

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: Мартыненко Д. М.

Группа: С22-501

Преподаватель: Овчаренко Е. С.

Москва — 2022

1. Формулировка индивидуального задания

Вариант №74: Вычислить значение функции в точке при помощи разложения в ряд:

$$\sin^3 x = \frac{3}{4} \left(\frac{8x^3}{3!} - \frac{80x^5}{5!} + \frac{728x^7}{7!} - \dots \right) = \frac{3}{4} \sum_{n=1}^{\infty} (-1)^{n+1} \frac{3^{2n} - 1}{(2n+1)!} x^{2n+1}, |x| < \infty \quad (1)$$

2. Описание использованных типов данных

При выполнении лабораторной работы были использованный следующие типы данных:

1. Тип `int` для `for-loop` циклов, а также определения количества членов ряда.
2. Тип `float` для подсчёта ряда с высокой точностью.

3. Описание использованного алгоритма

3.1. Описание алгоритма подсчёта ряда по количеству членов

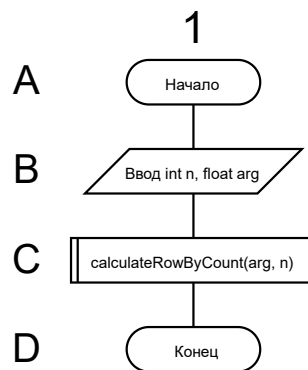


Рис. 1: Блок-схема алгоритма работы функции `main()` в библиотеке `rows`

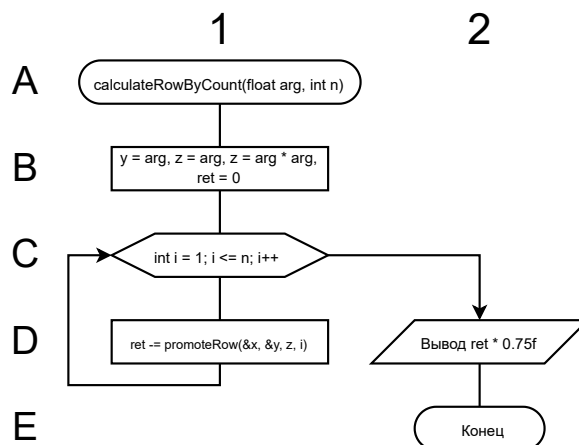


Рис. 2: Блок-схема алгоритма работы функции `calculateRowByCount()`

3.2. Описание алгоритма подсчёта ряда по заданной точности

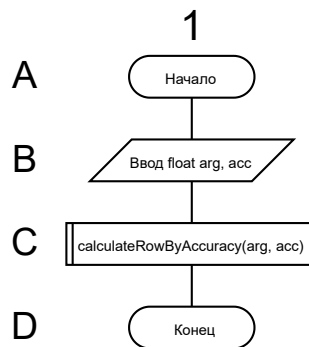


Рис. 3: Блок-схема алгоритма работы функции `main()` в библиотеке `accuracy`

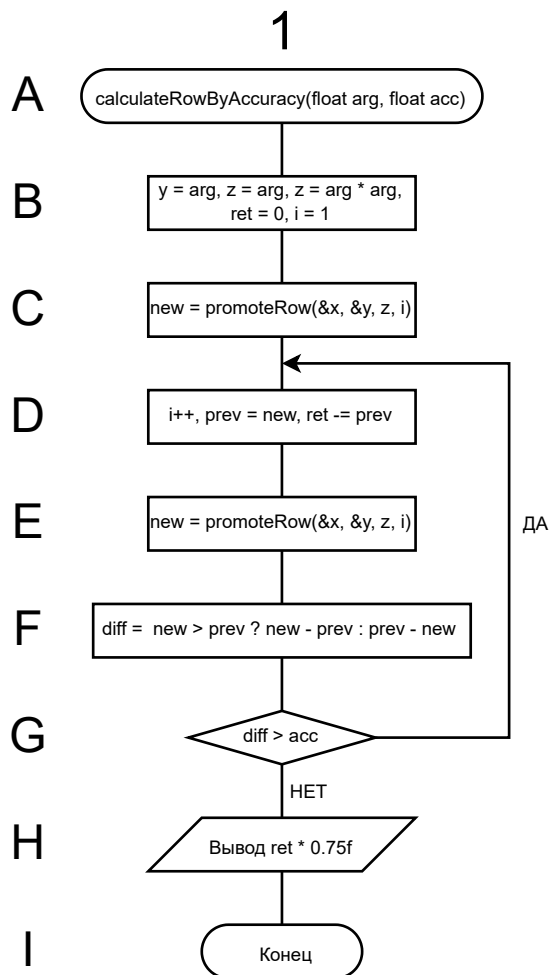


Рис. 4: Блок-схема алгоритма работы функции `calculateRowByAccuracy()`

3.3. Описание алгоритма общей части подсчёта ряда

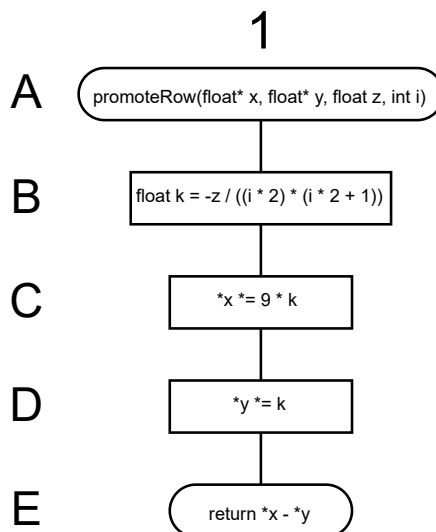


Рис. 5: Блок-схема алгоритма работы функции `promoteRow()`

4. Исходные коды разработанных программ

4.1. Исходный код программы для подсчёта по количеству членов ряда

Исходный код 1: библиотека `rows.c`

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #include "include/common.h"
5
6 float calculateRowByCount(float, int);
7
8 int main() {
9     float arg = 0;
10    int n = 1;
11
12    clearOutput();
13    printf("Specify the value (float) of function argument: ");
14    getFloat(&arg);
15    printf("Specify the value (int) of rows count: ");
16    while (1) {
17        getInt(&n);
