

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



# ОТЧЁТ

**О выполнении лабораторной работы №2**

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

**Студент:** Дзарданов Г. В.

**Группа:** Б22-525

**Преподаватель:** Половнева Ю. А.

Москва – 2022

# 1. Формулировка индивидуального задания

Вариант №32. Вычислить значение функции в точке при помощи разложения в ряд:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

где  $|x| < \infty$

## 2. Описание использованных типов данных

При выполнении программы использовались встроенные типы данных для хранения целых чисел (`int`) и встроенные типы данных для хранения вещественных чисел (`float`).

## 3. Описание использованного алгоритма

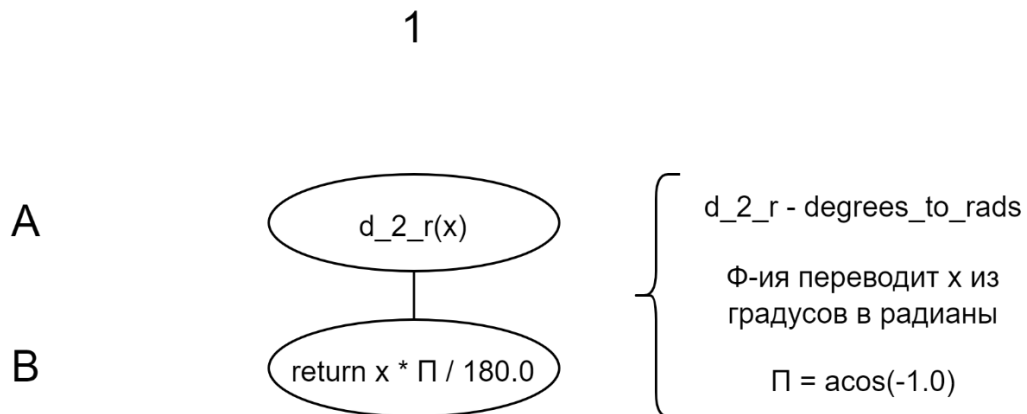


Рис. 1: Блок-схема алгоритма работы функции `degrees_to_rad()`, используемой в обеих программах и описанной в файлах `appendix.c` и `appendix.h`

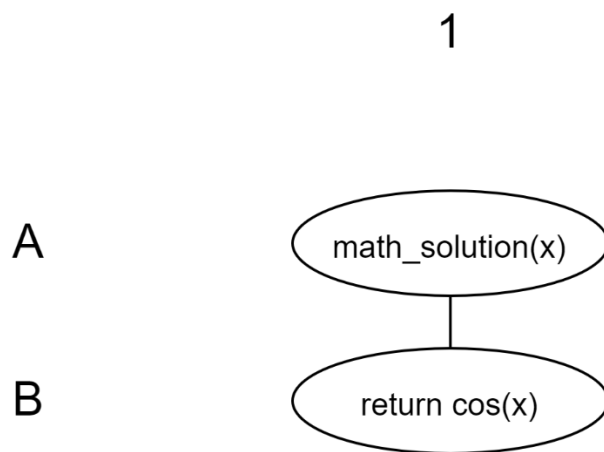


Рис. 2: Блок-схема алгоритма работы функции `math_solution()`, используемой в обеих программах и описанной в файлах `appendix.c` и `appendix.h`

