temp.rs

```rust
async fn task_a(
    mut receiver: UnboundedReceiver<Data>,
    send_to_b: UnboundedSender<Data>,
    _send_to_c: UnboundedSender<Data>,
    send_to_d: UnboundedSender<Data>,
) {
    let initial_data = receiver.recv().await.unwrap();
    send_to_b.send(initial_data).unwrap();
    send_to_d.send(initial_data).unwrap();

    let intermediate_data = receiver.recv().await.unwrap();
    let result = (intermediate_data + initial_data) * 0.7;
    send_to_b.send(result).unwrap();
}
async fn task_b(
    mut receiver: UnboundedReceiver<Data>,
    send_to_a: UnboundedSender<Data>,
    send_to_c: UnboundedSender<Data>,
    _send_to_d: UnboundedSender<Data>,
) {
    let initial_data = receiver.recv().await.unwrap();

    let intermediate_data = initial_data.powi(3);
    send_to_a.send(intermediate_data).unwrap();
    send_to_c.send(intermediate_data).unwrap();

    let first_summand = receiver.recv().await.unwrap();
    let second_summand = receiver.recv().await.unwrap();
    let third_summand = receiver.recv().await.unwrap();
    let result = first_summand + second_summand + third_summand;
    println!("[B] BINGO: {result}");
}
async fn task_c(
    mut receiver: UnboundedReceiver<Data>,
    mut file: UnboundedReceiver<Data>,
    _send_to_a: UnboundedSender<Data>,
    send_to_b: UnboundedSender<Data>,
    send_to_d: UnboundedSender<Data>,
) {
    let initial_data = receiver.recv().await.unwrap();

    let intermediate_data = initial_data.powi(3);
    send_to_d.send(intermediate_data).unwrap();

    let file_content = file.recv().await.unwrap();
    send_to_b.send(file_content).unwrap();
}
async fn task_d(
    mut receiver: UnboundedReceiver<Data>,
    _send_to_a: UnboundedSender<Data>,
    send_to_b: UnboundedSender<Data>,
    _send_to_c: UnboundedSender<Data>,
) {
    let first_input = receiver.recv().await.unwrap();
    let second_input = receiver.recv().await.unwrap();

    let result = (first_input * second_input) / (first_input + second_input);
    send_to_b.send(result).unwrap();
}

```