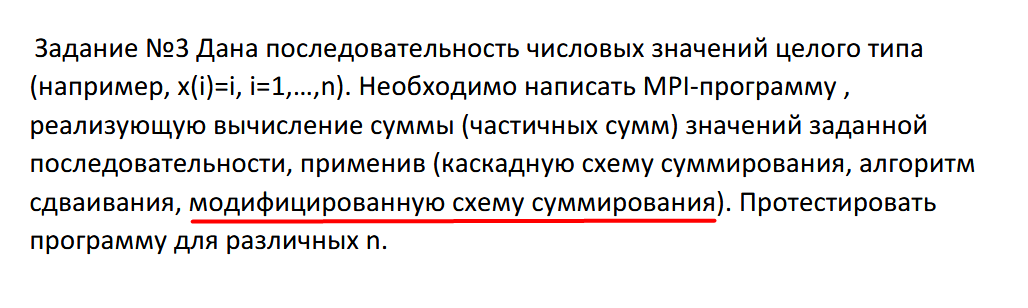
Контрольная №3

Задача на модифицированную каскадную схему суммирования

Боязитов Вадим, 932125

01.05.2025

## Задание 1



### Алгоритм

1.Инициализация и проверка входных данных

Инициализировать MPI-окружение.

Проверить корректность входного параметра n (положительное целое число).

Убедиться, что n является степенью двойки (для упрощения реализации).

Вычислить количество групп

Проверить, что количество процессов MPI соответствует num\_groups.

2. Распределение данных и локальное суммирование

Каждому процессу назначить группу из log₂n элементов.

Инициализировать данные в группе (например, x(i) = i).

Вычислить локальную сумму элементов внутри группы последовательным методом.

3. Каскадное суммирование результатов групп

Организовать иерархическое объединение частичных сумм:

Процессы с четными рангами:

Принимают данные от процесса с рангом current\_rank + step.

Проверяют, что ранг отправителя существует (current\_rank + step < total\_processes).

Суммируют полученное значение с локальным результатом

Процессы с нечетными рангами:

Отправляют свою сумму процессу с рангом current\_rank - step.

Проверяют, что ранг получателя существует (current\_rank - step >= 0).

Завершают участие в дальнейших вычислениях.

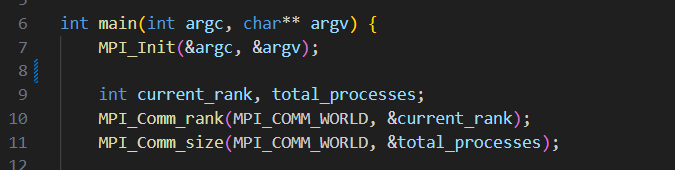
Повторять до полного объединения всех результатов.

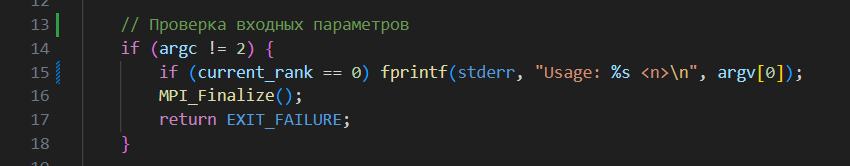
Вывести итоговую сумму на процессе-координаторе (ранг 0).

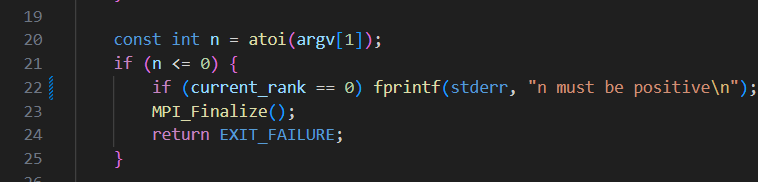
### Программная реализация

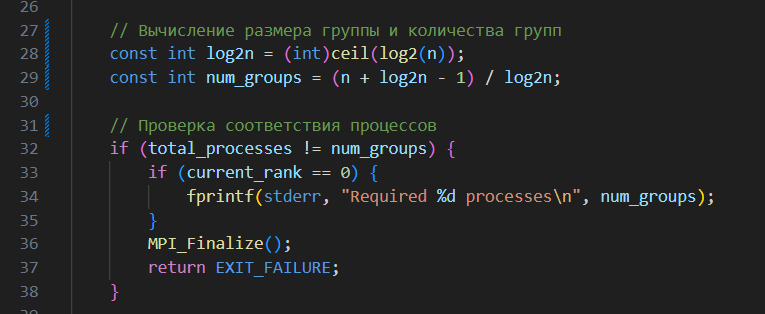
Окружение Linix. Язык C.