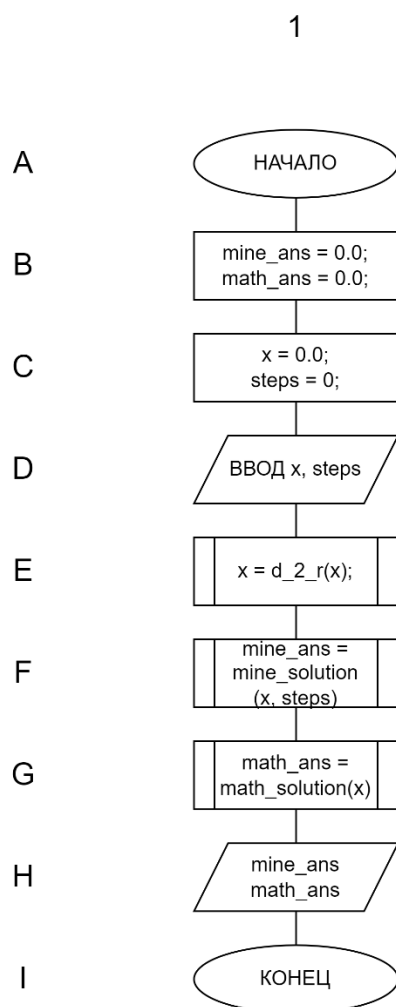


Рис. 3: Блок-схема алгоритма работы функции `main()`, используемой в файле `02-02_cosx_n.c`

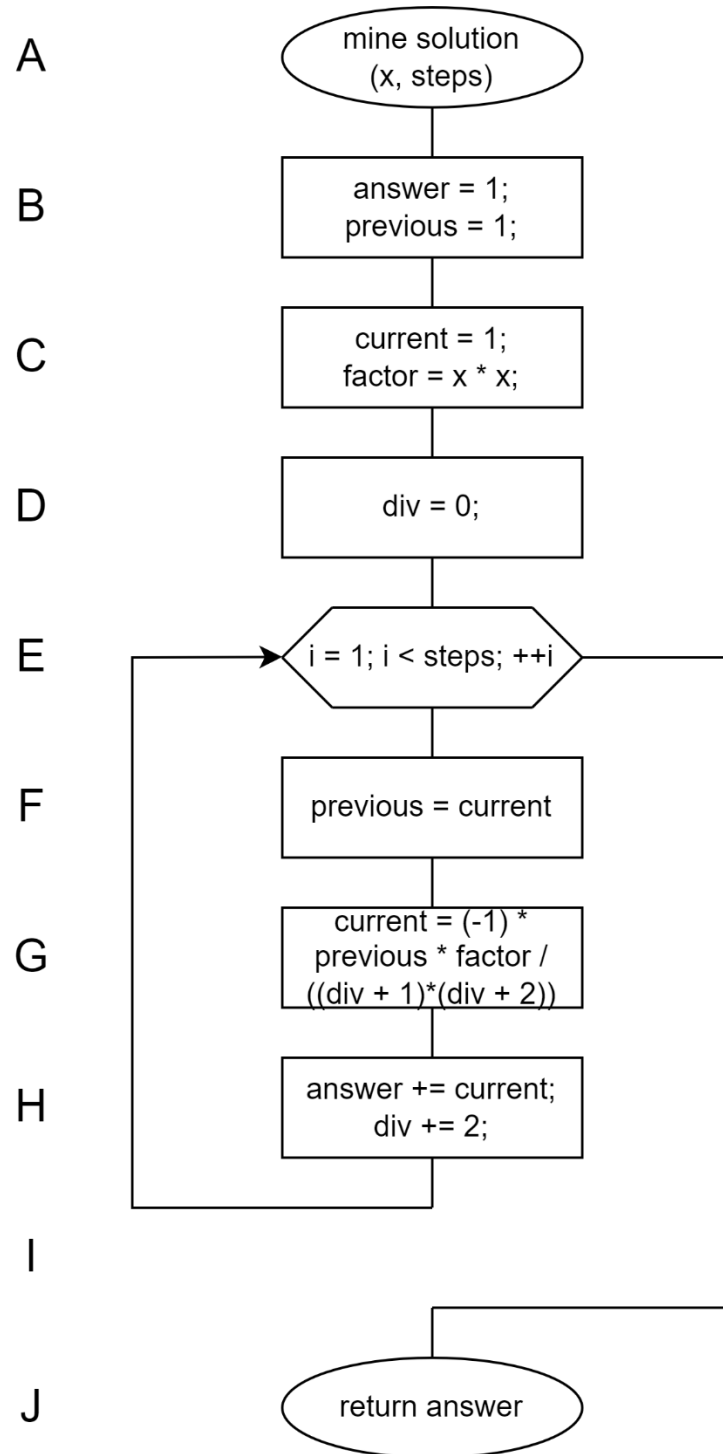


Рис. 4: Блок-схема алгоритма работы функции `mine_solution()`, используемой в файле `02-02_cosx_n.c`

