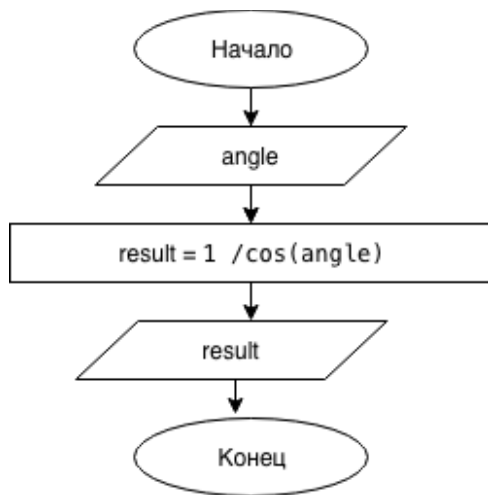
**Прототип:** double sec(const double val)

**Алгоритм:**

- псевдокод

вернуть  $1 / \cos(\text{angle})$

- блок-схема



---

**Имя:** calcZ1

**Назначение:** вычислить z1

**Входные данные:**

- a – угол в радианах

**Выходные данные:**

- вычисленное значение выражения

**Побочный эффект:** отсутствует.

**Тестовые данные:**

а	результат
35	5.67128
10	1.42815

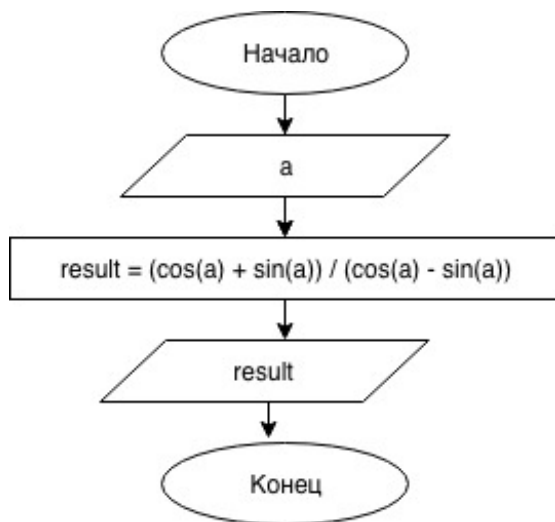
**Прототип:** double calcZ1(const double a)

**Алгоритм:**

- псевдокод

вернуть  $(\cos(a) + \sin(a)) / (\cos(a) - \sin(a))$

- блок-схема



---

**Имя:** calcZ2

**Назначение:** вычислить z2

**Входные данные:**

- a – угол в радианах

**Выходные данные:**

- вычисленное значение выражения

**Побочный эффект:** отсутствует.

**Тестовые данные:**

a	результат
35	5.67128
10	1.42815

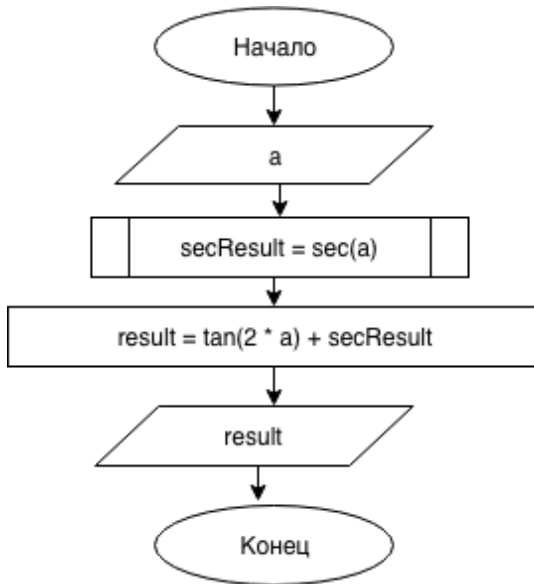
**Прототип:** double calcZ2(const double a)

**Алгоритм:**

- псевдокод

вернуть  $\tan(2 * a) + \sec(2 * a)$

- блок-схема



#### 4. Листинг программы

```
#include <iostream>
#include <math.h>
using namespace std;

const double PI =3.141592653589793238462;

double degToRad(const double angleInDegrees) {
    return angleInDegrees * PI / 180;
}

double sec(const double angle) {
    return 1 / cos(angle);
}

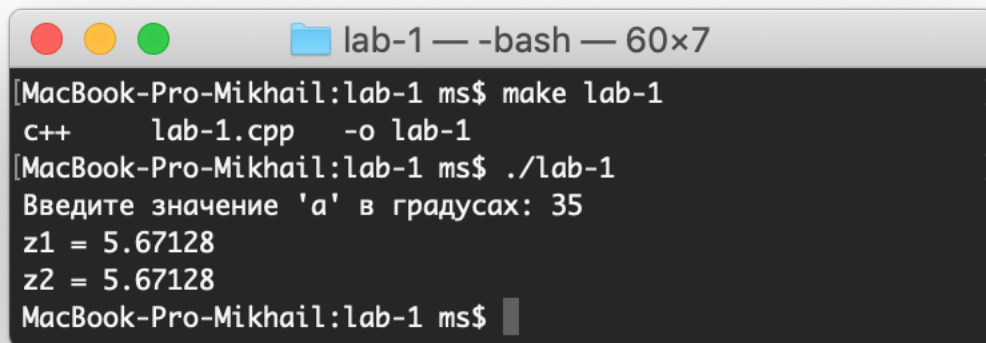
double calcZ1(const double a) {
    return (cos(a) + sin(a)) / (cos(a) - sin(a));
}

double calcZ2(const double a) {
    return tan(2 * a) + sec(2 * a);
}

int main() {
    double a, aInRadians;
    setlocale(LC_ALL, "russian");
    cout << "Введите значение 'а' в градусах: ";
    cin >> a;
    aInRadians = degToRad(a);
    cout << "z1 = " << calcZ1(aInRadians) << endl;
    cout << "z2 = " << calcZ2(aInRadians) << endl;
    return 0;
}
```

## 5. Пример выполнения программы

Ниже показан пример выполнения программы.



```
lab-1 — -bash — 60x7
[MacBook-Pro-Mikhail:lab-1 ms$ make lab-1
c++      lab-1.cpp  -o lab-1
[MacBook-Pro-Mikhail:lab-1 ms$ ./lab-1
Введите значение 'a' в градусах: 35
z1 = 5.67128
z2 = 5.67128
MacBook-Pro-Mikhail:lab-1 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).
- Каждое задание реализовано в виде отдельной функции, что позволяет эти функции в других проектах.

Из недостатков можно отметить:

- Не производится проверка входных данных.