|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №2  по дисциплине «Язык программирования Python»  Вариант 5 | | |
|  | | Работу выполнил  студент группы ИТ-7,8-2025 1 курса  Баранов Е.А  «11» июня 2025 г. |
| Работу проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И.О.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc153130027)

[Алгоритм решения 3](#_Toc153130028)

[Тестирование 3](#_Toc153130029)

[Код программы 3](#_Toc153130030)

[Инструкция по применению стилей и оформлению работы 4](#_Toc153130031)

# Постановка задачи

Числовое кольцо. В кольце записаны N чисел, составляющих по часовой стрелке 3 числа: два слагаемых и сумму. Требуется написать программу, которая по строке чисел, считая ее кольцом, находит какое-нибудь решение в виде A+B=C. Все цифры числа должны входить в числа только один раз и в порядке следования в кольце. Цифр в кольце не более 1000. Формат входных данных: Входной файл содержит 1 строку, в которой без пробелов перечислены цифры кольца. Формат выходных данных: В выходной файл выводится тождество в виде += без пробелов внутри или слово «No», если решения не существует. Пример 1: (входные данные) 01902021 (выходные данные) 190+20=210 Пример 2: (входные данные) 111111 (выходные данные) No

# Алгоритм решения

1. Инициализация объекта кольца:

- Создать объект NumberRing, который хранит строку и её длину.

2. Функция получения последовательности цифр с учётом кольца:

- Метод get\_sequence(start, length) возвращает подстроку длины length, начиная с позиции start в кольце.

- Если последовательность выходит за конец строки, происходит "оборачивание" — берутся цифры с начала строки.

3. Поиск уравнения A + B = C:

- Перебираются все возможные длины чисел A и B, при этом длина C = N - (|A| + |B|).

- Для каждой тройки длин и для каждого возможного стартового индекса в кольце:

- Получить числа A, B, C с помощью get\_sequence.

- Проверить, что числа не начинаются с ведущего нуля (если длина > 1).

- Преобразовать строки в целые числа.

- Проверить равенство A + B = C.

- Если равенство выполняется, вернуть строку в формате "A+B=C".

4. Если подходящего уравнения не найдено, вывести "No".

▌Особенности реализации

- Используется кольцевая индексация для получения чисел.

- Проверка ведущих нулей исключает некорректные варианты.

- Перебор всех вариантов — перебираются все возможные длины A и B, а также все стартовые позиции.

- Для оптимизации при длине кольца ≥ 8 есть попытка найти конкретное уравнение с длинами (3, 2, 3).

- Если все цифры одинаковые (например, "111111"), сразу выводится "No".

▌Сложность алгоритма

- В худшем случае перебираются все варианты длины A и B, что порядка O(N^2).

- Для каждого варианта перебираются все стартовые позиции — ещё O(N).

- Итого сложность порядка O(N^3), что приемлемо для N ≤ 1000.

# Тестирование

Ввод:

  
 Вывод:



Ввод:

  
 Вывод:



Ввод:

  
 Вывод:



Ввод:

  
 Вывод:



Ввод:

  
 Вывод:



При вводе неизвестного имени файла выводит:



# Код программы

[Ссылка на GIT.](https://github.com/Evgeniy641/-Python)