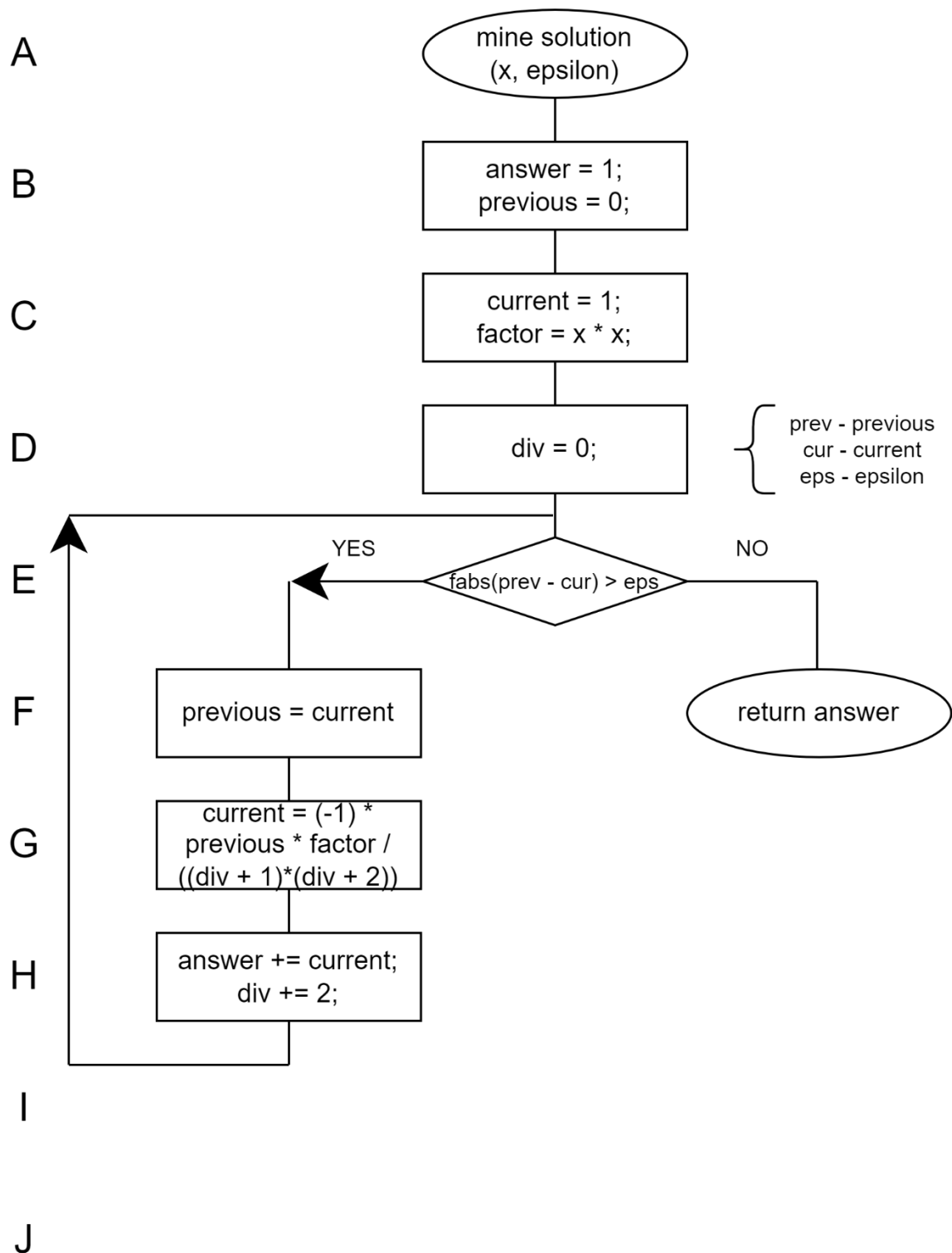


Рис. 5: Блок-схема алгоритма работы функции `mine_solution()`, используемой в файле `02-03_cosx_e.c`

