Network System Programming Quiz 5
2025/12/9 Tuesday

1. Implement a multi-process TCP application where four processes (one Router server and three Clients on the same system) communicate. Step 1: Establish persistent TCP connections from each Client (IDs 1, 2, 3) to the Router upon initialization. Step 2: Define a data structure for communication, including the sender's ID, the intended receiver's ID, and the message content. Step 3: Program the Router to facilitate message passing between any two specific clients. Step 4: Add a broadcast logic such that using a receiver ID of 0 forwards the message to all three connected clients. Verify the final output matches the provided sample.

```
$ ./a.out
 Please input the sender and the receiver: 1 2
 Node 2 receives "1 sends to 2"
 Please input the sender and the receiver: 1 0
 Node 2 receives "1 sends to all"
 Node 1 receives "1 sends to all"
 Node 3 receives "1 sends to all"
```

2. Continue Problem 1. Clients 1 and 2 must implement a high-throughput message distribution scheme to share all received Router data with N common child processes. The implementation must specifically address the challenge posed by a large number of children (N, e.g., 50) and a high message rate (small interarrival time t, e.g., 0.5 seconds), which will be provided as command-line arguments. Each message payload should be a simple random integer between 1 and 9. The final command-line argument defines the total number of messages each client will share. The output requirement is that each child process must compute and print the cumulative sum of all integers it successfully receives.

```
$ ./a.out 50 0.5 30    //50 clients, message arrival rate 0.5 seconds,
                       //totally 30 integers from each client
```