

Национальный исследовательский ядерный университет «МИФИ»
Институт интеллектуальных кибернетических систем
Кафедра №12 «Компьютерные системы и технологии»



ОТЧЕТ

О выполнении лабораторной работы №2
«Ряды. Числа с плавающей запятой в Си»

Студент: Титов Д.И.
Группа: Б22-505
Преподаватель: Вавренюк А.Б.

1 Формулировка индивидуального задания

Вариант №64

Задание

Вычислить значение функции в точке при помощи разложения в ряд:

$$\operatorname{sh} x = \frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots = \sum_{n=0}^{\infty} \frac{1}{(2n+1)!} x^{2n+1}$$

где $|x| < \infty$.

2 Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных `int`, предназначенный для работы с целыми числами, встроенный тип данных `long double`, предназначенный для работы с числами с плавающей запятой, а также указатели типа `int` и типа `long double`.

3 Описание использованного алгоритма

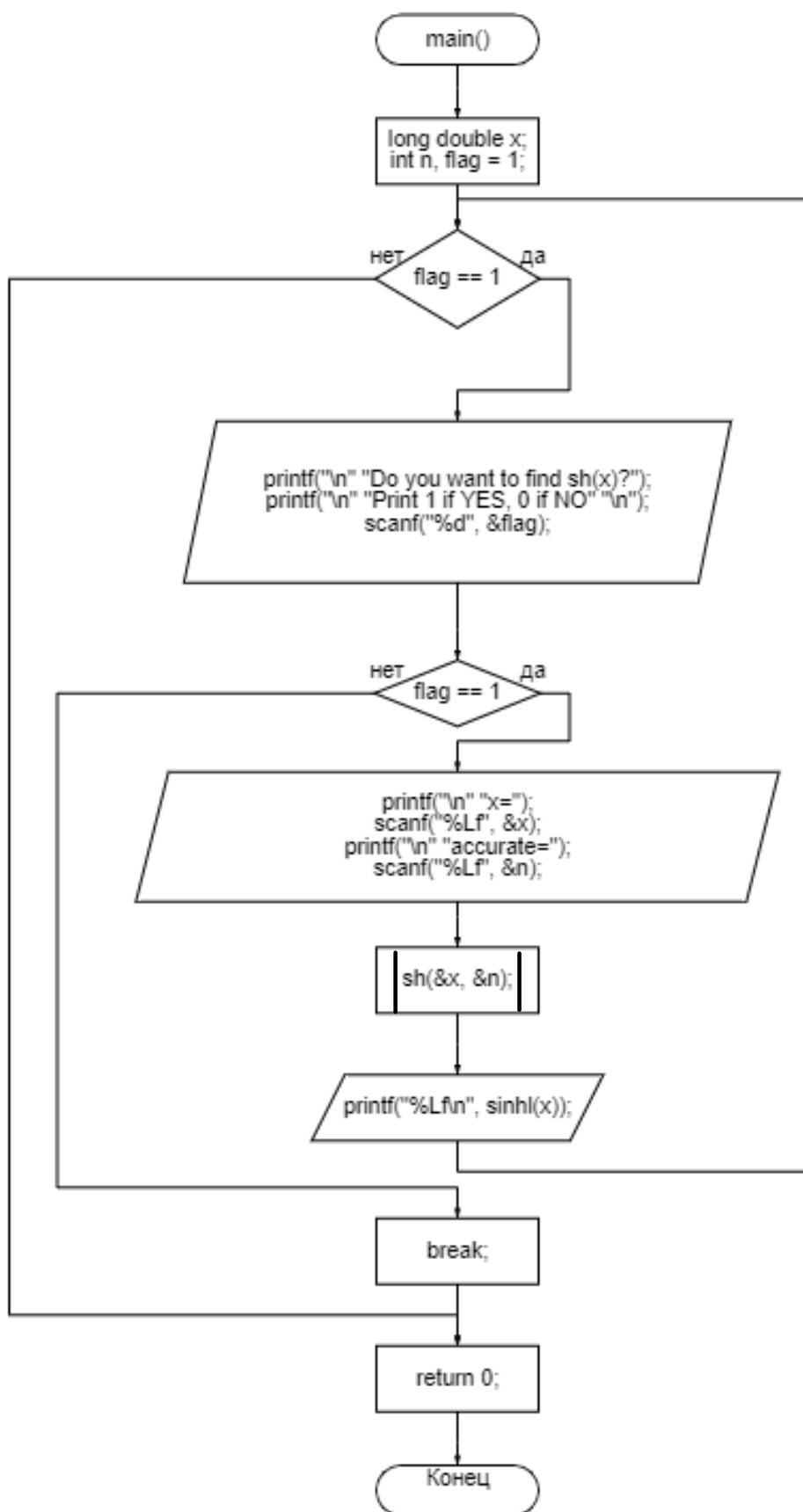


Рис. 1: Блок-схема алгоритма работы функции `main()` в коде подсчета через количество элементов.