18        if (n > 0) break;
19        printError("Rows count can't be less than or equals zero, try again: ");
20    }
21
22    printf("\n");
23    printf("Calculated sin(%g)^3 with %d rows: %.16lf\n", arg, n, calculateRowByCount(arg, n));
24    printf("Expected value (calculated with default impl) is: %.16lf\n", pow(sin((double)arg), 3));
25
26    return 0;
27 }
28
29 float calculateRowByCount(float arg, int n) {
```

```

30     float x = arg, y = arg, z = arg * arg, ret = 0;
31
32     for (int i = 1; i <= n; i++) {
33         ret -= promoteRow(&x, &y, z, i);
34     }
35
36     return ret * 0.75f;
37 }

```

4.2. Исходный код программы для подсчёта по заданной точности

Исходный код 2: библиотека accuracy.c

```

1  #include <stdio.h>
2  #include <math.h>
3
4  #include "include/common.h"
5
6  float calculateRowByAccuracy(float arg, float acc);
7
8  int main() {
9     float arg = 0, acc = 0;
10
11     clearOutput();
12     printf("Specify the value (float) of function argument: ");
13     getFloat(&arg);
14     printf("Specify the value (float) of function accuracy: ");
15     while (1) {
16         getFloat(&acc);
17         if (acc < 1) break;
18         printError("Accuracy cannot be more than or equal one, try again: ");
19     }
20
21     printf("\n");
22     printf("Calculated sin(%g)^3 with %g accuracy: %.16lf\n", arg, acc, calculateRowByA);
23     printf("Expected value (calculated with default impl) is: %.16lf\n", pow(sin((double)
24
25     return 0;
26 }
27
28 float calculateRowByAccuracy(float arg, float acc) {
29     float x = arg, y = arg, z = arg * arg, ret = 0;
30     float new = promoteRow(&x, &y, z, 1), prev, diff;
31
32     int i = 1;
33     do {
34         i++;
35         prev = new;
36         ret -= prev;
37         new = promoteRow(&x, &y, z, i);
38         diff = new > prev ? new - prev : prev - new;
39     } while (diff > acc);
40
41     return ret * 0.75f;
42 }

```

4.3. Исходный код общей библиотеки подсчёта ряда

Исходный код 3: библиотека common.c.

