

ГУАП

КАФЕДРА № 34

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Старший преподаватель  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

К.А. Жиданов  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 3145

\_\_\_\_\_  
подпись, дата

Ю.Д. Пьянов  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2022

## Вариант №5

### Цель работы

Изучить методы разработки консольных приложений, способы их запуска и обработки кодов возврата.

### Ход работы

1. Реализуем на языке Си функцию, вычисляющую корень квадратный из двух. Данная функция будет вычислять корень с помощью вавилонского метода по формуле:

Код функции:

```
double e_eps(double eps) {  
    int n = 0;  
    double sum = 1, temp = 1;  
    while (temp > eps) {  
        n++;  
        temp = temp * 1 / n;  
        sum += temp;  
    }  
    return sum;  
}
```

2. Составляем набор эталонных параметров и результатов для проверки работы программы. В качестве параметра будет выступать точность вычисления корня квадратного.

Параметр	Результат
0.1	2.7
0.0001	2.7182
0.000001	2.718281
0.00000001	2.71828182

3. Реализуем тестирующую функцию

Для корректной работы данной функции напомним вспомогательную функцию fabs, возвращающую модуль числа:

```
double nabs(double x) {  
    return x > 0 ? x : (-1 * x);  
}
```

4. Реализуем программу, возвращающую код ошибки, соответствующий результату работы тестирующей функции:

```
double nabs(double x) {
```

```

    return x > 0 ? x : (-1 * x);
}

double e_eps(double eps) {
    int n = 0;
    double sum = 1, temp = 1;

    while (temp > eps) {
        n++;
        temp = temp * 1 / n;
        sum += temp;
    }

    return sum;
}

double eps_test() {
    int r = 0;
    r = r || fabs(e_eps(0.1) - 2.7) >= 0.1 ;
    r = r || fabs(e_eps(0.0001) - 2.7182) >= 0.0001;
    r = r || fabs(e_eps(0.000001) - 2.718281) >= 0.000001;
    r = r || fabs(e_eps(0.00000001) - 2.71828182) >= 0.00000001;
    r = r || fabs(e_eps(0.0000000001) - 2.7182818284) >= 0.0000000001;
    r = r || fabs(e_eps(0.0000000000000001) - 2.718281828459045) >=
0.0000000000000001;
    return r;
}

int main() {
    return eps_test();
}

```

5. Компилируем программу, при имеющемся эталонном наборе данных программа работает корректно.  
 При добавлении неправильных данных в набор эталонных данных программа выдаёт код ошибки 1.