```
main:    addi $s0, $zero, 48       #set base address
         lw $s4, 0($s0)            #load input
         addi $a3, $s4, 0          #transfer input to function parameter
         addi $a0, $zero, 1        #previous value initialized to 1
         addi $s1, $zero, 0        #clear register
loop:    sw $a3, 4($s0)            #display value
         jal EVEN                  # test even or odd passing parameter a3
         bne $v0, $s1, case_odd    # branch if even case failed
         j case_even               #jump to even case
EVEN:    andi $v0, $a3, 1          # return result of test in $vo
         jr $ra                    #return
case_odd:  addi $a0, $a3, 0        #update old value
         add $a3, $a3, $a3         #update current value odd case
         j loop                    #return to loop
case_even:  add $a3, $a3, $a0      #update current value even case
         j loop                    #return to loop
```