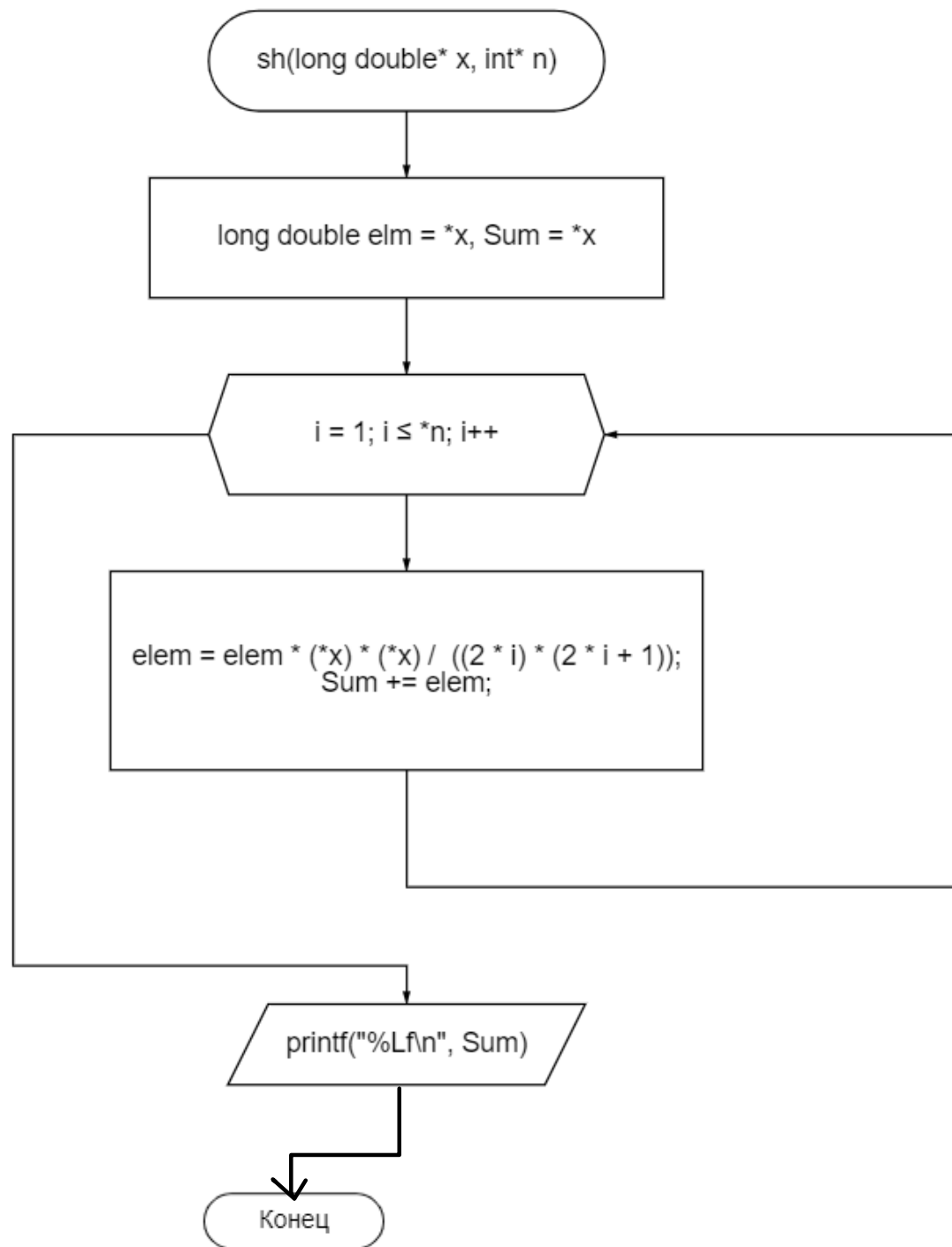


Рис. 2: Блок-схема алгоритма работы функции `sh()` в коде подсчета через количество элементов.

