Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»

**ОТЧЕТ**

**О выполнении лабораторной работы №2**

**«Вычисление значений числовых рядов и функций с заданной точностью»**

**Cтудент**: Гатченко А.С.

**Группа**: Б22-525

**Преподаватель**: Половнева Ю. А.

**1. Формулировка индивидуального задания**

Вычислить значение функции в точке при помощи разложения в ряд:  
где |x| ≤ 1.

**2. Описание использованных типов данных**

При выполнении данной лабораторной работы использовались встроенные типы данных int, double, предназначенные для работы с целыми числами и с числами с плавающей точкой двойной точности, а также указатели на целые числа.

**3. Описание использованного алгоритма**

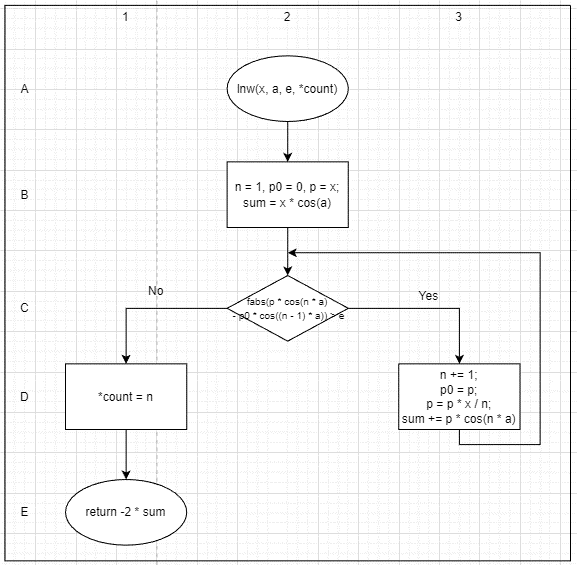
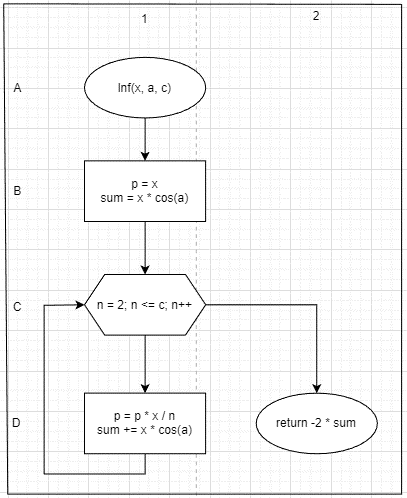
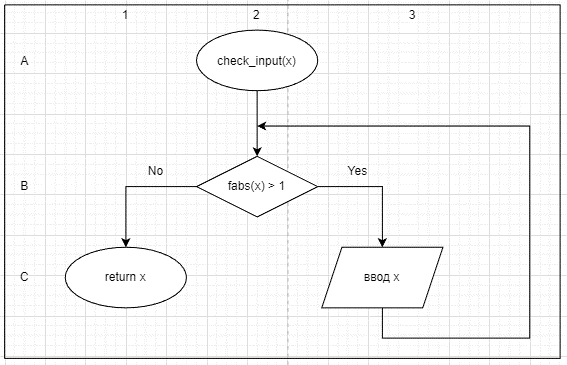
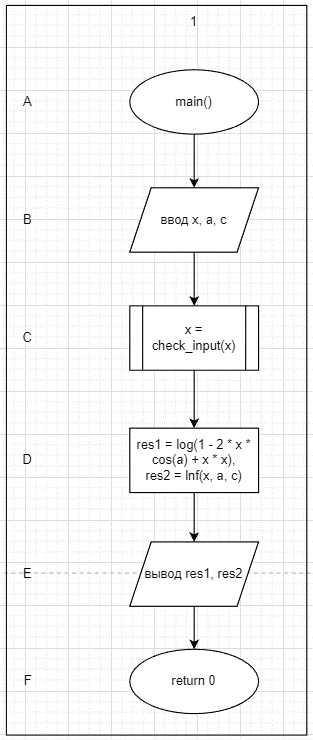


Рис. 1: Блок­схема алгоритма работы функции lnw()

Рис. 2: Блок-схема алгоритма работы функции lnf()





