## Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития Кафедра инфо коммуникаций

## ОТЧЕТ ПО ЛАБОРАТОРНОЙРАБОТЕ №2.23 Дисциплины «Программирование на Python»

	Выполнил: Волошин Алексей Вадимович 2 курс, группа ИТС-б-о-22-1, 11.03.02«Инфокоммуникационные технологии и системы связи», направленность (профиль) «Инфокоммуникационные системы и сети», очная форма обучения
	(подпись)
	Руководитель практики: <u>Воронкин Р.</u> А, канд. техн. наук, доцент кафедры инфокоммуникаций
	(подпись)
Отчет защищен с оценкой	Дата защиты

**Цель работы:** приобретение навыков написания многопоточных приложений на языке программирования Python версии 3.х.

## Индивидуальное задание

С использованием многопоточности для заданного значения найти сумму ряда с точностью члена ряда по абсолютному значению 1e-07 и произвести сравнение полученной суммы с контрольным значением функции для двух бесконечных рядов. Результат выполнения отображен на рисунке 1.

## Ответы на контрольные вопросы

- 1. Непосредственно **модуль sqlite3** это API к СУБД SQLite. Своего рода адаптер, который переводит команды, написанные на Питоне, в команды, которые понимает SQLite. Как и наоборот, доставляет ответы от SQLite в руthon-программу.
- 2. Для взаимодействия с базой данных SQLite3 в Python необходимо создать объект cursor. Вы можете создать его с помощью метода cursor() . Курсор SQLite3 – это метод объекта соединения. Для выполнения инструкций

SQLite3 сначала устанавливается соединение, а затем создается объект курсора с использованием объекта соединения

- 3. При создании соединения с SQLite3 автоматически создается файл базы данных, если он еще не существует. Этот файл базы данных создается на диске, мы также можем создать базу данных в оперативной памяти с помощью функции :memory: with the connect. Такая база данных называется базой данных в памяти.
  - 4. С помощью команды закрытия close().
- 5. Чтобы вставить данные в таблицу, используется оператор INSERT INTO.
- 6. Чтобы обновить данные в таблице, просто создайте соединение, затем создайте объект курсора с помощью соединения и, наконец, используйте оператор UPDATE.
- 7. Оператор SELECT используется для выбора данных из определенной таблицы. Если вы хотите выбрать все столбцы данных из таблицы, вы можете использовать звездочку (\*).
- 8. SQLite3 rowcount используется для возврата количества строк, которые были затронуты или выбраны последним выполненным SQL-запросом.
- 9. Чтобы перечислить все таблицы в базе данных SQLite3, вы должны запросить данные из таблицы sqlite\_master, а затем использовать fetchall() для получения результатов из инструкции SELECT
- 10. При создании таблицы мы должны убедиться, что она еще не существует. Аналогично, при удалении/удалении таблицы она должна существовать. Чтобы проверить, не существует ли таблица уже, мы используем IF NOT EXISTS с оператором CREATE TABLE следующим образом.
- 11. Метод executemany можно использовать для вставки нескольких строк одновременно.

12. В базе данных Python SQLite3 мы можем легко хранить дату или время, импортируя модуль datetime . Следующие форматы являются наиболее часто используемыми форматами для datetime::