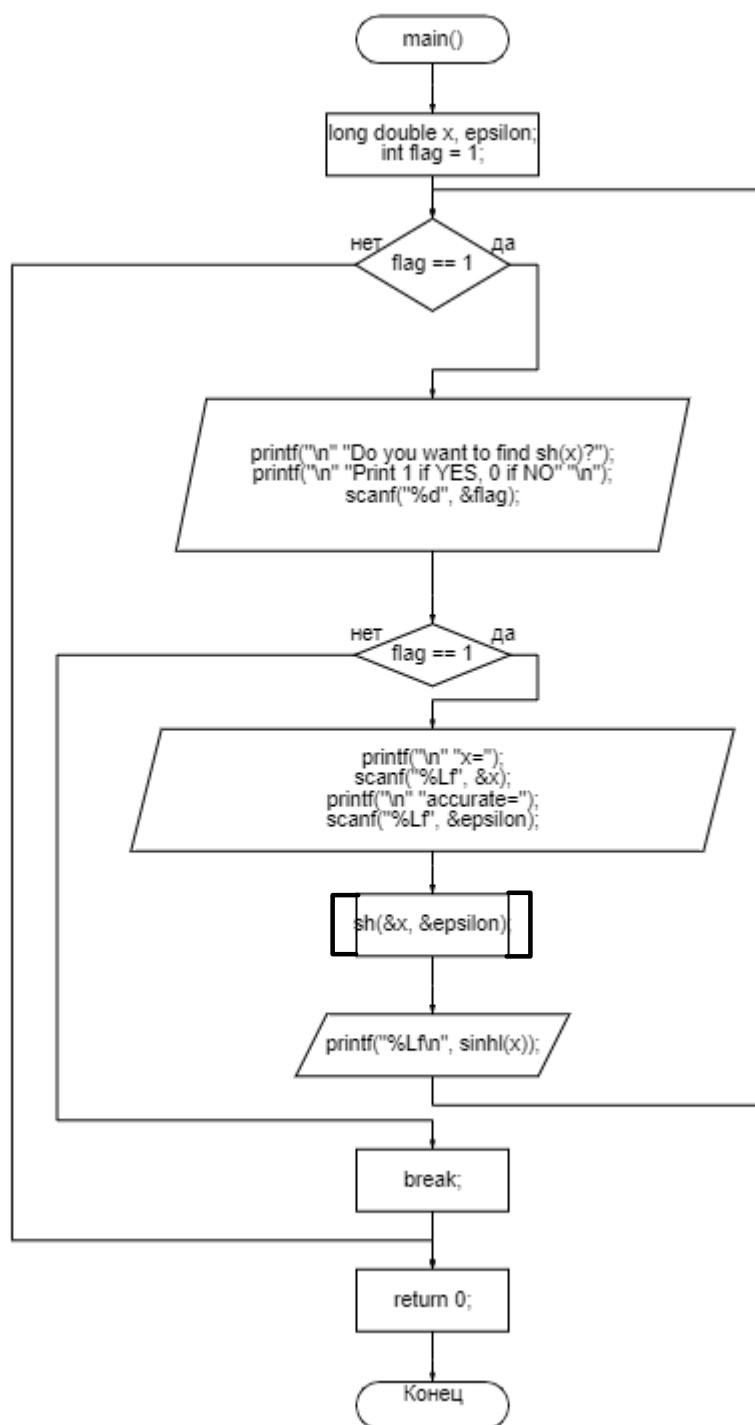


Рис. 3: Блок-схема алгоритма работы функции `main()` в коде подсчета через эпсилон.

