

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный морской технический университет»
(СПбГМТУ)

Факультет цифровых промышленных технологий
Направление подготовки 09.03.01.03 "Интеллектуальные технологии
киберфизических систем"

Лабораторная работа №1
по дисциплине “Программирование”

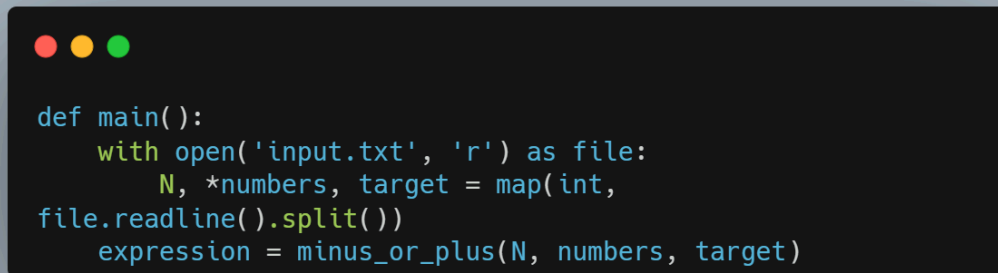
Студент 1 курса группы 20121
Очного отделения
Хохлов Д.Р.
Проверил:
Поделенюк П. П.

Цель работы - Составить на языке программирования Python программу, использующую рекурсивный метод, для установки знаков '+' и '-' между целыми числами таким образом, чтобы итоговое математическое выражение равнялось заранее заданной сумме.

Ход работы

Чтение данных:

- Программа читает данные из файла 'input.txt'.
- Извлекает первую строку из файла и записывает список целых чисел в переменную numbers .
- Извлекает количество чисел и заданную сумму из начала и конца списка и записывает в переменную N и target.
- После запускает функцию-рекурсию с полученными данными



```
def main():  
    with open('input.txt', 'r') as file:  
        N, *numbers, target = map(int,  
file.readline().split())  
        expression = minus_or_plus(N, numbers, target)
```

Основной код:

- Функция minus_or_plus() рекурсивно проверяет различные комбинации знаков для чисел.
- На каждом шаге рекурсии формируются два варианта: с использованием "+" и "-" перед текущим числом.
- При достижении последнего числа (index == N), проверяется, равна ли текущая сумма целевой сумме.
- Если равна, возвращается текущее выражение, иначе возвращается None.
- Если хотя бы один из рекурсивных вызовов возвращает заполненную строку, то эта строка передается далее по рекурсии и в итоге возвращается в главную программу.

```
def minus_or_plus(N:int, numbers:list[int,...], target:int, current_sum:int=0, expression:str='', index:int=0):
    if index == N:
        return expression if current_sum == target else None

    expression_plus = minus_or_plus(N, numbers, target, current_sum + numbers[index], expression + '+' + str(numbers[index]), index + 1)
    expression_minus = minus_or_plus(N, numbers, target, current_sum - numbers[index], expression + '-' + str(numbers[index]), index + 1)

    return expression_plus or expression_minus
```

Запись данных:

- Программа записывает полученные данные в отдельный файл “output.txt”(если данные отсутствуют, то записывается “no solution”)

```
with open('output.txt', 'w') as file:
    file.write(expression + '=' + str(target) if expression else 'no solution')
```

Запуск кода:

- Через условие запуска, запускается функция main()

```
if __name__ == "__main__":
    main()
```

Результат работы

В пример работы кода, в файл input.txt запишем значения:

1	3 2 5 3 10

В итоге, в файле output.txt, программа выдаст нам результат:

1	+2+5+3=10

Вывод

Таким образом, и работает программа, построенная на рекурсии для расстановки знаков между целыми числами, написанная на Python.

Весь код:

```
def minus_or_plus(N:int, numbers:list[int,...], target:int, current_sum:int=0, expression:str='', index:int=0):
    if index == N:
        return expression if current_sum == target else None

    expression_plus = minus_or_plus(N, numbers, target, current_sum + numbers[index], expression + '+' + str(numbers[index]), index + 1)
    expression_minus = minus_or_plus(N, numbers, target, current_sum - numbers[index], expression + '-' + str(numbers[index]), index + 1)

    return expression_plus or expression_minus

def main():
    with open('input.txt', 'r') as file:
        N, *numbers, target = map(int, file.readline().split())

    expression = minus_or_plus(N, numbers, target)

    with open('output.txt', 'w') as file:
        file.write(expression + '=' + str(target) if expression else 'no solution')

if __name__ == "__main__":
    main()
```