1

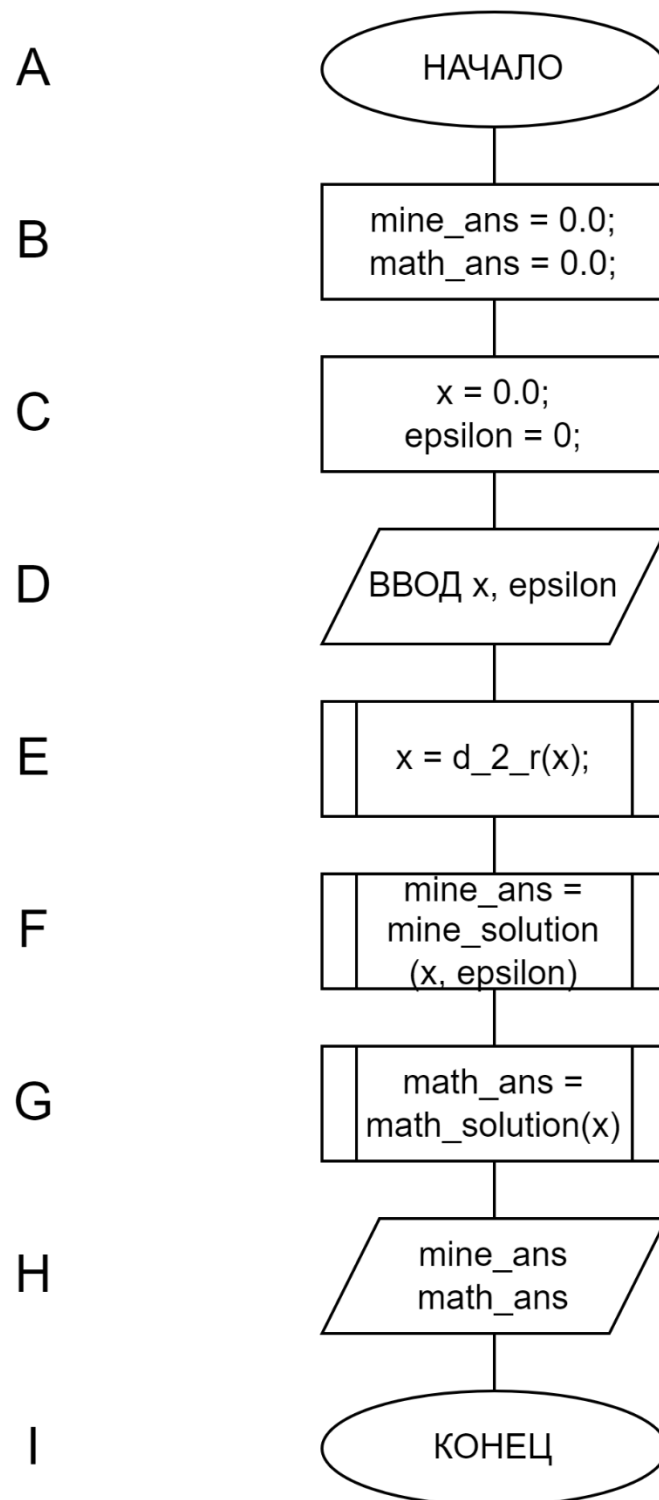


Рис. 6: Блок-схема алгоритма работы функции `main()`, используемой в файле `02-03_cosx_e.c`

