Pseudocode:

Exercise: Write the pseudo-code of **Primary-backup**

Assumption:
- Perfect Failure Detector
- Leader Election available at all processes (primary, backups and clients)
- Reliable Broadcast available between replicas (primary and backups)
- Perfect Point to Point Link between any pair of processes (primary, backups and clients)

Client Code

Init
$waiting_i$ = false
$pending_i$ = empty
$primary_i$ = r0

when operation start op starts
    if $waiting_i$
        $pending_i = pending_i$ U {op}
    else
        $waiting_i$ = true
        trigger pp2pSend (REQ, op, ts, ci) to $primary_i$

upon event pp2pDeliver(OP_COMPLETED, op
    waiting = false
    if pending != empty
        op = select_form(pending)
        trigger pp2pSend(REQ, op, ci) to $primary_i$

upon event leader(rj)
    primary = rj
    trigger pp2pSend(REQ, op, ci) to $primary_i$

Primary Code

Init

$busy_i$ = false
$pending_p$ = empty
ack_p = empty
$backups_p$ = {r1, r2, …rn}
state = default_value
running = null
%remember which is the last executed operation

upon event pp2Send(REQ, op, ts, ci) from cj
    if $busy_p$
        $pending_p = pending_p$ U {<op, cj>}

    else
        $busy_p$ = true
        $state_p$ = execute(op)
        running = <op, cj>
        trigger RBcast(UPDATE, $state_p$)

Upon event pp2pDeliver(ACK, rj)
    $ack_p = ack_p$ U {rj}

when backups is contains in ack_p
    trigger pp2pSend(OP_COMPLETED, running.op) to running.c
    pending = pending / running
    running = null
    ack = empty
    busy = false

when pending != empty and not busy
    % Select next operation

upon event crash(rj)
    backups = backups \ {rj}

Backup Code

Init

$state_i$ = default_value

$primary_i$ = r0

Upon event RBDeliver(UPDATE, state)

$state_i$ = state

trigger pp2pSend(ACK, rj) to primary