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: Башкатов Артемий Валерьевич группа 20121

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 5](#_Toc187676637)

[2.1.3. Листинг кода 5](#_Toc187676638)

[ЗАКЛЮЧЕНИЕ 7](#_Toc187676639)

[Список использованных источников 8](#_Toc187676640)

# 

# Цели и формулировка задачи

Цель работы: Научиться работать со следующими темами:

1. Работа с файлами,
2. Работа с несколькими функциями, а не только с main,
3. Реализации рекурсивной функции.

Формулировка задачи: Даны **N** целых чисел **X1, X2, …, XN**. Требуется расставить между ними знаки **+** и **-** так, чтобы значение получившегося выражения было равно заданному целому **S**.

Входные данные: считываются с файла через пробел, в следующем порядке:

N X1 X2 ... XN S  
**2 <= N <= 30**, **0 <= Xi <= 5\*107**, **-109 <= S <= 109**

Необходимо вывести в файл полученное равенство с расставленными знаками, либо **no solution**, если не найдено решения. Если найдено несколько решений, то необходимо вывести **любое**.

Пример:

Input:

3 2 3 4 3

Output:

2-3+4=3

# Результаты работы

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

Основная идея алгоритма была в том, чтобы программа искала нужное уравнение методом проб и ошибок с помощью рекурсивной функции, постоянно заходя в новую функцию, подставляя «+» или «-», таким образом, перебираются все возможные уравнения и их результаты, в итоге выводится то, которое дает нужную нам сумму в виде строчного формата данных.

## Ход работы

Функция read\_input: эта функция парсит входные данные из файла:

* filename — название файла из которого будем брать входные данные

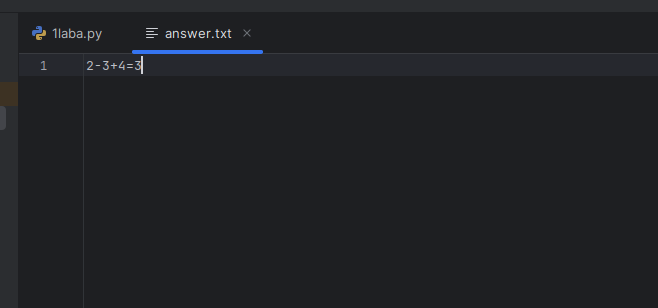
Функция solution: эта функция работает рекурсивно и принимает на вход:

* n, s и seq — количество чисел, число, которое нужно получить и сами числа между которыми нужно расставить знаки
* cur\_sum — текущая сумма
* cur\_eq (current equation) — текущее выражение
* idx — индекс с которого начинаем

В этой функции выход из рекурсии реализован в виде условия: если индекс равен длинне чисел между которыми нужно расставить знаки, то проверяется еще одно условие: совпадает ли текущая сумма со значением которое нам нужно, если не совпадает, то возвращает None.

Затем, функция вызывает саму себя со входным знаком «+», после этого с входным знаком «-»

## Демонстрация работы программы



## Листинг кода

def read\_input(filename: str) -> tuple:  
 with open("input.txt", 'r') as f:  
 inpt = list(map(int, f.readline().strip().split()))  
 return inpt  
   
def solution(n: int, s: int, seq: list, cur\_sum: int, cur\_eq="", idx = 0) -> str|None:  
 if idx == len(seq):  
 if cur\_sum == s:  
 return cur\_eq  
 else:  
 return None  
  
 next\_plus = solution(n, s, seq, cur\_sum+seq[idx], cur\_eq+f"+{seq[idx]}", idx+1)  
 if next\_plus is not None:  
 return next\_plus  
  
 next\_minus = solution(n, s, seq, cur\_sum-seq[idx], cur\_eq+f"-{seq[idx]}", idx+1)  
 if next\_minus is not None:  
 return next\_minus  
  
def main():  
 input = read\_input('input.txt')  
 n = input[0]  
 s = input[-1]  
 frst\_num = input[1]  
 seq = input[2:-1]  
 answ = solution(n, s, seq, frst\_num, cur\_eq=str(frst\_num))  
 if answ is None:  
 answ = 'no solution'  
 else:  
 answ += "=" + str(s)  
  
 with open("answer.txt", "w") as f:  
 f.write(answ)  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# ЗАКЛЮЧЕНИЕ

Были приобретены важные навыки работы с рекурсивными алгоритмами, которые действуют по методу проб и ошибок, и работы с файлами, а точнее запись и прочтение информации с файла.

# Список использованных источников