

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»**

Студент: Кругликова М. В.

Группа: Б22-504

Преподаватель: Комаров Т. И.

Москва – 2022

1. Формулировка индивидуального задания

Вариант №14. Вычислить значение функции в точке при помощи разложения в ряд:

$$\operatorname{arth} x = \ln \frac{\sqrt{1-x^2}}{1-x} = x + \frac{x^3}{3} + \frac{x^5}{5} + \dots = \sum_{n=0}^{\infty} \frac{1}{2n+1} x^{2n+1}$$

где $|x| < 1$

Тип данных: число с плавающей точкой одинарной точности — float
(спецификатор формата: %f)

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовались такие типы данных, как:

- встроенный тип данных int, предназначенный для работы с целыми числами
- встроенный тип данных float, предназначенный для работы с числами с плавающей точкой одинарной точности

3. Описание использованного алгоритма

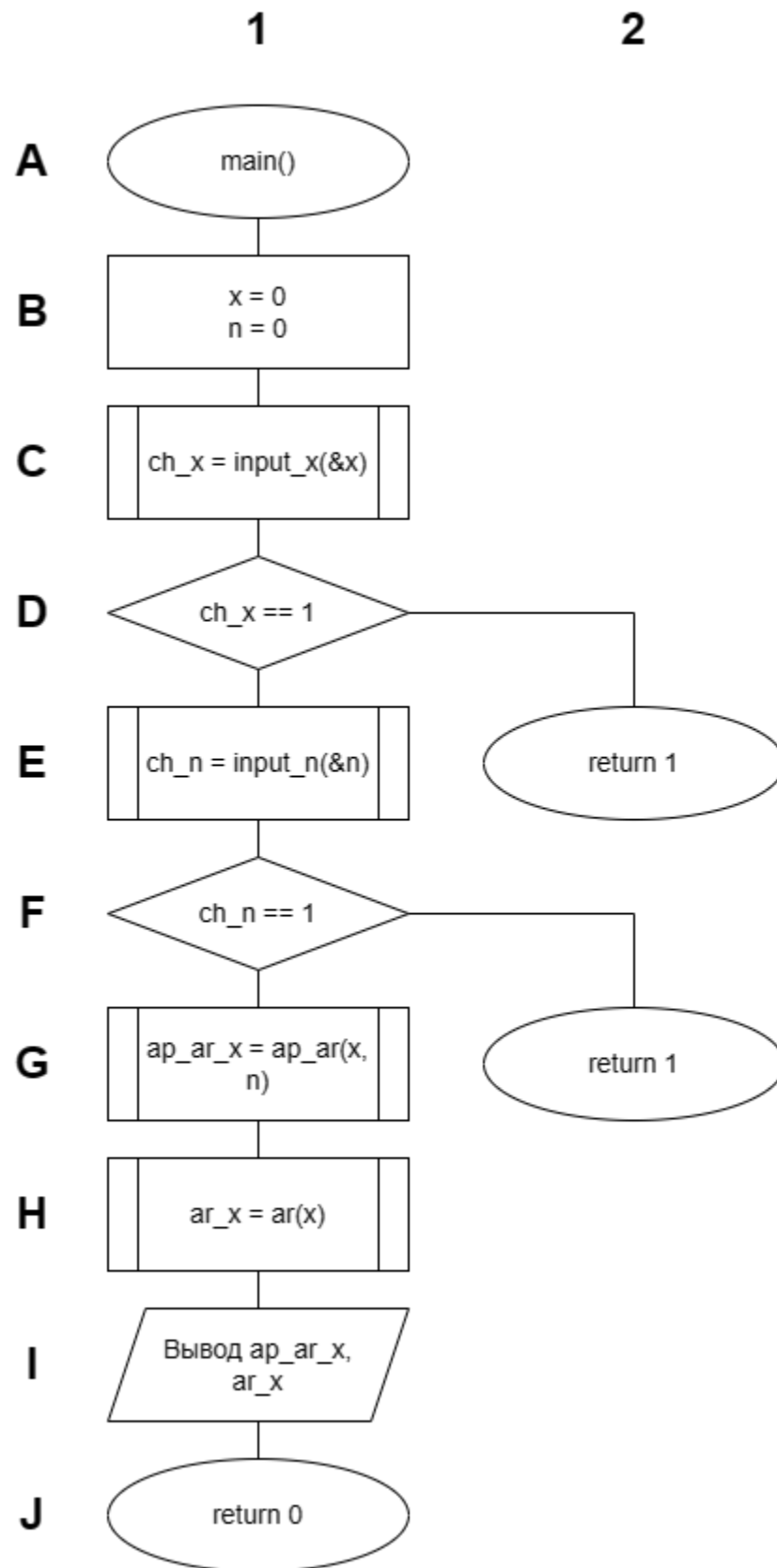


Рисунок 1. Блок-схема алгоритма работы функции main (используется в программе lab_2.1)

