```
# Test the MIPS processor.
# add, sub, and, or, slt, addi, lw, sw, beq, j
# If successful, it should write the value 7 to address 84

#         Assembly          Description                 Address    Machine
main:     addi $2, $0, 5    # initialize $2 = 5         0          20020005        20020005
          addi $3, $0, 12   # initialize $3 = 12        4          2003000c        2003000c
          addi $7, $3, -9   # initialize $7 = 3         8          2067fff7        2067fff7
          or   $4, $7, $2   # $4 <= 3 or 5 = 7          c          00e22025        00e22025
          and  $5, $3, $4   # $5 <= 12 and 7 = 4        10         00642824        00642824
          add  $5, $5, $4   # $5 = 4 + 7 = 11           14         00a42820        00a42820
          beq  $5, $7, end  # shouldn't be taken        18         10a7000a        10a7000a
          slt  $4, $3, $4   # $4 = 12 < 7 = 0           1c         0064202a        0064202a
          beq  $4, $0, around # should be taken         20         10800001        10800001
          addi $5, $0, 0    # shouldn't happen          24         20050000        20050000
around:   slt  $4, $7, $2   # $4 = 3 < 5 = 1            28         00e2202a        00e2202a
          add  $7, $4, $5   # $7 = 1 + 11 = 12          2c         00853820        00853820
          sub  $7, $7, $2   # $7 = 12 - 5 = 7           30         00e23822        00e23822
          sw   $7, 68($3)   # [80] = 7                  34         ac670044        ac670044
          lw   $2, 80($0)   # $2 = [80] = 7             38         8c020050        8c020050
          j    end          # should be taken           3c         08000011        08000011
          addi $2, $0, 1    # shouldn't happen          40         20020001        20020001
end:      sw   $2, 84($0)   # write adr 84 = 7          44         ac020054        ac020054
```