```

1  #include <stdio.h>

```

```

2  #include <stdlib.h>
3
4  #include "include/common.h"
5
6  float promoteRow(float* x, float* y, float z, int i) {
7      float k = -z / ((i * 2) * (i * 2 + 1));
8      *x *= 9 * k;
9      *y *= k;
10     return *x - *y;
11 }
12
13 void printError(const char* msg) {
14     printf("\033[1;31m");
15     printf("%s", msg);
16     printf("\033[0m");
17 }
18
19 void getFloat(float *f) {
20     while(!scanf("%f", f)) {
21         onBadInput();
22     }
23 }
24
25 void getInt(int *i) {
26     while(!scanf("%d", i)) {
27         onBadInput();
28     }
29 }
30
31 void onBadInput() {
32     scanf("%*[^\\n]");
33     printf("That is NOT a valid input, try again:\\n");
34 }
35
36 void clearOutput() {
37     #ifdef _WIN32
38         system("cls");
39     #else
40         system("clear");
41     #endif
42 }

```

Исходный код 4: заголовочный файл include/common.h.

```

1  #ifndef UTILS_H
2  #define UTILS_H
3
4  float promoteRow(float*, float*, float, int);
5  void printError(const char*);
6  void getFloat(float*);
7  void getInt(int*);
8  void onBadInput();
9  void clearOutput();
10
11 #endif //UTILS_H

```

5. Описание тестовых примеров

5.1. Тестовые примеры для подсчёта по количеству членов ряда

Таблица 1: Тестовые примеры (подсчёт по количеству членов ряда)

Аргумент функции	Количество членов	Значение с <code>math.h</code>	Полученное значение
1.4	20	0.9569812287889454	0.9569808840751648
-1	10	-0.5958232365909556	-0.5958232283592224
5	40	-0.8817651660366330	-0.8916047811508179
-3.17263	200	0.0000298846083317	0.0000606863395660
0	100	0.0000000000000000	0.0000000000000000

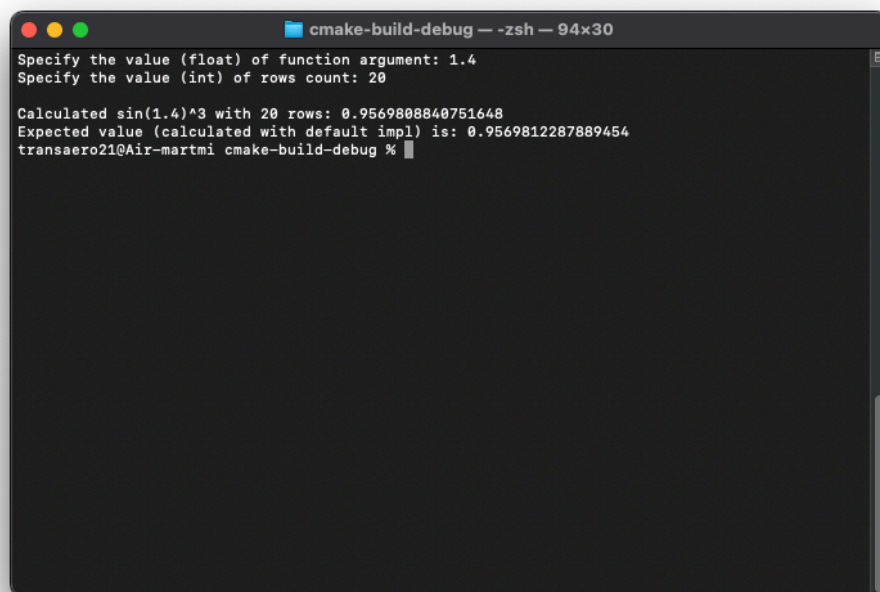
