Министерство науки и высшего образования Российской Федерации

(Минобрнауки)

Санкт-Петербургский Государственный Морской Технический Университет

(СПбГМТУ)

Работу выполнил

Студент СПбГМТУ

|  |
| --- |
| Мочкин Владимир Олегович |
| Подпись (Дата)  « » |

ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Лабораторная работа

по теме:

«Рекурсивные функции. Лабораторная работа №1»

Научную работу проверил

|  |
| --- |
| Поделенюк Павел Петрович |
| Подпись (Дата)  « » |

# Содержание

[Содержание 2](#_Toc124800337)

[Список исполнителей 3](#_Toc124800338)

[Цель работы 4](#_Toc124800339)

[Формулировка задания 4](#_Toc124800340)

[Ход работы 5](#_Toc124800341)

[Приложение А 6](#_Toc124800342)

[Листинг кода: 6](#_Toc124800343)

# Список исполнителей

|  |  |  |  |
| --- | --- | --- | --- |
| Студент СПбГМТУ | |  | | --- | |  |   Подпись, дата | Мочкин Владимир Олегович, единоличный исполнитель лабораторной работы |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Цель работы

Ключевые слова: РЕКУСРИЯ, ФАЙЛЫ, PYTHON, ФУНКЦИЯ, ДРЕВО РЕШЕНИЙ

* Базовая работа с файлами. Запись и чтение
* Создание функций вне тела main, главной программы
* Создать древовидную рекурсивную функцию

# Формулировка задания

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

# Ход работы

Составные элементы:

* Файл ввода: Lab\_1.txt
* Файл вывода: Lab\_1\_sol.txt
* Основной скрипт на Python: laba1.py
  + Функция: main
  + Функция: calc

Функция **main** не принимает никаких данных и не возвращает, является основным телом программы

Функция **calc**(m: list, idx: int, st: str, summ: int) принимает на вход переменные: idx (индекс), st (пустая строка для записи), summ (переменная для записи текущей суммы), m (входящие данные)

Основная идея программы – древовидное решение, перебор каждого возможного варианта расположения знаков. Реализуется древо решению в моей программе посредствам рекурсивной функции **calc**(m: list, idx: int, st: str, summ: int), а именно последовательным вызовом функции с передачей ей знака «+» или «−».

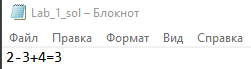
Функция углубляется, расставляя знаки и передавая получившуюся сумму в следующий вызов, когда же достигается базовый случай (количество расставленных знаков равняется: количеству чисел − 1) внутри функции происходит проверка равенства суммы (sm) и искомого числа (m), если проверка удачна функция возвращает накопленные ею на каждом уровне знаки

В главном теле скрипта, в функции main происходит считывание входных данных из файла Lab\_1.txt, вызов рекурсивной функции согласно полученным данным и запись вывода в файле Lab\_1\_sol.txt

Листинг скрипта: см. Приложение А

# Результат работы

По завершению работы программы в файл вывода в одну строку печатается получившиеся выражение с расставленными знаками + и – (см. Изображение экрана 1)



Изображение экрана 1

Программа не выводит ничего в консоль по ходу своего выполнения



# Приложение А

Листинг кода:

**def** **main**():

f = open('Lab\_1.txt', 'r') # открыли файл

m = list(map(int, f.read().split())) # список значений

#print(m)

f.close() # закрыли файл

**global** sln

sln= False # пока нет решения

calc(**1**,'',**0**,m)#расставляем знаки

**def** **calc**(idx, st, summ, m):

**global** sln

**if** **not** sln: # решение пока не найдено

# если первое число только сложение

**for** i **in** range(**2** **if** idx > **1** **else** **1**):

**if** sln: # решение найдено

**break**; # прекращаем расчет

**if** i: # вычитание

sm = summ - m[idx] # текущая сумма

s = st + '-' + str(m[idx]) # добавили в строку

**else**: # сложение

sm = summ + m[idx] # текущая сумма

s = st + ('+' **if** idx > **1** **else** '') + str(m[idx]) # добавили в строку

**if** idx < m[**0**]: # не все числа выбраны

calc(idx + **1**, s, sm, m) # идем к следующему

**else**:

**if** sm == m[len(m) - **1**]: # решение устраивает

**print**(s + '=' + str(m[len(m) - **1**])) # вывели строку

f = open('Lab\_1\_sol.txt', 'w') # создали файл вывода

f.write(s + '=' + str(m[len(m) - **1**])) # записали в него строку

f.close() # закрыли файл

**print**('end')

sln = True # решение существует

**if** idx == **1** **and** **not** sln: # если первое число и решения нет

**print**('no solution')

f = open('Lab\_1\_sol.txt', 'w')

f.write('no solution') # записали в файл

f.close()

**print**('end')

**if** \_\_name\_\_ == "\_\_main\_\_":

main()