МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования «Санкт-петербургский государственный морской технический университет»

ФАКУЛЬТЕТ ЦИФРОВЫХ ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЙ

Кафедра Киберфизических систем

ЛАБОРАТОРНАЯ РАБОТА

По дисциплине «Программирование»

Выполнил: СТЕЦЕНКО АЛЕКСАНДР АЛЕКСЕЕВИЧ

Проверил: ПОДЕЛЕНЮК ПАВЕЛ ПЕТРОВИЧ

Санкт-Петербург

2024

Оглавление

[1. Цели и формулировка задачи 3](#_Toc187676633)

[2. Результаты работы 4](#_Toc187676634)

[2.1. Реализация программы с использованием функционального программирования языка Python 4](#_Toc187676635)

[2.1.1. Ход работы 4](#_Toc187676636)

[2.1.2. Демонстрация работы программы 4](#_Toc187676637)

[2.1.3. Листинг кода 4](#_Toc187676638)

[ЗАКЛЮЧЕНИЕ 5](#_Toc187676639)

[Список использованных источников 6](#_Toc187676640)

# Цели и формулировка задачи

Даны N целых чисел X1,X2,…,XN​. Требуется расставить между ними знаки + и − так, чтобы значение получившегося выражения было равно заданному целому числу S.

#### Входные данные:

* Считываются из файла через пробел в следующем порядке:  
  N X1 X2 … XN S,

#### Выходные данные:

* В файл необходимо вывести равенство с расставленными знаками, которое соответствует заданной сумме S, в формате:  
  X1±X2±⋯±XN=S.
* Если решения нет, вывести "no solution".
* Если найдено несколько решений, можно вывести любое.

#### Условие:

Решение должно быть реализовано исключительно через **рекурсию**.

# Результаты работы

## **Реализация программы с использованием функционального программирования языка Python**

## Ход работы

Для решения задачи был разработан скрипт на языке Python, который находит арифметическое выражение с использованием знаков "+" и "-" между заданными числами, чтобы получить указанную сумму. Основой решения является использование рекурсии для перебора всех возможных комбинаций знаков.

#### Этапы решения:

#### 1. **Анализ задачи**

* Требуется расставить знаки между числами так, чтобы итоговая сумма была равна заданному значению.
* Числа и целевая сумма считываются из входного файла, где строка имеет формат:  
  N X1 X2 ... XN S, где:
  + N — количество чисел,
  + X1 … XN — массив чисел,
  + S — целевая сумма.
* Выходные данные записываются в файл в формате:  
  X1 +/- X2 +/- ... +/- XN = S. Если решения нет, записывается "no solution".

#### 2. **Разработка алгоритма**

##### Основной принцип работы:

* Используется рекурсивный подход для перебора всех комбинаций знаков.
* На каждом шаге текущий элемент массива либо прибавляется, либо вычитается из текущей суммы, после чего рекурсия вызывается для следующего элемента.
* Рекурсия продолжается до тех пор, пока не будут обработаны все элементы массива. Если итоговая сумма совпадает с целевой, формируется строка с выражением.

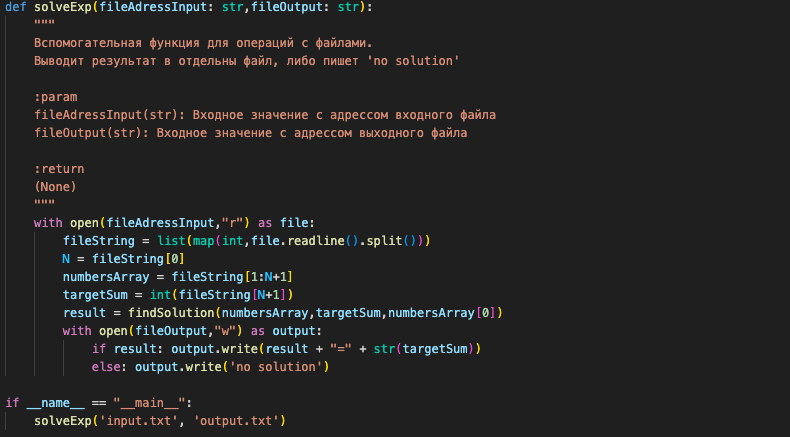
##### Разбиение алгоритма на функции:

1. **Рекурсивная функция findSolution**:
   * Принимает массив чисел, целевую сумму, текущее выражение, текущий индекс и текущую сумму.
   * На каждом уровне рекурсии добавляет текущий элемент массива с разными знаками (“+” и “-”).
   * Возвращает строку с выражением, если оно удовлетворяет условию, иначе — None.
2. **Вспомогательная функция solveExp**:
   * Считывает входные данные из файла.
   * Инициализирует рекурсивную функцию.
   * Записывает результат в выходной файл.

## Демонстрация работы программы

## 

## Листинг кода



**4. Пример работы программы**

**Входной файл (input.txt):**

5 1 2 3 4 5 9

**Выходной файл (output.txt):**

1+2-3+4+5=9

#### 5. **Преимущества и ограничения подхода**

##### Преимущества:

* Используется чистый рекурсивный подход, соответствующий требованиям задачи.
* Алгоритм находит решение для любой корректной входной строки.
* Код легко масштабируется и поддерживается.

##### Ограничения:

* Рекурсивный подход может быть неэффективным для больших массивов (), так как количество комбинаций растёт экспоненциально ().
* Требуется, чтобы входные данные строго соответствовали заданному формату.

# ЗАКЛЮЧЕНИЕ

Реализованное решение эффективно выполняет задачу в заданных ограничениях. Использование рекурсии позволило учесть все возможные комбинации знаков и гарантировать корректность результата. Код может быть дополнительно оптимизирован для работы с большими массивами за счёт использования мемоизации или других методов динамического программирования.

# Список использованных источников

1. Робертс А. "Искусство программирования. Теория и практика". – М.: Диалектика, 2019.
2. Дасгупта С., Пападимитриу Х., Увазани У. "Алгоритмы. Построение и анализ". – М.: Вильямс, 2014.
3. Knuth D. E. "The Art of Computer Programming", Volumes 1-3. – Addison-Wesley, 1997.
4. Официальная документация Python: https://docs.python.org/3/
5. Статьи по теории графов и комбинаторике на платформе Habrahabr: https://habr.com/ru/