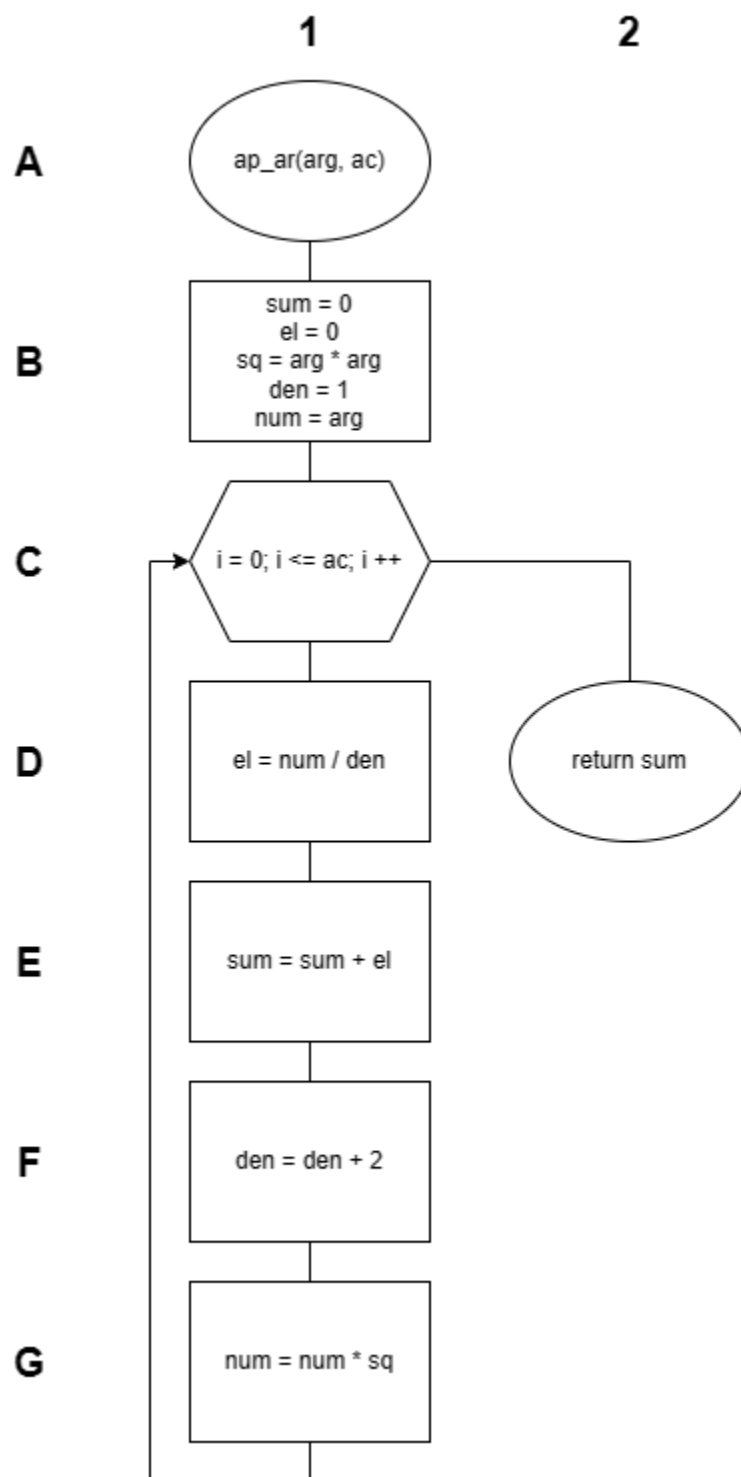


Рисунок 2. Блок-схема алгоритма работы функции `ap_ar` (используется в программе `lab_2.1`)

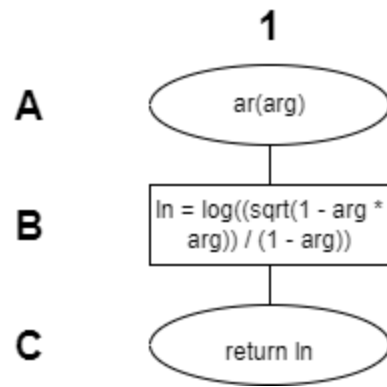


Рисунок 3. Блок-схема алгоритма работы функции ar (используется в программах lab_2.1 и lab_2.2)