5.2. Тестовые примеры для подсчёта по заданной точности

Таблица 2: Тестовые примеры (подсчёт по заданной точности)

Аргумент функции	Заданная точность	Значение с <code>math.h</code>	Полученное значение
1	0.1	0.5958232365909556	0.5947751402854919
-0.9	0.0003	-0.4806501848808332	-0.4806496202945709
4.12937	0.0009	-0.5817740149783298	-0.5823402404785156
1.234567	0.9	0.8412469450644317	0.9212406277656555
3.21	0.0000001	-0.0003193689160720	-0.0003066345525440

6. Скриншоты

6.1. Программы для подсчёта по количеству членов ряда

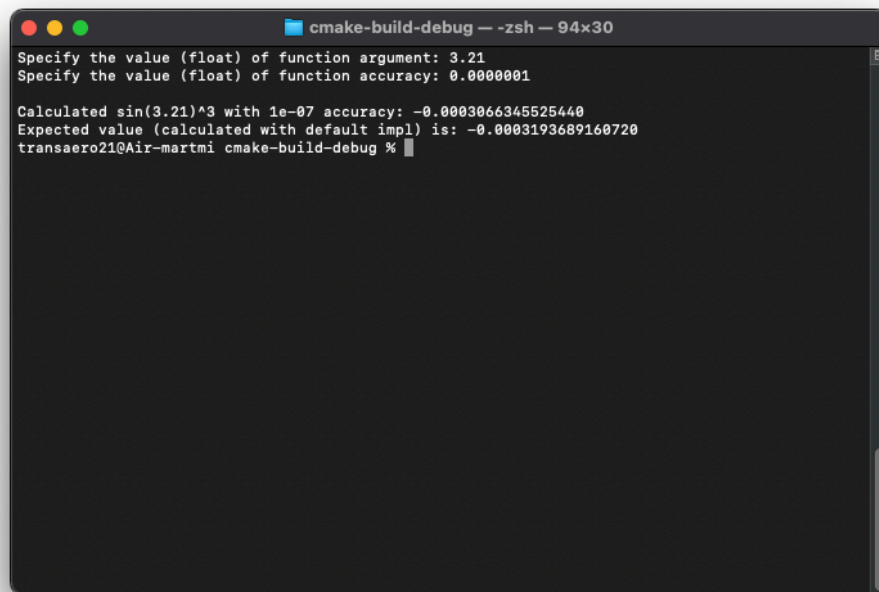


```
cmake-build-debug --zsh -- 94x30
Specify the value (float) of function argument: 1.4
Specify the value (int) of rows count: 20

Calculated sin(1.4)^3 with 20 rows: 0.9569808840751648
Expected value (calculated with default impl) is: 0.9569812287889454
transaero21@Air-martmi cmake-build-debug %
```

Рис. 6: Успешный подсчёт с аргументом 1.4 и 20 членами ряда.

6.2. Программы для подсчёта по заданной точности



```
cmake-build-debug — zsh — 94x30
Specify the value (float) of function argument: 3.21
Specify the value (float) of function accuracy: 0.0000001

Calculated sin(3.21)^3 with 1e-07 accuracy: -0.0003066345525440
Expected value (calculated with default impl) is: -0.0003193689160720
transaero21@Air-martmi cmake-build-debug %
```

Рис. 7: Успешный подсчёт с аргументов 3.21 и точностью 0.0000001.

7. Выводы

В ходе выполнения данной работы были:

1. Изучена работы с вещественный тип данных `float`.
2. Рассмотрены пределы целых и вещественных чисел.
3. Изучены ряды и рассмотрена работа с ними.
4. Изучена разработка программ с использования заголовочных файлов с общими библиотеками.