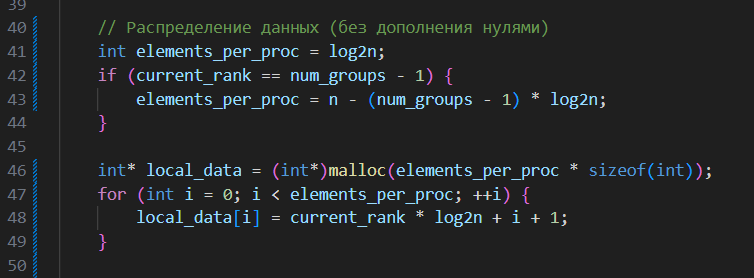
<https://github.com/katet3/tsu_parallel_programming/tree/main/lab_3>

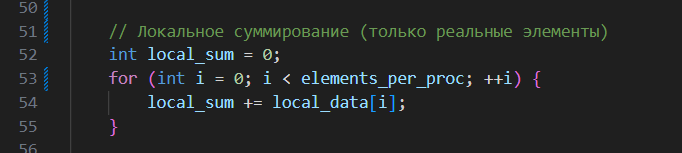


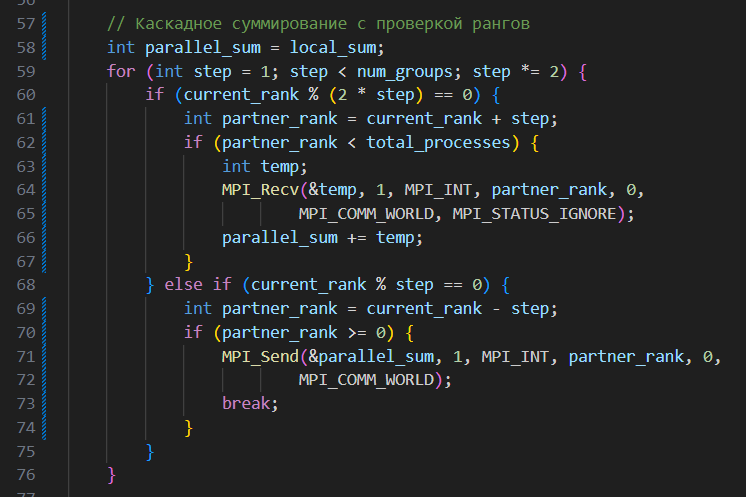


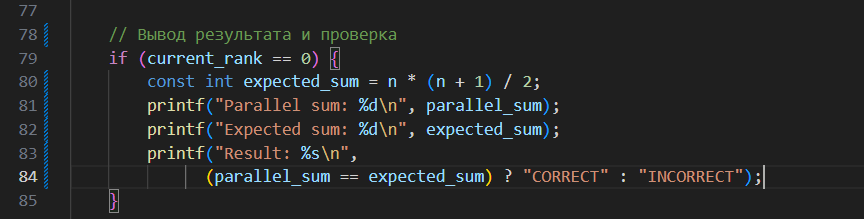