## 4. Исходные коды разработанных программ

Листинг 1: Исходные коды программы функций из библиотеки `appendix.h`

```
#ifndef APPENDIX_H
#define APPENDIX_H

float math_solution(float x);
float degrees_to_rads(float x);

#endif
```

```
#include <stdio.h>
#include <math.h>

// RETURNS THE MATH.H VALUE OF RESULT
float math_solution(float x){
    return cos(x);
}

// TRANSFORMS DEGREES TO RADIANS
float degrees_to_rads(float x){
    return x * acos(-1.0) / 180;
}
```

Листинг 2: Исходные коды программы 02-02\_cosx\_n (02-02\_cosx\_n.c)

```
#include <stdio.h>
#include <math.h>
#include "appendix.h"

// CALCULATES THE VALUE
float mine_solution(float x, int steps){
    float answer = 1.0, previous = 1.0;
    float current = 1.0, factor = x * x;
    int div = 0;

    for (int i = 1; i < steps; ++i){
        previous = current;
        current = (-1) * previous * factor / ((div + 1) * (div + 2)) ;
        answer += current;
        div += 2;
    }
    return answer;
}

int main(){
    float mine_ans = 0.0, math_ans = 0.0;
    float x = 0.0;
    int steps = 0;

    printf("Enter x (in degrees): ");
    scanf("%f", &x);

    printf("Enter n: ");
    scanf("%d", &steps);

    x = degrees_to_rads(x);

    mine_ans = mine_solution(x, steps);
    math_ans = math_solution(x);

    printf("\nMY ANSWER : %f", mine_ans);
    printf("\nMATH ANSWER: %f", math_ans);
    return 0;
}
```

Листинг 3: Исходные коды программы 02-03\_cosx\_e (02-03\_cosx\_e.c)

```
#include <stdio.h>
#include <math.h>
#include "appendix.h"

// CALCULATES THE VALUE
float mine_solution(float x, float epsilon){
    float answer = 1.0, previous = 0.0;
    float current = 1.0, factor = x * x;
    int div = 0;

    while (fabs(previous - current) > epsilon){
        previous = current;
        current = (-1) * previous * factor / ((div + 1) * (div + 2));
        answer += current;
        div += 2;
    }
    return answer;
}

int main(){
    float mine_ans = 0.0, math_ans = 0.0;
    float x = 0.0, epsilon = 0.0;

    printf("Enter x (in degrees): ");
    scanf("%f", &x);

    printf("Enter e: ");
    scanf("%f", &epsilon);

    x = degrees_to_rads(x);

    mine_ans = mine_solution(x, epsilon);
    math_ans = math_solution(x);

    printf("\nMY ANSWER : %f", mine_ans);
    printf("\nMATH ANSWER: %f", math_ans);
    return 0;
}
```



## 5. Описание тестовых примеров

Таблица 1: Тестовый пример для программы 02-02\_cosx\_n.c

main()			
mine ans	math ans	x	steps
0,000000	0,000000	0,000000	0
		45,000000	3
		0.785398	
0,707429			
	0.707107		

