

ГУАП

КАФЕДРА № 34

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

Жиданов К.А.

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

по курсу:

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

3145

А. А. Москаленко

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

Вариант №1

Вычислить значение синуса с заданной точностью

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 sin_acc( double x ) {
    double tmp = x;
    double sum = tmp;
    int n = 0;

    while (abs_v(tmp) > E) {
        n++;
        tmp = tmp * (-1) * (x * x) / (2 * n) / (2 * n + 1);
        sum += tmp;
    }

    return sum;
}
```

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

3. Реализуем тестирующую функцию

```
int test_sin() {
    int r = 0;

    r = r || (abs_v(abs_v(sin_acc(0)) - 0 >= E));
    r = r || (abs_v(abs_v(sin_acc(PI/2)) - 1 >= E));
    r = r || (abs_v(abs_v(sin_acc(PI)) - 0 >= E));
    r = r || (abs_v(abs_v(sin_acc(3 * PI/2)) - 1 >= E));
    r = r || (abs_v(abs_v(sin_acc(2 * PI)) - 0 >= E));

    // wrong:
    r = r || (abs_v(abs_v(sin_acc(PI/3)) - 0.866) >= E);

    return r;
}
```

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

```
double abs_v( double x ) {
    if ( x > 0 ) return x;
    else return -1 * x;
}
```

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

```
#define E 0.001
#define PI 3.1415

double abs_v( double x ) {
    if ( x > 0 ) return x;
    else return -1 * x;
}

double sin_acc( double x ) {
    double tmp = x;
    double sum = tmp;
    int n = 0;

    while (abs_v(tmp) > E) {
        n++;
        tmp = tmp * (-1) * (x * x) / (2 * n) / (2 * n + 1);
        sum += tmp;
    }

    return sum;
}

int test_sin() {
    int r = 0;

    r = r || (abs_v(abs_v(sin_acc(0)) - 0 >= E));
    r = r || (abs_v(abs_v(sin_acc(PI/2)) - 1 >= E));
    r = r || (abs_v(abs_v(sin_acc(PI)) - 0 >= E));
    r = r || (abs_v(abs_v(sin_acc(3 * PI/2)) - 1 >= E));
    r = r || (abs_v(abs_v(sin_acc(2 * PI)) - 0 >= E));

    // wrong:
    r = r || (abs_v(abs_v(sin_acc(PI/3)) - 0.866) >= E);

    return r;
}

int main() {
    return test_sin();
}
```

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