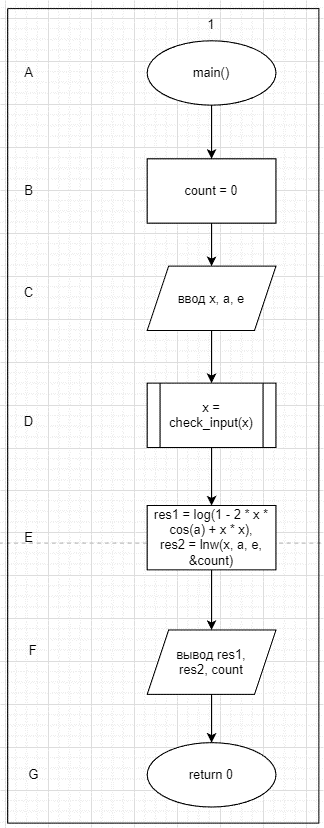


Рис. 3: Блок-схема алгоритма работы функции check\_input()

Рис. 4: Блок-схема алгоритма работы функции main()для цикла for

Рис. 5: Блок-схема алгоритма работы функции main()для цикла while

**4. Исходные коды разработанных программ**

Листинг 1: Исходные коды программы lab2f (файл: lab2f.c)

#include <stdio.h>

#include <math.h>

double lnf(double x, double a, int c){

double p = x;

double sum = x \* cos(a);

for (int n = 2; n <= c; n++){

p = p \* x / n;

sum += p \* cos(n \* a);

}

return -2 \* sum;

}

double check\_input(double x){

while (fabs(x) > 1){

printf("x value should belong to [-1; 1] range!\n");

printf("Input x: \n");

scanf("%lf", &x);

}

return x;

}

int main(){

double x = 0;

double a = 0;

int c = 0;

printf("Input x, a, number of cycles: \n");

scanf("%lf%lf%d", &x, &a, &c);

x = check\_input(x);

printf("%1.10lf -> %1.10lf", log(1 - 2 \* x \* cos(a) + x \* x), lnf(x, a, c));

return 0;

}

Листинг 2: Исходные коды программы lab2w (файл: lab2w.c)

#include <stdio.h>

#include <math.h>

double lnw(double x, double a, double e, int \*count){

int n = 1;

double p0 = 0;

double p = x;

double sum = x \* cos(a);

while (fabs(p \* cos(n \* a) - p0 \* cos((n - 1) \* a)) > e){

n += 1;

p0 = p;

p = p \* x / n;

sum += p \* cos(n \* a);

}

\*count = n;

return -2 \* sum;

}

double check\_input(double x){

while (fabs(x) > 1){

printf("x value should belong to [-1; 1] range!\n");

printf("Input x: \n");

scanf("%lf", &x);

}

return x;

}

int main(){

int count = 0;

double x = 0;

double a = 0;

double e = 0;

printf("Input x, a, delta: \n");

scanf("%lf%lf%lf", &x, &a, &e);

x = check\_input(x);

printf("%1.10lf -> %1.10lf\nSteps needed: %d", log(1 - 2 \* x \* cos(a) + x \* x), lnw(x, a, e, &count), count);

return 0;