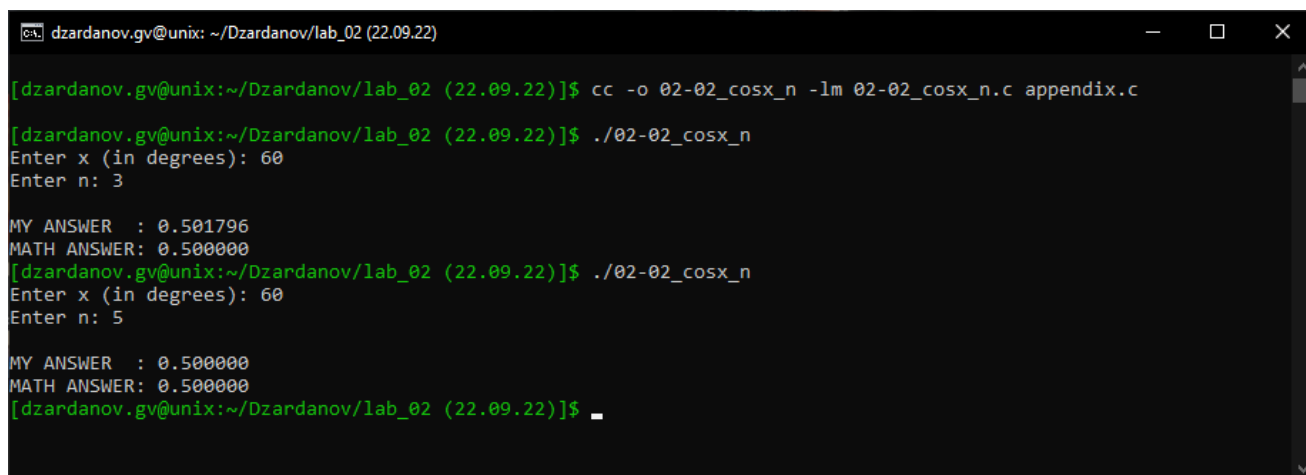
mine_solution()						
answer	previous	current	factor	x	steps	div
1,000000	1,000000	1,000000	0,616850	0.785398	3	0
	1,000000					
		-0,308425				
0,691575						
						2
	-0,308425					
		0,015854				
0,707429						
						4
0,707429						

Таблица 2: Тестовый пример для программы 02-03\_cosx\_e.c

main()			
mine ans	math ans	x	epsilon
0,000000	0,000000	0,000000	0,000000
		270,000000	0,010000
		4,712389	
-0,000011			
	0,000000		

mine_solution()						
answer	previous	current	factor	x	epsilon	div
1,000000	0,000000	1,000000	22,206610	#####	0,010000	0
	1,000000					
		-11,103305				
-10,103305						2
	-11,103305					
		20,5472305				
10,443925						4
	20,5472305					
		-15,209478				
-4,765552						6
	-15,209478					
		6,03126686				
1,265714						8
	6,03126686					
		-1,4881555				
-0,222441						10
	-1,4881555					
		0,25035521				
0,027914						12
	0,25035521					
		-0,0305469				
-0,002633						14
	-0,0305469					
		0,00282643				
0,000194						16
	0,00282643					
		-0,0002051				
-0,000011						18