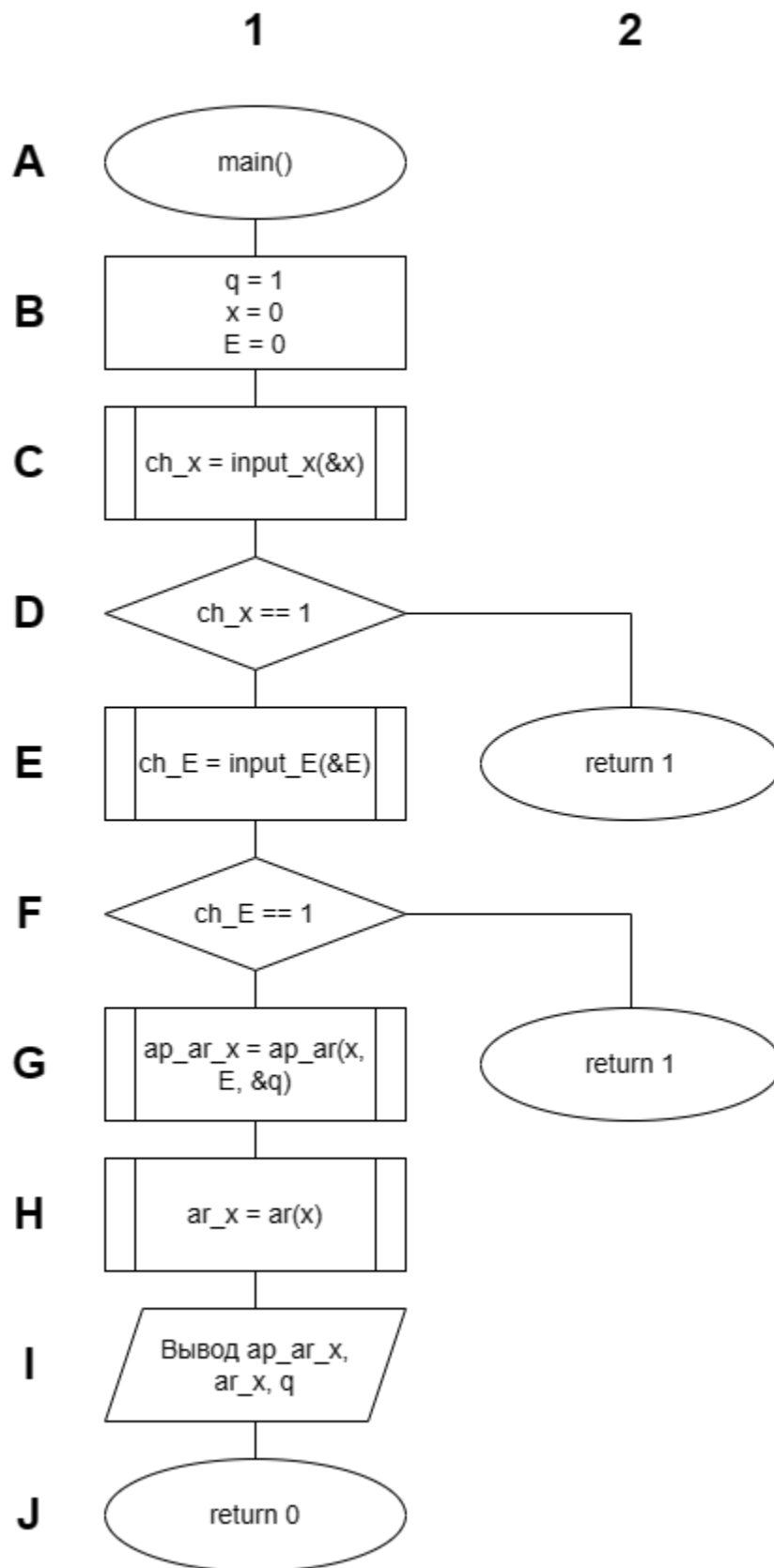


Рисунок 4. Блок-схема алгоритма работы функции main (используется в программе lab_2.2)

