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»

Выполнила: Подпорина Мария Тимофеевна группа 20121

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 5](#_Toc187676638)

[ЗАКЛЮЧЕНИЕ 7](#_Toc187676639)

[Список использованных источников 8](#_Toc187676640)

# Цели и формулировка задачи

Цели:

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.

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**. Решение, только через рекурсию!

# Результаты работы

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

## Ход работы

1. Чтение данных из файла и подготовка.

Программа открывает файл и считывает строку, содержащую числа. Считываемая строка разбивается на отдельные элементы, которые преобразуются в целые числа.

2. Определение рекурсивной функции.

Реализуется функция «combinations», которая принимает список чисел, нужную сумму, текущий индекс, текущую сумму и строку с операциями. Эта функция рекурсивно исследует все возможные комбинации чисел, применяя к ним операции сложения и вычитания. Если индекс достигает конца списка чисел, функция проверяет, равна ли текущая сумма нужной. Если да, то формируется строка с выражением и возвращается как результат. В противном случае функция продолжает рекурсивно вызывать себя для следующего числа, добавляя к текущей сумме или вычитая его.

3. Запись результатов в файл.

После выполнения функции «combinations», проверяется наличие решения. Если решение найдено, оно записывается в файл. Если же решения нет, в файл добавляется сообщение "no solution".

Программа завершает свою работу после записи результата в файл.

## Демонстрация работы программы

1. Создайте текстовый файл laba1.txt с содержимым:

N X1 X2 ... XN S

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

1. При запуске программы происходит следующее:

Файл laba1.txt открывается для чтения. Программа считывает первую строку и разбивает её на отдельные элементы, преобразуя их в список целых чисел. Программа запускает функцию «combinations», которая рекурсивно исследует все возможные комбинации чисел с операциями сложения и вычитания.

1. После завершения поиска программа проверяет, была ли найдена комбинация, соответствующая нужной сумме. После выполнения программы записывает нужную комбинацию обратно в файл laba1.txt. Можно изменять значения в файле laba1.txt для проверки других случаев.

## Листинг кода

with open('laba1.txt', 'r') as f:  
 sym=f.readline().split()  
 ''' Открытие файла и считывание значений '''  
r=[int(i) for i in sym]  
n=r[0]  
s=r[-1]  
r.pop(0)  
r.pop(-1)  
''' Присвоение значений переменным '''  
  
def combinations(numbers: list[int],goal:int,index:int,summ:int,symbols:str):  
 *''' Функция исследует все возможные комбинации чисел с операциями сложения и вычитания '''* if index==len(numbers):  
 if summ==goal:  
 return symbols+'='+str(goal)  
 return None  
 plus=combinations(numbers,goal, index+1, summ + numbers[index], symbols+'+'+str(numbers[index]))  
 if plus:  
 return plus  
 minus=combinations(numbers, goal, index+1, summ-numbers[index], symbols+'-'+str(numbers[index]))  
 if minus:  
 return minus  
 return None  
  
o=combinations(r,s,0,0,'')  
''' Приводит в действие функцию '''  
if o:  
 if o[0]=='+':  
 ''' Записывает решение в файл '''  
 with open('laba1.txt', 'a') as f:  
 f.write(o[1:]+'\n')  
 else:  
 with open('laba1.txt', 'a') as f:  
 f.write(o+'\n')  
else:  
 with open('laba1.txt', 'a') as f:  
 f.write('no solution' + '\n')

# ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы была разработана программа, которая эффективно находит все возможные комбинации чисел с операциями сложения и вычитания, соответствующие заданному значению. Программа демонстрирует использование функционального программирования и рекурсии для решения задачи.

# Список использованных источников

1. Учебные материалы и лекции.
2. Документация Python.