## 6. Скриншоты



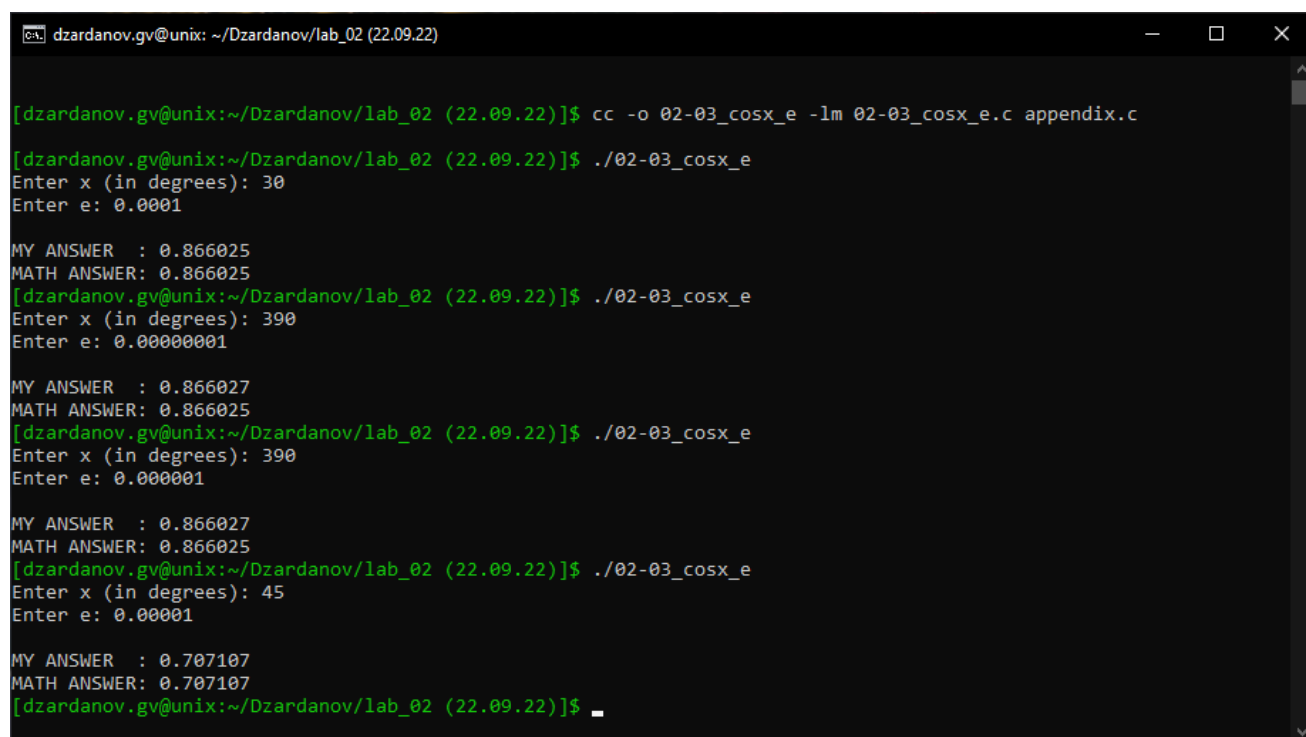
```
dzardanov.gv@unix: ~/Dzardanov/lab_02 (22.09.22)

[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ cc -o 02-02_cosx_n -lm 02-02_cosx_n.c appendix.c
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-02_cosx_n
Enter x (in degrees): 60
Enter n: 3

MY ANSWER : 0.501796
MATH ANSWER: 0.500000
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-02_cosx_n
Enter x (in degrees): 60
Enter n: 5

MY ANSWER : 0.500000
MATH ANSWER: 0.500000
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ _
```

Рис. 7: Сборка и запуск программы 02-02\_cosx\_n



```
dzardanov.gv@unix: ~/Dzardanov/lab_02 (22.09.22)

[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ cc -o 02-03_cosx_e -lm 02-03_cosx_e.c appendix.c
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-03_cosx_e
Enter x (in degrees): 30
Enter e: 0.0001

MY ANSWER : 0.866025
MATH ANSWER: 0.866025
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-03_cosx_e
Enter x (in degrees): 390
Enter e: 0.00000001

MY ANSWER : 0.866027
MATH ANSWER: 0.866025
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-03_cosx_e
Enter x (in degrees): 390
Enter e: 0.000001

MY ANSWER : 0.866027
MATH ANSWER: 0.866025
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ ./02-03_cosx_e
Enter x (in degrees): 45
Enter e: 0.0001

MY ANSWER : 0.707107
MATH ANSWER: 0.707107
[dzardanov.gv@unix:~/Dzardanov/lab_02 (22.09.22)]$ _
```

Рис. 8: Сборка и запуск программы 02-03\_cosx\_e

## 7. Выводы

В ходе выполнения данной лабораторной работы на примере двух программ, вычисляющих значение косинуса от введённого угла с помощью приближения с использованием разложения функции в ряды Тейлора, были рассмотрены базовые принципы построения программ на языке C и обработки вещественных чисел:

1. Организация ввода/вывода.
2. Разработка функций.
3. Объявление и использование переменных.
4. Выполнение простейших арифметических действий
5. Использование функций сторонних библиотек
6. Вынесение собственных функций в отдельные файлы