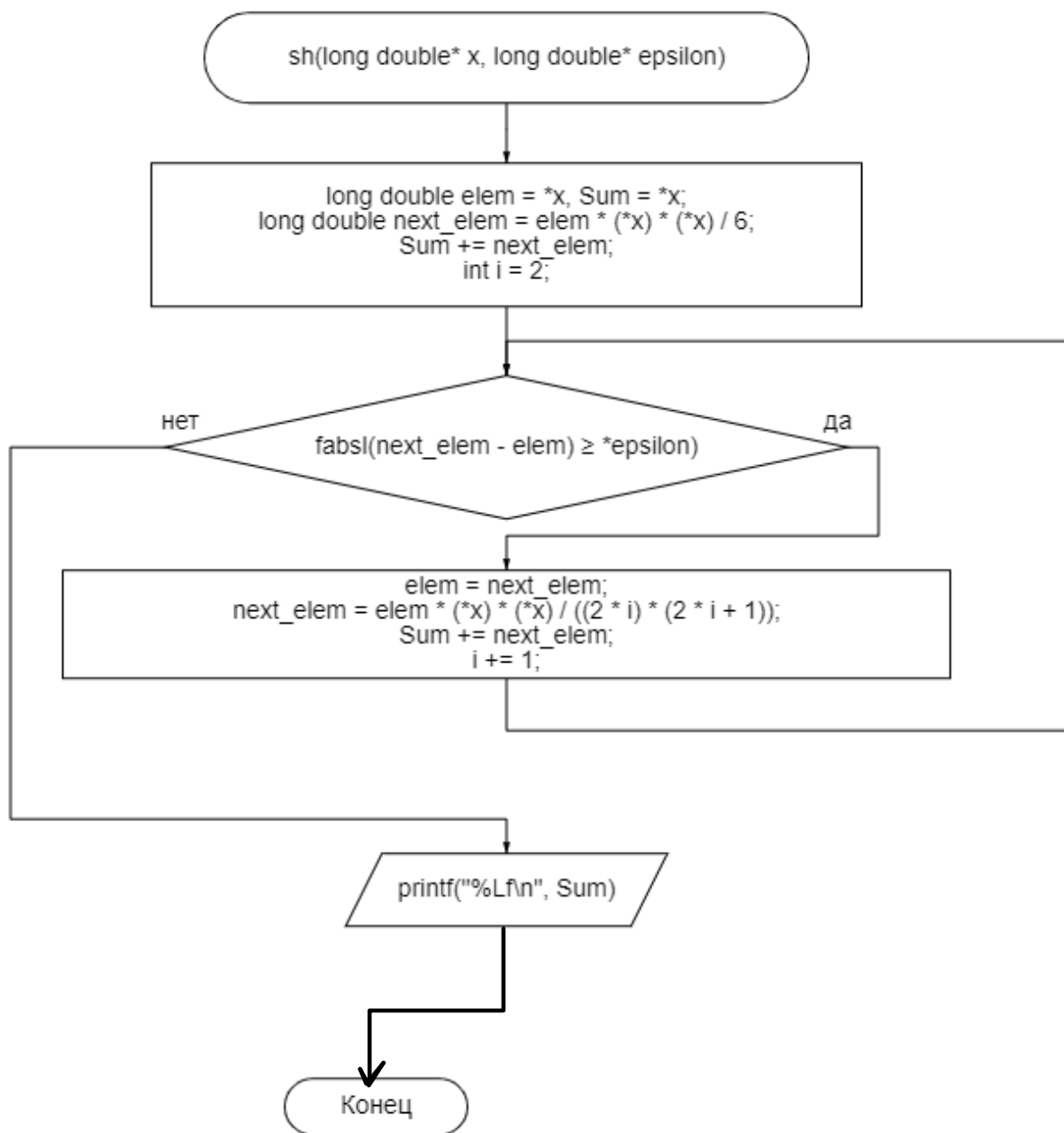


Рис. 4: Блок-схема алгоритма работы функции `sh ()` в коде подсчета через эпсилон.