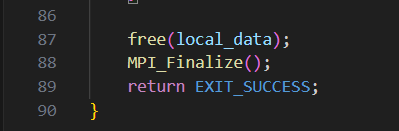




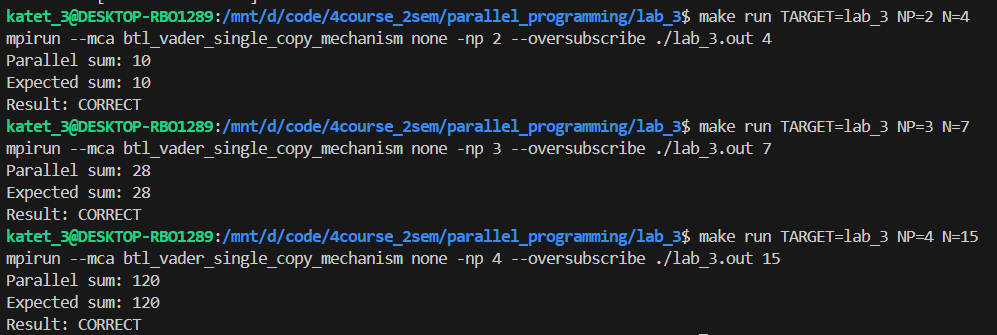


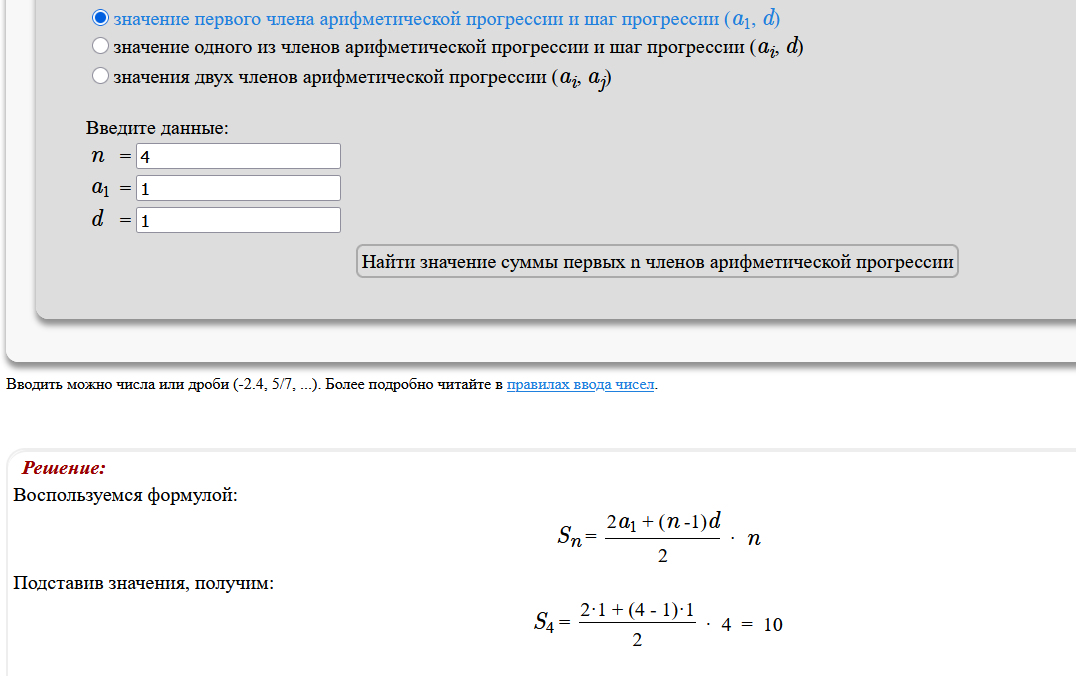


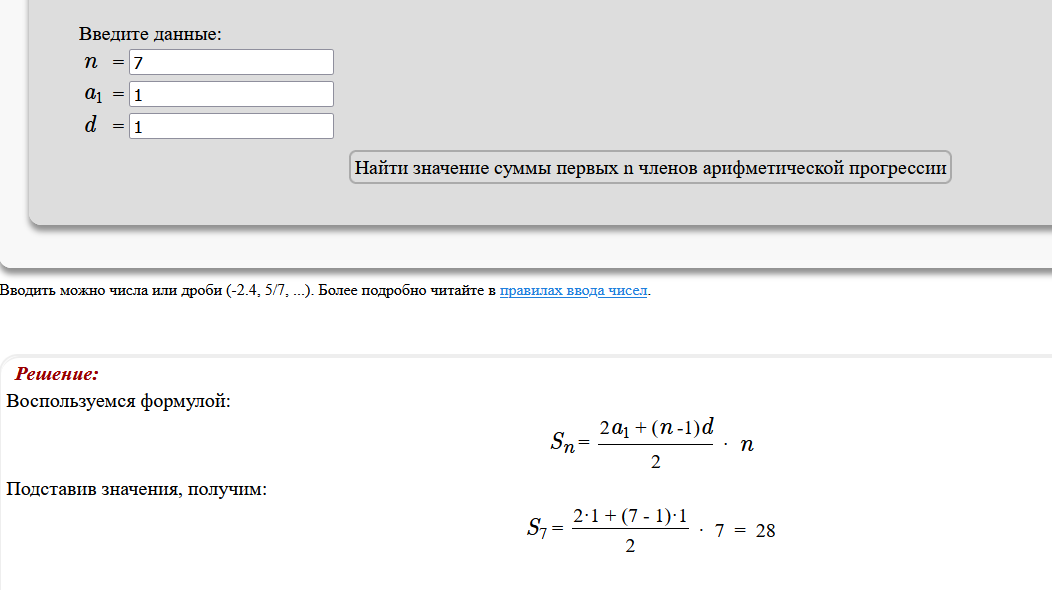


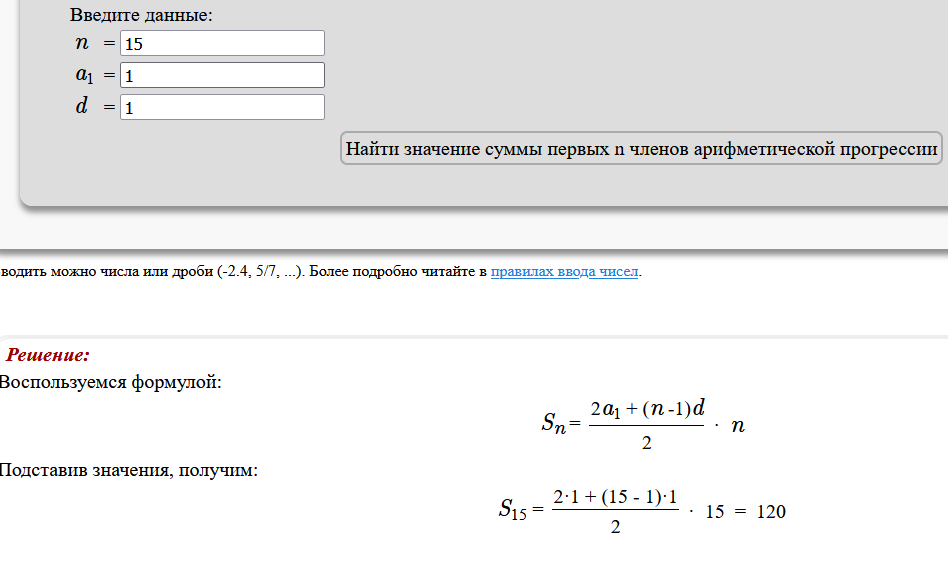


### Тестирование









### Вывод

Алгоритм реализован