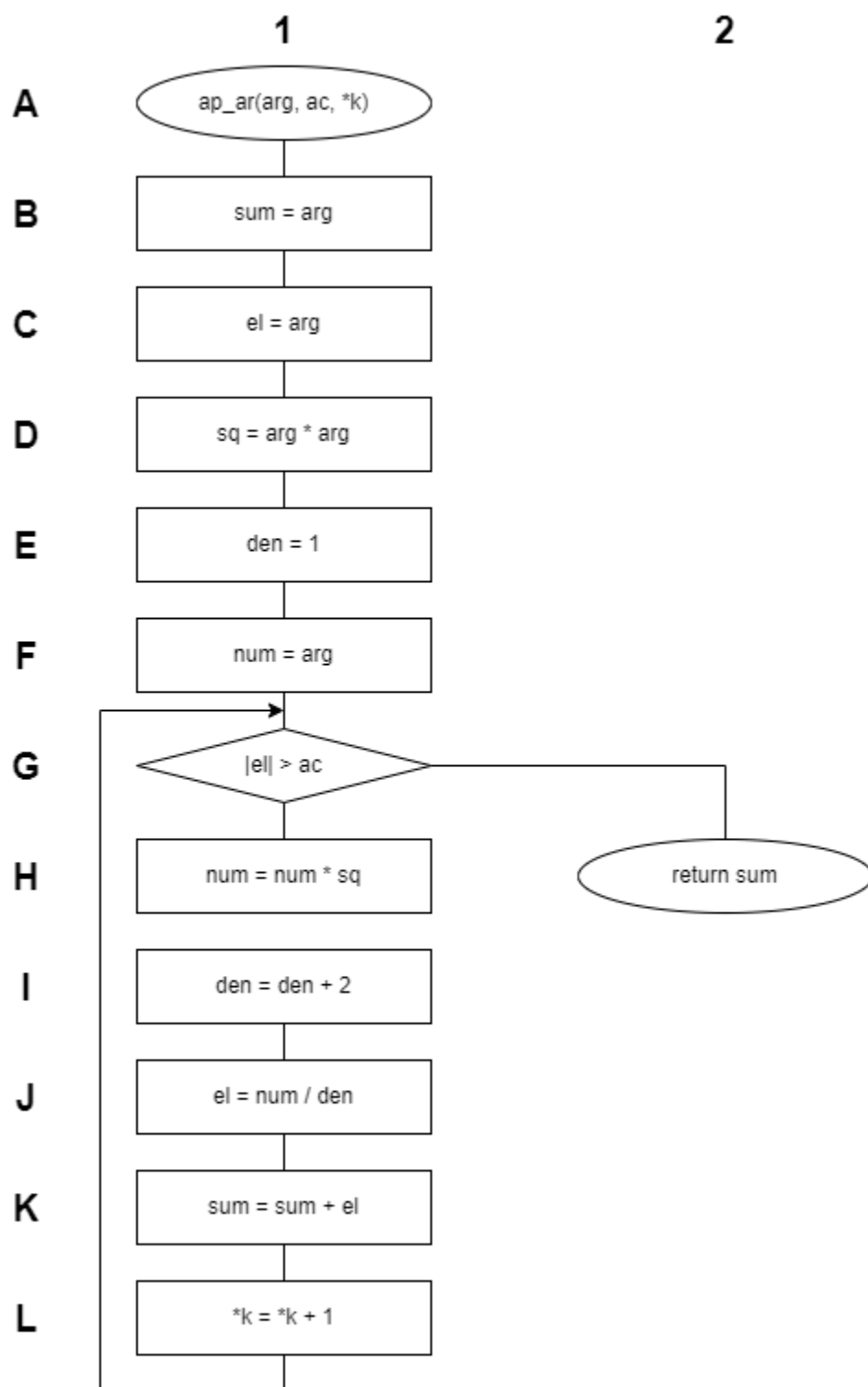


Рисунок 5. Блок-схема алгоритма работы функции `ap_ar` (используется в программе `lab_2.2`)

4. Исходные коды разработанных программ

Листинг 1. Исходные коды программы lab_2.1 (файл: lab_2.1.c)

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

float ap_ar(float arg, int ac) {
    float sum = 0;
    float el = 0;
    float sq = arg * arg;
    float den = 1;
    float num = arg;
    for (int i = 0; i <= ac; i++) {
        el = num / den;
        sum += el;
        den += 2;
        num *= sq;
    }
    return sum;
}

float ar(float arg) {
    float ln = log((sqrt(1 - arg * arg)) / (1 - arg));
    return ln;
}

int input_x(float *arg) {
    int t_f = 0;
    while ((t_f != 1) || (*arg <= -1) || (*arg >= 1)) {
        printf("Введите x (|x| < 1): ");
        t_f = scanf("%f", arg);
        if (t_f == EOF) {
            return 1;
        }
        if ((t_f != 1) || (*arg <= -1) || (*arg >= 1)) {
            printf("Введенное значение некорректно \n");
            scanf("%*[^\\n]");
        }
    }
    return 0;
}

int input_n(int *k) {
```



```

int t_i = 0;
while ((t_i != 1) || (*k <= 0)) {
    printf("Введите количество членов ряда: ");
    t_i = scanf("%d", k);
    if (t_i == EOF) {
        return 1;
    }
    if ((t_i != 1) || (*k <= 0)) {
        printf("Введенное значение некорректно \n");
        scanf("%*[^\\n]");
    }
}
return 0;
}

int main() {
    float x = 0;
    int n = 0;
    int ch_x = input_x(&x);
    if (ch_x == 1) {
        return 1;
    }
    int ch_n = input_n(&n);
    if (ch_n == 1) {
        return 1;
    }

    float ap_ar_x = ap_ar(x, n);
    float ar_x = ar(x);

    printf("Приближенный ареатенгенс: %f\\n", ap_ar_x);
    printf("Точный ареатангенс: %f\\n", ar_x);
    return 0;
}

```

Листинг 2. Исходные коды программы lab_2.2 (файл: lab_2.2.c)

```

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

float ap_ar(float arg, float ac, int *k) {
    float sum = arg;
    float el = arg;

```

