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:

Халилев Владислав Виленович

Проверил:

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

При работе над данной лабораторной работой будут затронуты следующие вопросы:

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

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**. Решение, только через рекурсию!

Пример:

Input:

3 2 3 4 3

Output:

2-3+4=3

# Результаты работы

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

## Ход работы

1. Считываем входные данные с файла input.
2. Создаём рекурсивную функцию, которая рекурсивно подставляет перебирает все возможные комбинации операций сложения и вычитания и ищет верное решение.

## Демонстрация работы программы

## 

<https://github.com/Cvif223/20121_Vlad_Khalilev.git>

## Листинг кода

f = open("Input.txt").readline()  
a = [int(x) for x in f.split(" ")]  
Xs = a[1:-1]  
N = a[0]  
S = a[-1]  
#print(Xs, N, S, sep="\n")  
def recursion\_comb(index = 1, final = f"{Xs[0]}", summ\_for\_now = Xs[0]):  
 if index == N:  
 if summ\_for\_now == S:  
 return final  
 return None  
  
 s1 = recursion\_comb(index + 1, final + " + " + str(Xs[index]), summ\_for\_now + Xs[index])  
 if s1 is not None:  
 #print(final, "1")  
 return s1  
  
 s2 = recursion\_comb(index + 1, final + " - " + str(Xs[index]), summ\_for\_now - Xs[index])  
 if s2 is not None:  
 #print(final, "2")  
 return s2  
  
print(recursion\_comb())

# ЗАКЛЮЧЕНИЕ

По результатам тестирования программы, поставленные задачи выполнены, все цели достигнуты.

# Список использованных источников

Конспекты лекций с занятий

https://stepik.org/course/195619/syllabus