```
#####################################################################
# Robert Maldonado, CS 2318-003, Assignment 2 Part 1 Program B
#############################
# prompt user to enter an integer, read the integer, and display if the
# integer is of type 0 (positve-even), 1 (positive-odd), 2 (negative-even)
# or 3 (negative-odd)
############################ data segment ###################################
                        .data
typeLegend:             .asciiz "0 = P&E, 1 = P&O, 2 = N&E, 3 = N&O\n"
                                # P&E = Positive&Even, P&O = Positive&Odd,
                                # N&E = Negative&Even, N&O = Negative&Odd.
inputPrompt:            .asciiz "Enter your integer: "
outputLabel:            .asciiz "Your integer is of type "
############################ code segment ###################################
                        .text
                        .globl main
main:
                        li $v0, 4
                        la $a0, typeLegend
                        syscall                 # print type legend
                        la $a0, inputPrompt
                        syscall                 # print input prompt
                        li $v0, 5
                        syscall                 # read integer
                        move $v1, $v0           # save integer read in $v1
                        li $v0, 11
                        li $a0, '\n'
                        li $v0, 4
                        la $a0, outputLabel
                        syscall                 # print output label

                        li $v0, 1
                        li $a0, 0               # initialize desired output
to 0


#########################################################
                        # Insert NO MORE THAN 5 lines of code that involve
ONLY
                        #   bit manipulating instructions (ANDing, ORing,
XORing,
                        #   NORing and shifting - only whatever that are
needed)
                        # so that the program will work just like the sample
runs
                        # shown at the bottom (some blank lines edited out).
                        # HINT: Risking telling the obvious, the
instructions you
                        #       insert are related to bringing the value in
$a0
                        #       from the initial value of 0 to the final
desired
                        #       value (which should be either 0, 1, 2 or 3
when
                        #       printed as an integer).
```

```
                        # You should test your completed program for AT
LEAST the
                        # test cases shown (what evaluator will do when
grading).

#############################################################

                        # the andi finds out if it is even or odd
                        andi $a0, $v1, 1
                        # the andi finds if it is negative with the leftmost
bit
                        andi $a1, $v1, 0x80000000
                        # we then shift the leftmost bit 30 to the right
which
                        # makes it 2 for negative value
                        srl $a1, $a1, 30
                        # we then add the two to the even or odd value $a0
                        or $a0, $a0, $a1



                        syscall                 # display desired output


#############################################################

                        li $v0, 10              # exit gracefully
                        syscall

########################## sample test runs ################################
#  0 = P&E, 1 = P&O, 2 = N&E, 3 = N&O
#  Enter your integer: 2147483646
#  Your integer is of type 0
#  -- program is finished running --
#
#  Reset: reset completed.
#
#  0 = P&E, 1 = P&O, 2 = N&E, 3 = N&O
#  Enter your integer: 2147483647
#  Your integer is of type 1
#  -- program is finished running --
#
#  Reset: reset completed.
#
#  0 = P&E, 1 = P&O, 2 = N&E, 3 = N&O
#  Enter your integer: -2
#  Your integer is of type 2
#  -- program is finished running --
#
#  Reset: reset completed.
#
#  0 = P&E, 1 = P&O, 2 = N&E, 3 = N&O
#  Enter your integer: -1
#  Your integer is of type 3
#  -- program is finished running --
```

```
######################## end sample test runs ############################
```