```

float sq = arg * arg;
float den = 1;
float num = arg;
while (fabs(el) > ac) {
    num *= sq;
    den += 2;
    el = num / den;
    sum += el;
    *k += 1;
}
return sum;
}

float ar(float arg) {
    float ln = log((sqrt(1 - arg * arg)) / (1 - arg));
    return ln;
}

int input_x(float *arg) {
    int t_f = 0;
    while ((t_f != 1) || (*arg <= -1) || (*arg >= 1)) {
        printf("Введите x (|x| < 1): ");
        t_f = scanf("%f", arg);
        if (t_f == EOF) {
            return 1;
        }
        if ((t_f != 1) || (*arg <= -1) || (*arg >= 1)) {
            printf("Введенное значение некорректно \n");
            scanf("%*[^\\n]");
        }
    }
    return 0;
}

int input_E(float *ac) {
    int t_f = 0;
    while ((t_f != 1) || (*ac <= 0)) {
        printf("Введите точность: ");
        t_f = scanf("%f", ac);
        if (t_f == EOF) {
            return 1;
        }
        if ((t_f == 0) || (*ac <= 0)) {

```

```

        printf("Введенное значение некорректно \n");
        scanf("%*[^\\n]");
    }
}
return 0;
}

int main() {
    int q = 1;
    float x = 0;
    float E = 0;
    int ch_x = input_x(&x);
    if (ch_x == 1) {
        return 1;
    }
    int ch_E = input_E(&E);
    if (ch_E == 1) {
        return 1;
    }

    float ap_ar_x = ap_ar(x, E, &q);
    float ar_x = ar(x);

    printf("Приближенный ареатангенс: %f\\n", ap_ar_x);
    printf("Точный ареатангенс: %f\\n", ar_x);
    printf("Количество элементов ряда: %d\\n", q);
    return 0;
}

```

5. Описание тестовых примеров

Таблица 1. Тестовые примеры работы программы lab_2.1

Значение x	Значение n	Значение ap_ar_x (приближенный ареатангенс)	Значение ar_x (точный ареатангенс)
0.367	15	0.384952	0.384952
0	3	0	0
-0.1964	16	-0.198985	-0.198985

Таблица 2. Тестовые примеры работы программы lab_2.2

Значение x	Значение E	Значение ap_ar_x (приближенный ареатангенс)	Значение ar_x (точный ареатангенс)	Значение q (количество элементов ряда)

-0.543	0.001	-0.608253	-0.608400	5
0.37	0.0001	0.388421	0.388423	5

6. Скриншоты

```
kruglikova.mv@unix: ~
login as: kruglikova.mv
Keyboard-interactive authentication prompts from server:
| Password:
End of keyboard-interactive prompts from server
Last login: Sun Oct 16 13:26:18 2022 from 90.154.71.220

[kruglikova.mv@unix:~]$ ./lab_2.1
Введите x ( $|x| < 1$ ): 0.367
Введите количество членов ряда: 15
Приближенный ареатангенс: 0.384952
Точный ареатангенс: 0.384952

[kruglikova.mv@unix:~]$ ./lab_2.1
Введите x ( $|x| < 1$ ): 0
Введите количество членов ряда: 3
Приближенный ареатангенс: 0.000000
Точный ареатангенс: 0.000000

[kruglikova.mv@unix:~]$ ./lab_2.1
Введите x ( $|x| < 1$ ): -0.1964
Введите количество членов ряда: 16
Приближенный ареатангенс: -0.198985
Точный ареатангенс: -0.198985
```

Рисунок 6. Запуск программы lab_2.1

```
[kruglikova.mv@unix:~]$ ./lab_2.2
Введите x ( $|x| < 1$ ): -0.543
Введите точность: 0.001
Приближенный ареатангенс: -0.608253
Точный ареатангенс: -0.608400
Количество элементов ряда: 5

[kruglikova.mv@unix:~]$ ./lab_2.2
Введите x ( $|x| < 1$ ): 0.37
Введите точность: 0.0001
Приближенный ареатангенс: 0.388421
Точный ареатангенс: 0.388423
Количество элементов ряда: 5
```

Рисунок 7. Запуск программы lab_2.2

7. Выводы

В ходе выполнения данной работы были изучены:

1. Способ проверки корректности входных данных
2. Принципы работы с указателями
3. Различия циклов for и while