```
#############################################################################
# Noah del Angel, CS 2318-002, Assignment 2 Part 1 Program B
#############################################################################
# 1st finds & shows position-weight of the rightmost 1 of a non-0 integer,
# then finds & shows the resulting value when that rightmost 1 is cleared.
############################ data segment ###################################
        .data
inPrompt .asciiz "Enter a non-zero integer: "
outLab1: .asciiz " has rightmost 1 @ weight position "
outLab2: .asciiz "\nClearing the rightmost 1 makes it "
############################ code segment ###################################
        .text
        .globl main
main:
        li $v0, 4
        la $a0, inPrompt
        syscall                 # print input prompt

        li $v0, 5
        syscall                 # read input integer x

        ###########################################################
        # Replace each "hole" indicated with "********" with an
        # an instruction so that the program will work just like
        # the sample runs shown at the bottom.
        # The last 3 instructions (replacing the last 3 "holes")
        # MUST involve bitwise operations.
        # Your completed program will be tested for AT LEAST the
        # test cases shown (so be sure to at least test them).
        ###########################################################

        move $t0, $v0           # $t0 gets copy of input x
        sub $t1, $zero, $t0     # $t1 gets mask1 that is "-x"

        li $v0, 1
        move $a0, $t0
        syscall

        li $v0, 4
        la $a0, outLab1
        syscall                 # print output label 1
        li $v0, 1
        and $a0, $t0, $t1       # $a0 gets "all bits of x cleared except the rightmost 1"
        syscall

        xori $t2,$a0,0xFFFFFFFF  # $t2 gets mask2 that is "$a0 with all its bits toggled"

        li $v0, 4
        la $a0, outLab2
        syscall                 # print output label 2
        li $v0, 1
        and $a0, $t0, $t2       # $a0 gets "all bits of x with the rightmost 1 cleared"
```

```
        syscall

        li $v0, 10              # exit
        syscall

########################## sample test runs ##############################
# Enter a non-zero integer: 1
# 1 has rightmost 1 @ weight position 1
# Clearing the rightmost 1 renders it 0
# -- program is finished running --
#
#
# Reset: reset completed.
# Enter a non-zero integer: -1
# -1 has rightmost 1 @ weight position 1
# Clearing the rightmost 1 makes it -2
# -- program is finished running --
#
#
# Reset: reset completed.
# Enter a non-zero integer: 3456
# 3456 has rightmost 1 @ weight position 128
# Clearing the rightmost 1 makes it 3328
# -- program is finished running --
#
#
# Reset: reset completed.
# Enter a non-zero integer: -123456
# -123456 has rightmost 1 @ weight position 64
# Clearing the rightmost 1 makes it -123520
# -- program is finished running --
#
#
# Reset: reset completed.
# Enter a non-zero integer: 1073741824
# 1073741824 has rightmost 1 @ weight position 1073741824
# Clearing the rightmost 1 makes it 0
# -- program is finished running --
#
#
# Reset: reset completed.
# Enter a non-zero integer: -2147483647
# -2147483647 has rightmost 1 @ weight position 1
# Clearing the rightmost 1 makes it -2147483648
# -- program is finished running --
######################## end sample test runs ############################
```