}

**5. Описание тестовых примеров**

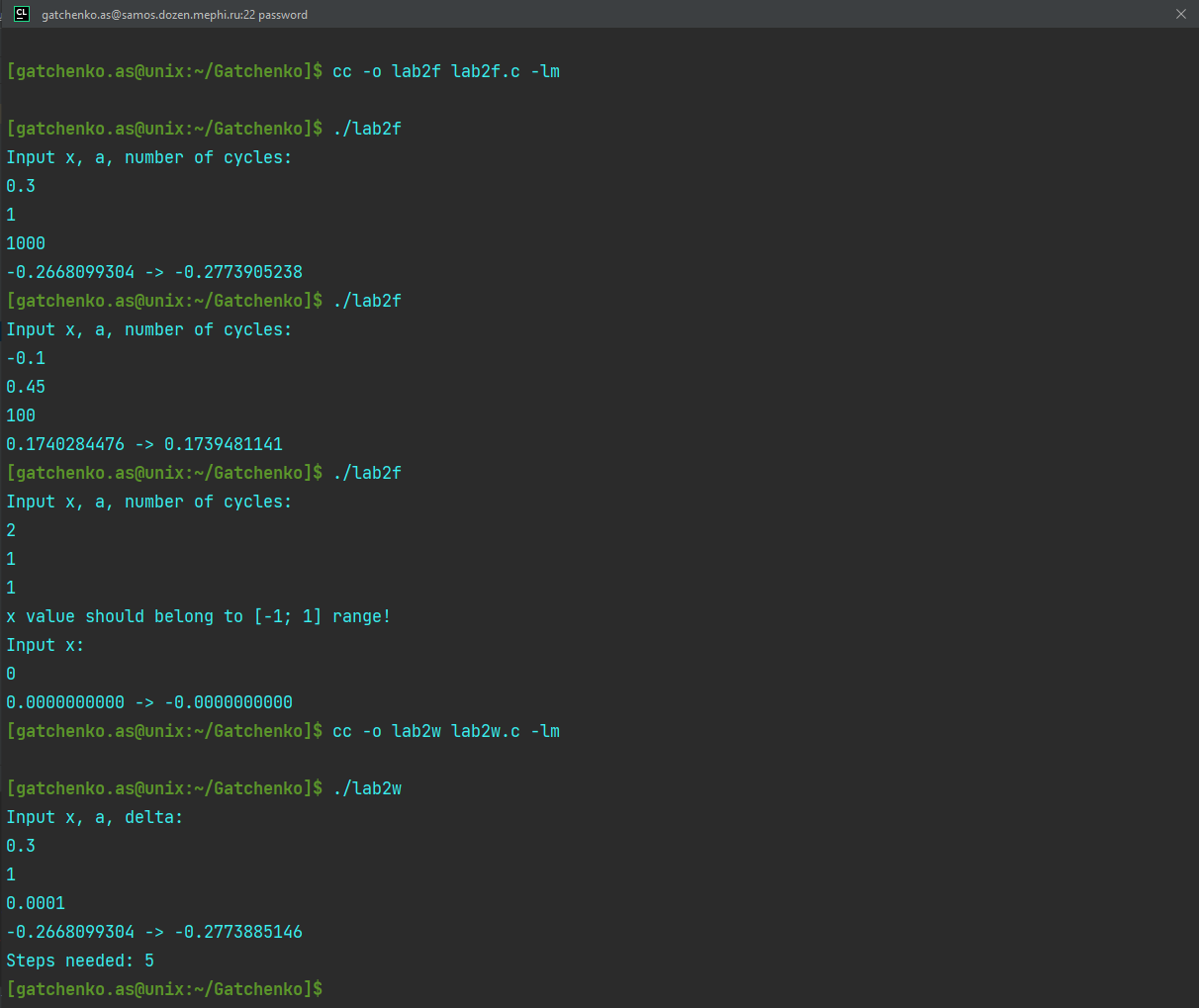
Таблица 1: Тестовые примеры lab2f.c

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Значение x | Значение a | Значение n | Ожидаемый вывод | Полученный вывод |
| 0.3 | 1 | 1000 | -0.2668099304 | -0.2773905238 |
| -0.1 | 0.45 | 100 | 0.1740284476 | 0.1739481141 |
| 2 | 1 | 1 | x value should belong to [-1; 1] range! | x value should belong to [-1; 1] range! |

Таблица 2: Тестовые примеры lab2w.c

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Значение x | Значение a | Значение e | Ожидаемый вывод | Полученный вывод | Количество итераций |
| 0.3 | 1 | 0.0001 | -0.2668… | -0.2773885146 | 5 |
| -0.1 | 0.45 | 0.0000000001 | 0.1740284476 | 0.1739481141 | 7 |
| 2 | 1 | 0.1 | x value should belong to [-1; 1] range! | x value should belong to [-1; 1] range! | 0 |

**6. Скриншоты**



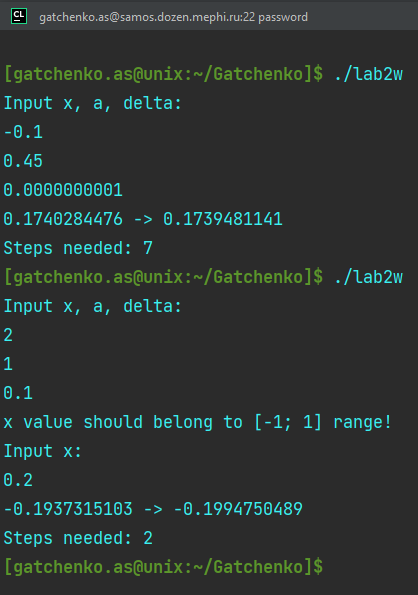
****

Рис. 5: Сборка и запуск программ lab2f и lab2w

**7. Выводы**

В ходе выполнения данной работы на примере программы, выполняющей вычисление значения функции в точке при помощи разложения в ряд, были рассмотрены базовые принципы работы построения программ на языке C и обработки чисел с плавающей запятой:

1. Организация ввода/вывода.
2. Разработка функций.
3. Объявление и использование переменных.
4. Выполнение простейших арифметических операций над целочисленными и дробными операндами.
5. Использование циклов и условий.
6. Использование указателей (параметров).