4 Исходные коды разработанных программ

Листинг 1: Исходные коды программы prog2 (файл: progrow.c)

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

void sh(long double* x, int* n);

int main(){
    long double x;
    int n, flag = 1;
    while(flag == 1){
        printf("\n" "Do you want to find sh(x)?");
        printf("\n" "Print 1 if YES, 0 if NO" "\n");
        scanf("%d", &flag);
        if(flag == 1){
            printf("\n" "x=");
            scanf("%Lf", &x);
            printf("\n" "accurate=");
            scanf("%d", &n);
            sh(&x, &n);
            printf("%Lf\n", sinh1(x));
        } else{
            break;
        }
    }
    return 0;
}

void sh(long double* x, int* n) {
    long double elem = *x, Sum = *x;
    for(int i = 1; i <= *n; i++){
        elem = elem * (*x) * (*x) / ((2 * i) * (2 * i + 1));
        Sum += elem;
    }
    printf("%Lf\n", Sum);
}
```

Листинг 2: Исходные коды программы prog2 (файл: progeps.c)

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

void sh(long double* x, long double *epsilon);

int main(){
    long double x, epsilon;
    int flag = 1;
    while(flag == 1){
        printf("\n" "Do you want to find sh(x)?");
        printf("\n" "Print 1 if YES, 0 if NO" "\n");
        scanf("%d", &flag);
        if(flag == 1){
            printf("\n" "x=");
            scanf("%Lf", &x);
            printf("\n" "accurate=");
            scanf("%Lf", &epsilon);
            sh(&x, &epsilon);
            printf("%Lf\n", sinh1(x));
        } else{
            break;
        }
    }
    return 0;
}

void sh(long double* x, long double* epsilon) {
    long double elem = *x, Sum = *x;
    long double next_elem = elem * (*x) * (*x) / 6;
    Sum += next_elem;
    int i = 2;
    while(fabs1(next_elem - elem) >= *epsilon){
        elem = next_elem;
        next_elem = elem * (*x) * (*x) / ((2 * i) * (2 * i + 1));
        Sum += next_elem;
        i += 1;
    }
    printf("%Lf\n", Sum);
}
```

5 Описание тестовых примеров

Таблица 1: Тестовые примеры в коде подсчета через количество элементов.

Значение x	Значение accurate	Ожидаемые значения	Полученные значения
1	20	1.175201	1.175201
-0.35	30	-0.357190	-0.357190
10	100	11013.232875	11013.232875

Таблица 2: Тестовые примеры в коде подсчета через эпсилон.

Значение x	Значение accurate	Ожидаемые значения	Полученные значения
---------------	----------------------	--------------------	---------------------

1	1	1.175201	1.166667
-0.35	0.1	-0.357190	-0.357190
10	0.001	11013.232875	11013.232874

6 Скриншоты

```
[titov.di@unix:~/Lab2]$ ./row1.out

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=1

accurate=20
1.175201
1.175201

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=-0.35

accurate=30
-0.357190
-0.357190

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=10

accurate=100
11013.232875
11013.232875

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
0
```

Рис. 5: Сборка и запуск программы `progrow.c`

```
[titov.di@unix:~/Lab2]$ ./eps.out

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=-0.35

accurate=0.1
-0.357190
-0.357190

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=10

accurate=0.001
11013.232874
11013.232875

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
1

x=1

accurate=1
1.166667
1.175201

Do you want to find sh(x)?
Print 1 if YES, 0 if NO
0
```

Рис. 6: Сборка и запуск программы `progeps.c`

7 Выводы

В ходе выполнения данной работы на примере программы, выполняющей вычисление гиперболического синуса заданного числа через разложения его в ряд по количеству элементов в ряду и через эпсилон, были рассмотрены принципы работы с числами с плавающими запятыми в языке Си.